

INSTYTUT BUDOWNICTWA  
POLITECHNIKI WROCŁAWSKIEJ

Raport serii PRE nr 5/08

*Analiza nowych zastosowań teorii szeregowania zadań  
w organizacji robót budowlanych*

Michał Podolski

Rozprawa doktorska

Promotor:

dr hab. inż. Zdzisław Hejducki

Słowa kluczowe:

organizacja robót budowlanych, planowanie robót budowlanych,  
optymalizacja, szeregowanie zadań, metaheurystyki

Wrocław, 2008 r.



## SPIS TREŚCI

1. Wstęp.....	7
1.1. Wprowadzenie w problematykę rozprawy .....	7
1.2. Przedmiot, cel, zakres, tezy pracy .....	10
1.3. Metodyka badań oraz wykorzystywane w pracy narzędzia i metody .....	12
2. Analiza stanu badań dotyczących metod harmonogramowania przedsięwzięć budowlanych z uwzględnieniem metod wykorzystujących system pracy potokowej .....	14
2.1. Aktualny stan badań dotyczących sieciowych metod harmonogramowania dla przedsięwzięć budowlanych .....	15
2.2. Aktualny stan badań dotyczących metod harmonogramowania wykorzystujących system pracy potokowej w przedsięwzięciach budowlanych .....	21
2.3. Podsumowanie.....	32
3. Analiza niektórych zagadnień teorii szeregowania zadań oraz metod ich rozwiązywania.....	34
3.1. Wybrane podstawowe pojęcia i definicje teorii szeregowania zadań .....	34
3.2. Problem przepływowy .....	37
3.3. Problem przepływowy z maszynami równoległymi .....	38
3.4. Metody rozwiązywania problemów szeregowania zadań .....	40
3.5. Aktualny stan badań dotyczący rozwiązywania problemów przepływowych .....	41
3.6. Zastosowania zagadnień teorii szeregowania zadań w budownictwie .....	43
3.7. Metody sprzężeń czasowych na tle teorii szeregowania zadań .....	44
3.8. Podsumowanie.....	47
4. Metaheurystyczne metody optymalizacji dyskretnej zastosowane w pracy oraz weryfikacja wyników obliczeń uzyskiwanych za ich pomocą .....	49
4.1. Metaheurystyczne metody optymalizacji dyskretnej zastosowane w pracy .....	49
4.1.1. Algorytm przeszukiwania z zabronieniami ( <i>taboo search</i> ).....	49
4.1.2. Algorytm przeszukiwania genetycznego ( <i>genetic search</i> ) .....	51
4.1.3. Algorytm symulowanego wyżarzania ( <i>simulated annealing</i> ) .....	52
4.1.4. Formy algorytmów metaheurystycznych zastosowanych w pracy.....	54
4.1.4.1. Formy algorytmów metaheurystycznych dla permutacyjnego problemu przepływowego .....	54
4.1.4.2. Formy algorytmów metaheurystycznych dla problemu przepływowego z maszynami równoległymi .....	58

4.2.	Weryfikacja wyników obliczeń uzyskiwanych za pomocą wybranych trzech algorytmów metaheurystycznych.....	61
4.2.1.	Weryfikacja wyników uzyskiwanych z zastosowaniem algorytmów metaheurystycznych dla permutacyjnego problemu przepływowego.....	63
4.2.2.	Weryfikacja wyników uzyskiwanych z zastosowaniem algorytmów metaheurystycznych dla problemu przepływowego z maszynami równoległymi.....	69
4.3.	Podsumowanie.....	75
5.	Nowe modele harmonogramowania przedsięwzięć wieloobektowych z zastosowaniem systemu pracy potokowej.....	78
5.1.	Nowe modele harmonogramowania przedsięwzięć wieloobektowych realizowanych w systemie pracy potokowej z relacjami kolejnościowymi między robotami wyrażonymi sekwencją.....	82
5.1.1.	Model systemu pracy potokowej z jedną grupą roboczą wykonującą robotę jednego rodzaju.....	83
5.1.2.	Model systemu pracy potokowej z wieloma grupami roboczymi wykonującymi robotę jednego rodzaju.....	86
5.1.3.	Model systemu pracy potokowej z relacjami kolejnościowymi między robotami wyrażonymi sekwencją i z uwzględnieniem kryteriów koszt/czas.....	89
5.2.	Nowe modele harmonogramowania przedsięwzięć wieloobektowych realizowanych w systemie pracy potokowej z relacjami kolejnościowymi między robotami danymi grafem.....	98
5.2.1.	Model systemu pracy potokowej z jedną grupą roboczą wykonującą robotę jednego rodzaju.....	101
5.2.2.	Model systemu pracy potokowej z wieloma grupami roboczymi wykonującymi robotę jednego rodzaju.....	104
5.2.3.	Model systemu pracy potokowej z relacjami kolejnościowymi między robotami danymi grafem i z uwzględnieniem kryteriów koszt/czas.....	107
5.3.	Podsumowanie.....	112
6.	Weryfikacja sformułowanych modeli harmonogramowania przedsięwzięć wieloobektowych wykorzystujących system pracy potokowej.....	115
6.1.	Harmonogramowanie robót budowlanych na przykładzie kompleksu budynków mieszkalnych.....	115

6.1.1. Przykład zastosowania modelu harmonogramowania przedstawionego w p. 5.1.1 .....	115
6.1.2. Przykład zastosowania modelu harmonogramowania przedstawionego w p. 5.1.2 .....	118
6.1.3. Przykład zastosowania modelu harmonogramowania przedstawionego w p. 5.1.3 .....	119
6.2. Harmonogramowanie robót budowlanych na przykładzie zespołu stacji paliw .....	126
6.2.1. Przykład zastosowania modelu harmonogramowania przedstawionego w p. 5.2.1 .....	127
6.2.2. Przykład zastosowania modelu harmonogramowania przedstawionego w p. 5.2.2 .....	129
6.2.3. Przykład zastosowania modelu harmonogramowania przedstawionego w p. 5.2.3 .....	131
6.3. Podsumowanie.....	139
7. Wnioski końcowe.....	141
Literatura.....	145
Załączniki.....	155



## **1. Wstęp**

### **1.1. Wprowadzenie w problematykę rozprawy**

Jednym z najważniejszych zagadnień analizowanych obecnie w budownictwie jest problematyka zarządzania realizacją przedsięwzięć budowlanych. W ramach niej badane są m.in.: problemy projektowania struktur organizacyjnych przedsięwzięć, projektowania obiektów budowlanych i procesów produkcyjnych w budownictwie [16], [68], [74], [85], [113]. W problemach zarządzania przedsięwzięciami budowlanymi istotne znaczenie mają zagadnienia ich organizacji, a szczególnie planowania. Dynamicznie zmieniająca się sytuacja gospodarcza powoduje, że powszechnie stosowane metody planowania wymagają udoskonalania, tak aby lepiej odzwierciedlały pojawiające się nowe potrzeby rynku. Koniunktura w budownictwie w Polsce przekłada się na stały wzrost PKB. Wzrost inwestycji jest obserwowany we wszystkich podstawowych gałęziach produkcji budowlanej, a szczególnie w infrastrukturze drogowej, miejskiej, ochronie środowiska. Jest to związane m.in. ze stałym dopływem środków z funduszy Unii Europejskiej. Z drugiej strony zauważalne są negatywne czynniki mające wpływ na organizację i koszt przedsięwzięć budowlanych. Do nich należą m.in.: znaczący wzrost kosztów pracy, wzrost cen materiałów budowlanych, energii, paliw, emigracja wielu wykształconych i wykwalifikowanych pracowników do krajów Unii Europejskiej. Sytuację pogarszył fakt ograniczonych zdolności produkcyjnych firm budowlanych, które zmniejszyły liczbę pracowników w latach poprzednich, w związku z dekoniunkturą. Wymienione wyżej czynniki, związane z obecną sytuacją ekonomiczną, powodują, że znaczenie optymalnego planowania przedsięwzięć budowlanych, które uwzględnia wszelkie możliwe ograniczenia realizacyjne, będzie ciągle wzrastać.

Planowanie jest integralną częścią przedsięwzięcia budowlanego. W ogólnym sensie jest ono procesem twórczym, który zmierza do określenia celów i ustalenia sposobów ich osiągnięcia [88]. Proces ten jest sekwencyjny. Na planowanie składają się kolejno: prognozowanie, programowanie i tworzenie planu. W przypadku planowania przedsięwzięcia budowlanego główną częścią tego procesu jest harmonogramowanie tj. utworzenie przez wykonawcę harmonogramu realizacji, w którym ustalone zostają terminy realizacji robót oraz uwzględniona jest przyjęta alokacja zasobów (kapitału, pracowników, sprzętu itd.). Utworzenie harmonogramu jest wynikiem przyjęcia założeń odnośnie do planu strategicznego i operacyjnego wykonawcy oraz przewidywanych ograniczeń realizacyjnych. Powinny być również uwzględnione ryzyko i niepewność związane z planowanym przedsięwzięciem, ograniczenia wynikające z budżetu oraz dyrektywnych terminów jego realizacji.

Harmonogramowanie zadań wchodzących w skład przedsięwzięcia jest zagadnieniem złożonym. Stworzono wiele metod, modeli i odpowiednich dla nich algorytmów postępowania dotyczących tego problemu. Dzięki użyciu programów komputerowych proces harmonogramowania może być znacząco ułatwiony i przyspieszony. Obecnie w dalszym ciągu są prowadzone badania nad tworzeniem takich metod harmonogramowania, które w najlepszy sposób odzwierciedlałyby rzeczywistość przedsięwzięć budowlanych i uwzględniały przyjęte kryteria.

W polskiej literaturze przedmiotu [68] zaznaczył się podział na dwie główne grupy modeli harmonogramowania przedsięwzięć budowlanych:

- dla przedsięwzięć typu kompleks operacji,
- dla przedsięwzięć, które mogą być realizowane zgodnie z tzw. zasadą pracy równomiernej (system pracy potokowej).

W literaturze zagranicznej, np. [83], [147], podział modeli harmonogramowania przedsięwzięć budowlanych jest następujący: dla przedsięwzięć z procesami o charakterze niepowtarzalnym (*nonrepetitive projects*) i dla przedsięwzięć z procesami o charakterze powtarzalnym (*repetitive projects*). Jest to podział analogiczny do przedstawionego w pracy [68].

W niniejszej rozprawie odniesiono się do systemu pracy potokowej, ze względu na wciąż niedoceniane zalety jego aplikacji podczas realizacji przedsięwzięć budowlanych. Ogólną zasadą harmonogramowania przedsięwzięć w systemie pracy potokowej jest podobieństwo planowania robót do przebiegu produkcji przemysłowej realizowanej z użyciem taśmy produkcyjnej [3], [68], [83], [88]. System pracy potokowej może być stosowany w organizacji przedsięwzięć budowlanych, w których jest możliwy podział inwestycji na części realizowane przez wyspecjalizowane grupy robocze (brygady, firmy, podwykonawcy). Częściami przedsięwzięcia najczęściej są wydzielone obiekty budowlane o takiej samej technologii lub działki robocze (fronty robót) realizowanego obiektu budowlanego. W rozprawie przyjęto, że przedsięwzięcia budowlane, które mogą być realizowane w tym systemie i szczególnie dla nich jest możliwe zastosowanie formułowanych modeli będą nazywane wieloobektowymi. Polegają one na realizacji kompleksów (zespołów) obiektów np. budynków mieszkalnych, handlowych lub przemysłowych.

Zaletami systemu pracy potokowej są m.in.: przejrzystość struktury harmonogramu robót, ścisłe określenie zadań dla realizatorów (grup roboczych), możliwy wzrost wydajności i jakości wykonywanych robót zgodnie z tzw. krzywą uczenia się, wykorzystywanie doświadczeń we współpracy (komunikacji) wykonawcy z inwestorem powstałych w trakcie realizacji przedsięwzięcia. Należy jednak również wymienić niedoskonałości



sformułowanych do tej pory modeli harmonogramowania wykorzystujących system pracy potokowej m.in.: brak możliwości uwzględnienia wielu dodatkowych ograniczeń występujących w praktyce, niedoskonałości algorytmów optymalizacji dyskretnej używanych obecnie podczas tworzenia harmonogramów np. najczęściej stosowanego algorytmu podziału i ograniczeń. Wady te można wyeliminować stosując m.in.: niektóre rozwiązania opisujące ograniczenia używane np. w modelach dla przedsięwzięć typu kompleks operacji oraz algorytmy optymalizacji dyskretnej stosowane obecnie w rozwiązywaniu współczesnych zagadnień szeregowania zadań. Stosowanie takich algorytmów umożliwi rozwiązywanie problemów harmonogramowania robót budowlanych z wieloma rodzajami ograniczeń i kryteriów.

Rozprawa jest podzielona na siedem rozdziałów. W pierwszym z nich przedstawiono krótkie wprowadzenie w problematykę rozprawy, przedmiot, cel, zakres, tezy pracy, jak również metodykę badań, wykorzystywane w pracy narzędzia i metody. W rozdziale drugim przeprowadzono analizę stanu badań w zakresie metod harmonogramowania dla przedsięwzięć typu kompleks operacji i metod stosujących system pracy potokowej. W rozdziale trzecim przedstawiono analizę wybranych zagadnień teorii szeregowania zadań, metod rozwiązywania tych zagadnień oraz zastosowań teorii szeregowania zadań w budownictwie. W rozdziale tym przedstawiono również identyfikację metod sprzężeń czasowych [94] wśród zagadnień teorii szeregowania zadań. W rozdziale czwartym przedstawiono trzy metaheurystyczne algorytmy optymalizacji dyskretnej stosowane w rozprawie. W rozdziale tym przedstawiono ponadto rezultaty badań weryfikujących rozwiązania uzyskiwane za pomocą tych algorytmów z zastosowaniem analizy eksperymentalnej. W rozdziale piątym zostały sformułowane nowe modele optymalizacyjne występujące w systemie pracy potokowej przedsięwzięć wieloobektowych. Zakłada się w nich występowanie relacji kolejnościowych między robotami w obiektach wyrażonych niezmienną dla każdego obiektu sekwencją wynikającą z ustalonego porządku technologicznego. Uogólnieniem tego założenia jest możliwość występowania relacji kolejnościowych między robotami, opisanych niezmiennym dla każdego obiektu grafem. W modelach zastosowano ponadto dodatkowe ograniczenia i kryteria, m.in. związane z uwzględnieniem kosztu wykonywania przedsięwzięcia. W rozdziale szóstym dokonano weryfikacji sformułowanych modeli przez zaprezentowanie przykładów ich aplikacji. Rozprawę zakończono wnioskami końcowymi zawierającymi podsumowanie rozprawy oraz propozycje dalszych kierunków badań (rozdział siódmy).

## 1.2. Przedmiot, cel, zakres, tezy pracy

W rozprawie przedstawiono i rozwiązano wybrane problemy planowania przedsięwzięć wieloobektowych w systemie pracy potokowej, przy założeniu deterministycznego charakteru parametrów występujących w przedsięwzięciu. W harmonogramowaniu przedsięwzięć, na które składa się realizacja obiektów budowlanych o charakterze jednorodnym lub jednotypowym (o takich samych lub proporcjonalnych czasach trwania robót jednego rodzaju dla każdego z obiektów), można zastosować metodę pracy równomiernej [22], [68]. Dla tego rodzaju przedsięwzięć najczęściej nie występuje potrzeba ustalenia optymalnego harmonogramu ze względu na narzucone kryteria (np. przy przyjęciu kryterium czasu trwania przedsięwzięcia i założeniu zastosowania niezależnych wykonawców dla realizacji poszczególnych robót). Problem ustalania optymalnego harmonogramu występuje podczas planowania przedsięwzięć związanych z realizacją obiektów niejednorodnych (o dowolnych czasach trwania robót dla obiektów) [68]. Wiąże się to z przyjęciem odpowiedniej kolejności realizacji obiektów dla założonych kryteriów np. czasu trwania przedsięwzięcia lub jego kosztu. Tego rodzaju przedsięwzięcia budowlane są przedmiotem rozprawy.

Obecnie w badaniach nad harmonogramowaniem pracy w systemie potokowym w budownictwie najczęściej przyjmowane są jako kryteria: czas wykonywania robót we wszystkich obiektach [54], [94] lub koszt rozumiany jako kary np. za niedotrzymanie terminów dyrektywnych zakończenia robót w obiektach i przestoje w pracy grup roboczych [88]. Brak jest natomiast badań dotyczących zastosowania kryterium kosztu całego przedsięwzięcia rozumianego jako suma kosztów realizacji poszczególnych robót wykonywanych przez grupy robocze. Kryterium to zostało zastosowane w niniejszej rozprawie.

Aktualnie w badaniach na temat przedsięwzięć budowlanych realizowanych z zastosowaniem systemu pracy potokowej przyjmuje się najczęściej relacje kolejnościowe między robotami wyrażone sekwencją wynikającą z przyjętego porządku technologicznego [54], [88]. Założenie to dla części modeli formułowanych w rozprawie zostało utrzymane. Ponadto rozpatrzono modele przedsięwzięć, w których relacje kolejnościowe między robotami obiektu są dane grafem (podobnie jak w przedsięwzięciach typu kompleks operacji). Wprowadza się odmienne od dotychczasowych założenia dotyczące grup roboczych stosowanych w przedsięwzięciu polegające na umożliwieniu wykonywania robót jednego rodzaju przez określoną ilość grup roboczych o takich samych albo różnych składach lub wydajnościach. Sformułowanie modeli uwzględniających opisane ograniczenia i kryteria,

poprzedzone analizą badań nad problemami organizacji i planowania przedsięwzięć budowlanych, będzie przedmiotem niniejszej rozprawy.

Utworzenie odpowiednich modeli dla przedsięwzięć wieloobektowych z przyjętą w rozprawie strukturą zasobów, ograniczeniami dotyczącymi kolejności realizacji robót w obiekcie, rozważenie wielu kryteriów optymalizacji umożliwiają aktualne osiągnięcia teorii szeregowania zadań. Wiąże się to z zastosowaniem odpowiednich ograniczeń i zmiennych decyzyjnych. W niniejszej rozprawie dla rozwiązania problemów optymalizacyjnych, ujętych w opisanych powyżej modelach, zastosowano narzędzia używane obecnie w teorii szeregowania zadań. Do nich należą metaheurystyczne algorytmy optymalizacji dyskretnej wykorzystujące zasady tzw. sztucznej inteligencji [134].

Celem naukowym niniejszej rozprawy jest identyfikacja i przedstawienie nowych modeli planowania realizacji przedsięwzięć budowlanych stosujących system pracy potokowej oraz podanie sposobów rozwiązywania zadań optymalizacji dyskretnej, które występują w tych modelach. Jest to możliwe dzięki analizie zastosowań aktualnych osiągnięć teorii szeregowania zadań. Analiza już istniejących modeli planowania przedsięwzięć budowlanych pozwala na przyjęcie w formułowanych w pracy modelach dodatkowych parametrów, ograniczeń i kryteriów, które nie były do tej pory stosowane w planowaniu przedsięwzięć realizowanych z zastosowaniem systemu pracy potokowej. Przedmiotowe modele mogą istotnie poszerzyć możliwości zastosowania systemu pracy potokowej w planowaniu przedsięwzięć budowlanych. Celem praktycznym przyjętym w rozprawie jest opracowanie odpowiedniego oprogramowania pozwalającego na rozwiązywanie występujących w modelach zadań optymalizacji dyskretnej. W pracy przewiduje się jego weryfikację przez przedstawienie możliwości zastosowania w planowaniu przedsięwzięć budowlanych w trakcie których wykorzystuje się system pracy potokowej.

Jak już wspomniano na początku tego rozdziału, w nowych modelach systemu pracy potokowej przyjęto deterministyczny charakter wszystkich danych liczbowych tj. parametrów, ograniczeń, zmiennych decyzyjnych, kryteriów w nich stosowanych. Opis modeli w podejściu deterministycznym umożliwi dalsze badania nad modelami o charakterze niedeterministycznym. W rozprawie uwzględniono kryteria czasu lub/i kosztu przedsięwzięcia. Ponadto zdecydowano się na zastosowanie niezależnych grup roboczych o stałych lub różnych wydajnościach lub składach. Ilość ich może być większa niż jedna dla każdego rodzaju robót. Istotnymi ograniczeniami przyjętymi w rozprawie jest możliwość wykonywania przez grupy robocze tylko robót jednego rodzaju oraz założenie stałej struktury zasobów (liczby grup roboczych oraz ich składów lub wydajności) przez cały czas trwania realizacji przedsięwzięcia. Innym ograniczeniem występującym w modelach jest

uwzględnienie w zasobach dla realizacji przedsięwzięcia tylko bezpośrednich wykonawców robót.

Wstępna analiza dotychczasowych badań nad metodami harmonogramowania przedsięwzięć budowlanych wykorzystujących system pracy potokowej oraz badań związanych z teorią szeregowania zadań pozwoliły na sformułowanie następującej tezy:

**Wykorzystanie elementów teorii szeregowania zadań oraz algorytmów metaheurystycznych do rozwiązywania zagadnień optymalizacji dyskretnej pozwala na powiększenie obszaru zastosowań systemu pracy potokowej w przedsięwzięciach budowlanych, szczególnie wieloobiektowych, oraz uwzględnienie w szerszym zakresie rzeczywistych warunków realizacyjnych.**

### **1.3. Metodyka badań oraz wykorzystywane w pracy narzędzia i metody**

W rozprawie przyjęto metodykę badań nad nowymi modelami planowania przedsięwzięć budowlanych realizowanych w systemie pracy potokowej, według zasad stosowanych w badaniach operacyjnych i polegającą na podziale tych badań na poszczególne etapy. Etapy te mają charakter sekwencyjny tj. nie można przejść do etapu następnego bez zakończenia badań nad etapami poprzednimi. Dla każdego z nich są stosowane metody i narzędzia badań naukowych wykorzystujących różne dziedziny wiedzy, często wykraczające poza problematykę organizacji robót budowlanych. Można więc stwierdzić, że badania te mają charakter interdyscyplinarny. Jest to jednak niezbędne dla prawidłowego rozwiązania stawianych w rozprawie problemów. Podział badań na poszczególne etapy, wraz z odpowiadającymi im metodami i narzędziami badań, jest następujący:

1. Analiza zagadnień pojawiających się w harmonogramowaniu przedsięwzięć budowlanych, ze szczególnym uwzględnieniem przedsięwzięć wykorzystujących system pracy potokowej (dokonanie przeglądu istniejących badań, wiedzy praktycznej, modeli).
2. Analiza zagadnień teorii szeregowania zadań (dokonanie przeglądu wybranych modeli optymalizacji dyskretnej stosowanych w ramach tej teorii).
3. Sformułowanie nowych modeli harmonogramowania przedsięwzięć budowlanych wykorzystujących system pracy potokowej (zastosowanie wiedzy związanej z już istniejącymi modelami harmonogramowania przedsięwzięć budowlanych i wykorzystanie odpowiednich elementów teorii szeregowania zadań po wcześniejszej ich analizie).
4. Formalny zapis nowych modeli (zastosowanie modelowania matematycznego oraz modelowania zagadnień znajdujących się obszarze teorii szeregowania zadań).

5. Przedstawienie metod i narzędzi do rozwiązywania problemów optymalizacyjnych w sformułowanych modelach (użycie metaheurystycznych algorytmów optymalizacji dyskretnej stosowanych w rozwiązywaniu zagadnień teorii szeregowania zadań oraz zastosowanie programowania komputerowego).
6. Weryfikacja wybranych w rozprawie metod rozwiązywania zagadnień z etapu poprzedniego (ocena wyników uzyskiwanych przez wybrane algorytmy poprzez zastosowanie analizy eksperymentalnej stosowanej w ramach teorii szeregowania zadań).
7. Weryfikacja sformułowanych modeli systemu pracy potokowej z użyciem opracowanego oprogramowania (zaprezentowanie przykładów planowania przedsięwzięć wieloobektowych z zastosowaniem utworzonych modeli).

## **2. Analiza stanu badań dotyczących metod harmonogramowania przedsięwzięć budowlanych z uwzględnieniem metod wykorzystujących system pracy potokowej**

Historia planowania przedsięwzięć budowlanych wiąże się ściśle z historią zarządzania projektami i historią organizacji produkcji. Jedne z pierwszych zasad dotyczących organizacji i planowania zostały podane przez K. Adamickiego [1]. Odnosiły się one m.in. do podziału pracy ze względu na specjalizację oraz harmonizacji działań uwzględniających kryterium minimalnych strat czasu w trakcie ich realizacji. Niezależnie prace nad przedmiotową tematyką prowadził H. Gantt [141]. Do jego szczególnych zasług należy m.in. wprowadzenie i upowszechnienie wykresów obrazujących realizację czynności w skali czasu, które są wykorzystywane w planowaniu przedsięwzięć budowlanych do dnia dzisiejszego. Szczególnie silny rozwój metod zarządzania i planowania projektów miał miejsce w połowie XX w. Wtedy, w ramach realizacji amerykańskich projektów wojskowych, zostały opracowane metody wykorzystujące sieci zależności między czynnościami m.in.: CPM (1957 r.), PERT (1958 r.), MPM (1958 r.), które są również w dalszym ciągu podstawowymi metodami harmonogramowania przedsięwzięć budowlanych na całym świecie [141]. Równocześnie w USA, dla przedsięwzięć budowlanych o charakterze powtarzalnym lub związanych z obiektami liniowymi, rozwijały się metody planowania bazujące na metodzie linii równowagi LOB np.: LSM, VPM, RSM [141]. W Polsce ten kierunek planowania został zapoczątkowany przez A. Dyżewskiego [22], który sformułował zasady pracy równomiernej.

Wg K.M. Jaworskiego [68] przedsięwzięcia budowlane można podzielić na dwa podstawowe rodzaje: przedsięwzięcia typu kompleks operacji oraz takie, które mogą być zorganizowane zgodnie z zasadami metody pracy równomiernej. Podział ten należy rozumieć następująco: każde przedsięwzięcie budowlane może być traktowane jako kompleks operacji, jednak tylko niektóre z nich mogą być organizowane według zasad metody pracy równomiernej [101]. Obecnie do harmonogramowania przedsięwzięć typu kompleks operacji (nie posiadających charakteru powtarzalnego) najczęściej stosowane są metody sieciowe: CPM o charakterze deterministycznym, PERT o charakterze niedeterministycznym. Metody do rozwiązywania zagadnień planowania dla przedsięwzięć organizowanych zgodnie z zasadami metody pracy równomiernej noszą nazwę potokowych metod organizacji robót [68].

W literaturze zagranicznej (np. w [147]) podział metod planowania przedsięwzięć zaznacza się ze względu na charakter przedsięwzięcia i rodzaj obiektu, do którego są one stosowane:

- metoda CPM/PERT dla złożonych, nie mających charakteru powtarzalnego przedsięwzięć z dużą liczbą czynności o charakterze dyskretnym,
- grupa technik opartych na metodzie LOB dla planowania przedsięwzięć powtarzalnych np. LSM - dla przedsięwzięć związanych z realizacją obiektów liniowych (rurociągi, drogi, autostrady, tunele itp.), VPM - dla planowania realizacji budynków wysokich z dużą liczbą powtarzalnych kondygnacji.

W następnych podrozdziałach zostanie przedstawiony aktualny stan badań dotyczących wyżej opisanych metod planowania przedsięwzięć budowlanych, ze szczególnym uwzględnieniem metod odnoszących się do planowania przedsięwzięć z procesami o charakterze powtarzalnym (do nich należą również przedsięwzięcia wieloobiettowe).

## **2.1. Aktualny stan badań dotyczących sieciowych metod harmonogramowania dla przedsięwzięć budowlanych**

Podstawową cechą metod analizy sieciowej jest przedstawienie organizacji czynności (robót) przedsięwzięcia w postaci grafu (sieci), który obrazuje zależności kolejnościowe między robotami. Analiza czasowa sieci pozwala na wskazanie ciągu robót decydującego o czasie trwania przedsięwzięcia (ścieżka krytyczna) oraz tych robót, które o nim nie decydują (czynności niekrytyczne, ze ściśle określonymi zapasami czasu).

Badania nad metodami analizy sieciowej przebiegały w kierunku ujęcia deterministycznego i probabilistycznego. Pierwszy z nich został zapoczątkowany przez metodę CPA (*Critical Path Analysis*). Na jej podstawie opracowano szereg metod w ujęciu deterministycznym m.in.:

- CPM (*Critical Path Method*) - podobnie jak metoda CPA uwzględnia ona jedynie analizę czasu trwania przedsięwzięcia,
- CPM-COST (*Critical Path Method for Cost*), LCES (*Least Cost Estimating and Scheduling*) - metody te uwzględniają, oprócz analizy czasu przedsięwzięcia, jego koszt, a celem analizy jest wyznaczenie terminu zakończenia przedsięwzięcia, dla którego całkowity koszt realizacji jest minimalny,
- MPM (*Metra Potential Method*) – uwzględnia dodatkowe ograniczenia występujące pomiędzy czynnościami (robotami) przedsięwzięcia oraz uwzględnia, podobnie jak w metodzie CPM, tylko analizę czasu trwania przedsięwzięcia,
- RAMPS (*Resource Allocation and Multi-Project Scheduling*) – obejmuje analizę wielu środków dla jednego lub więcej przedsięwzięć, które są realizowane jednocześnie.

Drugi kierunek rozwoju metod sieciowych uwzględnia ujęcie probabilistyczne. Reprezentowany jest on przez następujące metody:

- PERT (*Program Evaluation and Review Technique*) – uwzględnia jedynie analizę czasu trwania przedsięwzięcia,
- PERT – COST (*Program Evaluation and Review Technique for Cost*) - metoda ta uwzględnia, oprócz analizy czasu przedsięwzięcia, jego koszt, a celem analizy jest wyznaczenie terminu zakończenia przedsięwzięcia, dla którego całkowity koszt realizacji jest minimalny,
- GERT (*Graphical Evaluation and Review Technique*) – w odróżnieniu od metody PERT, gdzie zostały zastosowane sieci deterministyczne, w tej metodzie zastosowane są sieci stochastyczne, które dają możliwość wielowariantowego ustalania zależności pomiędzy czynnościami (robotami).

W niniejszym podrozdziale, ze względu na zakres rozprawy, rozważania na temat stanu badań nad metodami sieciowymi zostaną ograniczone do ujęcia deterministycznego, który najczęściej reprezentowany jest przez metodę CPM i jej rodzaje.

Aktualnie najczęściej rozważanym, dodatkowym kryterium, poza czasem realizacji przedsięwzięcia, jest optymalna alokacja zasobów (środków produkcji, kapitału). W związku z tym można dokonać następującego podziału metod sieciowych ze względu na ten czynnik [68]:

- metody uwzględniające czas i możliwość sumowania środków produkcji,
- metody uwzględniające czas i środki produkcji (metody z możliwością dyslokacji środków produkcji, metody uwzględniające zmienność kosztów wykonania czynności).

Pierwsza grupa zawiera metody wykorzystujące analizę harmonogramów sprawdzających tj. esogramów narastającego zużycia środków produkcji w przedsięwzięciu. Druga grupa zawiera metody, w których zakłada się ograniczenia w dostępie do środków produkcji, zależność między poniesionymi nakładami finansowymi niezbędnymi do realizacji czynności, a czasem trwania tych czynności (problem alokacji zasobów w planowaniu sieciowym), rozważany jest problem wyrównywania harmonogramu zużycia środków produkcji (zasobów, liczby pracowników itd.). Grupa tych metod umożliwia podjęcie decyzji optymalnych z punktu widzenia ww. kryteriów. Do nich należą m.in. wymienione wcześniej metody CPM-COST oraz LCSE. Stosowany w nich model zależności koszt/czas dla czynności jest liniowy. W modelu tym wyróżnia się normalny czas trwania czynności, któremu odpowiada najniższy koszt jej realizacji oraz graniczny czas trwania czynności (najkrótszy możliwy do uzyskania ze względów technologicznych), któremu odpowiada koszt graniczny (najwyższy) jej realizacji. Zależność koszt/czas pomiędzy tymi wartościami jest niemalejącą, ciągłą funkcją



liniową. Celem analizy np. w metodzie CPM-COST jest wyznaczenie terminu zakończenia przedsięwzięcia, dla którego całkowity koszt realizacji jest minimalny. Do rozwiązywania tego problemu może służyć np. algorytm kompresji sieci [6], [141].

Innego rodzaju zagadnienie stanowi następująca postać zadania optymalizacyjnego [101]: należy ustalić takie terminy realizacji i czasy trwania czynności, aby:

- zminimalizować czas trwania przedsięwzięcia przy zadanym koszcie całkowitym lub
- zminimalizować koszt całkowity przedsięwzięcia przy zadanym, nieprzekraczalnym terminie dyrektywnym jego realizacji.

Zagadnienie to (w literaturze zagranicznej stosuje się dla niego termin „*time-cost trade-off*” – w skrócie - TCT) stanowi problem optymalizacji jednokryterialnej [68] i do tej pory jest jednym z częściej rozważanych w ramach badań operacyjnych. Problem alokacji zasobów (w przypadku zagadnienia TCT - kosztów) w planowaniu sieciowym jest NP-trudny, w związku z tym nie jest możliwe skonstruowanie algorytmu, który będzie go rozwiązywał w czasie zależnym od pewnego wielomianu. Syntetyczne ujęcie tej problematyki zawiera m.in. praca J. Węglarza [144]. Najczęściej spotykaną formą podania zależności koszt/czas jest funkcja ciągła, liniowa lub nieliniowa oraz funkcja dyskretna. Problem ten, przy założeniu liniowej, ciągłej zależności koszt/czas dla czynności, można rozwiązać metodami programowania liniowego [60], [75] lub algorytmów należących do klasy metod Forda – Fulkersona [58]. W przypadku nieliniowej, ciągłej zależności koszt/czas problem można również sprowadzić do modeli programowania liniowego [60]. Do rozwiązania tego zagadnienia z uwzględnieniem postaci dyskretnych zależności koszt/czas stosowane było programowanie liniowe [89], programowanie całkowitoliczbowe [117], programowanie całkowitoliczbowe w połączeniu z programowaniem liniowym [11], [73], programowanie dynamiczne [120]. Stosowane były ponadto algorytmy heurystyczne o różnych stopniach dokładności w uzyskiwaniu wyników [27], [67], [86], [91], [114], [115], [133]. Algorytmy te dostarczają rozwiązań suboptymalnych. W ramach tej grupy algorytmów obecnie najczęściej stosowane są metaheurystyki np. algorytmy ewolucyjne (genetyczne). Użycie metaheurystyk pozwoliło na wprowadzenie wielu dodatkowych parametrów i ograniczeń, które dotyczą ściśle organizacji robót budowlanych.

Procedury wyboru projektów inwestycyjnych i wyznaczania optymalnych sposobów ich realizacji w warunkach deterministycznych i niedeterministycznych zostały przedstawione w pracy [59]. Procedury te mogą być również wykorzystywane podczas harmonogramowania przedsięwzięć budowlanych, które polegają na realizacji wielu obiektów budowlanych. W modelach uwzględnione są zagadnienia rozdziału środków finansowych do realizacji projektów inwestycyjnych. Poszukiwane są m.in. optymalne harmonogramy ich realizacji

przy kryterium maksymalizacji możliwych zysków będących do osiągnięcia w przyszłości. Modele te najczęściej zawierają zadania optymalizacji liniowej lub nieliniowej. Są rozwiązywane za pomocą m.in.: programowania liniowego, liniowego programowania stochastycznego, programowania dynamicznego, teorii nieliniowych przepływów w sieciach.

Najczęściej stosowanymi narzędziami do rozwiązywania problemów optymalizacyjnych w harmonogramowaniu z uwzględnieniem alokacji zasobów są algorytmy genetyczne [31], [106] np. w pracy [81], gdzie przyjęta zależność koszt/czas jest liniowa. W pracy tej jest rozważany przypadek przedsięwzięcia, którego realizacja jest opóźniona w stosunku do harmonogramu. W związku z tym, celem jest znalezienie harmonogramu przy założonym dyrektywnie terminie zakończenia przedsięwzięcia spełniającym kryterium minimum jego kosztu. Ze względu na NP-trudność zagadnienia algorytmy genetyczne pozwalają na uzyskiwanie tylko rozwiązań suboptymalnych (często stosunkowo bliskich optymalnym) dla instancji o rozmiarach dużo większych niż te, które były rozwiązywane za pomocą algorytmów heurystycznych, programowania liniowego czy całkowitoliczbowego. W pracy [41] przedstawione są narzędzia na bazie arkusza kalkulacyjnego, z zastosowaniem algorytmu genetycznego do planowania przedsięwzięć, z uwzględnieniem dyskretnej zależności koszt/czas. Zakres dodatkowych kryteriów jest szeroki, gdyż w ramach przeprowadzonej analizy TCT uwzględniane są jednocześnie przepływy finansowe w przedsięwzięciu, problem ograniczeń w dostępie do zasobów, problem wyrównywania harmonogramu zużycia zasobów (w tym przypadku zatrudnienia). Zagadnienie przedstawione w pracy [41] należy do problemów optymalizacji wielokryterialnej. Sprowadzenie tego problemu do zagadnienia jednokryterialnego polega na uwzględnianiu ich jako dodatkowego kosztu przedsięwzięcia za przekroczenie określonych przez decydenta (planistę) wartości granicznych. Jako kryterium optymalizacji przyjęto minimalizację kosztu całkowitego przy warunku ograniczenia czasu trwania przedsięwzięcia. Rozwinięcie tego modelu znajduje się w pracy [42], gdzie ponadto została zastosowana metoda łańcucha krytycznego [32] umożliwiającą lepszą kontrolę realizacji przedsięwzięcia przez zastosowanie buforów czasu. Przedstawiony model pozwala ponadto na kontrolę realizacji przedsięwzięcia i uaktualnianie harmonogramu w trakcie jego trwania. Jest to możliwe przez wprowadzanie do modelu danych o aktualnych stopniach zaawansowania prac i wybraniu optymalnego sposobu realizacji (wg zależności koszt/czas) pozostałej, niezrealizowanej części czynności (roboty).

Próba zastosowania algorytmów genetycznych do zagadnień harmonogramowania przedsięwzięć budowlanych z uwzględnieniem problemu TCT zaprezentowana została również w pracy [15]. Odpowiednie procedury zostały użyte w systemie komputerowym Planista. Kryteria problemu zostały ustalone podobnie jak w pracy [41]. Funkcjami celu są:

minimalizacja czasu realizacji przedsięwzięcia przy założeniu zadanego poziomu dostępności do zasobów oraz wyrównanie ich zużycia w czasie. Optymalizacyjny problem dwukryterialny został sprowadzony do jednokryterialnego przez wprowadzenie jednej funkcji celu - kosztów całkowitych. Składają się na nie: „koszt czasu” (koszt związany ze zmianą czasu trwania przedsięwzięcia w odniesieniu do zadanego przez decydenta) i „koszt zapotrzebowania na siły wytwórcze” (dodatkowy koszt użycia zasobów – pracowników - w razie przekroczenia ustalonych limitów lub nieosiągnięcia założonego poziomu).

Kolejnym przykładem zastosowania algorytmów genetycznych w rozwiązywaniu zagadnienia TCT jest praca [64]. Autor przedstawia w niej metodę projektowania struktury systemu wykonawczego, która uwzględnia ograniczenia w dostępności zasobów generalnego wykonawcy. Ze względu na nie, do zasobów realizujących przedsięwzięcia przynależą również podwykonawcy, którzy są rezerwą dla zasobów generalnego wykonawcy. Autor, jako ocenę możliwych wariantów budowy struktury systemu, proponuje zastosowanie trzech kryteriów decydujących o efektywności przedsięwzięcia: czasu realizacji przedsięwzięcia, jego kosztu całkowitego, kosztu użycia podwykonawców. Problem ten zostaje rozwiązany przez użycie funkcji celu skalaryzującej, która pozwala na wyznaczenie dobrej reprezentacji rozwiązań niezdominowanych (Pareto-optymalnych) i umożliwiającej sprowadzenie zagadnienia do zadania optymalizacji jednokryterialnej. Dobór wykonawców dokonywany jest za pomocą algorytmu genetycznego, a rozdział zasobów odnawialnych, których dostępność jest ograniczona i zmienna w czasie – za pomocą algorytmu heurystycznego opartego na regułach priorytetowych. Wybór odpowiedniego wariantu struktury systemu może być dokonywany za pomocą interaktywnego algorytmu R.E. Steuera.

Jako przykład innego rodzaju narzędzia wykorzystywanego do rozwiązywania problemu TCT w planowaniu sieciowym należy wymienić sieci neuronowe. Należą one do grupy metod opartych na tzw. sztucznej inteligencji. W pracy [2] został przedstawiony model harmonogramowania przedsięwzięć związany z budową autostrad. Łączy on cechy metod CPM i LSM wykorzystywanej do planowania przedsięwzięć liniowych. Model ten zapewnia właściwe modelowanie procesów powtarzalnych z warunkiem ciągłości ich realizacji, możliwości zatrudniania wielu zespołów roboczych o jednej specjalności, umożliwia monitorowanie przebiegu realizacji robót. Zakłada się nieliniową lub liniową postać zależności koszt/czas w modelu. Problem optymalizacji przedsięwzięcia sprowadzony jest do minimalizacji kosztów realizacji obiektu (uwzględnia się koszty bezpośrednio i pośrednio) i jest rozwiązywany za pomocą programu komputerowego wykorzystującego sieci neuronowe.

Problem alokacji zasobów (kosztów) w planowaniu sieciowym był rozwijany również w Polsce. W Instytucie Organizacji i Mechanizacji Budownictwa, na bazie metody CPM, opracowano metodę MDC (Metoda Decydujących Ciągów) [135]. MDC pozwala na analizę przedsięwzięcia pod względem czasu trwania, kosztów, użycia zasobów. Zastosowana jest optymalizacja z uwzględnieniem kryterium kosztów całkowitych, na które składają się koszty bezpośrednie, pośrednie oraz utracone korzyści. Zmiany w ujęciu problematyki kosztów w przedsięwzięciu doprowadziły do opracowania w latach siedemdziesiątych systemu MK-ps [71], w którym uwzględnia się również efekty społeczno - gospodarcze zależne od rodzaju realizowanej inwestycji. Optymalizacja w przypadku tej metody przeprowadzana jest dwuetapowo. W etapie pierwszym ustala się wstępnie optymalne czasy trwania czynności według kryterium efektów społeczno – gospodarczych inwestycji. W etapie drugim wartości czasów trwania czynności krytycznych (uzyskanych po analizie czasowej sieci zależności) są przyjmowane jako definitywnie optymalne. Optymalizacja w tym etapie polega na wydłużeniu czasów trwania czynności niekrytycznych w ramach obliczonych zapasów z uwzględnieniem kryterium minimum kosztów. Zależność koszt/czas w systemie MK-ps ma charakter ciągły, nieliniowy. Na podstawie tej metody opracowano model związany z optymalizacją pełnych cykli inwestycyjnych – system MK-27.

Innego rodzaju zagadnienie stanowią modele szeregowania sieciowego z ograniczeniami dotyczącymi zasobów [137] rozumianymi w poniżej opisany sposób. Istnieje zbiór zasobów zawierający po jednej jednostce każdego typu. Do realizacji czynności wymagane jest użycie jednej jednostki określonego zasobu, przy czym po ukończeniu czynności zasób ten jest ponownie dostępny. W związku z tym, żadne dwie czynności nie mogą korzystać jednocześnie z tego samego zasobu. W porównaniu do modelu CPM bez uwzględniania zasobów, istnieją tutaj dodatkowe ograniczenia dotyczące kolejności realizacji czynności w zbiorze czynności wykonywanych przez jedną jednostkę zasobu. Z punktu widzenia problematyki organizacji przedsięwzięć budowlanych zasoby w tym przypadku mogą stanowić grupy robocze, które mają wiele specjalności i mogą realizować różnego rodzaju prace w przedsięwzięciu. Problem w tym modelu polega na minimalizacji czasu trwania przedsięwzięcia (długości ścieżki krytycznej) z uwzględnieniem dodatkowych ograniczeń. Dla przedstawienia tego zagadnienia stosuje się sieci koniunkcyjne opisujące relacje kolejnościowe między czynnościami, z wprowadzeniem łuków dysjunkcyjnych, które narzucają dodatkowy porządek wykonywania czynności na jednostce zasobu. Problem ten, należący do NP-trudnych, można rozwiązać stosując np. algorytm Balasa [5], [137]. Rozwiązywanie tego problemu, jako element struktury systemu wykonawczego, znalazło się

również w pracy [64]. Narzędziem do rozwiązywania zagadnienia jest w tym przypadku algorytm heurystyczny bazujący na regułach priorytetowych.

Kolejnymi rodzajami kryteriów uwzględnianych w problemach planowania sieciowego są: kryterium nieprzekroczenia limitów środków produkcji, zagadnienie wyrównywania harmonogramu zużycia środków produkcji. Kryteria te występują w tych problemach w ramach optymalizacji, zarówno jedno- jak i wielokryterialnej. Problem wyrównywania harmonogramu zużycia środków produkcji przedstawiony jest w pracy [65]. Polega on na ustaleniu takiego poziomu limitów zasobów w kolejnych przedziałach czasu, aby koszty związane ze zmianą limitu zasobu i koszty ewentualnego ich niewykorzystania były jak najmniejsze. W celu rozwiązania zagadnienia stosowane są algorytmy programowania liniowego i elementy programowania dynamicznego. W pracy [106] do rozwiązywania problemu wyrównywania harmonogramów autor zastosował algorytmy ewolucyjne. Rozpatrywane są następujące postacie funkcji kryterialnych: minimalizacja średniego kwadratowego odchylenia zapotrzebowania na zasoby w danym momencie od średniego zapotrzebowania dziennego, minimalizacja maksymalnego odchylenia zapotrzebowania na zasoby od średniego zapotrzebowania dziennego, minimalizacja maksymalnego zapotrzebowania dziennego. Problem wyrównywania harmonogramu zużycia środków produkcji w przedsiębiorstwie realizowanym zgodnie z metodą łańcucha krytycznego [32], z zastosowaniem ewolucyjnego algorytmu hybrydowego HEA [8] przedstawiono również w pracy [122]. Został on przedstawiony jako zadanie optymalizacji polegające na minimalizacji odchylenia zatrudnienia od średniego poziomu zatrudnienia w przedsiębiorstwie.

## **2.2. Aktualny stan badań dotyczących metod harmonogramowania wykorzystujących system pracy potokowej w przedsiębiorstwach budowlanych**

Wg K.M. Jaworskiego [68] drugim rodzajem przedsięwzięć budowlanych, oprócz przedsięwzięć typu kompleks operacji, są takie, które można zorganizować zgodnie z zasadami pracy równomiernej. Takie podejście do planowania przedsięwzięć budowlanych powstało przez adaptację przemysłowej metody produkcji taśmowej do potrzeb budownictwa. Ogólną zasadą w systemie pracy potokowej jest dokonanie podziału realizowanego przedsięwzięcia na części (obiekty, działki robocze), które w najkorzystniejszym przypadku wymagają wykonania jednakowych ilości robót i powierzanych do wykonania zespołom roboczym (brygadam, grupom roboczym) o ustalonych specjalnościach. Zespoły te przechodząc z jednej działki do następnej wykonują stale taki sam rodzaj pracy. System ten stwarza warunki ciągłości i równomierności zatrudnienia grup roboczych, pracy maszyn,

zużycia materiałów. Umożliwia on również podniesienie wydajności grup roboczych przez wielokrotną powtarzalność powierzonych im zadań oraz specjalizację w wykonywaniu tych samych czynności. Ważnym zagadnieniem w metodzie pracy równomiernej jest odpowiedni podział realizowanego przedsięwzięcia na działki robocze. Odpowiednie zasady umożliwiające zastosowanie przedmiotowej metody i uwzględniające ograniczenia występujące w produkcji budowlanej podane są w pracach [22], [125]. Najkorzystniejsze warunki stosowania tej metody występują przy realizacji: przedsięwzięć z dużą liczbą typowych obiektów lub segmentów, obiektów wielokondygnacyjnych o znacznej długości (budynki wieloklatkowe), obiektów liniowych (w planie których przeważa jeden kierunek) np. dróg, linii kolejowych, kanałów wodnych, sieci wodociągowych, kanalizacyjnych, rurociągów itp. [101].

W pracy [126] podana jest klasyfikacja obiektów budowlanych, które mogą być realizowane według metody pracy równomiernej. Wyróżnia się trzy zasadnicze grupy, którym odpowiadają jednocześnie rodzaje procesów realizacji tych obiektów. Pierwsza grupa to obiekty i procesy jednotypowe (jednego typu), które występują, gdy przedsięwzięcie polega na realizacji obiektów podzielonych na działki robocze o jednakowej wielkości (czasy trwania odpowiednich procesów realizowanych na działkach są jednakowe). Druga grupa to obiekty i procesy jednorodne. Występują one wtedy, gdy działki robocze różnią się między sobą wielkością. Między wielkościami działek i pracochołnościami poszczególnych procesów występuje proporcjonalność, jak np. w zespołach budynków różniących się wielkością powierzchni zabudowy lub liczbą kondygnacji nadziemnych. Dla dwóch pierwszych grup obiektów mogą być stosowane klasyczne lub z niewielkimi modyfikacjami zasady pracy równomiernej [126]. Ostatnią, trzecią grupą są obiekty i procesy niejednorodne, które charakteryzują się brakiem proporcjonalności pomiędzy rozmiarami działek roboczych i pracochołnościami procesów, które muszą być w nich realizowane. W związku z tym, stosowanie metod pracy równomiernej do planowania realizacji tego rodzaju procesów nie pozwala na uzyskanie stałej wartości rytmu pracy grup roboczych, co jest cechą występującą w metodzie pracy równomiernej. Pojawia się nowy, nie występujący w dwóch pierwszych grupach czynnik - kolejność realizacji robót na działkach roboczych, która wpływa na czas trwania całego przedsięwzięcia. Zagadnienie to można więc przedstawić jako zadanie optymalizacji dyskretnej ze zmienną decyzyjną - kolejnością wykonywania działek oraz odpowiednią wybraną przez planistę funkcją celu.

Z powyższych rozważań można wysnuć wniosek, że niejednorodny model obiektu i procesu ma charakter najbardziej ogólny z pośród trzech wcześniej wymienionych modeli. Stosując założenie niejednorodności obiektów (działek roboczych) i procesów opracowano

wiele modeli organizacji przedsięwzięć z systemem pracy potokowej. Szczególne znaczenie mają badania prowadzone przez V.A. Afanasjewa [3], które zaowocowały utworzeniem metod organizacji robót, w których podstawową zasadą tworzenia są sprzężenia czasowe, czyli odległości w czasie między robotami (definiuje się sprzężenia między frontami roboczymi (działkami roboczymi), między środkami realizacji, diagonalne i odwrotnie diagonalne). Ich wspólną cechą jest przedstawienie organizacji przedsięwzięcia budowlanego jako zadania optymalizacyjnego, w którym zmienną decyzyjną jest kolejność wykonywania obiektów (działek roboczych, frontów). W Polsce zasady dla metod organizacji robót uwzględniających sprzężenia czasowe przedstawiono w pracach [49], [17], [95], gdzie kryterium wspólnym dla wszystkich metod przyjęto czas realizacji przedsięwzięcia. Do rozwiązania zadania optymalizacyjnego zastosowano metodę podziału i ograniczeń.

Kolejnym krokiem w badaniach nad metodami organizacji robót ze sprzężeniami czasowymi (nazywanymi obecnie metodami sprzężeń czasowych [50] lub TCM - „*Time Couplings Methods*”) było zdefiniowanie potoków o strukturze równoległej [45]. Wprowadzenie tej struktury pozwala na równoległe prowadzenie robót w kompleksach (większych grupach obiektów), z założeniem ciągłości realizacji procesu wiodącego np. montażu przy użyciu żurawia. Taka sytuacja może wystąpić przy planowaniu dużych przedsięwzięć budowlanych np. osiedli mieszkaniowych, gdzie występuje warunek ograniczonego czasu realizacji. W pracy [45] przedstawiono ponadto metodę rozwiązania zadania optymalizacyjnego polegającą na zastosowaniu odpowiednio zmodyfikowanego algorytmu podziału i ograniczeń. W kolejnej pracy [47] zaprezentowano inną strukturę, która zakłada pewne uszeregowanie w zbiorze potoków. Potokami w tym przypadku są uporządkowane już wcześniej, za pomocą odpowiednich metod organizacji wybranych przez planującego przedsięwzięcie, kompleksy (zbiory) obiektów budowlanych (działek roboczych). Zadaniem optymalizacyjnym jest odpowiednie uszeregowanie zaplanowanych już kompleksów z założeniem kryterium minimum czasu realizacji całego przedsięwzięcia. Zasada działania algorytmu optymalizacyjnego oparta jest również na metodzie podziału i ograniczeń. W pracy [54] opisany problem zakwalifikowany został do grupy metod organizacji robót budowlanych o szeregowej strukturze wewnętrznej.

W pracy [44] przedstawiony został model uwzględniający pracę wielu brygad jednej specjalności w ramach systemu pracy potokowej. Tak jak wcześniej kryterium optymalizacyjnym jest czas realizacji przedsięwzięcia. Występujące w modelu zagadnienia przydziału działek roboczych poszczególnym brygadam rozwiązano wykorzystując metodę symulację komputerowej i metody podziału i ograniczeń. W pracy [54] zagadnienie to zaklasyfikowano do grupy metod organizacji o równoległo – potokowej strukturze

wewnętrznej. Oryginalne podejście do problematyki równoległo – potokowych metod organizacji robót uwzględniających pracę wielu brygad jednej specjalności przedstawiono w pracy [84]. Do rozwiązania tych zagadnień stosowane są algorytmy: heurystyczny oraz metoda podziału i ograniczeń. Zagadnienie synchronizacji przy wzajemnej organizacji zależności potoków zostało poruszone w pracy [48]. Problem ten pojawia się podczas planowania równoległej realizacji wielu obiektów, przy czym występuje dodatkowe ograniczenie polegające na narzuceniu terminu zakończenia prac w jednym z obiektów związanym z realizacją jednego rodzaju robót (np. robót montażowych przy użyciu żurawia) w obiekcie poprzednim. Takie sytuacje mogą się pojawiać podczas realizacji budynków mieszkalnych w technologii uprzemysłowionej, jak również obiektów budownictwa przemysłowego. Powyższy problem rozwiązuje się stosując algorytm heurystyczny. Opisane zagadnienie przedstawiono w pracy [54] i zaklasyfikowano jako przypadek metody organizacji robót budowlanych o równoległej strukturze wewnętrznej.

Opisane powyżej zagadnienia dotyczące potoków równoległych, które są często spotykane w praktyce wykonawczej, są częścią ważnego kierunku w badaniach nad metodami sprzężeń czasowych [54], [94]. Kierunek ten można określić jako badania nad strukturą wewnętrzną planowanych potoków robót, która ma charakter złożony, zależny od ograniczeń występujących w planowanym przedsięwzięciu. Oprócz struktur równoległych należy zauważyć występowanie w przedsięwzięciach również różnego rodzaju struktur szeregowych [47], [54], [94]. Powstają one np. wtedy, gdy część robót przeznaczonych do realizacji należy zaplanować stosując różne metody organizacji, co wynika z ograniczeń technologicznych, koniecznego narzucenia określonych warunków pracy grup roboczych itp. Pojawia się więc problem synchronizacji (odpowiedniego połączenia) części kompleksów obiektów, które są modelowane różnymi metodami organizacji i zakładają dodatkowo różne kolejności ich realizacji. Kryterium optymalizacyjnym w tym przypadku jest, podobnie jak w metodach sprzężeń czasowych, czas realizacji przedsięwzięcia. W pracach [54], [94] dla rozwiązania zadanie optymalizacyjnego zastosowano oryginalny algorytm heurystyczny, który zawiera dwa etapy. W pierwszym dokonuje się optymalizacji kolejnych kompleksów robót utworzonych z uwzględnieniem przyjętych metod organizacji robót (wyznacza się nową kolejność realizacji obiektów zapewniając maksymalne skrócenia czasu realizacji robót). W drugim dokonuje się połączenia częściowych, już zoptymalizowanych kompleksów robót za pomocą algorytmu heurystycznego, który zapewnia minimalizację przerw w pracy grup roboczych w strefie połączenia kompleksów.

Innym kierunkiem badań nad metodami sprzężeń czasowych było rozszerzenie dotychczasowego pojęcia sprzężeń czasowych i przyjęcie podczas modelowania struktury



potokowej dodatkowych warunków wyznaczonych z ustalonych możliwych kombinacji najwcześniejszych i najpóźniejszych terminów rozpoczęcia i zakończenia robót. Prace nad tym kierunkiem badań zostały podjęte po dostrzeżeniu faktu, że sprzężenia czasowe (w formie przedstawionej w pracach [94], [95]) nie odzwierciedlają wielu rzeczywistych, dodatkowych ograniczeń występujących w trakcie realizacji przedsięwzięć budowlanych [49]. Prace nad nowymi wariantami (przypadkami) metody sprzężeń czasowych organizacji robót zostały zapoczątkowane w [49]. Rozwinięcie analiz nowych wariantów metod zostało dokonane w pracach [54], [94], [128]. W pracach [94], [128] przedstawiono nowe warianty powstałe na bazie metod organizacji przedstawionych w pracy [95]. Dokonano formalizacji matematycznej przypadków nowych sprzężeń, sprawdzono poprawność nowych zależności z praktycznego punktu widzenia, przedstawiono zastosowania praktyczne.

Podsumowując analizę badań na temat struktury wewnętrznej potoku i rozszerzenia pojęcia sprzężenia czasowego można przyjąć, że analiza ta spowodowała pojawienie się kryteriów podziału utworzonych metod organizacji robót budowlanych [54], [94]. W pracy [54] przedstawiono również nowe, nierozwiązane problemy modelowania organizacji robót budowlanych, które dotyczą metod kolejnego wykonania bez uwzględniania jednoczesności robót (modele o strukturze szeregowej) oraz metod równoległego wykonania z równoległym prowadzeniem robót jednego rodzaju lub jednoczesnym prowadzeniem robót różnego rodzaju (modele o strukturze równoległej). Odpowiednie metody, które można zaklasyfikować do powyższych modeli, tworzone są m.in. przez uwzględnienie dodatkowych ograniczeń wynikających z narzuconych przez decydenta warunków odnośnie do sprzężeń czasowych. Praca [54] prezentuje nowe metody i ich warianty powstałe z uwzględnieniem sprzężeń diagonalnych, odwrotnych diagonalnych, nowo utworzonych sprzężeń przekątnych i odwrotnie przekątnych oraz ich kombinacji. Dla części z nich podano algorytmy rozwiązywania zadania optymalizacyjnego z czasem trwania przedsięwzięcia jako kryterium. Rezultaty badań dotyczących metod sprzężeń czasowych zostały przedstawione również w pracach [46], [52], [53].

Problem optymalizacji poziomu zatrudnienia pracowników, przy założonym czasie trwania przedsięwzięcia budowlanego uzyskanym z zastosowaniem metody sprzężeń czasowych TCM III, jest rozpatrywany w pracy [123]. Jako miarę nierównomierności zapotrzebowania na pracowników przyjęto średnie odchylenie od średniego dziennego zapotrzebowania. Problem ten jest rozwiązywany za pomocą hybrydowego algorytmu ewolucyjnego HEA [8]. Ponadto w pracy [123] jest analizowany problem wpływu wielkości buforów czasowych robót na koszt przedsięwzięcia finansowanego ze środków zewnętrznych

(kredytu), przy założeniu czasu trwania przedsięwzięcia budowlanego uzyskanego z zastosowaniem metody sprzężeń czasowych TCM III.

W pracy [88] przedstawiono modele harmonogramowania przedsięwzięć budowlanych stosujących system pracy potokowej z innymi ograniczeniami i kryteriami niż podano powyżej. Przyjęto w nich możliwości zazębiania się robót (realizacji wielu rodzajów robót jednocześnie na danej działce roboczej). Innym, dodatkowym ograniczeniem jest dostępność zasobów (grup roboczych) oraz działek w określonych terminach. Narzucone są ponadto terminy dyrektywne zakończenia prac na poszczególnych działkach roboczych. W przedstawionych modelach przyjęto kryterium optymalizacji, które wyraża sumaryczne koszty strat z tytułu braku ciągłości pracy grup roboczych oraz „kar” z tytułu niedotrzymania terminów dyrektywnych zakończenia robót na działkach roboczych. Celem zadania optymalizacyjnego jest poszukiwanie kolejności realizacji działek roboczych, tak aby zminimalizować podane kryterium. Zadanie to rozwiązano za pomocą algorytmu podziału i ograniczeń. W innym modelu, w pracy [88], przedstawiony jest problem indywidualnego sterowania grup roboczych w realizacji prac na działkach roboczych, zastosowany w opisanym powyżej modelu systemu pracy potokowej. Dla każdej z grup roboczych przyjęto indywidualną kolejność realizacji działek. W związku z tym, wykładniczo zwiększyła się ilość możliwych kolejności realizacji działek i jednocześnie zbioru możliwych harmonogramów realizacji robót. Zadanie optymalizacyjne zostało rozwiązywane za pomocą algorytmu heurystycznego wykorzystującego elementy przeszukiwania losowego lub przeglądu zupełnego części zbioru rozwiązań. W pracy [87] podano z kolei inny sposób rozwiązania zadania optymalizacyjnego za pomocą metody symulacyjno – analitycznej, wykorzystującej mechanizm losowego generowania i bieżącej oceny jakościowej rozwiązań stanowiących kolejne (dopuszczalne) warianty harmonogramów robót.

Kolejny model systemu pracy potokowej jest zaprezentowany w pracy [66]. Odnosi się on do projektowania realizacji przedsięwzięć wieloobektowych. W przyjętym modelu wykonanie każdego z obiektów wymaga realizacji robót, między którymi zachodzą relacje kolejnościowe opisane siecią zależności. Przyjęto ponadto możliwość realizacji jednego rodzaju robót przez więcej niż jedną grupę roboczą (zestawy maszyn lub brygady wyspecjalizowanych pracowników). Zadanie optymalizacyjne w tym modelu polega na takim przydzieleniu zasobów do realizacji poszczególnych robót i jednocześnie ustaleniu kolejności ich realizacji w obiektach, aby czas trwania przedsięwzięcia był minimalny. Zadanie to jest rozwiązywane za pomocą algorytmu należącego do grupy metod podziału i ograniczeń, który wykorzystuje teorię grafów dysjunktywnych.

Powyższa analiza stanu badań nad systemem pracy potokowej dotyczyła przede wszystkim prac przedstawionych w Polsce. W pracach zagranicznych do planowania przedsięwzięć o charakterze powtarzalnym najczęściej wykorzystywana jest koncepcja metody tzw. linii równowagi („*Line of Balance*” - LOB). Metoda LOB, która jest oparta na technice wykresów liniowych, została wprowadzona w latach 40. ubiegłego wieku w USA głównie dla planowania zadań realizowanych w przemyśle, np. samochodowym, gdzie mamy do czynienia z realizacją serii powtarzalnych produktów o dużej liczebności. Ideą tej metody jest wprowadzenie równowagi (równomierności) w realizacji operacji (czynności, robót), wchodzących w skład przedsięwzięcia, które są realizowane w sposób ciągły [82]. Zaletami metody LOB jest przedstawienie harmonogramu przedsięwzięcia w formie graficznej (cyklogramu), pozwalającej na łatwe dostarczenie informacji o aktualnym stopniu wykonania operacji wchodzących w skład realizacji poszczególnych produktów oraz przyjęcie założenia ciągłości wykonywania prac przez zasoby. W literaturze zagranicznej wymienia się wiele technik opartych na założeniach tej metody. Mogą być one z powodzeniem stosowane w budownictwie w trakcie planowania różnego rodzaju przedsięwzięć powtarzalnych, które były wymienione na początku rozdziału. Spośród nich należy tutaj wymienić następujące techniki (na podstawie [83]):

- TSSM (*Time Space Scheduling Method*) [136],
- VPM (*Vertical Production Method*) [102],
- LSM (*Linear Scheduling Method*) [14], [70],
- RPM (*Repetitive Project Model*) [116],
- technika „*Line of Balance Scheduling*” [13],
- technika „*Velocity Diagrams*” [121].

Poniżej zostaną bliżej przedstawione niektóre z tych metod.

Technika LSM (*Linear Scheduling Method*) [14], [70] jest przeznaczona do planowania realizacji obiektów liniowych takich jak autostrady, drogi, rurociągi itp. Jednym z głównych jej elementów jest prezentacja wyniku procesu planowania przedsięwzięcia w postaci cyklogramu (najczęściej: oś pionowa to działki robocze np. odcinki drogi o stałej długości, oś pozioma - czas realizacji), z przedstawieniem czynności w postaci linii nachylonych pod kątem zależnym od przyjętego tempa realizacji. Wprowadzone zostają ponadto pojęcia minimalnych buforów odległości (minimalna odległość między miejscami realizacji robót: poprzedniej i następnej w kolejności) oraz czasu (minimalny czas pomiędzy realizacją robót: poprzedniej i następnej w kolejności w danym miejscu; jest to odpowiednik sprzężenia między frontami roboczymi w metodach sprzężeń czasowych). W pracy [70] podany jest również algorytm heurystyczny optymalizacji zagadnienia TCT dla

przedsięwzięcia o charakterze powtarzalnym. Wprowadza się ponadto pojęcie czynności dyskretnych (nie mających charakteru powtarzalnego) np. wykonanie przepustu pod autostradą, co ma swoje odzwierciedlenie w cyklogramie. Jest ponadto możliwość uwzględnienia wpływu zmian pogody na realizację przedsięwzięcia przez zastosowanie współczynników redukujących prędkość wykonywania robót lub przez ich wstrzymanie. Jest możliwe zastosowanie kontroli budżetu przedsięwzięcia z użyciem esogramów stosowanych w metodach sieciowych.

W pracy [102] została przedstawiona technika VPM (*Vertical Production Method*), która powstała jako dostosowanie metody LOB do planowania wznoszenia budynków wysokich (wielokondygnacyjnych). Zakłada się w niej, że realizacja każdej z kondygnacji budynku wysokiego przebiega według utworzonego planu sieciowego. Ze względu na to, że odpowiednia sieć zależności i harmonogram dla całego budynku byłby nieczytelny, dla kondygnacji powtarzalnych proponuje się zastosowanie cyklogramów (oś pionowa – kondygnacje budynku). Dla elementów niepowtarzalnych np. fundamentów proponuje się zastosowanie metody CPM.

Technika TSSM (*Time Space Scheduling Method*), przedstawiona w pracy [136], jest kolejnym rodzajem aplikacji metody LOB w zagadnieniach planowania przedsięwzięć o charakterze powtarzalnym. W zasadniczych punktach technika ta nie różni się od podstawowych założeń metody LOB (podział na działki robocze, założenie ciągłości pracy grup roboczych, które wywołuje efekt uczenia się i w związku z tym wzrost wydajności pracy, stosowanie cyklogramów). Praca [136] dostarcza wielu przykładów stosowania tej techniki: w budownictwie mieszkaniowym, drogowym, podczas realizacji rurociągów, mostów itp.

W pracy [116] podana jest technika RPM (*Repetitive Project Model*), która odnosi się do praktycznie wszystkich rodzajów przedsięwzięć powtarzalnych. Jej podstawowe elementy są przeniesione z metody LOB i podobne do opisanych w technice LSM. W tej technice zastosowane jest, jako podstawowy element optymalnego planowania, zagadnienie TCT. Zakłada się, że realizacja każdej działki roboczej przedsięwzięcia może być przedstawiona w formie planu sieciowego. Celem tej techniki jest zaplanowanie całego przedsięwzięcia tak, aby jego koszt bezpośredni był minimalny, zakładając ograniczenia dotyczące: terminu zakończenia realizacji przedsięwzięcia, ciągłości realizacji robót, utrzymania stałej prędkości wykonywania robót przez grupy robocze, utrzymania ustalonych buforów czasu i buforów odległości (zmienną decyzyjną będą tempa realizacji poszczególnych robót na działkach zależne od ich kosztów realizacji). Występujące zadanie optymalizacyjne proponuje się rozwiązać metodą programowania liniowego.

Odmienne podejście do problemów planowania realizacji obiektów o charakterze liniowym zostało przedstawione w pracach [38], [39]. Autorzy, niezależnie od siebie, zaproponowali nowe podejście do harmonogramowania obiektów liniowych. Zauważyli oni, że istotnym ograniczeniem powodującym słabą akceptację tej grupy metod przez przemysł (w tym budownictwo), jest brak odpowiednich narzędzi pozwalających na kontrolę i aktualizację harmonogramów liniowych. Autorzy proponują zastosowanie w tym celu idei „ścieżki krytycznej” występującej w metodzie CPM/PERT. Pozwala ona na wyróżnienie czynności lub ich części (w funkcji długości planowanego obiektu), których realizacja w zaplanowanych terminach jest niezbędna oraz tych, których realizacja ma określone zapasy czasu. Ponadto autorzy wprowadzają do harmonogramów czynności, których realizacja nie ma cech liniowych, powtarzalnych (podobnie jak w [70]).

Aktualne badania nad metodami planowania wywodzącymi się z metody LOB są skupione na szeroko rozumianej optymalizacji harmonogramów. Najczęściej rozwiązywane są problemy optymalizacyjne z kryteriami: minimalizacji czasu trwania przedsięwzięcia, minimalizacji łącznego czasu przestoju grup roboczych, minimalizacji kosztów realizacji przedsięwzięcia przy uwzględnieniu warunku narzuconego terminu jego zakończenia. Do rozwiązywania tego typu zadań optymalizacyjnych stosowane są następujące narzędzia: programowania liniowego [117], programowania dynamicznego [23], [92], [127], [130], sieci neuronowe [2], algorytmy genetyczne [43].

W pracy [130] przedstawiony jest model harmonogramowania dla obiektów liniowych, w którym funkcją celu jest czas trwania przedsięwzięcia powtarzalnego. Poszukuje się optymalnego wyboru sposobów realizacji poszczególnych czynności za pomocą algorytmu programowania dynamicznego, zakładając ciągłość prac grup roboczych. Autor zauważa również, że złamanie zasady ciągłości prac grup roboczych (jest to jedna z podstawowych zasad metody LOB) i wprowadzenie w harmonogramie przestojów w ich pracy pozwala na istotne skrócenie czasu trwania przedsięwzięcia. Problem ten został ujęty w pracy [127], gdzie poszukiwania optymalnego wyboru sposobów realizacji poszczególnych czynności zapewniają minimum czasu trwania przedsięwzięcia z uwzględnieniem możliwych przestojów w pracy grup roboczych. Dla tego zagadnienia w pracy [25] został z kolei podany inny algorytm wykorzystujący programowanie dynamiczne, który zawiera dodatkowo algorytm generowania (symulowania) możliwych układów przestojów w pracy. Metoda ta pozwala na osiąganie lepszych rozwiązań niż metoda, która została podana w pracy [127]. W pracy [24] zostało przedstawione podobne zagadnienie jak w [25], [127], rozwiązywane również za pomocą programowania dynamicznego. Autorzy wprowadzają w tym modelu dodatkową możliwość wykonywania czynności równolegle z innymi (w modelach tych

dopuszcza się relacje kolejnościowe między robotami wyrażone sekwencją). Istnieje jednak ograniczenie tylko do jednego, dodatkowego ciągu czynności równoległych.

Problem minimalizacji łącznego czasu przestoju grup roboczych został rozwiązany w pracy [26] za pomocą algorytmu heurystycznego. Model ten zakłada następujące ograniczenia: relacje kolejnościowe między robotami przedsięwzięcia wyrażone sekwencją, istnienie ograniczonego zbioru zasobów (grup roboczych), dla których mogą występować ograniczenia dostępności w czasie, różnice w wydajności poszczególnych grup roboczych wykonujących robotę jednego rodzaju. Ponadto czasy trwania czynności jednego rodzaju dla poszczególnych działek mogą różnić się od siebie ze względu na różną ilość robót do zrealizowania na działkach (np. robót ziemnych w trakcie realizacji odcinka drogi) oraz może zostać wprowadzona przez wykonawcę różna sekwencja (kolejność) realizacji działek roboczych. Zadanie optymalizacyjne polega na takim przydzieleniu grup roboczych do realizacji robót na działkach, aby suma przestojów w ich pracy była minimalna.

Zagadnienie minimalizacji kosztów realizacji przedsięwzięcia, przy uwzględnieniu warunku narzuconego terminu jego zakończenia w przedsięwzięciach powtarzalnych, zostało przedstawione w pracy [92]. Przyjęto w niej ograniczony zbiór dostępnych zasobów (grup roboczych). Zadanie polega na optymalnej alokacji grup roboczych do rodzajów robót, aby spełnić warunek kryterialny. W pracy [23] podjęto to samo zagadnienie uwzględniając jednocześnie drugą zmienną decyzyjną – możliwe przestoje w pracy grup roboczych. Zadania optymalizacyjne w obu pracach [23], [92] zostały rozwiązane za pomocą algorytmów wykorzystujących programowanie dynamiczne. Narzędzie to zostało wykorzystane również w pracy [131] do optymalnego planowania struktury zasobowej przedsięwzięcia powtarzalnego podobnego do przedstawionego w pracy [24]. W tym modelu dopuszcza się jednak występowanie większej niż jeden ilości dodatkowych ciągów czynności równoległych. Przyjęto jako kryterium optymalizacji koszt całego przedsięwzięcia, przy czym wprowadzono podział na koszty bezpośrednie i pośrednie. Zadanie optymalizacyjne w pracy [131] polegało na takim przydzieleniu grup roboczych do robót, aby kryterium osiągnęło wartość minimalną.

W pracy [93] zaprezentowano model optymalnego planowania przedsięwzięcia powtarzalnego np. wykonania drogi, z uwzględnieniem fizycznych przeszkód, które mogą pojawić się na jej długości np. rzeki, potoku. W modelu przyjęto możliwość występowania różnych ilości robót danego rodzaju dla poszczególnych odcinków przedsięwzięcia, różnych długości tych odcinków, czynności powtarzalnych oraz niepowtarzalnych do wykonania tylko w danym miejscu, ograniczonego zbioru zasobów (grup roboczych), wpływu warunków pogodowych na zmniejszenie ich wydajności. Model przedsięwzięcia pozwala na formowanie grup roboczych korzystając z utworzonej wcześniej bazy danych o sprzęcie i sile roboczej.

Zmienne decyzyjne są dwie: wielkość grup roboczych oraz alokacja ich do wykonywania czynności na danym odcinku. Funkcję celu wybiera planista, który ma dyspozycji czas przedsięwzięcia lub jego koszt całkowity składający się z kosztów bezpośrednich i pośrednich. Suboptymalne rozwiązanie tego problemu dostarcza utworzony algorytm wykorzystujący reguły heurystyczne oraz programowanie dynamiczne.

Model przeznaczony dla planowania przedsięwzięć o charakterze powtarzalnym np. kompleksów budynków mieszkalnych opisany jest pracy [72]. Autorzy wprowadzają pionowy podział na działki robocze przedsięwzięcia, związany z kolejnymi kondygnacjami budynków oraz poziomy podział obejmujący grupy wykonywanych budynków. W modelu uwzględniony jest problem odpowiedniego doboru składu grup roboczych, przydziału grup do realizacji czynności, strat spowodowanych przez przestoje w pracy. Na podstawie utworzonych reguł heurystycznych (równań) dobierana jest odpowiednia wielkość poziomych działek roboczych oraz odpowiedni przydział grup roboczych do realizacji czynności w celu zminimalizowania strat spowodowanych przez przestoje w pracy grup roboczych.

W pracy [43] przedstawiono model optymalizacji przedsięwzięć powtarzalnych uwzględniający relacje kolejnościowe między robotami dla pojedynczej działki roboczej dane grafem (jak metodzie CPM/PERT). W modelu rozwiązywany jest problem doboru składu i wielkości grup roboczych, aby nie został przekroczony zadany termin zakończenia przedsięwzięcia. Funkcja celu, która jest minimalizowana i odzwierciedla koszt całkowity przedsięwzięcia, jest ustalona podobnie jak w pracach [41], [42]. Uwzględnia ona m.in.: koszty bezpośrednie i pośrednie przedsięwzięcia, problem wyrównywania harmonogramu zatrudnienia i zapewnienia ciągłości pracy grup roboczych (stosowanie kar za przerwy w pracy grup roboczych i ponadplanowe ich wykorzystywanie). Zadanie optymalizacji dyskretnej jest rozwiązywane za pomocą algorytmu genetycznego aplikowanego w środowisku Microsoft Excel.

W pracy [40] został zaprezentowany model harmonogramowania przedsięwzięć infrastrukturalnych, które składają się z sektorów (działek roboczych). Oparty jest on na założeniach metody LOB i jest aplikacją modelu przedstawionego w pracy [41]. Cechą rozważanych przedsięwzięć są duże odległości pomiędzy sektorami, które powodują potrzebę uwzględnienia w tworzonym harmonogramie dodatkowych czasów potrzebnych na przerzucenie grup roboczych z sektora poprzedniego na następny. Zakłada się ciągłość realizacji robót przez grupy robocze, z możliwością generowania przerw w ich pracy, co może korzystnie wpłynąć na czas i koszt przedsięwzięcia. Ponadto, w modelu tym jest przyjęta możliwość uwzględniania zmian wydajności grup roboczych zależna od sektora i spowodowanych warunkami pogodowymi przyjętymi zależnie od miesiąca realizacji

przedsięwzięcia. Funkcją celu jest koszt przedsięwzięcia, na który składają się m.in.: koszty bezpośrednie, pośrednie, koszt transportu grup roboczych, przy spełnieniu warunku terminu zakończenia wszystkich robót. Zmiennymi decyzyjnymi są: metoda realizacji danego rodzaju roboty, liczba grup roboczych dla danego rodzaju roboty, kolejność realizacji sektorów, możliwe przerwy w realizacji robót w danym sektorze przez daną grupę roboczą. Narzędziem do rozwiązywania powstałego problemu optymalizacji jest algorytm genetyczny.

### 2.3. Podsumowanie

Zarówno w literaturze polskiej, jak i zagranicznej, badacze problemów planowania robót budowlanych wyróżniają grupy narzędzi dla przedsięwzięć typu kompleks operacji (w literaturze zagranicznej - *nonrepetitive projects*) i realizowanych w ramach systemu pracy potokowej, zgodnie z zasadami pracy równomiernej (w literaturze zagranicznej - *repetitive projects*). W ramach obu tych grup metod autorzy rozwiązywali szereg istotnych problemów, które występują w praktyce. Do nich należą m.in.: uwzględnienie w harmonogramowaniu kosztów robót i całego przedsięwzięcia, optymalna alokacja zasobów, wyrównanie zużycia zasobów w czasie przedsięwzięcia, skracanie czasu realizacji przedsięwzięcia. Problemy te z reguły sprowadzane są do jednokryterialnych zadań optymalizacyjnych. Rozwiązywane są za pomocą algorytmów, które dostarczają wyniki dokładne lub suboptymalne.

Tematyka rozprawy jest związana z metodami harmonogramowania wykorzystującymi system pracy potokowej. W literaturze polskiej [68], w porównaniu do prac zagranicznych na ten temat, zaznacza się możliwość wystąpienia w danym przedsięwzięciu obiektów (frontów robót, działek roboczych) niejednorodnych. Badania z nimi związane (m.in. metody sprzężeń czasowych) np. [54], [88], [94] wyróżniają się m.in. uwzględnieniem kolejności realizacji działek podczas tworzenia harmonogramów, co prawie nie jest spotykane w innych pracach polskich i zagranicznych. Z drugiej strony zauważalne są niedostatki tej grupy narzędzi, co istotnie zawęża ich zastosowania praktyczne:

- parametry, ograniczenia i funkcje celu w nich zastosowane mają charakter deterministyczny,
- relacje kolejnościowe między robotami wyrażone są niezmienną sekwencją dla poszczególnych obiektów (działek),
- utrudnione jest zgodne z rzeczywistością modelowanie ograniczeń technologicznych,
- ograniczone jest dowolne modelowanie struktury zasobów (grup roboczych) dla realizacji robót przedsięwzięcia,
- uwzględnia się najczęściej tylko kryterium czasu trwania całego przedsięwzięcia,



- dla rozwiązania zagadnienia kolejnościowego stosowane są najczęściej metody bazujące na schemacie podziału i ograniczeń.

W związku z powyższym, w rozprawie proponuje się rozszerzenie możliwości zastosowania systemu pracy potokowej w przedsiębiorstwach budowlanych m.in. przez zastosowanie niektórych rozwiązań, które były wykorzystywane w modelach przedsięwzięć typu kompleks operacji. Ze względu na istnienie analogii modeli stosujących systemu pracy potokowej (m.in. metod sprzężeń czasowych), uwzględniających działki niejednorodne, do problemów rozwiązywanych przez teorię szeregowania zadań (co przedstawiono w rozdziale 3), dla dokładnego opisu proponowanych modeli proponuje się zastosowanie pojęć, definicji i niektórych rozwiązań stosowanych w tej teorii. Zastosowanie aktualnie używanych w niej narzędzi służących do rozwiązywania problemów optymalizacji dyskretnej pozwoli na wprowadzenie innych, nowych ograniczeń i funkcji kryterialnych.

### **3. Analiza niektórych zagadnień teorii szeregowania zadań oraz metod ich rozwiązywania**

Problemy szeregowania zadań znajdują się w obszarze zagadnień poruszanych w ramach badań operacyjnych, optymalizacji dyskretnej, programowania dyskretnego i kombinatorycznego [134]. Celem w rozwiązywaniu tych problemów jest znalezienie optymalnej, dla przyjętej funkcji celu, kolejności wykonywania zadań przy użyciu zbioru maszyn przy ograniczeniach nałożonych na zadania, maszyny i ich wzajemne relacje. Modele tych zagadnień mają charakter deterministyczny. Stanowią one do tej pory przedmiot intensywnych badań w wielu środkach naukowych na całym świecie, co wynika z tego, że modelują one funkcjonowanie rzeczywistych systemów wytwarzania i produkcji przemysłowej. Dla zdecydowanej większości problemów praktycznych niemożliwe jest skonstruowanie efektywnych algorytmów ich rozwiązywania, głównie ze względu na silną NP-trudność problemów optymalizacyjnych w nich występujących. NP-trudność problemów optymalizacji dyskretnej polega na braku możliwości skonstruowania algorytmu dokładnego (znajdującego rozwiązanie optymalne), który rozwiązywałby dany problem w czasie zależnym od wyrażenia wielomianowego. Dla takich problemów są możliwe do skonstruowania tylko takie algorytmy dokładne, których czas obliczeń rośnie wykładniczo wraz ze wzrostem rozmiarów problemów. W związku z tym, nawet wielokrotny wzrost mocy obliczeniowej komputerów nie powoduje znaczącej poprawy szybkości rozwiązywania tego typu zagadnień algorytmami dokładnymi. Zagadnienia szeregowania zadań znajdują zastosowanie m.in. w następujących gałęziach przemysłu [134]:

- chemicznym (np. procesy przetwarzania ropy naftowej),
- samochodowym (np. projektowanie tras transportowych podczas kompletowania części przeznaczonych do montażu),
- elektronicznym (np. montaż elementów urządzeń elektronicznych),
- ciężkim (np. produkcja odkuwek w hucie),
- budownictwie (np. produkcja prefabrykatów żelbetowych, bloczków z betonu komórkowego).

#### **3.1. Wybrane podstawowe pojęcia i definicje teorii szeregowania zadań**

Zadania w teorii szeregowania zadań polegają na wykonaniu ciągu czynności zwanych operacjami, z których każda wymaga użycia określonych zasobów [134]. Zadaniem może być np. proces obróbki detalu, proces montażu produktu, realizacja inwestycji itd. Zasobami

natomiast są urządzenia, personel, grupy robocze, materiały, surowce czy kapitał. Posiadają one takie cechy jak: dostępność (czasowe przedziały dostępności), ilość, koszt, podzielność, przywłaszczalność. Dzielią się na następujące kategorie: odnawialne (np. maszyny, procesory, roboty), nieodnawialne (np. surowce, materiały podlegające zużyciu), podwójnie ograniczone (np. energia, kapitał). W teorii szeregowania zadań wykorzystywany jest najczęściej specyficzny rodzaj zasobów odnawialnych nazywanych maszynami (np. urządzenia przemysłowe, procesory, grupy robocze, roboty przemysłowe).

Większość problemów optymalizacyjnych analizowanych w teorii szeregowania zadań można opisać za pomocą, zaproponowanej m.in. w pracy [36], standardowej notacji trójpolowej  $\alpha|\beta|\gamma$ . Symbole lub wyrażenia w tej notacji oznaczają:

- $\alpha$  - typ zagadnienia,
- $\beta$  - dodatkowe ograniczenia zagadnienia,
- $\gamma$  - postać funkcji celu.

Symbol  $\alpha$ , który oznacza sposób przejścia zadań przez system maszyn, może być m.in. następująco określony [134]:

- **F** - przepływowy (*flow-shop*) - wszystkie zadania mają jednakową marszrutę technologiczną (kolejność przejścia przez maszyny), zaś każda z maszyn wymaga określenia oddzielnej sekwencji wprowadzania zadań,
- **FP** - przepływowy permutacyjny (*permutation flow-shop*) - założenie jak w systemie przepływowym z dodatkowym wymaganie, aby kolejność wprowadzania zadań na wszystkich maszynach była jednakowa (zgodna z kolejnością wprowadzania zadań do systemu),
- **J** - gniazdowy (*job-shop*) - różne zadania mogą posiadać różne, co do liczby jak i kolejności odwiedzania stanowisk, marszruty technologiczne,
- **O** - otwarty (*open-shop*) – operacje zadań nie mają określonej kolejności technologicznej.

Druga z cech (symbol  $\beta$ ) dotyczy dodatkowych założeń i ograniczeń, które mogą być następujące [63], [134] m.in.:

- **prec** – istnieje narzucony częściowy porządek technologiczny wykonywania zadań, który może przyjmować postać grafu,
- występują żądane terminy rozpoczęcia  $r_j$  i zakończenia wykonywania  $d_j$  zadań,
- **no idle** (bez przerwy w pracy maszyn) – poszczególne maszyny wykonują zadania bez zbędnych przerw,
- **no wait** (bez czekania) – poszczególne zadania są wykonywane bez przerwy,
- **no store** (bez magazynowania) – brak jest możliwości składowania zadania na składowisku międzyoperacyjnym,

- **pmtn** (przerywać) – dopuszcza się możliwość przerywania wykonywania operacji,
- **setup** (przebrojenia) – występują dodatkowe czasy pomiędzy wykonywaniem zadań, które są wymagane ze względu na przygotowanie maszyn do realizacji różnych zadań oraz mogą być zależne od sekwencji czynności (kolejności zadań) – **seq setup**,
- transport – występują dodatkowe czasy dla operacji, związane z przeniesieniem zadania z maszyny poprzedniej na następną; czasy te mogą przyjmować wartości ujemne (**time lag**) lub dodatnie (**transfer time**),
- ograniczenia zasobowe – występują ograniczenia na ilość zasobów dysponowanych do wykonania operacji na poszczególnych maszynach ( $\sum u_j \leq \hat{U}$ ),
- ograniczenia czasu zakończenia wszystkich zadań [63] – ( $C_{max} \leq \hat{C}$ ).

Trzecia z cech (symbol  $\gamma$ ) wiąże się z przyjętym kryterium optymalizacji (funkcją celu) [63], [134] m.in.:

- $C_{max}$  (*makespan*) – termin zakończenia wykonania wszystkich zadań:

$$C_{max} = \max_{1 \leq j \leq n} C_j,$$

gdzie:  $n$  – ilość zadań,  $C_j$  – termin zakończenia wykonania zadania  $j$ ,

- kryteria związane z terminami zakończenia wykonywania zadań: suma terminów zakończenia wykonywania wszystkich zadań, średni termin zakończenia wykonywania wszystkich zadań oraz kryteria ważone uwzględniające pilność różnych zadań,
- $L_{max}$  (*maximum lateness*)- maksymalna nieterminowość zakończenia wykonywania zadań (gdy są określone pożądane terminy zakończenia wykonywania zadań  $d_j$ ):

$$L_{max} = \max_{1 \leq j \leq n} L_j = \max_{1 \leq j \leq n} (C_j - d_j),$$

- kryteria związane z nieterminowością zakończenia wykonywania zadań: suma nieterminowości zakończenia wykonywania zadań, ważona suma nieterminowości zakończenia wykonywania zadań, średnia nieterminowość zakończenia wykonywania zadań,
- kryteria związane z oceną precyzji zakończenia wykonywania zadań np. maksymalna kara za odchylenie od żądanego przedziału wykonywania zadania  $[r_j, d_j]$ ,
- kryterium zasobowe [63] związane z ilością zasobów dysponowanych do wykonania operacji na poszczególnych maszynach ( $\sum u_j$ ).

Rozwiązaniem każdego problemu jest harmonogram rozpoczęcia i zakończenia wykonywania wszystkich operacji na poszczególnych maszynach. Zadanie optymalizacyjne polega na znalezieniu takiego rozwiązania danego problemu, aby funkcja celu osiągnęła wartość optymalną (zazwyczaj minimalną).

Poniżej zostaną przedstawione szczegółowo dwa problemy teorii szeregowania zadań: problem przepływowy z użyciem jednej maszyny w danym stanowisku i wielu równolegle używanych maszyn w stanowiskach. Stanowią one bazę wyjściową do modelowania zagadnień bardziej skomplikowanych, z dodatkowymi ograniczeniami i kryteriami. Na ich podstawie będą formułowane w rozprawie modele harmonogramowania przedsięwzięć wieloobektowych wykorzystujące system pracy potokowej.

### 3.2. Problem przepływowy

Problem przepływowy, stosujący tylko jedną maszynę do wykonywania danego rodzaju operacji, ma przedstawione następujące założenia. W systemie znajduje się  $m$  maszyn, na których należy wykonać  $n$  zadań produkcyjnych [134], [108]:

- zbiór maszyn:  $M = \{M_1, M_2, M_3, \dots, M_k, \dots, M_m\}$ ,
- zbiór zadań produkcyjnych:  $J = \{J_1, J_2, J_3, \dots, J_j, \dots, J_n\}$ .

Każde zadanie  $J_j \in J$  składa się z  $m$  operacji ze zbioru:

$$O_j = \{O_{j1}, O_{j2}, O_{j3}, \dots, O_{jk}, \dots, O_{jm}\},$$

które są wykonywane w takiej samej kolejności na wszystkich maszynach w systemie. Zakłada się, że operacja  $O_{jk} \in O_j$  jest realizowana nieprzerwanie na maszynie  $M_k$  przez czas  $p_{jk} > 0$  (zbiór czasów trwania  $p_j$  operacji ze zbioru  $O_j$ :  $p_j = \{p_{j1}, p_{j2}, p_{j3}, \dots, p_{jk}, \dots, p_{jm}\}$ ). Zakłada się ponadto, że w dowolnej chwili każda maszyna może wykonywać tylko jedną operację oraz to, że tylko jedna operacja danego zadania może być wykonywana w danej chwili w systemie. Powyższe założenia są parametrami i ograniczeniami modelu optymalizacyjnego problemu przepływowego. Zadanie optymalizacyjne polega na znalezieniu uszeregowania  $S$  (harmonogramu) określonego przez podanie terminów rozpoczęcia operacji oraz takiego, który optymalizuje (minimalizuje lub maksymalizuje) wartość funkcji celu, spełniając założone ograniczenia.

Zmienną decyzją w przedmiotowym problemie jest kolejność wykonywania operacji na maszynach. Jest ona opisana przez zestaw  $m$ -permutacji:

$$\pi = (\pi_1, \pi_2, \dots, \pi_i, \dots, \pi_m),$$

gdzie:  $\pi_i = (\pi_i(1), \pi_i(2), \dots, \pi_i(k), \dots, \pi_i(n)) \in P(N)$ , gdzie  $N = \{1, 2, 3, \dots, n\}$ ,  $P(N)$  oznacza zbiór wszystkich permutacji zbioru  $N$ . Wszystkie możliwe kolejności zawarte są w zbiorze  $\Pi = \{\pi = (\pi_1, \pi_2, \dots, \pi_i, \dots, \pi_m) : \pi_i \in P(N), M_k \in M\}$ . Liczność zbioru wszystkich permutacji dla danego problemu o rozmiarach  $n \times m$  wynosi  $(n!)^m$ .

Szczególnym przypadkiem problemu przepływowego jest sytuacja, gdy kolejność realizacji zadań na poszczególnych maszynach jest taka sama (w przypadku ogólnym

kolejność wykonywania zadań na maszynach może być różna). Przypadek ten nazywany jest problemem przepływowym permutacyjnym. Zmienną decyzyjną w tym przypadku jest kolejność wykonywania zadań na wszystkich maszynach, która dla wszystkich maszyn jest taka sama:

$$\begin{aligned}\pi_i(j) &= \pi_{i+1}(j), \\ i &= 1, \dots, m-1, j = 1, \dots, n.\end{aligned}$$

Kolejność wykonywania zadań jest wtedy następująca:

$$\pi = (\pi(1), \pi(2), \dots, \pi(k), \dots, \pi(n)).$$

Liczność zbioru wszystkich permutacji w tym przypadku, w problemie przepływowym permutacyjnym o rozmiarach  $n \times m$ , wynosi  $n!$ . Taka sama kolejność wykonywania zadań na maszynach wynika często z ograniczeń technologicznych systemu produkcyjnego np. występowanie taśmy produkcyjnej lub innych dodatkowych ograniczeń. Okazuje się ponadto, że dla pewnych problemów szeregowania rozwiązania optymalne leżą w klasie problemów permutacyjnych.

Problemy przepływowe z punktu widzenia złożoności obliczeniowej algorytmów optymalizacyjnych już dla trzech maszyn należą do problemów silnie NP-trudnych, co pokazano w pracy [28]. Rozważany problem jest rozwiązywalny w sposób dokładny tylko w przypadku wystąpienia dwóch maszyn i kryterium  $C_{max}$  (problem FP2 | |  $C_{max}$ ) za pomocą algorytmu Johnsona [69] w czasie rzędu  $O(n \log n)$ . Aktualny stan badań dotyczący rozwiązywania tego problemu z najczęściej występującą funkcją celu  $C_{max}$  zostanie podany w p. 3.5.

### 3.3. Problem przepływowi z maszynami równoległymi

Problem przepływowi z maszynami równoległymi polega na wykonaniu  $n$  zadań produkcyjnych ze zbioru  $J$ , na maszynach znajdujących się w  $m$  stanowiskach (centrach) obróbczych [100], [108], [134]:

- zbiór zadań produkcyjnych:  $J = \{J_1, J_2, J_3, \dots, J_j, \dots, J_n\}$ ,
- zbiór stanowisk (centrów) obróbczych:  $R = \{R_1, R_2, R_3, \dots, R_k, \dots, R_m\}$ .

W każdym stanowisku  $R_k \in R$  znajduje się  $m_k \geq 1$  identycznych maszyn ze zbioru  $M_k$ :

$$M_k = \{M_{k1}, M_{k2}, M_{k3}, \dots, M_{km_k}\} \subset M.$$

Każde zadanie  $J_j \in J$  składa się z  $m$  operacji ze zbioru:

$$O_j = \{O_{j1}, O_{j2}, O_{j3}, \dots, O_{jk}, \dots, O_{jm}\},$$

które są wykonywane w takiej samej kolejności, w każdym ze stanowisk ze zbioru  $R$ . Zbiór  $O = \bigcup_{j=1}^n \bigcup_{k=1}^m O_{jk}$  określa wszystkie operacje do wykonania w systemie. Zakłada się, że operacja  $O_{jk} \in O_j$  jest realizowana nieprzerwanie na jednej, wybranej maszynie ze zbioru  $M_k$  przez czas  $p_{jk} > 0$  (zbiór czasów trwania  $p_j$  operacji ze zbioru  $O_j : p_j = \{p_{j1}, p_{j2}, p_{j3}, \dots, p_{jk}, \dots, p_{jm}\}$ ). Podobnie jak w problemie przepływowym opisanym w poprzednim podrozdziale zakłada się, że w dowolnej chwili każda maszyna może wykonywać tylko jedną operację oraz to, że tylko jedna operacja danego zadania może być wykonywana w danej chwili w systemie. W celu rozwiązania danego problemu należy podać uszeregowanie  $(S, A)$ , gdzie:

$$S = [S_{jk}]_{n \times m}, \quad A = [a_{jk}]_{n \times m}.$$

$S_{jk}$  oznacza najwcześniejszy termin rozpoczęcia wykonywania operacji  $O_{jk}$  zadania  $J_j$  w stanowisku  $R_k$ ,  $a_{jk}$  oznacza numer maszyny przydzielonej do wykonania operacji  $O_{jk}$  zadania  $J_j$  w stanowisku  $R_k$ .

Zbiór  $O$  wszystkich operacji w danym problemie dzieli się na  $m$  podzbiorów operacji realizowanych w tym samym stanowisku  $O = \bigcup_{k=1}^m O_k$ ,  $O_k = \{O_{jk} : J_j \in J\}$ . Każdy z podzbiorów  $O_k$  dzieli się na  $m_k$  podzbiorów operacji wykonywanych na tych samych maszynach  $O_k = \bigcup_{i=1}^{m_k} O_{ki}$ ,  $O_{ki} = \{O_{jk} : J_j \in J \wedge a_{jk} = i\}$ .

Kolejność wykonywania operacji realizowanych na maszynie  $M_{ki}$  opisywana jest za pomocą permutacji:

$$\pi_{ki} = (\pi_{ki}(1), \pi_{ki}(2), \dots, \pi_{ki}(l), \dots, \pi_{ki}(n_{ki})) \in P(N_{ki}),$$

gdzie  $l = 1, \dots, n_{ki}$ ,  $P(N_{ki})$  – zbiór wszystkich permutacji zbioru  $N_{ki}$ ,  $N_{ki} = \{j : O_{jk} \in O_{ki}\}$  – zbiór numerów zadań realizowanych na maszynie  $M_{ki}$  w stanowisku  $R_k$ . Kolejność wykonywania zadań w każdym stanowisku określa permutacja podziału rozłącznego  $\pi_k = (\pi_{k1}, \pi_{k2}, \dots, \pi_{ki}, \dots, \pi_{km_k})$ . Kolejność wykonywania zadań we wszystkich stanowiskach określa permutacja (ściślej: zestaw permutacji podziału rozłącznego [98]):  $\pi = (\pi_1, \pi_2, \dots, \pi_k, \dots, \pi_m)$ . Możliwe kolejności wykonywania operacji wszystkich zadań zawarte są w zbiorze:

$$\Pi = \{\pi = (\pi_{ki} \in P(N_{ki}), N_{ki} = \{j : O_{jk} \in O_{ki}\}) \wedge O_{ki} \in O_k, i = 1, 2, \dots, m_k, k = 1, 2, \dots, m\},$$

gdzie  $O_{ki}$  jest podziałem zbioru  $O_k$ .

Problem przepływowy z maszynami równoległymi jest uogólnieniem problemu przepływowego przedstawionego w poprzednim podrozdziale. Pod względem złożoności obliczeniowej algorytmów optymalizacyjnych jest to również problem NP-trudny, poza nielicznymi przypadkami szczególnymi. Aktualny stan badań dotyczący rozwiązywania tego problemu z najczęściej występującą funkcją celu  $C_{\max}$  zostanie podany w podrozdziale 3.5.

### 3.4. Metody rozwiązywania problemów szeregowania zadań

Zagadnienia szeregowania zadań są problemami optymalizacji dyskretnej ze względu na to, że zmienne decyzyjne przyjmują wartości dyskretne, całkowitoliczbowe lub binarne. Są to zagadnienia trudniejsze z obliczeniowego punktu widzenia niż np. problemy optymalizacji liniowej, ze względu na m.in.: częsty brak własności analitycznych (np. różniczkowalności, liniowości itp.), wieloekstremalność ze znaczną liczbą ekstremów lokalnych [134]. Podstawową przeszkodą napotykaną przy rozwiązywaniu tych problemów jest to, że większość z nich jest NP-trudna [134]. W związku z tym stosowane są algorytmy, za pomocą których uzyskujemy wynik dokładny (wykładniczo-zależne dla problemów NP-trudnych) lub dające wynik zbliżony do optymalnego - algorytmy przybliżone [134].

Do grupy algorytmów dokładnych należą [34], [134]:

- efektywne algorytmy dedykowane,
- metody wykorzystujące schemat podziału i ograniczeń,
- metody wykorzystujące schemat programowania dynamicznego,
- metody wykorzystujące programowanie liniowe całkowitoliczbowe,
- metody wykorzystujące programowanie liniowe binarne,
- metody subgradientowe.

Efektywne algorytmy dedykowane mają wielomianową złożoność obliczeniową i stosowane są najczęściej dla problemów z klasy P (rozwiązywalnych w czasie zależnym od wielomianu) np. algorytm Johnsona dla rozwiązywania problemu  $FP2 \mid C_{\max}$  [69]. Pozostałe wymienione powyżej metody są zalecane dla problemów NP-trudnych. Są to jednak metody czasochłonne i nie dla wszystkich problemów efektywne [134].

W grupie algorytmów, za pomocą których uzyskuje się wynik przybliżony, można wyróżnić [34], [134]:

- algorytmy konstrukcyjne,
- algorytmy lokalnego przeszukiwania, zwane algorytmami poprawiającymi lub iteracyjnymi.

Algorytmy konstrukcyjne działają na zasadach heurystycznych szeregując według ustalonej kolejności zadania danego problemu [134]. Czas ich działania jest stosunkowo krótki ze względu na wielomianową złożoność obliczeniową. Dostarczają często jednak rozwiązań odległych od optymalnych. Wykorzystują metody: priorytetowe, wstawień, relaksacji (sprowadzenie do problemu łatwiejszego), kombinowane [134]. Obecnie są stosowane do znajdowania rozwiązań startowych dla innych algorytmów. Algorytmy poprawiające są dużą grupą metod, których celem nadrzędnym jest poprawianie w danej iteracji aktualnego



rozwiązania bazowego przez wybranie rozwiązania z jego otoczenia (podzbioru rozwiązań dopuszczalnych) o wartości funkcji celu lepszej niż dla bazowego. Algorytm kończy działanie po spełnieniu warunków: limitu czasu, liczby iteracji, osiągnięciu zadowalającej wartości funkcji celu lub wartości optymalnej. Z grupy algorytmów poprawiających należy wyróżnić grupę algorytmów metaheurystycznych, które wykorzystują mechanizmy poszukiwań analogicznych do zjawisk fizycznych lub występujących w naturze. Do nich należy zaliczyć algorytmy m.in.: przeszukiwania zstępującego (*descending search*), przeszukiwania z zabronieniami (*tabu search*), przeszukiwania ewolucyjnego (*evolutionary search*), symulowanego wyżarzania (*simulated annealing*), przeszukiwania rozproszonego (*scatter search*), przeszukiwania mrówkowego (*ant search*), algorytmy memetyczne (hybrydowe).

Istotną cechą zdecydowanej większości algorytmów poprawiających, a szczególnie metaheurystycznych, jest ich uniwersalność i elastyczność zastosowania do rozwiązywania zagadnień z różnymi ograniczeniami i kryteriami. Wynikają one z tego, że forma danego algorytmu jest zależna tylko od postaci zmiennej decyzyjnej. Zmiana kryterium i ograniczeń nie wpływa na formę algorytmu. Jest to ważna cecha odróżniająca te algorytmy od stosowanego często do tej pory, w potokowych metodach organizacji przedsięwzięć budowlanych, algorytmu podziału i ograniczeń, który przy każdorazowej zmianie ograniczeń lub parametrów wymaga ponownego, odpowiedniego formułowania funkcji dolnych i górnych ograniczeń.

### 3.5. Aktualny stan badań dotyczący rozwiązywania problemów przepływowych

Jak wspomniano w p. 3.2, problem przepływowy z najczęściej występującą funkcją celu  $C_{max}$ , jest NP-trudny już dla trzech maszyn [134]. Wprowadzanie dodatkowych ograniczeń, np. bez czekania, bez magazynowania, oraz rozpatrywanie innych funkcji celu nie zmniejsza złożoności obliczeniowej [134]. W grupie algorytmów dokładnych dla problemu  $FP \mid \mid C_{max}$  (jest to najczęściej stosowany problem testowy dla nowoutworzonych algorytmów) najczęściej wymienia się obecnie metody wykorzystujące schemat podziału i ograniczeń (B&B) [134]. Przeprowadzono wiele badań z użyciem tych metod [4], [10], [33], [57], [79]. Wyniki obliczeniowe w praktycznie użytecznym czasie są możliwe do uzyskania tylko dla stosunkowo niewielkich rozmiarów zadań (do 20 zadań i 5 maszyn). W grupie algorytmów przybliżonych mamy natomiast następujące ich rodzaje:

- konstrukcyjne: P [105], G [37], CDS [12], RA [19], NEH [97],
- poprawiające:
  - przeszukiwania z zabronieniami (TS) [35], [99], [140], [145],

- przeszukiwania genetycznego (GS) [118], [119],
- symulowanego wyżarzania (SA) [61], [103], [104],
- przeszukiwania zstępującego [19].

Metody powstałe na bazie algorytmu TS są jednymi z najsilniejszych narzędzi do rozwiązywania problemów optymalizacji dyskretnej, w tym problemów przepływowych [134]. Ich przeciętny błąd funkcji celu odniesiony do wartości optymalnej dla problemu  $FP \mid C_{max}$  może osiągać wartości do ok. 1,5% [35], [99]. Interesujące podejścia do zastosowań algorytmu TS w problemach teorii szeregowania zadań zawierają prace [140], [145]. Szczególnie duże znaczenie dla rozwoju tego rodzaju algorytmów dla problemów przepływowych miały prace [98], [99]. Dobre wyniki obliczeniowe z użyciem algorytmów GS dla problemów przepływowych z zastosowaniem operatorów genetycznych uzyskano w pracy [118], a szczególnie [119], gdzie został wykorzystany operator MSFX łączący cechy operatorów genetycznych i poszukiwań lokalnych.

Większe trudności w rozwiązywaniu zadań optymalizacyjnych są odnotowywane dla problemu przepływowego z maszynami równoległymi. W grupie metod dokładnych dla tego problemu, z funkcją celu  $C_{max}$ , można wymienić algorytmy: podziału i ograniczeń [132], programowania całkowitoliczbowego [134]. W grupie algorytmów przybliżonych można wymienić algorytmy:

- konstrukcyjne: DK1, DK2, DK3 [20], W [146], S [129], HS [56],
- typu popraw:
  - przeszukiwania z zabronieniami (TS) [100],
  - przeszukiwania genetycznego (GS) [143],
  - symulowanego wyżarzania (SA) [107],
  - metodę wstawień [9],
  - przeszukiwania zstępującego [96], [107].

W pracy [100], badając grupę algorytmów przybliżonych, najlepsze rezultaty, w porównaniu do algorytmów konstrukcyjnych, otrzymano stosując algorytm TS.

Aktualnie dobre rezultaty dla opisanych problemów przepływowych osiągają algorytmy będące połączeniem wyżej wymienionych metod - tzw. algorytmy memetyczne (hybrydowe) np. [8].

### 3.6. Zastosowania zagadnień teorii szeregowania zadań w budownictwie

Dotychczasowe zastosowania teorii szeregowania zadań w budownictwie związane są głównie z produkcją elementów prefabrykowanych, a ponadto w mniejszym zakresie planowaniem realizacji niektórych przedsięwzięć budowlanych. Zastosowania te dotyczą przypadków, w których występuje analogia do przemysłowych procesów produkcji (systemy pracy przepływowej zwane również pracą taśmową).

Zagadnienie produkcji prefabrykowanych elementów żelbetowych zostało przedstawione w pracy [80]. Autorzy stosują do opisu prefabrykacji metodę LOB. Zauważają, że występujące różnice czasów realizacji poszczególnych czynności związanych z wykonaniem elementu prefabrykowanego wymagają potraktowania badanego zagadnienia jako problemu przepływowego permutacyjnego. Jako funkcję celu przyjmują czas realizacji danej grupy elementów. Prowadzi to do złamania zasady ciągłości wykonywania prac przez grupy robocze jednego rodzaju (podobnie jak w [130]). W opisywanym problemie prefabrykacji występują ograniczenia w dostępie do zasobów (robotników, suwnic). Zmiennymi decyzyjnymi są: kolejność realizacji elementów prefabrykowanych i alokacja zasobów do czynności. Alokacja zasobów odbywa się na zasadzie FIFO. W rozpatrywanym przykładzie autorzy analizują różne wielkości zasobów (liczbę robotników z zasobów własnych wykonawcy i z zasobów zewnętrznych, liczbę suwnic). Celem tej analizy jest określenie wpływu zastosowania dodatkowych zasobów (liczby pracowników z zasobów zewnętrznych) na czas realizacji zadania. Do rozwiązania zagadnienia optymalizacyjnego stosowany jest algorytm genetyczny, który w populacji początkowej stosuje rozwiązania uzyskane za pomocą algorytmu konstrukcyjnego NEH [97].

Harmonogramowanie produkcji elementów prefabrykowanych z betonu komórkowego jest tematem pracy [108]. Proces technologiczny składa się z ośmiu czynności wykonywanych dla partii wyrobów różniących się między sobą m.in. wielkością. Autor rozważa systemy produkcyjne, w których znajduje się linia z tylko jedną maszyną w każdym stanowisku lub linia z wieloma maszynami w stanowisku (dla grup czynności od mieszania masy betonowej do cięcia związanej masy oraz proces autoklawizacji i transport do magazynu). W technologicznym ciągu czynności znajdują się również takie, dla których zakłada się brak ograniczeń w liczbie maszyn potrzebnych do ich realizacji (przygotowanie surowców, kontrola jakości, transport do odbiorcy). Związane jest to z wykonywaniem tych czynności przez zasoby, które są tanie i łatwe do pozyskania ze źródeł zewnętrznych (dodatkowe środki transportowe, zatrudnienie dodatkowych pracowników). Pomędzy poszczególnymi czynnościami występują ograniczenia wynikające z technologii: ograniczenie

„bez czekania” między czynnościami od mieszania masy betonowej do cięcia związanej masy oraz ograniczenie „bez magazynowania” między autoklawizacją i transportem wyrobów do magazynu. W związku z ograniczeniem liczby maszyn dla niektórych czynności istnieje potrzeba szeregowania zadań na maszynach. Zadanie optymalizacji polega na przyporządkowaniu zadań (wyrobów z poszczególnych partii) do maszyn na stanowiskach oraz ustaleniu kolejności wykonywania zadań (wyrobów). Jako kryterium przyjęto minimalizację czasu trwania wykonywania wszystkich partii wyrobów. Autor skupia się nad wykryciem szczególnych własności tego problemu, które uwzględni w aplikacjach algorytmów: konstrukcyjnych i poprawiających (m.in.: TS, GS, SA).

Rozdysponowanie mocy przerobowej przedsiębiorstwa budowlanego jest zagadnieniem przedstawionym w pracy [62]. Autor rozważa model systemu pracy przedsiębiorstwa budownictwa mieszkaniowego, które zgodnie z harmonogramem ma za zadanie wybudować określoną liczbę budynków w osiedlach mieszkaniowych. Dane przedsiębiorstwo posiada zdolności przerobowe wyrażone przez możliwość oddania do użytku określonej liczby tzw. umownych izb lub  $m^2$  powierzchni mieszkalnej w jednostce czasu. Wielkość ta jest zależna od liczby zatrudnionych pracowników oraz od wielkości parku maszynowego przedsiębiorstwa. Założono, że jego moc przerobowa jest odcinkami stałą funkcją czasu. Ponadto przyjęto, że każdy budynek ma wyznaczone dyrektywne terminy rozpoczęcia i zakończenia realizacji, które wyznacza się zgodnie z istniejącymi i przewidywanymi warunkami przedsiębiorstwa (mocą przerobową). Zadanie polega na rozdysponowaniu mocy przerobowej tak, aby budynki zostały zrealizowane w ramach terminów dyrektywnych oraz, aby przyjęte kryterium - czas wznoszenia wszystkich budynków przewidzianych do oddania osiągnął wartość minimalną. Do utworzenia modelu przedsiębiorstwa zostały użyte dynamiczne (różniczkowe) modele operacji typu prędkość/zasób ciągły [63]. Zakłada się w nich, że ilość zasobu może być zmienna w trakcie wykonywania operacji zadania (robót w budynkach) i mieć wpływ na prędkość wykonywania operacji. Ten model lepiej odzwierciedla sytuacje praktyczne, w odróżnieniu od modeli statycznych typu czas/zasób, w których ilość zasobu jest ustalana przed realizacją operacji i nie może być zmieniona w czasie wykonywania operacji.

### **3.7. Metody sprzężeń czasowych na tle teorii szeregowania zadań**

Analiza dokonana przez autora niniejszej pracy, dotycząca metod sprzężeń czasowych, wskazuje na istotny związek teorii szeregowania zadań (TSZ) z zagadnieniami harmonogramowania robót budowlanych. Metody sprzężeń czasowych, które umożliwiają

planowanie w systemie pracy potokowej przedsięwzięć budowlanych (aktualny stan badań nad nimi został przedstawiony w p. 2.2), wykazują analogię do niektórych zagadnień rozpatrywanych w ramach TSZ. Dla zobrazowania tego związku zostanie poniżej podany matematyczny model tych metod i porównany do modeli przyjętych w TSZ.

Model matematyczny metod sprzężeń czasowych ma charakter dyskretny, tak jak w TSZ. Parametry tego modelu mają charakter deterministyczny i są następujące:

- kompleks działek roboczych (obiektów):  $Z = \{Z_1, Z_2, Z_3, \dots, Z_j, \dots, Z_n\}$ ,
- zbiór grup roboczych:  $B = \{B_1, B_2, B_3, \dots, B_i, \dots, B_m\}$ ,
- zbiory robót:  $P_j = \{p_{1j}, p_{2j}, p_{3j}, \dots, p_{mj}\}$  realizowanych przez zbiór grup roboczych  $B$  na działkach  $Z_j \in Z$  (gdzie  $p_{1j}, \dots, p_{mj}$  – czasy trwania robót).

Czasy trwania robót  $p_{1j}, \dots, p_{mj}$  są funkcjami pracochłonności robót na działkach (obiektach) i wydajności grup roboczych.

Obszary rozwiązań dopuszczalnych określają:

- wynikająca z technologii kolejność robót:

$$p_{k-1,j} < p_{k,j} < p_{k+1,j} ,$$

- warunki dotyczące pracy grup roboczych na działkach:
  - w tym samym momencie dana grupa robocza może wykonywać tylko jedną robotę,
  - w tym samym momencie na jednej działce można wykonywać tylko jeden rodzaj robót (oznacza to, że wartości sprzężeń czasowych wszystkich rodzajów są zawsze nieujemne),
- warunki dotyczące sprzężeń czasowych, które można nakładać alternatywnie, a mianowicie:
  - ciągłość wykonywania jednego rodzaju robót (metoda I z zerowymi sprzężeniami między środkami realizacji),
  - ciągłość wykonywania robót na działce (metoda II z zerowymi sprzężeniami między działkami roboczymi),
  - brak dodatkowych założeń (metoda III ze sprzężeniami między środkami realizacji i między działkami roboczymi),
  - dodatkowe warunki dotyczące sprzężeń diagonalnych i odwrotnie diagonalnych (metody IV – VI uwzględniające te sprzężenia).

Zmienną decyzyjną jest kolejność realizacji działek roboczych (obiektach) ze zbioru  $Z$ .

Rozwiązaniem danego problemu jest harmonogram realizacji robót uwzględniającego przyjętą kolejność realizacji działek roboczych (obiektów).

Funkcją celu jest czas  $T$  realizacji wszystkich robót na wszystkich działkach roboczych (obiektach), przy czym poszukuje się wartości minimalnej tego czasu.

Zadanie optymalizacyjne dotyczące kolejności realizacji działek roboczych (obiektów) jest aktualnie rozwiązywane za pomocą metod wykorzystujących schemat podziału i ograniczeń (B&B).

Tabela 3.1. Równoważność pojęć modelu permutacyjnego przepływowego w TSZ i analogicznych pojęć wykorzystywanych w metodach sprzężeń czasowych

Pojęcia modelu permutacyjnego przepływowego w TSZ	Pojęcia w metodach sprzężeń czasowych
Zbiór zadań do wykonania: $J = \{J_1, J_2, J_3, \dots, J_n\}$	Kompleks działek roboczych: $Z = \{Z_1, Z_2, Z_3, \dots, Z_n\}$
Zbiór różnych typów maszyn potrzebnych do wykonania zbioru zadań $J$ : $M = \{M_1, M_2, M_3, \dots, M_n\}$	Zbiór grup roboczych: $B = \{B_1, B_2, B_3, \dots, B_m\}$
Każde zadanie $J_i \in J$ składa się z ciągu operacji wykonywanych na maszynach ( $n_i$ – liczba operacji dla zadania $i$ ): $J_i = \{O_{i1}, O_{i2}, O_{i3}, \dots, O_{in_i}\}$	Dla wykonania działki $Z_i \in Z$ potrzebne jest wykonanie zbioru robót: $Z_i = \{P_{i1}, P_{i2}, P_{i3}, \dots, P_{im}\}$
Operacje mogą być wykonane w ciągu technologicznym: $O_{i,k-1} < O_{i,k} < O_{i,k+1}$	Dla każdej działki obowiązuje wynikająca z technologii kolejność: $P_{i,k-1} < P_{i,k} < P_{i,k+1}$
Ograniczenie „no-idle” (bez przerwy w pracy maszyn)	Założenie metody I - zerowe sprzężenia między środkami realizacji
Ograniczenie „no-wait” (bez czekania)	Założenie metody II - zerowe sprzężenia między działkami roboczymi
Brak dodatkowego ograniczenia	Założenie metody III - równoczesne uwzględnianie sprzężeń między środkami realizacji i działkami roboczymi
Zmienna decyzyjna: permutacja $\pi$ zadań ze zbioru $J$	Zmienna decyzyjna: kolejność wykonywania działek ze zbioru $Z$
Funkcja celu: $C_{max}$ (makespan)	Funkcja celu: czas $T$ realizacji wszystkich robót na wszystkich działkach roboczych

Z przedstawionej charakterystyki modelu metod sprzężeń czasowych i założeń modelu permutacyjnego przepływowego rozważanego w TSZ (p. 3.2) wynika, że istnieje równoważność parametrów, ograniczeń, zmiennej decyzyjnej i przyjętego kryterium tych dwóch modeli, co zostało przedstawione w tabeli 3.1.

W związku z przedstawioną analogią tych dwóch modeli poszczególne metody sprzężeń czasowych można opisać jako modele TSZ stosując notację trójpolową  $\alpha|\beta|\gamma$  (wg p. 3.1):

- metoda I z zerowymi sprzężeniami między środkami realizacji – problem  $FP|no-idle|C_{max}$ ,
- metoda II z zerowymi sprzężeniami między działkami robót – problem  $FP|no-wait|C_{max}$ ,
- metoda III z równoczesnym uwzględnieniem sprzężeń między środkami realizacji i działkami roboczymi – problem  $FP||C_{max}$ ,

### 3.8. Podsumowanie

W niniejszym rozdziale przedstawiono podstawowe założenia, definicje i pojęcia teorii szeregowania zadań. Szczególny nacisk położono na przedstawienie problemów przepływowych z użyciem tylko jednej maszyny w stanowisku i z wieloma maszynami w stanowisku. Podano również aktualny stan badań nad metodami rozwiązywania problemów optymalizacyjnych, a szczególnie dwóch wymienionych powyżej problemów przepływowych. Z przeprowadzonej analizy wynika, że model metod sprzężeń czasowych, dla licznych przypadków przedsięwzięć budowlanych realizowanych w systemie pracy potokowej, jest odpowiedni dla ich opisu, jednakże nie jest on pozbawiony niedostatków, które utrudniają analizę przypadków bardziej złożonych i nietypowych. Uzasadniono również, że metody sprzężeń czasowych wykazują analogię do problemów przepływowych w TSZ. W związku z tym postanowiono w ramach niniejszej pracy wykorzystać aktualne osiągnięcia TSZ w ramach problemów przepływowych w celu zaproponowania nowych modeli planowania robót budowlanych, nie występujących dotychczas w modelach z systemem pracy potokowej (m.in. w metodach sprzężeń czasowych). Należy oczekiwać, iż modele te będą lepiej odzwierciedlać niektóre rzeczywiste sytuacje w produkcji budowlanej. Dla ujednolicenia zapisu modele formułowane będą zgodnie z metodologią stosowaną w TSZ. Analizowane nowe modele będą uwzględniały:

- możliwość użycia dodatkowych zasobów (grup roboczych) w realizacji robót (zastosowanie problemu przepływowego z maszynami równoległymi),

- możliwość uwzględniania dodatkowych ograniczeń występujących pomiędzy robotami jednego rodzaju i robotami jednego obiektu (działki roboczej),
- możliwość rozpatrywania innych kryteriów optymalizacyjnych np. koszt/czas (zagadnienie TCT),
- użycie uniwersalnych i elastycznych w zastosowaniu algorytmów metaheurystycznych.

Przewiduje się również rozszerzenie obszaru stosowania utworzonych modeli o przedsięwzięcia, które polegają na realizacji wielu obiektów budowlanych (przedsięwzięcia wieloobiektywne). Cechą takich przedsięwzięć jest istnienie ograniczeń występujących podczas planowania przedsięwzięć typu kompleks operacji (niepowtarzanych): możliwości jednoczesnego (równoległego) prowadzenia robót różnego rodzaju w obiekcie lub wystąpienie relacji kolejnościowych między robotami w obiekcie opisanych grafem, a także występowania dodatkowych zależności między robotami, wynikających z przyjętej technologii. Przedstawione dodatkowe ograniczenia są możliwe do uwzględnienia w nowych modelach przy wykorzystaniu aktualnych osiągnięć TSZ. W następnym rozdziale zostaną przedstawione trzy wybrane przez autora algorytmy metaheurystyczne, które zastosowano do rozwiązywania zadań optymalizacyjnych w modelach sformułowanych w rozdziale 5. Ponadto przedstawiono weryfikację uzyskiwanych rozwiązań na podstawie oceny eksperymentalnej.



#### **4. Metaheurystyczne metody optymalizacji dyskretnej zastosowane w pracy oraz weryfikacja wyników obliczeń uzyskiwanych za ich pomocą**

W nowych modelach harmonogramowania przedsięwzięć wieloobiektywnych, które zostaną przedstawione w rozdziale 5 niniejszej pracy, będą zastosowane wybrane metody optymalizacji dyskretnej. W celu dokonania wyboru rozważono następujące czynniki [134]:

- jakość dostarczanych rozwiązań,
- czas obliczeń prowadzących do rozwiązania konkretnego problemu,
- elastyczność w formułowaniu nowych zagadnień (parametrów, ograniczeń, funkcji celu),
- łatwość implementacji programowej.

Obecnie najlepiej powyższe warunki spełniają algorytmy metaheurystyczne należące do grupy algorytmów poprawiających [134]. Jakość dostarczanych rozwiązań z reguły jest dobra, lecz zależna od rodzaju algorytmu. Podstawową wadą tych algorytmów jest to, że za ich pomocą można uzyskać wynik przybliżony. Często jest on jednak bliski optymalnemu, zaś dla problemów o małych rozmiarach może być uzyskiwany wynik optymalny [134]. Czas obliczeń jest zależny od warunków zakończenia pracy algorytmu (np. liczby iteracji), lecz nie rośnie wykładniczo wraz ze wzrostem rozmiarów zadania, jak w przypadku metod opartych na schemacie B&B [134]. Z reguły są to algorytmy elastyczne w odniesieniu do nowych zagadnień, np. wymagane są tylko zmiany w procedurach obliczania funkcji celu w przypadku ich zmiany, w przeciwieństwie do np. metod opartych na schemacie B&B, które każdorazowo wymagają formułowania funkcji dolnych i górnych ograniczeń, zależnie od postaci rozwiązywanego zagadnienia [134]. Algorytmy poprawiające są na ogół łatwe do implementacji programowej, choć wersje bardziej rozbudowane mogą narażać na kłopoty [134]. Z grupy algorytmów metaheurystycznych, ze względu na jakość dostarczanych rozwiązań, wybrano trzy algorytmy: przeszukiwania z zabronieniami, przeszukiwania genetycznego, symulowanego wyżarzania.

#### **4.1. Metaheurystyczne metody optymalizacji dyskretnej zastosowane w pracy**

##### **4.1.1. Algorytm przeszukiwania z zabronieniami (*taboo search*)**

Metoda przeszukiwania z zabronieniami (TS) została zaproponowana przez F. Glovera w pracach [29], [30]. Powieliła ona naturalny proces poszukiwania rozwiązania problemu realizowany przez człowieka. Podstawowa wersja algorytmu TS rozpoczyna działanie od określonego rozwiązania startowego [134]. Następnie, dla tego rozwiązania znajduje się otoczenie. Otoczenie definiuje się jako zbiór rozwiązań, który można

utworzyć po wykonaniu ruchów w danym rozwiązaniu tj. przekształceń transformujących dane rozwiązanie w inne według ustalonych zasad. W otoczeniu poszukiwane jest rozwiązanie z najmniejszą wartością funkcji celu. To rozwiązanie staje się rozwiązaniem bazowym dla następnej iteracji. Wynikiem działania algorytmu jest najlepsze rozwiązanie z trajektorii poszukiwań.

Istotą metody TS jest zastosowanie historii poszukiwań, co zapobiega zatrzymaniu się trajektorii w ekstremum lokalnym oraz umożliwia wprowadzanie jej w bardziej obiecujące obszary poszukiwań. Najczęściej w tym celu używa się pamięci krótkoterminowej zwanej listą zabronień (lista tabu). Lista ta przechowuje przez ograniczony czas najświeższe atrybuty rozwiązań, ruchy prowadzące do tych rozwiązań lub atrybuty ruchów ostatnio rozpatrywanych rozwiązań. Zazwyczaj jest to dokonywane przez wprowadzenie stałej długości listy, co powoduje usunięcie elementu najstarszego w trakcie dopisywania elementu nowego. Z atrybutów elementów tej listy wynika, że niektóre przyszłe ruchy mogą dobrze nie rokować i dlatego będą traktowane jako zabronione. Zabronienie to może być anulowane, jeśli tzw. funkcja aspiracji uzna dany ruch za korzystny. Jest to dodatkowa funkcja, której wartość dla danego rozwiązania może spowodować, że ruchy prowadzące do rozwiązania lub jego atrybuty nie znajdą się na liście tabu. Warunkami zakończenia pracy algorytmu mogą być: limit czasu, maksymalna liczba iteracji, osiągnięcie zadowalającej wartości funkcji celu lub wartości optymalnej. Poniżej przedstawiony jest ogólny algorytm metody TS zastosowany do rozwiązywania problemów przepływowych [98].

Niech  $\pi \in \Pi$  będzie dowolną permutacją,  $LT$  listą tabu,  $c$  przyjętą funkcją celu, a  $\pi^*$  najlepszym do tej pory znalezionym rozwiązaniem (na początek przyjmujemy za  $\pi^*$  permutację  $\pi$ ).

**Krok 1.** Wyznaczyć otoczenie  $N_\pi$  permutacji  $\pi$  nie zawierające elementów zabronionych przez listę  $LT$ ;

**Krok 2.** Znaleźć permutację  $\delta \in N_\pi$  taką, że:

$$c(\delta) = \min\{c(\beta) : \beta \in N_\pi\};$$

**Krok 3.** Jeśli  $c(\delta) < c(\pi^*)$ , to  $\pi^* \leftarrow \delta$

Umieść atrybuty  $\delta$  na liście  $LT$ ;

$$\pi \leftarrow \delta;$$

**Krok 4.** Jeżeli Warunek\_Zakończenia, to STOP,

w przeciwnym przypadku idź do Kroku 1.

Metoda przeszukiwania z zakazami posiada wiele stopni swobody: wybór formy ruchu i określenia sąsiedztwa, formy realizującej mechanizm zabronień (np. długości listy tabu, formy atrybutów), strategii poszukiwań. Jakość uzyskiwanych rozwiązań zależy od odpowiedniego przyjęcia założeń. Obecnie algorytmy TS należą do najskuteczniejszych narzędzi wykorzystywanych w teorii szeregowania zadań.

#### **4.1.2. Algorytm przeszukiwania genetycznego (*genetic search*)**

Algorytmy przeszukiwania genetycznego (GS) wykorzystują zasady ewolucji w naturze, które prowadzą do najlepszego przystosowania się (optymalizacji) osobników do warunków występujących w danym środowisku [55]. Wykorzystują one populację osobników (rozwiązań), która następnie jest przetwarzana w trakcie selekcji, działań wywołanych przez użycie operatorów genetycznych oraz fazy przetrwania. Ze względów formalnych grupa tego rodzaju algorytmów jest ogólnie nazywana ewolucyjnymi. Ich szczególne przypadki, w których rozwiązanie jest reprezentowane przez wektor binarny, permutację itp., a proces przetwarzania jest realizowany przez zastosowanie operatorów krzyżowania i mutacji, nazywamy algorytmami przeszukiwania genetycznego (GS) [31].

Populacją w algorytmach GS jest zbiór osobników reprezentujących rozwiązania [134]. Każde rozwiązanie jest kodowane przez zbiór jego atrybutów zapisanych w materiale genetycznym (chromosomy, geny). Istnieje wiele sposobów kodowania, specyficznych dla różnych problemów optymalizacji, np. dla problemów przepływowych kolejne rozwiązania koduje się w chromosomie bezpośrednio za pomocą permutacji [31]. Populacja jest przetwarzana za pomocą cyklicznie następujących po sobie procesów: reprodukcji, krzyżowania i mutacji oraz przeżycia lub selekcji. W fazie reprodukcji osobniki są powielane proporcjonalnie do miary przystosowania do środowiska. Funkcją adaptacji, będącą miarą przystosowania, może być np. wartość funkcji celu dla danego rozwiązania. Proces ten powoduje, że osobniki o lepszym przystosowaniu będą miały więcej potomków w następnych pokoleniach. Osobniki wybrane z populacji tworzą tzw. pulę rodzicielską, z której zostają wybrane pary tzw. rodziców dostarczające osobniki nowego pokolenia. Są one tworzone za pomocą operatora krzyżowania genetycznego. Następnie realizowany jest proces mutacji powodujący zmiany w materiale genetycznym, które występują z reguły z małym prawdopodobieństwem pozwalającym na powolne wprowadzanie innowacji w pokoleniu. W fazie przeżycia (selekcji) wybierane są osobniki, które wejdą w skład nowej populacji. Selekcja często jest wykonywana zgodnie z zasadą ruletki, w której większe szanse dawane są osobnikom lepiej przystosowanym. Warunkami zakończenia pracy algorytmu mogą być,

podobnie jak w algorytmach TS: limit czasu, maksymalna liczba iteracji, osiągnięcie zadowalającej wartości funkcji celu lub wartości optymalnej. Metoda GS zawiera wiele elementów, które można swobodnie definiować: sposób kodowania chromosomów, operatory krzyżowania, mutacji, funkcja adaptacji, schemat wyboru puli rodzicielskiej i kojarzenia rodziców, schemat przeżywania. Poniżej przedstawiono ogólny algorytm metody GS [134].

**Krok 0.** Wygeneruj populację początkową  $P(0)$ ,  $t = 0$ , oceń  $P(0)$ .

**Krok 1.**  $t := t + 1$ ; Wybierz osobniki najbardziej dopasowane (selekcja).

**Krok 2.** Zmień wybrane osobniki (transformacja genetyczna: krzyżowanie, mutacja).

**Krok 3.** Oceń  $P(t)$ . Usuń najslabiej dopasowane osobniki (faza przetrwania).

**Krok 4.** Jeżeli spełniony jest warunek zatrzymania to STOP, w przeciwnym przypadku wróć do kroku 1.

Algorytmy GS są obecnie z powodzeniem stosowane do rozwiązywania szeregu problemów optymalizacyjnych, w tym związanych z TSZ. Ich implementacja zawsze wymaga wykonania dużej liczby eksperymentów obliczeniowych w celu właściwego doboru szeregu parametrów sterujących [134]. Metoda GS ma niedoskonałości, które wyrażają się wczesną zbieżnością do lokalnego ekstremum lub słabą zbieżnością do rozwiązań optymalnych lub im bliskich. Zachowuje się dosyć dobrze dla przykładów o małych rozmiarach, lecz dla przykładów o rozmiarach dużych błędy przybliżenia mogą być znaczne. Dla opisywanych wcześniej problemów przepływowych stosuje się najczęściej sposób kodowania bezpośrednio w postaci permutacji. W pracach [31], [142] przedstawiono szereg operatorów genetycznych (krzyżowania i mutacji), które są stosowane na tej formie chromosomu.

#### 4.1.3. Algorytm symulowanego wyżarzania (*simulated annealing*)

W celu wyprowadzenia trajektorii poszukiwań z lokalnego ekstremum algorytm symulowanego wyżarzania (SA) wykorzystuje analogię do procesu termodynamicznego chłodzenia ciała stałego [76], [134]. Stany ciała stałego są postrzegane jako poszczególne rozwiązania problemu, natomiast energia ciała – wartości funkcji celu. W trakcie fizycznego procesu studzenia temperatura redukowana jest powoli w celu utrzymania równowagi energetycznej. Algorytm SA został zaproponowany po raz pierwszy w pracy [76].

Algorytm SA startuje z rozwiązania początkowego, najczęściej wybranego losowo [104], [134]. Następnie, w każdej iteracji z otoczenia rozwiązania bazowego  $\pi$  wybiera się, według ustalonych zasad lub losowo, rozwiązanie  $\pi'$ . Staje się ono rozwiązaniem bazowym w następnej iteracji, jeśli wartość funkcji celu jest lepsza od aktualnego rozwiązania bazowego lub w przypadku przeciwnym może nim stać się z prawdopodobieństwem:

$$p = \exp(-\Delta / T_i),$$

gdzie  $\Delta = c(\pi') - c(\pi)$ ,  $T_i$  – temperatura w bieżącej iteracji  $i$ ,  $c$  – funkcja celu. W każdej iteracji wykonywanych jest  $m$  losowań z otoczenia bieżącego rozwiązania bazowego. Parametr zwany temperaturą zmniejsza się tak, jak w naturalnym procesie wyżarzania. Najczęściej przyjmuje się jeden z dwóch schematów studzenia:

- geometryczny  $T_{i+1} = \lambda_i T_i$ ,
- logarytmiczny  $T_{i+1} = T_i / (1 + \lambda_i T_i)$ ,

gdzie  $i = 0, \dots, N - 1$ ,  $T_0$  – temperatura początkowa,  $T_N$  - temperatura końcowa,  $N$  – liczba iteracji,  $\lambda_i$  – parametr. W algorytmie z reguły przyjmuje się na początku określone wartości parametrów  $T_0$ ,  $T_N$ ,  $N$  oraz wylicza się parametr  $\lambda_i$ . Powinna zachodzić zależność  $T_0 > T_N$ , należy przyjmować  $T_N$  odpowiednio małe, bliskie zeru. Poniżej przedstawiony jest ogólny algorytm metody SA zastosowany do rozwiązywania problemów przepływowych [134].

**Krok 0.** Wyznacz rozwiązanie początkowe  $\pi^0 \in \Pi$ . Podstaw  $\pi^{SA} = \pi^0$ ,  $k = 0$ ,  $T = T_0$ .

**Krok 1.** Wykonaj  $x$ -razy kroki 1.1 – 1.3.

**Krok 1.1.** Podstaw  $k := k + 1$ . Wybierz losowo  $\pi \in N(V, \pi^{k-1})$ .

**Krok 1.2.** Jeżeli  $c(\pi') < c(\pi^{SA})$  to podstaw  $\pi^{SA} = \pi'$ .

**Krok 1.3.** Jeżeli  $c(\pi') < c(\pi^{k-1})$  to podstaw  $\pi^k = \pi'$ . W przeciwnym przypadku zaakceptuj rozwiązanie  $\pi'$  z prawdopodobieństwem  $p = \exp((c(\pi^{k-1}) - c(\pi'))/T)$ , tj.  $\pi^k = \pi'$ , jeżeli zaakceptowano rozwiązanie  $\pi'$  lub  $\pi^k = \pi^{k-1}$ , jeżeli nie zaakceptowano rozwiązania  $\pi'$ .

**Krok 2.** Zmień temperaturę  $T$  według określonego schematu chłodzenia.

**Krok 3.** Jeżeli  $T > T_N$  idź do kroku 1, w przeciwnym wypadku STOP.

Algorytmy SA znajdują zastosowanie w rozwiązywaniu wielu zagadnień optymalizacyjnych, w tym problemów optymalizacji dyskretnej. Dobre wyniki uzyskiwane w aplikacjach pozwalają na traktowanie algorytmów SA jako jednych z najsilniejszych narzędzi.

#### 4.1.4. Formy algorytmów metaheurystycznych zastosowanych w pracy

W nowych modelach planowania przedsięwzięć wieloobektowych podanych w dalszych rozdziałach rozprawy zostaną przyjęte założenia dwóch modeli rozważanych w TSZ: permutacyjnego problemu przepływowego z jedną maszyną w danym stanowisku (przedstawionego w p. 3.2) oraz z maszynami równoległymi w stanowiskach (przedstawionego w p. 3.3). Wybrane przez autora algorytmy do rozwiązywania tych dwóch zagadnień optymalizacyjnych w nowych modelach (poszczególne ich kroki zostały podane w p. 4.1.1 – 4.1.3) mają liczne punkty swobody. Szczegółowa forma tych algorytmów, jest zależna od rodzaju problemów i zostanie podana poniżej.

##### 4.1.4.1. Formy algorytmów metaheurystycznych dla permutacyjnego problemu przepływowego

###### Algorytm przeszukiwania z zabronieniami (*taboo search*)

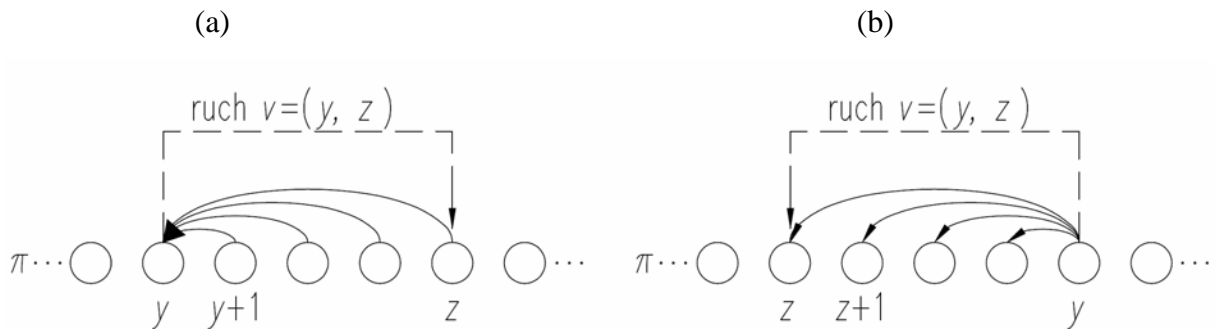
Forma algorytmu TS została przyjęta na podstawie prac [98], [99]. W kroku 1 przyjmuje się rozwiązanie początkowe będące permutacją  $\pi$  utworzoną losowo. Następnie, w tym kroku wyznaczane jest otoczenie permutacji  $\pi$ . Otoczenie  $N_\pi$  zawiera wszystkie możliwe permutacje powstałe po wykonaniu ruchu „włóż” (*insert*). Ruch ten polega na pobraniu operacji (numeru zdania) z pozycji  $y$  i włożeniu jej na pozycję  $z$ ,  $y \neq z$ . Oznaczmy ruch przez  $v = (y, z)$ . Przykład: niech  $\pi = (2, 3, 1, 5, 4)$ , a ruch „włóż”  $v = (2, 4)$ . Permutacja po wykonaniu ruchu  $v$  jest następująca  $\pi^v = (2, 1, 5, 3, 4)$ . Ogólnie można powiedzieć, że liczność zbioru wszystkich możliwych ruchów w permutacji  $\pi$  i liczność zbioru otoczenia  $N_\pi$  wynosi  $(n - 1)^2$ . Z powodu kwadratowo rosnącej liczności zbioru elementów otoczenia  $N_\pi$  wprowadza się możliwe jego ograniczanie przez wykonywanie ruchów w danej permutacji  $\pi$  tylko do pewnej głębokości (określonej pozycji operacji w permutacji  $\pi$ ).

Następnym punktem swobody w algorytmie TS jest postać atrybutów permutacji i ruchu (krok 3 algorytmu). Na podstawie [98] przyjęto, że atrybutem permutacji  $\pi$  i ruchu  $v$  będzie następująca para:

$$AV(\pi, v) = \begin{cases} \{(\pi(y), \pi(y+1))\}, & \text{jeżeli } y < z, \\ \{(\pi(y-1), \pi(y))\}, & \text{jeżeli } y > z. \end{cases}$$

Kolejnym elementem algorytmu TS jest podanie warunków sprawdzających, czy ruch  $v$  dla danej permutacji  $\pi$  jest zabroniony ze względu na zawartość listy tabu  $LT$  (zabronienie

powoduje nieuwzględnianie ruchu  $v$  w kroku 1 algorytmu przy wyznaczaniu otoczenia  $N(\pi)$ . Podobnie, na podstawie [98] przyjęto kolejne, dalej przedstawione warunki. Ruch  $v = (y, z)$  wykonywany w prawo (tj.  $y < z$ ) jest zabroniony dla permutacji  $\pi$  wtedy i tylko wtedy, gdy na liście tabu  $LT$  znajduje się przynajmniej jedna para operacji  $(j_1, j_2)$  taka, że  $j_1 = \pi(i)$  dla pewnego  $y < i \leq z$  oraz  $j_2 = \pi(y)$ . Symetrycznie, ruch  $v = (y, z)$  wykonywany w lewo (tj.  $y > z$ ) jest zabroniony, gdy na liście tabu  $LT$  znajduje się przynajmniej jedna para operacji  $(j_1, j_2)$  taka, że  $j_1 = \pi(y)$  oraz  $j_2 = \pi(i)$  dla pewnego  $z \leq i < y$ . Obrazowo warunki te zostały przedstawione na rys. 4.1.



Rys. 4.1. Warunki określające zabronienie wykonywania ruchu  $v = (y, z)$  - co najmniej jedna para operacji połączonych łukiem musi znajdować się na liście  $LT$ : (a) ruch w prawo, (b) ruch w lewo [98]

Praktycznie warunki najłatwiej jest sprawdzić dysponując permutacją odwrotną  $\pi^{-1}$  do permutacji  $\pi$ . Utworzenie permutacji odwrotnej  $\pi^{-1}$  do permutacji  $\pi$  polega na dokonaniu transformacji: dla permutacji  $\pi$  mamy  $\pi(i) = j$ , to dla permutacji odwrotnej  $\pi^{-1}$  będziemy przyjmować  $\pi^{-1}(j) = i$ . Stwierdzenie, czy para operacji  $(j_1, j_2)$  z listy tabu  $LT$  zabrania wykonywania ruchu  $v = (y, z)$  w prawo, ogranicza się wtedy do zweryfikowania prawdziwości zależności  $y < \pi^{-1}(j_1) \leq z, \pi^{-1}(j_2) = y$ . Analogicznie wygląda sprawdzenie dla ruchu  $v = (y, z)$  wykonywanego w lewo.

Ponadto wprowadza się dodatkowy warunek, który umożliwi zaliczenie danego ruchu do zbioru ruchów niezabronionych, mimo posiadania statusu ruchu zabronionego ze względu na listę tabu  $LT$ . Warunek ten polega na przypisaniu ruchowi zabronionemu  $v = (y, z)$  wykonywanemu w permutacji  $\pi$  funkcji aspiracji [98]. Jeśli spełnione jest kryterium odnośnie do wartości tej funkcji, to dany ruch  $v$  będzie zaliczony do ruchów niezabronionych, a dokładniej będzie spełniony warunek odnośnie do poziomu aspiracji – wartości funkcji aspiracji permutacji  $\pi$  po wykonaniu ruchu  $v$ . Na podstawie pracy [98] w algorytmie TS przyjęto najprostszą formę funkcji aspiracji tj. rozpatrywaną w danym problemie funkcję celu.

Poziomem aspiracji, który pozwala na zaliczenie danego ruchu  $v$  w permutacji  $\pi$  do ruchów niezabronionych będzie poprawienie lub wyrównanie najlepszej wartości funkcji celu znalezionej do tej pory (iteracji algorytmu), obliczonej dla przekształconej permutacji  $\pi$  przez ruch  $v$ .

W przyjętej postaci algorytmu TS nie wykorzystywano elementów zaawansowanej metody przeszukiwania z zakazami (pamięć długoterminowa - metoda skoku powrotnego, detektor cyklu) [98].

Wartości parametrów sterujących algorytmu TS (maksymalna liczba iteracji, długość listy tabu  $LT$ , parametr ograniczający licznosc zbioru otoczenia  $N_{\pi}$ ) będą ustalane każdorazowo w danej aplikacji.

### **Algorytm przeszukiwania genetycznego (*genetic search*)**

Wyjściową formę algorytmu GS przyjęto na podstawie pracy [118], a następnie algorytm ten zmodyfikowano. W kroku 0 jest tworzona populacja, która składa się z osobników – permutacji utworzonych losowo. Następnie dla osobników utworzonej populacji obliczane są wartości funkcji przystosowania (ocena populacji). Jako funkcję tę przyjęto wartość funkcji celu. W kroku 1 dana populacja jest sortowana zgodnie z rosnącymi wartościami funkcji celu. W etapie selekcji wybierane są osobniki (permutacje) do krzyżowania, a następnie do mutacji. W etapie krzyżowania rodzic pierwszy jest wybierany na zasadzie koła ruletki. W tym kole większe szanse mają osobniki o lepszej wartości funkcji celu. Ścisłej, wybór osobnika jako rodzica pierwszego do krzyżowania będzie następował zgodnie z prawdopodobieństwem [118]:

$$p([k]) = \frac{2k}{M(M+1)},$$

gdzie  $[k]$  oznacza numer osobnika (permutacji) w populacji posortowanej w kolejności zmniejszania się wartości funkcji celu,  $M$  – licznosc populacji. Drugi z rodziców do krzyżowania wybierany jest losowo z populacji powstałej po usunięciu osobnika wybranego jako rodzica pierwszego.

W pracy [118] zastosowano tylko operator krzyżowania porządkowego OX [106], [134]. W formie algorytmu GS użytej w rozprawie zastosowano trzy różne operatory krzyżowania: wyżej wymieniony operator krzyżowania porządkowego OX, operator krzyżowania z częściowym odwzorowaniem PMX oraz operator krzyżowania cyklicznego CX [106], [134]. Wybór operatora krzyżowania w kroku 2 następuje w sposób losowy. Jest on zmieniany na inny również drogą losową po wykonaniu określonej liczby iteracji. Po



dokonaniu wyboru operatora następuje z określonym prawdopodobieństwem krzyżowanie osobników i utworzenie nowego pokolenia. Następnie, w kroku 2, z tego pokolenia losowo wybierana jest część osobników do przeprowadzenia mutacji. Mutacja jest realizowana z określonym prawdopodobieństwem za pomocą wykonania ruchu „włóż” w osobnikach (permutacjach) wcześniej wybranych do mutacji. Ruch „włóż” wykonywany jest według zasad przedstawionych dla algorytmu przeszukiwania z zakazami.

W kroku 3 następuje ocena przystosowania nowego pokolenia (obliczenie wartości funkcji celu). Faza przetrwania jest realizowana przez wymianę najgorszych osobników populacji (o największej wartości funkcji celu) na osobniki pokolenia powstałe przez transformację genetyczną (krzyżowanie i mutację). W każdym z etapów następuje zapamiętanie najlepszego dotychczas znalezionej rozwiązania i wartości funkcji celu.

W przedstawionej formie algorytmu może wystąpić sytuacja, w której po wykonaniu części iteracji, w populacji będą znajdowały się tylko osobniki reprezentujące lokalne ekstremum (z taką samą wartością funkcji celu). Aby ponownie zróżnicować osobniki stosuje się wymianę części osobników najgorzej przystosowanych na nowe utworzone w sposób losowy. Operację tę wykonuje się cyklicznie, zawsze po zrealizowaniu określonej liczby iteracji. Jednocześnie następuje zmiana operatora krzyżowania na losowo wybrany z grupy trzech wyżej wymienionych. Operacja ta nie była stosowana w pracy [118].

W tej formie algorytmu GS występuje znaczna ilość parametrów sterujących, które w danej aplikacji przyjmują określone wartości. Tymi parametrami są: licznosc populacji, licznosc pokolenia, prawdopodobienstwo krzyżowania, licznosc zbioru osobników mutowanych, prawdopodobienstwo mutacji, licznosc zbioru osobników najgorzej przystosowanych z populacji, które są wymieniane na nowe, liczba iteracji, po której następuje wymiana, całkowita liczba iteracji. Ich wartości będą ustalone dla konkretnej aplikacji algorytmu.

### **Algorytm symulowanego wyżarzania (*simulated annealing*)**

Forma prezentowanego algorytmu SA nie odbiega znacznie od algorytmu przedstawionego w p. 4.1.3. Algorytm startuje z rozwiązania początkowego (permutacji) wybranego losowo [134]. Jako otoczenie przyjęto wykonywanie ruchu „włóż” w rozwiązaniu bazowym (permutacji), podobnie jak w algorytmie TS. Przyjęto geometryczny schemat chłodzenia. Badania w pracach [108], [134] wykazują, że oba wymienione w p. 4.1.3 schematy chłodzenia zapewniają podobne wyniki obliczeniowe. W celu ograniczenia wahań trajektorii poszukiwań (szczególnie w trakcie początkowych iteracji, gdy wartość temperatury

jest dość duża) zastosowano dodatkowe ograniczenie za pomocą parametru  $\varepsilon$  przyjmowania nowych rozwiązań bazowych [108]. Prawdopodobieństwo  $p$  akceptacji nowego rozwiązania będzie miało następującą postać (krok 1.3 algorytmu):

$$p = \begin{cases} 1 & c(\pi') \leq c(\pi^{k-1}) \\ 0 & c(\pi') > (1 + \varepsilon)c(\pi^{k-1}) \\ \exp\left(\frac{(c(\pi^{k-1}) - c(\pi'))}{T}\right) & \text{w przeciwnym przypadku} \end{cases}$$

W tej formie algorytmu SA, podobnie jak we wcześniejszych, występują liczne parametry sterujące dotyczące wartości temperatur początkowej  $T_0$  i końcowej  $T_N$ , liczby losowań występującej w kroku 1, wartości parametru  $\varepsilon$ , całkowitej liczby iteracji. Wartości tych parametrów będą ustalone w danej aplikacji algorytmu.

#### 4.1.4.2. Formy algorytmów metaheurystycznych dla problemu przepływowego z maszynami równoległymi

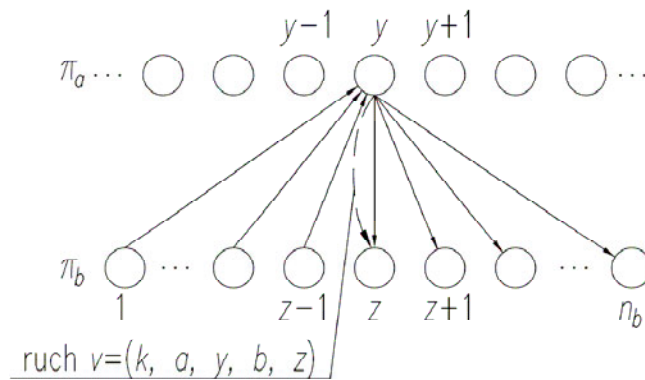
##### Algorytm przeszukiwania z zabronieniami (*taboo search*)

Forma algorytmu TS została przyjęta na podstawie prac [98], [100]. W kroku 1 przyjmuje się rozwiązanie początkowe będące zestawem permutacji podziału rozłącznego (dalej zwane dla uproszczenia permutacją)  $\pi$  utworzonym losowo [98]. Następnie, w tym kroku wyznaczane jest otoczenie permutacji  $\pi$ . Otoczenie  $N_\pi$  zawiera wszystkie możliwe permutacje powstałe po wykonaniu ruchu „włóż” (*insert*) w ramach jednej, losowo wybranej permutacji  $\pi_k$ . W związku z tym, że permutacja  $\pi_k$  jest podzielona rozłącznie, ruch „włóż” będzie oznaczony przez  $v = (k, a, y, b, z)$ . Polega on na pobraniu operacji (numeru zadania) z pozycji  $y$  na maszynie  $a$  znajdującej się w stanowisku  $k$  i włożeniu jej na pozycję  $z$  na maszynie  $b$  znajdującej się w stanowisku  $k$ . Przykład: niech  $\pi_k = ((2, 3, 1), (5, 4))$ , a ruch „włóż”  $v = (k, 1, 2, 2, 1)$ . Permutacja po wykonaniu ruchu  $v$  jest następująca  $\pi_k^v = ((2, 1), (3, 5, 4))$ . Podobnie jak wcześniej z powodu przynajmniej kwadratowo rosnącej liczności zbioru elementów otoczenia  $N_\pi$ , wprowadza się możliwe jego ograniczanie przez wykonywanie ruchów w danej permutacji  $\pi$  tylko do pewnej głębokości (określonej pozycji operacji w permutacji  $\pi$ ).

Postać atrybutów permutacji i ruchu (krok 3 algorytmu) przyjęto na podstawie [98], [100]. Dla ruchu  $v = (k, a, y, b, z)$  atrybuty są następujące:

$$AV(\pi, v) = \begin{cases} \{(\pi_{ka}(y), \pi_{ka}(y+1))\}, & b = a, y < z \\ \{(\pi_{ka}(y-1), \pi_{ka}(y))\}, & b = a, y > z \\ \{(\pi_{ka}(y-1), \pi_{ka}(y))\}, \{(\pi_{ka}(y), \pi_{ka}(y+1))\} & b \neq a. \end{cases}$$

Kolejnym elementem algorytmu TS jest podanie warunków sprawdzających, czy ruch  $v$  dla danej permutacji  $\pi$  jest zabroniony ze względu na zawartość listy tabu  $LT$  (zabronienie powoduje nieuwzględnianie ruchu  $v$  w kroku 1 algorytmu przy wyznaczaniu otoczenia  $N_{\mathcal{P}}$ ). Podobnie na podstawie [98] przyjęto następujące warunki. Dla ruchów wykonywanych w obrębie jednej maszyny ( $b = a$ ) warunki te są takie same jak przedstawione wcześniej w p. 4.1.4.1 dla algorytmu TS. Dla  $b \neq a$  ruch  $v = (k, a, y, b, z)$  jest zabroniony dla permutacji  $\pi$  wtedy i tylko wtedy, gdy na liście tabu  $LT$  znajduje się przynajmniej jedna para operacji  $(j_1, j_2)$  taka, że  $j_1 = \pi_b(i)$  dla pewnego  $1 \leq i < z$ ,  $j_2 = \pi_a(y)$  lub  $j_1 = \pi_a(y)$ ,  $j_2 = \pi_b(i)$  dla pewnego  $z \leq i \leq n_b$  ( $n_b$  – długość permutacji  $\pi_b$ ). Obrazowo warunki te zostały przedstawione na rys. 4.2.



Rys. 4.2. Warunki określające zabronienie wykonywania ruchu  $v = (k, a, y, b, z)$ ,  $b \neq a$  - co najmniej jedna para operacji połączonych łukiem musi znajdować się na liście  $LT$  [98]

Praktycznie te dodatkowe warunki dla  $b \neq a$  najłatwiej sprawdzić dysponując tzw. uogólnioną permutacją odwrotną  $\pi^{-1} = (\pi^{-1}(1), \pi^{-1}(2), \dots, \pi^{-1}(n))$  do permutacji  $\pi$  (opisaną w pracy [98]) oraz dodatkową zmienną  $m(i)$  określającą maszynę, na której wykonywana jest operacja  $i$  w permutacji  $\pi$ , gdzie  $i$  – numer zadania. Stwierdzenie, czy para operacji  $(j_1, j_2)$  z listy tabu  $LT$  zabrania wykonywania ruchu  $v = (k, a, y, b, z)$ ,  $b \neq a$  ogranicza się wtedy do zweryfikowania prawdziwości zależności  $m(j_1) = b$ ,  $\pi_b^{-1}(j_1) < z$ ,  $j_2 = \pi_a(y)$  lub  $j_1 = \pi_a(y)$ ,  $m(j_2) = b$ ,  $z \leq \pi_b^{-1}(j_2)$ . Ponadto zastosowano postać funkcji poziomu aspiracji jak w formie algorytmu TS przedstawionej w p. 4.1.4.1.

Podobnie jak w p. 4.1.4.1, w przyjętej postaci algorytmu TS nie wykorzystano elementów zaawansowanej metody tabu. Wartości parametrów sterujących algorytmu TS

(maksymalna liczba iteracji, długość listy tabu  $LT$ , parametr ograniczający licznosc otoczenia  $N_{\pi}$ ) będą ustalane każdorazowo w danej aplikacji.

### **Algorytm przeszukiwania genetycznego** (*genetic search*)

Forma algorytmu GS została przyjęta na podstawie pracy [118] w formie podanej w p. 4.1.4.1. W algorytmie tym dokonano modyfikacji w związku ze zmianą rodzaju problemu, którego aktualnie dotyczy, tj. problemu przepływowego z maszynami równoległymi. W kroku 0 jest tworzona populacja, która składa się z osobników – permutacji podziału rozłącznego utworzonych losowo. Następnie dla osobników utworzonej populacji obliczane są wartości funkcji przystosowania (ocena populacji). Jako funkcję tę przyjęto wartość funkcji celu. W kroku 1 dana populacja jest sortowana zgodnie z rosnącymi wartościami funkcji celu. W etapie selekcji wybierane są osobniki (permutacje podziału rozłącznego) do krzyżowania, a następnie do mutacji. W etapie krzyżowania rodzic pierwszy jest wybierany za zasadzie koła ruletki według zasad i funkcji prawdopodobieństwa podanych w p. 4.1.4.1, a drugi w sposób losowy.

Zastosowano te same operatory genetyczne krzyżowania (OX, CX, PMX) z tym, że rodzicami dla wykonania operacji krzyżowania są wybrane w sposób losowy permutacje  $\pi_k$  z permutacji podziału rozłącznego  $\pi$ . Mutacja w tej formie algorytmu GS polega na wykonaniu losowego ruchu „włóż” w wybranej w sposób losowy permutacji  $\pi_k$  z permutacji podziału rozłącznego  $\pi$ .

Pozostałe elementy algorytmu GS przyjęto takie same jak w p. 4.1.4.1. Podobnie jak wcześniej parametry sterujące to: licznosc populacji, licznosc pokolenia, prawdopodobieństwo krzyżowania, licznosc zbioru osobników mutowanych, prawdopodobieństwo mutacji, licznosc zbioru osobników najgorzej przystosowanych z populacji, które są wymieniane na nowe, liczba iteracji, po której następuje wymiana, całkowita liczba iteracji.

### **Algorytm symulowanego wyżarzania** (*simulated annealing*)

Forma algorytmu SA w problemie z maszynami równoległymi jest podobna jak w p. 4.1.4.1. Różnice polegają na przyjęciu, jako reprezentacji rozwiązania, zestawu permutacji podziału rozłącznego zamiast permutacji, która reprezentuje rozwiązanie w permutacyjnym problemie przepływowym z jedną maszyną w stanowisku. W związku z tym, algorytm startuje z rozwiązania początkowego (w tym przypadku z zestawu permutacji

podziału rozłącznego) wybranego losowo. Jako otoczenie przyjęto wykonywanie ruchu „włóż” w losowo wybranej permutacji  $\pi_k$  z rozwiązania bazowego tj. permutacji podziału rozłącznego  $\pi$ . Pozostałe elementy tego algorytmu są takie same jak podane w p. 4.1.4.1.

Podobnie jak wcześniej, parametrami sterującymi są: wartości temperatur początkowa  $T_0$  i końcowa  $T_N$ , liczba losowań występująca w kroku 1, wartość parametru  $\varepsilon$ , całkowita liczba iteracji.

#### **4.2. Weryfikacja wyników obliczeń uzyskiwanych za pomocą wybranych trzech algorytmów metaheurystycznych**

Wybrane trzy algorytmy metaheurystyczne będą narzędziami służącymi do rozwiązywania problemów optymalizacji dyskretnej stosowanymi w nowych modelach planowania przedsięwzięć wieloobektowych przedstawionych w rozdziale 5 pracy. Jak podano w p. 3.4 należą one do grupy algorytmów poprawiających, za pomocą których można uzyskać wynik przybliżony. W związku z tym, ważnym zagadnieniem jest ocena uzyskiwanych przez nie wyników. W teorii szeregowania zadań wprowadzono szereg możliwości oceny „jakości” wybranych algorytmów. Jedną z najważniejszych cech podlegających tej ocenie jest dokładność przybliżenia uzyskiwanych wyników (odległość wartości funkcji celu uzyskiwanego rozwiązania suboptymalnego od wartości optymalnej, wartości dolnego ograniczenia lub wartości funkcji celu uzyskiwanej za pomocą innego algorytmu).

W pracy J. Błażewicza [7] zostało przedstawionych szereg metod oceny dokładności przybliżenia. Do nich należą metody analityczne i eksperymentalne. W grupie metod analitycznych wymienia się metody analizy najgorszego przypadku i analizy probabilistycznej. Te dwie metody są z reguły kłopotliwe w zastosowaniach ze względu na potrzebę użycia złożonego aparatu matematycznego. Mają one jednak tę podstawową zaletę, że dostarczają ocen niezależnych od rozpatrywanych przykładów badanego problemu. Analiza eksperymentalna jest obecnie najbardziej popularną i łatwą do zastosowania metodą oceny. Jest jednak metodą subiektywną, zależną od przyjętej próbki przykładów. Ze względu na popularność i łatwość stosowania, do weryfikacji wybranych w rozprawie algorytmów przyjęto metodę analizy eksperymentalnej (przyjętą również np. w pracy [108]).

Metoda analizy eksperymentalnej polega na ocenie a posteriori zachowania się badanych algorytmów (w tym przypadku błędu przybliżenia) na podstawie wyników obliczeń na ograniczonej, lecz reprezentatywnej próbce konkretnych przykładów. Próbkę ta najczęściej tworzona jest przez wygenerowanie w sposób losowy podzbiorów przykładów danego

problemu lub wykorzystanie utworzonych wcześniej „trudnych” przykładów testowych wykorzystywanych przez innych badaczy. Jakość uzyskiwanych rezultatów porównuje się z jedną z wartości: wartością optymalną, dolnym ograniczeniem, wartością suboptymalną otrzymaną innym algorytmem przybliżonym lub wartością uzyskaną w sposób losowy. W rozprawie wybrane algorytmy będą weryfikowane za pomocą dwóch poniżej przedstawionych metod.

Metoda pierwsza polegać będzie na porównaniu wyników otrzymanych za pomocą każdego z analizowanych algorytmów z wynikami jednego, konkretnego algorytmu  $B$  lub z wartością będącą rozwiązaniem optymalnym danego przykładu. Sposób ten będzie wykorzystywany w niniejszej pracy podczas rozpatrywania przykładów, które są podane w literaturze i były analizowane już przez innych badaczy. Na podstawie wartości  $CA$ ,  $A \in \mathcal{A}$  (wartość funkcji celu uzyskana za pomocą algorytmu  $A$ ) oraz  $CB$  (wartość funkcji celu uzyskana za pomocą algorytmu  $B$  lub wartość optymalna funkcji celu dla danego przykładu) dla każdego przykładu danego problemu definiuje się dodatkową wartość:

$PRD(A) = 100\%(CA - CB) / CB$  - średni błąd względny (średnia procentowa różnica między wartością funkcji celu uzyskaną algorytmem  $A$  i wartością  $CB$  odniesiona do wartości  $CB$ ).

Metoda druga polegać będzie na porównaniu wyników otrzymanych ocenianym algorytmem  $A$ , dla badanej próbki przykładów, z najlepszymi wynikami osiągniętymi przez zbiór testowanych algorytmów  $\mathcal{A}$ ,  $A \in \mathcal{A}$ . W związku z tym, definiuje się następujące wartości:

$CA$  – wartość funkcji celu otrzymana algorytmem  $A$ ,

$C^*$  - najlepsza wartość funkcji celu otrzymana jednym z badanych algorytmów, tj.

$$C^* = \min_{A \in \mathcal{A}} CA,$$

$PRD(A) = 100\%(CA - C^*) / C^*$  - średni błąd względny (średnia procentowa różnica między wartością funkcji celu uzyskaną algorytmem  $A$  i wartością  $C^*$ , odniesiona do wartości  $C^*$ ).

W przedstawionych w dalszej części rozprawy, nowych modelach planowania przedsięwzięć wieloobektowych zastosowano permutacyjny problem przepływowy (opisany w p. 3.2) i problem przepływowy z maszynami równoległymi (opisany w p. 3.3). Te dwa problemy są szczególnymi przypadkami formułowanych modeli planowania przedsięwzięć wieloobektowych. W związku z tym, w trakcie weryfikacji algorytmów, rozważano permutacyjny problem przepływowy i problem przepływowy z maszynami równoległymi bez uwzględnienia dodatkowych ograniczeń i z funkcją celu  $C_{max}$  (te dwa problemy z przyjętą funkcją celu  $C_{max}$  są szczególnymi przypadkami modeli sformułowanych w rozdziale 5).

Implementację programową wybranych algorytmów dokonano w środowisku Mathematica. Środowisko to zostało wybrane ze względu na łatwość jej przeprowadzenia dla wybranych algorytmów. Autor niniejszej rozprawy opracował wszystkie programy stosowane w pracy. Obliczenia wykonano korzystając z komputera PC z procesorem AMD Athlon XP 2500+.

#### 4.2.1. Weryfikacja wyników uzyskiwanych z zastosowaniem algorytmów metaheurystycznych dla permutacyjnego problemu przepływowego

Permutacyjny problem przepływowy bez dodatkowych ograniczeń i z funkcją celu  $C_{max}$  (problem FP | |  $C_{max}$ ) jest zagadnieniem, na bazie którego wielu badaczy zajmujących się algorytmami szeregowania zadań przeprowadza weryfikację nowych algorytmów, np. [99], [118]. Model matematyczny tego problemu został przedstawiony w p. 3.2. Znalezienie harmonogramu dla przyjętej zmiennej decyzyjnej – permutacji  $\pi$  wiąże się z podaniem terminów zakończenia wykonywania zadań ze zbioru  $J$ . Terminy te można określić ze wzoru o postaci rekurencyjnej:

$$C_{i, \pi(j)} = \max\{C_{i, \pi(j-1)}, C_{i-1, \pi(j)}\} + p_{i, \pi(j)},$$

gdzie:  $j = 1, \dots, n, i = 1, \dots, m, \pi(0) = 0, C_{i,0} = 0, C_{0,j} = 0$ .

Terminy zakończenia wykonywania poszczególnych operacji mogą być znalezione w czasie rzędu  $O(nm)$ . Dla wybranego kryterium  $C_{max}$  (minimum czasu realizacji wszystkich zadań) poszukiwana jest permutacja  $\pi^* \in \Pi$ , dla której:

$$C_{max}(\pi^*) = \min_{\pi \in \Pi} C_{max}(\pi),$$

$$\text{gdzie } C_{max}(\pi) = C_{m, \pi(n)}.$$

W niniejszej pracy ocena wyników, otrzymywanych za pomocą trzech wybranych algorytmów metaheurystycznych dla permutacyjnego problemu przepływowego, będzie miała dwa etapy. W pierwszym zostaną zastosowane przykłady testowe podane w literaturze związanej z teorią szeregowania zadań [138], [139]. Ze względu na zbyt małą reprezentatywność tych przykładów dla problemów planowania przedsięwzięć budowlanych (liczba rozpatrywanych zadań w [138], [139] wynosi  $n = 20, 50, 100, 200, 500$ , co w planowaniu przedsięwzięć budowlanych jest zbyt szerokim spektrum; w tego typu zagadnieniach można spotkać się w praktyce z około  $n \leq 30$  zadaniami) zdecydowano się na przeprowadzenie drugiego etapu weryfikacji polegającej na dokładniejszej ocenie uzyskiwanych rezultatów dla  $n \leq 30$  zadań (działek roboczych, obiektów). Etap ten będzie polegał na poddaniu ocenie rezultatów wybranych algorytmów uzyskanych przy

zastosowaniu przykładów wygenerowanych losowo o rozmiarach  $n = 10, 15, 20, 25, 30$  zadań.

Jak już wspomniano, w etapie pierwszym zastosowane zostaną przykłady z pracy [138], [139]. Dla oceny rezultatów obliczeń uzyskiwanych za pomocą trzech wybranych algorytmów TS, GS, SA wybrano trzy rozmiary przykładów o wielkości  $n \times m = 20 \times 5, 20 \times 10, 20 \times 20$  ( $n$  – liczba zadań,  $m$  – liczba maszyn), które najbardziej odpowiadają praktycznym problemom planowania przedsięwzięć budowlanych. W każdym rozmiarze wybrano po pięć przykładów z dziesięciu dostępnych. Dane odnośnie czasów trwania poszczególnych operacji są dostępne na stronie internetowej [139] i w pracy [138]. Wartości parametrów sterujących w trzech testowanych algorytmach przyjęto następująco:

- algorytm TS: długość listy tabu  $LT = 0.5n = 10$ , nie wprowadzono dodatkowych ograniczeń liczności zbioru otoczenia  $N_{\pi}$ , maksymalna liczba iteracji – 20000,
- algorytm GS: licznosc populacji –  $10n = 200$ , licznosc pokolenia jako odsetek licznosci populacji –  $0,2 \times 10n = 2n = 40$ , prawdopodobienstwo krzyzowania –  $1,0$ , licznosc zbioru osobnikow mutowanych w pokoleniu  $0,02 \times 10n = 4$ , prawdopodobienstwo mutacji –  $0,1$ , licznosc zbioru osobnikow najgorzej przystosowanych z populacji, ktore sa wymieniane na nowe –  $0,9 \times 10n = 180$ , liczba iteracji, po ktorej nastepuje wymiana i zmiana operatora krzyzowania - 100, calkowita liczba iteracji – 20000,
- algorytm SA: wartosci temperatur poczatkowa  $T_0 = 60$ , koncowa  $T_N = 0,5$ , liczba losowan wystepujaca w kroku 1 -  $0,5n = 10$ , wartosc parametru  $\varepsilon = 0,3$ , calkowita liczba iteracji – 20000.

Podane powyzej parametry zostaly ustalone na podstawie prac [99], [106], [108], [118], [134]. Kazdy z rozpatrywanych przykladow testowych byl rozwiazywany siedmiokrotnie. Nastepnie srednia arytmetyczna z tych obliczen byla odniesiona do wartosci optymalnej kryterium dla danego przykladu (wartosci te sa podane w [138], [139]) poprzez podanie sredniego bledu wzglednego  $PRD$  dla danego algorytmu:

$$PRD(A) = 100\%(C^A - C^{opt}) / C^{opt},$$

gdzie:  $C^A$  – wartosc funkcji celu uzyskana danym algorytmem,  $C^{opt}$  – wartosc optymalna funkcji celu.

Szczegolowe rezultaty obliczeniowe uzyskane za pomoca algorytmow TS, GS, SA znajduja sie w tabelach Z.1 – Z.9 zawartych w zalaczniku do rozprawy. Jako informacje dodatkowe, obrazujace przebieg obliczen dla poszczegolnych prob, podano w nich wartosci funkcji celu uzyskane w iteracjach algorytmow, wartosci funkcji celu uzyskane na koncu obliczen ( $C_{max}$ ) oraz numer iteracji, w ktorej uzyskano wartosc optymalna funkcji celu



( $iter_{opt}$ ). Podsumowanie wyników obliczeń dla przykładów testowych w pierwszym etapie weryfikacji wybranych algorytmów przedstawiono w tabeli 4.1.

Tabela 4.1. Podsumowanie wyników obliczeń dla przykładów testowych wg [138], [139]

Rozmiar przykładowu [ $n \times m$ ]	Oznaczenie przykładowu wg [139]	$PRD(TS)$ [%]	$PRD(GS)$ [%]	$PRD(SA)$ [%]
1	2	3	4	5
20x5	tai001	0,00	1,49	0,00
	tai002	0,06	0,36	0,01
	tai003	0,00	1,03	0,00
	tai004	0,00	1,30	0,17
	tai006	0,00	1,26	0,06
	<b>PRD średnio</b>	<b>0,01</b>	<b>1,09</b>	<b>0,05</b>
20x10	tai011	0,00	1,44	0,23
	tai012	0,00	1,79	0,69
	tai015	0,00	1,58	0,16
	tai017	0,00	1,59	0,04
	tai018	0,07	1,80	0,65
	<b>PRD średnio</b>	<b>0,01</b>	<b>1,64</b>	<b>0,35</b>
20x20	tai022	0,03	1,50	0,33
	tai024	0,01	1,56	0,25
	tai025	0,13	1,50	0,39
	tai027	0,41	1,58	0,31
	tai029	0,03	1,76	0,26
	<b>PRD średnio</b>	<b>0,12</b>	<b>1,58</b>	<b>0,31</b>
<b>PRD średnio dla <math>n=20</math></b>		<b><u>0,05</u></b>	<b><u>1,44</u></b>	<b><u>0,24</u></b>

W etapie drugim przeprowadzono weryfikację polegającą na dokładniejszej ocenie uzyskiwanych rezultatów dla  $n \leq 30$  zadań. Wybrano następujące rozmiary przykładów o wielkości  $n \times m = 10 \times 5, 10 \times 10, 10 \times 15, 15 \times 5, 15 \times 10, 15 \times 15, 20 \times 5, 20 \times 10, 20 \times 15, 25 \times 5, 25 \times 10, 25 \times 15, 30 \times 5, 30 \times 10, 30 \times 15$  ( $n$  – liczba zadań,  $m$  – liczba maszyn). W każdym rozmiarze utworzono po siedem przykładów. Czasy trwania poszczególnych operacji zostały wygenerowane losowo z zastosowaniem rozkładu jednostajnego [1, 99].

Wartości parametrów sterujących zostały ustalone następująco:

- algorytm TS: długość listy tabu  $LT = 0.5n$ , wprowadzono dodatkowe ograniczenia liczności zbioru otoczenia  $N_\pi$  przez ograniczenie liczby ruchów „włóż” wykonywanych

na operacji z danej pozycji do  $0,5(n-1)$  pozycji (maksymalnie możliwych jest  $n-1$  wstawień), maksymalna liczba iteracji – 5000,

- algorytm GS: licznosc populacji –  $5n$ , licznosc pokolenia jako odsetek licznosci populacji –  $0,2 \times 5n = n$ , prawdopodobienstwo krzyzowania –  $1,0$ , licznosc zbioru osobnikow mutowanych w pokoleniu  $0,02 \times 5n = 0,1n$ , prawdopodobienstwo mutacji –  $0,1$ , licznosc zbioru osobnikow najgorzej przystosowanych z populacji, ktore sa wymieniane na nowe –  $0,9 \times 5n = 4,5n$ , liczba iteracji, po ktorej nastepuje wymiana i zmiana operatora krzyzowania - 100, calkowita liczba iteracji – 10000,
- algorytm SA: wartosci temperatur poczatkowa  $T_0 = 60$ , koncowa  $T_N = 0,5$ , liczba losowan wystepujaca w kroku 1 -  $0,5n$ , wartosc parametru  $\varepsilon = 0,3$ , calkowita liczba iteracji – 10000.

Wartosci parametrów zostały ustalone również na podstawie prac [99], [106], [108], [118], [134]. Zmniejszenie liczby iteracji i wartosci niektórych parametrów w stosunku do wartosci z etapu pierwszego zostało podyktowane zbyt dlugim czasem trwania obliczen. Przyjeto dodatkowy warunek, ktory wynikał z mozliwosci praktycznego przeprowadzenia badan, ze czas obliczen jednej próby dla danego przykladu wybranym algorytmem nie powinien przekraczac trzech godzin. Kazdy z rozpatrywanych przykladów testowych był rozwiazywany siedmiokrotnie. Nastepnie srednia arytmetyczna z siedmiu prób była odniesiona do minimalnej sredniej arytmetycznej z siedmiu prób, dla wartosci kryterium wyznaczonego trzema algorytmami TS, GS, SA, przez obliczenie sredniego błedu względnego według metody drugiej podanej we wstepie do niniejszego rozdzialu:

$PRD(A) = 100\%(C^A - C^*) / C^*$  - sredni bład względný (srednia procentowa różnica między wartoscią funkcji celu uzyskaną algorytmem  $A$  i wartoscią  $C^*$  odniesiona do wartosci  $C^*$ ), gdzie:

$C^A$  – wartosc funkcji celu otrzymana wybranym algorytmem  $A$ ,  $A \in \mathcal{A}$ ,

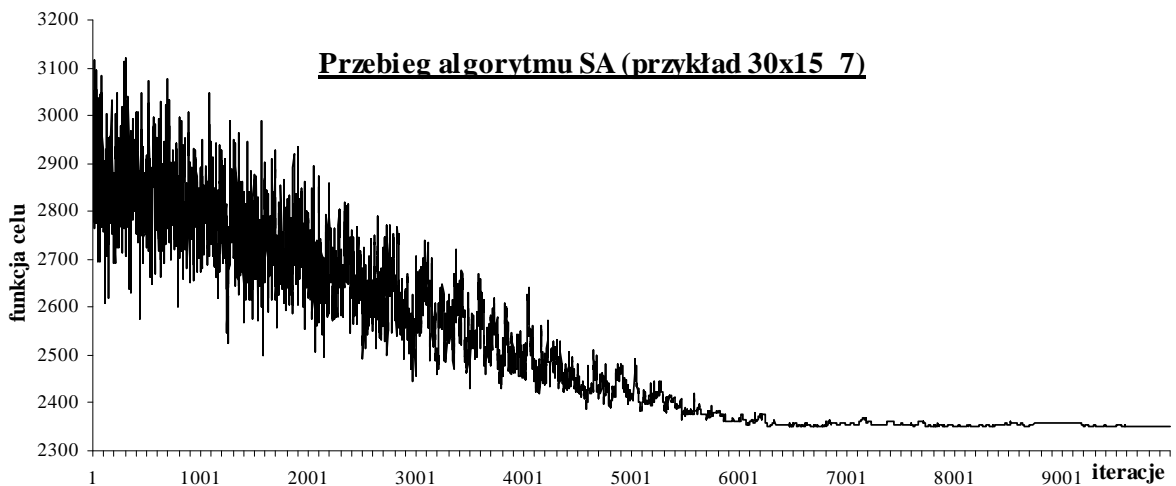
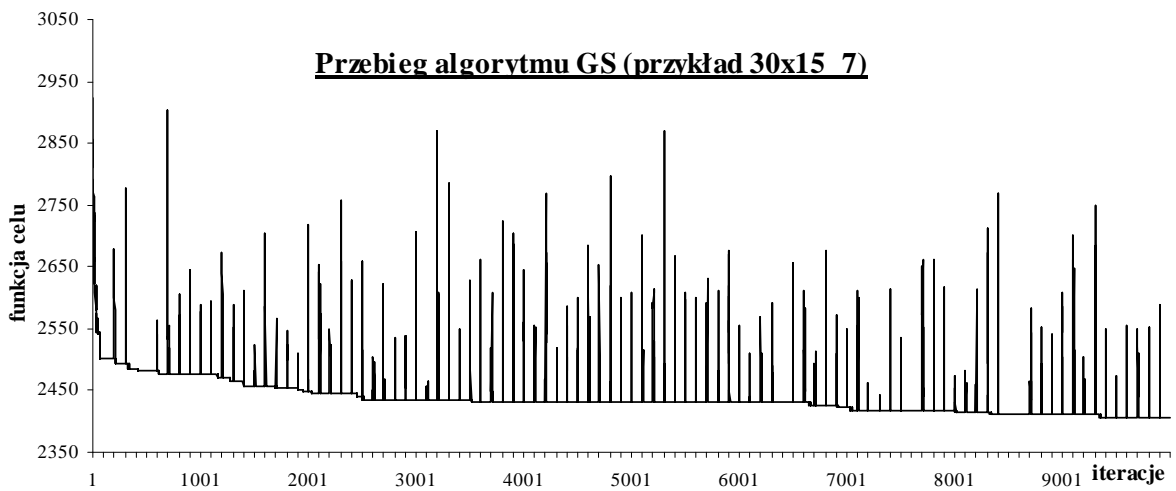
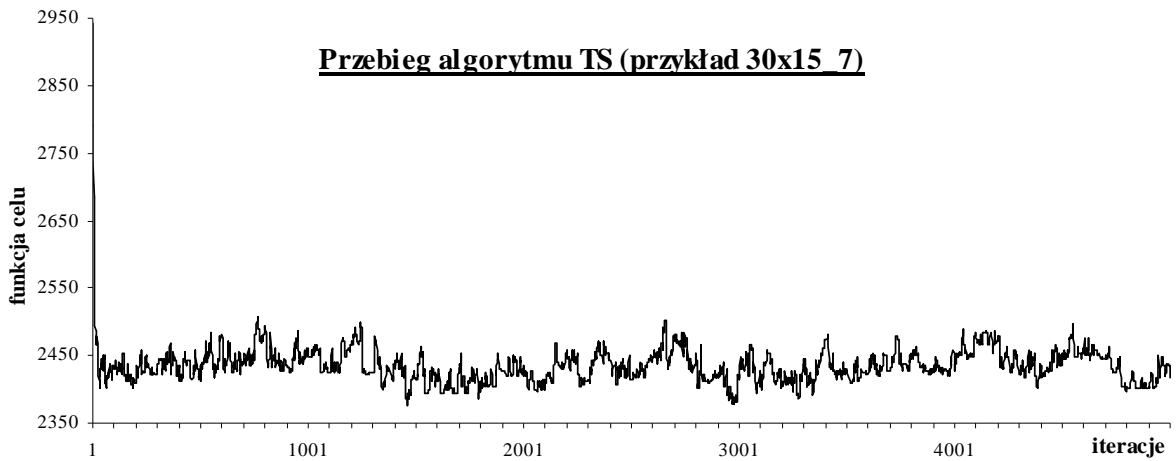
$C^*$  - najlepsza wartosc funkcji celu otrzymana jednym z badanych algorytmów, tj.

$$C^* = \min_{A \in \mathcal{A}} C^A.$$

Do zbioru  $\mathcal{A}$  badanych algorytmów należą algorytmy TS, GS, SA. Podsumowanie wynikow obliczen dla przykladów testowych w drugim etapie weryfikacji przedstawiono w tabeli 4.2. Przykladowe przebiegi poszukiwan lokalnych przedstawiono na rys. 4.3.

Tabela 4.2. Podsumowanie wyników obliczeń dla przykładów testowych wygenerowanych losowo dla  $n \leq 30$

Rozmiar przykładu [ $n \times m$ ]	$PRD(TS)$ [%]	$PRD(GS)$ [%]	$PRD(SA)$ [%]
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>
10x5	0,00	0,05	0,00
10x10	0,00	0,26	0,01
10x15	0,11	0,38	0,00
<b><i>PRD</i> średnio dla <math>n=10</math></b>	<b>0,04</b>	<b>0,23</b>	<b>0,00</b>
15x5	0,60	0,25	0,00
15x10	0,19	0,86	0,04
15x15	0,16	1,35	0,09
<b><i>PRD</i> średnio dla <math>n=15</math></b>	<b>0,32</b>	<b>0,82</b>	<b>0,04</b>
20x5	0,34	0,11	0,00
20x10	0,36	1,70	0,01
20x15	0,88	2,04	0,00
<b><i>PRD</i> średnio dla <math>n=20</math></b>	<b>0,53</b>	<b>1,28</b>	<b>0,00</b>
25x5	0,77	0,43	0,00
25x10	1,28	1,72	0,00
25x15	0,74	1,93	0,00
<b><i>PRD</i> średnio dla <math>n=25</math></b>	<b>0,93</b>	<b>1,36</b>	<b>0,00</b>
30x5	0,57	0,46	0,01
30x10	1,30	1,70	0,00
30x15	1,04	1,89	0,00
<b><i>PRD</i> średnio dla <math>n=30</math></b>	<b>0,97</b>	<b>1,35</b>	<b>0,00</b>
<b>średnie <i>PRD</i> [%]</b>	<b><u>0,56</u></b>	<b><u>1,01</u></b>	<b><u>0,01</u></b>



Rys. 4.3. Przebiegi poszukiwań lokalnych algorytmów TS, GS, SA dla przykładu nr 7 o rozmiarze 30x15

#### 4.2.2. Weryfikacja wyników uzyskiwanych z zastosowaniem algorytmów metaheurystycznych dla problemu przepływowego z maszynami równoległymi

Problem przepływowo z maszynami równoległymi jest uogólnieniem permutacyjnego problemu przepływowego z użyciem jednej maszyny w danym stanowisku. W niniejszym podrozdziale zostanie przeprowadzona weryfikacja wybranych algorytmów TS, GS, SA, których forma dla przedmiotowego problemu istotnie różni się od postaci tych algorytmów dla permutacyjnego problemu przepływowego. Dla weryfikacji algorytmów przyjęto problem przepływowo z maszynami równoległymi bez dodatkowych ograniczeń, z funkcją celu  $C_{max}$ . Jest to zagadnienie, na bazie którego przeprowadza się weryfikację nowych algorytmów dla tego rodzaju problemu np. [100]. Model matematyczny tego problemu został przedstawiony w p. 3.3. Znalezienie harmonogramu dla przyjętej zmiennej decyzyjnej – permutacji podziału rozłącznego  $\pi$  - wiąże się z podaniem terminów zakończenia wykonywania zadań ze zbioru  $J$ . Terminy te można określić ze wzoru o postaci rekurencyjnej:

$$C_{k, \pi_{ki}(l)} = \max\{C_{k, \pi_{ki}(l-1)}, C_{k-1, \pi_{ki}(l)}\} + p_{k, \pi_{ki}(l)},$$

gdzie:  $j = 1, \dots, n, l = 1, \dots, n_{ki}, i = 1, \dots, m_k, k = 1, \dots, m, \pi_{ki}(0) = 0, C_{k,0} = 0, C_{0j} = 0$ .

Terminy zakończenia wykonywania poszczególnych operacji mogą być znalezione w czasie rzędu  $O(nm)$ . Dla wybranego kryterium  $C_{max}$  (minimum czasu realizacji wszystkich zadań) poszukiwana jest permutacja  $\pi^* \in \Pi$ , dla której:

$$C_{max}(\pi^*) = \min_{\pi \in \Pi} C_{max}(\pi), \quad \text{gdzie } C_{max}(\pi) = \max_{j \in J} C_{m,j}.$$

Podobnie jak w p. 4.2.1, ocena wyników otrzymywanych za pomocą trzech wybranych algorytmów metaheurystycznych będzie miała dwa etapy. W pierwszym zostaną zastosowane przykłady testowe podane w literaturze związanej z teorią szeregowania zadań [146]. Należy tutaj zaznaczyć, że liczba przykładów testowych dla przedmiotowego problemu jest zdecydowanie mniejsza niż podanych w pracach [138], [139] dla permutacyjnego problemu przepływowego, co wiąże się z mniejszą ilością prac badaczy na ten temat. Podobnie jak w p. 4.2.1, ze względu na zbyt małą reprezentatywność tych przykładów dla problemów planowania przedsięwzięć budowlanych, zaplanowano przeprowadzenie drugiego etapu weryfikacji. Etap ten będzie polegał na poddaniu ocenie rezultatów wybranych algorytmów uzyskanych przy zastosowaniu przykładów wygenerowanych losowo o rozmiarach  $n = 5, 10, 15, 20, 25$  zadań.

W etapie pierwszym zastosowane zostaną przykłady z pracy [146], które są związane z praktycznymi problemami szeregowania zadań linii produkcyjnej w firmie wytwarzającej karty obwodów drukowanych. Firma ta przyjmuje dziennie sześć zleceń wyprodukowania

określonej liczby różnych kart. Montaż każdej karty stanowi zadanie składające się zawsze z trzech operacji i jest ono wykonywane na trzech rodzajach maszyn. Każdą z trzech operacji może wykonywać określona liczba maszyn pracujących równolegle. Problemem do rozwiązania jest znalezienie kolejności montażu kart wchodzących w skład dziennego zlecenia, aby czas trwania pracy był jak najkrótszy. Zlecenia te stanowią przykłady testowe, które można zamodelować jako problem przepływowy z maszynami równoległymi. Rozmiary tych przykładów są następujące  $n \times m = 51 \times 3$  („day 1”),  $38 \times 3$  („day 2”),  $38 \times 3$  („day 3”),  $36 \times 3$  („day 4”),  $40 \times 3$  („day 5”),  $30 \times 3$  („day 6”). Dane odnośnie czasów trwania operacji są zawarte w pracy [146]. Wartości parametrów sterujących w trzech testowanych algorytmach przyjęto następująco:

- algorytm TS: długość listy tabu  $LT = 1,0mn$ , nie wprowadzono dodatkowych ograniczeń liczności zbioru otoczenia  $N_{\pi}$ , maksymalna liczba iteracji – 10000,
- algorytm GS: licznosc populacji –  $10n$ , licznosc pokolenia jako odsetek licznosci populacji –  $0,2 \times 10n = 2n$ , prawdopodobienstwo krzyzowania –  $1,0$ , licznosc zbioru osobnikow mutowanych w pokoleniu  $0,2 \times 10n$ , prawdopodobienstwo mutacji –  $1,0$ , licznosc zbioru osobnikow najgorzej przystosowanych z populacji, ktore sa wymieniane na nowe –  $0,9 \times 10n = 9n$ , liczba iteracji, po ktorej nastepuje wymiana i zmiana operatora krzyzowania - 100, calkowita liczba iteracji – 20000,
- algorytm SA: wartosci temperatur poczatkowa  $T_0 = 60$ , koncowa  $T_N = 0,5$ , liczba losowan wystepujaca w kroku 1 -  $1,0n$ , wartosc parametru  $\varepsilon = 0,3$ , liczba iteracji – 20000.

Podane powyzej parametry zostaly ustalone na podstawie prac [100], [106], [108], [134]. Kazdy z rozpatrywanych przykladow testowych byl rozwiazywany siedmiokrotnie. Nastepnie srednia arytmetyczna z tych obliczen byla odniesiona do wartosci oszacowania dolnego ograniczenia funkcji celu  $LBC_{\max}$ , podanej dla tych przykladow w pracy [146], poprzez obliczenie dla kazdego przykladu sredniego bledu wzglednego  $PRD$  danego algorytmu:

$$PRD(A) = 100\%(C^A - LBC_{\max}) / LBC_{\max},$$

gdzie:  $C^A$  – wartosc funkcji celu uzyskana danym algorytmem,  $LBC_{\max}$  – wartosci oszacowania dolnego ograniczenia funkcji celu. Ponadto, dla porownania jakosci otrzymanych rezultatow, podano rowniez bledy wzgledne algorytmu heurystycznego W zaproponowanego w pracy [146] oraz algorytmu TSAB podanego w pracy [100] (przedmiotowe przyklady testowe byly rowniez w tej pracy analizowane) i odniesione do wartosci oszacowania dolnego ograniczenia funkcji celu  $LBC_{\max}$  z pracy [146]. Podsumowanie wynikow obliczen dla przykladow testowych w pierwszym etapie weryfikacji wybranych algorytmow przedstawiono w tabeli 4. 3.

Oznaczenie przykładowy wg [146]	$C_{\max}$ uzyskany w danej próbie dla $i=$							Wartość średnia $C_{\max}$	Oszacowanie $LBC_{\max}$ wg [146]	Wartość $C_{\max}$ wg [146] - algorytm W	Wartość $C_{\max}$ wg [100] - algorytm TSAB	$PRD$ [%]	$PRD(W)$ [%]	$PRD(TSAB)$ [%]
	1	2	3	4	5	6	7							
<i>l</i>	2	3	4	5	6	7	8	9	10	11	12	13	14	15
<b>Algorytm TS</b>														
“day1”	799	776	782	787	777	779	785	783,57	720	784	765	<b>8,83</b>	8,89	6,25
“day2”	782	793	777	777	790	783	784	783,71	715	789	753	<b>9,61</b>	10,35	5,31
“day3”	787	785	786	787	782	785	780	784,57	694	785	760	<b>13,05</b>	13,11	9,51
“day4”	788	785	789	786	786	785	785	786,29	694	796	761	<b>13,30</b>	14,70	9,65
“day5”	968	963	971	967	968	968	967	967,43	963	964	963	<b>0,46</b>	0,10	0,00
“day6”	681	678	678	687	683	678	678	680,43	584	686	661	<b>16,51</b>	17,47	13,18
<b>średni <math>PRD(TS)</math>:</b>											<b>10,29</b>			
<b>Algorytm GS</b>														
“day1”	815	820	831	828	810	815	825	820,57	720	784	765	<b>14,58</b>	8,89	6,25
“day2”	825	826	817	803	811	810	822	816,29	715	789	753	<b>14,97</b>	10,35	5,31
“day3”	807	814	842	795	800	799	821	811,14	694	785	760	<b>18,30</b>	13,11	9,51
“day4”	802	819	802	809	807	812	805	808,00	694	796	761	<b>15,99</b>	14,70	9,65
“day5”	977	996	977	982	969	975	980	979,43	963	964	963	<b>1,77</b>	0,10	0,00
“day6”	683	684	685	689	688	689	681	685,57	584	686	661	<b>16,61</b>	17,47	13,18
<b>średni <math>PRD(GS)</math>:</b>											<b>13,70</b>			
<b>Algorytm SA</b>														
“day1”	786	799	771	781	792	783	779	784,43	720	784	765	<b>8,19</b>	8,89	6,25
“day2”	786	778	782	772	775	782	791	780,86	715	789	753	<b>10,63</b>	10,35	5,31
“day3”	786	782	791	777	782	781	789	784,00	694	785	760	<b>13,69</b>	13,11	9,51
“day4”	793	792	792	799	786	786	800	792,57	694	796	761	<b>15,27</b>	14,70	9,65
“day5”	963	963	963	963	963	963	963	963,00	963	964	963	<b>0,00</b>	0,10	0,00
“day6”	685	691	678	680	684	685	691	684,86	584	686	661	<b>18,32</b>	17,47	13,18
<b>średni <math>PRD(SA)</math>:</b>											<b>11,02</b>			

W etapie drugim przeprowadzono weryfikację polegającą na dokładniejszej ocenie uzyskiwanych rezultatów dla  $n \leq 25$  zadań. Wybrano następujące rozmiary przykładów:  $n \times m = 5 \times 5, 5 \times 10, 5 \times 15, 10 \times 5, 10 \times 10, 10 \times 15, 15 \times 5, 15 \times 10, 15 \times 15, 20 \times 5, 20 \times 10, 20 \times 15, 25 \times 5, 25 \times 10, 25 \times 15$ , ( $n$  – liczba zadań,  $m$  – liczba stanowisk). W każdym rozmiarze utworzono po siedem przykładów. Czasy trwania poszczególnych operacji zostały wygenerowane losowo z zastosowaniem rozkładu jednostajnego [1,99]. Liczba maszyn w każdym ze stanowisk była ustalona w sposób losowy dla każdego rozmiaru przykładu  $n = 5, 10, 15, 20, 25$  i przyjęta jako stała dla każdego z badanych algorytmów. Wartości parametrów sterujących zostały ustalone następująco:

- algorytm TS: długość listy tabu  $LT = \frac{1}{3}mn$ , wprowadzono dodatkowe ograniczenia liczości zbioru otoczenia  $N_\pi$  przez ograniczenie liczby ruchów „włóż” wykonywanych na operacji z danej pozycji do  $0,5(n-1)$  pozycji (maksymalnie możliwych jest  $n-1$  wstawień), maksymalna liczba iteracji – 5000,
- algorytm GS: liczość populacji –  $5n$ , liczość pokolenia jako odsetek liczości populacji –  $0,2 \times 5n = n$ , prawdopodobieństwo krzyżowania – 1,0, liczość zbioru osobników mutowanych w pokoleniu  $0,2 \times 5n = n$ , prawdopodobieństwo mutacji – 1,0, liczość zbioru osobników najgorzej przystosowanych z populacji, które są wymieniane na nowe –  $0,9 \times 5n = 4,5n$ , liczba iteracji, po której następuje wymiana i zmiana operatora krzyżowania - 100, całkowita liczba iteracji – 10000,
- algorytm SA: wartości temperatur początkowa  $T_0 = 60$ , końcowa  $T_N = 0,5$ , liczba losowań występująca w kroku 1 -  $1,0n$ , wartość parametru  $\varepsilon = 0,3$ , całkowita liczba iteracji – 10000.

Wartości parametrów zostały ustalone na podstawie prac [100], [106], [108], [134]. Zmniejszenie liczby iteracji i wartości niektórych parametrów w stosunku do wartości z etapu pierwszego zostało podyktowane czasem trwania obliczeń. Przyjęto dodatkowy warunek (jak w p. 4.2.1), który wynikał z możliwości praktycznego przeprowadzenia badań, że czas obliczeń jednej próby dla danego przykładu wybranym algorytmem nie powinien przekraczać trzech godzin. Każdy z rozpatrywanych przykładów testowych był rozwiązywany siedmiokrotnie. Następnie średnia arytmetyczna z siedmiu prób była odniesiona do minimalnej średniej arytmetycznej z siedmiu prób dla wartości kryterium wybranego obliczeń trzema algorytmami TS, GS, SA przez obliczenie średniego błędu względnego według metody drugiej podanej we wstępie do niniejszego podrozdziału:

$PRD(A) = 100\%(C^A - C^*) / C^*$  - średni błąd względny (średnia procentowa różnica między wartością funkcji celu dla algorytmu  $A$  i wartością  $C^*$  odniesiona do wartości  $C^*$ ), gdzie:



$C^A$  – wartość funkcji celu otrzymana wybranym algorytmem  $A$ ,  $A \in \mathcal{A}$ ,

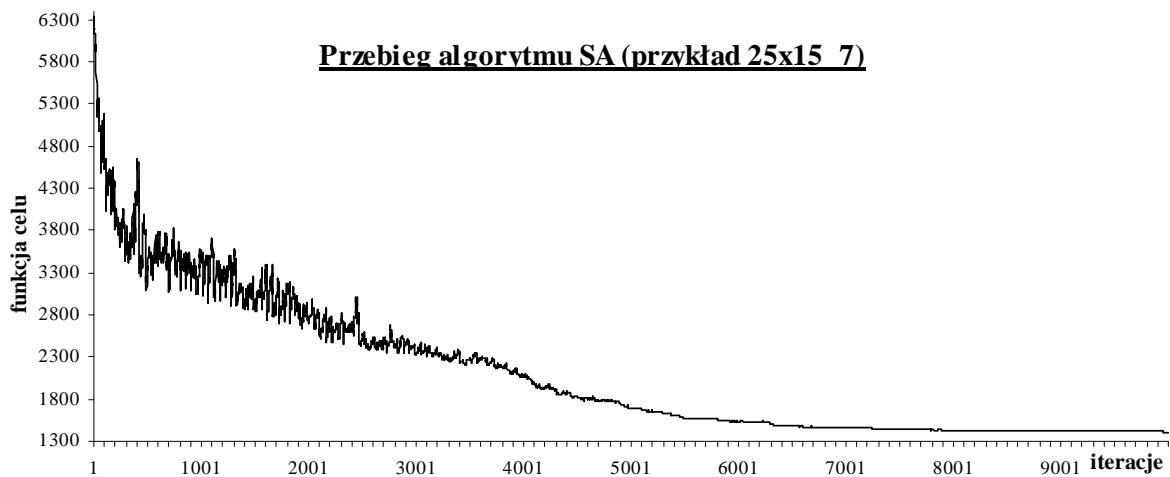
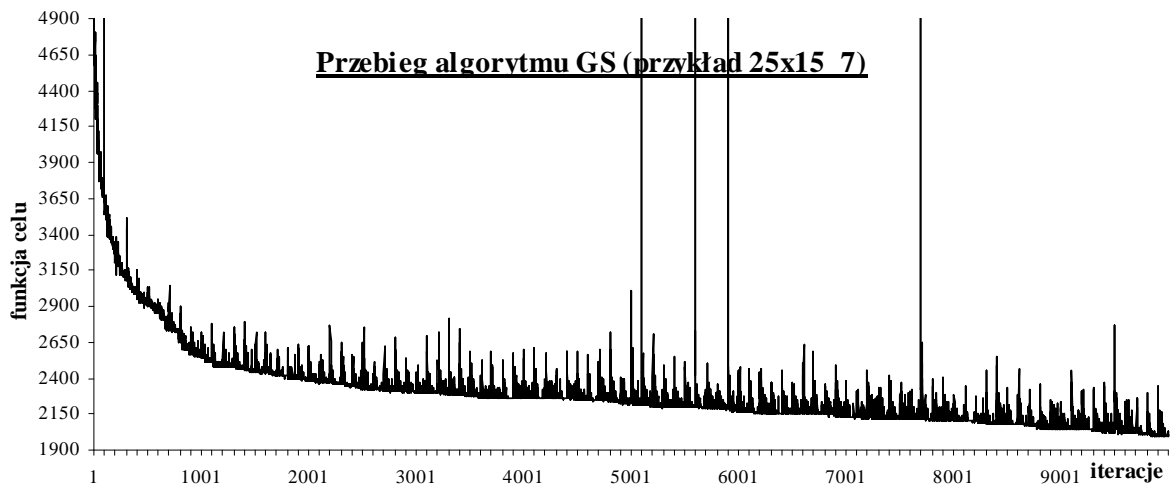
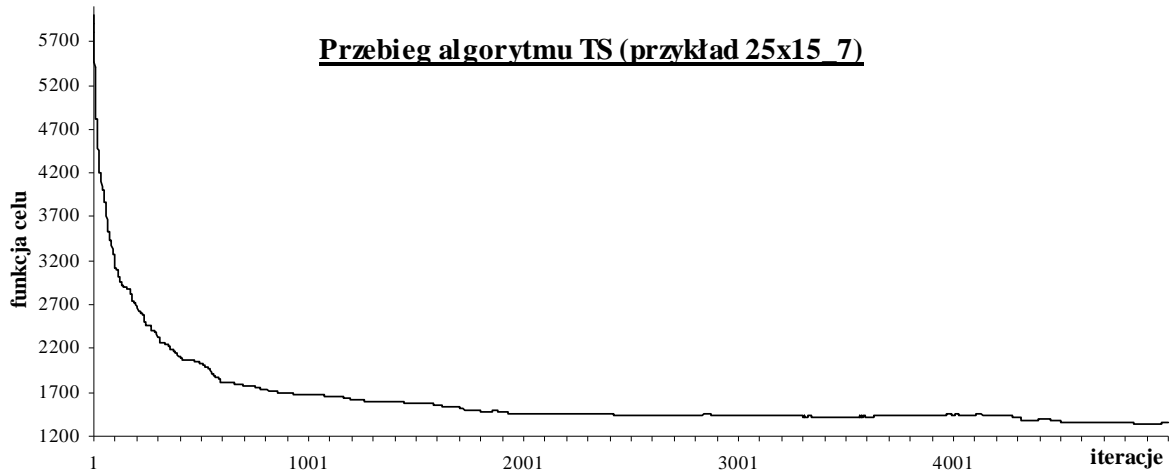
$C^*$  - najlepsza wartość funkcji celu otrzymana jednym z badanych algorytmów, tj.

$$C^* = \min_{A \in \mathcal{A}} C^A.$$

Do zbioru  $\mathcal{A}$  badanych algorytmów należą algorytmy TS, GS, SA. Podsumowanie wyników obliczeń dla przykładów testowych w drugim etapie weryfikacji wybranych algorytmów przedstawiono w tabeli 4.4. Przykładowe przebiegi poszukiwań lokalnych zastosowanych algorytmów przedstawiono na rys. 4.4.

Tabela 4.4. Podsumowanie wyników obliczeń dla przykładów testowych wygenerowanych losowo dla  $n \leq 25$

Rozmiar przykładu [ $n \times m$ ]	$PRD(TS)$ [%]	$PRD(GS)$ [%]	$PRD(SA)$ [%]
<i>l</i>	2	3	4
5x5	0,00	0,86	0,49
5x10	0,00	5,93	2,21
5x15	0,06	5,43	1,89
<b><i>PRD</i> średnio dla <math>n=5</math></b>	<b>0,02</b>	<b>4,07</b>	<b>1,53</b>
10x5	1,00	5,92	0,33
10x10	2,18	5,75	0,00
10x15	2,35	6,56	0,08
<b><i>PRD</i> średnio dla <math>n=10</math></b>	<b>1,84</b>	<b>6,08</b>	<b>0,14</b>
15x5	3,17	8,99	0,12
15x10	1,44	11,65	0,61
15x15	2,61	17,20	0,00
<b><i>PRD</i> średnio dla <math>n=15</math></b>	<b>2,41</b>	<b>12,61</b>	<b>0,24</b>
20x5	3,20	11,61	0,00
20x10	1,04	18,17	0,46
20x15	0,78	27,94	0,86
<b><i>PRD</i> średnio dla <math>n=20</math></b>	<b>1,68</b>	<b>19,24</b>	<b>0,44</b>
25x5	6,17	7,61	0,00
25x10	3,21	27,76	0,06
25x15	0,90	40,28	1,16
<b><i>PRD</i> średnio dla <math>n=25</math></b>	<b>3,43</b>	<b>25,22</b>	<b>0,41</b>
<b>średnie <i>PRD</i> [%]</b>	<b><u>1,88</u></b>	<b><u>13,44</u></b>	<b><u>0,55</u></b>



Rys. 4.4. Przebiegi poszukiwań lokalnych algorytmów TS, GS, SA dla przykładu nr 7 o rozmiarze 25x15

### 4.3. Podsumowanie

W niniejszym rozdziale przedstawiono trzy algorytmy metaheurystyczne: TS, GS, SA, które wybrano do poszukiwania rozwiązań zagadnień optymalizacyjnych w sformułowanych w rozdziale 5 modelach przedsięwzięć wieloobiektowych. Ponadto przedstawiono ich formy w zastosowaniu do rozwiązywania problemów: permutacyjnego przepływowego oraz przepływowego z maszynami równoległymi. Problemy te będą stanowiły podstawę tworzenia nowych modeli harmonogramowania przedsięwzięć wieloobiektowych. Celem niniejszego rozdziału była również ocena wyników dostarczanych przez przedmiotowe algorytmy w trakcie rozwiązywania problemów przepływowych. W związku z tym, przeprowadzono odpowiednie badania weryfikacyjne, osobno dla obu problemów.

W badaniach weryfikacyjnych algorytmów dla permutacyjnego problemu przepływowego w pierwszym etapie wykorzystano niektóre przykłady testowe stosowane aktualnie przez badaczy zagadnień teorii szeregowania zadań i znajdujące się w pracy [138]. Wyniki tych badań pokazują, że najlepsze rezultaty dostarcza algorytm TS (średni błąd względny – 0,05% w stosunku do rozwiązań optymalnych dla rozmiaru  $n=20$ ), następnie algorytm SA (średni błąd względny – 0,24%) i algorytm GS (średni błąd względny – 1,44%). W etapie drugim przeprowadzono weryfikację rezultatów otrzymywanych na bazie przykładów wygenerowanych losowo rozmiarach  $n = 10 - 30$ . Ze względu na długi czas obliczeń dla algorytmu TS (spowodowany głównie potrzebą wyznaczania wartości funkcji celu dla całego otoczenia  $N_\pi$ ) ograniczono licznosc zbioru otoczenia  $N_\pi$  oraz liczbę iteracji, co znacznie wpłynęło na osiągane rezultaty. W wyniku przyjętych założeń parametrów sterujących algorytmów w tym etapie, najlepsze rezultaty osiągnął algorytm SA (średni błąd względny – 0,01% w stosunku do najlepszego wyniku osiąganego przez jeden z trzech rozpatrywanych algorytmów), a następnie TS (0,56%) i GS (1,01%). Badania weryfikacyjne wybranych algorytmów dla permutacyjnego problemu przepływowego dostarczają następujących wniosków:

1. Najsilniejszymi narzędziami spośród rozpatrywanych algorytmów są algorytmy TS i SA, a najsłabszym algorytm GS.
2. Zastosowane ze względów praktycznych (zbyt długi czas trwania obliczeń) ograniczenia dla wartości niektórych parametrów sterujących (szczególnie dla algorytmu TS) istotnie wpływają na jakość uzyskiwanych rezultatów.
3. Wyniki dostarczane przez testowane algorytmy (szczególnie TS i SA) są często bardzo zbliżone do optymalnych lub optymalne, a rozrzut średniego błędu względnego między wynikami dostarczonymi przez trzy algorytmy jest stosunkowo niewielki - do ok. 1,5%.

4. Stwierdzono zbyt długi czas obliczeń (szczególnie dla algorytmu TS dla przykładów o rozmiarach  $n = 25 - 30$ ), który wynikał z cech środowiska Mathematica, w którym dokonano implementacji programowej i w którym nie była możliwa kompilacja procedur przyspieszająca działanie algorytmów.

Badania weryfikacyjne algorytmów dla problemu przepływowego z maszynami równoległymi zostały przeprowadzone również w dwóch etapach. W pierwszym testowano sześć przykładów z pracy [146]. W tym etapie najlepsze rezultaty osiągnął algorytm TS (średni błąd względny – 10,29% w stosunku do dolnych ograniczeń rozpatrywanych przykładów), a następnie SA (11,02%) i GS (13,70%). Są to rezultaty porównywalne, w niektórych przypadkach lepsze (szczególnie algorytm TS) od wyników uzyskanych przez algorytm W z pracy [146]. Dużo lepsze wyniki uzyskiwane są jednak przez algorytm TSAB z pracy [100], który jest zaawansowanym algorytmem przeszukiwania z zakazami, stosującym dodatkowe narzędzia (detektor cyklu, mechanizm skoku powrotnego). Drugi etap badań weryfikacyjnych polegał na testowaniu przykładów losowych o rozmiarach  $n = 5 - 25$ . Podobnie, jak wcześniej, zastosowano ograniczenia wartości niektórych parametrów sterujących (algorytm TS) spowodowane zbyt długim czasem obliczeń. Najlepsze rezultaty osiągnął algorytm SA (średni błąd względny – 0,55% w stosunku do najlepszego wyniku osiąganego przez jeden z trzech rozpatrywanych algorytmów), a następnie TS (1,88%) i GS (13,44%). Wnioski z badań weryfikacyjnych wybranych algorytmów dla problemu przepływowego z maszynami równoległymi są zbliżone do wniosków z badań nad algorytmami dla permutacyjnego problemu przepływowego:

1. Najsilniejszymi narzędziami spośród rozpatrywanych algorytmów są również algorytmy TS i SA, a najsłabszym algorytm GS.
2. Zauważono duży wpływ ograniczania wartości niektórych parametrów sterujących na jakość uzyskiwanych rezultatów (spowodowany zbyt długim czasem trwania obliczeń – algorytm TS). Ważnym w tym przypadku pozostaje wniosek 4 z badań weryfikacyjnych algorytmów dla permutacyjnego problemu przepływowego.
3. W pierwszym etapie obliczeń rezultaty uzyskane przez trzy algorytmy są zbliżone do wyników uzyskanych przez algorytm W z pracy [146] i jednoznacznie gorsze niż uzyskane przez algorytm TSAB z pracy [100]. Rozrzut wyników badanych algorytmów w tym etapie jest większy niż w badaniach dla permutacyjnego problemu przepływowego i wynosi średnio 3,4%. W drugim etapie rozrzut rezultatów wzrasta do średnio 12,9%. W obu etapach wzrost rozrzutu wyników jest spowodowany słabymi wynikami osiąganymi przez algorytm GS.

Na podstawie przeprowadzonych badań można stwierdzić, że trzy wybrane algorytmy metaheurystyczne TS, GS, SA (a szczególnie TS i SA) są skutecznymi narzędziami do rozwiązywania problemów optymalizacyjnych, które będą występowały w nowych modelach planowania przedsięwzięć wieloobektowych przedstawionych w rozdziale 5.

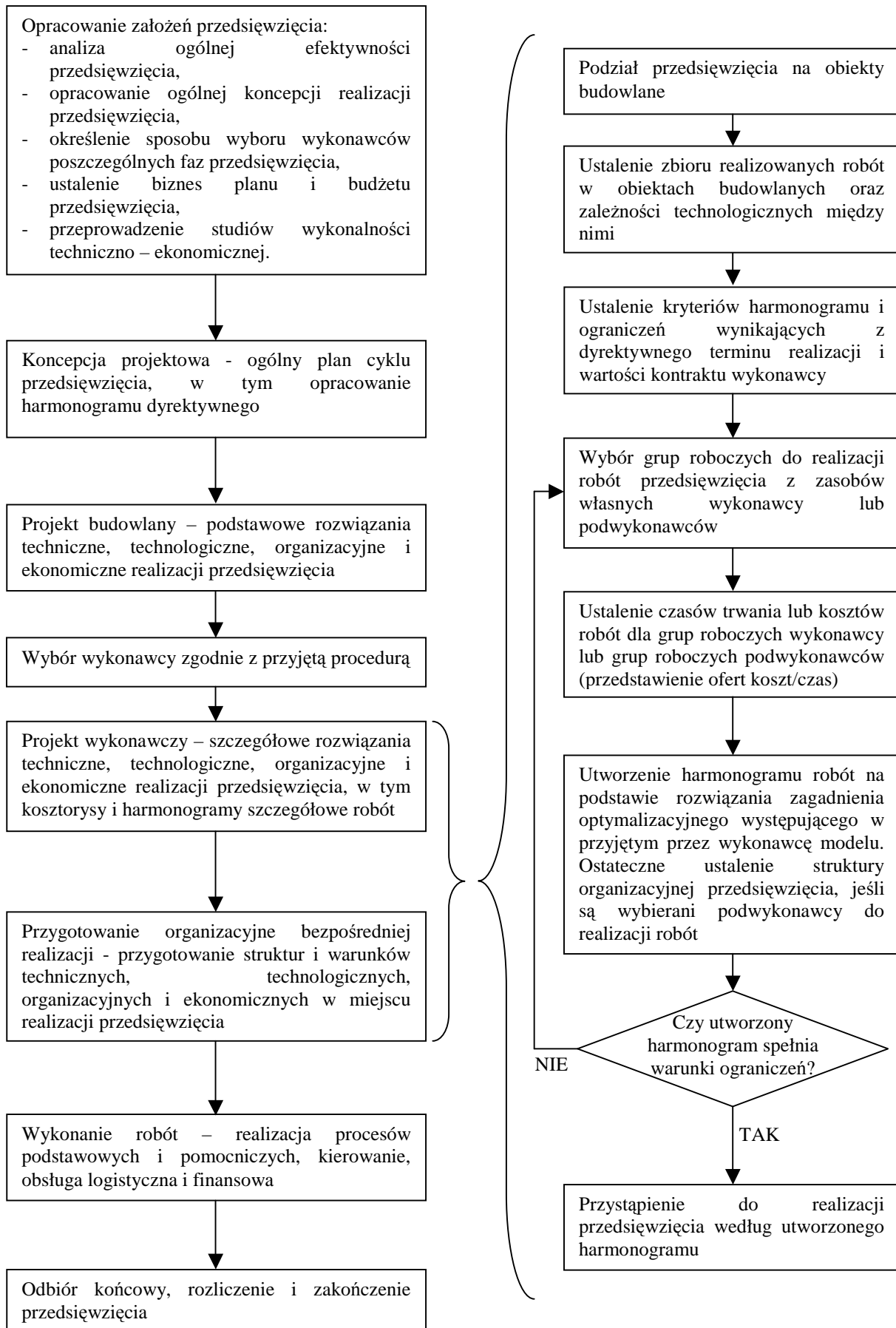
## **5. Nowe modele harmonogramowania przedsięwzięć wieloobektowych z zastosowaniem systemu pracy potokowej**

Proponowane w niniejszej pracy nowe modele harmonogramowania wykorzystujące aktualne osiągnięcia teorii szeregowania zadań (TSZ) należą do grupy modeli harmonogramowania stosujących system pracy potokowej. Analiza przedstawionych w rozdziale 3 niektórych zagadnień TSZ oraz przedstawionej w p. 3.6 analogii metod sprzężeń czasowych (stanowiących uogólnienie metody pracy równomiernej) z tymi zagadnieniami pozwalają na postawienie tezy, że jest możliwe rozszerzenie obszaru zastosowań modeli pracy potokowej w planowaniu przedsięwzięć budowlanych. Rozszerzenie to może być dokonane przez sformułowanie nowych modeli pracy potokowej uwzględniających dodatkowe parametry, ograniczenia i kryteria, które nie były do tej pory rozpatrywane w modelach przedstawionych w literaturze, a występujące w ramach TSZ, w modelach i metodach planowania przedsięwzięć typu kompleks operacji.

Cechą wszystkich rozpatrywanych w pracy, nowych modeli jest przyjęcie sytuacji deterministycznej, gdy znane są warunki techniczne, technologiczne i organizacyjne, możliwy jest dokładny przedmiar robót, zasoby (grupy robocze, kapitał) są dostępne w chwili i na czas ich zapotrzebowania oraz realizują roboty z przyjętą wydajnością i jakością, brak jest istotnych zakłóceń wykonywania robót [74]. Wspólnym ograniczeniem występującym w pierwszej grupie formułowanych modeli będą relacje kolejnościowe między robotami, wyrażone niezmienną dla każdego obiektu sekwencją (rozumiane tak samo jak w metodach sprzężeń czasowych, w pracy [88] i analogicznie do kolejności wykonywania operacji w problemach przepływowych w TSZ). Występowanie relacji kolejnościowych między robotami, wyrażone sekwencją, jest ograniczeniem spotykanym najczęściej w obiektach budownictwa mieszkaniowego (np. budynki mieszkalne), uprzemysłowionego, obiektach liniowych. W drugiej grupie modeli przyjęto relacje kolejnościowe robót dla określonego obiektu budowlanego, które są dane grafem. Przyjmuje się, że graf ten jest taki sam dla wszystkich obiektów wchodzących w skład przedsięwzięcia. Ograniczenie to stanowi uogólnienie ograniczenia, które zostało założone w pierwszej grupie modeli. Uwzględnienie jego pozwoli na rozszerzenie możliwości stosowania modeli na przedsięwzięcia wymagające równoległego prowadzenia robót w obiekcie, zastosowania powiązań pomiędzy robotami jak w metodzie CPM, innych, dodatkowych zależności kolejnościowych nie występujących w modelach z grupy pierwszej. Ze względu na ujednoczenie nazewnictwa oraz wyróżnienie głównego przeznaczenia proponowanych w rozprawie modeli, przedsięwzięcia, w których mogą być stosowane proponowane modele, będą nazywane w dalszej części

przedsięwzięciami wieloobiekowymi. W skład danego przedsięwzięcia wieloobiektowego wchodzi obiekty budowlane, które mają takie same cechy jak fronty robót/działki robocze w metodach sprzężeń czasowych, zadania w TSZ. Inne, dodatkowe parametry, ograniczenia i kryteria dotyczące przedstawionych dwóch grup modeli będą przedstawiane w poszczególnych podrozdziałach.

Proponowane nowe modele harmonogramowania mogą znaleźć zastosowanie podczas organizacji przedsięwzięć budowlanych, w których jest możliwe zastosowanie systemu pracy potokowej. Szczegółowe określenie, które przedsięwzięcia budowlane mogą być realizowane w tym systemie zostało przedstawione w p. 2.2. Pozycja proponowanych modeli w procesie realizacji przedsięwzięcia budowlanego oraz poszczególne etapy tworzenia harmonogramu na podstawie przyjętego modelu zostały przedstawione na rys. 5.1. Przedstawiony schemat odnosi się wprost do systemów realizacji przedsięwzięć najczęściej obecnie spotykanych, które zakładają istnienie dwóch oddzielnych uczestników: inwestora i wykonawcy [74]. Inwestor jest uczestnikiem, który narzuca warunki ograniczające: termin zakończenia przedsięwzięcia lub jego koszt (wartość kontraktu wykonawcy). Usytuowanie formułowanych w pracy modeli zostało przedstawione na rys. 5.1 w sytuacji, gdy wykonawca zawarł już umowę na realizację przedsięwzięcia, a jego zadanie polega na utworzeniu harmonogramu zgodnego z ograniczeniami narzuconymi przez inwestora i z możliwościami własnymi lub jego podwykonawców. Harmonogram ten będzie rozwiązaniem minimalizującym przyjęte przez wykonawcę kryterium (termin zakończenia przedsięwzięcia lub jego koszt) w zależności od przyjętego modelu przedsięwzięcia. Sformułowane modele mogą być zastosowane również w sytuacji, gdy wykonawca jest na etapie składania oferty inwestorowi dotyczącej realizacji przedsięwzięcia. W tym przypadku utworzony harmonogram stanie się składnikiem takiej oferty. Przedstawione zadania inwestora i wykonawcy oraz sposób zastosowania przedmiotowych modeli mogą istnieć w systemach generalnego wykonawcy, „projektowanie i wykonanie” z generalnym wykonawcą robót czy systemie menedżerskim z generalnym wykonawcą robót [74]. W innych systemach tego typu, gdzie rolę generalnego wykonawcy przejmuje menedżer budowy zastosowanie przedmiotowych modeli jest analogicznie do systemów z generalnym wykonawcą. Istnieje również możliwość zastosowania rozpatrywanych modeli w systemach, gdzie inwestor jednocześnie pełni rolę wykonawcy. Takim systemem jest system wykonawstwa własnego (ma on współcześnie ograniczone znaczenie), gdzie inwestor jest odpowiedzialny za projektowanie i wykonawstwo robót [74].



Rys. 5.1. Fazy realizacji przedsięwzięcia budowlanego (na podstawie [74]) ze szczególnym uwzględnieniem zastosowania sformułowanych modeli



Z związku z tym, on sam lub jego służby mogą dokonać wyboru zasobów do realizacji przedsięwzięcia, ustalenia ograniczeń terminu jego zakończenia lub kosztu oraz ustalić harmonogram spełniający warunki własnego biznes planu oraz budżetu posługując się przyjętym modelem.

Formułowane modele systemu pracy potokowej mogą znaleźć zastosowanie m. in. podczas kontraktowania realizacji serii obiektów o podobnym charakterze (*serial contracting*) [78]. W kontraktach takich, po należytych wywiązaniu się z realizacji pierwszego obiektu, wykonawca ma realną możliwość otrzymania zlecenia na wykonanie serii analogicznych obiektów. Ten sposób kontraktowania umożliwia wykorzystanie doświadczenia zdobytego przez wykonawcę zgodnie z tzw. krzywą uczenia, a także wypracowanych, korzystnych zasad współpracy z inwestorem.

We wszystkich rozpatrywanych modelach zostało zastosowane ogólne pojęcie grup roboczych jako zasobu realizującego roboty jednego rodzaju w obiektach. Mogą być nimi brygady np. generalnego wykonawcy, zespoły ludzkie, przedsiębiorstwa lub dowolne jednostki organizacyjne realizujące tylko ściśle określony rodzaj robót. Ten rodzaj zasobu można zaklasyfikować jako zasób czynny, czyli środek mający określone możliwości działania oraz nie ulegający zużyciu w toku działania [74], [88]. We wszystkich rozpatrywanych w pracy modelach przyjmuje się, że dostępność grup roboczych i obiektów przedsięwzięcia jest nieograniczona. W niektórych modelach dodatkowym parametrem ekonomicznym zasobów czynnych będzie koszt robót wybranych grup roboczych np. podwykonawców wykonujących niektóre roboty dla generalnego wykonawcy [74], [88]. Według pojęć stosowanych w TSZ, analizowane grupy robocze w przedmiotowych modelach są zasobami odnawialnymi (analogicznymi do maszyn rozważanych w TSZ), których podzielność jest dyskretna, z dokładnością do jednostki i które mogą posiadać swój koszt [134].

Formułowane modele stanowią wynik badań autora nad problematyką metod harmonogramowania wykorzystujących system pracy potokowej. We wcześniejszej ich fazie sformułowano modele stosujące relacje kolejnościowe wyrażone sekwencją, a w późniejszej - modele stanowiące uogólnienie poprzednich, z relacjami kolejnościowymi danymi grafem. Dla przejrzystości ich prezentacji, każdy z modeli przedstawiono oddzielnie, mimo że część z parametrów lub ograniczeń jest taka sama.

Implementację programową przedstawionych i zweryfikowanych w rozdziale 4 algorytmów metaheurystycznych TS, GS i SA dla formułowanych nowych modeli systemów pracy potokowej dokonano w środowisku Mathematica. W rozdziale 6 zostanie

przedstawiona weryfikacja tych modeli w przykładach zastosowań dla przypadków planowania przedsięwzięć budowlanych.

### **5.1. Nowe modele harmonogramowania przedsięwzięć wieloobektowych realizowanych w systemie pracy potokowej z relacjami kolejnościowymi między robotami wyrażonymi sekwencją**

W niniejszym podrozdziale przedstawione zostaną nowe modele pracy potokowej przedsięwzięć wieloobektowych uwzględniające kryterium czasu realizacji całego przedsięwzięcia oraz kryteria związane z jego kosztem. Wspólnym ograniczeniem występującym w tej grupie modeli są relacje kolejnościowe między robotami wyrażone sekwencją, która dla każdego obiektu jest niezmienna. Pierwsze z kryteriów jest najczęściej stosowane przy tworzeniu optymalnych harmonogramów realizacji przedsięwzięć budowlanych, zarówno typu kompleks operacji, jak i wykonywanych w systemie pracy potokowej (np. w metodach sprzężeń czasowych). Z tego powodu zostało ono rozpatrzone w pierwszej kolejności. Modele takie mogą mieć zastosowanie w sytuacji, gdy dane przedsięwzięcie budowlane jest realizowane w systemie pracy potokowej, a wykonawca np. w systemie generalnego wykonawstwa czy w systemach z menedżerem budowy) ma zamiar użyć grup roboczych wyspecjalizowanych do wykonywania jednego rodzaju robót. Zakłada się, że wykonawca dokonał już jednoznacznego wyboru grup roboczych (mogą one stanowić np. zasoby własne generalnego wykonawcy) i w związku z tym, nie istnieją inne, dodatkowe parametry czy ograniczenia mające wpływ na alokację grup roboczych. Wykonawca może dysponować tylko jedną grupą roboczą do wykonywania robót jednego rodzaju (model przedstawiony w p. 5.1.1) lub może być ich więcej (liczba ich może być różna dla każdego rodzaju robót - model przedstawiony w p. 5.1.2). W pierwszym przypadku alokacja grup roboczych jest jednoznacznie określona, gdyż tylko jedna, wskazana przez wykonawcę grupa robocza, może wykonywać robotę jednego rodzaju we wszystkich obiektach, przechodząc z obiektu poprzedniego na następny w kolejności ustalonej tak samo dla wszystkich grup roboczych. Poszukiwana jest taka kolejność realizacji obiektów, aby przyjęte kryterium (termin realizacji całego przedsięwzięcia) zostało zminimalizowane. W drugim modelu (p. 5.1.2) alokacja i kolejność realizacji obiektów będzie ustalana dla każdej z grup roboczych indywidualnie. Poszukiwana z kolei będzie taka kolejność realizacji obiektów oraz alokacja poszczególnych grup roboczych do realizacji robót w obiektach, aby kryterium zostało zminimalizowane. W trzecim modelu przedstawionym w p. 5.1.3 zostaną zastosowane kryteria koszt/czas, podobnie jak w zagadnieniu TCT dla metod sieciowych

opisanym w rozdziale 2 rozprawy. W związku z tymi kryteriami zbiór parametrów tego modelu zostanie rozszerzony o koszty realizacji poszczególnych robót w obiektach. Podstawą do utworzenia tego modelu będzie problem przepływowy z maszynami równoległymi z teorii szeregowania zadań, podobnie jak w modelu z p. 5.1.2.

### 5.1.1. Model systemu pracy potokowej z jedną grupą roboczą wykonującą robotę jednego rodzaju

Rozpatrywany model systemu pracy potokowej wykorzystuje założenia permutacyjnego problemu przepływowego z kryterium czasu wykonywania wszystkich zadań (problem  $FP \mid \mid C_{max}$ ), który jest rozważany w TSZ i został przedstawiony w p. 3.2 rozprawy. Na bazie tego problemu został sformułowany rozważany model w którym zastosowano dodatkowe parametry i ograniczenia. W modelu zakłada się, że robota jednego rodzaju będzie wykonywana przez nie więcej niż jedną grupę roboczą. Formułowany model został przez autora zasygnalizowany w pracy [110].

Model optymalizacyjny przedmiotowego zagadnienia jest następujący:

#### Parametry:

- Przedsięwzięcie tworzy zbiór obiektów budowlanych  $Z = \{Z_1, Z_2, Z_3, \dots, Z_j, \dots, Z_n\}$ .
- Roboty w obiektach wykonują grupy robocze wyspecjalizowane do wykonywania roboty jednego rodzaju. Tworzą one zbiór  $B = \{B_1, B_2, B_3, \dots, B_k, \dots, B_m\}$ , gdzie  $B_k$  reprezentuje jedną grupę roboczą.
- Każdy obiekt  $Z_j \in Z$  wymaga realizacji  $m$  robót, które tworzą zbiór  $O_j = \{O_{j1}, O_{j2}, O_{j3}, \dots, O_{jk}, \dots, O_{jm}\}$ .
- Zakłada się, że robota  $O_{jk} \in O_j$  może być realizowana przez grupę roboczą  $B_k$ . Czas trwania roboty  $O_{jk}$  wykonywanej przez nią wynosi  $p_{jk} > 0$ . Zbiór czasów trwania  $p_j$  robót ze zbioru  $O_j$  określa wektor  $p_j = [p_{j1}, p_{j2}, p_{j3}, \dots, p_{jk}, \dots, p_{jm}]$ . Czasy trwania robót  $p_{jk}$  są ustalane na podstawie nakładów pracy (wyrażonych w roboczogodzinach lub maszynogodzinach) wyznaczonych na podstawie bazy normatywnej (KNR, KSNR, norm zakładowych, kalkulacji własnej) i wielkości grupy roboczej (liczby pracowników)  $B_k$  wykonujących robotę  $O_{jk}$ .
- Zakłada się możliwość występowania przerw technologicznych między robotami i jednoczesnej pracy wielu grup roboczych w obiektach (co było niemożliwe do uwzględnienia np. w metodach sprzężeń czasowych). Czasy trwania przerw między daną robotą i następną ( $s_{jk}^F > 0$ ) lub długość jednoczesnego czasu trwania danej roboty i następną ( $s_{jk}^F < 0$ ) w obiekcie dla zbioru robót  $O_j$  są podane w wektorze  $s_j^F = [s_{j1}^F, s_{j2}^F,$

$s_{j3}^F, \dots, s_{jk}^F, \dots, s_{jm}^F$ ]. Są one odpowiednikami sprzężeń między frontami roboczymi, które są zastosowane w pracy [88]. Podobnie jak w pracy [88], czasy te należy rozumieć jako ograniczenia minimalne (sprzężenia warunkowe). W odróżnieniu od definicji sprzężeń między frontami z pracy [88], w rozpatrywanym modelu czasy te mogą przyjmować wartości dowolne. Dla ujednolicenia nazewnictwa, w dalszej części rozprawy czasy te będą nazywane sprzężeniami między obiektami.

- Dodatkowe czasy niezbędne dla przemieszczania grup roboczych (np. maszyn i sprzętu) między obiektami, zależne są od rodzaju grupy roboczej oraz kolejności realizacji obiektów  $S_k^S = [s_{gh}^S]_{n \times n}$ , gdzie  $g \in [1..n]$ ,  $h \in [1..n]$ ,  $k \in [1..m]$ . Czasy te są uwzględniane w przypadku znaczących odległości między obiektami, w przypadku potrzeby przetransportowania sprzętu (np. żurawi budowlanych) między obiektami, uwzględnienia czasów demontażu i montażu sprzętu w obiektach itd. Należy rozumieć je ograniczenia minimalne. Charakter ich jest taki sam jak sprzężeń warunkowych zdefiniowanych w pracy [88].

#### Ograniczenia:

- Zakłada się kolejność (sekwencję) wykonywania robót wynikająca z technologii:

$$O_{j,k-1} < O_{j,k} < O_{j,k+1} .$$

- Zakłada się, że w dowolnej chwili każda grupa robocza  $B_k$  może wykonywać tylko jedną robotę.
- Zakłada się, że robota  $O_{jk} \in O_j$  jest realizowana nieprzerwanie przez grupę roboczą  $B_k$  przez czas  $p_{jk} > 0$ .

Zmienną decyzyjną jest kolejność  $\pi$  wykonywania obiektów, która dla każdej z grup roboczych jest taka sama i jest następująca:

$$\pi = (\pi(1), \pi(2), \dots, \pi(k), \dots, \pi(n)).$$

Liczność zbioru  $\Pi$  wszystkich permutacji  $\pi \in \Pi$  w tym modelu wynosi  $n!$ .

Kryterium (funkcją celu) jest termin  $C_{max}$  realizacji wszystkich robót we wszystkich obiektach. Zadanie optymalizacyjne w modelu polega na znalezieniu harmonogramu wykonywania robót, który minimalizuje wartość funkcji celu, spełniając ograniczenia podane powyżej.

Przedstawiony model można zidentyfikować wśród modeli TSZ jako problem  $m$  – maszynowy o następujących cechach:

- typ zagadnienia: przepływowy permutacyjny,
- dodatkowe założenia i ograniczenia:

- możliwość występowania dodatkowych sprzężeń między obiektami odpowiadają czasom transportu zadań między maszynami, które mogą mieć wartości dodatnie (*transfer time*) lub ujemne (*time lag*) [134],
- dodatkowe czasy wymagane ze względu na przemieszczenie grup roboczych między obiektami, które są zależne od rodzaju grupy roboczej oraz kolejności realizacji obiektów, odpowiadają czasom przebrojeń maszyn zależnych od rodzaju maszyny i sekwencji zadań (tzw. przebrojenia sekwencyjnie zależne [63] - *seq setup*),
- funkcja celu:  $C_{max}$  (*makespan*).

Model o opisanej postaci w ramach TSZ nie był badany w znanych autorowi pracach.

Przedmiotowy model można przedstawić w postaci grafu [134]. Jest on zależny od kolejności realizacji obiektów tj. permutacji  $\pi$ :

$$G(\pi) = (M \times N, E),$$

gdzie  $M \times N = \{1, \dots, k, \dots, m\} \times \{1, \dots, j, \dots, n\}$  jest zbiorem obciążonych wierzchołków reprezentujących roboty. Obciążenie wierzchołka  $(k, j)$  jest równe  $p_{k,\pi(j)}$ . Zbiór  $E = E^F \cup E^S$  zawiera krawędzie pionowe (technologiczne)  $E^F$  i krawędzie poziome (sekwencyjne, reprezentujące kolejność realizacji obiektów)  $E^S$ . Krawędzie pionowe  $((k, j), (k+1, j))$  grafu  $G(\pi)$  są obciążone wartościami  $s_{k,\pi(j)}^F$ , zaś krawędzie poziome  $((k, j), (k, j+1))$  wartościami  $s_{k,\pi(j)\pi(j+1)}^S$ . Graf  $G(\pi)$  dla badanego przypadku będzie więc regularną, prostokątną siatką  $m \times n$  z obciążeniami węzłów i wszystkich łuków zależnych od permutacji  $\pi$  (rys. 5.2).

Terminy zakończenia poszczególnych robót można określić ze wzoru o postaci rekurencyjnej:

$$C_{k, \pi(j)} = \max \{ C_{k, \pi(j-1)} + s_{k,\pi(j-1)\pi(j)}^S, C_{k-1, \pi(j)} + s_{k-1,\pi(j)}^F \} + p_{k,\pi(j)},$$

gdzie:  $j = 1, \dots, n, k = 1, \dots, m, \pi(0) = 0, C_{k,0} = 0, C_{0,j} = 0$ .

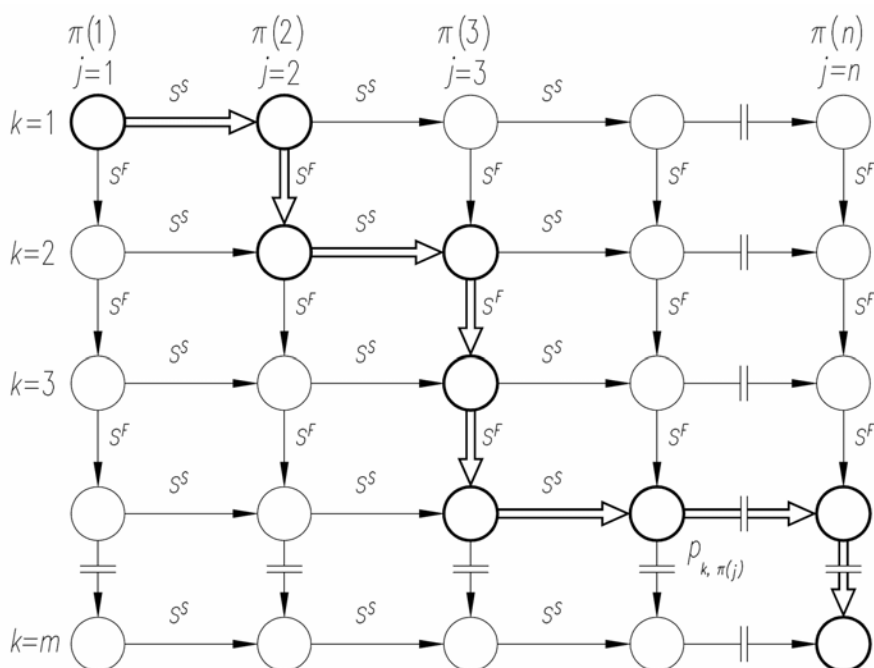
Dla kryterium  $C_{max}$  (minimum czasu realizacji wszystkich robót w obiektach) poszukiwana jest permutacja  $\pi^* \in \Pi$ , dla której:

$$C_{max}(\pi^*) = \min_{\pi \in \Pi} C_{max}(\pi), \quad \text{gdzie } C_{max}(\pi) = C_{m, \pi(n)}.$$

W grafie  $G(\pi)$  można wyróżnić ścieżkę krytyczną (przykład jej jest podany na rys. 5.2).

Przedstawiony powyżej model systemu pracy potokowej jest NP-trudnym zagadnieniem optymalizacyjnym. Ze względu na postać przyjętego ograniczenia dotyczącego możliwości występowania sprzężeń między obiektami (wartości tych sprzężeń mogą być dowolne) nie jest możliwe zastosowanie algorytmu B&B do rozwiązywania zagadnienia optymalizacyjnego, gdyż nie są możliwe do ustalenia funkcje dolnych i górnych ograniczeń. Przyjęto rozwiązanie tego zagadnienia za pomocą trzech rozpatrzonych w rozdziale 4 algorytmów metaheurystycznych: TS, GS, SA. Ze względu na występowanie takiej samej

zmiennej decyzyjnej, jak w permutacyjnym problemie przepływowym, forma tych algorytmów jest taka sama, jak dla tego problemu, i została przedstawiona w p. 4.1.4.1.



Rys. 5.2. Graf  $G(\pi)$  dla rozważanego w p. 5.1.1 modelu systemu pracy potokowej

### 5.1.2. Model systemu pracy potokowej z wieloma grupami roboczymi wykonującymi robotę jednego rodzaju

Model ten stosuje założenia problemu przepływowego z maszynami równoległymi oraz z kryterium czasu wykonywania wszystkich zadań, który jest rozważany w TSZ i został przedstawiony w p. 3.3 niniejszej rozprawy. Na bazie tego problemu sformułowano model stosując dodatkowe parametry i ograniczenia, które zostały zastosowane w p. 5.1.1. W modelu zakłada się, że robota jednego rodzaju może być wykonywana przez więcej niż jedną grupę roboczą.

Model optymalizacyjny przedmiotowego zagadnienia jest następujący:

#### Parametry:

- Przedsięwzięcie tworzy zbiór obiektów budowlanych  $Z = \{Z_1, Z_2, Z_3, \dots, Z_j, \dots, Z_n\}$ .
- Do wykonywania robót przedsięwzięcia istnieją zespoły grup roboczych wykonujące robotę jednego rodzaju i tworzące zbiór  $B = \{B_1, B_2, B_3, \dots, B_k, \dots, B_m\}$ .
- W każdym zespole grup roboczych  $B_k \in B$  znajduje się  $m_k \geq 1$  takich samych grup roboczych (mają takie same wydajności lub składy):  $B_k = \{B_{k1}, B_{k2}, B_{k3}, \dots, B_{ki}, \dots, B_{km_k}\}$ .
- Każdy obiekt  $Z_j \in Z$  wymaga realizacji  $m$  robót, które tworzą zbiór  $O_j = \{O_{j1}, O_{j2}, O_{j3}, \dots, O_{jk}, \dots, O_{jm}\}$ .

- Zakłada się, że robota  $O_{jk} \in O_j$  może być realizowana przez grupę roboczą  $B_{ki} \subset B_k$ . Czas trwania roboty  $O_{jk}$  wykonywanej przez nią wynosi  $p_{jk} > 0$ . Zbiór czasów trwania  $p_j$  robót ze zbioru  $O_j$  określa wektor  $p_j = [p_{j1}, p_{j2}, p_{j3}, \dots, p_{jk}, \dots, p_{jm}]$ . Czasy trwania robót  $p_{jk}$  są wyznaczone według zasad przedstawionych w p. 5.1.1.
- Zakłada się, podobnie jak w modelu przedstawionym w p. 5.1.1, możliwość występowania przerw technologicznych między robotami i jednoczesnej pracy wielu grup roboczych w obiektach. Czasy trwania przerw między daną robotą i następną ( $s_{jk}^F > 0$ ) lub długość jednoczesnego czasu trwania danej roboty i następnej ( $s_{jk}^F < 0$ ) w obiekcie, dla zbioru robót  $O_j$ , są podane w wektorze  $s_j^F = [s_{j1}^F, s_{j2}^F, s_{j3}^F, \dots, s_{jk}^F, \dots, s_{jm}^F]$ . W modelu czasy te mogą przyjmować wartości dowolne. Podobnie, jak w modelu z p. 5.1.1, w dalszej części pracy czasy te będą nazywane sprzężeniami między obiektami.
- Dodatkowe czasy niezbędne dla przemieszczania grup roboczych między obiektami, zależne od rodzaju grupy roboczej oraz kolejności realizacji obiektów określa macierz  $S_k^S = [s_{gh}^S]_{n \times n}$ , gdzie  $g \in [1..n]$ ,  $h \in [1..n]$ ,  $k \in [1..m]$ .

Ograniczenia:

- Zakłada się kolejność wykonywania robót wynikającą z technologii:

$$O_{j,k-1} < O_{j,k} < O_{j,k+1} .$$

- Zakłada się, że w dowolnej chwili każda grupa robocza z zespołu  $B_k$  może wykonywać tylko jedną robotę.
- Zakłada się, że robota  $O_{jk} \in O_j$  jest realizowana nieprzerwanie przez jedną grupę roboczą z zespołu  $B_k$  przez czas  $p_{jk} > 0$ .

Zmienną decyzyjną jest kolejność  $\pi$  wykonywania poszczególnych robót w obiektach, którą stanowi zestaw permutacji podziału rozłącznego (jak w p. 3.3)  $\pi = (\pi_1, \pi_2, \dots, \pi_k, \dots, \pi_m)$ . Kolejność wykonywania roboty  $k$  w obiektach przez zespół grup roboczych  $B_k$  określa permutacja podziału rozłącznego  $\pi_k = (\pi_{k1}, \pi_{k2}, \dots, \pi_{ki}, \dots, \pi_{km_k})$ , gdzie  $\pi_{ki} = (\pi_{ki}(1), \pi_{ki}(2), \dots, \pi_{ki}(l), \dots, \pi_{ki}(n_{ki}))$  określa kolejność wykonywania roboty  $k$  w obiektach przydzielonych do grupy roboczej  $B_{ki}$  z zespołu grup roboczych  $B_k \subset B$  o liczności  $m_k$ .

Kryterium (funkcją celu) jest termin  $C_{max}$  realizacji wszystkich robót we wszystkich obiektach. Zadanie optymalizacyjne w modelu polega na znalezieniu harmonogramu wykonywania robót, który minimalizuje wartość funkcji celu, spełniając ograniczenia podane powyżej.

Rozpatrywany model można zidentyfikować wśród modeli TSZ jako problem  $m$  – maszynowy o cechach podanych w p. 5.1.1 uwzględniając tylko zmianę typu zagadnienia na problem przepływowy z maszynami równoległymi. Model ten można określić jako

uogólnienie w stosunku do modelu podanego w p. 5.1.1 przez zastosowanie więcej niż jednej grupy roboczej wykonującej roboty jednego rodzaju. Nie był on badany w znanych autorowi pracach, w ramach TSZ.

Rozpatrywany model przedstawia się również w formie grafu dysjunktywnego [100], [108], [134]. Forma tego grafu jest zależna od przyjętego zestawu permutacji podziału rozłącznego  $\pi$ :

$$G(\pi) = (N', E(\pi)).$$

gdzie  $N'$  to zbiór węzłów  $E(\pi)$  – zbiór krawędzi. Przyjęto  $N' = N \cup \{(start), (koniec)\}$ , gdzie  $N = \{1, \dots, k, \dots, m\} \times \{1, \dots, j, \dots, n\}$  jest zbiorem węzłów reprezentujących roboty dla poszczególnych obiektów zaś  $(start)$ ,  $(koniec)$  są węzłami fikcyjnymi reprezentującymi operację początkową i końcową. Obciążenie wierzchołka  $(k, j)$  jest równe  $p_{k, \pi_{ki}(l)}$ , natomiast wagi węzłów  $(start)$ ,  $(koniec)$  są równe zero. Zbiór krawędzi  $E(\pi) = E^F \cup E^S(\pi)$  ma budowę zależną od przyjętej zmiennej decyzyjnej  $\pi$ . Krawędzie poziome (sekwencyjne, reprezentujące kolejność realizacji obiektów i mające charakter dysjunktywny) ze zbioru  $E^S(\pi)$  znajdują się pomiędzy wierzchołkami  $\pi_{ki}(l-1)$  i  $\pi_{ki}(l)$  dla  $i = 1, \dots, m_k$ ,  $k = 1, \dots, m$ ,  $l = 2, \dots, n_{ki}$ . Krawędzie technologiczne ze zbioru  $E^F$  znajdują się pomiędzy wierzchołkami, które oznaczają roboty  $k-1$  i  $k$  dla obiektu  $\pi_{ki}(j)$ . Położenie tych wierzchołków jest zależne od pozycji w permutacjach podziału rozłącznego  $\pi_{k-1}$  i  $\pi_k$ . Krawędzie ze zbioru  $E^S(\pi)$  są obciążone wartościami  $s_{k, \pi_{ki}(l-1)\pi_{ki}(l)}^S$  a krawędzie ze zbioru  $E^F$  – wartościami  $s_{k, \pi_{ki}(j)}^F$ . (rys. 5.3). Terminy zakończenia poszczególnych robót można określić ze wzoru o postaci rekurencyjnej:

$$C_{k, \pi_{ki}(l)} = \max\{C_{k, \pi_{ki}(l-1)} + s_{k, \pi_{ki}(l-1)\pi_{ki}(l)}^S, C_{k-1, \pi_{ki}(l)} + s_{k-1, \pi_{ki}(j)}^F\} + p_{k, \pi_{ki}(l)},$$

gdzie:  $j = 1, \dots, n$ ,  $l = 1, \dots, n_{ki}$ ,  $i = 1, \dots, m_k$ ,  $k = 1, \dots, m$ ,  $\pi_{ki}(0) = 0$ ,  $C_{k,0} = 0$ ,  $C_{0,j} = 0$ .

Terminy zakończenia poszczególnych robót mogą być znalezione w czasie rzędu  $O(nm)$ , podobnie jak dla problemu przepływowego z maszynami równoległymi. Dla kryterium  $C_{max}$  (minimum czasu realizacji wszystkich robót we wszystkich obiektach) poszukiwany jest zestaw permutacji podziału rozłącznego  $\pi^* \in \Pi$ , dla którego:

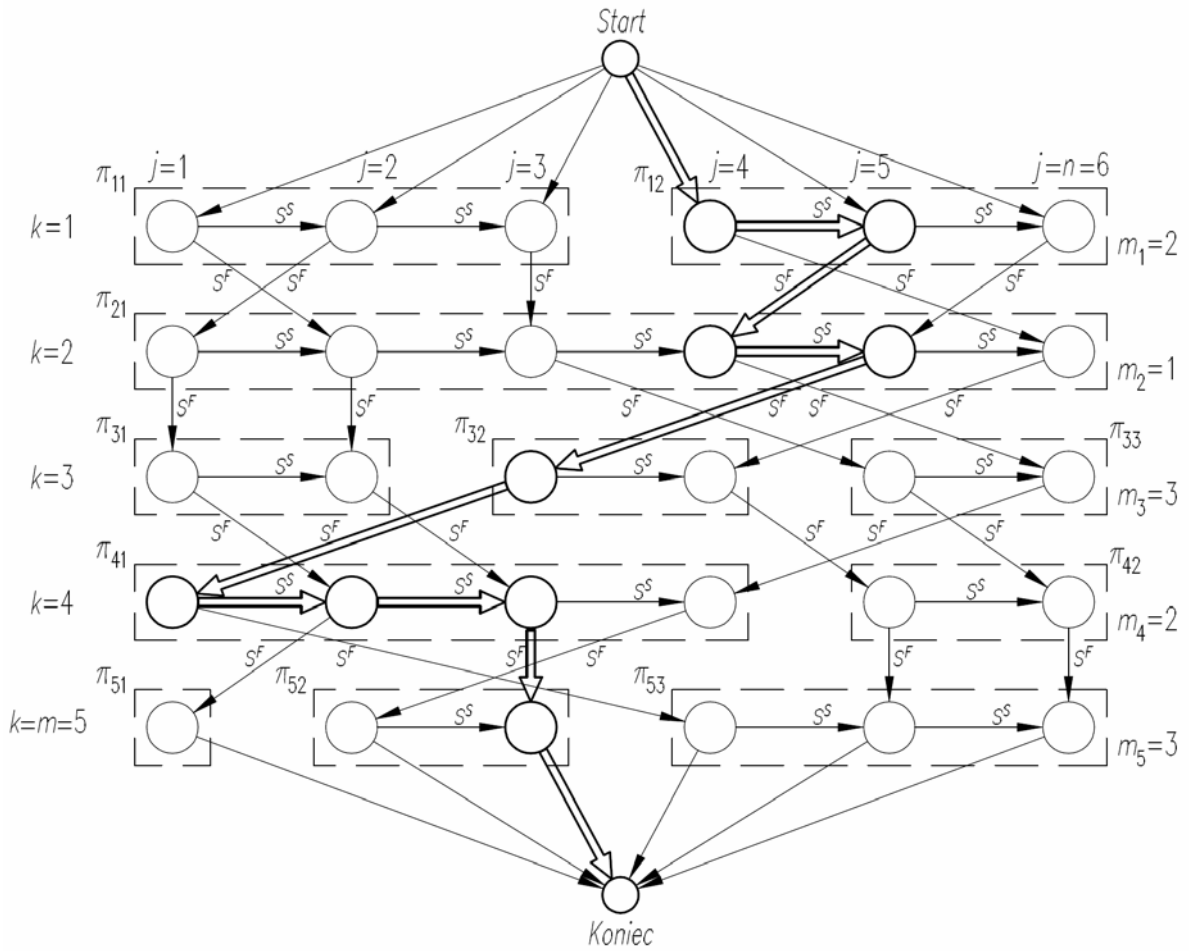
$$C_{max}(\pi^*) = \min_{\pi \in \Pi} C_{max}(\pi), \quad \text{gdzie } C_{max}(\pi) = \max_j C_{m,j}.$$

Dowolny graf  $G(\pi)$  dla przedmiotowego modelu posiada własność ścieżki krytycznej (przykład jej jest przedstawiony rys. 5.3).

Model systemu pracy potokowej z wieloma grupami roboczymi wykonującymi robotę jednego rodzaju, podobnie jak model przedstawiony w p. 5.1.1, jest NP-trudnym zagadnieniem optymalizacyjnym i ze względu na przyjęte ograniczenia nie jest możliwe



utworzenie algorytmu B&B do rozwiązywania zagadnienia optymalizacyjnego (podobnie jak w p. 5.1.1 nie są możliwe do ustalenia funkcje dolnych i górnych ograniczeń).



Rys. 5.3. Przykładowy graf  $G(\pi)$  dla rozważanego w p. 5.1.2 modelu systemu pracy potokowej ( $m = 5, n = 6$ )

Przyjęto rozwiązanie tego zagadnienia w analizowanym modelu za pomocą trzech rozpatrzonych w rozdziale 4 algorytmów metaheurystycznych: TS, GS, SA. Forma tych algorytmów jest taka sama jak dla problemu przepływowego z maszynami równoległymi i została przedstawiona w p. 4.1.4.2.

### 5.1.3. Model systemu pracy potokowej z relacjami kolejnościowymi między robotami wyrażonymi sekwencją i z uwzględnieniem kryteriów koszt/czas

Przedstawione w p. 5.1.1 i 5.1.2 dwa modele harmonogramowania w systemie pracy potokowej uwzględniały kryterium czasu trwania przedsięwzięcia budowlanego. Jednym z najczęściej wymienianych czynników wpływających na planowanie danego przedsięwzięcia jest odpowiednie wykorzystanie zasobów do jego wykonania. W modelach tych zostały

zastosowane zasoby czynne (odnawialne), którymi były grupy robocze. Liczba tych grup dla wykonywania roboty jednego rodzaju była ściśle ograniczona - tylko jedna lub więcej niż jedna. Były one niepodzielne (dana grupa robocza nie mogła wykonywać robót w więcej niż jednym obiekcie) oraz przyjęto, że ich dostępność jest zapewniona w każdej chwili realizacji przedsięwzięcia.

W modelu formułowanym w niniejszym podrozdziale zostanie rozważony, oprócz czasów trwania robót, parametr ekonomiczny zasobów czynnych tj. ich koszt użycia. W badaniach operacyjnych i teorii szeregowania zadań ten rodzaj parametru mówi o nieodzownym do wykonania danej roboty zapotrzebowaniu na zasób nieodnawialny, którym jest kapitał (zasób finansowy). W zależności od wyboru systemu realizacji przedsięwzięcia budowlanego i roli jaką pełnią grupy robocze, koszt ten jest rozumiany następująco. W systemach: generalnego wykonawcy, z menedżerem budowy, pełniącym rolę generalnego wykonawcy, czy w systemie wykonawstwa własnego, gdy istnieją grupy robocze zatrudnione przez wykonawcę na okres dłuższy niż realizacja danego przedsięwzięcia, jest on obliczany według kalkulacji kosztu wykonania danej roboty przez daną grupę roboczą wykonawcy [74]. Składa się on z kosztu bezpośredniego związanego z wykonaniem danej roboty przez grupę roboczą (m.in. kosztu robocizny, kosztu pracy sprzętu, zużytych materiałów) oraz udziału kosztów pośrednich i zysku przedsięwzięcia w przeliczeniu na tę grupę [74]. W tym przypadku wykonawca ma możliwość wyboru grupy do realizacji danej roboty w obiekcie spośród wielu własnych grup roboczych, kierując się określonymi w modelu kryteriami i ograniczeniami kosztu i czasu trwania roboty. W sytuacjach, gdy zostają wykorzystani podwykonawcy robót (sytuacja ta może mieć miejsce we wszystkich stosowanych obecnie systemach [74]), kosztem będzie przychód powstały ze sprzedaży usług przez danego podwykonawcę dla wykonawcy lub inwestora (generalnego wykonawcy, inwestora pełniącego również rolę wykonawcy). Najważniejsze elementy tego przychodu to: koszt bezpośredni, pośredni oraz zysk danego podwykonawcy realizującego daną robotę w danym obiekcie. Przedstawiony sposób definiowania kosztów jest stosowany zarówno w literaturze polskiej np. [74], jak również zagranicznej np. [40], [43]. W tym przypadku wykonawca ma możliwość wyboru grupy do realizacji danej roboty w obiekcie, spośród wielu podwykonawców, którzy przedstawili mu oferty zawierające koszt i czas realizacji danej roboty w obiekcie. Zakłada się, że wykonawca dokonał wcześniej selekcji podwykonawców pod względem innych parametrów np. jakości robót i tylko taki zbiór podwykonawców dopuścił do składania ofert. Dla wszystkich tych przedstawionych sytuacji nie jest ściśle określona postać zależności pomiędzy czasem trwania roboty i jego kosztem. Funkcja ta będzie miała charakter dyskretny. W tego rodzaju problemach najczęściej

przyjmuje się, że czasy realizacji robót są pewnymi wypukłymi, malejącymi funkcjami ilości przydzielonych im zasobów (kosztów) [63]. W rozpatrywanym modelu przyjęto podobnie, że funkcja czas/koszt dla roboty w obiekcie będzie miała również takie cechy, co umożliwi przeprowadzenie optymalizacji [63]. W danej aplikacji mogą jednak pojawiać się wartości czasu i kosztu, gdzie ta zależność nie jest spełniona (wraz ze wzrostem czasu trwania roboty lub dla takiego samego czasu trwania roboty rośnie jego koszt). Przyjęto, że nie dokonuje się eliminacji takich danych przed rozwiązywaniem danego zadania. W związku z postacią przyjętego kryterium, w trakcie przeprowadzanej optymalizacji, odpowiednie (wyższe) wartości czasu i kosztu, nie spełniające przyjętej postaci zależności koszt/czas, będą pomijane w poszukiwaniu rozwiązania suboptymalnego.

W teorii szeregowania zadań przedstawioną powyżej sytuację można zaklasyfikować do statycznych modeli operacji typu czas/zasób dyskretny [63]. Zgodnie z nim ilość zasobu (kapitału) przydzielona jednorazowo na początku wykonywania operacji (roboty) nie może być zmieniana oraz, aby rozpocząć wykonywanie operacji (roboty), należy przydzielić ściśle określoną (nie mniejszą ani większą) ilość zasobu (kapitału). W ogólnym przypadku czasy wykonywania operacji (robót) są dyskretnymi, nieujemnymi i nierosnącymi funkcjami ilości zasobu (kapitału) przedzielonego do wykonywania operacji (roboty) przez daną grupę roboczą.

Formułowany model bazuje na modelu systemu pracy potokowej z wieloma grupami roboczymi wykonującymi robotę jednego rodzaju. Dodatkowymi parametrami, oprócz czasów trwania robót, będą ich koszty. Kryteriami mogą być zarówno czas trwania całego przedsięwzięcia, jak również jego koszt. W związku z tym, że są one niezależne i przeciwstawne, rozpatrywane będą w modelu następujące postacie funkcji celu:

- czas trwania całego przedsięwzięcia przy założonym ograniczeniu jego kosztu,
- koszt całego przedsięwzięcia przy założonym ograniczeniu czasu jego trwania,
- koszt i czas trwania całego przedsięwzięcia.

Przyjęte postacie funkcji celu są analogiczne do funkcji celu, które są rozpatrywane dla zagadnienia TCT w metodach sieciowych, dla przedsięwzięć typu kompleks operacji, opisanego w p. 2.1.

W formułowanym modelu zakłada się, że koszt całego przedsięwzięcia stanowi sumę kosztów wszystkich robót zrealizowanych we wszystkich obiektach przedsięwzięcia. Pierwsze dwie funkcje celu tworzą zagadnienia optymalizacji jednokryterialnej, takiej samej jak w modelach z p. 5.1.1 i p. 5.1.2. Trzecia funkcja celu jest zagadnieniem optymalizacji dwukryterialnej. Rozwiązaniem jego jest jedno z rozwiązań Pareto optymalnych, tworzących zbiór Pareto [63].

Rozważany model stosuje założenia problemu przepływowego z maszynami równoległymi. Podstawą jego są parametry, ograniczenia i postać zmiennej decyzyjnej modelu przedstawionego w p. 5.1.2. W związku z przyjęciem innych funkcji celu, zbiór parametrów przedmiotowego modelu zostaje rozszerzony o dodatkowy parametr - koszty wykonywania robót. Podstawowe założenia formułowanego modelu zostały zasygnalizowane przez autora rozprawy w pracy [111].

Model optymalizacyjny przedmiotowego zagadnienia jest następujący:

Parametry:

- Przedsięwzięcie tworzy zbiór obiektów budowlanych  $Z = \{Z_1, Z_2, Z_3, \dots, Z_j, \dots, Z_n\}$ .
- Do wykonywania prac istnieją zespoły grup roboczych wykonujących robotę jednego rodzaju i tworzących zbiór  $B = \{B_1, B_2, B_3, \dots, B_k, \dots, B_m\}$ .
- W każdym zespole grup roboczych  $B_k \in B$  znajduje się  $m_k \geq 1$  grup roboczych, które mogą posiadać takie same albo różne wydajności lub składy:  $B_k = \{B_{k1}, B_{k2}, B_{k3}, \dots, B_{ki}, \dots, B_{km_k}\}$ .
- Każdy obiekt  $Z_j \in Z$  wymaga realizacji  $m$  robót, które tworzą zbiór  $O_j = \{O_{j1}, O_{j2}, O_{j3}, \dots, O_{jk}, \dots, O_{jm}\}$ .
- Zakłada się, że robota  $O_{jk} \in O_j$  może być realizowana przez grupę roboczą  $B_{ki} \subset B_k$ . Czas trwania roboty  $O_{jk}$  wykonywanej przez nią wynosi  $p_{jki} > 0$ . Zbiór możliwych czasów trwania  $p_j$  robót ze zbioru  $O_j$  określa wektor  $p_j = [p_{j1}, p_{j2}, p_{j3}, \dots, p_{jk}, \dots, p_{jm}]$ , gdzie  $p_{jk} = [p_{jk1}, p_{jk2}, p_{jk3}, \dots, p_{jki}, \dots, p_{jmm_k}]$ . Czasy trwania robót  $p_{jki}$  są wyznaczone jak to przedstawiono w p. 5.1.1.
- Podobnie, jak wyżej, zakłada się, że robota  $O_{jk} \in O_j$  może być realizowana przez grupę roboczą  $B_{ki} \subset B_k$ . Koszt realizacji roboty  $O_{jk}$  przez grupę roboczą  $B_{ki}$  określa zmienna  $u_{jki} \geq 0$ . Zbiór możliwych kosztów  $u_j$  robót ze zbioru  $O_j$  określa wektor  $u_j = [u_{j1}, u_{j2}, u_{j3}, \dots, u_{jk}, \dots, u_{jm}]$ , gdzie  $u_{jk} = [u_{jk1}, u_{jk2}, u_{jk3}, \dots, u_{jki}, \dots, u_{jmm_k}]$ . Koszt roboty  $u_{jki}$  jest wyznaczany według kalkulacji kosztu wykonania roboty  $O_{jk}$  przez grupę roboczą  $B_{ki}$  znajdującą się w zasobach wykonawcy. Może on również być ofertą kosztu wykonania roboty  $O_{jk}$  przez podwykonawcę reprezentowanego przez grupę roboczą  $B_{ki}$ .
- Zakłada się, podobnie jak w modelu przedstawionym w p. 5.1.1, możliwość występowania przerw technologicznych między robotami i jednoczesnej pracy wielu grup roboczych w obiektach (sprzężenia między obiektami)  $s^F_j = [s^F_{j1}, s^F_{j2}, s^F_{j3}, \dots, s^F_{jk}, \dots, s^F_{jm}]$ . W przedmiotowych dwóch modelach przyjęto możliwość dwójakiej interpretacji wartości  $s^F_{jk}$ . Pierwsza interpretacja jest taka sama jak w modelach przedstawionych w p. 5.1.1 i w p. 5.1.2. Wartości  $s^F_{jk}$  przedstawiają wtedy konkretne odległości czasowe między

terminem zakończenia roboty poprzedniego rodzaju  $O_{jk}$  i terminem rozpoczęcia roboty  $O_{j,k+1}$  następnego rodzaju w tym samym obiekcie. Implementacja programowa w środowisku Mathematica dla tej interpretacji polega na przyjęciu przez zmienną reprezentującą wartość  $s_{jk}^F$  typu Integer (liczba całkowita). W drugiej interpretacji wartości  $s_{jk}^F$  przedstawiają mnożnik dla czasu trwania  $p_{jki}$  roboty poprzedniego rodzaju  $O_{jk}$ . W wyniku tego działania matematycznego otrzymywana jest konkretna wartość  $s_{jk}^F$ . Przyczyną wprowadzenia takiej interpretacji jest umożliwienie podania wartości  $s_{jk}^F$  jako wielkości proporcjonalnej do czasu trwania  $p_{jki}$  roboty poprzedniego rodzaju  $O_{jk}$ , który może zmienny i zależny od wyboru grupy roboczej wykonującej robotę  $O_{jk}$ . Implementacja programowa w środowisku Mathematica dla tej interpretacji polega na przyjęciu przez zmienną reprezentującą wartość  $s_{jk}^F$  typu Real (liczba rzeczywista), co pozwoli na odróżnienie tych dwóch postaci sprzężeń między obiektami. W formułowanym modelu ich wartości mogą być dowolne.

- Dodatkowe czasy niezbędne dla przemieszczania grup roboczych między obiektami, zależne od rodzaju grupy roboczej oraz kolejności realizacji obiektów, określa macierz  $S_k^S = [s_{gh}^S]_{n \times n}$ , gdzie  $g \in [1..n]$ ,  $h \in [1..n]$ ,  $k \in [1..m]$ .

#### Ograniczenia:

- Zakłada się kolejność wykonywania robót wynikającą z technologii:

$$O_{j,k-1} < O_{j,k} < O_{j,k+1} .$$

- Zakłada się, że w dowolnej chwili każda grupa robocza z zespołu  $B_k$  może wykonywać tylko jedną robotę.
- Zakłada się, że robota  $O_{jk} \in O_j$  jest realizowana nieprzerwanie przez grupę roboczą  $B_{ki} \subset B_k$  przez czas  $p_{jki} > 0$ .
- Zakłada się (alternatywnie) ograniczenie kosztu realizacji całego przedsięwzięcia  $U \leq \hat{U}$ ,

$$U = \sum_{j=1}^n \sum_{k=1}^m u_{jk} \text{ lub ograniczenie czasu zakończenia wszystkich robót w obiektach}$$

$$C_{max} \leq \hat{C} , \text{ gdzie } C_{max} \text{ jest czasem trwania przedsięwzięcia.}$$

Zmienną decyzyjną jest kolejność  $\pi$  wykonywania obiektów, którą stanowi zestaw permutacji podziału rozłącznego (jak w p. 3.3)  $\pi = (\pi_1, \pi_2, \dots, \pi_k, \dots, \pi_m)$ . Kolejność wykonywania obiektów przez zespół grup roboczych  $B_k$  określa permutacja podziału rozłącznego  $\pi_k = (\pi_{k1}, \pi_{k2}, \dots, \pi_{ki}, \dots, \pi_{km_k})$ , gdzie  $\pi_{ki} = (\pi_{ki}(1), \pi_{ki}(2), \dots, \pi_{ki}(l), \dots, \pi_{ki}(n_{ki}))$  określa kolejność wykonywania obiektów przydzielonych do grupy roboczej  $B_{ki} \subset B_k$ ,  $B_k \subset B$ . Postać zmiennej decyzyjnej jednoznacznie określa alokację grup roboczych do realizacji robót w obiektach. W związku z tym, za pomocą zestawu permutacji podziału rozłącznego  $\pi$ , jednoznacznie są

ustalone czasy trwania poszczególnych robót wykonywanych w obiektach oraz ich koszt. Po przyjęciu permutacji  $\pi$  zbiór czasów trwania  $p_j$  robót ze zbioru  $O_j$  jest następujący:

$$p_j = \{p_{j1}, p_{j2}, p_{j3}, \dots, p_{jk}, \dots, p_{jm}\},$$

gdzie  $p_{jk}$  jest czasem trwania wykonywania roboty  $k$  w obiekcie  $j$ .

Podobnie po przyjęciu permutacji  $\pi$  zbiór kosztów  $u_j$  robót ze zbioru  $O_j$  jest następujący:

$$u_j = \{u_{j1}, u_{j2}, u_{j3}, \dots, u_{jk}, \dots, u_{jm}\},$$

gdzie  $u_{jk}$  jest kosztem realizacji roboty  $k$  w obiekcie  $j$ .

W pierwszym przypadku funkcją celu będzie czas trwania całego przedsięwzięcia przy założonym ograniczeniu jego kosztu. Dokładniej, poszukiwany będzie minimalny termin zakończenia wykonywania wszystkich robót w obiektach  $C_{max}(\pi^*)$  dla zestawu permutacji podziału rozłącznego  $\pi^* \in \Pi$ , dla którego:

$$C_{max}(\pi^*) = \min_{\pi \in \Pi} C_{max}(\pi), \quad \text{gdzie } C_{max}(\pi) = \max_j C_{m,j},$$

przy założeniu, że koszt całego przedsięwzięcia nie przekroczy wartości dyrektywnej narzuconej przez wykonawcę:

$$U(\pi^*) \leq \hat{U}, \quad U(\pi) = \sum_{j=1}^n \sum_{k=1}^m u_{jk},$$

gdzie  $n$  – liczba wszystkich obiektów roboczych,  $m$  – liczba robót wymaganych realizacji w każdym z obiektów.

W drugim przypadku funkcją celu będzie koszt przedsięwzięcia przy założonym ograniczeniu jego terminu zakończenia. Dokładniej poszukiwany będzie minimalny koszt całego przedsięwzięcia (będący sumą kosztów poszczególnych robót) dla zestawu permutacji podziału rozłącznego  $\pi^* \in \Pi$ , dla którego:

$$U(\pi^*) = \min_{\pi \in \Pi} U(\pi), \quad \text{gdzie } U(\pi) = \sum_{j=1}^n \sum_{k=1}^m u_{jk},$$

gdzie  $n$  – liczba wszystkich obiektów roboczych,  $m$  – liczba robót wymaganych realizacji w każdym z obiektów, przy założeniu, że termin zakończenia całego przedsięwzięcia  $C_{max}(\pi^*)$  nie przekroczy terminu dyrektywnego narzuconego przez inwestora  $\hat{C}$ :  $C_{max}(\pi^*) \leq \hat{C}$ .

Terminy zakończenia wykonywania poszczególnych robót w obu modelach mogą być znalezione w czasie rzędu  $O(nm)$ , podobnie jak dla problemu przepływowego z maszynami równoległymi.

W modelach przedstawionych w p. 5.1.1 i p. 5.1.2 każde rozwiązanie należy do zbioru rozwiązań dopuszczalnych, czyli spełniających wszystkie warunki ograniczające. W modelach przedstawionych w niniejszym podrozdziale mogą istnieć zbiory rozwiązań niedopuszczalnych. Będą to te rozwiązania, które tworzą harmonogramy nie spełniające

ograniczeń kosztu całego przedsięwzięcia lub terminu jego zakończenia nałożonych w danym modelu. Jest to istotna różnica między modelami z kryteriami koszt/czas a modelami z kryterium czasu trwania przedsięwzięcia, która znajduje odzwierciedlenie w dokonanej implementacji programowej.

Rozpatrywany model można zidentyfikować wśród modeli TSZ jako problem  $m$  – maszynowy o cechach podanych w p. 5.1.1, z typem zagadnienia przepływowym, z maszynami równoległymi. Formułowany model przedstawia się również w formie grafu dysjunktywnego [100], [108], [134]. Forma tego grafu jest zależna od przyjętego zestawu permutacji podziału rozłącznego  $\pi$  i jest taka sama jak podano w p. 5.1.2.

Podobnie jak modele przedstawione w p. 5.1.1 i p. 5.1.2, przedmiotowy model jest NP-trudnym zagadnieniem optymalizacyjnym. Przyjęto rozwiązanie zagadnienia optymalizacyjnego za pomocą trzech rozpatrzonych w rozdziale 4 algorytmów metaheurystycznych: TS, GS, SA. Forma tych algorytmów jest taka sama jak dla problemu przepływowego z maszynami równoległymi i została przedstawiona w p. 4.1.4.2.

Jak podano wyżej, w modelu mogą wystąpić zbiory rozwiązań niedopuszczalnych ze względu na przyjęte ograniczenia. Założono eliminację tego rodzaju rozwiązań przez przyjęcie dla nich odpowiednio dużej wartości funkcji celu. Będzie to powodować skierowanie trajektorii poszukiwań we wszystkich trzech algorytmach do obszarów rozwiązań dopuszczalnych, gdzie jest możliwość znalezienia rozwiązania optymalnego. Ważnym problemem w rozwiązywaniu zadań optymalizacyjnych dla sformułowanego modelu jest odpowiednie przyjęcie rozwiązań początkowych w algorytmach. W poprzednio rozważanych modelach, gdzie każde możliwe rozwiązanie było dopuszczalne, przyjęto jako rozwiązanie początkowe permutację  $\pi$ , która była tworzona losowo. W rozpatrywanym modelu takie przyjęcie rozwiązania startowego może powodować, że będzie ono rozwiązaniem niedopuszczalnym. Przyjęte bardziej restrykcyjne ograniczenia mogą powodować znaczne zmniejszenie liczności zbioru rozwiązań dopuszczalnych. W związku z tym, algorytmy mogą przebywać przez dłuższy lub nawet cały czas obliczeń w obszarach rozwiązań niedopuszczalnych. Aby uniknąć takich sytuacji, przyjęto podany dalej sposób wyznaczania rozwiązań startowych ze zbioru rozwiązań dopuszczalnych.

W modelu z kryterium czasu trwania całego przedsięwzięcia, przy założonym ograniczeniu jego kosztu, rozwiązania startowe w algorytmach TS, GS, SA są tworzone w sposób losowy z założeniem, że poszczególne roboty w obiektach są wykonywane przez grupy robocze, których koszt jest najmniejszy spośród danego zespołu grup roboczych. W związku z tym, takie przyjęcie rozwiązań startowych gwarantuje spełnienie każdego ograniczenia kosztu całego przedsięwzięcia. Po wybraniu takiego rozwiązania trajektoria

poszukiwań w algorytmach przebiega w kierunku rozwiązań spełniających nałożone ograniczenie i minimalizacji przyjętego kryterium.

W modelu z kryterium kosztu przedsięwzięcia przy założonym ograniczeniu jego terminu zakończenia dobranie rozwiązań startowych w algorytmach TS, GS, SA jest trudniejsze ze względu założoną postać ograniczenia. Może bowiem wystąpić sytuacja, w której przyjęte ograniczenie związane z terminem zakończenia przedsięwzięcia powoduje silne zmniejszenie liczości zbioru rozwiązań dopuszczalnych i które będzie można wyznaczyć tylko przez poszukiwanie rozwiązania zagadnienia z kryterium czasu trwania całego przedsięwzięcia, przy założonym ograniczeniu jego kosztu. W związku z tym założono, że jako rozwiązania startowe w algorytmach TS, GS, SA przyjmuje się dowolne rozwiązania suboptymalne uzyskane z optymalizacji przeprowadzonej w modelu pierwszym i spełniające warunek ograniczenia czasu trwania przedsięwzięcia. Rozwiązywanie zadań optymalizacyjnych w modelu drugim będzie dwuetapowe. W pierwszym etapie rozwiązywane będzie zadanie optymalizacyjne przy kryterium czasu trwania całego przedsięwzięcia, przy założonym ograniczeniu jego kosztu oraz przyjmując, że poszczególne roboty w obiektach są wykonywane przez grupy robocze, których koszt jest najwyższy w danym zespole grup roboczych (zapewni to możliwie najkrótszy czasu trwania przedsięwzięcia). W drugim etapie uzyskane rozwiązanie jest traktowane jako rozwiązanie startowe zadania optymalizacyjnego z założonym modelem, przy drugim kryterium. Może wystąpić jednak sytuacja, że dane rozwiązanie startowe nie spełni ograniczenia nałożonego w tym modelu (czas trwania przedsięwzięcia będzie dłuższy niż przyjęte ograniczenie czasu). W tym przypadku przyjmuje się, że dane zadanie optymalizacyjne nie posiada zbioru rozwiązań dopuszczalnych i nie jest możliwe wyznaczenie rozwiązania optymalnego. Wykonawca uzyskuje informację, że przy tak dobranych parametrach nie jest możliwe uzyskanie harmonogramu spełniającego termin dyrektywny podany przez inwestora. Wymagana jest wtedy zmiana parametrów zadania (zwiększenie liczby grup roboczych w zespołach, wymiana grup roboczych na wydajniejsze lub zmiana ograniczenia - terminu dyrektywnego, w porozumieniu z inwestorem.

W sformułowanym modelu jest możliwe rozpatrzenie podejścia dwukryterialnego. Polega ono na jednoczesnej minimalizacji obu niezależnych kryteriów: całkowitego czasu i kosztu przedsięwzięcia wieloobiektowego. Ze względu na przeciwstawność obu kryteriów rozwiązaniem tego zagadnienia jest zbiór rozwiązań Pareto optymalnych (niezdominowanych, efektywnych). Ogólna definicja elementu ze zbioru rozwiązań Pareto optymalnych jest następująca [63]. Jedno z rozwiązań Pareto optymalnych  $s$ , będące rozwiązaniem dopuszczalnym, musi spełniać warunek, że dla przeciwstawnych kryteriów  $F_1$



i  $F_2$  nie istnieje żadne inne rozwiązanie  $q$  takie, że spełnione są następujące nierówności:  $F_1(q) \leq F_1(s)$  oraz  $F_2(q) \leq F_2(s)$  i przynajmniej jedna z tych nierówności jest ostra. Zbiór  $P$  będzie nazywany zbiorem Pareto, jeśli jest on zbiorem rozwiązań Pareto optymalnych takich, że nie zawiera on dwóch rozwiązań  $s, q \in P$  z wartościami  $F_1(q) = F_1(s)$  oraz  $F_2(q) = F_2(s)$ . Punkt  $(F_1(s), F_2(s))$  będzie nazywany punktem kompromisowym w przestrzeni funkcji kryterialnych  $F_1$  i  $F_2$ , jeśli  $s \in P$ . Zbiór wszystkich punktów kompromisowych pomiędzy kryteriami  $F_1$  oraz  $F_2$  będzie oznaczany przez  $K$  [63]. W przypadku rozważanego modelu

kryteriami  $F_1$  oraz  $F_2$  są całkowity koszt realizacji przedsięwzięcia  $U(\pi) = \sum_{j=1}^n \sum_{k=1}^m u_{jk}$  oraz

całkowity czas realizacji przedsięwzięcia  $C_{\max}(\pi) = \max_j C_{m, j}$ . Zbiór rozwiązań Pareto

optymalnych  $s$  będzie oznaczony dla rozpatrywanego modelu (podobnie jak w przypadku ogólnym) przez  $P$ , natomiast zbiór wszystkich punktów kompromisowych pomiędzy przedmiotowymi kryteriami przez  $K$ . W modelu przyjmuje się, że czasy trwania robót w obiektach są wypukłymi, malejącymi funkcjami dyskretnymi ilości przydzielonych im zasobów (kosztów). W związku z tym zbiór wszystkich punktów kompromisowych  $K$  będzie zbliżony kształtem do krzywej (łamanej), która powstałaby przy założeniu ciągłej funkcji czas/zasób dla roboty w obiekcie [63]. Krzywa ta jest wypukła, malejąca odcinkami [63].

Autor rozprawy nie spotkał się w literaturze z algorytmem pozwalającym na wyznaczenie punktów kompromisowych ze zbioru  $K$  w sformułowanym modelu. W pracy [63] podano algorytm, za pomocą którego można wyznaczyć punkty kompromisowe w modelu przy założeniu ciągłej funkcji czas/zasób dla roboty w obiekcie. W związku z tym, proponuje się wyznaczenie części zbioru punktów kompromisowych  $K$  odpowiadającego wartościom progowym kosztu całego przedsięwzięcia za pomocą następującego algorytmu:

**Krok 1.** Wyznacz minimalny  $U_{\min}$  i maksymalny  $U_{\max}$  możliwy koszt całego przedsięwzięcia.

**Krok 2.** Ustal zbiór wartości progowych całkowitego kosztu przedsięwzięcia, które będą znajdowały się pomiędzy wartościami  $U_{\min}$  i  $U_{\max}$ .

**Krok 3.** Dla wartości kosztu  $U_{\min}$  i  $U_{\max}$  oraz ustalonych wartości progowych kosztu znajdź minimalne czasy trwania całego przedsięwzięcia odpowiadające tym wartościom kosztu, korzystając z modelu z kryterium czasu trwania całego przedsięwzięcia przy założonym ograniczeniu jego kosztu.

Przez wartości progowe kosztu całkowitego rozumie się takie wartości kosztu, które ustalił indywidualnie wykonawca na podstawie wartości planowanego budżetu przedsięwzięcia.

Istnieje więc dowolność w ustalaniu liczby punktów kompromisowych, zależna tylko od wykonawcy. W przypadku skrajnym, przy wystarczająco dużym zagęszczeniu wartości progowych kosztu możemy otrzymać wszystkie lub prawie wszystkie punkty kompromisowe ze zbioru  $K$ . Przedstawiony algorytm wyznaczania punktów kompromisowych może mieć wartość użyteczną dla wykonawcy, który otrzyma w ten sposób informację o postaci zależności koszt/czas dla całego przedsięwzięcia oraz podejmie odpowiednie działania dotyczące jego planowania.

## **5.2. Nowe modele harmonogramowania przedsięwzięć wieloobiektyowych realizowanych w systemie pracy potokowej z relacjami kolejnościowymi między robotami danymi grafem**

W niniejszym podrozdziale przedstawione zostaną modele pracy potokowej przedsięwzięć wieloobiektyowych, w których pomiędzy robotami danego obiektu budowlanego, wchodzącego w jego skład, zachodzą relacje kolejnościowe dane niezmiennym, dla każdego z obiektów, grafem. Rozpatrując problemy organizacji robót tylko dla pojedynczych obiektów wchodzących w skład przedsięwzięcia wieloobiektyowego, w takim przypadku właściwymi narzędziami dla ich harmonogramowania są metody dla przedsięwzięć typu kompleks operacji [68]. Badania operacyjne dostarczają dla tego rodzaju przedsięwzięć wielu modeli i metod do ich optymalizacji (np. metodę CPM), gdyż przedsięwzięcia budowlane, ograniczając się tylko do problemów harmonogramowania, są najczęściej traktowane jako projekty. Dla przypadku rozpatrywanego w niniejszym podrozdziale przedmiotowe przedsięwzięcia wieloobiektywne (w sensie harmonogramowania) w badaniach operacyjnych są nazywane programami. Programy tworzą złożone struktury składające się ze zbioru projektów [21]. W sensie badań operacyjnych przedsięwzięcia wieloobiektywne są programami wieloprojektowymi (*multi-project*), których zarządzanie opiera się na wielu równolegle realizowanych projektach, pomiędzy którymi nie ma bezpośrednich, narzuconych relacji kolejnościowych (ograniczeń w kolejności ich realizacji), a wzajemne uzależnienie projektów wynika z wykorzystywania wspólnych zasobów [109]. Według wiedzy autora rozprawy badania operacyjne (w tym również badania związane w teorią szeregowania zadań) nie dostarczają modeli i metod, które umożliwiłyby optymalizację programów wieloprojektowych uwzględniających system pracy potokowej, z występującymi w nich dodatkowymi relacjami kolejnościowymi odnośnie projektów. Brak modeli i metod optymalizacji wynika przypuszczalnie ze zbyt długiego czasu trwania takich programów wieloprojektowych, co w połączeniu z charakterem systemu pracy potokowej

(tworzone są harmonogramy przed przystąpieniem do realizacji przedsięwzięcia, bez możliwości dalszych zmian) powodowałyby, że utworzony harmonogram byłby zbyt sztywny i mało użyteczny dla realizatora. Zagadnienia takie mogą się jednak pojawiać dla przedsięwzięć budowlanych, które polegają na realizacji wielu obiektów (przedsięwzięć wieloobektowych). Z punktu widzenia teorii szeregowania zadań założono, że formułowane modele należą do grupy problemów przepływowych, ze względu na możliwość rozwiązywania zadań optymalizacyjnych metodami stosowanymi dla tych problemów.

Cechą charakterystyczną wyróżniającą tę grupę modeli od grupy przedstawionej w p. 5.1 jest występowanie relacji kolejnościowych między robotami dla danego obiektu, danych niezmiennym dla każdego obiektu grafem. Relacje te odwzorowują logiczną kolejność wykonywania robót dla danego obiektu. Dla jednoznacznego określenia relacji konieczne jest podanie poprzednika (roboty, która poprzez relację warunkuje rozpoczęcie lub zakończenie innej roboty) oraz następnika (roboty, której możliwości wykonywania są podane poprzez relację). W pracy [77] wymienione są następujące rodzaje relacji:

- zakończenie – rozpoczęcie (ZR) – następnik może rozpocząć się nie wcześniej niż po zakończeniu poprzednika; jest to najczęściej stosowany typ relacji,
- zakończenie – zakończenie (ZZ) – następnik może zakończyć się nie wcześniej niż po zakończeniu poprzednika ,
- rozpoczęcie – rozpoczęcie (RR) - następnik może rozpocząć się nie wcześniej niż po rozpoczęciu poprzednika,
- rozpoczęcie – zakończenie (RZ) - następnik może zakończyć się nie wcześniej niż po rozpoczęciu poprzednika.

Dodatkowo dla każdej relacji mogą wystąpić opóźnienia czasowe (*lag*) [77]. Wartości tych opóźnień mogą mieć wartość dodatnią, co wywołuje opóźnienie w czasie dostępności następnika lub wartość ujemną powodując wcześniejszą jego dostępność. Wielkość opóźnień rozumiana jest jako wartość graniczna: minimalna lub maksymalna. Wielkość minimalna oznacza, że następnik może rozpocząć lub zakończyć się po upływie opóźnienia od zakończenia lub rozpoczęcia poprzednika. Wielkość maksymalna oznacza, że następnik musi się rozpocząć lub zakończyć się nie później niż po upływie opóźnienia od rozpoczęcia lub zakończenia poprzednika. W modelach rozpatrywanych w niniejszym podrozdziale zostaną zastosowane wyżej opisane relacje. Nie uwzględniono w modelach relacji rozpoczęcie – zakończenie (RZ) ze względu na jej rzadkie występowanie w problemach organizacji przedsięwzięć budowlanych. W przedmiotowych modelach przyjęto ponadto występowanie opóźnień jako wartości minimalnych. Ten sposób rozumienia opóźnień jest stosowany

w metodzie ścieżki krytycznej (CPM), która jest obecnie najczęściej stosowana w harmonogramowaniu przedsięwzięć budowlanych.

W formułowanych modelach zostanie zastosowana reprezentacja problemu harmonogramowania w formie sieci z czynnościami (robotami) w węzłach. Jest to wtedy tzw. sieć czynności (AON – *Activity-On-Node*) w odróżnieniu od sytuacji, gdy dla reprezentacji problemu stosowana jest sieć zdarzeń z robotami występującymi na łukach (AOA – *Activity-On-Arc*) [68], [137]. Sieć czynności AON tworzy acykliczny graf skierowany  $K = (M, F)$ , w którym  $M$  jest zbiorem obciążonych czasami trwania realizacji roboty wierzchołków reprezentujących roboty obiektów wchodzących w skład przedsięwzięcia, natomiast  $F$  jest zbiorem obciążonych łuków wyrażających relacje kolejnościowe pomiędzy robotami. W rozpatrywanych modelach przyjęto, że sieci czynności dla wszystkich obiektów w przedsięwzięciu posiadają takie same relacje poprzedzania i następowania oraz występują tego samego rodzaju roboty dla każdego z obiektów. Ponadto, dla każdego z nich istnieje dowolność w definiowaniu konkretnego rodzaju relacji (ZR, ZZ, RR) między robotami oraz wartości występującego opóźnienia. W związku z tym, sieć czynności AON dla całego przedsięwzięcia wieloobiektowego będzie składała się z sieci reprezentujących realizację pojedynczych obiektów (następniki i poprzedniki dla każdego z nich są takie same). Między robotami tego samego rodzaju zachodzą dodatkowe relacje poprzedzania i następowania wynikające ze struktury zasobów użytych do jej realizacji (liczby grup roboczych). Każda sieć czynności przedsięwzięcia będzie miała dołączone dwa dodatkowe wierzchołki odpowiadające jego rozpoczęciu i zakończeniu (źródło i odpływ). Zdecydowano się na ten sposób przedstawiania reprezentacji sieciowej ze względu na jego popularność w literaturze zagranicznej i w obecnie stosowanym oprogramowaniu np. Microsoft Project, Primavera Project Planner, Planista oraz łatwość stosowania relacji w sieciach czynności (w sieciach zdarzeń konieczne byłoby zastosowanie obciążonych łuków pozornych, zamiast relacji).

W niniejszym podrozdziale będą przedstawione modele pracy potokowej przedsięwzięć wieloobiektowych z kryterium czasu realizacji przedsięwzięcia oraz kryteriami związanymi z jego kosztem. Przyjęcie wyżej opisanego ograniczenia, związanego z występowaniem relacji kolejnościowych między robotami dla danego obiektu danych grafem, jest rozszerzeniem i uogólnieniem ograniczenia odnośnie do relacji kolejnościowych w porównaniu do modeli z relacjami wyrażonymi sekwencją. Możliwości zastosowania modeli przedstawionych w tym punkcie dla przedsięwzięć wieloobiektowych są podobne jak modeli opisywanych w punkcie 5.1. Istotnie rozszerzają się one na te rodzaje przedsięwzięć, w skład których wchodzi obiektów budowlanych wymagających bardziej skomplikowanych relacji kolejnościowych między robotami. Takimi obiektami budowlanymi mogą być np. zespoły

(kompleksy) budynków o podobnej lub różnej technologii, budynki wraz z równolegle realizowaną infrastrukturą, obiekty liniowe (np. drogi, autostrady) z infrastrukturą itd.

W p. 5.2.1 przedstawiony zostanie model z kryterium czasu trwania przedsięwzięcia wieloobiektowego, w którym wykonawca może dysponować tylko jedną grupą roboczą do wykonywania robót jednego rodzaju. W p. 5.2.2 zostanie przedstawiony model z tym samym kryterium, lecz z założeniem, że grup roboczych do wykonywania robót jednego rodzaju może być więcej niż jedna. Te dwa modele zostały zasygnalizowane przez autora rozprawy w pracy [112]. Kryteria koszt/czas wykorzystujące zagadnienie TCT zostaną zastosowane w modelu przedstawionym w p. 5.2.3, który będzie bazował na założeniach problemu przepływowego z maszynami równoległymi, podobnie jak w modelu z p. 5.2.2.

### **5.2.1. Model systemu pracy potokowej z jedną grupą roboczą wykonującą robotę jednego rodzaju**

W rozpatrywanym modelu są zastosowane niektóre założenia permutacyjnego problemu przepływowego z kryterium czasu wykonywania wszystkich zadań (problem  $FP \parallel C_{\max}$ ), który został przedstawiony w p. 3.2 rozprawy. Podstawowym założeniem przyjętym w modelu, związanym ze strukturą zasobów, jest zastosowanie tylko jednej grupy roboczej dla roboty jednego rodzaju. Model optymalizacyjny przedmiotowego zagadnienia jest następujący:

#### Parametry:

- Przedsięwzięcie tworzy zbiór obiektów budowlanych  $Z = \{Z_1, Z_2, Z_3, \dots, Z_j, \dots, Z_n\}$ .
- Roboty w obiektach wykonują grupy robocze wyspecjalizowane do wykonywania roboty jednego rodzaju. Tworzą one zbiór  $B = \{B_1, B_2, B_3, \dots, B_k, \dots, B_m\}$ , gdzie  $B_k$  reprezentuje jedną grupę roboczą.
- Każdy obiekt  $Z_j \in Z$  wymaga realizacji  $m$  robót, które tworzą zbiór  $O_j = \{O_{j1}, O_{j2}, O_{j3}, \dots, O_{jk}, \dots, O_{jm}\}$ .
- Zakłada się, że robota  $O_{jk} \in O_j$  może być realizowana przez grupę roboczą  $B_k$ . Czas trwania roboty  $O_{jk}$  wykonywanej przez nią wynosi  $p_{jk} > 0$ . Zbiór czasów trwania  $p_j$  robót ze zbioru  $O_j$  określa wektor  $p_j = [p_{j1}, p_{j2}, p_{j3}, \dots, p_{jk}, \dots, p_{jm}]$ .
- Zależności kolejnościowe między robotami  $O_j$  dla obiektu  $Z_j$  są zadane acyklicznym grafem skierowanym  $K = (M, F)$ , gdzie  $M$  jest zbiorem wierzchołków obciążonych czasami trwania realizacji robót  $p_{jk}$  i reprezentujących roboty  $O_j$  obiektu  $Z_j$  ( $|M| = m$ ), natomiast  $F$  jest zbiorem obciążonych łuków wyrażających ograniczenia technologiczne (przyjęte relacje kolejnościowe z wartościami opóźnień). Obciążenia łuków są podane

w postaci relacji pomiędzy robotami RR, ZZ, ZR, z przyjętymi dla nich wartościami opóźnień, które w rozważanym modelu mogą przyjmować wartości dowolne. Wyrażone są za pomocą  $n$  macierzy górnych, trójkątnych  $RR_j, ZR_j, ZZ_j$ . o rozmiarach  $m \times m$ .

- Dodatkowe czasy niezbędne dla przemieszczania grup roboczych między obiektami, zależne od rodzaju grupy roboczej oraz kolejności realizacji obiektów, określa macierz  $S_k^S = [s_{gh}^S]_{n \times n}$ , gdzie  $g \in [1..n], h \in [1..n], k \in [1..m]$ .

#### Ograniczenia:

- Zakłada się, że w dowolnej chwili każda grupa robocza  $B_k$  może wykonywać tylko jedną robotę.
- Zakłada się, że robota  $O_{jk} \in O_j$  jest realizowana nieprzerwanie przez grupę roboczą  $B_k$  przez czas  $p_{jk} > 0$ .

Zmienną decyzyjną jest kolejność  $\pi$  wykonywania obiektów (permutacja), która dla każdej z grup roboczych jest taka sama i jest następująca:

$$\pi = (\pi(1), \pi(2), \dots, \pi(k), \dots, \pi(n)).$$

Liczność zbioru  $\Pi$  wszystkich permutacji  $\pi \in \Pi$  w tym modelu wynosi  $n!$ .

Kryterium (funkcją celu) jest termin  $C_{max}$  realizacji wszystkich robót na wszystkich obiektach. Zadanie optymalizacyjne w modelu polega na znalezieniu harmonogramu wykonywania robót, który minimalizuje wartość funkcji celu, spełniając ograniczenia podane powyżej.

Rozważany model można zidentyfikować wśród zagadnień teorii szeregowania zadań jako problem  $m$ -maszynowy, przepływowy z nałożonymi ograniczeniami: relacjami kolejnościowymi zadanymi grafem  $K$  i przebrojeniami sekwencyjnie zależnymi oraz funkcją celu  $C_{max}$ . Model o opisanej postaci w ramach TSZ nie był badany w znanych autorowi pracach. Dla przedmiotowego modelu można jednak stosować metody optymalizacyjne, które są używane w problemach przepływowych i w modelach z p. 5.1, ze względu na przyjęcie dla każdego obiektu takich samych relacji kolejnościowych (poprzedników i następników) zadanym grafem  $K$  i różniących się tylko rodzajem relacji i wartością opóźnienia. Zakłada się niezmienną dla wszystkich obiektów liczbę robót.

Przedmiotowy model można przedstawić również w postaci grafu. Jest on zależny od kolejności realizacji obiektów tj. permutacji  $\pi$ :

$$G(\pi) = (M \times N, E),$$

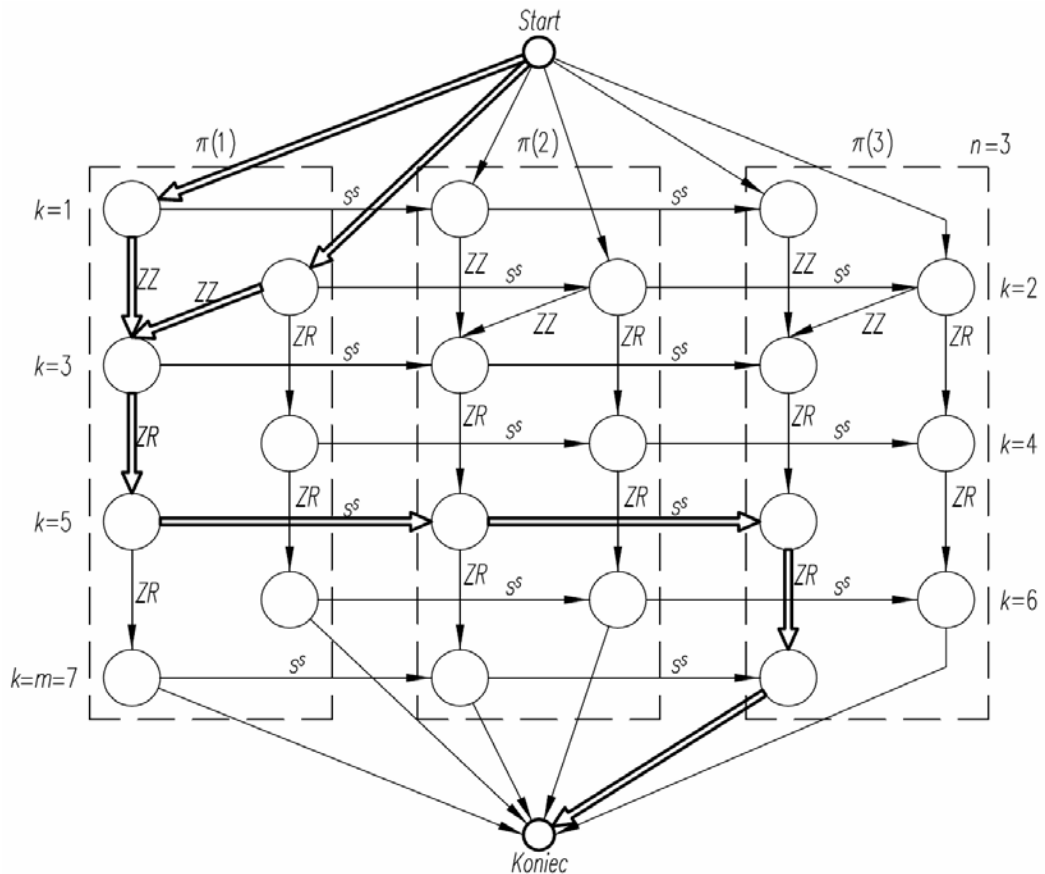
gdzie  $M \times N = \{1, \dots, k, \dots, m\} \times \{1, \dots, j, \dots, n\}$  jest zbiorem obciążonych wierzchołków reprezentujących roboty. Obciążenie wierzchołka  $(k, j)$  jest równe  $p_{k, \pi(j)}$ . Zbiór  $E = E^F \cup E^S$  zawiera krawędzie przebiegające w kierunku pionowym (technologiczne)  $E^F$  i krawędzie poziome (sekwencyjne, reprezentujące kolejność realizacji obiektów)  $E^S$ . W odróżnieniu od modelu przedstawionego w p. 5.1.1, graf  $G(\pi)$  nie tworzy struktury siatki o wymiarach  $m \times n$ ,

lecz jego postać jest zależna od przyjętych relacji kolejnościowych pomiędzy robotami dla obiektu  $Z_j$ . Krawędzie  $E^F$  przebiegające w kierunku pionowym grafu  $G(\pi)$  są obciążone wartościami przyjętych relacji kolejnościowych i ich opóźnień. Krawędzie poziome  $((k, j), (k, j+1))$  są obciążone wartościami  $s_{k,\pi(j)\pi(j+1)}^S$ . Przykładowa postać tego grafu jest podana na rys. 5.4.

Terminy zakończenia poszczególnych robót można określić ze wzoru o postaci rekurencyjnej:

$$C_{k, \pi(j)} = \max \{ C_{k, \pi(j-1)} + s_{k,\pi(j-1)\pi(j)}^S, \max_f \{ C_{f, \pi(j)} + r_{f, \pi(j)} \} \} + p_{k,\pi(j)},$$

gdzie:  $j = 1, \dots, n, k = 1, \dots, m, \pi(0) = 0, C_{k,0} = 0, C_{0,j} = 0, f = 1, \dots, m_{poprz}$ .



Rys. 5.4. Przykładowy graf  $G(\pi)$  dla rozważanego w p. 5.2.1 modelu systemu pracy potokowej ( $m = 7, n = 3$ )

Zmienna  $m_{poprz}$  określa liczbę poprzedników roboty  $k$ , zaś zmienna  $f$  określa numer roboty znajdującej się na liście poprzedników roboty  $k$ . Zmienna  $r_{f, \pi(j)}$  określa wartość opóźnienia terminu zakończenia roboty  $f$ , znajdującej się na liście poprzedników roboty  $k$ , wynikającą z przyjętej postaci relacji (RR, ZZ, ZR) w stosunku do terminu rozpoczęcia roboty  $k$ .

Dla kryterium  $C_{max}$  (minimum czasu realizacji wszystkich robót w przedsiębiorstwie) poszukiwana jest permutacja  $\pi^* \in \Pi$ , dla której:

$$C_{max}(\pi^*) = \min_{\pi \in \Pi} C_{max}(\pi), \text{ gdzie } C_{max}(\pi) = \max_i C_{i, \pi(n)}.$$

W grafie  $G(\pi)$  można wyróżnić ścieżkę krytyczną (przykład jej jest przedstawiony rys. 5.4).

Przedstawiony powyżej model systemu pracy potokowej, podobnie jak model przedstawiony w p. 5.1.1, jest NP-trudnym zagadnieniem optymalizacyjnym. Nie jest również możliwe zastosowanie algorytmu B&B do rozwiązywania zagadnienia optymalizacyjnego ze względu na możliwość występowania dowolnych (dodatnich lub ujemnych) wartości opóźnień w występujących relacjach RR, ZZ, ZR. Przyjęto rozwiązanie tego zagadnienia w modelu za pomocą trzech rozpatrzonych w rozdziale 4 algorytmów metaheurystycznych: TS, GS, SA. Ze względu na występowanie takiej samej zmiennej decyzyjnej, jak w permutacyjnym problemie przepływowym, i przyjęcie opisu relacji kolejnościowych między robotami przez taki sam dla każdego z obiektów graf  $K$ , forma tych algorytmów jest taka sama jak dla tego problemu i została przedstawiona w p. 4.1.4.1.

### 5.2.2. Model systemu pracy potokowej z wieloma grupami roboczymi wykonującymi robotę jednego rodzaju

Podobnie jak model przedstawiony w p. 5.1.2 przedmiotowy model stosuje niektóre założenia problemu przepływowego z maszynami równoległymi i z kryterium czasu wykonywania wszystkich zadań, który jest rozważany w TSZ. W odróżnieniu od modelu przedstawionego w p. 5.2.1, zostało przyjęte założenie, że jest zastosowana dowolna liczba wyspecjalizowanych grup roboczych do wykonywania roboty jednego rodzaju. Model optymalizacyjny przedmiotowego zagadnienia jest następujący:

#### Parametry:

- Przedsięwzięcie tworzy zbiór obiektów budowlanych  $Z = \{Z_1, Z_2, Z_3, \dots, Z_j, \dots, Z_n\}$ .
- Do wykonywania prac istnieją zespoły grup roboczych wykonujących robotę jednego rodzaju i tworzących zbiór  $B = \{B_1, B_2, B_3, \dots, B_k, \dots, B_m\}$ .
- W każdym zespole grup roboczych  $B_k \in B$  znajduje się  $m_k \geq 1$  takich samych grup roboczych (mają takie same wydajności lub składy):  $B_k = \{B_{k1}, B_{k2}, B_{k3}, \dots, B_{ki}, \dots, B_{km_k}\}$ .
- Każdy obiekt  $Z_j \in Z$  wymaga realizacji  $m$  robót, które tworzą zbiór  $O_j = \{O_{j1}, O_{j2}, O_{j3}, \dots, O_{jk}, \dots, O_{jm}\}$ .
- Zakłada się, że robota  $O_{jk} \in O_j$  może być realizowana przez grupę roboczą  $B_{ki} \subset B_k$ . Czas trwania roboty  $O_{jk}$  wykonywanej przez nią wynosi  $p_{jk} > 0$ . Zbiór czasów trwania  $p_j$  robót ze zbioru  $O_j$  określa wektor  $p_j = [p_{j1}, p_{j2}, p_{j3}, \dots, p_{jk}, \dots, p_{jm}]$ .
- Zależności kolejnościowe między robotami  $O_j$  dla obiektu  $Z_j$  są zadane acyklicznym grafem skierowanym  $K = (M, F)$ , gdzie  $M$  jest zbiorem wierzchołków obciążonych czasami trwania realizacji robót  $p_{jk}$  i reprezentujących roboty  $O_j$  obiektu  $Z_j$  ( $|M| = m$ ),



natomiast  $F$  jest zbiorem obciążonych łuków wyrażających ograniczenia technologiczne (przyjęte relacje kolejnościowe z wartościami opóźnień). Obciążenia łuków są podane w postaci relacji pomiędzy robotami  $RR$ ,  $ZZ$ ,  $ZR$  z przyjętymi dla nich wartościami opóźnień, które mogą przyjmować wartości dowolne. Wyrażone są za pomocą  $n$  macierzy górnych, trójkątnych  $RR_j$ ,  $ZR_j$ ,  $ZZ_j$ , o rozmiarach  $m \times m$ .

- Dodatkowe czasy niezbędne dla przemieszczania grup roboczych między obiektami, zależne od rodzaju grupy roboczej oraz kolejności realizacji obiektów, określa macierz  $S_k^S = [s_{gh}^S]_{n \times n}$ , gdzie  $g \in [1..n]$ ,  $h \in [1..n]$ ,  $k \in [1..m]$ .

#### Ograniczenia:

- Zakłada się, że w dowolnej chwili każda grupa robocza z zespołu  $B_k$  może wykonywać tylko jedną robotę.
- Zakłada się, że robota  $O_{jk} \in O_j$  jest realizowana nieprzerwanie przez jedną grupę roboczą z zespołu  $B_k$  przez czas  $p_{jk} > 0$ .

Zmienną decyzyjną jest kolejność  $\pi$  wykonywania poszczególnych robót w obiektach, którą stanowi zestaw permutacji podziału rozłącznego (jak w p. 3.3)  $\pi = (\pi_1, \pi_2, \dots, \pi_k, \dots, \pi_m)$ . Kolejność wykonywania roboty  $k$  w obiektach przez zespół grup roboczych  $B_k$  określa permutacja podziału rozłącznego  $\pi_k = (\pi_{k1}, \pi_{k2}, \dots, \pi_{ki}, \dots, \pi_{km_k})$ , gdzie  $\pi_{ki} = (\pi_{ki}(1), \pi_{ki}(2), \dots, \pi_{ki}(l), \dots, \pi_{ki}(n_{ki}))$  określa kolejność wykonywania roboty  $k$  w obiektach przydzielonych do grupy roboczej  $B_{ki}$  z zespołu grup roboczych  $B_k \subset B$  o liczności  $m_k$ .

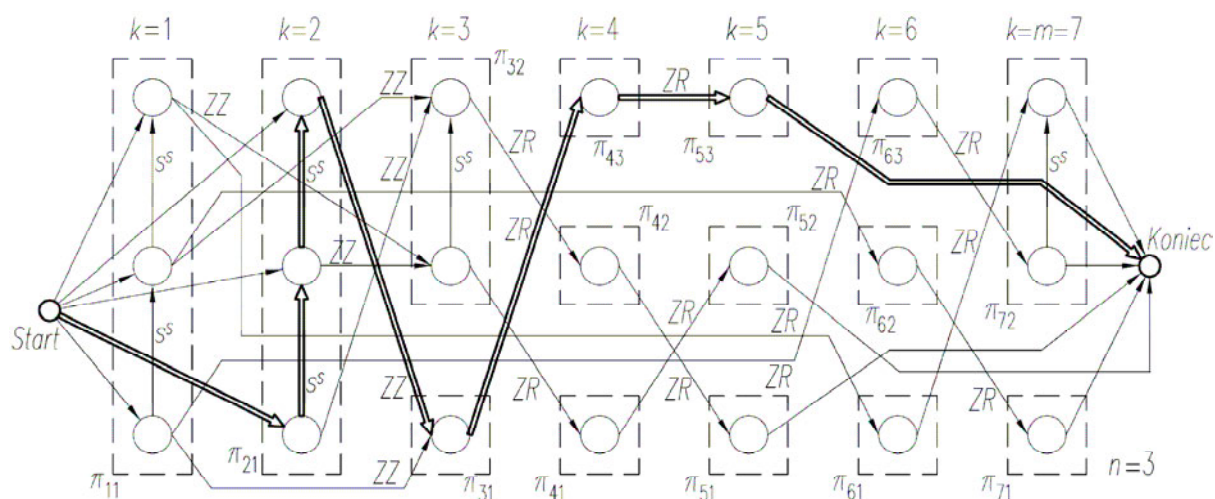
Kryterium (funkcją celu) jest termin  $C_{max}$  realizacji wszystkich robót we wszystkich obiektach. Zadanie optymalizacyjne w modelu polega na znalezieniu harmonogramu wykonywania robót, który minimalizuje wartość funkcji celu, spełniając ograniczenia podane powyżej.

Rozważany model można zidentyfikować wśród zagadnień teorii szeregowania zadań jako problem  $m$ -maszynowy, przepływowy, z maszynami równoległymi oraz z nałożonymi ograniczeniami: relacjami kolejnościowymi zadanymi grafem  $K$  i przezbrojeniami sekwencyjnie zależnymi oraz funkcją celu  $C_{max}$ . Model o opisanej postaci w ramach TSZ również nie był badany w znanych autorowi pracach. Dla przedmiotowego modelu można jednak stosować metody, które są stosowane w problemach przepływowych z maszynami równoległymi i w modelach z p. 5.1.2 i p. 5.1.3. ze względu na przyjęcie dla każdego obiektu takich samych zależności kolejnościowych (poprzedników i następników) zadanym grafem  $K$  różniących się tylko rodzajem relacji i wartością opóźnienia oraz niezmienną dla wszystkich obiektów liczbą robót. Model ten można określić jako uogólnienie w stosunku do modelu podanego w p. 5.2.1 przez zastosowanie więcej niż jednej grupy roboczej wykonującej roboty jednego rodzaju.

Rozpatrywany model można przedstawić, podobnie jak model opisany w p. 5.1.2, w formie grafu dysjunktywnego [100], [108], [134]. Forma tego grafu jest zależna od przyjętego zestawu permutacji podziału rozłącznego  $\pi$  (przykład grafu jest podany na rys. 5.5):

$$G(\pi) = (N', E(\pi)).$$

gdzie  $N'$  to zbiór węzłów  $E(\pi)$  – zbiór krawędzi. Przyjęto  $N' = N \cup \{(start), (koniec)\}$ , gdzie  $N = \{1, \dots, k, \dots, m\} \times \{1, \dots, j, \dots, n\}$  jest zbiorem węzłów reprezentujących roboty dla poszczególnych obiektów, zaś  $(start), (koniec)$  są węzłami fikcyjnymi reprezentującymi operację początkową i końcową. Obciążenie wierzchołka  $(k, j)$  jest równe  $p_{k, \pi_{ki}(l)}$ , natomiast wagi węzłów  $(start), (koniec)$  są równe zero. Zbiór krawędzi  $E(\pi) = E^F \cup E^S(\pi)$  ma budowę zależną od przyjętej zmiennej decyzyjnej  $\pi$ . Krawędzie pionowe (sekwencyjne, reprezentujące kolejność realizacji obiektów i mające charakter dysjunktywny) ze zbioru  $E^S(\pi)$  znajdują się pomiędzy wierzchołkami  $\pi_{ki}(l-1)$  i  $\pi_{ki}(l)$  dla  $i = 1, \dots, m_k, k = 1, \dots, m, l = 2, \dots, n_{ki}$ . Krawędzie technologiczne ze zbioru  $E^F$  znajdują się pomiędzy wierzchołkiem oznaczającym robotę  $k$  i wierzchołkami będącymi poprzednikami tej roboty dla obiektu  $\pi_{ki}(j)$ . Położenie tych wierzchołków jest zależne od pozycji w permutacjach podziału rozłącznego  $\pi_{k-1}$  i  $\pi_k$ . Krawędzie ze zbioru  $E^S(\pi)$  są obciążone wartościami  $s^S_{k, \pi_{ki}(l-1)\pi_{ki}(l)}$ , a krawędzie ze zbioru  $E^F$  – wartościami opóźnień wynikających z przyjętych relacji (rys. 5.5).



Rys. 5.5. Przykładowy graf  $G(\pi)$  dla rozważanego w p. 5.2.2 modelu systemu pracy potokowej ( $m = 7, n = 3$ )

Terminy zakończenia poszczególnych robót można określić ze wzoru o postaci rekurencyjnej:

$$C_{k, \pi_{ki}(l)} = \max \{ C_{k, \pi_{ki}(l-1)} + s^S_{k, \pi_{ki}(l-1)\pi_{ki}(l)}, \max_f \{ C_{f, \pi_{ki}(l)} + r_{f, \pi_{ki}(l)} \} \} + p_{k, \pi_{ki}(l)},$$

gdzie:  $j = 1, \dots, n$ ,  $l = 1, \dots, n_{ki}$ ,  $i = 1, \dots, m_k$ ,  $k = 1, \dots, m$ ,  $\pi_{ki}(0) = 0$ ,  $C_{k,0} = 0$ ,  $C_{0,j} = 0$ ,  $f = 1, \dots, m_{poprz}$ . Zmienna  $m_{poprz}$  określa liczbę poprzedników roboty  $k$ , zaś zmienna  $f$  określa numer roboty znajdującej się na liście poprzedników roboty  $k$ . Zmienna  $r_{f, \pi(j)}$  określa wartość opóźnienia terminu zakończenia roboty  $f$  znajdującej się na liście poprzedników roboty  $k$  i wynikającą z przyjętej postaci relacji (RR, ZZ, ZR) w stosunku do terminu rozpoczęcia roboty  $k$ . Dowolny graf  $G(\pi)$  dla przedmiotowego modelu posiada własność ścieżki krytycznej.

Model systemu pracy potokowej z wieloma grupami roboczymi wykonującymi robotę jednego rodzaju, podobnie jak model przedstawiony w p. 5.2.1, jest NP-trudnym zagadnieniem optymalizacyjnym i ze względu na przyjęte ograniczenia (dowolne wartości opóźnień w przyjętych relacjach) nie jest możliwe utworzenie algorytmu B&B do rozwiązywania zagadnienia optymalizacyjnego. Przyjęto rozwiązanie tego zagadnienia w analizowanym modelu za pomocą trzech rozpatrzonych w rozdziale 4 algorytmów metaheurystycznych: TS, GS, SA. Ze względu na występowanie takiej samej zmiennej decyzyjnej, jak w permutacyjnym problemie przepływowym z maszynami równoległymi, i przyjęcie opisu relacji kolejnościowych między robotami przez taki sam dla każdego z obiektów graf  $K$ , forma tych algorytmów jest taka sama, jak dla tego problemu, i została przedstawiona w p. 4.1.4.2.

### **5.2.3. Model systemu pracy potokowej z relacjami kolejnościowymi między robotami danymi grafem i z uwzględnieniem kryteriów koszt/czas**

W p. 5.2.1 i p. 5.2.2 przedstawiono dwa modele planowania w systemie pracy potokowej uwzględniające kryterium czasu trwania przedsięwzięcia budowlanego. Podobnie jak w modelach z p. 5.1.1 i 5.1.2, w celu wykonania robót przedsięwzięcia, zostały w nich zastosowane zasoby czynne (odnawialne), którymi były grupy robocze. Podobnie jak dla modelu rozważanego w p. 5.1.3, dla modeli z relacjami kolejnościowymi między robotami obiektu danymi niezmiennym grafem (p. 5.2.1 i 5.2.2), jest możliwe, oprócz czasu trwania wykonywania roboty, uwzględnienie parametru ekonomicznego użytych zasobów czynnych tj. kosztów ich użycia. Metodę ich określania oraz możliwości zastosowania przedstawiono w p. 5.1.3. Funkcja koszt/czas będzie w rozpatrywanym modelu miała charakter dyskretny. Przyjęto również, że czasy realizacji robót są wypukłymi, malejącymi funkcjami ilości przydzielonych im zasobów (kosztów) [63], przyjęto takie same założenia dotyczące tej funkcji, jak w p. 5.1.3. W modelu zastosowano statyczny i dyskretny model operacji typu czas/zasób, w rozumieniu teorii szeregowania zadań [63].

Formułowany model stosuje niektóre założenia modelu systemu pracy potokowej z wieloma grupami roboczymi wykonującą robotę jednego rodzaju, podanego w p. 5.2.2. Dodatkowymi parametrami, oprócz czasów trwania wykonywania robót, będą koszty realizacji robót. W związku z tym, że są one niezależne i przeciwstawne, podobnie jak w modelu w p. 5.1.3, w formułowanym modelu rozpatrywane są następujące funkcje celu:

- czas trwania całego przedsięwzięcia przy założonym ograniczeniu jego kosztu,
- koszt całego przedsięwzięcia przy założonym ograniczeniu czasu jego trwania,
- koszt i czas trwania całego przedsięwzięcia.

Zakłada się, że koszt całego przedsięwzięcia stanowi sumę kosztów wszystkich robót zrealizowanych na wszystkich obiektach przedsięwzięcia. Pierwsze dwie funkcje celu tworzą zagadnienia optymalizacji jednokryterialnej, takiej samej jak w modelach z p. 5.2.1 i p. 5.2.2. Trzecia funkcja celu jest zagadnieniem optymalizacji dwukryterialnej. Rozwiązaniem tego zagadnienia jest jedno z rozwiązań Pareto optymalnych, tworzących zbiór Pareto [63].

Rozważany w niniejszym podrozdziale model stosuje niektóre założenia problemu przepływowego z maszynami równoległymi, z kryterium czasu wykonywania wszystkich zadań, który jest rozważany w TSZ. Mimo przyjęcia relacji kolejnościowych między robotami danych niezmiennym grafem należy on nadal do grupy problemów przepływowych, podobnie jak modele przedstawione w p. 5.2.1 i p. 5.2.2. Przyjmuje się założenie, że zastosowana jest dowolna liczba wyspecjalizowanych grup roboczych do wykonywania roboty jednego rodzaju.

Model optymalizacyjny przedmiotowego zagadnienia jest następujący:

#### Parametry:

- Przedsięwzięcie tworzy zbiór obiektów budowlanych  $Z = \{Z_1, Z_2, Z_3, \dots, Z_j, \dots, Z_n\}$ .
- Do wykonywania prac istnieją zespoły grup roboczych wykonujących robotę jednego rodzaju i tworzących zbiór  $B = \{B_1, B_2, B_3, \dots, B_k, \dots, B_m\}$ .
- W każdym zespole grup roboczych  $B_k \in B$  znajduje się  $m_k \geq 1$  grup roboczych, które mogą posiadać takie same albo różne wydajności lub składy:  $B_k = \{B_{k1}, B_{k2}, B_{k3}, \dots, B_{ki}, \dots, B_{km_k}\}$ .
- Każdy obiekt  $Z_j \in Z$  wymaga realizacji  $m$  robót, które tworzą zbiór  $O_j = \{O_{j1}, O_{j2}, O_{j3}, \dots, O_{jk}, \dots, O_{jm}\}$ .
- Zakłada się, że robota  $O_{jk} \in O_j$  może być realizowana przez grupę roboczą  $B_{ki} \subset B_k$ . Czas trwania roboty  $O_{jk}$  wykonywanej przez nią wynosi  $p_{jki} > 0$ . Zbiór możliwych czasów trwania  $p_j$  robót ze zbioru  $O_j$  określa wektor  $p_j = [p_{j1}, p_{j2}, p_{j3}, \dots, p_{jk}, \dots, p_{jm}]$ , gdzie  $p_{jk} = [p_{jk1}, p_{jk2}, p_{jk3}, \dots, p_{jki}, \dots, p_{jmm_k}]$ .

- Koszt realizacji roboty  $O_{jk}$  przez grupę roboczą  $B_{ki}$  określa zmienna  $u_{jki} \geq 0$ . Zbiór możliwych kosztów  $u_j$  robót ze zbioru  $O_j$  określa wektor  $u_j = [u_{j1}, u_{j2}, u_{j3}, \dots, u_{jk}, \dots, u_{jm}]$ , gdzie  $u_{jk} = [u_{jk1}, u_{jk2}, u_{jk3}, \dots, u_{jki}, \dots, u_{jmm_k}]$ . Koszt roboty  $u_{jki}$  jest wyznaczany według kalkulacji kosztu wykonania roboty  $O_{jk}$  przez grupę roboczą  $B_{ki}$  znajdującą się w zasobach wykonawcy. Może on również być ofertą kosztu wykonania roboty  $O_{jk}$  przez podwykonawcę reprezentowanego przez grupę roboczą  $B_{ki}$ .
- Zależności kolejnościowe między robotami  $O_j$  dla obiektu  $Z_j$  są zadane acyklicznym grafem skierowanym  $K = (M, F)$ , gdzie  $M$  jest zbiorem wierzchołków obciążonych czasami trwania realizacji robót  $p_{jk}$  i reprezentujących roboty  $O_j$  obiektu  $Z_j$  ( $|M| = m$ ), natomiast  $F$  jest zbiorem obciążonych łuków wyrażających ograniczenia technologiczne (przyjęte relacje kolejnościowe z wartościami opóźnień). Obciążenia łuków są podane w postaci relacji pomiędzy robotami RR, ZZ, ZR i przyjętych dla nich wartości opóźnień. W przedmiotowych dwóch modelach przyjęto możliwość dwojakiej interpretacji wartości opóźnień dla przyjętych relacji. Pierwsza interpretacja jest taka sama jak w modelach przedstawionych w p. 5.2.1 i 5.2.2 dotyczących tych wartości. Wartości opóźnień dla przyjętej relacji RR, ZZ lub ZR przedstawiają wtedy konkretne, minimalne odległości czasowe między terminami rozpoczęcia lub zakończenia roboty występującej na liście poprzedników roboty  $O_{jk}$  i terminami rozpoczęcia lub zakończenia  $O_{jk}$  w tym samym obiekcie  $Z_j$ . Implementacja programowa w środowisku Mathematica dla tej interpretacji polega na przyjęciu przez zmienną reprezentującą wartość opóźnienia typu Integer (liczba całkowita). W drugiej interpretacji wartości opóźnień, dla przyjętej relacji RR, ZZ lub ZR, przedstawiają mnożnik dla czasu trwania roboty, która jest na liście poprzedników roboty  $O_{jk}$ . W wyniku tego działania matematycznego otrzymywana jest konkretna wartość opóźnienia. Przyczyną wprowadzenia takiej interpretacji, podobnie jak w modelu przedstawionym w p. 5.1.3, jest umożliwienie podania wartości opóźnienia jako wielkości proporcjonalnej do czasu trwania roboty będącej poprzednikiem roboty  $O_{jk}$ , który jest zmienny i zależny od wyboru dla niej grupy roboczej. Implementacja programowa w środowisku Mathematica dla tej interpretacji polega na przyjęciu przez zmienną reprezentującą wartość opóźnienia typu Real (liczba rzeczywista). W rozpatrywanym modelu wartości opóźnień mogą przyjmować wartości dowolne.
- Dodatkowe czasy niezbędne dla przemieszczania grup roboczych między obiektami, zależne od rodzaju grupy roboczej oraz kolejności realizacji obiektów określa macierz  $S_k^S = [s_{gh}^S]_{n \times n}$ , gdzie  $g \in [1..n]$ ,  $h \in [1..n]$ ,  $k \in [1..m]$ .

Ograniczenia:

- Zakłada się, że w dowolnej chwili każda grupa robocza z zespołu  $B_k$  może wykonywać tylko jedną robotę.
- Zakłada się, że robota  $O_{jk} \in O_j$  jest realizowana nieprzerwanie przez grupę roboczą  $B_{ki} \subset B_k$  przez czas  $p_{jki} > 0$ .
- Zakłada się (alternatywnie) ograniczenie kosztu realizacji całego przedsięwzięcia do wartości  $\hat{U} : U \leq \hat{U}$ ,  $U = \sum_{j=1}^n \sum_{k=1}^m u_{jk}$  lub ograniczenie czasu zakończenia wszystkich robót w obiektach do terminu  $\hat{C} : C_{max} \leq \hat{C}$ , gdzie  $C_{max}$  jest czasem trwania przedsięwzięcia.

Zmienną decyzyjną jest kolejność  $\pi$  wykonywania obiektów, którą stanowi zestaw permutacji podziału rozłącznego (zwany dla uproszczenia permutacją), podobnie jak w modelu w p. 5.1.3,  $\pi = (\pi_1, \pi_2, \dots, \pi_k, \dots, \pi_m) \in \Pi$ , gdzie  $\Pi$  jest zbiorem wszystkich możliwych permutacji,  $\pi_k = (\pi_{k1}, \pi_{k2}, \dots, \pi_{ki}, \dots, \pi_{km_k})$ ,  $\pi_{ki} = (\pi_{ki}(1), \pi_{ki}(2), \dots, \pi_{ki}(l), \dots, \pi_{ki}(n_{ki}))$ . Przyjęta zmienna decyzyjna jednoznacznie określa alokację grup roboczych do realizacji robót w obiektach. Podobnie jak w p. 5.1.3, za pomocą permutacji  $\pi$  jednoznacznie ustalone są czasy trwania poszczególnych robót wykonywanych w obiektach oraz ich koszt. Po przyjęciu permutacji  $\pi$  wektor czasów trwania  $p_j$  robót ze zbioru  $O_j$  jest następujący:  $p_j = [p_{j1}, p_{j2}, p_{j3}, \dots, p_{jk}, \dots, p_{jm}]$ , a zbiór kosztów  $u_j$  robót ze zbioru  $O_j$  przedstawia wektor  $u_j = [u_{j1}, u_{j2}, u_{j3}, \dots, u_{jk}, \dots, u_{jm}]$ .

Pierwszą rozważaną funkcją celu, podobnie jak w modelu w p. 5.1.3, jest czas trwania całego przedsięwzięcia przy założonym ograniczeniu jego kosztu. Poszukiwany będzie minimalny termin zakończenia wykonywania wszystkich robót  $C_{max}(\pi^*)$  dla permutacji  $\pi^* \in \Pi$ , dla której:  $C_{max}(\pi^*) = \min_{\pi \in \Pi} C_{max}(\pi)$ , gdzie  $C_{max}(\pi) = \max_{i,j} C_{i,j}$ , przy założeniu, że koszt całego przedsięwzięcia nie przekroczy wartości dyrektywnej narzuconej przez wykonawcę  $U(\pi^*) \leq \hat{U}$ ,  $U(\pi^*) = \sum_{j=1}^n \sum_{k=1}^m u_{jk}$ , gdzie  $n$  – liczba wszystkich obiektów,  $m$  – liczba robót wymaganych realizacji w każdym z obiektów.

Drugą rozważaną funkcją celu, podobnie jak w modelu w p. 5.1.3, jest koszt całego przedsięwzięcia przy założonym ograniczeniu jego terminu zakończenia. Poszukiwany będzie minimalny koszt przedsięwzięcia (rozumiany jako suma kosztów robót) dla permutacji  $\pi^* \in \Pi$ , dla której  $U(\pi^*) = \min_{\pi \in \Pi} U(\pi)$ , gdzie  $U(\pi) = \sum_{j=1}^n \sum_{k=1}^m u_{jk}$ , przy założeniu, że termin

zakończenia całego przedsięwzięcia  $C_{\max}(\pi^*)$  nie przekroczy terminu dyrektywnego narzuconego przez inwestora  $\hat{C} : C_{\max}(\pi^*) \leq \hat{C}$ .

Rozważany model można zidentyfikować wśród zagadnień teorii szeregowania zadań jako problem  $m$ -maszynowy, przepływowy, z maszynami równoległymi, z nałożonymi ograniczeniami: relacjami kolejnościowymi między robotami dla obiektu zadanymi grafem  $K$  i przebrojeniami sekwencyjnie zależnymi oraz z założeniem statycznego, dyskretnego modelu operacji typu czas/zasób. Model o opisanej postaci w ramach TSZ również nie był badany w znanych autorowi pracach. Dla przedmiotowego modelu, podobnie jak dla modelu przedstawionego w p. 5.2.2, można stosować metody, które są używane w problemach przepływowych z maszynami równoległymi i w modelach z p. 5.1.2 i p. 5.2.2 ze względu na przyjęcie dla każdego obiektu takich samych relacji kolejnościowych (poprzedników i następników) zadanymi grafem  $K$  i różniących się tylko rodzajem relacji, wartością opóźnienia. Zakłada się niezmienną dla wszystkich obiektów liczbę robót. Rozważany model można określić jako uogólnienie w stosunku do modelu podanego w p. 5.1.3 przez zastosowanie relacji kolejnościowych danych grafem  $K$ .

Rozpatrywany model przedstawia się również w formie grafu dysjunktywnego [100], [108], [134]. Forma tego grafu jest zależna od przyjętego zestawu permutacji podziału rozłącznego  $\pi$  i jest taka sama jak podano w p. 5.2.2.

Podobnie jak wszystkie poprzednio przedstawione modele, rozpatrywany model jest NP-trudnym zagadnieniem optymalizacyjnym. Przyjęto rozwiązanie zagadnienia optymalizacyjnego za pomocą trzech rozpatrzonych w rozdziale 4, algorytmów: TS, GS, SA. Forma tych algorytmów jest taka sama jak dla problemu przepływowego z maszynami równoległymi i została przedstawiona w p. 4.1.4.2.

W sformułowanym modelu występuje również, podobnie jak w modelu z p. 5.1.3, zagadnienie zbiorów rozwiązań niedopuszczalnych ze względu na przyjęte ograniczenia. Przyjęto taki sam sposób rozwiązania tego zagadnienia jak podany w p. 5.1.3. Oddzielnym zagadnieniem jest odpowiednie przyjęcie rozwiązań startowych w algorytmach. Dla obu postaci funkcji celu przyjęto taki sam sposób postępowania przy losowym wyznaczaniu tych rozwiązań, jaki został przedstawiony w p. 5.1.3.

Podobnie jak w modelu z relacjami kolejnościowymi wyrażonymi sekwencją, w przedmiotowym modelu jest możliwe rozpatrzenie podejścia dwukryterialnego, polegającego na jednoczesnej minimalizacji obu niezależnych kryteriów: całkowitego czasu i kosztu przedsięwzięcia wieloobiektowego. Rozwiązaniem tego zagadnienia jest zbiór rozwiązań Pareto optymalnych (niezdominowanych, efektywnych) ze względu na występującą w praktyce przeciwstawność obu kryteriów. Odpowiednie definicje rozwiązania

Pareto optymalnego, zbioru Pareto  $P$ , zbioru wszystkich punktów kompromisowych  $K$  zostały przedstawione w p. 5.1.3. Rozważane kryteria są następujące: całkowity koszt realizacji

przedsięwzięcia  $U(\pi) = \sum_{j=1}^n \sum_{k=1}^m u_{jk}$  oraz całkowity czas realizacji przedsięwzięcia

$C_{max}(\pi) = \max_{i,j} C_{i,j}$ . W modelu przyjmuje się, podobnie jak w p. 5.1.3, że czasy trwania robót

w obiektach są wypukłymi, malejącymi funkcjami dyskretnymi ilości przydzielonych im zasobów (kosztów), co powoduje, że charakterystyka krzywej, która przybliża punkty kompromisowe ze zbioru  $K$ , jest taka sama jak krzywej w p. 5.1.3.

Rozpatrywany problem poszukiwania zbioru punktów kompromisowych w przedmiotowym modelu, jest taki sam jak przedstawiony w p. 5.1.3. W związku z tym, proponuje się zastosowanie algorytmu przedstawionego w p. 5.1.3 dotyczącego znajdowania elementów zbioru punktów kompromisowych  $K$ , które odpowiadają wartościom progowym kosztu całego przedsięwzięcia wieloobiektowego.

### 5.3. Podsumowanie

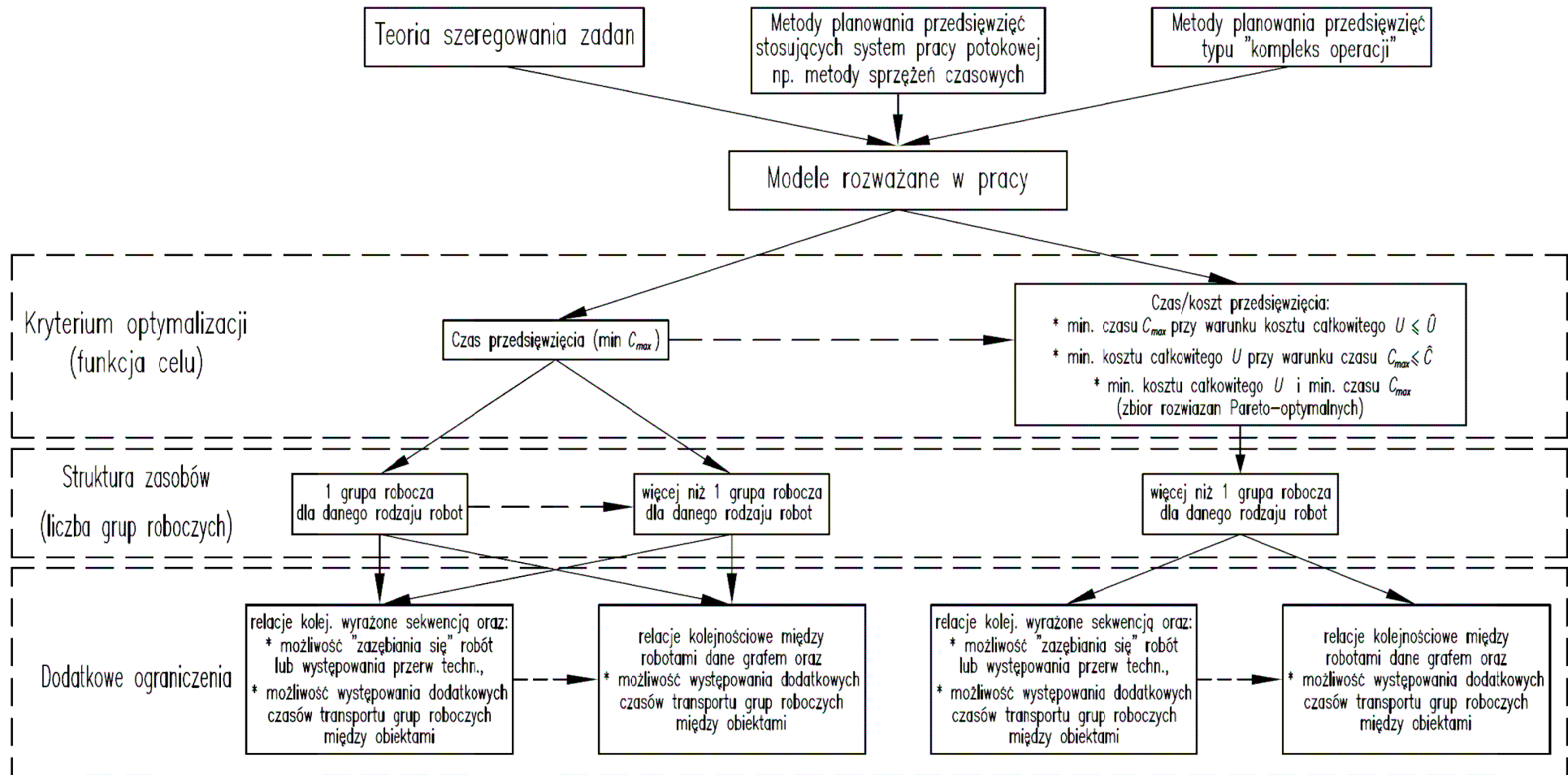
Przedstawione w niniejszym rozdziale nowe modele dla harmonogramowania przedsięwzięć wieloobiektowych wykorzystujące system pracy potokowej, stosują wiedzę z różnych dziedzin nauki: badań operacyjnych, teorii szeregowania zadań, aktualnych wyników badań w organizacji robót budowlanych. Była ona nieodzowna, aby sformułować modele oraz dostarczyć narzędzi, które będą w stanie rozwiązywać występujące w nich problemy optymalizacyjne. Zaprezentowane modele, utworzone dzięki tej wiedzy i uwzględniające niektóre nowe, niestosowane do tej pory, parametry, ograniczenia i funkcje celu, istotnie poszerzają obszar zastosowań modeli w systemie pracy potokowej. Modele te zostały przedstawione zgodnie z kierunkiem uogólniania tj. rozpoczynając od modeli uwzględniających przypadek szczególny - z relacjami kolejnościowymi wyrażonymi sekwencją (modele podane w p. 5.1), a następnie przechodząc do przypadku ogólnego - z relacjami kolejnościowymi między robotami w obiektach wyrażonymi grafem (modele podane w p. 5.2). Sposób ich przedstawienia jest związany z kolejnością prowadzenia badań nad nimi przez autora rozprawy. W ramach poszczególnych podrozdziałów przyjęto również podobny sposób przedstawiania modeli (z jedną, a następnie wieloma grupami roboczymi wykonującymi robotę jednego rodzaju; z kryterium czasu trwania przedsięwzięcia, a następnie kryteriami koszt/czas i przy podejściu dwukryterialnym). Rozważane w pracy



nowe modele zostały schematycznie przedstawione na rys. 5.6, gdzie poziomymi strzałkami zaznaczono kierunek uogólnienia kryteriów, struktury zasobów i ograniczeń.

Zaprezentowane w rozdziale 5 modele mogą być przydatnym narzędziem planistycznym dla wykonawcy, który zdecyduje się na realizację przedsięwzięcia w systemie potokowym. Istotną ich cechą jest możliwość dalszego rozwoju związanego z lepszym dostosowaniem do potrzeb planistycznych realizatorów przedsięwzięć budowlanych. Będzie to możliwe poprzez uwzględnianie dodatkowych parametrów, ograniczeń lub funkcji celu. Pozwalają na to uniwersalne i elastyczne metody optymalizacji dyskretnej użyte do rozwiązywania problemów optymalizacyjnych występujących w modelach.

W rozdziale 6 zostanie przedstawiona weryfikacja sformułowanych modeli przez zaprezentowanie możliwości ich zastosowania w przykładach przedsięwzięć budowlanych.



Rys. 5.6. Nowe modele w systemie pracy potokowej przedsięwzięć wieloobektowych przedstawione w pracy – schemat

## **6. Weryfikacja sformułowanych modeli harmonogramowania przedsięwzięć wieloobektowych wykorzystujących system pracy potokowej**

W rozdziale 5 niniejszej pracy przedstawiono nowe modele harmonogramowania w systemie pracy potokowej dla przedsięwzięć wieloobektowych. W celu weryfikacji przydatności sformułowanych modeli do planowania tych przedsięwzięć w niniejszym rozdziale przedstawione zostaną przykłady takiego planowania. Każdy z przykładów ma indywidualnie zdefiniowane ograniczenia związane z warunkami technologicznymi przedsięwzięcia oraz warunkami inwestora i wykonawcy. W przedsięwzięciach przyjęto, że obiekty mają charakter niejednorodny. Do analizy procesów wznoszenia takich obiektów, w tym poszukiwania optymalnych harmonogramów realizacji robót opracowane i przedstawione poprzednio modele i algorytmy ich rozwiązywania okazują się w pełni przydatne.

### **6.1. Harmonogramowanie robót budowlanych na przykładzie kompleksu budynków mieszkalnych**

W niniejszym podrozdziale, na przykładzie planowania robót budowlanych dwunastu budynków mieszkalnych zostaną zaprezentowane możliwości zastosowania modeli, które przedstawiono w p. 5.1. Zakłada się, że inwestor nie narzuca wykonawcy warunków odnośnie do kolejności oddawania obiektów do użytku. W przedsięwzięciu przyjęto założenie, że budynki są odległe od siebie na tyle, że występuje potrzeba uwzględniania w harmonogramie dodatkowych czasów ze względu na transport grup roboczych, maszyn i sprzętu między obiektami. Charakter robót w obiektach przedsięwzięcia i możliwości ich wykonywania są odzwierciedlane przez występujące ograniczenia polegające na przyjęciu relacji kolejnościowych między robotami wyrażonych sekwencją, z możliwościami ich zazębienia się lub występowania przerw technologicznych pomiędzy nimi.

#### **6.1.1. Przykład zastosowania modelu harmonogramowania przedstawionego w p. 5.1.1**

Wykonawca na zlecenie inwestora ma zrealizować przedsięwzięcie w postaci  $n = 12$  budynków mieszkalnych odległych od siebie. Każdy z nich wymaga wykonania  $m = 11$  robót w ustalonej sekwencji wynikającej z technologii. Wykonawca dysponuje tylko własnymi grupami roboczymi, wyspecjalizowanymi do wykonywania jednego rodzaju robót i prowadzącymi roboty w każdym z obiektów ze stałą wydajnością. Zakłada się, że dysponuje

on po jednej grupie dla danego rodzaju robót. Na podstawie pracochłonności robót w poszczególnych obiektach oraz składu i wydajności grup roboczych wykonawcy ustalono czasy trwania realizacji robót w obiektach, które podano w tabeli 6.1. Między robotami i w ciągu technologicznym występują sprzężenia między obiektami, które zostały ustalone na podstawie istniejących ograniczeń technologicznych. Są one przedstawione w tabeli 6.2.

Tabela. 6.1. Czasy trwania realizacji robót przez grupy robocze wyrażone w dniach roboczych

Numer i rodzaj roboty $k =$	Obiekty $j =$											
	1	2	3	4	5	6	7	8	9	10	11	12
1 (roboty ziemne)	3	3	3	3	3	3	2	3	2	3	3	3
2 (fundamenty)	12	10	9	11	12	15	14	15	14	14	13	10
3 (ściany, stropy)	22	25	23	26	23	27	27	19	23	25	22	25
4 (wieżba dachowa)	14	15	16	17	16	13	16	11	12	16	18	15
5 (instalacje)	10	8	13	10	11	9	9	12	13	9	12	8
6 (stolarka)	3	3	4	4	2	3	4	3	4	3	2	2
7 (tynki, posadzki, poddasze)	20	22	19	18	21	25	19	26	26	20	25	23
8 (ogrodzenie, podjazdy)	15	15	18	19	18	11	16	15	10	20	20	11
9 (glazura, malowanie)	8	10	8	7	8	7	10	7	7	8	7	8
10 (posadzki)	8	11	7	8	10	6	8	5	7	7	9	5
11 (biały montaż)	5	5	5	5	4	4	3	6	5	5	7	6

Tabela. 6.2. Sprzężenia między obiektami występujące między robotami  $k$  i  $k+1$  [dni robocze]

Roboty $k =$	Obiekty $j =$											
	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	0	0	0	0	0
2	5	5	5	5	5	5	5	5	5	5	5	5
3	0	0	0	0	0	0	0	0	0	0	0	0
4	-4	-5	-5	-5	-5	-4	-5	-3	-4	-5	-5	-5
5	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
6	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
7	0	0	0	0	0	0	0	0	0	0	0	0
8	-15	-15	-18	-19	-18	-11	-16	-15	-10	-20	-20	-11
9	10	10	10	10	10	10	10	10	10	10	10	10
10	-2	-3	-2	-2	-3	-2	-2	-1	-2	-2	-2	-1
11	0	0	0	0	0	0	0	0	0	0	0	0

Dane dotyczące czasów wymaganych ze względu na przemieszczanie grup roboczych między obiektami (zależnych od rodzaju grupy roboczej oraz kolejności realizacji obiektów) są zapisane w postaci 11 macierzy  $S^S_k$  ( $k = 1...11$ ):  $S^S_1 = S^S_2 = S^S_3 = S^S_7 = S^S_8 = [s^S_{gh}]_{n \times n} = 1$ ,  $S^S_4 = S^S_5 = S^S_6 = S^S_9 = S^S_{10} = S^S_{11} = [s^S_{gh}]_{n \times n} = 0$ . W związku z tak ustaloną strukturą

i charakterystyką użytych zasobów (grup roboczych), wykonawca ma możliwość utworzenia optymalnego harmonogramu z uwzględnieniem kryterium czasu trwania całego przedsięwzięcia  $C_{max}$ , co odpowiada założeniom modelu systemu pracy potokowej przedstawionego w p. 5.1.1.

Problem optymalizacyjny występujący w zaprezentowanym przykładzie rozwiązano za pomocą opisanych w pracy algorytmów metaheurystycznych: TS, GS, SA. Forma tych algorytmów została przedstawiona w p. 4.1.4.1. Zostały zastosowane następujące wartości parametrów sterujących w algorytmach (na podstawie prac [104], [106], [108], [118], [134]):

- algorytm TS: długość listy tabu  $LT = 0,5n$ , nie wprowadzono dodatkowych ograniczeń liczności zbioru otoczenia  $N_{\pi}$ , maksymalna liczba iteracji – 20000,
- algorytm GS: licznosc populacji –  $10n$ , licznosc pokolenia jako odsetek licznosci populacji –  $0,2 \times 10n = 2n$ , prawdopodobienstwo krzyzowania – 1,0, licznosc zbioru osobnikow mutowanych w pokoleniu  $0,02 \times 10n = 0,2n$ , prawdopodobienstwo mutacji – 0,1, licznosc zbioru osobnikow najgorzej przystosowanych z populacji, ktore sa wymieniane na nowe –  $0,9 \times 10n = 9n$ , liczba iteracji, po ktorej nastepuje wymiana i zmiana operatora krzyzowania - 100, calkowita liczba iteracji – 20000,
- algorytm SA: wartosci temperatur: poczatkowej  $T_0 = 60$ , koncowej  $T_N = 0,5$ , liczba losowan wystepujaca w kroku 1 algorytmu -  $1,0n$ , wartosc parametru  $\varepsilon = 0,3$ , calkowita liczba iteracji – 20000.

Obliczenia wykonano dla kazdego z algorytmow trzykrotnie. Wyniki obliczen z najlepszymi wartosciami  $C_{max}$  sa przedstawione w tabeli 6.3.

Tabela. 6.3. Wyniki obliczeń

	Algorytmy optymalizacyjne		
	TS	GS	SA
<i>l</i>	<i>2</i>	<i>3</i>	<i>4</i>
Najlepsza wartość $C_{max}$ [dni robocze]	<b><u>384</u></b>	<b><u>384</u></b>	<b><u>384</u></b>
Numer iteracji w której osiągnięto najlepszą wartość $C_{max}$	2	17	50
Permutacja dla najlepszej wartości $C_{max}$	{3,12,1,10,8,11,9,5,4,7,2,6}	{3,8,7,9,1,5,4,11,10,12,2,6}	{3,1,5,8,6,9,4,7,11,2,10,12}

Warte uwagi jest bardzo szybkie znalezienie przez trzy algorytmy takich samych najlepszych wartości  $C_{max}$ . Można z dużym prawdopodobieństwem przypuszczać, że została znaleziona wartość optymalna tego kryterium. W załączniku przedstawiono harmonogram realizacji przedmiotowego przedsięwzięcia w postaci sieci zależności (rys. Z.1).

### 6.1.2. Przykład zastosowania modelu harmonogramowania przedstawionego w p. 5.1.2

W celu weryfikacji przydatności modelu sformułowanego w p. 5.1.2, który zakłada możliwość zastosowania w przedsięwzięciu więcej niż jednej grupy roboczej dla wykonywania roboty jednego rodzaju, posłużono się również przykładem planowania realizacji  $n = 12$  budynków mieszkalnych z p. 6.1.1, gdzie są przedstawione szczegółowe czasy trwania robót, wartości sprzężeń między obiektami, czasy niezbędne dla transportu grup roboczych między obiektami. Zakłada się, że wykonawca dysponuje następującymi ilościami grup roboczych (o takiej samej wydajności i wielkości) dla poszczególnych robót ( $m = 11$ ):  $k = 1$  (roboty ziemne) – 3 grupy,  $k = 2$  (fundamenty) – 4 grupy,  $k = 3$  (ściany, stropy) – 4 grupy,  $k = 4$  (więźba dachowa) – 4 grupy,  $k = 5$  (instalacje) – 3 grupy,  $k = 6$  (stolarka) – 2 grupy,  $k = 7$  (tynki, posadzki, poddasze) – 4 grupy,  $k = 8$  (ogrodzenie, podjazdy) – 3 grupy,  $k = 9$  (glazura, malowanie) – 3 grupy,  $k = 10$  (posadzki) – 3 grupy,  $k = 11$  (biały montaż) – 2 grupy. W przedmiotowym przykładzie rozpatrywane będzie kryterium czasu trwania całego przedsięwzięcia  $C_{max}$ , co odpowiada założeniom modelu systemu pracy potokowej przedstawionego w p. 5.1.2.

Zawarte w przykładzie zadanie optymalizacyjne rozwiązano za pomocą algorytmów TS, GS i SA. Forma tych algorytmów została przedstawiona w p. 4.1.4.2. Zastosowano następujące wartości parametrów sterujących w algorytmach (na podstawie prac [100], [106], [108], [134]):

- algorytm TS: długość listy tabu  $LT = \frac{1}{3}mn$ , nie wprowadzono dodatkowych ograniczeń liczności zbioru otoczenia  $N_{\alpha}$  maksymalna liczba iteracji – 20000,
- algorytm GS: licznosc populacji –  $10n$ , licznosc pokolenia jako odsetek licznosci populacji –  $0,2 \times 10n = 2n$ , prawdopodobienstwo krzyzowania – 1,0, licznosc zbioru osobnikow mutowanych w pokoleniu  $0,2 \times 10n = 2n$ , prawdopodobienstwo mutacji – 1,0, licznosc zbioru osobnikow najgorzej przystosowanych z populacji, ktore sa wymieniane na nowe –  $0,9 \times 10n = 9n$ , liczba iteracji, po ktorej nastepuje wymiana i zmiana operatora krzyzowania - 100, calkowita liczba iteracji – 20000,
- algorytm SA: wartosci temperatur: poczatkowej  $T_0 = 60$ , koncowej  $T_N = 0,5$ , liczba losowan wystepujaca w kroku 1 -  $1,0n$ , wartosc parametru  $\varepsilon = 0,3$ , calkowita liczba iteracji – 20000.

Obliczenia wykonano dla kazdego z algorytmow trzykrotnie. Wyniki obliczen z najlepszymi wartosciami  $C_{max}$  sa przedstawione w tabeli 6.4.

Tabela. 6.4. Wyniki obliczeń

	Algorytmy optymalizacyjne		
	TS	GS	SA
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>
Najlepsza wartość $C_{max}$ [dni robocze]	<b><u>179</u></b>	<b>190</b>	<b>190</b>
Numer iteracji w której osiągnięto najlepszą wartość $C_{max}$	2638	19233	19952
Permutacja dla najlepszej wartości $C_{max}$	{{{1,4,6,3,2,10,8,9},{12,11},{5,7}}, {{5,3,11},{12,6,2},{4,9,7,10},{1,8}}, {{12,11,6},{5,2,7},{4,9,10},{1,3,8}}, {{4,2,10},{1,9},{12,3,8,7},{5,11,6}}, {{12,9,7},{1,4,11,6},{5,3,2,8,10}}, {{1,5,3,7,8,10},{12,4,11,2,9,6}}, {{12,2,8},{4,3,6},{5,11,10},{1,9,7}}, {{5,12,3,6,7},{4,11,8},{1,2,9,10}}, {{12,11,6,8},{1,5,2,10},{4,3,9,7}}, {{1,12,11,2,8},{9,7},{4,3,5,6,10}}, {{1,12,9,4,10,7},{11,2,5,6,3,8}}}	{{{},{5,1},{8,10,11,4,9,2,12,3,6,7}}, {{1},{5},{8,9,2,12,4},{10,11,3,6,7}}, {{10,2,6},{5,11,4},{9,3},{8,1,12,7}}, {{8,2,4},{1},{10,11,3},{5,9,12,6,7}}, {{11},{5,10,1,3,4},{8,9,2,12,6,7}}, {{3},{8,5,10,1,9,2,12,11,4,6,7}}, {{11,6},{5,12},{8,1,3,2},{10,9,4,7}}, {{12,11,6},{5,1,3,2},{8,10,9,4,7}}, {{{},{3,1,11,2,6},{5,8,9,12,10,4,7}}, {{5,1,11,2},{12,10,6},{9,8,3,4,7}}, {{3,10,11,2,7},{5,8,12,1,9,4,6}}}	{{{7},{5,9,12,2,6},{3,4,10,1,8,11}}, {{5,2,12,11,6},{3,4,10},{7,8},{9,1}}, {{3,1,4},{7,8,11},{5,2,10},{9,12,6}}, {{3,9},{5,2,4},{7,8,11,10},{1,12,6}}, {{4,6,12},{5,9,1},{3,7,8,2,11,10}}, {{3,7,9,4,11},{5,8,1,2,6,10,12}}, {{3,9,11,10},{1,6},{7,2},{5,8,4,12}}, {{7,2,4,10},{1,8,6,12},{3,9,5,11}}, {{2,4,6,10},{7,9,8,11},{3,5,1,12}}, {{9,7,2,8,6,12},{3,1,4},{11,5,10}}, {{1,11,7,3,5,10},{9,8,2,4,6,12}}}

Wyniki obliczeń potwierdzają wnioski z analizy eksperymentalnej przeprowadzonej w p. 4. Najlepsze wyniki dostarczył algorytm TS, w którym nie zastosowano dodatkowych warunków ograniczających przeszukiwanie otoczenia  $N_{\pi}$ . W załączniku został przedstawiony harmonogram realizacji przedsięwzięcia w formie sieci zależności (rys. Z.2).

### 6.1.3. Przykład zastosowania modelu harmonogramowania przedstawionego w p. 5.1.3

W modelu przedstawionym w p. 5.1.3 stosowane są kryteria koszt/czas. Podczas tworzenia tego modelu stosowano założenia modelu z p. 5.1.2, w którym jest możliwość użycia więcej niż jednej grupy roboczej do wykonywania roboty jednego rodzaju. Dla zaprezentowania możliwości aplikacji przedmiotowego modelu wykorzystano przykład planowania realizacji  $n = 12$  budynków mieszkalnych. Przedsięwzięcie w tym przypadku będzie realizowane przez wykonawcę, który dysponuje własnymi, wyspecjalizowanymi do realizacji jednego rodzaju robót, zespołami grup roboczych o stałych lub różnych wydajnościach. Zakłada się, że pracownicy wchodzący w skład grup roboczych wykonawcy są zatrudnieni na stałe (np. na umowę o pracę) i dla nich nie wyznacza się kosztów ich użycia, gdyż wykonawca zawsze ponosi ten sam koszt ich zatrudnienia. Liczba grup roboczych wykonawcy do realizacji poszczególnych robót jest ograniczona. Ponadto jego grupy robocze mogą realizować tylko niektóre z rodzajów robót. W związku z tym, zamierza on użyć

zasobów zewnętrznych - podwykonawców (dodatkowych grup roboczych wykonujących tylko robotę jednego rodzaju), którzy będą realizować roboty  $k = 1, 2, 3, 4, 7$ . Każdy z podwykonawców przedstawił wykonawcy swoją ofertę zawierającą cenę i czas realizacji roboty w danym obiekcie. Na podstawie pracochłonności robót w poszczególnych obiektach oraz wydajności własnych grup roboczych wykonawca ustalił czasy trwania realizacji robót w obiektach. Szczegółowe dane dotyczące ofert podwykonawców, czasów trwania realizacji robót przez grupy robocze wykonawcy i ilości poszczególnych grup roboczych są podane w tabelach 6.5 i 6.6. Między robotami występującymi w ciągu technologicznym występują sprzężenia między obiektami, które zostały ustalone na podstawie istniejących ograniczeń technologicznych i są przedstawione w tabeli 6.7. Dodatkowe czasy transportu grup roboczych pomiędzy obiektami przedsięwzięcia zostały przyjęte jak w p. 6.1.1.

Tabela 6.5. Czasy trwania robót  $k = 5, 6, 10, 11$  wykonywanych przez zespoły grup roboczych wykonawcy składających się z grup roboczych o stałych składach i wydajnościach, wyrażone w dniach roboczych

Roboty $k =$	Liczba grup wykonawcy	Obiekty $j =$											
		1	2	3	4	5	6	7	8	9	10	11	12
5 (instalacje)	4	10	8	13	10	11	9	9	12	13	9	12	8
6 (stolarka)	4	3	3	4	4	2	3	4	3	4	3	2	2
10 (posadzki)	4	8	11	7	8	10	6	8	5	7	7	9	5
11 (biały montaż)	3	5	5	5	5	4	4	3	6	5	5	7	6

Tabela 6.6. Czasy trwania i koszty robót, które są wykonywane przez podwykonawców albo przez grupy robocze wykonawcy o różnych składach lub wydajnościach

Numer grupy roboczej dla poszczególnych rodzajów robót		Obiekty $j =$											
		1	2	3	4	5	6	7	8	9	10	11	12
Roboty ziemne ( $k = 1$ )													
1	Czas [dni robocze]	3	3	3	3	3	3	2	3	2	3	3	3
	Koszt [tys. PLN]	5,50	5,48	6,37	6,20	4,82	4,95	3,86	5,56	4,41	5,87	6,07	6,25
2	Czas [dni robocze]	3	3	3	3	3	4	2	2	2	3	2	4
	Koszt [tys. PLN]	5,08	6,31	7,49	6,75	5,09	3,59	3,20	6,98	4,13	6,42	7,71	4,98
3	Czas [dni robocze]	2	3	3	4	4	3	2	2	2	3	3	3
	Koszt [tys. PLN]	6,84	6,43	6,14	5,02	4,01	5,01	4,60	6,96	5,30	6,47	5,61	6,13
4	Czas [dni robocze]	4	4	4	4	3	4	2	3	2	4	4	3
	Koszt [tys. PLN]	3,88	4,37	4,67	4,80	4,90	3,67	3,15	5,83	3,91	4,66	4,71	6,41



Tabela 6.6. cd.

Numer grupy roboczej dla poszczególnych rodzajów robót		Obiekty $j =$											
		1	2	3	4	5	6	7	8	9	10	11	12
<b>Fundamenty (<math>k = 2</math>)</b>													
1	Czas [dni robocze]	12	10	9	11	12	15	14	15	14	14	13	10
	Koszt [tys. PLN]	-	-	-	-	-	-	-	-	-	-	-	-
2	Czas [dni robocze]	15	13	9	13	11	13	16	17	14	11	11	9
	Koszt [tys. PLN]	19,03	14,42	19,12	19,21	26,90	35,52	24,36	25,50	27,17	34,40	29,82	21,68
3	Czas [dni robocze]	13	11	9	9	13	12	19	14	11	17	12	11
	Koszt [tys. PLN]	21,85	17,74	18,05	28,16	22,15	37,16	20,22	31,52	34,38	23,23	27,58	17,85
4	Czas [dni robocze]	11	10	11	14	12	17	17	12	16	13	14	9
	Koszt [tys. PLN]	25,90	19,50	14,86	16,92	25,12	27,31	22,27	36,74	23,96	30,22	24,67	21,73
<b>Ściany, stropy (<math>k = 3</math>)</b>													
1	Czas [dni robocze]	22	25	23	26	23	27	27	19	23	25	22	25
	Koszt [tys. PLN]	-	-	-	-	-	-	-	-	-	-	-	-
2	Czas [dni robocze]	21	34	18	29	32	22	24	27	25	20	18	25
	Koszt [tys. PLN]	-	-	-	-	-	-	-	-	-	-	-	-
3	Czas [dni robocze]	24	24	23	34	19	21	25	22	20	24	19	21
	Koszt [tys. PLN]	52,22	67,15	60,73	51,33	75,17	90,07	77,03	43,61	70,44	66,69	68,94	76,79
4	Czas [dni robocze]	18	22	18	29	19	38	22	19	22	21	24	21
	Koszt [tys. PLN]	69,57	72,79	75,99	60,80	74,81	50,25	87,02	50,74	62,15	76,91	54,73	75,29
5	Czas [dni robocze]	25	28	28	21	25	21	22	15	28	29	20	23
	Koszt [tys. PLN]	49,62	56,85	50,45	85,53	55,62	90,66	86,43	65,19	48,59	55,59	63,18	69,83
<b>Wieżba dachowa (<math>k = 4</math>)</b>													
1	Czas [dni robocze]	14	15	16	17	16	13	16	11	12	16	18	15
	Koszt [tys. PLN]	-	-	-	-	-	-	-	-	-	-	-	-
2	Czas [dni robocze]	14	20	23	16	16	12	14	13	11	13	15	14
	Koszt [tys. PLN]	57,81	45,82	46,17	73,89	63,83	56,82	74,80	36,46	52,71	79,37	87,35	65,24
3	Czas [dni robocze]	14	16	16	20	17	14	13	9	12	22	16	20
	Koszt [tys. PLN]	57,69	58,04	67,13	57,55	58,78	46,98	76,72	51,61	47,97	46,73	82,81	44,98
4	Czas [dni robocze]	20	14	15	16	22	12	13	9	13	22	17	13
	Koszt [tys. PLN]	39,78	68,11	68,80	74,22	46,70	55,70	77,54	52,84	43,25	47,66	75,70	69,69
<b>Tynki, posadzki, poddasze (<math>k = 7</math>)</b>													
1	Czas [dni robocze]	20	22	19	18	21	25	19	26	26	20	25	23
	Koszt [tys. PLN]	60,35	65,11	56,57	55,81	63,84	74,75	57,97	77,21	77,92	61,69	74,19	68,42
2	Czas [dni robocze]	24	18	26	14	17	22	18	23	24	19	19	19
	Koszt [tys. PLN]	50,20	78,71	40,57	72,32	78,73	86,25	60,70	89,19	84,18	64,55	96,10	83,91
3	Czas [dni robocze]	20	19	18	15	18	29	17	27	21	17	24	22
	Koszt [tys. PLN]	59,79	77,07	59,00	67,32	73,83	65,41	64,21	74,23	97,10	74,07	77,64	70,54
4	Czas [dni robocze]	23	26	15	20	22	28	26	36	24	19	20	31
	Koszt [tys. PLN]	53,32	54,30	73,19	49,88	59,65	67,76	42,58	55,67	84,85	66,25	91,58	51,48
5	Czas [dni robocze]	21	25	22	17	26	25	24	35	35	24	30	30
	Koszt [tys. PLN]	56,95	58,17	48,11	59,66	51,71	74,95	46,28	58,17	57,94	50,73	62,62	51,73

Tabela 6.6. cd.

Numer grupy roboczej dla poszczególnych rodzajów robót		Obiekty $j =$											
		1	2	3	4	5	6	7	8	9	10	11	12
Ogrodzenie, podjazdy ( $k = 8$ )													
1	Czas [dni robocze]	15	15	18	19	18	11	16	15	10	20	20	11
	Koszt [tys. PLN]	-	-	-	-	-	-	-	-	-	-	-	-
2	Czas [dni robocze]	17	17	20	21	20	12	18	17	11	22	22	12
	Koszt [tys. PLN]	-	-	-	-	-	-	-	-	-	-	-	-
3	Czas [dni robocze]	14	14	16	17	16	10	14	14	9	18	18	10
	Koszt [tys. PLN]	-	-	-	-	-	-	-	-	-	-	-	-
4	Czas [dni robocze]	20	20	23	25	23	14	21	20	13	26	26	14
	Koszt [tys. PLN]	-	-	-	-	-	-	-	-	-	-	-	-
Glazura, malowanie ( $k = 9$ )													
1	Czas [dni robocze]	8	10	8	7	8	7	10	7	7	8	7	8
	Koszt [tys. PLN]	-	-	-	-	-	-	-	-	-	-	-	-
2	Czas [dni robocze]	6	8	6	6	6	6	8	6	6	6	6	6
	Koszt [tys. PLN]	-	-	-	-	-	-	-	-	-	-	-	-
3	Czas [dni robocze]	10	12	10	8	10	8	12	8	8	10	8	10
	Koszt [tys. PLN]	-	-	-	-	-	-	-	-	-	-	-	-
4	Czas [dni robocze]	6	7	6	5	6	5	7	5	5	6	5	6
	Koszt [tys. PLN]	-	-	-	-	-	-	-	-	-	-	-	-

Tabela 6.7. Sprzężenia między obiektami  $S^F$  występujące między robotami  $k$  i  $k+1$

	Roboty $k =$										
	1	2	3	4	5	6	7	8	9	10	11
Wartość $S^F$ [dni robocze]	0	5	0	$-0.3p_{j4i}$	-1	-1	0	$-1.0p_{j8i}$	10	$-0.25p_{j10i}$	0

W pierwszym przypadku wykonawca ma do dyspozycji w swoim budżecie tej inwestycji dla podwykonawców kwotę 1 350 000 PLN. W związku z tym ograniczeniem, planista wykonawcy ma za zadanie ustalić taki harmonogram wykonywania robót przedsięwzięcia, aby zminimalizować czas jego trwania. Zadanie to odpowiada założeniom modelu przedstawionego w p. 5.1.3 z kryterium czasu trwania przedsięwzięcia  $C_{max}$  przy ograniczeniu jego kosztu  $\hat{U}$  wynoszącym w tym przypadku 1 350 000 PLN. Wartości wszystkich parametrów sterujących w rozpatrywanych trzech algorytmach TS, GS i SA przyjęto takie same jak w p. 6.1.2. Podobnie jak poprzednio, obliczenia dla każdego z algorytmów wykonano trzykrotnie. Uzyskane wyniki suboptymalne są zawarte w tabeli 6.8.

Tabela 6.8. Wyniki obliczeń optymalizacyjnych dla rozpatrywanego przykładu z uwzględnieniem kryterium czasu trwania przedsięwzięcia  $C_{max}$  przy ograniczeniu jego kosztu  $\hat{U} = 1\,350\,000$  PLN

1	Algorytmy optymalizacyjne		
	TS	GS	SA
2	3	4	
Ograniczenie $\hat{U}$ na koszt podwykonawców [tys. PLN]	1350.00		
Koszt podwykonawców $U$ [tys. PLN]	<b><u>1345,99</u></b>	<b>1341,05</b>	<b>1338,17</b>
Minimalny czas przedsięwzięcia $C_{max}$ [dni robocze]	<b><u>195</u></b>	<b>220</b>	<b>210</b>
Numer iteracji w której osiągnięto najlepszą wartość $C_{max}$	10994	4450	19875
Permutacja dla najlepszej wartości $C_{max}$	((11,1,12),(3,6),(9),(2,4,5,7,8,10)), ((11,5,2,4,10,6,7),(9,12),(3,8),(1)), ((9,5,4,6),(3,12,1,10,7),(0),(2),(11,8)), ((3,11,5,9,1,4,7),(8,2,6),(12),(10)), ((8,5,3,7),(11,2,1,6),(12,9,4),(10)), ((11,2,1),(8,9,6),(7),(5,3,12,10,4)), ((5,10),(2,3,7),(8,1,6),(12,4),(11,9)), ((5,9,7),(8,3),(11,12,10,6),(2,1,4)), ((8),(5,2,9,1,12,6),(11,3,10),(4,7)), ((8,11,4),(9,1,3,6),(10,7),(5,2,12)), ((11,5,4),(2,10,1,8,7),(9,3,12,6))	((0),(4),(11,5,7,8,1,2,9,10,3,12,6)), ((11,5,8,6,12),(1,2,9),(4),(7,10,3)), ((4,5,3,8,2),(11,7,10,12),(1),(6),(9)), ((11,7,4,3,9,8,2,6),(10),(5,12),(1)), ((7,4,1),(11,9,8,2),(5,10,3,12),(6)), ((12,11),(5,10,4,1,9,6),(8,2),(7,3)), ((1,8),(5,3),(9,11),(7,4,2),(10,12,6)), ((7,4,8),(5,1,9,11),(2),(10,3,12,6)), ((3),(4,11),(1,5),(10,7,8,9,2,12,6)), ((7,10,5,4,1,11,6),(3,8),(2),(9,12)), ((4,11),(5,10,7,8,6),(1,3,9,2,12))	((10),(4,5,12),(11,9,2,7,1,6),(8,3)), ((4,10,7,2,9,1,6),(11,3,5),(12),(8)), ((3,10,1,2),(11,12,7,9,6),(4),(5),(8)), ((8,11,3,5,12,7,9,2),(6),(4),(10,1)), ((2),(11,8,3,4),(12,1,9),(5,7,10,6)), ((3,12,5,1),(4,6,2),(11),(8,7,9,10)), ((8,12,7,6),(3,1,2),(0),(4,9),(11,5,10)), ((3,9,6),(5,2),(11,4,12,1,10),(8,7)), ((4,11,7,3,10,6),(12,8,1),(0),(5,9,2)), ((8,11,4,10,6),(5,7,3),(12,1,9),(2)), ((10,2),(3,1,9),(12,11,5,7,4,8,6))

W drugim przypadku wykonawca musi spełnić dyrektywny czas trwania przedsięwzięcia narzucony przez inwestora i wynoszący 180 dni roboczych (tak, aby inwestycja została zrealizowana w trakcie jednego sezonu). W związku z tym ograniczeniem, planista wykonawcy ma za zadanie ustalić taki harmonogram wykonywania robót przedsięwzięcia, aby zminimalizować koszt użytych zasobów zewnętrznych - podwykonawców. Zadanie to odpowiada założeniom modelu przedstawionego w p. 5.1.3

z kryterium kosztu realizacji przedsięwzięcia  $U = \sum_{j=1}^n \sum_{k=1}^m u_{jk}$ , przy ograniczeniu czasu jego

trwania  $\hat{C}$ . Wartości wszystkich parametrów sterujących w rozpatrywanych trzech algorytmach TS, GS i SA przyjęto takie same jak w p. 6.1.2. Wyniki suboptymalne, które są uzyskane po trzykrotnym zastosowaniu każdego z algorytmów, przedstawiono w tabeli 6.9.

Tabela 6.9. Wyniki obliczeń optymalizacyjnych dla rozpatrywanego przykładu realizacji

przedsięwzięcia  $U = \sum_{j=1}^n \sum_{k=1}^m u_{jk}$ , przy ograniczeniu czasu jego trwania  $\hat{C} = 180$  dni

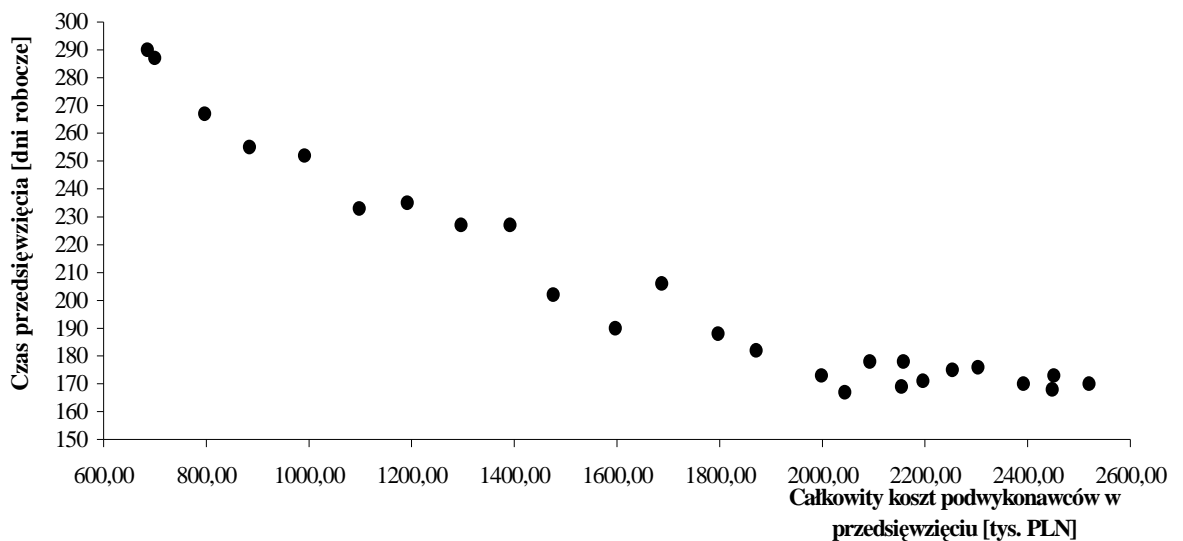
roboczych

	Algorytmy optymalizacyjne		
	TS	GS	SA
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>
Ograniczenie $\hat{C}$ na czas realizacji przedsięwzięcia [dni robocze]	180		
Minimalny koszt podwykonawców $U$ [tys. PLN]	<b>1490,02</b>	<b><u>1446,29</u></b>	<b>1494,63</b>
Czas przedsięwzięcia $C_{max}$ [dni robocze]	<b>180</b>	<b><u>180</u></b>	<b>180</b>
Numer iteracji, w której osiągnięto najlepszą wartość $U$	23	17590	18621
Permutacja dla najlepszej wartości $U$	((4,8,9,10,11),(3,12),(6,7,5),(2,1)), ((6,4,9,7,5,12),(8),(3,10,1),(2,11)), (2,1,5),(3,10,7,12),(8,4),(6),(11,9)), (3,8,6,9,7,12),(2),(10,4),(11,1,5)), (2,1,4,12),(8,10,11,9),(3,6,5),(7)), (0),(10,4,7,5),(3,6,11,9,12),(2,8,1)), (1,4),(3,10,9,12),(6),(8,7),(2,11,5)), (1,12),(2,9,7),(8,10,4),(3,6,11,5)), (11,1,4),(12),(8,3,7),(6,2,10,9,5)), (8),(10,6,11,12),(4,5),(2,3,9,1,7)), ((6,10,2,9,12),(1,4,7),(8,11,3,5)))	((8),(6,12),(3,5),(2,4,11,9,10,7,1)), ((6,4,9,7,5,12),(8),(3,10,1),(2,11)), (11,1,5),(3,10,7,12),(8,4),(6),(2,9)), (3,8,11,4,7,12),(2),(6,9,5),(10,1)), (0),(2,11,1,7),(3,8,6,10,9,4,5,12)), (0),(9),(3,2,8,6,10,11,1,4,7,5,12)), (10,4),(3,9,12),(6,1),(8,7),(2,11,5)), (1),(8,9,7),(2,10,4,5),(3,6,11,12)), (0),(5),(0),(2,3,8,6,10,11,9,1,4,7,12)), (1),(5),(3,7),(2,8,6,10,11,9,4,12)), (0),(5),(3,2,8,6,10,11,9,1,4,7,12)))	((2),(12,1,7),(5),(3,11,10,8,9,4,6)), (3,10,4,6,5,12),(2),(1,8,7),(11,9)), (2,4,5),(3,10,7,12),(8),(11,6),(1,9)), (3,2,1,4,5,12),(8),(10,9,7,6),(11)), (1),(8,7,5),(10,4),(3,2,11,9,6,12)), (0),(3,2,1,9),(11,4,6),(8,10,7,5,12)), (10,9),(3,1),(8,7,12),(2,4,6),(11,5)), (2,1,4,6),(11,9,10),(3,7,5),(8,12)), (1,7,6),(5,12),(2),(11,3,8,4,9,10)), (3,1,4,9,10,6),(8,12),(2,5),(11,7)), (4,2,7,9,5),(11,1,8,10,6),(3,12)))

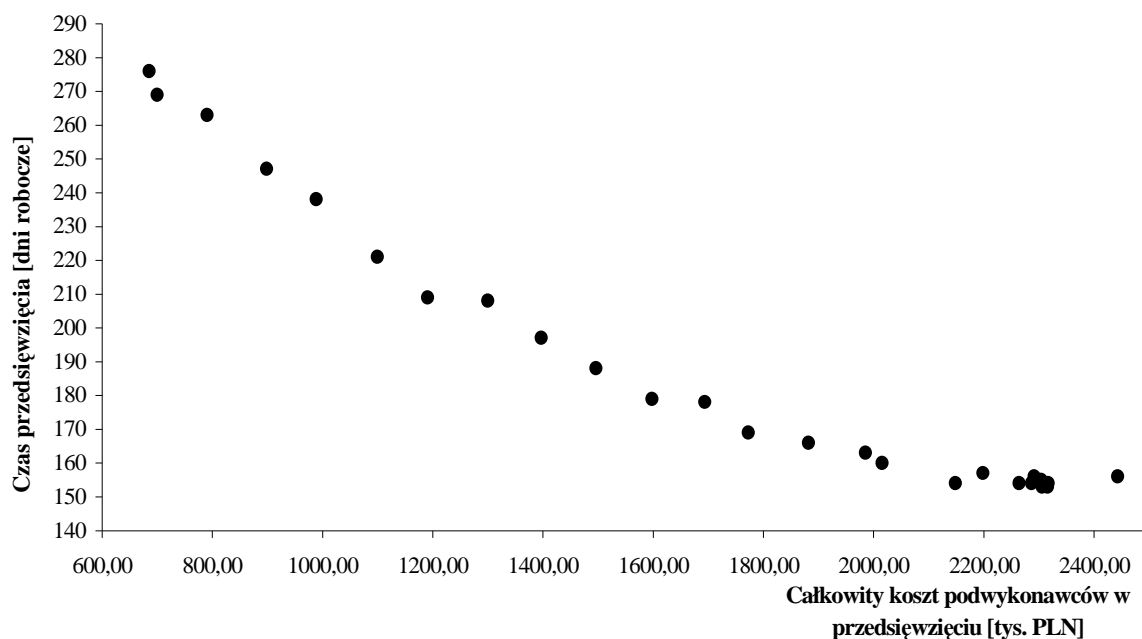
Komentarza wymaga sposób doboru rozwiązań startowych w obu rozpatrywanych przypadkach kryteriów. W pierwszym z nich postąpiono tak jak to opisano w p. 5.1.3. W drugim przypadku (minimalizacja kosztu przedsięwzięcia przy warunku ograniczonego czasu jego trwania) jako rozwiązania startowe do algorytmów TS i SA wybrano rozwiązania uzyskane z optymalizacji przedmiotowego przykładu z kryterium czasu trwania przedsięwzięcia przy warunku maksymalnie możliwego kosztu użytych podwykonawców, który wynosi 3 091 470 PLN (dla każdej z prób oddzielnie generowano rozwiązania startowe). Dla algorytmu GS istnieje problem doboru rozwiązań, które znajdują się w populacji początkowej o liczności  $10n = 120$ . W tym przypadku, do tej populacji przyjęto rozwiązania, które uzyskano przy podejściu dwukryterialnym dla rozpatrywanych trzech algorytmów w tym przykładzie (wyniki optymalizacji w tym podejściu opisano poniżej) i spełniające warunek ograniczający czas trwania przedsięwzięcia  $C_{max} = 180$  dni roboczych. Łącznie w populacji początkowej algorytmu GS znalazło się 41 rozwiązań dopuszczalnych. Pozostałe

rozwiązania w populacji zostały uzyskane w sposób losowy i prawie w każdym przypadku były to rozwiązania niedopuszczalne (nie spełniające warunku ograniczenia czasu trwania przedsięwzięcia).

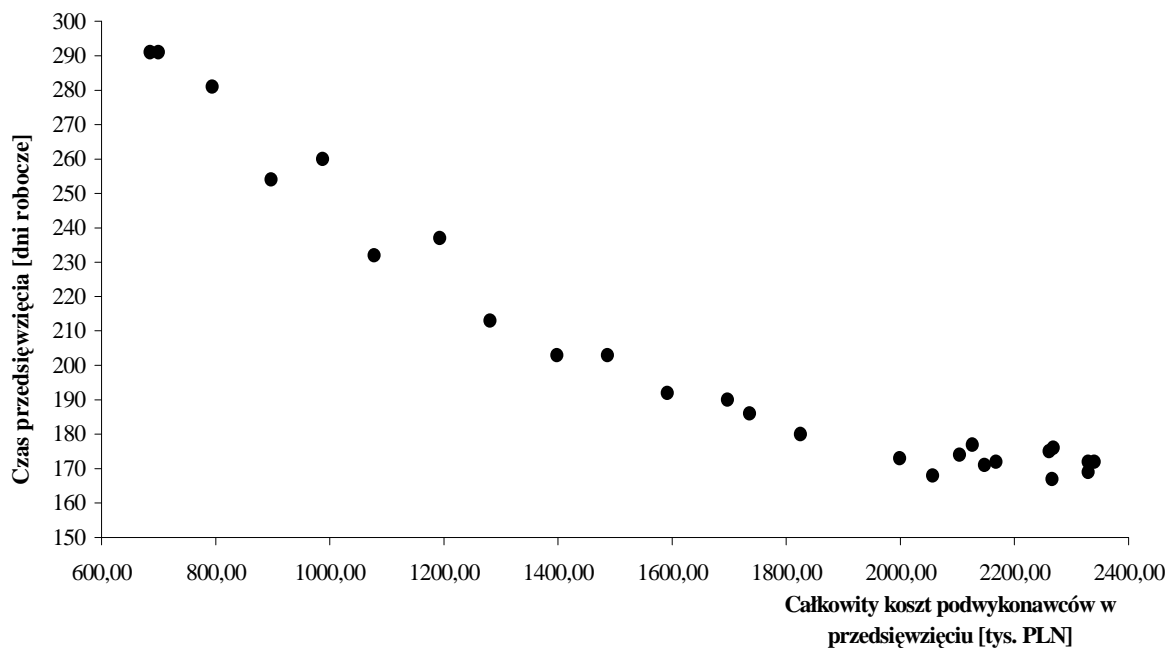
W celu przedstawienia zbioru wszystkich rozwiązań kompromisowych  $K$  dla przedmiotowego przykładu zastosowano podejście dwukryterialne koszt/czas, które opisano w p. 5.1.3. Ze względu na brak odpowiedniego algorytmu, za pomocą którego byłoby możliwe wyznaczenie wszystkich punktów kompromisowych zdecydowano się na wyznaczenie części rozwiązań ze zbioru  $K$  za pomocą algorytmu przedstawionego w p. 5.1.3. W pierwszym kroku tego algorytmu wyznaczono: minimalny koszt użycia podwykonawców w przedsięwzięciu, który wynosi 685 380 PLN oraz maksymalny koszt ich użycia, który wynosi 3 091 470 PLN. Pomędzy tymi wartościami znajdują się wartości kosztów robót podwykonawców dla wszystkich punktów kompromisowych w rozpatrywanym przykładzie przedsięwzięcia. W drugim kroku ustalono wartości progowe kosztów wszystkich robót podwykonawców realizowanych we wszystkich obiektach przedsięwzięcia. Przyjęto, że wykonawca akceptuje wartości progowe tych kosztów ustalone co 100 000 PLN jako wystarczające uszczegółowienie poszukiwanego zbioru rozwiązań kompromisowych  $K$ . Rozwiązanie tego problemu optymalizacyjnego otrzymano za pomocą trzech algorytmów TS, GS i SA. Algorytmy te dostarczą trzy różne przybliżenia zbioru rozwiązań kompromisowych  $K$ . Wartości wszystkich parametrów sterujących w algorytmach przyjęto takie same jak w p. 6.1.2. Ze względu na długi czas obliczeń komputerowych (np. dla algorytmu TS przekraczający kilka godzin dla jednego punktu kompromisowego) obliczenia dla każdego z algorytmów wykonano jednokrotnie. Uzyskane wyniki suboptymalne są zawarte na rysunkach 6.1, 6.2 i 6.3 oraz w tabelach Z.10 – Z.12 przedstawionych w załączniku do pracy.



Rys. 6.1. Zbiór punktów kompromisowych wyznaczony za pomocą algorytmu GS



Rys. 6.2. Zbiór punktów kompromisowych wyznaczony za pomocą algorytmu TS



Rys. 6.3. Zbiór punktów kompromisowych wyznaczony za pomocą algorytmu SA

## 6.2. Harmonogramowanie robót budowlanych na przykładzie zespołu stacji paliw

Celem niniejszego podrozdziału jest weryfikacja modeli sformułowanych w p. 5.2 przez zaprezentowanie możliwości ich zastosowania. Podstawową cechą odróżniającą te modele od modeli z p. 5.1 jest założenie relacji kolejnościowych pomiędzy robotami pojedynczego obiektu wyrażonymi grafem. Przykładem takiego przedsięwzięcia wieloobektowego z tym założeniem jest realizacja zespołu stacji paliw odległych od siebie. W każdej ze stacji, oprócz budynku głównego stacji z wiatą, planuje się również wzniesić

budynki dodatkowe: myjnię samochodową, budynek stacji obsługi samochodów. W związku z tym, w przedsięwzięciu pojawia się możliwość realizowania niektórych robót w sposób równoległy z innymi, na co pozwalają założenia modeli sformułowanych w p. 5.2.

### 6.2.1. Przykład zastosowania modelu harmonogramowania przedstawionego w p. 5.2.1

Wykonawca na zlecenie inwestora ma zrealizować przedsięwzięcie polegające na wybudowaniu  $n = 12$  stacji paliw, które są odległe od siebie. Powoduje to potrzebę uwzględniania dodatkowych czasów transportu pomiędzy poszczególnymi stacjami. Wykonawca zakłada, że zrealizuje przedsięwzięcie tylko własnymi środkami tj. grupami roboczymi, które są wyspecjalizowane do realizacji tylko jednego rodzaju robót i mają stałą wydajność. W przedsięwzięciu każda ze stacji wymaga wykonania  $m = 15$  robót, pomiędzy którymi zachodzą relacje kolejnościowe przedstawione w tabeli 6.11. Na podstawie pracochłonności robót w poszczególnych stacjach oraz składu i wydajności grup roboczych wykonawcy ustalono czasy trwania realizacji robót. Czasy trwania realizacji robót przez grupy robocze są podane w tabeli 6.10. Ponadto, dla każdej z relacji kolejnościowej zachodzą dodatkowe warunki pomiędzy terminami rozpoczęcia lub zakończenia roboty poprzedzającej i następującej (wynikające z technologii robót), co zostało przedstawione w tabeli 6.12. Warunki te mają charakter niezbędnego minimum do spełnienia.

Tabela. 6.10. Czasy trwania robót wykonywanych przez grupy robocze, wyrażone w dniach roboczych

Numer i rodzaj roboty $k =$	Obiekty $j =$											
	1	2	3	4	5	6	7	8	9	10	11	12
1 (zagosp. terenu)	10	13	15	7	19	17	7	8	17	13	4	9
2 (roboty ziemne)	12	18	8	22	10	13	22	9	7	16	13	7
3 (fundamenty)	31	33	43	45	44	17	26	39	22	40	16	17
4 (zbiorniki i frejlity)	5	3	10	7	9	7	6	4	9	10	2	9
5 (sieć wod. – kan.)	50	42	39	17	17	39	45	41	18	49	28	39
6 (sieć elektr.)	25	31	17	10	30	26	29	11	16	9	18	18
7 (roboty murowe)	29	33	36	14	55	57	52	37	56	44	51	13
8 (konstr. stalowe)	15	20	22	29	25	28	10	18	25	10	7	20
9 (pokrycia dachowe)	18	16	10	14	25	9	10	30	31	16	28	29
10 (drogi i plac)	33	27	31	40	25	20	15	32	12	28	20	20
11 (instalacje)	24	43	35	19	37	37	39	42	19	43	12	10
12 (tynkowanie)	29	16	26	20	29	19	31	24	34	20	26	26
13 (posadzki, glazura)	25	39	49	41	41	25	12	25	40	27	33	17
14 (malowanie)	14	11	25	18	22	15	5	5	14	23	13	7
15 (montaż urządzeń)	24	44	32	46	37	31	38	23	10	43	38	20

Tabela. 6.11. Relacje kolejnościowe w rozpatrywanym przykładzie

	Roboty $k =$														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Poprzedniki	start	start	1, 2	3	1, 2	5	3	7	8	8,6,4	9	9,7	11	12	10,13,14
Następniki	5, 3	5, 3	4, 7	10	6	10	8,12	9, 10	11, 12	15	13	14	14, 15, koniec	15	koniec

Tabela. 6.12. Rodzaje relacji kolejnościowych w przykładzie oraz wartości opóźnień w relacjach dla poszczególnych obiektów wyrażone w dniach roboczych

Poprzednik $k =$	Następnik $k =$	Rodzaj relacji	Wartości opóźnień dla poszczególnych obiektów $j =$											
			1	2	3	4	5	6	7	8	9	10	11	12
1	3	RR	5	7	8	4	10	9	4	4	9	7	2	5
1	5	RR	5	7	8	4	10	9	4	4	9	7	2	5
2	3	RR	5	7	3	9	4	5	9	4	3	6	5	3
2	5	RR	5	7	3	9	4	5	9	4	3	6	5	3
3	4	ZZ	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3
3	7	ZR	-3	-3	-4	-5	-4	-2	-3	-4	-2	-4	-2	-2
4	10	ZR	0	0	0	0	0	0	0	0	0	0	0	0
5	6	ZZ	0	0	0	0	0	0	0	0	0	0	0	0
6	10	ZR	0	0	0	0	0	0	0	0	0	0	0	0
7	8	RR	19	21	23	9	36	37	34	24	36	29	33	8
7	12	RR	15	17	18	7	28	29	26	19	28	22	26	7
8	9	RR	5	7	8	10	9	10	4	6	9	4	2	7
8	10	ZR	0	0	0	0	0	0	0	0	0	0	0	0
9	11	RR	7	6	4	6	10	4	4	12	12	6	11	12
9	12	ZZ	-2	-2	-1	-1	-3	-1	-1	-3	-3	-2	-3	-3
10	15	FS	0	0	0	0	0	0	0	0	0	0	0	0
11	13	RR	12	22	18	10	19	19	20	21	10	22	6	5
12	14	ZR	0	0	0	0	0	0	0	0	0	0	0	0
13	14	ZR	0	0	0	0	0	0	0	0	0	0	0	0
13	15	ZR	0	0	0	0	0	0	0	0	0	0	0	0
14	15	RR	6	4	10	7	9	6	2	2	6	9	5	3

W przykładzie przyjęto, że nie występują dodatkowe warunki odnośnie do relacji pomiędzy węzłami: początkowym i końcowym a węzłami reprezentującymi roboty. Dane dotyczące czasów wymaganych ze względu na przemieszczanie grup roboczych między stacjami (zależnych od rodzaju grupy roboczej oraz kolejności realizacji obiektów) są zapisane w postaci 15 macierzy  $S^S_k$  ( $k = 1...15$ ):  $S^S_2 = S^S_3 = S^S_7 = S^S_8 = S^S_{12} = [s^S_{gh}]_{n \times n} = 1$ ,  $S^S_1 = S^S_4 = S^S_5 = S^S_6 = S^S_9 = S^S_{11} = S^S_{13} = S^S_{14} = S^S_{15} = [s^S_{gh}]_{n \times n} = 0$ ,  $S^S_{10} = [s^S_{gh}]_{n \times n} = 2$ . W związku z tak ustaloną strukturą i charakterystyką użytych zasobów (grup roboczych), wykonawca ma możliwość utworzenia optymalnego harmonogramu z uwzględnieniem kryterium czasu



trwania całego przedsięwzięcia  $C_{max}$ , co odpowiada założeniem modelu systemu pracy potokowej przedstawionego w p. 5.2.1.

Problem optymalizacyjny występujący w przedstawionym przykładzie rozwiązano za pomocą algorytmów TS, GS, SA. Forma tych algorytmów została przedstawiona w p. 4.1.4.1. Wartości parametrów sterujących w rozpatrywanych trzech algorytmach przyjęto takie same jak w p. 6.1.1. Obliczenia dla każdego z algorytmów wykonano trzykrotnie. Wyniki obliczeń z najlepszymi wartościami  $C_{max}$ , uzyskane za ich pomocą, są przedstawione w tabeli 6.13.

Tabela. 6.13. Wyniki obliczeń

	Algorytmy optymalizacyjne		
	TS	GS	SA
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>
Najlepsza wartość $C_{max}$ [dni robocze]	<b><u>569</u></b>	<b><u>569</u></b>	<b><u>569</u></b>
Numer iteracji, w której osiągnięto najlepszą wartość $C_{max}$	10	14	38
Permutacja dla najlepszej wartości $C_{max}$	{11,1,8,12,6,4,2,3,5,10,9,7}	{11,2,5,6,12,10,3,1,4,8,9,7}	{11,8,2,4,6,3,1,5,12,10,9,7}

Podobnie jak w przykładzie przedstawionym w p. 6.1.1, algorytmy dość szybko znalazły najlepsze rozwiązania, które są takie same dla każdego z nich. Można również z dużym prawdopodobieństwem przypuszczać, że przez wszystkie trzy algorytmy została znaleziona wartość optymalna. W załączniku został przedstawiony harmonogram realizacji przedmiotowego przedsięwzięcia w postaci sieci zależności (rys. Z.3).

### 6.2.2. Przykład zastosowania modelu harmonogramowania przedstawionego w p. 5.2.2

Dla opisanego powyżej przedsięwzięcia, na które składa się realizacja 12 stacji paliw odległych od siebie, ten sam wykonawca planuje użycie wielu grup roboczych do wykonywania robót jednego rodzaju, ze względu na zbyt długi czas realizacji przedsięwzięcia. (przy założeniu zastosowania tylko jednej grupy roboczej dla robót jednego rodzaju, czas ten wynosi 569 dni roboczych). Sytuacja ta odpowiada założeniom modelu z kryterium czasu trwania przedsięwzięcia  $C_{max}$  przedstawionego w p. 5.2.2. W tym przypadku wykonawca zakłada, że dysponuje następującymi ilościami grup roboczych dla poszczególnych robót, które mają taką samą wydajność i wielkość ( $m = 15$ ):  $k = 1$  (zagospodarowania terenu) – 3 grupy,  $k = 2$  (roboty ziemne) – 3 grupy,  $k = 3$  (fundamenty) –

4 grupy,  $k = 4$  (zbiorniki i frejlity) – 2 grupy,  $k = 5$  (sieć wod. – kan.) – 4 grupy,  $k = 6$  (sieć elektryczna) – 3 grupy,  $k = 7$  (roboty murowe) – 4 grupy,  $k = 8$  (konstrukcje stalowe) – 3 grupy,  $k = 9$  (pokrycia dachowe) – 4 grupy,  $k = 10$  (drogi i plac) – 5 grup,  $k = 11$  (instalacje) – 4 grupy,  $k = 12$  (tynkowanie) – 3 grupy,  $k = 13$  (posadzki, glazura) – 3 grupy,  $k = 14$  (malowanie) – 3 grupy,  $k = 15$  (montaż urządzeń) – 3 grupy. Szczegółowe dane dotyczące czasów trwania robót, relacji kolejnościowych, przyjętego rodzaju i opóźnienia tych relacji, czasów niezbędnych dla transportu grup roboczych są takie same jak przedstawione w p. 6.2.1. W tym przypadku wykonawca ma możliwość utworzenia optymalnego harmonogramu przedsięwzięcia z kryterium czasu jego trwania  $C_{max}$ . Zadanie to będzie rozwiązywane, podobnie jak wcześniej, za pomocą algorytmów TS, GS i SA, których forma jest zaprezentowana w p. 4.1.4.2, a parametry sterujące zostały dobrane tak jak przedstawiono w p. 6.1.2. Obliczenia wykonano dla każdego algorytmu trzykrotnie. Wyniki obliczeń z najlepszymi wartościami  $C_{max}$  są przedstawione w tabeli 6.14.

Tabela. 6.14. Wyniki obliczeń

	Algorytmy optymalizacyjne		
	TS	GS	SA
$I$	2	3	4
Najlepsza wartość $C_{max}$ [dni robocze]	<b><u>264</u></b>	<b>292</b>	<b>286</b>
Numer iteracji, w której osiągnięto najlepszą wartość $C_{max}$	16250	17088	19510
Permutacja dla najlepszej wartości $C_{max}$	{{{11,10,9,7,1},{12,8,5,4},{2,6,3}}, {{{12,10,4,8},{11,6,1,7,9},{2,5,3}}, {{{12,6,3},{2,4,8},{10,1,7},{11,5,9}}, {{{12,11,9,2,4,5},{10,7,6,3,1,8}}, {{{6,4,10},{3,12},{1},{11,5,2,9,8,7}}, {{{3,12,10},{2,6,4},{5,11,1,8,7,9}}, {{{2,4,9},{12,6,3},{11,5,8},{10,1,7}}, {{{12,2,5,7,9},{6,1,3},{11,10,4,8}}, {{{6},{11,1},{12,2,10,4,5,3,7,8},{9}}, {{{2,4,8},{6},{1,3,7},{11},{12,10,5,9}}, {{{12,6,1,3,8},{11,10},{4,9},{2,5,7}}, {{{6,4,10,1,3,8},{11,2,9},{12,5,7}}, {{{2,4,7,8},{12,6,5,9},{11,10,1,3}}, {{{10,1,4,8},{6,12},{11,2,5,7,3,9}}, {{{11,12,10,7,9},{2,5,3},{6,1,4,8}}}	{{{2},{5,1},{7,3,4,8,6,10,9,11,12}}, {{{2},{7},{5,1,8,9,6,3,10,11,12,4}}, {{{2},{5,6,3,4},{1,10},{7,8,9,11,12}}, {{{},{2,3,5,7,1,8,6,4,10,9,11,12}}, {{{12,3},{6,5},{2,7,8},{1,10,11,4,9}}, {{{7,6,9},{11},{2,1,5,10,8,3,12,4}}, {{{7,1,11},{2,8,3,9},{6,12},{5,10,4}}, {{{1,10},{7,5,3,12,9},{2,8,6,11,4}}, {{{12},{5,11},{7,8,10,1,4,9},{2,6,3}}, {{{1},{2,5,11},{7,12},{8,3,9},{6,10,4}}, {{{2,6,3,12,9},{1},{5,10},{7,8,11,4}}, {{{5,11,9,1},{8},{7,2,6,10,3,12,4}}, {{{5,11,12,1},{7,8,10,4,9},{2,6,3}}, {{{2,5},{8,11,12,3,1,9},{7,6,10,4}}, {{{5,6,12,1},{2,8,10,3,9},{7,11,4}}}	{{{6,12,8,11,10},{2},{1,3,4,7,5,9}}, {{{6,12,11},{2,4},{1,3,8,10,7,5,9}}, {{{12,11,10},{1,8,5,9},{2,3,7},{6,4}}, {{{12,6,4,10,11,1,2,8},{3,7,5,9}}, {{{12,3,10,5},{11,2,9},{6},{8,4,7,1}}, {{{12,4,1,5,9},{6,2},{3,11,8,10,7}}, {{{6,10},{2,8,7,9},{1,4,3,5},{12,11}}, {{{6,11,3,10,7,5,9},{2,12,8},{1,4}}, {{{2},{1,4,11},{5,7,9},{6,8,10,3,12}}, {{{11,2},{12,8},{4,7},{3,10},{6,1,5,9}}, {{{11,3},{2,5,9},{4,1,10,12,7},{6,8}}, {{{1,4,5,12},{10,9},{2,6,8,11,3,7}}, {{{4,1,12},{2,8,10,5},{6,11,3,7,9}}, {{{8},{10,12,5,9},{6,2,4,11,1,3,7}}, {{{6,2,1,7},{4,10,3,9},{8,11,12,5}}}

Wyniki obliczeń również potwierdzają wnioski z analizy eksperymentalnej przeprowadzonej w rozdziale 4. Najlepsze wyniki dostarczył algorytm TS, a gorsze SA i GS, co miało miejsce przy braku dodatkowych warunków ograniczających przeszukiwanie

otoczenia  $N_x$  w algorytmie TS. W załączniku został przedstawiony harmonogram realizacji przedmiotowego przedsięwzięcia w postaci sieci zależności (rys. Z.4).

### 6.2.3. Przykład zastosowania modelu harmonogramowania przedstawionego w p. 5.2.3

W przedstawionym powyżej przedsięwzięciu wykonawca decyduje się na zastosowanie własnych grup roboczych o takich samych albo różnych wydajnościach lub składach. Ich liczba oraz specjalizacja będzie jednak ograniczona, co powoduje potrzebę użycia grup roboczych spoza zasobów wykonawcy, czyli podwykonawców. Będzie to wiązało się z poniesieniem przez niego dodatkowych kosztów zależnych od wyboru grupy roboczej do wykonywania roboty w danym obiekcie. Zakłada się, podobnie jak w przykładzie w p. 6.1.3, że pracownicy wchodzący w skład grup roboczych wykonawcy są zatrudnieni na stałe (np. na umowę o pracę) i dla nich nie wyznacza się kosztów ich użycia, gdyż wykonawca zawsze ponosi ten sam koszt ich zatrudnienia. Podwykonawcy składają wykonawcy oferty na realizację robót, które zawierają czas trwania danej roboty i jej koszt. Na podstawie pracochłonności robót w poszczególnych obiektach oraz wydajności własnych grup roboczych wykonawca ustalił czasy trwania realizacji robót w obiektach. Dane dotyczące czasów trwania robót i ich kosztów dla podwykonawców są zawarte w tabelach 6.15. i 6.16. Między robotami dla pojedynczej stacji zachodzą relacje kolejnościowe, które są takie same jak przedstawione w przykładach w p. 6.2.1 i p. 6.2.2. Ich rodzaj oraz wartości opóźnień dla rozpatrywanego przykładu są podane w tabeli 6.17. Dodatkowe czasy transportu grup roboczych pomiędzy obiektami przedsięwzięcia zostały przyjęte jak w p. 6.2.1.

Tabela 6.15. Czasy trwania robót  $k = 1, 4, 5, 6, 15$  wykonywanych przez zespoły grup roboczych wykonawcy składających się z grup roboczych o stałych składach i wydajnościach

Roboty $k =$	Liczba grup wykonawcy	Obiekty $j =$											
		1	2	3	4	5	6	7	8	9	10	11	12
1 (zagosp. terenu)	3	10	13	15	7	19	17	7	8	17	13	4	9
4 (zbiorniki i frejlity)	2	5	3	10	7	9	7	6	4	9	10	2	9
5 (sieć wod. – kan.)	5	50	42	39	17	17	39	45	41	18	49	28	39
6 (sieć elektr.)	4	25	31	17	10	30	26	29	11	16	9	18	18
15 (montaż urządzeń)	4	24	44	32	46	37	31	38	23	10	43	38	20



Tabela 6.16. cd.

Numer grupy roboczej dla poszczególnych rodzajów robót		Obiekty $j =$											
		1	2	3	4	5	6	7	8	9	10	11	12
<b>Pokrycia dachowe (<math>k = 9</math>)</b>													
1	Czas [dni robocze]	18	16	10	14	25	9	10	30	31	16	28	29
	Koszt [tys. PLN]	-	-	-	-	-	-	-	-	-	-	-	-
2	Czas [dni robocze]	23	14	9	13	20	8	11	34	27	15	27	28
	Koszt [tys. PLN]	35,4	45,8	27,6	40,9	77,9	24,0	24,2	67,1	90,2	42,6	74,7	73,6
3	Czas [dni robocze]	19	13	9	12	28	7	12	38	26	17	26	26
	Koszt [tys. PLN]	44,4	47,9	25,5	41,1	57,3	28,2	21,0	59,9	92,3	38,9	76,1	82,3
4	Czas [dni robocze]	20	18	8	17	31	10	12	39	32	15	33	30
	Koszt [tys. PLN]	44,0	36,7	31,0	29,2	51,6	21,9	22,2	59,0	77,5	42,3	59,3	72,2
<b>Drogi i plac (<math>k = 10</math>)</b>													
1	Czas [dni robocze]	33	27	31	40	25	20	15	32	12	28	20	20
	Koszt [tys. PLN]	-	-	-	-	-	-	-	-	-	-	-	-
2	Czas [dni robocze]	41	34	36	42	21	19	12	27	13	32	20	19
	Koszt [tys. PLN]	-	-	-	-	-	-	-	-	-	-	-	-
3	Czas [dni robocze]	34	24	40	41	33	17	15	35	14	34	24	21
	Koszt [tys. PLN]	302,8	286,4	230,6	374,2	180,4	227,8	146,1	275,8	100,8	216,7	160,4	180,6
4	Czas [dni robocze]	27	36	33	48	29	17	19	30	12	23	27	18
	Koszt [tys. PLN]	379,8	194,6	278,2	316,7	206,6	217,7	109,7	323,5	114,2	324,8	142,8	207,2
5	Czas [dni robocze]	31	23	36	36	21	18	14	31	13	28	22	20
	Koszt [tys. PLN]	319,5	306,0	252,6	416,9	278,3	212,5	153,7	312,9	103,3	262,5	170,1	189,8
6	Czas [dni robocze]	32	32	31	46	32	23	13	37	12	31	26	19
	Koszt [tys. PLN]	318,2	216,0	293,8	332,4	188,6	168,7	163,3	263,2	117,4	237,6	145,3	195,6
<b>Instalacje (<math>k = 11</math>)</b>													
1	Czas [dni robocze]	24	43	35	19	37	37	39	42	19	43	12	10
	Koszt [tys. PLN]	-	-	-	-	-	-	-	-	-	-	-	-
2	Czas [dni robocze]	19	34	28	15	30	30	31	34	15	34	10	8
	Koszt [tys. PLN]	-	-	-	-	-	-	-	-	-	-	-	-
3	Czas [dni robocze]	29	52	42	23	44	44	47	50	23	52	14	12
	Koszt [tys. PLN]	-	-	-	-	-	-	-	-	-	-	-	-
4	Czas [dni robocze]	17	30	25	13	26	26	27	29	13	30	8	7
	Koszt [tys. PLN]	-	-	-	-	-	-	-	-	-	-	-	-
<b>Tynkowanie (<math>k = 12</math>)</b>													
1	Czas [dni robocze]	29	16	26	20	29	19	31	24	34	20	26	26
	Koszt [tys. PLN]	-	-	-	-	-	-	-	-	-	-	-	-
2	Czas [dni robocze]	24	15	25	21	28	23	29	21	38	24	25	22
	Koszt [tys. PLN]	-	-	-	-	-	-	-	-	-	-	-	-
3	Czas [dni robocze]	33	15	24	19	27	21	25	24	36	18	22	23
	Koszt [tys. PLN]	67,7	46,1	73,8	54,6	83,8	46,3	101,5	65,2	86,7	59,9	81,7	78,0
4	Czas [dni robocze]	25	15	30	18	30	15	27	25	44	22	25	22
	Koszt [tys. PLN]	89,5	44,7	61,0	60,5	75,3	62,6	95,9	64,5	70,3	48,2	72,9	80,2
5	Czas [dni robocze]	34	13	23	23	24	17	30	23	31	21	24	28
	Koszt [tys. PLN]	65,6	50,6	79,2	47,3	92,2	55,5	83,6	65,3	101,0	51,7	70,8	64,6

Tabela 6.16. cd.

Numer grupy roboczej dla poszczególnych rodzajów robót		Obiekty $j =$											
		1	2	3	4	5	6	7	8	9	10	11	12
Posadzki, glazura ( $k = 13$ )													
1	Czas [dni robocze]	25	39	49	41	41	25	12	25	40	27	33	17
	Koszt [tys. PLN]	-	-	-	-	-	-	-	-	-	-	-	-
2	Czas [dni robocze]	21	36	48	49	37	29	14	29	36	26	32	19
	Koszt [tys. PLN]	77,6	107,1	126,2	88,6	116,3	55,3	26,5	56,0	113,4	73,1	87,0	39,9
3	Czas [dni robocze]	29	31	63	37	43	24	12	21	44	29	34	17
	Koszt [tys. PLN]	56,1	124,3	97,2	116,6	99,0	67,7	30,3	75,7	92,5	63,7	81,9	42,3
4	Czas [dni robocze]	24	37	42	46	39	25	10	30	43	23	41	18
	Koszt [tys. PLN]	67,7	105,1	147,9	93,3	110,4	66,8	36,6	54,0	94,4	82,4	68,3	41,7
Malowanie ( $k = 14$ )													
1	Czas [dni robocze]	14	11	25	18	22	15	5	5	14	23	13	7
	Koszt [tys. PLN]	-	-	-	-	-	-	-	-	-	-	-	-
2	Czas [dni robocze]	12	11	22	19	19	18	4	6	16	29	17	5
	Koszt [tys. PLN]	12,5	8,4	22,0	13,2	20,3	9,9	7,8	9,1	9,6	14,3	11,2	8,8
3	Czas [dni robocze]	16	10	23	21	26	13	6	4	13	20	12	6
	Koszt [tys. PLN]	9,6	10,6	22,3	12,3	14,8	13,1	9,7	10,6	11,7	21,0	10,7	9,9

Tabela 6.17. Relacje kolejnościowe i wartości opóźnień występujące w przykładzie

Poprzednik $k =$	Następnik $k =$	Rodzaj relacji	Wartość opóźnienia
1	3	RR	$0.5p_{j1i}$
1	5	RR	$0.5p_{j1i}$
2	3	RR	$0.4p_{j2i}$
2	5	RR	$0.4p_{j2i}$
3	4	ZZ	-3
3	7	ZR	$-0.1p_{j3i}$
4	10	ZR	0
5	6	ZZ	0
6	10	ZR	0
7	8	RR	$0.65p_{j7i}$
7	12	RR	$0.5p_{j7i}$
8	9	RR	$0.35p_{j8i}$
8	10	ZR	0
9	11	RR	$0.4p_{j9i}$
9	12	ZZ	$-0.1p_{j9i}$
10	15	ZR	0
11	13	RR	$0.5p_{j11i}$
12	14	ZR	0
13	14	ZR	0
13	15	ZR	0
14	15	RR	$0.4p_{j14i}$

W pierwszym przypadku wykonawca ma do dyspozycji w budżecie tej inwestycji dla podwykonawców kwotę 3 000 0000 PLN. Planista wykonawcy ma za zadanie ustalić taki harmonogram wykonywania robót przedsięwzięcia, aby zminimalizować czas jego trwania oraz uwzględnić przyjęte ograniczenie kosztu podwykonawców. Zadanie to odpowiada założeniom modelu przedstawionego w p. 5.2.3 z kryterium czasu trwania przedsięwzięcia  $C_{max}$  przy ograniczeniu jego kosztu  $\hat{U}$  wynoszącym 3 000 0000 PLN. Wartości wszystkich parametrów sterujących w rozpatrywanych trzech algorytmach metaheurystycznych TS, GS i SA przyjęto takie same jak w p. 6.2.2. Podobnie jak poprzednio, obliczenia dla każdego z algorytmów wykonano trzykrotnie. Uzyskane wyniki suboptymalne są zawarte w tabeli 6.18.

Tabela 6.18. Wyniki obliczeń optymalizacyjnych dla rozpatrywanego przykładu z uwzględnieniem kryterium czasu trwania przedsięwzięcia  $C_{max}$  przy ograniczeniu jego kosztu  $\hat{U} = 3\,000\,0000$  PLN

	Algorytmy optymalizacyjne		
	TS	GS	SA
<i>l</i>	2	3	4
Ograniczenie $\hat{U}$ na koszt podwykonawców [tys. PLN]	3000		
Koszt podwykonawców $U$ [tys. PLN]	<b><u>2998,80</u></b>	<b>2993,90</b>	<b>2985,00</b>
Minimalny czas przedsięwzięcia $C_{max}$ [dni robocze]	<b><u>340</u></b>	<b>386</b>	<b>367</b>
Numer iteracji, w której osiągnięto najlepszą wartość $C_{max}$	11929	17117	18823
Permutacja dla najlepszej wartości $C_{max}$	((2,3,5),(1,6,11,9),(4,8,12,7,10)), (3,10,5,9),(2,12,8,11),(1,4,6,7)), (4),(2,10,5),(1,9,3),(12,11,8),(6,7)), ((2,7,4,12,10,1),(9,11,8,6,3,5)), (5,8),(2,9,4),(1,11),(10,6),(12,7,3)), (9,12,3),(4),(2,8,10,6,11),(1,7,5)), (2,6,3,11,9),(1,8,5),(4,12,7),(10)), (4,6,3),(12,10,8,11),(1,2,7,5,9)), (1,4,2,12,10,8,5,11,6,9),0,(3),(7)), (10,3,12,5,6,9),(2,7,4,8,1,11),0,0,0,0)), (2,5,8),(1,4,9),(10,7),(12,3,11,6)),	((4),(3,12),(11,5,6,8,1,2,7,9,10)), (0),(11,5,2,8),(3,12,6,4,1,7,9,10)), (0),(5),(3,1,4),(11,6,8,2),(12,7,9,10)), (0),(1,2,11,8,12,10,6,5,9,4,7,3)), (0,0),(10,2),(3,8,1),(5,4,11,9,12,6,7)), (0),(9,7,2),(11),(3,5,8,12,6,4,10,1)), (11,5,4,2,7,1),(8,9),(12),(3,6,10)), (11,3,9,2),(5,8,7),(12,6,4,10,1)), (12,3,11,5,6,4,8,9,2,1),(7),(10),0)), (6,9,5,4,2),(8,11,10,7,1,3,12),0,0,0,0)), (11,3,4,9,10),(12,8),(6,2),(5,7,1)),	((1,4,12,7),(6,5,8,10,9),(3,2,11)), (6,12,3),(11,2,4,9,10),(8,1,5,7)), (1,2,4),(11),(6,9,10),(12,8),(3,7,5)), (7,8,11,1,10,2,6),(9,4,5,12,3)), (4,2,10),(5),(7,8,9,12),(1,3),(11,6)), (3,4,7,11),0,(2,6,1,8),(9,5,12,10)), (6,2,3,5,10,9),(8,4,7),(12,11),(1)), (1),(11,4,5,10),(12,6,2,8,3,7,9)), (1,12,6,2,4,8,3,7,5,10,9),0,0,(11)), (1,3,2,5,8,9),(6,4,12,7,11,10),0,0,0,0)), (11,12,4,8,7),(2,10,9),(1),(6,3,5)),

W drugim przypadku wykonawca musi spełnić dyrektywny czas trwania przedsięwzięcia narzucony przez inwestora i wynoszący 300 dni roboczych. Planista wykonawcy ma możliwość ustalić taki harmonogram wykonywania robót przedsięwzięcia,

aby zminimalizować koszt użytych zasobów zewnętrznych – podwykonawców, przy ograniczeniu czasu jego trwania. Zadanie to odpowiada założeniom modelu przedstawionego

w p. 5.2.3 z kryterium kosztu realizacji przedsięwzięcia  $U = \sum_{j=1}^n \sum_{k=1}^m u_{jk}$ , przy ograniczeniu

czasu jego trwania  $\hat{C}$ . Wartości wszystkich parametrów sterujących w rozpatrywanych trzech algorytmach TS, GS i SA przyjęto takie same jak w p. 6.2.2. Wyniki suboptymalne, które są uzyskane po trzykrotnym zastosowaniu każdego z algorytmów, przedstawiono w tabeli 6.19.

Tabela 6.19. Wyniki obliczeń optymalizacyjnych dla rozpatrywanego przykładu realizacji

przedsięwzięcia  $U = \sum_{j=1}^n \sum_{k=1}^m u_{jk}$ , przy ograniczeniu czasu jego trwania  $\hat{C} = 300$  dni roboczych

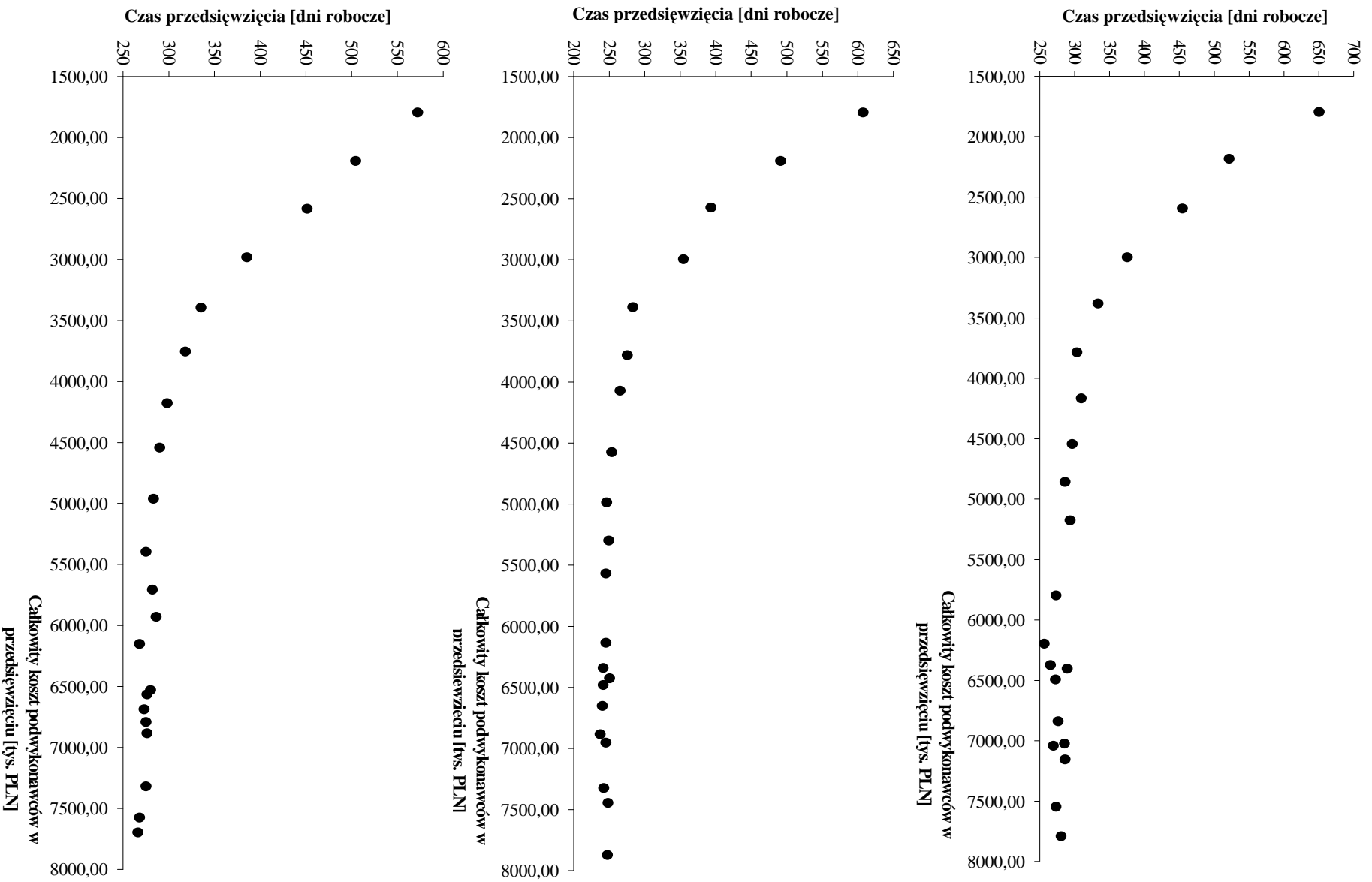
	Algorytmy optymalizacyjne		
	TS	GS	SA
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>
Ograniczenie $\hat{C}$ na czas realizacji przedsięwzięcia [dni robocze]	300		
Minimalny koszt podwykonawców $U$ [tys. PLN]	<b>3163,80</b>	<b>3103,50</b>	<b><u>3071,90</u></b>
Czas przedsięwzięcia $C_{max}$ [dni robocze]	<b>300</b>	<b>300</b>	<b><u>300</u></b>
Numer iteracji, w której osiągnięto najlepszą wartość $U$	191	15545	18401
Permutacja dla najlepszej wartości $U$	((8,10,12),(4,11,5,9,1),(6,2,7,3)), ((6,10,5,12),(8,2,3,9,11),(4,1,7)), ((4,12),(6,3),(5,9),(10,8,1),(2,11,7)), ((4,2,9,11),(5,6,8,10,12,7,1,3)), ((6,4,9,5,12),(7,1),(8,2),(10,11,3),0), ((5,10,3,11,7),(4,6),(12,9),(2,1,8)), ((6,2,11,7),(5,8),(12,9),(4,10,3,1)), ((12,3,9),(6,10,5,11,1,7),(4,2,8)), ((4,12,2,5,3,8,1,7,9),(6),(10),(11)), ((4,2,10,11,1,8),(6,12,5,3,7,9),0,0,0,0), ((6,2),(4,5,1,7),(8),(10,12,3,11,9)),	((0,(4,10,2),(6,5,1,7,11,12,3,8,9)), ((6,10,5,12),(2,3,8,11,9),(4,1,7)), ((11),(5),(4,12,3,9),(2,1,8),(6,10,7)), ((0,(2,4,1,5,6,10,7,11,12,3,8,9)), ((0,0,0),(10,2,1,3),(4,6,5,12,11,8,7,9)), ((3),0,(6,2,1),(10,4,12,5,11,8,7,9)), ((6,12,2,3,7),(5,9),(11,8),(4,10,1)), ((4,10,2,7),(6,5,11,8),(12,1,3,9)), ((4,6,12,2,5,3,11,1,9),0,(10),(8,7)), ((10,2,6,1,8),(4,12,5,11,3,7,9),0,0,0,0), ((4,5,1),(11,9),(10,8),(6,12,2,3,7)),	((12,11,5,8,2),(6),(4,1,10,9,3,7)), ((6,10,5),(11,9,8,2,3),(12,1,4,7)), ((11,4,7),(5),(12,10,8,9),(6,1,2),(3)), ((3,1,10,7,9,8),(12,4,5,11,2,6)), ((10,7,2),(6,12,4,11),0,(1,9,8),(3,5)), ((7,10,2),(12,6,1,9),(11,8,5,3),(4)), ((6,4,2,8,7),(12,10,3,9),(11,5),(1)), ((6),(12,11,4,5,9),(1,10,2,3,8,7)), ((12,6,11,4,10,5,2,8,7,9),(1),(3),0), ((6,11,10,1,3),(12,4,5,2,7,8,9),0,0,0,0), ((12,6,5,7),(4,3,8,9),(1,2),(11,10)),

Problem doboru wartości startowych rozwiązano w podobny sposób jak w przykładzie w p. 6.1.3. W pierwszym przypadku postąpiono tak jak to opisano w p. 5.1.3. W drugim z nich (minimalizacja kosztu przedsięwzięcia, przy warunku ograniczonego czasu jego trwania), dla algorytmów TS i SA jako rozwiązania startowe przyjęto rozwiązania uzyskane z optymalizacji przedmiotowego przykładu z kryterium czasu trwania przedsięwzięcia, przy



warunku maksymalnie możliwego kosztu użytych podwykonawców, który wynosi 9 897 000 PLN (dla każdej z prób oddzielnie generowano rozwiązania startowe). W algorytmie GS do populacji startowej o licznosci  $10n = 120$  rozwiązań, podobnie jak w przykładzie w p. 6.1.3, przyjęto te rozwiązania, które uzyskano z rozwiązań przy podejściu dwukryterialnym dla rozpatrywanych trzech algorytmów (wyniki optymalizacji opisano poniżej). Rozwiązania te powinny spełniać warunek ograniczający czas trwania przedsięwzięcia  $C_{max} = 300$  dni roboczych. Łącznie w populacji początkowej algorytmu GS znalazło się 46 rozwiązań dopuszczalnych. Pozostałe rozwiązania w populacji zostały uzyskane w sposób losowy i prawie w każdym przypadku były to rozwiązania niedopuszczalne.

Wykonawca przedsięwzięcia może być zainteresowany uzyskaniem informacji o zbiorze wszystkich rozwiązań kompromisowych  $K$ , które może otrzymać stosując podejście dwukryterialne koszt/czas. Podejście to opisano w p. 5.2.3. Ze względu na brak odpowiedniego algorytmu, za pomocą którego byłoby możliwe wyznaczenie wszystkich punktów kompromisowych, wykonawca ma możliwość wyznaczenia części rozwiązań ze zbioru  $K$ , odpowiadające przyjętym przez niego wartościom kosztu, za pomocą algorytmu przedstawionego w p. 5.1.3. W pierwszym kroku tego algorytmu wyznaczono: minimalny koszt użycia podwykonawców w przedsięwzięciu, który wynosi 1 795 300 PLN oraz maksymalny koszt ich użycia, który wynosi 9 897 000 PLN. Przyjęto, że wykonawca akceptuje wartości progowe tych kosztów ustalone co 400 000 PLN jako wystarczające uszczegółowienie poszukiwanego zbioru rozwiązań kompromisowych  $K$ . Podobnie jak w przykładzie w p. 6.1.3, zagadnienie optymalizacyjne rozwiązano za pomocą algorytmów TS, GS i SA. Ze względu na zbyt długi czas obliczeń komputerowych (np. dla algorytmu TS przekraczający jedną dobę dla jednego punktu kompromisowego) obliczenia dla każdego z algorytmów wykonano jednokrotnie. Uzyskane wyniki suboptymalne są zawarte na rysunkach 6.4, 6.5 i 6.6 oraz w tabelach Z.13 – Z.15 przedstawionych w załączniku do pracy.



Rys. 6.4 – 6.6. Zbiory punktów kompromisowych wyznaczone za pomocą algorytmów (od góry): GS, TS i SA

### 6.3. Podsumowanie

W niniejszym rozdziale przedstawiono weryfikację, sformułowanych w rozdziale 5, nowych sześciu modeli harmonogramowania w systemie pracy potokowej przedsięwzięć wieloobektowych przez prezentację i analizę przykładów ich aplikacji. Przykłady te opracowano na podstawie dwóch przedsięwzięć: realizacji kompleksu budynków mieszkalnych oraz kompleksu stacji paliw. Rozpatrywane były kryteria czasu trwania oraz kryteria koszt/czas. Ze względu na stosunkowo niewielkie rozmiary zadań ( $n = 12$  obiektów,  $m = 11$  robót lub  $m = 15$  robót) w przykładach w p. 6.1.1 i p. 6.2.1 wszystkie trzy algorytmy uzyskały taką samą wartość funkcji celu w bardzo wczesnej fazie poszukiwań. Można przypuszczać, że zostały znalezione rozwiązania optymalne. Wyniki optymalizacji uzyskane za pomocą rozważanych w pracy algorytmów TS, GS i SA dla przykładów z kryterium czasu trwania przedsięwzięcia oraz z więcej niż jedną grupą roboczą dla wykonywania roboty jednego rodzaju (p. 6.1.2, p. 6.2.2) potwierdzają wnioski z przeprowadzonej w rozdziale 4 analizy eksperymentalnej tych algorytmów. Najlepsze wyniki dostarczał algorytm TS, najgorsze - algorytm GS, który uzyskał wyniki gorsze od najlepszych o 6,5 i 10,6 % (dla przykładów w p. 6.1.2 i p. 6.2.2).

Podobne wyniki uzyskano rozpatrując kryterium czasu trwania przedsięwzięcia przy ograniczeniu jego kosztu (przykłady w p. 6.1.3, p. 6.2.3). Różnica między najlepszym (algorytm TS) i najgorszym wynikiem (algorytm GS) wyniosła 12,8% i 13,5%. Inne wyniki, (nie potwierdzające rezultatów z rozdziału 4) zostały uzyskane w przykładach dla kryterium kosztu przedsięwzięcia, przy ograniczeniu czasu jego trwania (przykłady w p. 6.1.3, p. 6.2.3). W obu przykładach za pomocą algorytmu TS otrzymano najgorsze wyniki. W p. 6.1.3 z tym kryterium najlepszy wynik uzyskano za pomocą algorytmu GS (różnica 3,0% od najgorszego wyniku). W p. 6.2.3 najlepszy wynik uzyskano za pomocą algorytmu SA (również różnica 3,0% od najgorszego wyniku). Przyczyną takich rezultatów jest prawdopodobnie występowanie tzw. „rozległych dolin” [134] w przeszukiwanym zbiorze rozwiązań dopuszczalnych (obszarów w zbiorze rozwiązań dopuszczalnych, które charakteryzują się zbliżonymi wartościami funkcji celu), który został ponadto silnie ograniczony przez założenie odnośnie do czasu trwania danego przedsięwzięcia. W związku z tym algorytm TS, w formie jaka została zastosowana, może przeszukiwać obszar „rozległej doliny” nie natrafiając na minimum lokalne [98]. Lepsze rozwiązania dostarczał w tym przypadku algorytm GS, który prowadzi swoje poszukiwania za pomocą populacji rozwiązań, co zwiększa możliwość znalezienia minimum lokalnego w „dolinie”.

Uzyskane wyniki poszukiwań zbioru rozwiązań kompromisowych w podejściu dwukryterialnym dla obu przykładów potwierdzają przyjęte w p. 5.1.3 i p. 5.2.3 założenie, że punkty tego zbioru są zbliżone kształtem do wypukłej, malejącej odcinkami krzywej (łamanej), która powstałaby przy założeniu ciągłej funkcji czas/koszt. Cechą takiej krzywej jest brak możliwości przecięcia jej przez styczną poprowadzoną do niej w każdym z punktów tego zbioru, co jest widoczne na przedstawionych wykresach przybliżeń zbiorów punktów kompromisowych. Najlepsze wyniki dostarczał algorytm TS, najgorsze - algorytm GS, co zostało szczegółowo przedstawione w tabelach w załączniku i co jest zgodne z wnioskami z analizy eksperymentalnej.

## 7. Wnioski końcowe

W rozprawie analizuje się problematykę planowania robót w systemie pracy potokowej stosowanym w przedsiębiorstwach budowlanych. W celu wyróżnienia tego rodzaju przedsięwzięć, dla których może być użyty taki system pracy, w rozprawie określono je jako przedsięwzięcia wieloobektowe. Dla nich sformułowano nowe modele pracy potokowej, które zawierają niestosowane do tej pory parametry, ograniczenia i kryteria. W tym celu przeprowadzono analizę wiedzy z zakresu modeli systemu pracy potokowej w przedsiębiorstwach budowlanych (m.in. metod sprzężeń czasowych [50], [94]), z zakresu metod planowania przedsięwzięć typu kompleks operacji oraz wiedzy z zakresu teorii szeregowania zadań, która do tej pory nie była szeroko stosowana w harmonogramowaniu przedsięwzięć wieloobektowych. Problematykę tę przedstawiono w rozdziałach 2 i 3 niniejszej pracy. W rozdziale 5 zostały zidentyfikowane i sformułowano nowe modele planowania w systemie pracy potokowej, w których najważniejsze założenia są następujące:

- wszystkie parametry, ograniczenia i funkcje celu w nich użyte mają charakter deterministyczny,
- występują (alternatywnie): relacje kolejnościowe między robotami w obiektach wyrażone niezmienną dla każdego obiektu sekwencją, wynikającą z ustalonego porządku technologicznego (w ramach tych relacji mogą występować ząbienia lub przerwy technologiczne - sprzężenia między obiektami), relacje kolejnościowe między robotami opisane niezmiennym dla każdego obiektu grafem,
- są stosowane różne struktury zasobów, których ilość jest ograniczona w przedsięwzięciu: jedna grupa robocza dla wykonywania jednego rodzaju robót, wiele takich samych grup roboczych dla wykonywania jednego rodzaju robót, struktury z użyciem takich samych lub różnych grup roboczych wykonawcy oraz użycie podwykonawców, którzy powodują zwiększenie limitu dostępności zasobów,
- występują dodatkowe czasy niezbędne dla przemieszczania grup roboczych między obiektami, zależne od rodzaju grupy roboczej oraz kolejności realizacji obiektów,
- jest stosowany dodatkowy parametr opisujący grupę roboczą wykonującą daną robotę w obiekcie tj. koszt jej użycia w przedsięwzięciu,
- stosowane jest kryterium czasu trwania całego przedsięwzięcia oraz kryteria koszt/czas.

Elementy nowości w pracy, które według wiedzy autora nie występowały dotychczas w badaniach związanych z systemem pracy potokowej, są następujące:

- zastosowanie relacji kolejnościowych między robotami opisanymi niezmiennym dla każdego obiektu grafem, z założeniem wystąpienia danego rodzaju relacji i wartości opóźnienia,
- zastosowanie struktur zasobów z użyciem wielu takich samych lub różnych grup roboczych dla wykonywania jednego rodzaju robót,
- zastosowanie dodatkowych czasów niezbędnych dla przemieszczania grup roboczych między obiektami, zależnych od rodzaju grupy roboczej oraz kolejności realizacji obiektów,
- zastosowanie dodatkowego parametru opisującego grupę roboczą wykonującą daną robotę w obiekcie tj. kosztu jej użycia w przedsięwzięciu,
- zastosowanie kryteriów koszt/czas przedsięwzięcia w związku z zastosowaniem dodatkowego parametru, którym jest koszt realizacji robót.

Zaprezentowane w rozdziale 5 modele są zagadnieniami optymalizacji dyskretnej, które, wg wiedzy autora, nie były dotychczas badane w ramach teorii szeregowania zadań (TSZ). Z przeprowadzonej analizy obecnie stosowanych metod harmonogramowania w systemie pracy potokowej i aktualnych osiągnięć TSZ można wnioskować, że metody te (np. metody sprzężeń czasowych) są najczęściej szczególnymi przypadkami modeli rozważanych w ramach TSZ. W związku z tym identyfikacja oraz formułowanie nowych modeli wymagało również stosowania pojęć i narzędzi związanych z TSZ, co zostało uwzględnione w rozprawie. W utworzonych modelach, w celu umożliwienia rozwiązywania problemu optymalizacyjnego z przyjętą funkcją celu, zastosowano wybrane trzy algorytmy metaheurystyczne: poszukiwania z zakazami, przeszukiwania genetycznego i symulowanego wyżarzania. Wybrano je ze względu na jakość dostarczanych rozwiązań, uniwersalność ich aplikacji do różnych problemów szeregowania zadań, jak również łatwość implementacji programowej. Dla tych algorytmów przeprowadzono weryfikację jakości uzyskiwanych rozwiązań na przykładach analizowanych już w literaturze oraz przykładach z danymi generowanymi losowo, które zostały utworzone dla potrzeb niniejszej pracy (rozdział 4). Wybrane algorytmy nie były stosowane do tej pory w rozwiązywaniu zagadnień optymalizacyjnych w systemie pracy potokowej przedsięwzięć budowlanych. Użycie ich dla sformułowanych modeli wymagało dokonania implementacji programowej, która została przeprowadzona w środowisku Mathematica przez autora niniejszej rozprawy. W związku z tym, na potrzeby prac badawczych opracowano w tym środowisku 30 różnych programów uwzględniających odmienne parametry, ograniczenia i funkcje celu.

Przedstawione modele harmonogramowania są przeznaczone do zastosowania przez wykonawcę w trakcie etapu przygotowywania realizacji przedsięwzięcia np. przygotowania

oferty dla inwestora lub tworzenia optymalnego harmonogramu przez wykonawcę po zawarciu umowy z inwestorem. Są ponadto możliwe do zastosowania w aktualnie wykorzystywanych systemach wykonawczych i poszerzają obszar zastosowań systemu pracy potokowej. W rozdziale 6, w ramach weryfikacji sformułowanych modeli, zaprezentowano przykłady ich użycia w fazie planowania przedsięwzięć wieloobektowych: kompleksu budynków mieszkalnych i kompleksu stacji paliw.

Teza sformułowana we wstępie do rozprawy (p. 1.2) była następująca:

Wykorzystanie elementów teorii szeregowania zadań oraz algorytmów metaheurystycznych do rozwiązywania zagadnień optymalizacji dyskretnej pozwala na powiększenie obszaru zastosowań systemu pracy potokowej w przedsięwzięciach budowlanych, szczególnie wieloobektowych, oraz uwzględnienie w szerszym zakresie rzeczywistych warunków realizacyjnych.

Teza ta została udowodniona przez sformułowanie nowych modeli stosujących elementy teorii szeregowania zadań (rozdział 3 i 5), podanie sposobów rozwiązywania problemów optymalizacji dyskretnej z ich weryfikacją (rozdział 4) oraz weryfikację modeli przez przedstawienie i analizę przykładów ich aplikacji w przedsięwzięciach wieloobektowych (rozdział 6). W modelach tych zostały zastosowane parametry, ograniczenia i kryteria spotykane w praktyce realizacji przedsięwzięć budowlanych. Rozwiązywanie zagadnień optymalizacyjnych, występujących w nowych obszarach zastosowań systemu pracy potokowej, było możliwe dzięki wykorzystaniu modelowania stosowanego w teorii szeregowania zadań (zastosowanie problemów przepływowych) oraz algorytmów metaheurystycznych.

Przedstawione modele planowania w systemie pracy potokowej, z dodatkowymi parametrami, ograniczeniami i kryteriami, uwzględniają w znacznym zakresie złożone sytuacje podczas realizacji przedsięwzięć budowlanych. Zakres stosowania modeli sformułowanych w pracy wynika z następujących przyjętych założeń:

- niezmiennej dostępności do zasobów (grup roboczych, kapitału) oraz niezmiennej dostępności do obiektów w czasie wykonywania przedsięwzięcia,
- braku dodatkowych ograniczeń dotyczących relacji kolejnościowych między realizowanymi obiektami,
- możliwości wprowadzania w harmonogramie zmian terminów rozpoczęcia lub zakończenia robót w obiektach ze względu na np. warunki pogodowe, brak dostępu do zasobów lub obiektów, podczas gdy powinien być on zapewniony, zmiany składów lub wydajności grup roboczych, zmiany limitów dostępności zasobów następujące w trakcie wykonywania przedsięwzięcia itp., zgodnie z zasadami występującymi w metodzie

ścieżki krytycznej (każdy z harmonogramów powstałych na bazie sformułowanych modeli posiada własność występowania ścieżki krytycznej),

- niestosowania niektórych ograniczeń, które były podstawą utworzenia metod sprzężeń czasowych [94] np. założonej ciągłości wykonywania robót jednego rodzaju, ciągłości wykonywania robót w obiekcie,
- uwzględniania w sformułowanych modelach kryterium czasu trwania przedsięwzięcia oraz kryteriów koszt/czas.

Uzyskane w niniejszej pracy rezultaty wskazują na celowość kontynuowania badań modeli stosujących system pracy potokowej, w których byłyby uwzględniane:

- niepewność i ryzyko przy założeniu podejścia deterministycznego wykorzystując np. założenia metody łańcucha krytycznego (*critical chain*) przedstawionej przez Goldratta [32] (sposoby aplikacji tej metody dla przedsięwzięć budowlanych są przedstawione np. w pracach [18], [51], [90], [124]),
- planowanie w warunkach niedeterministycznych,
- ograniczona, podczas realizacji przedsięwzięcia, dostępność do zasobów (grup roboczych, kapitału) i obiektów,
- zmiany dotyczących parametrów lub ograniczeń modelu w czasie realizacji przedsięwzięcia,
- różne zbiory robót dla poszczególnych obiektów w przedsięwzięciach wieloobektowych,
- dodatkowe ograniczenia: ciągłość wykonywania robót jednego rodzaju, ciągłość wykonywania robót w obiekcie,
- inne funkcje celu np. związane z nieterminowością wykonania przedsięwzięcia, karami za nieterminowość realizacji obiektów, brakiem ciągłości pracy grup roboczych, wyrównywaniem poziomu wykorzystania zasobów, jakością realizowanych robót, a ponadto funkcje celu uwzględniające wiele kryteriów.

Celowe jest ponadto dokonanie implementacji programowej formułowanych modeli np. w środowisku C++, które pozwala na kompilację poleceń i procedur znacznie przyspieszającą poszukiwanie rozwiązań rozpatrywanych zadań optymalizacyjnych.



LITERATURA

- [1] ADAMIECKI K., O nauce organizacji, Warszawa, PWE, 1985.
- [2] ADELI H., KARIM A., Scheduling/cost optimisation and neural dynamic model for construction, *Journal of Construction Engineering and Management*, 1997, 123(4), 450-458.
- [3] AFANASJEV V.A, AFANASJEV A.V., Potocnaja organizacija rabot v stroitelstwie, Sankt-Petersburg, 2000.
- [4] ASHOUR S., A branch-and-bound algorithm for flow-shop scheduling problem, *AIIE Transactions* 2, 1970, 172, 176.
- [5] BALAS E., Machine sequencing via disjunctive graphs: An implicit enumeration algorithm, *Operational Research*, 1969, 17, 941-957.
- [6] BLADOWSKI S., Metody sieciowe w planowaniu i organizacji pracy, Warszawa, PWE, 1970.
- [7] BŁAŻEWICZ J., Złożoność obliczeniowa problemów kombinatorycznych, WNT, Warszawa, 1988.
- [8] BOŻEJKO W., WODECKI M., A hybrid evolutionary algorithm for some discrete optimization problems, *IEEE Computer Society*, P2286 ISBN 0-7695-2286-6, 2005.
- [9] BRAH S.A., LOO L.L., Heuristics for scheduling in a flow shop with multiple processors, *European Journal of Operational Research*, 1999, 113, 113-122.
- [10] BROWN A.P.G., ŁOMNICKI Z.S., Some applications of the branch-and-bound algorithm for the machine scheduling problem. *Operations Research Quarterly*, 1966, 17, 173-189.
- [11] BURNS S.A., LIU L., FENG C-W., The LP/IP hybrid method for construction time-cost trade-off analysis, *Construction Management and Economics*, 1996, 14, 265-276.
- [12] CAMPBELL H.G., DUDEK R.A., SMITH M.L., A heuristic algorithm for the  $n$  job  $m$  machine sequencing problem, *Management Science*, 1970, 16, 630-637.
- [13] CARR R.I., MEYER W.L., Planning construction of repetitive building units, *Journal of Construction Division*, 1974, 105, 117-128.
- [14] CHRZANOWSKI E.N., JOHNSTON D., Application of linear construction, *Journal of Construction Engineering and Management*, 1986, 112(4), 476-491.
- [15] CHYLIŃSKI P., WOŁCZAŃSKI J., Próba zastosowania algorytmów ewolucyjnych do harmonogramowania realizacji procesów budowlanych, *Konferencja Naukowo-Techniczna Procesy Budowlane 2000, Gliwice-Kokotek, 28 IX - 01 X 2000*.

- [16] CZAPLIŃSKI K., MROZOWICZ J., MUSIAŁ W., Podstawy metodologii projektowania w budownictwie, Wydawnictwo Politechniki Wrocławskiej, Wrocław, 1980.
- [17] CZAPLIŃSKI K., MROZOWICZ J., Realizacja obiektów budowlanych. Podstawy teoretyczne. Wrocław, Wydawnictwo PWr., 1983.
- [18] CZARNIGOWSKA A., JAŚKOWSKI P., SOBOTKA A., Zastosowanie metody łańcucha krytycznego w harmonogramowaniu przedsięwzięć budowlanych. Konferencja Naukowa „Zarządzanie procesami inwestycyjnymi w budownictwie”, Kraków, 2004.
- [19] DANNENBRING D.G., An evaluation of flow-shop sequencing heuristics, *Management Science*, 1977, 23, 1174-1182.
- [20] DING F.Y., KITII CHARTPHAYAK, Heuristics for scheduling flexible flow lines, *Computers Industrial Engineering*, 1994, 26, 27-34.
- [21] DIXON M. (red.), *Project Management. Body of Knowledge*. Association for Project Management, 4 edycja, 2000.
- [22] DYŻEWSKI A., *Technologia i organizacja budowy. Część II: Organizacja i planowanie budowy*, Arkady, Warszawa 1971.
- [23] ELDIN N., SENOUCI A., Scheduling and control of linear projects, *Can. J. Civ. Eng.*, 1994, 21, 219-230.
- [24] ELDIN N., SENOUCI A., Scheduling of linear projects with single loop structures, *Advances in Engineering Software*, 2000, 31, 803-814.
- [25] EL-RAYES K., MOSELHI O., Optimising resource utilization for repetitive construction projects, *Journal of Construction Engineering and Management*, 2001, 127(1), 1-18.
- [26] EL-RAYES K., MOSELHI O., Resource-driven scheduling of repetitive activities, *Construction Management and Economics*, 1998, 16, 433-446.
- [27] FONDAHL J.W., A non-computer approach to the critical path method for the construction industry, 1961, Technical Report No. 9, Department of Civil Engineering, Salford University, CA.
- [28] GAREY M.R., D.S. JOHNSON, R. SETI, The complexity of flowshop and jobshop scheduling, *Mathematics of Operations Research*, 1976, 1, 117-129.
- [29] GLOVER F., Tabu Search, Part I., *ORSA Journal of Computing*, 1989, 1, 190-206.
- [30] GLOVER F., Tabu Search, Part II., *ORSA Journal of Computing*, 1990, 2, 4-32.
- [31] GOLDBERG D.E., *Algorytmy genetyczne i ich zastosowanie*, WNT, Warszawa 1995.

- [32] GOLDRATT E.M., *Critical Chain*, Great Barrington, MA: The North River Press, 1997.
- [33] GRABOWSKI J., A new algorithm of solving the flow-shop problem, *Operations Research in Progress*, D. Reidel Publishing Company, 1982, 57-75.
- [34] GRABOWSKI J., NOWICKI E., SMUTNICKI C., *Metoda blokowa w zagadnieniach szeregowania zadań*, Akademicka Oficyna Wydawnicza EXIT, Warszawa, 2003.
- [35] GRABOWSKI J., WODECKI M., A very fast taboo search algorithm for the permutation flow shop problem with makespan criterion, *Raport PRE 64/2002*, Prace Naukowe Instytutu Cybernetyki Technicznej PWr., 2002.
- [36] GRAHAM R.L., LAWLER E.L., LENSTRA J.K., RINNOOY KAN A.H.G., Optimisation and approximation in deterministic sequencing and scheduling, *Annals of Discrete Mathematics* 1979, 5, 287-326.
- [37] GUPTA J.N.D., A functional heuristic algorithm for the flow-shop scheduling problem, *Operations Research Quarterly*, 1971, 22, 39-47.
- [38] HAMERLINK D.J., ROWINGS J.E., Linear scheduling model: Development of controlling activity path, *Journal of Construction Engineering and Management*, 1998, 124(4), 266-268.
- [39] HARRIS R.B., IOANNOU P.G., Scheduling projects with repeating activities, *Journal of Construction Engineering and Management*, 1998, 124(4), 269-278.
- [40] HEGAZY T., ELHAKEEM A., ELBELTAGI E., Distributed Scheduling Model for Infrastructure Networks, *Journal of Construction Engineering and Management*, 2004, 130(2), 160-167.
- [41] HEGAZY T., ERSAHIN T., Simplified spreadsheet solution. II: Overall schedule optimization, *Journal of Construction Engineering and Management*, 2001, 127(6) , 469-475.
- [42] HEGAZY T., PETZOLD K., Genetic optimization for dynamic project control, *Journal of Construction Engineering and Management*, 2001, 127(6) , 469-475.
- [43] HEGAZY T., WASSEF N., Cost optimisation in projects with repetitive nonserial activities, *Journal of Construction Engineering and Management*, 2001, 127(3), 183-191.
- [44] HEJDUCKI Z., MROZOWICZ J., Harmonizacja potoków z uwzględnieniem ustalonych zasobów (liczby brygad roboczych), *Aktualne problemy naukowo – badawcze budownictwa. XXXII Konferencja Naukowa, Komitet Inżynierii Lądowej i Wodnej PAN i Komitetu Nauki PZiTb*, Krynica, 1986.

- [45] HEJDUCKI Z., MROZOWICZ J., Problemy harmonogramowania realizacji kompleksu obiektów budowlanych, Zeszyty Naukowe Politechniki Śląskiej, seria Automatyka, z. 75, Gliwice 1984, 19-27.
- [46] HEJDUCKI Z., MROZOWICZ J., Stream methods of construction work organization: an introduction to the problem, Engineering Construction and Architectural Management. 2001 (8)2, s. 80-89.
- [47] HEJDUCKI Z., MROZOWICZ J., Ustalenie kolejności w zbiorze potoków obejmujących kompleks obiektów budowlanych, Zeszyty Naukowe Politechniki Śląskiej, seria Automatyka, z. 85, Gliwice 1986, 85-92.
- [48] HEJDUCKI Z., MROZOWICZ J., Zagadnienie synchronizacji kompleksów robót budowlanych, Zeszyty Naukowe Politechniki Śląskiej, seria Automatyka, z. 95, Gliwice 1988, 79-85.
- [49] HEJDUCKI Z., Próba zastosowania potokowych metod organizacji robót do planowania realizacji wielkich kompleksów budowlanych, Praca doktorska, Wrocław, 1985.
- [50] HEJDUCKI Z., ROGALSKA M., Metody sprzężeń czasowych TCM, Przegląd Budowlany 2005 R.76 nr.2 s. 38-45,
- [51] HEJDUCKI Z., ROGALSKA M., Shorting the realization time of building projects with application of theory of constraints and critical chain scheduling, Journal of Civil Engineering and Management, 2004, 10(2), 99-105.
- [52] HEJDUCKI Z., Scheduling model of construction activity with time couplings, Journal of Civil Engineering and Management, 2003, 9(4), s. 284-291.
- [53] HEJDUCKI Z., Sequencing problems in methods of organizing construction processes, Engineering Construction and Architectural Management. 2004, 11(1), s. 20-32.
- [54] HEJDUCKI Z., Sprzężenia czasowe w metodach organizacji złożonych procesów budowlanych, Prace Naukowe Instytutu Budownictwa Politechniki Wrocławskiej. Monografie nr 34, 2000.
- [55] HOLLAND J.H., Adaptation in natural and artificial systems, Ann Arbor, University of Michigan Press, 1975.
- [56] HUNSUCKER J.L., SHAH J.R., Comparative performance analysis of priority rules in a constrained flow shop with multiple processors environment, European Journal of Operational Research, 1994, 72, 102-114.
- [57] IGNALL E., SCHRAGE L., Application of the branch and bound technique to some: flow shop scheduling problem, Operations Research 1956, 13, 400-402.
- [58] IGNASIAK E., Badania operacyjne, PWE, Warszawa, 1998.

- [59] IGNASIAK E., Optymalizacja projektów inwestycyjnych, PWE, Warszawa, 1994.
- [60] IGNASIAK E., Teoria grafów i planowanie sieciowe, PWE, Warszawa, 1982.
- [61] ISHIBUCHI H., MISAKI S., TANAKA H., Modified simulated annealing algorithms for the flow shop sequencing problem, European Journal of Operational Research 1995, 81, 388-398.
- [62] JANIĄK A., Czasowo optymalne sterowanie sekwencją kompleksów operacji niezależnych, Praca doktorska, (nie publikowana), Komunikaty ICT nr 426 i 427, Politechnika Wrocławska, Wrocław, 1976.
- [63] JANIĄK A., Wybrane problemy i algorytmy szeregowania zadań i rozdziału zasobów, Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1999.
- [64] JAŚKOWSKI P., Metoda projektowania struktury systemu wykonawczego przedsięwzięcia budowlanego, Praca doktorska, Lublin, 2003.
- [65] JAWORSKI K.M., Problemy optymalizacyjne w wyrównywaniu wykresów sprawdzających, Konferencja Naukowo-Techniczna: Procesy Budowlane 2000. Realizacja procesów i obiektów budowlanych, Gliwice-Kokotek, 2000.
- [66] JAWORSKI K.M., BOROWIK A., Określenie optymalnej kolejności prowadzenia robót budowlanych przy realizacji wieloobektowego zadania budowlanego, XXXI Konferencja Naukowa KILiW PAN i KN PZiTb, Kraków - Krynica, 1985.
- [67] JAWORSKI K.M., MARCINKOWSKI R., Harmonizacja niejednorodnych procesów technologicznych w planowaniu przedsięwzięć budowlanych, Przegląd Budowlany nr 12/1990.
- [68] JAWORSKI K.M., Metodologia projektowania realizacji budowy, Wydawnictwo Naukowe PWN. Warszawa 1999.
- [69] JOHNSON S.M., Optimal two- and three-stage production schedules with setup times included, Naval Res. Logist. Quart., 1954, 1, 61-68.
- [70] JOHNSTON D.W., Linear scheduling method for highway construction, J. Constr. Div., Am. Soc. Civ. Eng, 1981, 107(2), 247 –261.
- [71] KALABIŃSKI B., MICHNOWSKI Z., Optymalizacja pełnych cykli inwestycyjnych, PWN, Warszawa, 1977.
- [72] KANG L.S., PARK I.C., LEE B.H, Optimal schedule planning for multiple, repetitive construction process, Journal of Construction Engineering and Management, 2001, 127(5), 382-390.
- [73] KARSHANAS S., HABER D., Economic optimization of construction project scheduling, J. Constr. Manage. Econ. E&FN Spon, 1990, 8(2), 135-146.

- [74] KASPROWICZ T., Inżynieria przedsięwzięć budowlanych, Wydawnictwo i Zakład Poligrafii Instytutu Technologii Eksploatacji, Radom – Warszawa, 2002.
- [75] KELLY J.E., Critical path planning and scheduling: mathematical basis, *Operational Research*, 1961, 9(3), 296-320.
- [76] KIRKPATRICK S., GELATT C.D., VECCHI M.P., Optimization by simulated annealing, *Science* 1983, 220, 671-680.
- [77] KOLISCH R., PADMAN R., An integrated survey of deterministic project scheduling, *Omega*, 29, 249-272, 2001.
- [78] KOSECKI A., Kontraktowanie realizacji serii obiektów budowlanych, *Technologia i zarządzanie w budownictwie, Prace Naukowe Instytutu Budownictwa Politechniki Wrocławskiej nr 91, Seria: Studia i Materiały nr 20, Wrocław, 2008.*
- [79] LAGEWEG B.J., J.K., LENSTRA, A.H.G. RINNOOY KAN A.H.G, A General Bounding Scheme for the Permutation Flow-Shop Problem, *Opns. Res.*, 1978, 26, 53-67.
- [80] LEU S.S., HWANG S.T., Optimal repetitive scheduling model with shareable resource constraint, *Journal of Construction Engineering and Management*, 2001, 127(4), 270-280.
- [81] LI H., LOVE P., Using Improved Genetic Algorithms to Facilitate Time-Cost Optimization, *Journal of Construction Engineering and Management*, 1997, 123, 233-237.
- [82] LUMDSEN P., *The line of balance method*, Pergamon Press, London, 1968.
- [83] LUTZ J.D., HIJAZI A., Planning repetitive construction: Current practise, *Construction Management and Engineering*, 1993, 11 , 99-110.
- [84] MAHER M.J., *Równoległo-potokowe metody organizacji robót budowlanych*, Praca doktorska, Wrocław, 1990.
- [85] MARCINKOWSKA E., *Problemy decyzyjne w projektowaniu obiektów i procesów budowlanych*, Monografie PWr., Wrocław, 1986.
- [86] MARCINKOWSKI R., *Harmonogramowanie zadań inżynieryjno-budowlanych według wybranych kryteriów decyzyjnych*, Praca doktorska. WAT, Warszawa 1990.
- [87] MARCINKOWSKI R., JAKUBOWSKI G., *Symulacyjno – analityczne harmonogramowanie przedsięwzięć realizowanych w systemach pracy potokowej*, Konferencja naukowo – techniczna: *Technologia w zarządzanie w budownictwie*, Karłów, 2006.
- [88] MARCINKOWSKI R., *Metody rozdziału zasobów realizatora w działalności inżynieryjno –budowlanej*, Warszawa, WAT, 2002.

- [89] MEYER W.L., SHAFFER L.R., Extensions of the critical path method through the application of integer programming, Civil Engineering Construction Research Series No. 2, University of Illinois, Urbana-Champaign, 1963, II.
- [90] MILIAN Z., Szacowanie buforów czasu w metodzie CCPM. Konferencja Naukowa „Zarządzanie procesami inwestycyjnymi w budownictwie”, Kraków, 2004.
- [91] MOSEHLI O., Schedule comparison using the direct stiffness method, Canadian Journal of Civil Engineering, 1993, 20, 65-72.
- [92] MOSELHI O. EL-RAYES K., Scheduling of repetitive projects with cost optimisation, Journal of Construction Engineering and Management, ASCE, 1993, 119(4), 681-697.
- [93] MOSELHI O., HASSANEIN A., Optimised scheduling of linear projects, Journal of Construction Engineering and Management, 2003, 129(6), 667-673.
- [94] MROZOWICZ J., Metody organizacji procesów budowlanych uwzględniające sprzężenia czasowe, Dolnośląskie Wyd. Edukacyjne, Wrocław, 1997.
- [95] MROZOWICZ J., Potokowe metody organizacji procesów budowlanych o charakterze deterministycznym, Monografia nr 14, Wyd. Politechniki Wrocławskiej, 1982.
- [96] MUSIER R.F.H., EVANS L.B., An approximate method for the production scheduling of industrial bath processes with parallel units, Computers and Chemical Engineering, 1989, 13, 229-238.
- [97] NAWAZ M., ENSCORE Jr E.E., HAM I., A heuristic algorithm for the  $m$ -machine,  $n$ -job flow-shop sequencing problem, 1983, OMEGA 11/1, 91-95.
- [98] NOWICKI E., Metoda tabu w problemach szeregowania zadań produkcyjnych, Prace Naukowe Instytutu Cybernetyki Technicznej PWr., Wyd. Politechniki Wrocławskiej, 1999.
- [99] NOWICKI E., SMUTNICKI C., A fast taboo search algorithm for the permutation flow-shop problem, European Journal of Operational Research, 1996, 91, 160-175.
- [100] NOWICKI E., SMUTNICKI C., The flow shop with parallel machines: A taboo search approach, European Journal of Operational Research, 1998, 106, 226-253.
- [101] NOWICKI K., Organizacja i ekonomika budowy, Politechnika Wrocławska, Wrocław, 1992.
- [102] O'BRIEN J.J., VPM scheduling for high rise buildings, J. Constr. Div., Am. Soc. Civ. Eng., 1975, 101, 895-905.
- [103] OGBU F., SMITH D., The application of the simulated annealing algorithm to the solution of the  $n/m/C_{max}$  flowshop problem, Computers & Operations Research, 1990, 17(3), 243-253.

- [104] OSMAN I., POTTS C., Simulated annealing for permutation flow-shop scheduling, *OMEGA*, 1989, 17(6), 551-557.
- [105] PALMER D.S., Sequencing jobs through a multi-stage process in the minimum total time. A quick method of obtaining near optimum, *Operations Research Quarterly*, 1965, 16, 101-107.
- [106] PAWLAK M., Algorytmy ewolucyjne jako narzędzie harmonogramowania produkcji, Wydawnictwo Naukowe PWN, Warszawa 1999.
- [107] PEKKNY J.F., MILLER D.L., Exact solution of the no-wait flow shop scheduling problem with a comparison to heuristic methods. *Computers and Chemical Engineering*; 1991, 15, 741-748.
- [108] PEMPERA J., Algorytmy szeregowania zadań w pewnym dyskretnym procesie produkcyjnym, Praca doktorska, ICT PWr, Raport serii: Preprinty nr 117/2000, Wrocław, 2000.
- [109] PENNYPACKER J.S., DYE L.D. (red.), *Managing Multiple Projects. PM Practices*. Marcel Dekker, 2002.
- [110] PODOLSKI M., Modelowanie realizacji robót budowlanych z zastosowaniem teorii szeregowania zadań. Konferencja Naukowo – Techniczna „Budownictwo polskie w rok po wstąpieniu do Unii Europejskiej”, Gdańsk, 2005.
- [111] PODOLSKI M., Zagadnienie optymalizacji dyskretnej z kryteriami koszt/czas w planowaniu przedsięwzięć wieloobektowych, *Technologia i zarządzanie w budownictwie*, Prace Naukowe Instytutu Budownictwa Politechniki Wrocławskiej nr 91, Seria: Studia i Materiały nr 20, Wrocław, 2008
- [112] PODOLSKI M., Zastosowanie metaheurystycznych algorytmów szeregowania zadań w harmonogramowaniu wieloobektowych przedsięwzięć budowlanych, Konferencja naukowo – techniczna: *Technologia w zarządzanie w budownictwie*, Karłów, 2006.
- [113] Praca zbiorowa (kier. Z. Michnowski), *Podstawy organizacji, zarządzania i technologii w budownictwie*, Arkady, Warszawa, 1985.
- [114] PRAGER W., A structural method of computing project cost polygons, *Management Science*, 1963, 9(3), 394-404.
- [115] PRYSTUPA M., Modele alokacji zasobów w procesie realizacji obiektów budowlanych, Wydawnictwo Politechniki Warszawskiej, Warszawa, 1988.
- [116] REDA R.M., RPM: Repetitive project modelling, *Journal of Construction Engineering and Management*, 1990, 116, 316-330.
- [117] REDA R.M., Time-cost trade-off modeling of related activities by using mixed integer programming, thesis submitted to the University of Michigan, Ann Arbor, MI, 1986.



- [118] REEVES C., A genetic algorithm for flowshop sequencing problem, *Computer and Operational Research*, 1995, 80, 397-403.
- [119] REEVES C., YAMADA T., Genetic algorithms, path relinking and the flowshop sequencing problem, *Evolutionary Computations*, 1998, 6, 45-60.
- [120] ROBINSON D.R., A dynamic programming solution to time-cost trade-off for CPM, *Management Science*, 1975, 22(2), 158-166.
- [121] ROECH W., Network planning and velocity-diagrams in housing construction, *Proceedings of the Third INTERNET Congress, Book II, Stockholm, Sweden, 1972*, 415-422.
- [122] ROGALSKA M., BOŻEJKO W., HEJDUCKI Z., Harmonogramowanie przedsięwzięć budowlanych z zastosowaniem hybrydowego algorytmu ewolucyjnego, *Technologia i zarządzanie w budownictwie, Prace Naukowe Instytutu Budownictwa Politechniki Wrocławskiej*, nr 18, Wrocław, 2006.
- [123] ROGALSKA M., BOŻEJKO W., HEJDUCKI Z., Time/cost optimization using hybrid evolutionary algorithm in construction project scheduling, *Automation in Construction* (2008), doi: 10.1016/j.autcon.2008.04.002 (w druku).
- [124] ROGALSKA M., HEJDUCKI Z., Time buffers in construction process scheduling, *Journal of Civil Engineering and Management*, 2007, 13(2), s. 143-148.
- [125] ROWIŃSKI L., *Organizacja i ekonomika budownictwa. Cz. I: Organizacja i planowanie w budownictwie*, PWN, Warszawa, 1976.
- [126] ROWIŃSKI L., *Organizacja procesów budowlanych*, PWN, Warszawa, 1979.
- [127] RUSSEL A., CASELTON W., Extensions to linear scheduling optimisation, *Journal of Construction Engineering and Management*, 1988, 114(1), 36-52.
- [128] SAWICKI M., *Sprzężenia czasowe jako podstawa nowych metod organizacji robót*, Praca doktorska, Wrocław, 1997.
- [129] SAWIK T., A scheduling algorithm for flexible flow lines with limited intermediated buffers, *Applied Stochastic Models and Data Analysis* 1993, 9, 127-138.
- [130] SELINGER S., Construction planning for linear projects, *J. Constr. Div., Am. Soc. Civ. Eng.*, 1981, 108(3), 445 –457.
- [131] SENOUCI A., ELDIN N., Dynamic programming approach to scheduling of nonserial linear projects, *Journal of Construction Engineering and Management*, 1996, 10(2), 106-114.
- [132] SHAUKAT A.B., HUNSUCKER J.L., Branch and bound algorithm for the flow shop with multiple processors, *European Journal of Operational Research* 1991, 51 88-99.

- [133] SIEMENS N., A simple CPM time-cost trade-off algorithm, *Management Science*, 1971, 17(6), 354-363.
- [134] SMUTNICKI C., *Algorytmy szeregowania*, Akademicka Oficyna Wydawnicza EXIT, Warszawa, 2002.
- [135] STANISZKIS M., Metoda decydujących ciągów czynności (MDC), *Zeszyty PZiTB*, luty 1965.
- [136] STRADAL O., CACHA J., Time space scheduling method, *J. Constr. Div., Am. Soc. Civ. Eng.*, 1982, 108, 445-457.
- [137] SYSŁO M.M., DEO N., KOWALIK J.S., *Algorytmy optymalizacji dyskretnej z programami w języku Pascal*, PWN, Warszawa, 1993.
- [138] TAILLARD E., Benchmarks for basic scheduling problems, *European Journal of Operational Research*, 1993, 64, 278-285.
- [139] TAILLARD E., Home page, <http://www.eivd.ch/ina/collaborateurs/etd/default.htm>.
- [140] TAILLARD E., Some efficient heuristic methods for flow-shop sequencing problems, *European Journal of Operational Research*, 1990, 47, 65-74.
- [141] TROCKI M., GRUCZA B., OGONEK K., *Zarządzanie projektami*, PWE, Warszawa, 2003.
- [142] WALA K. CHMIEL W., Operatory genetyczne dla permutacyjnych zagadnień optymalizacji. *Zeszyty Naukowe Politechniki Śląskiej, seria Automatyka*, 1996, 84, 223-233, 1995, 116, 67-77.
- [143] WALA K., Algorytm ewolucyjny harmonogramowania procesów wytwarzania w systemach komputerowo zintegrowanych, *Zeszyty Naukowe Politechniki Śląskiej, seria Automatyka*, 1998, 125, 249-259.
- [144] WĘGLARZ J., *Sterowanie w systemach typu kompleks operacji*, PWN, Warszawa-Poznań, 1981.
- [145] WIDMER, M., HERTZ, A., A new heuristic method for the flowshop sequencing problem. *European Journal of Operational Research* 1989, 41, 186-193.
- [146] WITTRUCK R.J., An adaptable scheduling algorithm for flexible flow lines, *Operations Research*, 1988, 36, 445-453.
- [147] YAMIN R.A., HARMELINK D.J., Comparison of Linear Scheduling Model (LSM) and Critical Path Method, *Journal of Construction Engineering and Management*, 2001, 127, 374-381.

# **ZAŁĄCZNIK**



Tabela Z.1. Szczegółowe wyniki obliczeń weryfikacyjnych dla problemu przepływowego (p. 4.2.1 rozprawy) przeprowadzonych za pomocą algorytmu TS dla wybranych przykładów testowych wg [138], [139], o rozmiarach  $n = 20$ ,  $m = 5$

Oznaczenie przykładu wg [139]	Numer iteracji	$C_{max}$ uzyskany w danej iteracji dla próby $i=$							Wartość średnia $C_{max}$	Wartość optymalna $C_{max}$ wg [139]	PRD(TS) [%]
		1	2	3	4	5	6	7			
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>	<i>12</i>
tai001	1000	-	-	-	-	-	-	-			
	5000	-	-	-	-	-	-	-			
	10000	-	-	-	-	-	-	-			
	15000	-	-	-	-	-	-	-			
	$C_{max}$	1278	1278	1278	1278	1278	1278	1278	1278,00	1278	<b>0,00</b>
	$iter_{opt}$	34	754	135	379	167	303	36			
tai002	1000	1360	1360	1360	1360	-	1360	1360			
	5000	1360	1360	1360	1360	-	1360	1360			
	10000	1360	1360	1360	1360	-	1360	1360			
	15000	1360	1360	1360	1360	-	1360	1360			
	$C_{max}$	1360	1360	1360	1360	1359	1360	1360	1359,86	1359	<b>0,06</b>
	$iter_{opt}$	-	-	-	-	35	-	-			
tai003	1000	-	-	-	-	1099	-	-			
	5000	-	-	-	-	-	-	-			
	10000	-	-	-	-	-	-	-			
	15000	-	-	-	-	-	-	-			
	$C_{max}$	1081	1081	1081	1081	1081	1081	1081	1081,00	1081	<b>0,00</b>
	$iter_{opt}$	681	84	215	273	1711	115	41			
tai004	1000	1304	1297	1297	1299	1297	1299	1297			
	5000	-	-	1297	1299	1297	1297	-			
	10000	-	-	-	1297	1297	-	-			
	15000	-	-	-	-	-	-	-			
	$C_{max}$	1293	1293	1293	1293	1293	1293	1293	1293,00	1293	<b>0,00</b>
	$iter_{opt}$	1272	1059	7442	10401	11532	5871	2118			
tai006	1000	1210	1210	1210	-	1210	1210	-			
	5000	-	-	-	-	1210	1210	-			
	10000	-	-	-	-	-	-	-			
	15000	-	-	-	-	-	-	-			
	$C_{max}$	1195	1195	1195	1195	1195	1195	1195	1195,00	1195	<b>0,00</b>
	$iter_{opt}$	1868	1275	1716	374	7980	6571	56			
<b>średni PRD(TS) <math>n = 20</math>, <math>m = 5</math>:</b>											<b><u>0,01</u></b>

$iter_{opt}$  – numer iteracji, w której uzyskano najlepszą wartość  $C_{max}$  w danej próbie  $i$

Tabela Z.2. Szczegółowe wyniki obliczeń weryfikacyjnych dla problemu przepływowego (p. 4.2.1 rozprawy) przeprowadzonych za pomocą algorytmu TS dla wybranych przykładów testowych wg [138], [139], o rozmiarach  $n = 20$ ,  $m = 10$

Oznaczenie przykładu wg [139]	Numer iteracji	$C_{max}$ uzyskany w danej iteracji dla próby $i=$							Wartość średnia $C_{max}$	Wartość optymalna $C_{max}$ wg [139]	PRD(TS) [%]	
		1	2	3	4	5	6	7				
<i>l</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>	<i>12</i>	
tai011	1000	1594	1585	-	-	-	1590	1588				
	5000	-	-	-	-	-	1587	1585				
	10000	-	-	-	-	-	-	1585				
	15000	-	-	-	-	-	-	-				
	$C_{max}$	1582	1582	1582	1582	1582	1582	1582	1582	1582,00	1582	<b>0,00</b>
	$iter_{opt}$	4340	1079	575	578	745	8314	11764				
tai012	1000	1669	1667	1667	1667	1669	1667	1664				
	5000	1664	1667	-	1664	1660	-	1663				
	10000	1664	1664	-	1660	-	-	1660				
	15000	-	1660	-	-	-	-	-				
	$C_{max}$	1659	1659	1659	1659	1659	1659	1659	1659	1659,00	1659	<b>0,00</b>
	$iter_{opt}$	13436	17085	2381	13412	8914	3354	11095				
tai015	1000	-	-	1424	1423	-	1422	1423				
	5000	-	-	1422	-	-	-	1422				
	10000	-	-	-	-	-	-	1420				
	15000	-	-	-	-	-	-	-				
	$C_{max}$	1419	1419	1419	1419	1419	1419	1419	1419	1419,00	1419	<b>0,00</b>
	$iter_{opt}$	285	560	6734	1873	814	2007	12804				
tai017	1000	1486	1487	1488	-	1490	-	1488				
	5000	-	-	-	-	1486	-	-				
	10000	-	-	-	-	-	-	-				
	15000	-	-	-	-	-	-	-				
	$C_{max}$	1484	1484	1484	1484	1484	1484	1484	1484	1484,00	1484	<b>0,00</b>
	$iter_{opt}$	1195	1071	2083	590	6233	881	3509				
tai018	1000	1546	1545	1544	1550	1546	1544	1544				
	5000	1545	1545	-	1548	1545	1544	-				
	10000	1545	1545	-	1545	1544	-	-				
	15000	1544	-	-	1544	-	-	-				
	$C_{max}$	1544	1538	1538	1540	1538	1538	1538	1539,14	1538	<b>0,07</b>	
	$iter_{opt}$	-	13913	1178	-	11810	6917	2778				
<b>średni PRD(TS) <math>n = 20</math>, <math>m = 10</math>:</b>											<b>0,01</b>	

$iter_{opt}$  – numer iteracji, w której uzyskano najlepszą wartość  $C_{max}$  w danej próbie  $i$

Tabela Z.3. Szczegółowe wyniki obliczeń weryfikacyjnych dla problemu przepływowego (p. 4.2.1 rozprawy) przeprowadzonych za pomocą algorytmu TS dla wybranych przykładów testowych wg [138], [139], o rozmiarach  $n = 20$ ,  $m = 20$

Oznaczenie przykładu wg [139]	Numer iteracji	$C_{max}$ uzyskany w danej iteracji dla próby $i=$							Wartość średnia $C_{max}$	Wartość optymalna $C_{max}$ wg [139]	PRD(TS) [%]		
		1	2	3	4	5	6	7					
<i>l</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>	<i>12</i>		
tai022	1000	2105	2101	2105	2105	-	2105	2101			<b>0,03</b>		
	5000	2105	2101	2101	2101	-	2101	2101					
	10000	2100	2101	2101	2101	-	2100	2101					
	15000	2100	2101	2101	-	-	-	2100					
	$C_{max}$	2099	2101	2100	2099	2099	2099	2100				2099,57	2099
	$iter_{opt}$	17438	-	-	12895	755	13514	-					
tai024	1000	2234	2233	-	-	2234	-	2234			<b>0,01</b>		
	5000	-	2224	-	-	2233	-	2224					
	10000	-	2224	-	-	2233	-	2224					
	15000	-	2224	-	-	2224	-	-					
	$C_{max}$	2223	2223	2223	2223	2224	2223	2223				2223,14	2223
	$iter_{opt}$	4002	15622	736	760	-	619	14208					
tai025	1000	2299	2298	2306	2299	2300	2304	2299			<b>0,13</b>		
	5000	2299	2298	2298	-	2298	2298	2296					
	10000	2298	2298	2297	-	2298	2296	-					
	15000	2297	2298	2297	-	2297	-	-					
	$C_{max}$	2296	2296	2297	2291	2296	2291	2291				2294,00	2291
	$iter_{opt}$	-	-	-	4411	-	11839	8617					
tai027	1000	2298	2299	2298	2273	2278	2277	2278			<b>0,41</b>		
	5000	2296	2298	2298	-	2278	-	2278					
	10000	2296	2294	2298	-	-	-	-					
	15000	2296	2294	2294	-	-	-	-					
	$C_{max}$	2296	2294	2294	2273	2273	2273	2273				2282,29	2273
	$iter_{opt}$	-	-	-	4157	9221	1715	7546					
tai029	1000	2242	2242	2240	-	2242	2240	2242			<b>0,03</b>		
	5000	2242	-	2240	-	2242	2238	2240					
	10000	-	-	2240	-	2240	-	2240					
	15000	-	-	2240	-	2238	-	2238					
	$C_{max}$	2237	2237	2240	2237	2238	2237	2237				2237,57	2237
	$iter_{opt}$	5625	2399	-	526	-	8193	17224					
<b>średni PRD(TS) <math>n = 20</math>, <math>m = 20</math>:</b>											<b>0,12</b>		

$iter_{opt}$  – numer iteracji, w której uzyskano najlepszą wartość  $C_{max}$  w danej próbie  $i$

Tabela Z.4. Szczegółowe wyniki obliczeń weryfikacyjnych dla problemu przepływowego (p. 4.2.1 rozprawy) przeprowadzonych za pomocą algorytmu GS dla wybranych przykładów testowych wg [138], [139], o rozmiarach  $n = 20$ ,  $m = 5$

Oznaczenie przykładowg [139]	Numer iteracji	$C_{max}$ uzyskany w danej iteracji dla próby $i=$							Wartość średnia $C_{max}$	Wartość optymalna $C_{max}$ wg [139]	PRD(GS) [%]
		1	2	3	4	5	6	7			
<i>l</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>	<i>12</i>
tai001	1000	1297	1297	1297	1297	1297	1297	1297			
	5000	1297	1297	1297	1297	1297	1297	1297			
	10000	1297	1297	1297	1297	1297	1297	1297			
	15000	1297	1297	1297	1297	1297	1297	1297			
	$C_{max}$	1297	1297	1297	1297	1297	1297	1297	1297,00	1278	<b>1,49</b>
	$iter_{opt}$	-	-	-	-	-	-	-			
tai002	1000	1368	1373	1366	1365	1383	1373	1368			
	5000	1365	1368	1366	1365	1368	1365	1360			
	10000	1365	1366	1366	1365	1366	1365	1360			
	15000	1365	1366	1366	1365	1366	1365	1360			
	$C_{max}$	1365	1366	1366	1365	1360	1365	1360	1363,86	1359	<b>0,36</b>
	$iter_{opt}$	-	-	-	-	-	-	-			
tai003	1000	1098	1098	1128	1098	1100	1132	1098			
	5000	1098	1098	1098	1090	1100	1089	1098			
	10000	1098	1088	1098	1087	1100	1088	1098			
	15000	1098	1088	1098	1087	1098	1088	1098			
	$C_{max}$	1098	1088	1098	1087	1098	1088	1088	1092,14	1081	<b>1,03</b>
	$iter_{opt}$	-	-	-	-	-	-	-			
tai004	1000	1309	1321	1331	1318	1309	1322	1315			
	5000	1309	1312	1315	1300	1309	1318	1315			
	10000	1309	1309	1309	1300	1309	1318	1315			
	15000	1309	1309	1309	1300	1309	1318	1315			
	$C_{max}$	1309	1309	1309	1300	1309	1318	1315	1309,86	1293	<b>1,30</b>
	$iter_{opt}$	-	-	-	-	-	-	-			
tai006	1000	1210	1210	1224	1210	1224	1219	1210			
	5000	1210	1210	1210	1210	1210	1212	1210			
	10000	1210	1210	1210	1210	1210	1210	1210			
	15000	1210	1210	1210	1210	1210	1210	1210			
	$C_{max}$	1210	1210	1210	1210	1210	1210	1210	1210,00	1195	<b>1,26</b>
	$iter_{opt}$	-	-	-	-	-	-	-			
<b>średni PRD(GS) <math>n = 20</math>, <math>m = 5</math>:</b>										<b>1,09</b>	

$iter_{opt}$  – numer iteracji, w której uzyskano najlepszą wartość  $C_{max}$  w danej próbie  $i$



Tabela Z.5. Szczegółowe wyniki obliczeń weryfikacyjnych dla problemu przepływowego (p. 4.2.1 rozprawy) przeprowadzonych za pomocą algorytmu GS dla wybranych przykładów testowych wg [138], [139], o rozmiarach  $n = 20$ ,  $m = 10$

Oznaczenie przykładu wg [139]	Numer iteracji	$C_{max}$ uzyskany w danej iteracji dla próby $i=$							Wartość średnia $C_{max}$	Wartość optymalna $C_{max}$ wg [139]	PRD(GS) [%]
		1	2	3	4	5	6	7			
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>	<i>12</i>
tai011	1000	1628	1609	1625	1623	1644	1613	1622			
	5000	1627	1609	1604	1607	1618	1608	1599			
	10000	1627	1609	1604	1598	1613	1608	1598			
	15000	1627	1609	1604	1598	1613	1604	1598			
	$C_{max}$	1610	1609	1604	1598	1613	1604	1596			
	$iter_{opt}$	-	-	-	-	-	-	-	-		
tai012	1000	1721	1707	1710	1704	1694	1707	1716			
	5000	1704	1704	1701	1679	1691	1707	1714			
	10000	1677	1686	1701	1679	1691	1707	1714			
	15000	1667	1669	1701	1679	1691	1707	1714			
	$C_{max}$	1667	1669	1701	1679	1691	1707	1707			
	$iter_{opt}$	-	-	-	-	-	-	-	-		
tai015	1000	1453	1493	1461	1498	1455	1476	1437			
	5000	1453	1441	1449	1438	1443	1467	1437			
	10000	1448	1426	1438	1438	1443	1466	1437			
	15000	1448	1426	1438	1438	1443	1466	1437			
	$C_{max}$	1448	1426	1438	1432	1443	1466	1437			
	$iter_{opt}$	-	-	-	-	-	-	-	-		
tai017	1000	1499	1550	1536	1526	1510	1503	1515			
	5000	1499	1526	1536	1526	1497	1498	1515			
	10000	1499	1526	1536	1526	1497	1498	1515			
	15000	1499	1507	1511	1526	1497	1498	1515			
	$C_{max}$	1499	1507	1511	1526	1497	1498	1515			
	$iter_{opt}$	-	-	-	-	-	-	-	-		
tai018	1000	1641	1613	1592	1593	1577	1576	1588			
	5000	1571	1612	1580	1592	1574	1566	1552			
	10000	1571	1582	1580	1565	1571	1566	1552			
	15000	1571	1573	1571	1558	1571	1566	1552			
	$C_{max}$	1571	1571	1571	1558	1571	1566	1552			
	$iter_{opt}$	-	-	-	-	-	-	-	-		
<b>średni PRD(GS) <math>n = 20</math>, <math>m = 10</math>:</b>											<b>1,64</b>

$iter_{opt}$  – numer iteracji, w której uzyskano najlepszą wartość  $C_{max}$  w danej próbie  $i$

Tabela Z.6. Szczegółowe wyniki obliczeń weryfikacyjnych dla problemu przepływowego (p. 4.2.1 rozprawy) przeprowadzonych za pomocą algorytmu GS dla wybranych przykładów testowych wg [138], [139], o rozmiarach  $n = 20$ ,  $m = 20$

Oznaczenie przykładu wg [139]	Numer iteracji	$C_{max}$ uzyskany w danej iteracji dla próby $i=$							Wartość średnia $C_{max}$	Wartość optymalna $C_{max}$ wg [139]	PRD(GS) [%]
		1	2	3	4	5	6	7			
<i>l</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>	<i>12</i>
tai022	1000	2154	2140	2129	2151	2167	2147	2151			
	5000	2126	2140	2123	2150	2143	2147	2133			
	10000	2126	2140	2122	2150	2143	2147	2132			
	15000	2126	2133	2122	2137	2143	2147	2132			
	$C_{max}$	2126	2133	2122	2137	2117	2147	2132	2130,57	2099	<b>1,50</b>
	$iter_{opt}$	-	-	-	-	-	-	-			
tai024	1000	2298	2262	2292	2263	2304	2255	2267			
	5000	2267	2250	2255	2263	2250	2255	2267			
	10000	2267	2250	2251	2263	2250	2255	2267			
	15000	2267	2250	2251	2263	2250	2255	2267			
	$C_{max}$	2267	2250	2251	2263	2250	2255	2267	2257,57	2223	<b>1,56</b>
	$iter_{opt}$	-	-	-	-	-	-	-			
tai025	1000	2325	2340	2336	2361	2338	2363	2376			
	5000	2325	2328	2336	2335	2329	2363	2335			
	10000	2325	2328	2327	2335	2317	2358	2330			
	15000	2325	2323	2327	2335	2317	2321	2330			
	$C_{max}$	2325	2323	2327	2335	2317	2321	2330	2325,43	2291	<b>1,50</b>
	$iter_{opt}$	-	-	-	-	-	-	-			
tai027	1000	2326	2328	2316	2329	2330	2337	2317			
	5000	2310	2303	2302	2322	2310	2337	2317			
	10000	2310	2300	2302	2322	2310	2321	2317			
	15000	2310	2300	2302	2313	2308	2321	2317			
	$C_{max}$	2310	2300	2302	2313	2305	2321	2311	2308,86	2273	<b>1,58</b>
	$iter_{opt}$	-	-	-	-	-	-	-			
tai029	1000	2322	2305	2277	2273	2340	2332	2325			
	5000	2268	2289	2249	2240	2340	2295	2287			
	10000	2254	2289	2249	2240	2340	2294	2287			
	15000	2254	2289	2247	-	2326	2294	2287			
	$C_{max}$	2254	2289	2247	2237	2326	2294	2287	2276,29	2237	<b>1,76</b>
	$iter_{opt}$	-	-	-	14101	-	-	-			
<b>średni PRD(GS) <math>n = 20</math>, <math>m = 20</math>:</b>										<b>1,58</b>	

$iter_{opt}$  – numer iteracji, w której uzyskano najlepszą wartość  $C_{max}$  w danej próbie  $i$

Tabela Z.7. Szczegółowe wyniki obliczeń weryfikacyjnych dla problemu przepływowego (p. 4.2.1 rozprawy) przeprowadzonych za pomocą algorytmu SA dla wybranych przykładów testowych wg [138], [139], o rozmiarach  $n = 20$ ,  $m = 5$

Oznaczenie przykładu wg [139]	Numer iteracji	$C_{max}$ uzyskany w danej iteracji dla próby $i=$							Wartość średnia $C_{max}$	Wartość optymalna $C_{max}$ wg [139]	PRD(SA) [%]			
		1	2	3	4	5	6	7						
<i>l</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>	<i>12</i>			
tai001	1000	1317	1314	1297	1314	1320	1312	1305						
	5000	1297	1297	1297	1297	1297	1297	1297						
	10000	1286	1291	1290	1295	1291	1280	1293						
	15000	1282	-	1290	-	1279	-	-						
	$C_{max}$	1278	1278	1278	1278	1278	1278	1278				1278,00	1278	<b>0,00</b>
	$iter_{opt}$	17236	12394	17225	13132	17555	12448	11232						
tai002	1000	1373	1370	1372	1367	1373	1368	1380						
	5000	1368	1368	1368	1367	1367	1367	1367						
	10000	1360	1366	1366	1366	1366	1365	1365						
	15000	-	1365	1360	-	-	1360	-						
	$C_{max}$	1359	1360	1359	1359	1359	1359	1359				1359,14	1359	<b>0,01</b>
	$iter_{opt}$	14217	-	15900	13603	10568	15869	14664						
tai003	1000	1160	1146	1165	1158	1153	1147	1163						
	5000	1121	1132	1123	1122	1132	1114	1123						
	10000	1087	1091	1098	1098	1090	1098	1096						
	15000	1087	1087	1084	-	1087	1084	1082						
	$C_{max}$	1081	1081	1081	1081	1081	1081	1081				1081,00	1081	<b>0,00</b>
	$iter_{opt}$	15331	16629	15775	13974	18774	16684	15455						
tai004	1000	1385	1397	1400	1397	1390	1383	1382						
	5000	1351	1351	1360	1349	1345	1344	1353						
	10000	1312	1309	1305	1298	1302	1310	1309						
	15000	1298	-	1299	-	-	-	1299						
	$C_{max}$	1298	1293	1297	1293	1293	1293	1299				1295,14	1293	<b>0,17</b>
	$iter_{opt}$	-	14877	-	14164	13182	12998	-						
tai006	1000	1258	1253	1233	1235	1272	1258	1269						
	5000	1224	1230	1227	1224	1224	1224	1235						
	10000	1210	1210	1205	1203	1206	1198	-						
	15000	-	1200	-	-	-	-	-						
	$C_{max}$	1195	1200	1195	1195	1195	1195	1195				1195,71	1195	<b>0,06</b>
	$iter_{opt}$	13250	-	12294	12215	10451	12684	9500						
<b>średni PRD(SA) <math>n = 20</math>, <math>m = 5</math>:</b>											<b>0,05</b>			

$iter_{opt}$  – numer iteracji, w której uzyskano najlepszą wartość  $C_{max}$  w danej próbie  $i$

Tabela Z.8. Szczegółowe wyniki obliczeń weryfikacyjnych dla problemu przepływowego (p. 4.2.1 rozprawy) przeprowadzonych za pomocą algorytmu SA dla wybranych przykładów testowych wg [138], [139], o rozmiarach  $n = 20, m = 10$

Oznaczenie przykładowg [139]	Numer iteracji	$C_{max}$ uzyskany w danej iteracji dla próby $i=$							Wartość średnia $C_{max}$	Wartość optymalna $C_{max}$ wg [139]	PRD(SA) [%]
		1	2	3	4	5	6	7			
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>	<i>12</i>
tai011	1000	1710	1702	1719	1727	1734	1719	1677			
	5000	1641	1652	1664	1649	1660	1656	1651			
	10000	1598	1596	1599	1604	1598	1593	1609			
	15000	1586	1586	1586	1583	1586	1586	1586			
	$C_{max}$	1586	1586	1586	1583	1586	1586	1586	1585,57	1582	<b>0,23</b>
	$iter_{opt}$	-	-	-	-	-	-	-			
tai012	1000	1786	1787	1772	1779	1798	1778	1792			
	5000	1753	1758	1721	1761	1737	1758	1742			
	10000	1682	1687	1691	1691	1694	1691	1679			
	15000	1675	1669	1671	1669	1667	1669	1673			
	$C_{max}$	1675	1669	1671	1669	1667	1669	1673	1670,43	1659	<b>0,69</b>
	$iter_{opt}$	-	-	-	-	-	-	-			
tai015	1000	1523	1562	1555	1562	1549	1560	1541			
	5000	1505	1506	1509	1517	1498	1507	1504			
	10000	1432	1440	1444	1425	1435	1440	1446			
	15000	1423	-	1425	-	-	-	1425			
	$C_{max}$	1423	1419	1425	1419	1419	1419	1425	1421,29	1419	<b>0,16</b>
	$iter_{opt}$	-	10185	-	10786	13108	12078	-			
tai017	1000	1599	1583	1575	1585	1589	1559	1575			
	5000	1545	1542	1531	1552	1543	1536	1545			
	10000	1500	1490	1486	1487	1501	1494	1494			
	15000	1486	-	-	-	-	-	1486			
	$C_{max}$	1486	1484	1484	1484	1484	1484	1486	1484,57	1484	<b>0,04</b>
	$iter_{opt}$	-	12993	12410	12245	13686	13464	-			
tai018	1000	1683	1658	1661	1672	1681	1652	1651			
	5000	1605	1604	1616	1612	1606	1598	1620			
	10000	1569	1560	1564	1565	1558	1551	1560			
	15000	1550	1544	1550	1551	1544	1551	1546			
	$C_{max}$	1550	1544	1550	1551	1544	1551	1546	1548,00	1538	<b>0,65</b>
	$iter_{opt}$	-	-	-	-	-	-	-			
<b>średni PRD(SA) <math>n = 20, m = 10</math>:</b>										<b>0,35</b>	

$iter_{opt}$  – numer iteracji, w której uzyskano najlepszą wartość  $C_{max}$  w danej próbie  $i$

Tabela Z.9. Szczegółowe wyniki obliczeń weryfikacyjnych dla problemu przepływowego (p. 4.2.1 rozprawy) przeprowadzonych za pomocą algorytmu SA dla wybranych przykładów testowych wg [138], [139], o rozmiarach  $n = 20$ ,  $m = 20$

Oznaczenie przykładu wg [139]	Numer iteracji	$C_{max}$ uzyskany w danej iteracji dla próby $i=$							Wartość średnia $C_{max}$	Wartość optymalna $C_{max}$ wg [139]	PRD(SA) [%]			
		1	2	3	4	5	6	7						
<i>l</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>	<i>12</i>			
tai022	1000	2220	2242	2234	2233	2241	2241	2251						
	5000	2173	2183	2190	2182	2195	2179	2177						
	10000	2120	2113	2121	2109	2119	2111	2123						
	15000	-	2113	-	2101	2105	2111	2113						
	$C_{max}$	2099	2113	2099	2101	2105	2111	2113				2105,86	2099	<b>0,33</b>
	$iter_{opt}$	11256	-	10608	-	-	-	-						
tai024	1000	2354	2377	2369	2350	2347	2359	2362						
	5000	2310	2305	2316	2299	2298	2299	2312						
	10000	2240	-	2240	2248	2229	2240	2233						
	15000	2232	-	2229	2225	2229	2229	2233						
	$C_{max}$	2232	2223	2229	2225	2229	2229	2233				2228,57	2223	<b>0,25</b>
	$iter_{opt}$	-	9641	-	-	-	-	-						
tai025	1000	2427	2426	2437	2450	2421	2430	2414						
	5000	2388	2380	2384	2374	2364	2372	2354						
	10000	2317	2309	2313	2306	2324	2304	2319						
	15000	2297	2296	2298	2306	2305	2298	2300						
	$C_{max}$	2297	2296	2298	2306	2305	2298	2300				2300,00	2291	<b>0,39</b>
	$iter_{opt}$	-	-	-	-	-	-	-						
tai027	1000	2405	2389	2412	2393	2386	2360	2401						
	5000	2365	2351	2367	2336	2355	2360	2369						
	10000	2296	2300	2299	2308	2302	2294	2300						
	15000	2284	2288	-	2288	2278	2278	-						
	$C_{max}$	2284	2288	2273	2287	2278	2278	2273				2280,14	2273	<b>0,31</b>
	$iter_{opt}$	-	-	13586	-	-	-	12995						
tai029	1000	2380	2385	2376	2394	2353	2418	2366						
	5000	2317	2324	2324	2316	2323	2333	2315						
	10000	2242	2251	2244	2255	2248	2253	2252						
	15000	2242	2242	2242	2242	2246	2244	2242						
	$C_{max}$	2242	2242	2242	2242	2246	2244	2242				2242,86	2237	<b>0,26</b>
	$iter_{opt}$	-	-	-	-	-	-	-						
<b>średni PRD(SA) <math>n = 20</math>, <math>m = 20</math>:</b>											<b>0,31</b>			

$iter_{opt}$  – numer iteracji, w której uzyskano najlepszą wartość  $C_{max}$  w danej próbie  $i$

Tabela Z.10. Wyniki obliczeń przeprowadzonych za pomocą algorytmu GS, związane z poszukiwaniem zbioru punktów kompromisowych w przykładzie przedstawionym w p. 6.1.3

Lp.	Ograniczenie $\hat{U}$ na całkowity koszt przedsięwzięcia [tys. PLN]	Wyniki po optymalizacji		
		Całkowity koszt przedsięwzięcia [tys. PLN]	Czas przedsięwzięcia (najlepsza wartość $C_{max}$ ) [dni robocze]	Permutacja dla najlepszej wartości $C_{max}$
1	2	3	4	5
1	685,38	<b>685,38</b>	<b>290</b>	((8),(6,12),(5),(1,2,3,4,7,9,10,11)), ((6,2,3,12,9,8,1,10,11,4,5,7),(0,0,0)), ((2,12,8,1,5),(6,3,9,10,11,4,7),(0,0,0)), ((6,2,3,12,9,8,1,10,11,5,4,7),(0,0,0)), (0,(2),(5),(6,3,12,9,8,10,1,11,4,7)), ((2,6,3),(12,9,8,1),(10,11),(4,5,7)), (0,(3,1),(6),(12,2,8,4,7),(9,10,11,5)), (3),(10,2),(6,9,8,11,1,5),(12,4,7)), (9,2,3,12),(4),(6,8,10,1),(11,5,7)), (1),(3,12,4),(8,2,10,11,5),(9,6,7)), (1),(8,4,12,9),(3,6,10,11,2,5,7))
2	700,00	<b>699,99</b>	<b>287</b>	((0,0,(8),(1,2,3,5,4,6,7,9,10,11,12)), ((8,5,4,1,10,11,9,7,12,3,6,2),(0,0,0)), (5,1,10,7,12),(8,4,11,9,3,6,2),(0,0,0)), ((8,5,4,1,10,11,9,7,3,6,12,2),(0,0,0)), (0,(1),(5),(8,4,10,11,9,7,3,6,12,2)), (0,(2),(5),(1,8,4,11,9,10,7,3,6,12)), (2),(1,3),(6),(8,4,7,12),(5,11,9,10)), ((7,3,6,2),(10),(4,1,5),(8,11,9,12)), (5),(10,12),(8,7),(11,9,1,4,3,6,2)), (5,1,4,8),(2),(11,9,10,7),(3,6,12)), (11,1,8,3,5),(10,4,12,9),(7,6,2))
3	800,00	<b>797,04</b>	<b>267</b>	((0,(2),(1),(3,4,7,5,6,8,12,9,10,11)), ((1,7,12,4,8,5,6,9,3,11,10),(0,0,2)), ((2,12,8,5,1,10),(7,4,6,9,3,11),(0,0,0)), ((2,7,12,4,8,5,6,1,9,3,11),(0,0,10)), (0,0,(10),(2,7,12,4,8,5,6,1,9,3,11)), (0,0,(11),(2,7,12,4,5,6,8,1,9,3,10)), (11),(3),(1,6),(2,4,7,8,10),(12,5,9)), (0,(6,10),(7,5,1,3),(12,2,4,8,9,11)), (0,(8),(6),(7,2,1,4,12,5,9,3,10,11)), (7),(10),(8,5,6,2),(1,4,12,9,3,11)), (7),(12,2,5),(1,8,4,6,3,9,10,11))
4	900,00	<b>884,01</b>	<b>255</b>	((0,0,(2),(9,10,8,3,6,11,5,4,12,1,7)), ((2,8,3,5,6,11,1,4),(0,9),(10,12,7)), ((9,8,3,5,4),(2,10,6,11,12,1,7),(0,0,0)), ((9,2,8,10,3,6,11,4,12,1,7),(0,0,5)), (0,0,(3),(8,9,10,2,6,11,5,4,12,1,7)), (0,0,0),(8,9,10,3,2,6,11,5,4,12,1,7)), (7),(3,1),(6,11),(8,2,5,4),(10,9,12)), (11,1),(2,6,12),(5,4),(10,3,9,8,7)), (1),(10,3,9),(2,6,8,11),(5,4,12,7)), (4),(6,10,11,8),(2,5),(9,3,12,1,7)), (8,5,9),(3),(6,10,2,4,11,12,1,7))
5	1000,00	<b>991,37</b>	<b>252</b>	((1),(0,(10),(2,3,6,5,11,4,7,8,9,12)), ((10,6,5,8,7,3,1,4),(0,0),(2,11,12,9)), (2,11,8,7,4),(10,6,12,9,3,1),(0,0,(5)), ((10,6,5,12,8,7,11,2,9,1),(3),(4),(0)), (4),(2,1),(3),(5,6,10,12,7,11,8,9)), (0,0,(6,1),(5,10,12,7,11,4,3,2,8,9)), (4,2),(10,3),(1),(5,6,7,8),(12,11,9)), (8),(11),(10,4,2,1),(5,6,12,7,3,9)), (0,(2,1),(4,3),(5,6,12,10,7,11,8,9)), (2),(5,3,10),(11,6,12,1),(4,7,8,9)), (5,6,2),(10,12,1),(11,7,3,8,4,9))
6	1100,00	<b>1098,14</b>	<b>233</b>	((3),(0,(2),(10,1,4,12,11,5,6,7,8,9)), ((10,1,4,9,8,6),(0,(11),(2,3,12,5,7)), (2,12,3,4,1),(10,11,9,8,6,7),(0,0,(5)), ((2,11,12,9,3,4,6,1,7),(8),(0,(10,5)), (0,0,(3,4),(2,10,11,9,12,8,5,6,1,7)), (0,0,0),(2,11,10,9,12,3,4,5,8,6,1,7)), (8),(3,7),(11,6),(9,4,1),(2,10,12,5)), (6),(1),(10,3,4,5),(2,11,9,12,8,7)), (0,0,(9,5),(2,10,11,12,3,4,8,6,1,7)), (1),(4),0,(2,10,11,12,9,3,8,6,5,7)), (12,1),0,(2,11,10,9,3,4,8,6,5,7))

Tabela Z.10. cd.

1	2	3	4	5
7	1200,00	<b>1191,40</b>	<b>235</b>	((0),(3),(12),(4,1,10,2,5,6,7,8,9,11)), ((12,1,10,6,7,5),(0),(3),(4,2,8,11,9)), ((4,2,8,5),(3,1,11,6,9,7),(0),(12,10)), (3,12,1,10,2,8,5,9,6,7),(0),(4),(11)), (0),(0),(3,8,6,7),(4,1,12,10,2,11,5,9)), (0),(1,6),(10,4,2,8),(12,3,11,5,9,7)), (2),(8,9),(11),(1,12,4,6),(10,3,5,7)), ((10,2),(8,5,3),(12,11,1,9,6),(4,7)), (0),(9,6),(4,1),(10,2,8,11,12,5,3,7)), (0),(1,5),(2,10,8,11,9,3,6),(4,12,7)), (9),(11,2,6),(4,10,1,8,5,3,12,7))
8	1300,00	<b>1296,15</b>	<b>227</b>	((0),(4,12,1),(6,2),(7,5,3,10,9,11,8)), (4,2,12,1,3,10,7,9),(0),(6,11),(5,8)), (6,12,3,5),(4,11,1,10,9,7),(8),(2),(0)), (2,8,4,11,10,5,7),(3),(12,6,9),(1)), (4,8,3,5),(12,9,7),(1),(2,11,10,6)), (9),(8,3,5),(12,11,7),(4,1,2,10,6)), (1,3,9),(5,7),(1),(4,10,2),(12,8,6)), (2),(1,12,5,9,8),(3,11,7),(4,10,6)), (10,9),(2,8),(12),(4,5,1,11,6,3,7)), (11),(2,6,3),(10,5,8),(12,4,1,9,7)), (2,12,6,3),(9,7),(4,10,5,11,1,8))
9	1400,00	<b>1391,69</b>	<b>227</b>	((0),(0),(4),(5,8,1,2,3,11,7,6,10,12,9)), (1,6),(5),(4,11,7),(8,2,3,12,9,10)), (1,8,6,3),(4,11,7,10,2),(0),(5),(12,9)), (5,4,8,12,7,9,3,1,2),(0),(0),(11,6,10)), (0),(0),(8,3,1),(5,4,11,7,6,10,12,9,2)), (0),(0),(5,4,11,8,7,6,10,12,3,1,9,2)), (7,2),(3),(8,6,1),(4,12,9),(5,11,10)), (3),(1,9),(4,8,7,10,5),(11,6,12,2)), (0),(9),(10,5),(4,11,8,7,6,12,3,1,2)), (5),(8,3,1,9),(7),(4,6,11,10,12,2)), (4,1,9),(5,8),(7,6,10,11,3,12,2))
10	1500,00	<b>1476,20</b>	<b>202</b>	((0),(2),(3),(1,4,5,7,11,6,8,10,9,12)), (2,4),(1,8,10),(11,5,9),(3,7,6,12)), (2,4,1,5),(3,11,7,10,9),(8),(6),(12)), (3,2,4,7,1,5,9),(8,6),(12),(11,10)), (8,12),(4,7,1,10),(5,9),(3,2,11,6)), (2,3,6,1,5),(4),(11,7,10,9),(8,12)), (7,10),(3,1,5),(4,6,9),(2,8),(11,12)), (7,4,10),(3,1,9),(11,2,8,5),(6,12)), (2,7,1),(11,5),(3,12,8),(6,4,10,9)), (2,6,1,5,9),(4),(11,7,10),(3,12,8)), (2,4,7,3,11,6),(10,5,12),(1,8,9))
11	1600,00	<b>1596,62</b>	<b>190</b>	((3),(0),(5,8),(2,10,6,1,4,9,7,11,12)), (8,7,4,1,11),(3),(2,6),(5,10,9,12)), (8,9,1),(3,6,7,10),(0),(5,4),(2,12,11)), (3,8,6,9,12,1,11),(2),(4),(5,7,10)), (10),(7,4),(5,6),(3,8,2,9,12,1,11)), (0),(0),(4),(3,5,2,6,7,12,8,9,1,10,11)), (4,8),(2,7,1),(6,10),(12,11),(3,5,9)), (1,8),(2,4,11),(6,7,12,9),(3,5,10)), (9),(8),(2,3,4,1),(6,7,5,10,12,11)), (6,2),(1,8),(3,5,10,4,9),(7,12,11)), (4,6,8),(3,10,5,1,9),(2,7,12,11))
12	1700,00	<b>1687,40</b>	<b>206</b>	((0),(0),(7),(12,5,1,3,8,11,10,9,6,4,2)), (7,3,9,11,2),(1,10),(5,8,6),(12,4)), (12,1,9,11,2),(7,10,4),(3,6),(5),(8)), (5,3,1,9,11,2),(6),(7,8,10),(12,4)), (1,9,11),(3,6,7,2),(12,5,10,4),(8)), (3,10),(9,11,2),(5,1),(8,6,12,4,7)), (10,3,9),(1),(5,12,2),(6,4,11),(8,7)), (5,11),(10,9),(6,12,4,3),(8,1,7,2)), (4,1,9),(10,7,8),(6,12),(5,3,11,2)), (5,1,9,3,2),(8,6),(12,7,11),(10,4)), (7,4,1,11,12),(2),(10,5,8,6,9,3))
13	1800,00	<b>1797,06</b>	<b>188</b>	((0),(9,2),(1),(4,6,3,11,5,12,7,8,10)), (4),(6,2),(9,11,5,3,7),(1,12,10,8)), (9,12,8),(1,11,10,3,7),(6,2),(5),(4)), (4,6,5,3,7),(8),(9,2),(1,11,12,10)), (0),(5),(1,2,8,3,7),(4,9,6,12,10,11)), (8),(0),(3,7),(1,4,9,5,2,6,12,10,11)), (12,6),(1,2,3),(8,7),(5,10),(4,9,11)), (2,3),(1,8,7),(5,10,6),(4,9,12,11)), (0),(2,3),(11),(4,9,5,1,12,8,10,6,7)), (3),(4,9,2,5,8,11),(0),(1,12,10,6,7)), (9,1,3,10),(4,12,2,8,7),(5,6,11))

Tabela Z.10. cd.

1	2	3	4	5
14	1900,00	1870,98	182	((0,0),(1,8),(2,4,11,3,6,9,12,5,7,10)), ((1,9),(2,6),(11,4,5,10),(8,3,12,7)), ((8,6,2),(11,3),(4,5),(1,12,7),(9,10)), ((1,11,6,3,2,5),(4),(12),(8,9,7,10)), (0,(1,7,5),(3,2),(8,11,12,4,9,6,10)), (0,(2,5),(1,4,7),(8,11,12,3,9,6,10)), (4),(3,6),(11,9,5),(8,7,10),(1,12,2)), ((12,3,6),(8,1,7,5),(4,2),(11,9,10)), (6),(7,2,5),(12,4,8),(1,11,3,9,10)), (2,6),(4,5),(1,12,7,8),(11,3,9,10)), ((1,2,6),(12,3,8,7,5),(11,4,9,10)))
15	2000,00	1998,58	173	((5),0,(1,2),(12,9,3,6,8,4,7,11,10)), (1),(5,2,3),(9,6,7,4),(12,8,11,10)), ((12,8,2),(1,3,4),(7),(5,6),(9,11,10)), (5,1,6,4),(0,(12,3,2),(9,8,11,7,10)), (0,(1,2),(9,3,6),(5,12,8,11,7,4,10)), (0,0,(2),(5,12,9,1,8,3,11,6,7,4,10)), (3,10),(1,2),(5,9,7),(12,11,4),(8,6)), (8),(3,2,6),(12,1,7,4),(5,9,11,10)), (0,(2),(3,6),(5,9,12,1,11,7,8,4,10)), (0,(2),(1,8,4,6),(9,5,12,11,7,3,10)), (0,(1,8,9,3,6),(5,12,11,2,7,4,10)))
16	2100,00	2092,69	178	((0),(3),(6,2),(12,5,11,1,8,7,9,4,10)), (3,2,7,4),(5),(6,11,10),(12,8,9,1)), (3,2),(6,7,10),(5,4),(12,8),(11,9,1)), (6,2,10),(0,(5,8,7,1),(12,11,3,4,9)), (10,1),(5,2),(12,11,3,9),(6,8,7,4)), (0,(6,2,7,3,10,9),(11),(5,12,8,4,1)), (3),(7,12),(8,6),(5,11,10,9),(2,4,1)), (5,3,6,9),(11,2,1),(7,12,4),(8,10)), (0,(12,2,3,1),(7,6),(11,5,10,8,4,9)), (11,8,3,7),(12,9),(2),(5,10,6,4,1)), ((4,6,7),(11,10,8,12,9),(2,3,5,1)))
17	2200,00	2154,43	169	((4),(8),(10),(3,12,2,7,1,11,5,6,9)), (3,1,6,2),(4),(8,7),(10,12,11,5,9)), (8,1,9),(10,11,5),(12,2),(3),(4,7,6)), (4,1,2),(7,5),(10,3),(8,12,11,6,9)), (9),(4,11,2,5),(8,10,3,1),(12,7,6)), (4),(8),(10,12,7,3,1,2,5),(11,6,9)), (8,7,2),(12,11,5),(10,9),(1),(4,3,6)), (8,4,6),(12,11,5),(10,3,1,9),(7,2)), (2),(4,3,5,6),(8,11,1),(10,12,7,9)), (12,4,2),(10,5),(3,1,6),(7,11,8,9)), (10,8,1,4,6),(11,7,2,5),(3,12,9))
18	2300,00	2195,84	171	((5),(9),(2,4),(6,1,12,3,8,7,10,11)), (6,1),(5),(9,4,3,7),(2,12,8,10,11)), (2,4),(6,1,11),(9,12),(5,3,7),(8,10)), (3,7,4),(5,10),(2,1),(9,6,12,8,11)), (0,(3),(5,2,1,7,10,4),(9,6,12,8,11)), (0,0,(4),(5,9,6,2,12,1,3,8,7,10,11)), (2,7),(6,8,4),(9,3),(12,10),(5,1,11)), (10),(5,7,1),(9,6,12,8,4),(2,3,11)), (0,(4),(5,8,9,1),(2,6,12,3,7,10,11)), (0,(5,8,9,1),(2,6,12,4),(3,7,10,11)), ((12,8,2,5,4,1),0,(6,7,3,10,9,11)))
19	2400,00	2391,93	170	((9),0,(6,1),(12,2,5,7,11,3,4,8,10)), (5,2),(1),(6,9,3,4,8),(12,11,7,10)), (1,2),(9),(6,3,10),(5,7),(12,11,4,8)), (1,9,2),(4),(5,3,7,8),(6,12,11,10)), (2,4),(1),(5,3,10),(6,12,9,11,7,8)), (0,(2,7,4),(6,12,1,3,8),(9,5,11,10)), (9),(5,7),(12,3,8),(1,11,10),(6,2,4)), (12,9,2),(3,4),(6,5,7,8),(1,11,10)), (11),(9,2,7,10),(12,5),(6,3,1,4,8)), (3),(1,2),(6,5,4,10),(12,9,11,7,8)), (3,11),(5,2,4,10),(1,6,12,9,7,8))
20	2500,00	2253,54	175	((0),(4),(1,3),(5,6,9,8,2,7,10,11,12)), (5,3,2,7),(6),(4,9),(1,8,10,11,12)), (5),(6,3,11),(9,10),(1,2),(4,8,7,12)), (5,3,11),(4),(1,9,2,10),(6,8,7,12)), (9),0,(1,4,5,2,7,10),(6,8,3,11,12)), (0,0,0),(1,6,4,9,8,2,3,5,7,10,11,12)), (8,5),(4,2,7),(6,10,1),(9,11),(3,12)), (9,10,1),(4,7,3),(2,6,5,12),(8,11)), (5),(7,11,1),(6),(4,2,9,8,10,3,12)), (8,4,12),(2,7,3),(9,6,5,1),(11,10)), (10,3),(8,6,2,7,4,1),(9,5,11,12))



Tabela Z.10. cd.

1	2	3	4	5
21	2600,00	<b>2303,30</b>	<b>176</b>	((0),(1,5),(6),(7,2,11,3,8,4,9,10,12)), ((1,2,10,12),(5),(6,11,3,4,9),(7,8)), (6,1,12),(7,4),(11,10,3),(5,2,9),(8)), (2),(6,10,4),(7,11,1,3),(8,5,9,12)), (1,1,12),0,(8,10,4,3),(7,6,2,5,11,9)), (0,0),(11,3),(8,7,6,2,10,1,5,4,12,9)), (8,1,12),(6,10,11),(2,4),(5,9),(7,3)), (8,1,12),(6,5,11),(2,4,9),(7,10,3)), (1),(10,4,12,3),(6,7),(2,8,5,11,9)), (0),(6,7,1,10,4,11),(2,12,3),(8,5,9)), (11),(7,6,8,10,1,12,3),(5,4,2,9))
22	2700,00	<b>2044,11</b>	<b>167</b>	((0),(4),(2,7),(8,5,6,10,3,1,11,9,12)), (2,7,3,9),(10),(4,6),(8,5,11,1,12)), (4,3,1),(6,11,12),(2,5),(10),(8,7,9)), (8,6,3,1),(2,5,7),(10,9),(4,11,12)), (10,3,1),(4,11,9),(6,5),(8,2,7,12)), (8,7),(4,9),(2,10,1),(6,5,3,11,12)), (3,11),(8,2,12),(6,7),(10,9),(4,5,1)), (8,3,1),(6,7,11),(10,5,9),(4,2,12)), (7,9),(12),(3,2,1),(10,4,8,6,5,11)), (4,6,5,11),(10,7,12,1),(8,3,2),(9)), (5,2,11),(7,3,6,4,1),(8,10,9,12))
23	2800,00	<b>2157,82</b>	<b>178</b>	((0),(2),(1,4,12),(7,6,8,10,9,5,3,11)), (4,8),(7),(1,10,9,5,3,11),(2,6,12)), (6),(1,10,11),(2,12,5),(4,9),(7,8,3)), (6,9,5,11),(4),(1,10),(7,8,12,2,3)), (0),(6,11),(1,8,10,9),(7,4,12,2,5,3)), (0),(5,11),(1,2,4),(7,8,6,12,10,9,3)), (10,5),(8,9,11),(12),(1,7,2),(6,4,3)), (1,7,4,11),(12,9),(10,5,3),(6,8,2)), (7,3),(4,2),0,(1,12,6,8,10,9,5,11)), (12,2),(8,1,10),(7,4,11),(6,9,5,3)), (10,1,4,11),(5,7,3),(6,12,8,9,2))
24	2900,00	<b>2447,61</b>	<b>168</b>	((0),(4),(2,6,11,1),(9,7,10,5,12,3,8)), (1),(6,3,9),(4,7,10,5),(2,12,11,8)), (1),(6,11,9),(12,10,5),(2,3),(4,7,8)), (7,5),(4,1,10,9),(2,3),(6,12,11,8)), (6),(3,8),(2,1,12,10,5),(4,7,11,9)), (12),(2,6,1,8,9),(4,7,5),(3,11,10)), (1),(2,12,5),(4,7,10),(6,11,9),(3,8)), (1),(4,6,3,9),(7,10,5),(2,12,11,8)), (4,9),(1,6,10,2,5),(12),(7,3,11,8)), (2,9),0,(10,12,5),(4,7,3,1,6,11,8)), (3,10,2,1,7,9),(12,6,11,5),(4,8))
25	3000,00	<b>2451,28</b>	<b>173</b>	((3),(2),(1),(5,4,7,8,6,9,11,10,12)), (2),(1,7),(4,3,6,9),(5,8,10,11,12)), (4),(1,3,9),(5,6),(2,7,11),(8,10,12)), (3),(1,4,10,11),(2,8,6),(5,7,9,12)), (0),(9),(2,5,4,6,3,11),(1,8,7,10,12)), (0,0,0),(2,1,8,5,4,7,6,10,3,9,11,12)), (6,12),(5,4,9),(8,11),(1,10,3),(2,7)), (6,9,3),(1,4),(2,8,7,11),(5,10,12)), (0),(6,10,3),(1),(2,5,4,7,8,9,11,12)), (9,3),(5,7),(8,2,11),(4,6,1,10,12)), (3),(9,2,11),(5,6,4,1,8,7,10,12))
26	3091,47	<b>2519,75</b>	<b>170</b>	((8,4,7),(3),(1,5,2,10,6,9,11,12)), (4),(1,7,2),(3,5,9),(8,10,11,12,6)), (4),(10),(3,5,11,6),(1,9,2),(8,7,12)), (2,6),(1,10,11),(3,5),(8,4,9,7,12)), (0),(4,2),(3,5,7),(8,1,10,9,11,12,6)), (0,0),(9,2),(8,1,4,3,5,10,7,11,12,6)), (3),(11,6),(4,5,12),(1,9,2),(8,10,7)), (5),(1,4,12,6),(3,9,11,7),(8,10,2)), (6),(7),(1,5,11),(3,8,4,10,9,12,2)), (7),(3,2),(1,5,4,8),(10,9,11,12,6)), (5,8,7),(1,10,3,9,4,2),(11,12,6))

Tabela Z.11. Wyniki obliczeń przeprowadzonych za pomocą algorytmu TS, związane z poszukiwaniem zbioru punktów kompromisowych w przykładzie przedstawionym w p. 6.1.3

Lp.	Ograniczenie $\hat{U}$ na całkowity koszt przedsięwzięcia [tys. PLN]	Wyniki po optymalizacji		
		Całkowity koszt przedsięwzięcia [tys. PLN]	Czas przedsięwzięcia (najlepsza wartość $C_{max}$ ) [dni robocze]	Permutacja dla najlepszej wartości $C_{max}$
1	2	3	4	5
1	685,38	<b>685,38</b>	<b>276</b>	((8),(12,6),(5),(1,11,7,4,2,10,3,9)), ((12,11,2,7,5,10,3,9,8,6,1,4),0,0,0), ((12,2,5,8,9,4),(11,7,3,10,6,1),0,0,0), ((12,11,2,7,5,3,10,8,6,9,1,4),0,0,0), ((11,7,8,9),(12,2,6,4),(3,10),(5,1)), (3,1,4),(5,8),(2,7,6),(11,12,10,9)), (0,(3,1),(6),(2,7,12,8,4),(11,5,10,9)), (2,7,3,10,9),0,(11,12,5,8,1),(6,4), (0,(2,6,9,1),(11,8,10),(12,3,5,7,4)), (9,8),(5,3,11,12,7,6),(2,1),(10,4), (9,10,1),(5,12,8,6),(11,2,7,3,4))
2	700,00	<b>699,76</b>	<b>269</b>	((0,(11,6),(9,3),(12,8,2,1,10,7,4,5)), ((11,3,2,1,5,12,9,6,8,7,10,4),0,0,0), (2,5,9,8,10),(11,3,1,12,6,7,4),0,0,0), ((11,3,2,1,5,12,9,8,6,7,10,4),0,0,0), (2,1,9),0,(3,5,8,7,10,4),(11,12,6), (2,3,8,7,1),(11,5,12,4),(9,10),(6)), (4),(3,1),(6),(2,12,8,7),(11,5,9,10)), (8,1,7),(11,10),(6,2,4),(3,12,5,9)), (11,6,12,7),(9,1,5,10),(3),(2,8,4), (1,5,7),(3,2,11,4),(10),(9,8,12,6), (12,3,7,5),(8,1,11,9,2),(6,10,4))
3	800,00	<b>790,60</b>	<b>263</b>	((12,4),(11,2,7,9,3,1),(10),(8,5,6)), ((12,9,5,7,8,4,3,1,2,10),(11),0,(6)), ((12,9,8,3,4,10),(11,6,5,7,1,2),0,0,0), ((11,12,6,9,5,8,7,3,4,1,2,10),0,0,0), (11,12,8),(7,2),(6,5,9,3,1),(4,10)), (11,3),(12,7,9,4,1),(2,10),(8,6,5)), (4),(3,1),(6,10),(12,7,8),(11,5,9,2)), (5,12,9,1),(7),(11,6,3,8,10),(4,2)), (12,6,3),(9,11,7,1,10),(2),(8,4,5)), (4,6),(12,1,10),(11,5,3,2),(8,9,7)), (11,3,8,9),(4,7,1,6,2),(12,5,10))
4	900,00	<b>898,29</b>	<b>247</b>	((2,12,5),(9,4),(6),(3,10,1,8,11,7)), (2,5,7,10,4,11,8,6,1),(9,12),(3),0), (2,5,4,8,11,1),(9,7,10,12,6,3),0,0,0), (2,9,5,7,4,12,8,11,3,1),0,(10,6),0), (12,8,1),(5,10,4),(9),(2,7,6,11,3)), (12,4,1),(9,7,8),(2,5,10,3),(6,11)), (4,8),(3),(6,1),(5,2,7),(9,10,12,11)), (4,5,8,1),(2),(10,9,12,7,6,3),(11)), (5),(10,4,2,8,1),(7),(9,12,6,11,3)), (5,11,1),(2,10,3),(12,9,4),(8,6,7)), (5,10,1),(8,6,4,3),(2,9,11,12,7))
5	1000,00	<b>988,47</b>	<b>238</b>	((4,5,12,1),(2,11,6),(9,8,10),(7,3)), (9,12,1,11,10,8,6,7,3,5),0,(4),(2)), (4,12,9,8,7,5),(2,1,10,11,6,3),0,0,0), (4,2,12,9,10,11,6,3,5),(8),0,(1,7)), (4,1,6),(2,10,5),(12,9,8,3),(11,7)), (12),(8,10),(2,4,3,5),(1,9,11,6,7)), (4,1,8,10),(5),(6),(12,7),(2,9,11,3)), (11),(1,4,12,10,3),(9,8,7,5),(2,6)), (4,12,3),(1,2,11,9,5),0,(10,8,7,6)), (11),(12,10,2,7,3),(4,1),(8,9,6,5)), (12,4,3),(10,1,9,11,7),(2,8,6,5))
6	1100,00	<b>1099,62</b>	<b>221</b>	((5,12,1),(11,10),(8,3,9),(2,7,4,6)), (2,4,5,9,1,7,8,6),(11),(12),(10,3)), (12,5,1,8,7),(11,10,3,9,4,6),(2),0,0), (11,12,2,5,9,1,8,4,7),(3,6),(10),0), (9,1,8,7),(2,5,4),0,(11,10,12,3,6)), (0,(5,11,4),(3,6),(2,10,9,12,1,8,7)), (2,9),(3,1,6),(8,7),(12,4),(10,5,11)), (2,5,8,7),(3,12,6),(9,1,4),(10,11)), (8,4),(2,12,5,6),0,(3,10,1,9,11,7)), (9,11,4,6),(12,2,8,5,7),(3,1),(10)), (1,12,9,7),(10,3,2,11,8,5,6),(4))

Tabela Z.11. cd.

1	2	3	4	5
7	1200,00	<b>1190,55</b>	<b>209</b>	((11,9),(2,4,8),(3),(1,7,10,12,5,6)), ((2,5,6,8,1,11,10),(3,7),(12),(4,9)), ((2,5,4,9,12),(3,7,6,1,11,10),(8),(0),(0)), ((3,2,7,6,8,4,11,10),(9),(1),(5,12)), ((3,8,1,9,10),(2,6,12),(11),(5,7,4)), ((6,8,1,12),(5),(2,3,11,10),(7,4,9)), ((7,11),(3,1,12),(4,10),(5,8),(2,6,9)), ((9),(2,3,1,10),(5,8,6,12),(7,4,11)), (0),(5,3,12),(1,7,9),(2,4,8,6,11,10)), ((8,11,12),(3,4,5,10),(7,9),(2,1,6)), ((3,6,11,12),(2,1,7,9,5),(4,8,10)))
8	1300,00	<b>1299,94</b>	<b>208</b>	((3,12,2,10),(9,7,1),(4),(6,11,8,5)), (9,11,7,6,10,12),(3,2,8),(1,5),(4)), (4,5,2,6),(3,11,8,10,12),(7),(0),(1,9)), (3,4,11,8,7,2,12,6),(1),(10),(9,5)), (11,10),(4,8,7,12),(3,5,2),(1,9,6)), (11,8,7,5,2,10),(0),(12),(3,1,4,9,6)), (4,1,8),(9,2,6),(0),(7,12),(3,11,5,10)), (9,5),(4,1,2,6),(7,3,10),(11,8,12)), (7),(12),(8,9),(1,4,5,2,11,3,10,6)), (11,10),(8,9,7,3,6),(2,5,12),(1,4)), ((8,5,9,10),(7,3,12),(2,11,1,4,6)))
9	1400,00	<b>1396,67</b>	<b>197</b>	((8,1),(11,7,5),(12,4,3),(2,10,9,6)), (7,3,1,12,10,5),(8,6),(11,9),(2,4)), (2,9,1,5),(11,4,12,3,10),(8),(6),(7)), (11,7,6,9,1,5,10),(8,2,3),(4,12),(0)), (9,5,10),(7,6,12,1,3),(8,4,2),(11,1)), (2,10),(11,5),(7,1,12),(4,8,6,9,3)), (7,9,3),(1),(11,4,8,10),(6,12),(2,5)), (8,3),(7,4,9,5),(2,12,10),(11,6,1)), (11,12),(6,2,7,3),(0),(4,9,1,8,5,10)), (11,3),(7,1,12),(8,9,10),(4,2,6,5)), ((1,6,12,9,5),(4,3,8,7),(11,2,10)))
10	1500,00	<b>1496,30</b>	<b>188</b>	((3),(11,9),(8,4,6,1,2),(12,10,7,5)), (3,1,2,5,4,6,10),(11,8,9),(12),(7)), (3,1,2,10),(11,7,9,6),(8),(4),(12,5)), (11,3,8,1,9,4,10),(7),(12,2,6),(5)), (12,7,9),(2),(11,5,6),(3,8,1,4,10)), (11,9),(8,2,4),(10),(3,12,7,1,5,6)), (3,5),(9,4),(11,7,8,10),(12,2),(1,6)), (9,4),(11,2,6),(7,12,8,5),(3,1,10)), (8),(7,11,3,9,6),(5),(1,12,2,4,10)), (1,3,4,6),(9,5),(11,12,7,8),(2,10)), ((12,7,4,10),(5,2,6),(1,9,11,8,3)))
11	1600,00	<b>1598,13</b>	<b>179</b>	((4,8,9,10,11),(3,12),(6,7,1,5),(2)), (6,4,9,7,5,12),(8),(3,10,1),(2,11)), (2,5),(3,10,7,12),(8,4),(6,1),(11,9)), (3,8,6,9,7,12),(2,5),(10,4),(1,1,1)), (1,4,12),(8,10),(3,6,9,5),(2,11,7)), (4),(10,1,7,5),(3,11,9,12),(2,8,6)), (2,1),(3,10,9,12),(6,4),(8,7),(11,5)), (2,1),(3,10,9,7),(8,4,5),(6,11,12)), (11,1,4),(12),(3,7),(6,2,10,8,9,5)), (8),(10,6,12),(1,4,5),(2,11,3,9,7)), ((10,2,9,12),(1,6,4,7),(11,3,8,5)))
12	1700,00	<b>1693,99</b>	<b>178</b>	((4),(11,12,2),(8,10,6),(7,1,5,9,3)), (8,5,4),(11,6,7),(12,3,2),(10,9,1)), (8,6,9),(11,3,7,4),(5),(12,2),(10,1)), (11,8,2,7,5,4),(3,9),(12,10),(6,1)), (6,1,9),(11,12,4),(2,10,7),(8,3,5)), (2,1,7),(12,8,4),(6,10,5),(11,3,9)), (11,1),(3,7,4),(8,5),(12,10,9),(2,6)), (8,10,6),(11,1,5),(12,2,7,4),(3,9)), (12,1),(2,5,6),(8,11,3,9),(10,7,4)), (11,12,3,7),(5,6),(10,1,4),(2,8,9)), (8,3,9,4),(11,2,12,1,7,10),(5,6)))
13	1800,00	<b>1772,50</b>	<b>169</b>	((7,3,5),(2),(1,12,4,9,11,8),(10,6)), (1,9,6,10),(3,11,7),(12,5,8),(2,4)), (3,4,7),(1,11,6,10),(2,5),(12,9),(8)), (3,11,8,6,10),(2,9),(12,4),(1,5,7)), (12,5,6),(9,7),(2,11,8),(3,1,4,10)), (8,6,4,7),(1,5,10),(12,2),(3,11,9)), (3,5,4),(11,6),(2,8),(12,9,10),(1,7)), (11,6,10),(2,9,4),(12,5,8),(3,1,7)), (2,9,8),(3,5,12,10),(1,4),(11,6,7)), (3,2,6,12,8),(5,9,10),(11,1,4,7),(0)), (3,12,4,7),(2,5,8),(11,1,6,9,10)))

Tabela Z.11. cd.

1	2	3	4	5
14	1900,00	1881,73	166	((3),(8,10,7,11,5),(1,2),(9,6,12,4)), ((9,5,6),(1),(3,11,8,7),(2,10,12,4)), ((1,5,7),(3,10,6),(2,12),(9,4),(11,8)), ((3,9,10,1,7),(5,8),(2,6),(11,12,4)), ((3,9,10,8),(2,6,4),(11,5),(12,1,7)), ((9,1,6,8,4),(3,12,7),(2,10),(11,5)), ((9,8),(3,12,7),(11,1,4),(10,6),(2,5)), ((11,1,7),(9,10,8),(2,12,4),(3,5,6)), ((9,5,10,6),(12,4),(11),(2,3,1,8,7)), ((7),(5,8),(11,12,3,10,4),(9,2,1,6)), ((9,1,8,6),(11,10,12,4),(3,2,5,7)))
15	2000,00	1985,45	163	((4),(11,2,10,8),(1,12,3,6),(5,7,9)), ((12,2,6,4),(1,8),(11,9),(5,3,7,10)), ((1,2,10),(11,3,6),(12,7),(5,8,9),(4)), ((11,3,7,4),(12,8,2,10),(1,6),(5,9)), ((11,8,2),(1,6,9),(12,10),(5,3,7,4)), ((1,3,9,4),(12),(8,7,10),(5,11,2,6)), ((8,10),(12,7,4),(5,2,9),(11,6),(1,3)), ((12,11,7,10),(8,6),(1,3,4),(5,2,9)), ((5,8,2,10),(11,3,6),(0),(12,1,7,4,9)), ((12,11,7,4),(5,8,1,10),(9),(3,2,6)), ((7,6,9),(5,12,2,1,10),(11,8,3,4)))
16	2100,00	2015,54	160	((11,3,1),(5,6),(8,4,10,7,2,12),(9)), ((3,6,1),(11,12),(8,4,9,2),(5,7,10)), ((3,1),(11,7,10),(12,4),(5,9),(8,6,2)), ((11,12,1,10),(3,7,2),(5,6),(8,9,4)), ((6,1,4),(5,3,2),(8,12,7,10),(11,9)), ((1),(8,5,9,2),(11,6,7,4),(12,3,10)), ((12),(5,3,1),(11,9,2),(6,10),(8,7,4)), ((5,3,10),(11,6,4),(8,7,1),(12,9,2)), ((11,1),(6,9,10,4),(5,12),(3,8,7,2)), ((8,10),(5,7,9,4),(11,1),(12,6,3,2)), ((5,8,11,12,9,1),(3,7,2),(6,10,4)))
17	2200,00	2198,66	157	((5,9,12,6,1),(11,10,4,7,3),(8),(2)), ((10,7),(11,6,1),(5,9,8),(2,4,12,3)), ((2,9,8),(11,6,3),(10,12),(5,7,4),(1)), ((5,3),(2,7,4),(10,9,1),(11,6,12,8)), ((11,9,1,4),(2,10,7,3),(12,8),(5,6)), ((5,2,10,3,8),(11,12,4),(6,1),(7,9)), ((5,7,1),(11,6,8),(2,12,4),(10,9),(3)), ((2,7,1),(11,6,8),(5,10,9,4),(12,3)), ((10,6,8),(11,2,7,3),(12,1),(5,9,4)), ((10,5,9,8),(2,11,6,1),(7,4),(12,3)), ((2,12,3,4),(10,5,1),(11,6,7,9,8)))
18	2300,00	2148,78	154	((0),(11,6,7,12,1),(8,3,4),(2,5,10,9)), ((11,6,7,10),(3,5,1),(2,9),(8,12,4)), ((12,4),(3,6,10),(11,9),(2,5,7),(8,1)), ((11,6,1),(3,5,10),(8,12,7),(2,9,4)), ((2,5,10),(3,6,1,7),(8,12,9,4),(11)), ((11,12),(2,3,10),(8,6,9,4),(5,1,7)), ((2,12,10),(8,6,4),(5,7),(3,9),(11,1)), ((6,9,4),(11,7),(8,3,12,1),(2,5,10)), ((3,4),(6,1,7),(2,8,12),(11,5,9,10)), ((3,2,8,1,10),(11),(12,9,7),(6,5,4)), ((12,8,10),(2,5,11,1,4),(3,9,6,7)))
19	2400,00	2306,25	153	((12,8,7,4,2,6),(3,9),(11,10,1),(5)), ((5,2,6),(11,10,8),(3,4,9),(12,1,7)), ((5,9),(11,10,7),(12,8),(3,2,1),(4,6)), ((3,4,1),(5,10,9),(11,2,7),(12,6,8)), ((11,9),(3,10,1),(12,4,2,8),(5,6,7)), ((3,8,7),(10,6,9),(1),(12,5,4,11,2)), ((12,10,1),(4,8),(5,2,7),(3,11,9),(6)), ((5,2,9),(12,11,1),(3,4,6,7),(10,8)), ((0),(2,1,8),(10,6,9),(3,12,5,11,4,7)), ((3,10,6),(2,11,8),(5,12,9),(4,1,7)), ((5,11,8),(3,10,4,9),(12,2,6,1,7)))
20	2500,00	2264,35	154	((2,6,10,5),(4,1,11,12),(3),(8,7,9)), ((2,7,10),(3,1),(4,11,9,6),(8,5,12)), ((8,5),(3,11,10),(2,7),(1,9),(4,12,6)), ((2,12,10),(4,11,7,6),(8,1,9),(3,5)), ((3,5),(4,11,9,6),(2,1,12),(8,7,10)), ((4,11,5,9),(2,1,12,10),(8,7,6),(3)), ((1,9),(2,5,6),(4,12,10),(3,11),(8,7)), ((11,7),(2,12,10),(3,8,5,9),(4,1,6)), ((3,11,8,9),(4,2,5,1,6),(0),(12,7,10)), ((4,11,5,10),(3,7,6),(8,12,9),(2,1)), ((7,5,6),(11,8,2,4,3,9),(1,12,10)))

Tabela Z.11. cd.

1	2	3	4	5
21	2600,00	<b>2315,59</b>	<b>153</b>	((3,8,1),(11,9,4),(2,12,6,10),(7,5), (2,4,8),(11,1,10),(3,9,6),(5,12,7), (9,7),(3,6),(5,12),(2,1,10),(11,4,8), (5,1,8,7),(11,10),(3,9,4),(2,6,12)), (11,5,12),(1,10),(2,9,4),(3,6,8,7), (3,2,10),(1,12),(11,9,5,8,7),(6,4), (3,1,8),(11,5,12),(2,4,7),(9,10),(6), (3,5,4),(11,9,10),(1,12,8),(2,6,7)), (2),(5,9,11,12,7),(3,6,4),(1,10,8), (3,6),(11,12,8),(1,2,10),(5,9,4,7), (11,10,8),(2,1,4,7),(3,5,9,12,6)))
22	2700,00	<b>2286,75</b>	<b>154</b>	((8,4),(3,11,6,9),(2),(5,7,10,1,12)), (12,6),(5,9,1),(3,4,7,10),(8,11,2), (4,6),(9,10),(5,2),(3,12,1),(8,11,7), (3,11,7),(8,4,10),(5,12,1),(9,2,6), (9,6),(8,5,4,1,10),(3,12),(11,2,7), (8,2),(5,12,11,7,10),(3,9,6),(4), (12,6),(8,11,10),(5,2,7),(3,9),(4,1)), (3,12,2,6),(11,7),(5,4,1),(8,9,10), (3,8,9,10),(5,12,6),(11),(4,2,1,7), (3,11,12,7),(4,8,1),(2,10),(5,9,6), (5,8,2,1,6),(9,4,10),(3,11,12,7)))
23	2800,00	<b>2316,88</b>	<b>154</b>	((12,9),(3,10,1,2),(8,7,4,11,6),(5), (5,9,6),(12,11,10,2),(8,4),(3,1,7)), (12),(3,1,7),(11,6),(5,9,10),(8,4,2), (12,4,7),(5,9,10),(3,1,6),(8,11,2), (3,9,6,2),(4),(8,5,11,7),(12,1,10), (3,2),(5,10,7),(8,12,1,4,9),(11,6), (8,1,7),(5,11,2),(12,9),(3,4,10),(6), (5,4,2),(3,11,7),(12,9,10),(8,1,6), (6),(7),(9,10),(3,8,12,5,1,4,11,2), (5,9,10),(8,2),(3,12,11,7),(4,1,6), (4,12,9,2),(8,3,11,10),(5,1,6,7)))
24	2900,00	<b>2304,08</b>	<b>155</b>	((6,11),(9,10,2,7,8,5),(1,3,4),(12)), (6,11),(12,10,5),(9,4,3,8),(1,2,7), (2,4),(1,10),(9,11,3),(12,5,8),(6,7), (6,5,4),(9,10,8),(1,2,7),(12,11,3), (12,7),(9,10,4),(6,2,5,8),(1,11,3), (5,4),(6,10,7),(9,2,8),(12,1,11,3), (6,7,3),(12,8),(1,11,4),(9,5),(2,10), (12,6,5,8),(9,11,3),(1,7,10),(2,4), (6,5),(1,7,10),(9,8),(12,11,2,4,3), (1,2,3),(9,12,6,5,8),(7,4),(11,10), (5,10,3),(12,1,7,8),(6,9,2,11,4)))
25	3000,00	<b>2443,35</b>	<b>156</b>	((9,12),(2,8,1,10,7),0,(6,3,5,11,4), (6,8,11),(3,12),(9,1,7,4),(2,5,10)), (5,4),(3,10),(9,12,11),(2,1,7),(6,8), (6,5),(9,1,10,4),(3,8,11),(2,12,7), (10,11),(3,8),(2,6,1,7,4),(9,5,12), (7,4),(8,5),(1,12),(2,9,3,6,11,10), (1),(2,8,11),(9,12,4),(3,5,10),(6,7), (9,5,10),(6,11),(2,1,12,4),(3,8,7), (9,1,7),(6,3,10),(2,8),(5,12,11,4), (9,11,7),(2,8,3,5),(1,10),(6,12,4), (8,1,5,4),(12,6,2,10),(3,9,11,7)))
26	3091,47	<b>2291,24</b>	<b>156</b>	((2,3,11,5),(8),(7,6,9,4),(1,12,10)), (1,4,5,10),(7,6,12),(8,3,9),(2,11), (11,5),(7,12),(1,9),(2,3,10),(8,6,4), (2,3,4),(9,11),(1,5),(8,7,6,12,10), (2,4),(7,6,12,5),(1,3,11),(8,9,10), (9,4),(3,11),(8,7,6),(2,1,12,5,10), (2,6,5),(8,9),(7,12,10),(3,11),(1,4), (12,5),(8,3,10),(2,1,9,4),(7,6,11), (2,3,12,4),(8,7,6,10),(1,9,11),(5), (6,5),(8,1,3,4),(2,12,9,10),(7,11), (4,5),(1,6,8,7,2,11),(12,9,3,10)))

Tabela Z.12. Wyniki obliczeń przeprowadzonych za pomocą algorytmu SA, związane z poszukiwaniem zbioru punktów kompromisowych w przykładzie przedstawionym w p. 6.1.3

Lp.	Ograniczenie $\hat{U}$ na całkowity koszt przedsięwzięcia [tys. PLN]	Wyniki po optymalizacji		
		Całkowity koszt przedsięwzięcia [tys. PLN]	Czas przedsięwzięcia (najlepsza wartość $C_{max}$ ) [dni robocze]	Permutacja dla najlepszej wartości $C_{max}$
1	2	3	4	5
1	685,38	<b>685,38</b>	<b>291</b>	((8),(12,6),(5),(2,10,9,1,7,11,3,4)), ((8,1,11,12,5,3,9,7,2,6,4,10),(0,0,0)), ((8,12,3,9,4,2),(1,11,5,7,6,10),(0,0,0)), ((8,1,11,12,5,3,7,9,6,4,10,2),(0,0,0)), ((11),(5),(8,1,12,9),(7,3,6,4,10,2)), ((7,3),(9,4),(1,11,12),(5,8,10,6,2)), (0,(3,1),(6),(12,8,7,4,2),(5,11,9,10)), ((5,12,11,9,6),(8,3,4,2),(7,1,10,0)), ((3,11),(7,5,4,1,9,10),(8,12,6),(2)), ((1,6),(3,2),(5,4,8,9),(11,12,7,10)), ((5,4,7),(9,1,6,2),(3,12,10,8,11)))
2	700,00	<b>699,78</b>	<b>291</b>	((0,0),(7,3,11,12),(10,8,4,2,9,5,6,1)), ((7,10,12,5,8,1,3,4,2,9,11,6),(0,0,0)), ((7,5,8,4,2,6),(10,12,1,3,9,11),(0,0,0)), ((7,10,12,5,8,1,3,4,9,11,6,2),(0,0,0)), ((12,3),(10,8,1,4,6,2),(7,9,11),(5)), ((4,6,11,2),(12,5,10,3,8,9),(0,7,1)), (0,(3,1),(6),(7,5,8,4,2),(12,10,9,11)), ((8,4),(12,3,7,11),(1,5,6),(10,9,2)), ((11),(1,9,2),(8,3,12,5,10,6),(4,7)), ((3,6),(12,7,5),(4,8,2),(9,1,10,11)), ((12,6,2),(4,7,1,9),(3,8,5,10,11)))
3	800,00	<b>793,87</b>	<b>281</b>	((7),(9,8,2,4,10,6),(1,3,11),(5,12)), ((1,4,7,12,8,10,3,11,2,9,6),(0,5,0)), ((5,4,8,11,2),(1,7,12,10,3,9,6),(0,0,0)), ((1,5,7,4,12,10,11,8,2,9,6,3),(0,0,0)), ((8),(12,6),(5,10,2,9,11,3),(4,7,1)), (0,(9),(5,7,4,1,12,10,11),(8,2,6,3)), ((4,2),(7,9,3),(1,6),(12,8),(5,10,11)), ((9,6,3),(1,4,7,2,8,11),(5,12,10,0)), ((1,2,11),(7,10,8),(9,5,6),(12,4,3)), ((1,2,5,9),(12,7,4,10),(8,6,11),(3)), ((1,6),(12,11),(4,7,10,2,5,8,9,3)))
4	900,00	<b>897,07</b>	<b>254</b>	((4,5),(3,7,6,11,12),(0,(8,2,1,9,10)), ((3,8,11,2,12,9,10,5,4),(1,6,0,(7))), ((7,2,9,5,8,4),(3,1,11,12,10,6),(0,0,0)), ((3,7,1,11,12,9,10,6,5,8,4),(2),(0,0)), ((1,7,2,10),(9,3,6,8,4),(11,12),(5)), ((2,11),(12),(1,10,3,6,8,4),(7,9,5)), ((1,10,4),(3),(6,8),(7,2,9,5),(11,12)), ((11,8),(12,9),(1,2,7,10,5,6),(3,4)), (0,(11,5),(1,12,7,2,8),(9,3,10,6,4)), ((11,1,9),(7,10,5,6,8),(0,(3,2,12,4)), ((6,4),(1,3,11,5,9,2),(7,12,10,8)))
5	1000,00	<b>987,52</b>	<b>260</b>	((0),(11,6,5,4,10,1),(2,12,3,8),(9,7)), ((11,8,3,2,7,1,4,5),(12,10,6),(0,(9))), ((9,12,7,4,3),(11,8,10,1,5,2),(0,(6,0)), ((11,9,8,12,10,6,7,4,3,5,2),(0,0,(1))), ((11,12),(1,2),(9,8,4,10),(6,7,3,5)), ((12,8),(11,1,7,3,5),(6,4,10,2),(9)), ((4,3,2),(10),(5),(8,6,12),(11,9,1,7)), ((11,9,10,2),(4,3),(1,8,7,12),(6,5)), ((5),(11,7,10,6,1,4,2),(12,9),(3,8)), ((9),(11,6,1,12,3,2),(8,10,7),(4,5)), ((6,4,11,7,9,2),(12,3),(1,10,5,8)))
6	1100,00	<b>1078,05</b>	<b>232</b>	((8,2),(3,5,9,10),(7,11,12),(4,1,6)), ((3,1,4,9,8,11),(5,7),(12),(6,10,2)), ((5,9,12,4,8),(3,6,2,11,10,7),(0,0,(1))), ((3,5,6,1,9,2,11,4,8,7),(0,(12),(10))), ((3,9),(1,11,10,4,8),(6,2,7),(12,5)), ((1,12,4,8,7),(6),(9,2,3,5,10),(11)), ((2,11),(1,3,10,8),(7),(6,12,4),(9,5)), ((2,10,7),(3,11,8),(6,5,4),(1,12,9)), ((3,10,8),(11,6,5),(2,12,9),(1,4,7)), ((10,12),(11,6,4),(2,9,1,5,8),(3,7)), ((1,10,3,12),(6,4,11,7),(9,5,2,8)))

Tabela Z.12. cd.

1	2	3	4	5
7	1200,00	<b>1192,97</b>	<b>237</b>	((2),(5,4),(11,12,3,6,10,1,9,7,8),(0), (11,12,8,5,9,1,7),(3,6),(0),(2,10,4), (2,12,3,5,7),(11,6,10,4,1,8),(0),(9),(0), (11,2,6,12,3,4,9,7,5),(1,8),(0),(10), (6,12,11,7,8),(10,4,1),(2,5),(3,9)), (12,11,3,4),(2,10,9,7),(6,1,8,5),(0), (4),(12,3,7,8),(6,1,5),(10,2,9),(11)), (10,2,7),(11,8),(6,12,1,3),(4,9,5)), (0),(4,12,6,1,2,3,7,5),(11,9,10,8),(0), (12,11,2,7),(4,1),(9,5),(6,3,10,8)), (2,6,1),(4,12,5,10),(11,9,7,3,8)))
8	1300,00	<b>1281,15</b>	<b>213</b>	((9,6,7,10,12),(3),(1),(8,2,4,11,5), (9,12,8,10,7,6),(11),(3,4,5,2),(1)), (1,5,10,2),(3,12,8,6,7),(0),(4,11),(9)), (1,9,4,11,10,3,2,7),(12,8),(5,6),(0), (12,8,3,2,7),(9),(4),(1,5,11,10,6), (1,5,10,6,2),(3),(9,11),(12,4,8,7)), (4,11),(10,3),(6,7),(1,5,12),(9,8,2)), (9,4,8,6,3),(11,2),(12,10),(5,1,7)), (11,9),(10,8,6,3),(1,12),(5,4,2,7)), (1,9,8,6,4,10),(12,2),(0),(11,5,3,7)), (11,12,5,6,8,10,1),(4,7),(9,3,2)))
9	1400,00	<b>1398,04</b>	<b>203</b>	((11,5,4,10,3,7,1),(8,12),(6),(9,2)), (8,10,1,4,7),(12,2),(6),(9,11,3,5)), (12,10,2,1),(6,11,3,4,7),(8),(0),(9,5)), (6,12,11,3,2,1,7),(0),(10,5,4),(8,9)), (8),(3),(12,9,10,11,5,4,1,7),(6,2)), (6,12,10,1),(0),(8,9,3,5,7),(11,2,4)), (10,2,7),(9,3,1),(6,5,4),(8,11),(12)), (9,11,3),(0),(8,6,5,4,7),(12,10,2,1)), (4,3),(5,11,10,1,7),(6,8),(9,12,2)), (6,12,5,3,10,7),(2),(0),(11,8,9,4,1)), (5,6,11,9,10,2),(12,8,7),(3,4,1)))
10	1500,00	<b>1487,06</b>	<b>203</b>	((1,11,10,8),(3,2,5),(4,7,9,12),(6)), (3,8,6,12,2,9),(11,5),(4,7,10),(1)), (1,8,6,2),(3,11,7,12,9),(4),(10),(5)), (3,1,4,7,12,2),(8,6,9),(5,10),(11)), (8),(5,10,4,2),(1,6,7,12,9),(11,3)), (3,5,10,9),(8,2),(1,11),(6,4,7,12)), (8,12),(10,9),(1,3,6),(5,4),(11,7,2)), (10,6,12,9),(11),(3,5,4),(1,8,7,2)), (1,10),(3,11,8,4,7,12,2),(5,6),(9)), (0),(10,3,8,11),(5,1,6,9),(4,7,12,2)), (3,1,4),(8,11,12,6,9),(5,10,7,2)))
11	1600,00	<b>1591,81</b>	<b>192</b>	((4,1,5),(2),(3,10,12,11),(9,8,7,6)), (9,12,8,11,10,7),(0),(3,2,1,6),(5,4)), (9,4,7),(3,12,6,11,10),(5,1),(2),(8)), (3,5,12,9,1,4,10),(6),(2),(8,11,7)), (12,9,4),(3,6),(2,11,10),(5,8,1,7)), (2,12,6,1,11,7,10),(5,9,4),(3),(8)), (3,8),(6,4,10),(12,1,7),(2,9),(5,11)), (5,2,1,7),(12,11),(8,6,4,10),(3,9)), (8,2,3,12,1,11),(6,4,9,7,10),(0),(5)), (12,2,7),(5,9),(6,8,4,1,11),(3,10)), (7,11),(8,5,1,10),(12,6,2,9,4,3)))
12	1700,00	<b>1697,42</b>	<b>190</b>	((1,5,6),(12,7),(8,3,2),(10,11,4,9)), (10,2,7,9),(12,11),(4,3),(8,5,1,6)), (8,4,5,6),(12,1,7,3),(10,9),(11),(2)), (8,11,4,5,3),(12,1,7),(2,9),(10,6)), (0),(8,11,2,5),(12,10,9,6),(1,4,7,3)), (12,7),(8),(4,5),(10,2,1,11,9,6,3)), (1,5,3),(2,9),(10,4),(12,7,6),(8,11)), (2,1,7),(10,4,9,12,3),(11),(8,5,6)), (10),(2,7,12,8,3,6),(11,5,4),(1,9)), (10,11,9,1),(4),(2,5,8),(12,7,3,6)), (1,8,7,5,3),(4,11),(9,10,12,2,6)))
13	1800,00	<b>1735,75</b>	<b>186</b>	((3),(6,11,8,5,12,10),(2),(9,7,1,4)), (2,1,10,8),(3,11,5),(9,7,12,4),(6)), (9,5,8),(3,1,10,6),(2,4),(12,11),(7)), (3,2,7,10,11,6),(12),(1,4),(9,5,8)), (2,4,6),(12),(9,3,8,10,11),(1,7,5)), (7,8),(9,2,1,12,5,10,6,11),(3),(4)), (9,4),(7,5,11),(2,12),(1,10,6),(3,8)), (9,7,4),(12,5,3),(2,1,8,11),(10,6)), (8),(7,5,2,3,11),(0),(1,9,4,10,12,6)), (9,4),(2,1,12,6),(8,10,7),(5,3,11)), (3,8,10),(9,2,11),(4,5,1,12,7,6)))

Tabela Z.12. cd.

1	2	3	4	5
14	1900,00	1825,25	180	((6,4,10,5),(1,12,3,9),(11,8,2,7),(0), (1,3,7,8,12),(6,10,2,5),(4,9),(11)), (1,9,2),(11,10,7,12),(6,8),(3,5),(4)), (11,10,5,7,12),(3,9,8),(4,1,2),(6)), (4,8,7,12),(11,9),(6,10),(3,1,5,2)), (3,4,7),(6,11,9,2),(5),(1,10,8,12)), (1,8),(6,10),(3,2,12),(11,9,7),(4,5)), (11,5,8),(3,9,2,12),(4,10),(6,1,7)), (9,5,7),(6,10,3),(0),(1,4,11,2,8,12)), (11,4,8),(0),(5,10,7),(6,9,1,3,2,12)), (11,6,4,8,10,7),(9),(1,3,5,2,12)))
15	2000,00	1998,92	173	((1,9,8,7),(3,5),(2,10,4,12,6,11),(0), (1,8),(7,10,6),(2,12,4,5,11),(3,9)), (1,9,6),(3,10,11),(2,12),(8,5),(7,4)), (3,1,7,11),(8,12),(2,9,10),(5,6,4)), (10,6,4),(2,1),(3,7,9,5),(8,12,11)), (2,5),(3,1,7,10,9,11),(4),(8,12,6)), (1,10,6),(2,12,9,4),(7,11),(8),(3,5)), (2,3,10,6),(12,1,5),(7,9,4),(8,11)), (3,2,9),(1,11,4),(12,5),(7,8,10,6)), (0),(7,12,10,4),(2,3,8,9,5),(1,11,6)), (7,2,12,5,4),(1,8,3,9,10,6),(11)))
16	2100,00	2056,66	168	((8,12,6),(9,1,2),(11,5),(4,3,10,7)), (2,7,4,3),(11,5),(9,1),(8,12,10,6)), (8,1,6),(9,7,10),(2),(5,12,3),(11,4)), (8,9,12,3),(2,1,6),(11,7,10),(5,4)), (11,5,1,6),(9,7,4,3),(2,8,12,10),(0)), (9,1,8,4,12),(11),(5,2,7,3),(6,10)), (2,8),(7,12),(5,4,3),(11,9,6),(1,10)), (2,12,6),(5,4,8),(9,7,3),(11,1,10)), (9),(4,3),(7,10),(2,11,5,1,12,8,6)), (11),(2,5,12,6),(9,1,3,8),(4,7,10)), (5,4,9,7,3,8),(6),(2,1,12,11,10)))
17	2200,00	2167,89	172	((9,5,3,11,7),(2,10,1,12,4),(6),(8)), (8,3,11),(6,10,12,1,4),(9,5,7),(2)), (6,10),(9,11,12),(8),(3,5,4),(2,7,1)), (3,5),(2,12),(9,11,7,1),(8,6,10,4)), (8,3,5,7,12,4),(2,11,1),(9),(6,10)), (3,2,5,10,7,12,1),(8,6,11,4),(9),(0)), (8,5,1),(6,7),(2,12),(9,11,4),(3,10)), (8,5,10),(11,12),(9,3,2,1),(6,7,4)), (9,6,8),(2,10,12,4),(11,5,3,1),(7)), (10,12,4),(9,11,7),(6,2,8,1),(5,3)), (7,11,5,9,10),(8,12,1),(2,6,3,4)))
18	2300,00	2260,91	175	((4),(6),(12,3,1,7,10,2,8),(5,9,11)), (5),(6,11,3,10),(4,9,8),(12,7,2,1)), (4,3),(6,9,10),(12,11,1),(5,7,2),(8)), (12,4,9),(6),(5,8,7,2,10),(11,3,1)), (8,9,2),(12,7,3,1),(5,4),(6,11,10)), (11,1,3,10),(5,6,8,7),(12),(4,9,2)), (6,8,3),(12,5,9,10),(4,7,1),(11),(2)), (5,7,2),(4,9,1),(11,3),(12,6,8,10)), (5,8,1),(12,11,7,2),(9,6,10),(4,3)), (12,9,3),(2,10),(7,8,4,1),(11,5,6)), (9,7,12,1,10),(5,8,2,4,6),(3,11)))
19	2400,00	2104,08	174	((5,9,8,11,2),(6),(7,4,3,1,12),(10)), (7,12,4),(10),(6,8,2,9,1),(5,11,3)), (10,4,2),(7,11,3),(6),(5,8,9,1),(12)), (10,12,3),(6),(7,11,8,2),(5,4,9,1)), (7,4,3),(6,5),(12,8,2),(10,11,9,1)), (6,8,3,1),(7,12,9),(10,2),(5,11,4)), (6,11,2),(5,8),(12,9),(10,3),(7,4,1)), (10,6,8),(5,3),(12,11,9,1),(7,4,2)), (5,6,2),(0),(10,7,11,4,3,12,9,8,1)), (0),(5,10,9,2),(7,6,4,12,1),(11,8,3)), (10,4,7,8,2),(3),(5,6,12,11,9,1)))
20	2500,00	2265,87	167	((11,12),(1,4,10,7,9),(3,2,5,6,8),(0), (12),(3,5,6,10),(11,7,4,1),(2,8,9)), (2,4),(11,10,1),(12,9),(3,5),(8,6,7)), (11,2,6,1),(8,9,10),(3,5),(12,4,7)), (8,4,7),(12,2,6),(11,5,9,10,1),(3)), (12,6,10),(7),(2),(3,11,8,5,9,4,1)), (5,4,7),(8),(3,2,9,10),(11),(12,6,1)), (3,4,10),(2,9),(11,5,12,1),(8,6,7)), (5,7),(3,12,6,9,10),(11,2),(8,4,1)), (11,2,6,4),(12,8,9,1),(10),(3,5,7)), (5,3,2,11,8,6,1),(4,10),(12,9,7)))



Tabela Z.12. cd.

1	2	3	4	5
21	2600,00	<b>2147,50</b>	<b>171</b>	((11,8),(12,9),(7,4),(6,3,1,5,10,2)), (4),(6,10,3,2),(11,9,7,8),(12,1,5), (11,9,3),(4,10,7),(6,5,2),(12),(1,8)), (1,10,2,7),(11,5),(6,9,8,3),(12,4), (4,5,9,2),(10,7),(1,8,3),(12,6,11)), (11,6),(10,9,8,3,7),(4,5,2),(12,1)), (12,8),(11,9,7),(4,10,2),(5,3),(1,6)), (1,6,2),(10,3),(12,4,5,8,7),(11,9)), (0,(6,9,8,7),(4,12,10,3),(5,1,11,2)), (1,12,3,7),(4,10,5,2),(11),(6,9,8)), (11,8,6,7),(10,1,4),(5,12,9,2,3)))
22	2700,00	<b>2339,92</b>	<b>172</b>	((0,(4,10,3,11,2,5),(6,9),(8,7,12,1)), (4,5),(9,3,11,1,10),(6),(8,2,12,7)), (4,3),(6,11,7),(9,12),(5,1,10),(8,2)), (6,3),(5,12,7,10),(9,11,1),(8,4,2)), (9,2,11,12,7,5),(6,4),(8,1,10),(3)), (3),(2,11,1),(5,10),(8,6,4,9,12,7)), (9,1,10),(8,4,11,5),(6,12),(3,7),(2)), (9,3,7),(8,1),(6,2,12,10),(4,11,5)), (2,9),(12,5),(8,6,4,3,1),(11,7,10)), (8,9,1,7),(3,5),(6,11,4,12),(2,10)), (1,3,2,5),(7,11),(6,8,4,12,9,10)))
23	2800,00	<b>2126,18</b>	<b>177</b>	((2,12,5,3),(4,1),(9,6),(11,10,8,7)), (4,3,7),(10,6,5),(9),(11,8,2,1,12)), (10,2),(4,1,6),(3,5,12),(9,8),(11,7)), (4,7,1,6),(11,8,5),(9,3),(10,2,12)), (7,5,12,6),(4,10,8),(9,3,2,1),(11)), (11,7,1,12),(9,4,6),(8),(3,10,5,2)), (10,7),(8,6),(9,4,5),(11,12),(3,2,1)), (3,2,12),(8,6),(10,4,1),(9,11,5,7)), (9,3,10,1),(4,2,12,7),(11),(5,8,6)), (11,5,6,9),(4,2,12),(8,3,1),(10,7)), (11,4,9),(10,12,6,7),(8,5,2,3,1)))
24	2900,00	<b>2329,46</b>	<b>169</b>	((0,(12,3,4,6,2,11),(1),(9,7,10,5,8)), (7),(3,12,11,5),(9,4,6),(1,8,10,2)), (7,10,4),(3,6),(9,11,5),(1,12),(8,2)), (12,10),(8,5),(9,3,6),(1,7,11,2,4)), (7,8,5),(9,1,3,2,4),(10),(12,6,11)), (0,(9,12,6,8),(7,3,10,11,5,2,4),(1)), (1,2),(7,8),(9,12,5,4),(3,11),(6,10)), (1,8,10),(3,11,4),(9,12,5),(7,6,2)), (9,11,8),(1,6),(3,7,2),(12,5,10,4)), (12,1,9,11,5),(8,2),(4),(3,7,6,10)), (9,1,6,8,5,10),(7,2),(12,3,11,4)))
25	3000,00	<b>2329,45</b>	<b>172</b>	((3,7),(1,9,10,4),(8,12,5,2,11,6),(0)), (8,5,11),(3,12,10,2),(1),(9,7,6,4)), (12),(1,11,6),(8,10),(3,5,2),(9,7,4)), (1,2),(5,10,7,6),(8,12,9),(3,11,4)), (3,1,12,2,6),(9),(8,7,10,4),(5,11)), (1,5,2,7,4),(3,12,11,9,6,10),(0),(8)), (5,7),(12,11,6),(1,9,10),(3),(8,2,4)), (1,8,7,11),(12,10),(9,6),(3,5,2,4)), (3,7,10),(1,8,12),(5,9,6),(2,11,4)), (2,10,4),(8),(5,1,3,7,6),(12,9,11)), (1,12,6,11),(8,9,10),(5,3,2,7,4)))
26	3091,47	<b>2268,23</b>	<b>176</b>	((9,7,4),(1,8,6,2,5),(3,11,10),(12)), (12,4,3),(11),(9,8,5),(1,6,7,2,10)), (12,5,2),(11,3),(9,6),(8,7,10),(1,4)), (4,5,10),(11,1,6),(9,8,3),(12,7,2)), (9,3,4,2),(12,8,7),(11,1,6,5,10),(0)), (4,6),(11,7,10,2),(12,8,1),(3,9,5)), (8,5),(12,4,2),(11,9),(3,7,10),(1,6)), (12,3,6,2),(8,1),(11,4,7),(9,5,10)), (11,1,12,9,8),(0),(4,7),(3,6,5,2,10)), (4,11),(3,12,7,2),(8),(1,6,5,9,10)), (5,12,2),(4,3,8,10),(11,7,6,9,1)))

Tabela Z.13. Wyniki obliczeń przeprowadzonych za pomocą algorytmu GS, związane z poszukiwaniem zbioru punktów kompromisowych w przykładzie przedstawionym w p. 6.2.3

Lp.	Ograniczenie $\hat{U}$ na całkowity koszt przedsięwzięcia [tys. PLN]	Wyniki po optymalizacji		
		Całkowity koszt przedsięwzięcia [tys. PLN]	Czas przedsięwzięcia $C_{max}$ [dni robocze]	Permutacja dla najlepszej wartości $C_{max}$
1	2	3	4	5
1	1795,30	<b>1795,30</b>	<b>650</b>	((0,0),(1,2,10,7,4,5,9,8,6,3,11,12)), ((6,5,10,12),(2,3,8,9,11),(1,7,4)), (11),(5,6),(3,4,9,12),(1,2,8,10),(7)), ((12),(1,2,3,4,5,6,7,8,9,10,11)), (0,0,0,0),(1,2,3,4,5,6,7,8,9,10,11,12)), (0,0,0),(1,2,3,4,5,6,7,8,9,10,11,12)), (1,7,2,5,8,9,10,11,6,3,4,12),(0,0,0), (0,0),(1,2,5,7,8,9,10,11,6,3,4,12)), (1,2,5,7,8,9,10,11,6,3,4,12),(0,0,0), (0,(1,2,5,7,8,9,10,11,6,3,4,12),(0,0,0,0)), (0,0,0),(1,2,5,7,8,9,10,11,6,3,4,12)), (0,(1,2,5,7,8,9,6,10,11,3,4,12),(0,0,0)), (1,2,5,7,8,9,10,11,6,3,4,12),(0,0,0)), (1,2,5,7,8,9,6,10,11,3,4,12),(0,0), (2,6),(12),(1,5,7,8,9,10,11),(3,4))
2	2200	<b>2184,50</b>	<b>521</b>	((0,(3),(6,1,2,4,5,7,8,10,11,12,9)), (6),(3),(1,2,4,5,7,12,8,9,10,11)), (4),(5),(3),(1,2,12,8),(6,7,10,9,11)), (0,(1,2,3,4,5,6,7,12,8,9,10,11)), (5),(2,4),(3,1,12),(8),(6,7,10,9,11)), (0,(5,4,3,11),(6),(2,12,8,1,10,7,9)), (6,3,5,10,7,4,1,11,2,9),(12,8),(0,0), (10,7,4,1),(3,5,8,6,11),(12,2,9)), (3,12,5,8,10,6,7,4,1,11,2,9),(0,0,0), (10,3,1,4),(5,6,12,7,8,2,11,9),(0,0,0,0)), (4,2),(3,5,6),(8,7),(10,11,12,1,9)), (7,4,3,11,6,2),(12,10,5,8,1,9),(0,0,0,0)), (3,5,10,8,6,4,11,12,2,9),(7),(1,0,0)), (7,10,5,6,11,3,2,12),(0,(8,4,1,9)), (2),(4,3),(7,5,6,12,9),(10,8,1,11))
3	2600	<b>2596,50</b>	<b>454</b>	((0,(5),(8,2,4,3,6,7,10,12,9,1,11)), (0,(5,9,3,2),(8,1,4,6,7,10,12,11)), (0,(5),(4,9,10),(2,1,11),(8,3,6,12,7)), (4),(5,12,8,3,9,10,2,1,11,6,7)), (1),(8,2,4),(7,5),(3,6,11,10),(12,9)), (2,11,9),(8,1,4,6),(0,(7,10,12,5,3)), (8,4,2,9,11,6,10,7),(5),(0,(3,12,1)), (8,4),(6),(5,12,9,1,2,3,11,10,7)), (5,12,4,8,9,3,11,1,6,2,10),(7),(0,0,0)), (4,1,3,9),(8,11,2,5,6,10,7,12),(0,0,0,0)), (12,6,2),(4),(0,(5,8,9,3,11,1,10,7)), (12,8,3,5,11,9,2),(4,1,6,10,7),(0,0,0,0)), (5,12,4,9,8,3,1,2),(6),(10,7),(11)), (4,12,5,8,3,1,10,2),(9),(11,6,7)), (3,4),(11,8,1,9,7),(5,2),(6,10,12))
4	3000	<b>2999,00</b>	<b>375</b>	((0,(12,1),(7,5,6,2,3,4,8,9,10,11)), (12),(7),(1,3,5,6,2,4,8,9,10,11)), (0,(3),(7,5,4),(1,10),(12,6,2,8,9,11)), (0,(3,1,6,7,5,10,2,12,8,4,11,9)), (10),(0,(1,6),(7,5,8,3),(12,4,2,9,11)), (1),(6,11,12,2),(7,5,10),(8,9,3,4)), (7,5,3,10,9),(12,6,8,11),(0,(1,2,4)), (1,3,10,9),(12),(7,6,5,2,4,8,11)), (12,7,1,6,5,4,2,10,11,9),(3),(0,(8)), (6,12,7,3,10,2),(1,5,4,8,9,11),(0,0,0,0,0)), (8,2,9),(1,4),(12,3),(7,6,5,10,11)), (7,6,1,3,5,4,2,9),(12,8,10,11),(0,0,0,0)), (12,7,1,6,5,4,8,10,2,9),(0,(3),(11)), (6,12,1,3,5,10,2),(7,8),(4,11,9)), (6,10),(1,11,9),(7,3,2),(12,4,5,8))
5	3400	<b>3381,00</b>	<b>333</b>	((9),(5),(4,11,1,8,12,2,10,3,6,7)), (0,(5,2),(12,9,1,8,3,10,4,6,7,11)), (2),(10,6),(12,4),(9,1,8),(5,3,7,11)), (2,4),(5,3,1,12,10,8,7,9,6,11)), (5,2),(4,12),(3,11,7),(8,9,10,1),(6)), (7,11,1),(0,(3,6),(5,9,8,2,4,10,12)), (9,1,8,7,11),(10,6),(5),(12,2,4,3)), (9,2,1,4,8,6,11),(12,3),(5,10,7)), (5,2,10,1,3,12,6,8,11),(7),(0,(9,4)), (5,2,4,3,1,12),(10,8,7,9,6,11),(0,0,0,0,0)), (4,9,12,3,1),(6),(10),(2,5,7,8,11)), (5,2,10,7,12,8),(9,1,4,6,3,11),(0,0,0,0)), (2,5,7,9,1,12),(4,6,11),(3),(10,8)), (4,5,6,9,3,12),(7,1),(2,10,8,11)), (10,5,3),(2,7,12),(4,8,9,1),(6,11))

Tabela Z.13. cd.

1	2	3	4	5
6	3800	3785,00	303	((0,(6),(7,1,10,2,3,4,11,5,8,9,12)), (0,(7,2,3),(6,10,1,11,4,5,8,9,12)), (0,(3),(6,1,11,5),(7,2,4,8),(10,9,12)), (7),(1,2,10,6,9,3,5,4,8,11,12)), (0,(1),(2),(3,7,5,8,9),(6,10,4,11,12)), (0,(8),(6,2,10,3,7,1),(11,4,5,12,9)), (6,11,9,8),(1,3,12),(7,2,4,5),(10)), (1,5),(2,3,9,8),(7,6,10,11,4,12)), (1,11,7,12,9,8),(3),(10,2),(6,4,5)), (10,6,3,4,1),(2,11,5,7,12,8,9),(0,(0,(0,0)), (0,(10,2),(6,3,1),(11,4,7,5,9,12,8)), (10,4,6,1,5,8),(2,11,3,7,9,12),(0,(0,0)), (11,2,6,9),(4),(10,5,1),(3,7,12,8)), (11,10,3,2,1,8,9),(5),(4,7,6,12)), (2,1),(11,7,6,9),(3,5,8),(10,4,12)))
7	4200	4166,60	309	((1),(7,2,3),(8,4,11,12,9,5,10,6)), (8),(7,2),(1,3,12,9,11,5,10,4,6)), (4),(8,11),(1,3,5),(7,2),(12,9,10,6)), (0,(1,2,3,7,8,4,11,12,5,6,10,9)), (0,(2,10,1),(3,5,7),(8,9,12,4,6,11)), (0,(0,(8,2,7,1,3),(5,12,4,9,10,6,11)), (7,3,10,5),(12,9,4),(1,11),(8,2,6)), (3,4,6),(8,2),(12,7,1,11,9,10,5)), (12,1,3,9,4,10,7),(5),(11),(8,2,6)), (12,2,4,3),(8,7,10,6,5,1),(0,(9),(0,(11)), (11,7),(8),(2,4,6),(12,1,9,3,10,5)), (11,1,10,7,5),(12,9,3,6,4),(8,2),(0,0)), (1,12,2,8,7,3),(10,5),(11,4,6),(9)), (2,11,7,6,3),(8,12,10,1,5),(9,4)), (9,12,11,7),(8,4,3),(6,1),(2,10,5)))
8	4600	4543,80	296	((2),(5),(6,1,3,10,7,4,8,11,12,9)), (0,(6,2),(5,1,3,8,10,7,11,12,4,9)), (0,(2),(6,8,11),(5,3,7,9),(1,10,12,4)), (6),(2,5,10,12,8,3,11,1,4,7,9)), (5,1,8,9),(2,7,4),(0,(11,3,12),(6,10)), (7,9),(5,1,11,8,10,3),(12),(2,6,4)), (6,2,12,11),(10,9),(1,3,7),(5,8,4)), (2,12,11),(6,3,7,4),(5,1,10,8,9)), (5,2,12,6,9),(10,11),(1,3),(8,4,7)), (6,2,5,10,1,12),(8,11,4),(3),(7),(9),(0)), (10,12,6),(2,8),(3,1),(5,11,4,7,9)), (10,5,8,6),(1,2,12,3,11,7,9),(0,(0,(4)), (5,10,11,12,7,9),(2),(8,1,6),(3,4)), (10,8,1,12,7,6),(2,3),(5,11,4,9)), (10,7),(2,3,4),(5,8,1,6),(11,12,9)))
9	5000	4857,80	286	((0,(11,1,2,5),(8,3,4,7,6,9,12,10)), (5),(11,1,2,4),(8,3,7,6,9,12,10)), (0,(1,5),(3,4,9),(11,7,2),(8,6,12,10)), (0,(1,3,4,8,2,11,9,6,5,7,12,10)), (0,(3),(1,2,5),(7,12),(8,6,9,11,4,10)), (0,(1,9,4),(3,2,7),(8,6,5,11,12,10)), (8,3,2,9),(1,6),(7,4,10),(11,12,5)), (7,6,10),(8,12,2,9),(1,11,3,4,5)), (11,1,12,10,9),(8),(7,4,2,5),(3,6)), (1,3,2,5,9),(8,4,7,11),(0,(6),(0,(12,10)), (4,8),(6,5),(7,12,11,10,9),(3,1,2)), (8,12,2,9,1),(7,4,10,5),(11),(3),(6)), (1,8,11,9),(3,5),(6,2),(7,4,12,10)), (8,2,5,11,1,9),(3,4),(6,12,7,10)), (2,7),(6,8,10,9),(3,12,5,1),(4,11)))
10	5400	5176,90	293	((0,(3,2,6,5),(8,4,9,7,10,1,11,12)), (2),(8,4,1),(3,9,5,6,10,11,12,7)), (2,1),(3,10,11),(4,12,7),(9),(8,6,5)), (8,3),(2,1,9,6,5,4,10,7,11,12)), (3,10),(8,6),(0,(4,2,5,1),(9,7,11,12)), (0,(3,8,7,10),(6,4,1),(2,9,5,11,12)), (3,4,10,7),(2,1,11),(8,6),(9,12,5)), (8,4,6),(3,12,1,10,2),(9,5,11,7)), (8,6,1,10,7),(3,9,4,12,2),(11),(5)), (3,6,1,10),(9,8,5,4,12),(0,(2),(0,(7,11)), (12,4,10),(8,6),(1,2),(3,9,5,11,7)), (3),(9,10,7,12,2),(1),(11),(6,8,5,4)), (3,4,10),(5,2),(8,9,1,11),(6,12,7)), (9,5,11,2,12,1),(3,6,7,4,10),(8)), (9,8,5,2),(6,4,1),(3,7,10),(11,12)))
11	5800	5796,80	273	((6),(1,4,3),(11,2,7,8,10,5,12,9)), (2),(1,3,4),(11,6,7,8,10,5,12,9)), (5),(2,4),(6,8),(11,3,10),(1,7,12,9)), (0,(1,2,3,4,11,6,7,8,10,12,5,9)), (11,2),(3),(4),(1,5,7,12),(6,8,10,9)), (0,(0,(1,3,2,10,5),(6,4,8,11,7,12,9)), (11,3,8,9),(6,10),(2,4,12,5),(1,7)), (1,6,3),(12,8),(2,4,11,10,5,7,9)), (11,10,7),(6,4,12,5),(1),(2,3,8,9)), (1,6,4,2),(3,5,7,12),(0,(11),(8),(10,9)), (4),(3,5,7),(1,8),(2,6,11,12,10,9)), (0,(2,1,4,11),(3,7,8),(6,10),(5,12,9)), (4,3,7),(11,10,5),(1,2,8),(6,12,9)), (7,3),(6,4,11,5),(1,2,10,8,12,9)), (4,8,7),(6,1,11,5),(2,3),(10,12,9)))

Tabela Z.13. cd.

1	2	3	4	5
12	6200	6197,40	256	((3),(11,6),(12,7,4,5,8,10,2,1,9)), ((11,6),(12,7),(3,4,5,8,10,2,1,9)), (3),(11,2),(7,8),(12,4,5,1),(6,10,9)), (8),(3,6,11,7,12,4,2,10,5,1,9)), (7),(5),(3),(6,12,4,11,8),(10,2,1,9)), (4,2),(6,3,5,12,8),(7,10,1),(11,9)), (7,4,2),(6,10),(11,5,1),(12,3,8,9)), (3,4,10,1),(12,11,2,9),(7,6,5,8)), (11,10),(6,7,2),(3,8),(12,4,5,1,9)), (12,5,2),(6,4,8),(0,0),(3,11),(7,10,1,9)), (11,10,8),(12,3,4),(6,5),(7,2,1,9)), (9),(3,2,5,8,1),(12,4,10,6),(7),(11)), (12,6,4,9),(7,11,5),(2,1),(3,10,8)), (7,12,3,2,8),(11,4,5,1),(6,10,9)), (4,1),(3,6,2),(12,11,5,9),(7,10,8))
13	6600	6373,10	265	((2),(8,3),(5,12,1,4,6,7,11,9,10)), (3,6),(2,1,4),(8,12,5,11,7,10,9)), (12),(2),(3,4),(5,6,7,9),(8,1,11,10)), (0),(1,2,3,4,6,8,12,5,11,7,10,9)), (0),(11,10),(4,3),(12,6,5,8),(2,1,9,7)), (0),(3,1),(4,2,6,8,12),(5,11,7,9,10)), (8,11,10),(12,5,4,9),(3,6),(2,1,7)), (8,3,1,4,6),(12),(2,5,11,7,10,9)), (3,8,1,6,9),(5,4),(2,7),(12,11,10)), (2,1),(3,4),(6,12),(8,9),(5,7),(11,10)), (5),(2,4,6),(12,1,11,9),(3,8,7,10)), (3,4,6),(1,12),(2,8),(5,7,11,9),(10)), (5,7,11,9),(3,1),(4,6),(12,2,8,10)), (5,8,10),(2,7,1,4,11,12),(3,6,9)), (3,8,6,12),(5,11),(2,1,4,9),(7,10))
14	7000	6403,20	289	((3),(4,1),(2,6,7,10,12,8,5,9,11)), (2),(4,1,6),(3,10,5,9,8,12,7,11)), (10),(4,1,8),(2,6),(3,5,9),(12,7,11)), (0),(1,2,7,3,8,5,10,11,12,4,6,9)), (2,1),(3,4,10),(6,5,7),(11,12),(8,9)), (10,6),(3,5),(1,2,8),(7,4,12,9,11)), (4,6,8,9),(1,12,5),(3,10,11),(2,7)), (7,8,11),(4,1,10,5,9),(3,2,12,6)), (10,7,8,11),(4,1,6,5),(2,9),(3,12)), (2,4,11),(3,10),(8,5),(6,1,7),(9),(12)), (6,1,9),(7,5),(3,12,4),(2,10,8,11)), (4,1),(6,8,5),(3,2,10),(9),(12,7,11)), (7,8,5),(10,11),(12,6,4),(3,2,1,9)), (2,8,6,4),(5,1,9),(12,10,3,7,11)), (8,6,5),(10,4,9),(12,3,7),(2,11,1))
15	7400	6493,40	272	((0),(3,6,1),(11,9,8,4,10,2,5,7,12)), (3),(1,2,4),(11,9,6,8,10,5,7,12)), (1,2),(11,4),(6,10,7,12),(3),(9,8,5)), (12),(1,2,3,4,11,9,6,8,10,5,7)), (0,0),(1,10),(3,6,5,12),(11,4,2,9,8,7)), (0),(9,2,7),(6,11,4,8,12),(3,1,10,5)), (3,4,2,12,7),(11,1),(6,10),(9,8,5)), (3,4,2,12,7),(9,8),(11,6,1,10,5)), (3,4,2,12,7),(9,8),(11),(6,1,10,5)), (1,8),(6,11,5),(4,7),(0),(10,12),(3,9,2)), (4,2),(9,11,8,5),(6,12),(3,1,10,7)), (6,2,1,4),(11,7),(0,0),(9,3,10,8,12,5)), (6,8,4,7),(9,11),(1,2,12),(3,10,5)), (1,2,8,4),(12,3),(9,6,10,11,7,5)), (1,8,4),(6,2,5),(9,10,12,3),(11,7))
16	7800	7546,20	273	((0),(11,6,5),(2,7,9,1,4,8,3,10,12)), (2),(6,7),(11,5,9,8,1,4,10,12,3)), (3),(1),(2,5,10),(11,7,4),(6,9,8,12)), (0),(1,2,4,5,6,7,10,11,3,8,9,12)), (0,0),(6),(2,5,1,10),(11,7,4,8,9,3,12)), (0),(1),(7,2,5,10),(11,6,4,8,9,3,12)), (11,2,12,1),(6,8),(7,4,10),(9,5,3)), (11,7,9),(4,1,12),(6,2,8,5,10,3)), (4,1),(6,2),(7,9,8,10,12),(11,5,3)), (7,6),(11,5),(2),(1,9),(8,3),(4,10,12)), (6,1),(11,4,7,10),(8,12),(2,9,5,3)), (2,8),(0),(4,11,5,10),(7,9,1,12),(6,3)), (11,1,10),(6,5),(2,7,8,12,9),(4,3)), (7,10,9),(4,8,5,1),(2,11,6,12,3)), (2,6,1,12),(11,8,5,9),(7,10),(4,3))
17	8200	7154,90	286	((11,5),(6),(1,10,2,3,7,4,8,9,12)), (2),(6,5,7),(10,1,9,8,3,11,4,12)), (10),(0),(7,2),(5,3,4),(6,1,11,9,8,12)), (0),(1,7,5,11,6,4,10,3,8,2,9,12)), (0),(2),(5,3),(1,7,4,8),(10,6,11,9,12)), (4,2),(5,11),(9,1),(6,10,3,7,8,12)), (5,9,12,1),(11,4,8),(10,3,2),(6,7)), (5,3,4,8,1),(7,9),(10,6,11,2,12)), (3,7,2),(11,4),(10,5,9,12,1),(6,8)), (11),(10,5),(4),(3,7,9),(6,2,8,1),(12)), (3,5),(10,11,4),(7,9),(6,2,8,1,12)), (5,4,1),(6,7,10),(2,3),(11),(9,8,12)), (10,11,4,9),(3,5,1),(6,8),(7,2,12)), (11,4,5),(3,8,1),(6,10,7,2,9,12)), (7,5),(11,3,8,1),(10,4),(6,2,9,12))

Tabela Z.13. cd.

1	2	3	4	5
18	8600	<b>7791,70</b>	<b>280</b>	((9),(1,2,5),(12,3,4,8,7,10,6,11)), ((1),(4,2),(12,9,3,5,6,10,8,7,11)), ((10),(1,2),(9,5),(3,6),(12,4,8,7,11)), ((0),(1,2,3,4,5,6,7,8,9,10,11,12)), ((0,0),(1),(3,2,6,10),(9,4,5,8,12,7,11)), ((0,0),(1,5,9,10),(4,3,2,8,6,12,7,11)), ((4,5),(12,1,2,8),(3,10,7),(9,6,11)), ((3,8,7),(4,5,10,11),(12,1,9,2,6)), ((8,7,10,11),(4,5),(1,2),(12,3,9,6)), ((0,9,6),(3),(12,10),(4,5,1),(2,8,7,11)), ((12,3,2),(4,8),(6,10),(1,9,5,7,11)), ((2,6),(9,12,1,10),(4),(5),(3,8,7,11)), ((8),(1,9,5,11),(3,2),(12,4,6,7,10)), ((4,3,2,1),(12,5,1,6),(9,8,7,10)), ((8,1,10),(4,5,9,6),(2,3),(12,7,11))
19	9000	<b>7041,30</b>	<b>269</b>	(((1),(10,5,2),(8,4,3,12,6,11,7,9)), ((5,4),(10),(8,1,3,12,6,11,2,7,9)), ((1,6),(10),(4,2,7),(5),(8,12,11,3,9)), ((0),(1,4,8,5,11,12,10,6,9,3,7,2)), ((0),(8),(2,1),(5,6,11,7,9),(4,10,3,12)), ((2,12),(4,1),(5,10,6,7,3,9),(8,11)), ((4,1,2,7),(10,11),(12,6),(8,5,3,9)), ((8,4,6,11),(12,1),(10,5,2,3,7,9)), ((5,6,2,9),(1,11,7),(8),(10,4,12,3)), ((4,3),(8,12),(11),(1),(10,5,2),(6,7,9)), ((10),(5,2,1),(12,6),(4,8,11,3,7,9)), ((0),(6),(4,8,12,11),(1,10,5),(2,3,7,9)), ((5,6,7,1),(12,3,9),(10,8,2,11),(4)), ((10,2,11),(8,6,1,9),(4,12,5,7,3)), ((8,6,11),(5,3,9),(10,2,1),(4,12,7))
20	9400	<b>7023,60</b>	<b>285</b>	(((12,6),(9),(1,2,4,11,8,10,3,5,7)), ((0),(6,1,4,2),(9,11,12,8,3,10,5,7)), ((10),(4),(6,1),(9,11,3,5),(12,8,2,7)), ((0),(1,2,3,9,6,11,12,5,4,8,10,7)), ((3),(8),(6,1,9),(11,7,10),(12,4,2,5)), ((9),(4,7),(1,6,11,3),(8,12,5,2,10)), ((6,11),(1,4,2),(12,10,7),(9,8,3,5)), ((6,2,7),(1,4),(12,9,8,10,11,3,5)), ((12,6,1,8,3),(11,7,5),(9,4,2),(10)), ((7,4),(1,11),(9,2,5),(3),(6,12),(8,10)), ((12,6,4,2,3),(9,11,7),(8),(1,10,5)), ((11,3),(6,9,7),(1,10,8),(12,5),(2,4)), ((10),(1,6,11,5),(9,4,8,7),(12,2,3)), ((12,1,11,7,4),(6,9,10,5),(2,8,3)), ((4),(1,6,7,10),(12,11,8,3),(9,2,5))
21	9897,00	<b>6839,70</b>	<b>276</b>	(((7),(9,1,3,5),(8,12,2,4,6,11,10)), ((7),(9,1,4,5,2),(8,3,12,6,11,10)), ((1,5,10),(12,4),(7,6,11,2),(9,3),(8)), ((0),(4,6,5,12,7,11,8,9,1,2,3,10)), ((0),(2,9),(1,11,6),(8,12,5,4),(7,3,10)), ((9),(2,1,6),(12,7,3,11,8,5,10),(4)), ((7,11),(12,3,5),(8,4,1,2,10),(9,6)), ((8,1,11,5),(9,7,6),(12,4,3,2,10)), ((8,7),(12,9,6,5),(1,2),(4,3,11,10)), ((12,4),(9,11,6,8),(2,1),(7,3,5),(10)), ((1,11),(8,6),(12,7,9,5),(4,3,2,10)), ((4,2,1),(6),(12,7,3,8,11),(5),(9,10)), ((8,7,6,9),(3,5),(1,2),(12,4,11,10)), ((3,1),(7,2,8,9,5),(12,4,6,11,10)), ((11,4),(6,9,5),(3,8,1),(12,7,2,10))

Tabela Z.14. Wyniki obliczeń przeprowadzonych za pomocą algorytmu TS, związane z poszukiwaniem zbioru punktów kompromisowych w przykładzie przedstawionym w p. 6.2.3

Lp.	Ograniczenie $\hat{U}$ na całkowity koszt przedsięwzięcia [tys. PLN]	Wyniki po optymalizacji		
		Całkowity koszt przedsięwzięcia [tys. PLN]	Czas przedsięwzięcia $C_{max}$ [dni robocze]	Permutacja dla najlepszej wartości $C_{max}$
1	2	3	4	5
1	1795,30	1795,30	607	((0),(8,10,6,7,3),(11,1,12,2,9,5,4)), ((12,6,5,10),(11,2,9,3,8),(7,1,4)), (11),(6,5),(12,4,9,3),(2,1,10,8),(7)), (0),(8,7,1,5,10,4,9,2,6,12,3,11)), (1),(10,9,4,5,7),(6,11),0,(8,12,3,2)), (1,12,4),0,(9,8,11,7,6,10,5,2,3),0), (11,6,2,12,4,3,1,9,7,10,8,5),0,0,0), (7),(6,3,10),(11,2,4,1,12,9,8,5)), (2,6,4,11,12,3,9,7,1,10,8,5),0,0,0,0), (1,7,11,5),(6,12,2,4,3,10,9,8),0,0,0,0,0), (2,3,8),(4,10),(11,6,9),(12,7,1,5)), (6,11,1,7,12),(2,9,4,3,10,8,5),0,0,0,0), (2,4,6,11,12,3,7,1,9,10,8,5),0,0,0,0), (11,3,4,7,12,1,2,10,8,6,5,9),0,0,0), (6),(7,10,5),(2,9),(11,3,1,12,4,8)))
2	2200	2191,80	491	((5),(7,9,2,1,4),(10,12,6,3,11,8)), ((10,5,3,6,12),(8,9,11,1),(7,2,4)), (10),(5,6),(3,4,12,9),(7,1,8,2,11),0), (7,11,1,9,5,3,6,4,2,10,8),(12)), (12,11,10),(4,7),(5,1),(9,2,8),(6,3)), (9),(5),(3,10,12),(2,11,7,6,8,1,4)), (7,5,3,12,4,2,11,9,1,8),(6),0,(10)), (10,6,12,2,9,8),(7,3,4),(5,11,1)), (10,7,5,3,12,4,2,11,9,1,8),(6),0,0,0), (2,5,12,7,8),(3,4,6,11,9,10,1),0,0,0,0,0), (10,4,12),(2,11),(7,6),(5,3,9,1,8)), (7,6,10,1,9,8),(5,12,3,4,11,2),0,0,0,0), (10,7,5,3,4,6,11,2,9,1,8),0,(12),0), (4,10,6,7,5,12,3,11,2,9,1),(8),0), (7,10,1),(5,11,3,9),(4),(6,12,2,8)))
3	2600	2574,00	393	((6,7,2,8),(11,1,12),(3,4,5,10,9)), ((11,12,6,10,8,5,1,9),(3),(4,2,7)), (1),(11,6,5),(4,9),(12,2,8,10),(3,7)), (8,9,3,5,7,12,6,2,1),(10,11,4)), (5,12,2),(10,8),(7,9,11,6),(4,1,3),0), (2),(5),(4,8,10,9,3,12),(1,7,6,11)), (11,3,1,7,5,10,8),(12,6),(4,9),(2)), (12,4,6),(2,11,10),(3,1,7,9,5,8)), (12,4,3,2,6,1,11,7,9,5,10,8),0,0,0,0), (1,6,11,2,10),(4,7,12,5,3,9,8),0,0,0,0,0), (11,5),(2,10),(1,6,9),(12,4,3,7,8)), (3,4,1,11,2,7,10,8),(9,12,6,5),0,0,0,0), (12,4,3,2,6,1,5,10,9),(7),(8),(11)), (3,6,1,5,8,9),(11,7,2),(12,4,10)), (4,7,10),(6,11,3,9),(2,5,8),(12,1)))
4	3000	2996,10	354	((7,10,3,1),(11,2,12,9,5),(6,4,8)), ((6,9,10,5),(12,8,4,2,3),(11,1,7)), (9,1),(12,7,5,3),(11,4,8),(10),(6,2)), (4,8,9),(11,7,10,12,6,5,3,2,1)), (2,11,4),(9,3),(10,1),(7),(6,12,8,5)), (5,9),(12,11,2,6,1),(4,8,7),(10,3)), (6,10,5,3,1,7),(11,8),(12,9),(4,2)), (11,12,2,10,5),(9),(6,4,8,3,1,7)), (11,4,9,2,10,5,8,1,7),(3),(6),(12)), (4,12,6,9,2,3,7),(8,10,5,11,1),0,0,0,0,0), (6,2,12),(1,8),(9,10),(11,4,5,3,7)), (12,11,4,6,8,3,7),(9,2,10,5,1),0,0,0,0), (11,4,2,5,3,1),(6,10),(9,12,8),(7)), (6,4,11,9,2,12,5,3),(10,8,7,1),0), (5,3),(4,12,1),(10,9,2,8),(6,11,7)))
5	3400	3387,80	283	((4,10),(5,3,7,9,1),(6,8,12,2,11)), ((6,10,5,8,9),(11,12,3),(4,2,7,1)), (4,11,1),(12,3,7),(5,9),(10,8),(6,2)), (11,12,2,10),(4,9,5,8,7,1,6,3)), (12,9,11,5),(2,7,1),(8),(10,6,3),(4)), (6,3,11,1,7),(9),(4,8),(2,12,10,5)), (6,2,11,7),(12,5,9),(4,3,8),(10,1)), (12,10,5,3,7),(6,9),(4,2,11,8,1)), (4,12,2,5,3,8,1,9),(7),(10,11),(6)), (10,2,6,1,8),(4,12,5,11,3,7,9),0,0,0,0,0), (6,5,7),(2,11,1),(12,8),(4,10,3,9)), (4,12,3,11,1,9),(10,5,2,6,8,7),0,0,0,0), (4,12,10,2,8,9),(11),(3,7),(6,5,1)), (10,5,11,7),(12,4,6,2,1,8),(3,9)), (12,5,3,9),(10,2,8),(11,1),(4,6,7)))

Tabela Z.14. cd.

1	2	3	4	5
6	3800	<b>3781,30</b>	<b>275</b>	((5,1,8,3),(12,6),(2,4,10,7,9,11)), ((5,7,10,6),(12,4,8,9,11,3),(2,1)), ((5,6,10),(2,1),(8,3),(4),(12,7,9,11)), ((12,4,7,2,6,3,1,10,8),(9,5,11)), ((4,2),(11,9,6,3),(5,12,10),(1,8),(7)), (9,7,3,8),(2,4,10,5),(12),(11,1,6)), (5,11,9),(12,2,1,3),(6,8),(4,7,10)), (12,7,11),(4,3),(2,1,5,6,10,8,9)), (2,4,1,5,11,8,9),(7,3),(12,10,6),(0)), (2,1,10,11,6),(12,4,7,5,3,8,9),(0),(0),(0,0)), (5,8),(12,4,7,6),(1,10),(2,11,3,9)), (1,4,5,11,8,6),(12,2,7,10,9),(0),(3),(0)), (12,2,4,11,6,9),(5,8),(1,10),(7,3)), (12,2,1,4,10),(7,5,3),(11,8,6,9)), (4,1,3),(2,5,10,9),(6),(7,12,11,8)))
7	4200	<b>4071,90</b>	<b>265</b>	((8,10,2),(1,12,5,9),(11,4,7,6,3)), (8,3,12,5,9),(11,1,2,10),(4,7,6)), (4,6,10),(11,2,9),(1,3),(5),(8,12,7)), (12,5,11,3,10,7,6),(1,4,9,2,8)), (11,5,4,2,3),(7,6),(12),(8),(1,10,9)), (2),(11,12,6,3),(5,4,7,10),(8,1,9)), (8,2,3,6),(11,12,10),(4,1,7),(5,9)), (4,5,3,12,9),(10),(11,8,1,2,7,6)), (4,1,5,3,12,6),(8,7),(11,2,10),(9)), (4,2,10,1,12),(11,8,5,7,3,6,9),(0),(0),(0,0)), (8,1,9,12),(11,5,3),(4,7),(2,10,6)), (1,4,10,3,6,9),(11,8,5,2,7,12),(0),(0,0)), (4,5,1,9),(11,2,3),(8,10,6),(7,12)), (5,10,3,9),(4,7),(8,11,2,1,6,12)), (10,9),(4,5,6),(8,11,7,3),(2,1,12)))
8	4600	<b>4574,10</b>	<b>253</b>	((6,8,2),(11,7,4,10,5,9,1),(3,12)), (12,6,10,9,5),(7,11,2),(3,1,8,4)), (3,2),(10,9),(6,8,1),(7,11,4),(12,5)), (5),(7,8,3,11,4,6,12,10,1,9,2)), (11,12,9),(7,3),(2,1),(6,10,5),(4,8)), (7,6,10,9),(12),(11,2),(3,4,8,1,5)), (12,11,2,9),(7,10),(6,4,3),(8,5,1)), (12,6,4,2,1),(10,3),(7,11,8,5,9)), (12,11,8,5,1),(4,10),(7,6,2,9),(3)), (11,7,4,12,3),(6,8,10,1,9),(5),(0),(0),(2)), (7,2,9),(11,4,10),(3,1),(12,6,8,5)), (12,7,4,2,10,5,9),(11,3,8,1),(6),(0,0)), (12,7,6,10,5),(11,3,9),(8,2,1),(4)), (7,6,11,10,2,3),(12,5),(4,8,1,9)), (4,5),(10,3),(6,11,2,9),(7,8,12,1)))
9	5000	<b>4985,20</b>	<b>246</b>	((2,3,9),(11,12,8,10),(4,6,1,7,5)), (2,6,10),(4,7,5),(11,8,12,3,1,9)), (2,10),(4,7,6),(11,12,1),(3,9),(8,5)), (11,7),(2,8,4,1,9,10,3,6,5,12)), (2,11,12),(4,8,3,9),(1,7),(5,6),(10)), (2),(8,11,7,1,9),(4,10,3,12,6),(5)), (11,1,6),(2,10,7),(4,3,12,9),(8,5)), (4,10,9),(11,8,1,5,7),(2,3,12,6)), (4,11,5,7),(10,1,9),(3,12,6),(2,8)), (4,3,7,1),(8,11,12,6,5,9),(0),(0),(2,10)), (7),(2,10,6),(11,3),(4,8,12,1,5,9)), (4,11,3,5,12,9),(2,8,7),(10),(1),(6)), (4,8,10,6,9),(11,5),(12,7),(2,3,1)), (2,4,1,6),(8,3,12,5),(11,10,7,9)), (2,10,6),(8,7,9),(4,3,1),(11,12,5)))
10	5400	<b>5298,70</b>	<b>249</b>	((6,1,9,10,4),(2,3,12,7),(11,5,8)), (2,8,3,1),(11,6,4,10),(5,12,9,7)), (6,7,10,1),(2,12),(11,3),(5,4),(8,9)), (5,10),(2,6,12,7,11,4,9,8,3,1)), (2,1),(11,6,4),(5,3),(8,10,12),(9,7)), (2,10,11),(3),(5,7,12),(9,8,6,1,4)), (6,4,10),(2,8,9),(5,3,12),(11,7,1)), (2,8,4,12,9),(5,7,1),(11,6,3,10)), (8,7,9),(2,4),(6,5,3,10,1),(11,12)), (2,6,5,4,1),(8,3,10,9),(0),(11),(12),(7)), (6,4),(11,8,10,1),(7),(2,5,3,12,9)), (11,6,7),(8,5,10,9),(2,3,4,1),(0),(12)), (11,3),(2,5,10,9),(6,4,1),(8,7,12)), (2,11,5,8,12,10),(6,1),(7,4,3,9)), (8,12,1),(2,11,10),(5,4),(6,7,3,9)))
11	5800	<b>5568,50</b>	<b>245</b>	((12,1,10),(2,4,5,3),(11,6,8,9,7)), (2,9,10,3),(11,4,1),(12,6,5,8,7)), (6,10),(12,1),(11,9,3),(4,5),(2,8,7)), (9,12,4,2),(7,11,8,5,10,6,3,1)), (5,6),(3,7),(0),(2,10,1,9),(11,4,8,12)), (4,2,6),(10,8,12,9),(3,1,5),(11,7)), (11,10,9),(6,1,7),(4,8,3),(12,2,5)), (4,10,1,5,7),(11),(12,2,6,8,3,9)), (12,4,6,1,3,9),(10,5),(11,7),(2,8)), (4,6,8,7,9),(11,10,5),(3),(2),(12),(1)), (9),(11,4,10,5),(2,8),(6,1,12,3,7)), (4,7),(12,10,1,9),(2,5),(11,3),(6,8)), (4,10,8,7),(11,2,1,5),(12,9),(6,3)), (12,1,10,7,9),(2,6,5),(11,4,8,3)), (6,12,7),(11,8,3),(4,1,5),(2,10,9)))

Tabela Z.14. cd.

1	2	3	4	5
12	6200	<b>6134,70</b>	<b>245</b>	((12,6,9),(3,10,8),(4,11,2,7,5,1)), ((12,2,6),(4,7,1,8),(3,11,10,9,5)), ((11,6,1),(12,10),(2,7),(4,5),(3,9,8)), ((11,6,4,9,3,5,12,2,8),(10,1,7)), ((11),(7,1,5),(4,3,12,8),(6,2),(10,9)), (2),(11,9,3),(12,6,8,5),(4,10,7,1)), (3,5),(11,1,7),(12,4,10,9),(2,6,8)), ((11,7,8),(12,2,6),(4,3,10,1,5,9)), ((12,10,7,9),(4,5),(2,3,6,8),(11,1)), (4,3,1),(12,10,6),(2,9),(5),(7),(11,8)), ((6,9),(4,10,5),(11,1),(12,2,3,7,8)), (3,11),(6,9,8),(10,7,5),(2),(4,12,1)), (4,10,9),(2,5),(11,1,6,8),(12,3,7)), (4,12,3,1,8),(10,7,5,9),(11,2,6)), ((12,7,8),(11,3,1,9),(2,6),(4,10,5)))
13	6600	<b>6340,10</b>	<b>241</b>	((11,4,5,9),(10,12,2,7),(6,3,1,8)), (10,3,12,9,5),(11,6,2),(4,1,8,7)), (3),(10,2),(6,7,12),(4,5,9),(11,1,8)), (7,10,4,1,9,2),(6,8,3,5,11,12)), (3,8),(6,5,11,2),(10),(4,9,7),(12,1)), ((6,11,8,3,2),(12,9,7,5),(4,10,1,0)), (11,1,12,9),(10,2),(4,3,7),(6,5,8)), (6,1,5,8),(11,3,2),(4,10,12,7,9)), ((1,2,8),(10,5,9),(4,6,12,7),(11,3)), (4,7),(10,12,2),(0,(1,3),(11,8),(6,5,9))), (1,5,8),(4,6,12,7),(11,3),(10,2,9)), (11,4,9),(1,10,3,2,8),(5),(6),(12,7)), (4,3,8),(12,5),(11,10,7),(6,1,2,9)), (6,11,2),(1,7,5,8),(4,10,12,3,9)), (1,12,3,9),(11,2),(6,10,5),(4,7,8)))
14	7000	<b>6480,90</b>	<b>241</b>	((3,6,10),(2,4,1),(11,5,9,7,8,12)), (11,5,10,6),(3,1,7,12,8),(2,4,9)), (11,1,12),(2,10,8),(3,6,7),(4,9),(5)), (2,4,1,9,6,8,7,10),(3,11,5,12)), (2,9,10,6),(11,1,4),(7,3),(5,8,12,0)), (9,10,1),(5,2,4,6),(11,7,8,3),(12)), (11,12,10),(2,4,1,6),(5,9),(3,7,8)), (11,3,1,7,8),(5,10,9),(2,4,12,6)), (4,3,10),(11,5,8),(2,1,6,9),(12,7)), (7,1),(4,10,8),(6),(5),(12),(11,2,3,9)), (9),(11,3,12,7,8),(1,6),(2,4,5,10)), (5),(11,2,1,7),(3,12,9),(4,6,8),(10)), (3,6),(5),(11,4,12,7,8),(2,1,10,9)), (2,4,7,10,3,9),(12,1,8),(11,5,6)), (4,1,8),(2,5,3),(12,10,9),(11,7,6)))
15	7400	<b>6424,70</b>	<b>250</b>	((12,9,1,5),(10,11,2,7,4,8),(3,6)), (3,9,7,4),(10,6,8,11),(12,1,5,2)), (10,2),(9,11,8,1),(12,6,7,4),(3),(5)), (6,9,2,8,4,3,5),(11,7,10,12,1)), (0),(3,6,1),(10),(7,12,11,8),(9,5,4,2)), (7,5,6),(3,2,1,11,8),(10,9),(4,12)), (6,5),(12,10,2,4,8),(3,7),(9,11,1)), (10,9,2,11,8,1),(3,4),(12,6,7,5)), (12,3,6,5,1),(9,4,8),(2),(10,7,11)), (9,7,4),(12,6,5),(10,11),(3,1),(8),(2)), (9,7,8),(4,1),(3,2),(12,10,6,11,5)), (12,11),(7,5),(9,4,2,1),(3,10,6),(8)), (10,6,7,5),(9,11),(12,2,8),(3,4,1)), (12,7,2,4,1),(10,6,9,5,8),(3,11)), (3,4),(12,6,11,8),(10,2,1),(7,9,5)))
16	7800	<b>7445,20</b>	<b>248</b>	((4,10,11,8,6,1),(12,7,9,5),(3,2)), (4,8,6,1),(12,10,5,9),(3,7,2,11)), (4,6,2),(7,11,9),(10,1),(3,5),(12,8)), (3,6,1,10,8,4,7,9),(12,11,2,5)), (7,9,1),(12),(4,8),(10,6,11),(3,2,5)), (6),(12,7,4,3,1,10,11),(8,9),(2,5)), (12,1,11),(4,10,6),(3,8,2),(7,5,9)), (4,7,2,6),(12,8,11),(3,10,1,5,9)), (12,7,2),(10,5,11),(3,1,6,9),(4,8)), (6),(7,2),(10),(1),(4,5,9),(12,3,8,11)), (4,10,2,9),(3,1,5,6),(0),(12,7,8,11)), (7),(10,12,1,9),(3,2),(5,6),(4,8,11)), (7,1,12,11),(3,5,9),(4,2),(10,8,6)), (7,4,10,1,5,9),(3,12,8,2),(11,6)), (3,1,2),(4,8,11),(7,10,6),(12,5,9)))
17	8200	<b>7872,80</b>	<b>247</b>	((8,10,11,4,5,3),(9,2),(12,6,7,1)), (6,1,11),(12,4,7),(8,9,10,3,5,2)), (12,10,3),(4,2),(6,1),(9,7),(8,11,5)), (8,6,12,9,7,4,2,3),(10,5,11,1)), (1,11,2),(4,7),(8,5,3),(10),(9,12,6)), (8,9,1,6),(3,10),(11,7),(12,4,5,2)), (12,11,2),(6,5),(8,10,3),(9,4,1,7)), (6,10,11,3),(9,2),(12,8,4,1,5,7)), (6,7),(4,11,5),(9,1,2),(12,8,10,3)), (11),(3),(4),(12,10,1),(9,8,5,7),(6,2)), (9,4,3),(12,6,11,2),(8,1,7),(10,5)), (12,9,10,7),(8),(3),(6,4,2,1),(11,5)), (4,1,7),(12,9,10,3),(6,11,2),(8,5)), (9,8,6,4,7),(10,2,5),(12,11,1,3)), (10,5),(6,1,7),(9,12,4,3),(8,11,2)))



Tabela Z.14. cd.

1	2	3	4	5
18	8600	<b>6651,70</b>	<b>240</b>	((6,1,9),(12,10,7,3),(11,4,5,8,2), ((11,10,3,1,9),(12,6,7,8),(4,5,2)), (5,1),(4),(6,7,2,8),(12,3,9),(11,10)), (12,7,4,10,1,8),(5,6,11,3,9,2), (12,7,9),(2,3),(8,1),(6),(11,10,4,5)), (3),(7,4,11,12,6),(5,1),(2,8,10,9)), (11,5,8),(12,4,10,2),(7,1),(6,3,9)), (11,6,3,1,9),(8),(12,4,7,10,5,2), (4,1,8),(5,9),(7,10,3,6),(12,11,2)), (4,3,1),(11,12),(5,6),(8),(10,2),(7,9)), (12,5,9),(11,10,2,1),(6),(4,7,3,8)), (3),(12,8),(7,5),(4,6),(11,10,1,2,9)), (4,3,9),(11,12,5,1),(7,2,8),(10,6)), (5,8),(7,11,10,6,2,1),(4,12,3,9)), (7,10,2),(11,6,1),(4,5,9),(12,3,8)))
19	9000	<b>6952,00</b>	<b>245</b>	((9,7,3,10,8),(2,5,1),(11,4,6,12)), (9,2,6,10,8),(11,7,3),(4,5,1,12)), (6,10),(4,1),(11,7),(9,5),(2,3,8,12)), (7,6,10,2,9,3,1,12,8),(4,5,11)), (9,12,1),(7,8),(6,11),(5,4,10,3),(2)), (5,11,9),(7),(6,4,1,3),(2,10,8,12)), (11,7,8),(4,2,1,10),(6,3),(9,5,12)), (4,7,1,3),(2,8),(11,9,6,5,10,12)), (9,3),(4,2,5,12),(6,8),(11,7,1,10)), (4,5),(7),(2,1),(11,10,3),(8),(9,6,12)), (11,9,7,12),(2,1,3),0,(4,6,5,10,8)), (11,4,2,1,12),(9,7,5),(6,10),(8),(3)), (4,2,10),(5),(11,9,1,12,8),(6,7,3)), (1,9,8),(4,11,2,5,3),(6,7,10,12)), (7,10),(4,6,1,8),(2,9,3),(11,5,12)))
20	9400	<b>6882,60</b>	<b>237</b>	((7,10,6,1,12),(11,8,9),(2,4,3,5)), (2,4,9,6,7),(11,12,5),(8,10,3,1)), (10,6),(11,4),(2,5),(12,7,1),(8,3,9)), (2,9,1,3,6,12),(11,4,8,7,5,10)), (10),(8,12,9),0,(4,2,6),(11,3,7,5,1)), (0),(2,5),(4,7,12),(11,10,8,9,3,6,1)), (11,3,9),(2,4,12,7),(10,5,1),(8,6)), (8,4,5,9),(11,12,6),(2,10,3,7,1)), (2,10,3,1),(8,5),(11,4,12,6,9),(7)), (7),(4,3,6,9),(10,5),(1),(8,2,12),(11)), (4,6,1),(11,8,5),(3),(2,10,12,7,9)), (2,11,12),(7,9),(4,8),(10,3,6,1),(5)), (12,9),(8,5),(11,4,7,6),(2,10,3,1)), (2,4,3,9),(11,7,5,1),(8,10,12,6)), (8,4,3,9),(7,5),(10,12,1),(11,2,6)))
21	9897,00	<b>7324,40</b>	<b>242</b>	((10,5),(8,7,2,1,3),(12,6,4,9,11)), (7,4,9,8),(12,2,11),(10,6,5,3,1)), (10,3),(2,1),(7,4),(12,5,11,9),(6,8)), (2,12,5,3,9),(7,6,10,11,4,1,8)), (11,3),(10,8,1),(5,4,9),(2,6,12),(7)), (11,4,6),(7,5,3),(2,10,9),(8,12,1)), (10,3,1),(6,9),(12,2,5,8),(7,4,11)), (7,10,4,5,1),(2,11,8),(12,6,3,9)), (7,2,1,9),(10,5,11),(4,3),(12,6,8)), (2,12,11),(3),(6,8,9),(10,5,1),(4),(7)), (7,4,9),(6,5),(2,1),(12,10,3,11,8)), (3,9),(8),(2,5),(7,10,6,1),(12,4,11)), (4,11),(12,6,5,9),(7,2,8),(10,3,1)), (2,1,8),(12,7,4,5),(6,10,3,11,9)), (10,11),(2,3,8),(7,6,12,5,9),(4,1))

Tabela Z.15. Wyniki obliczeń przeprowadzonych za pomocą algorytmu SA, związane z poszukiwaniem zbioru punktów kompromisowych w przykładzie przedstawionym w p. 6.2.3

Lp.	Ograniczenie $\hat{U}$ na całkowity koszt przedsięwzięcia [tys. PLN]	Wyniki po optymalizacji		
		Całkowity koszt przedsięwzięcia [tys. PLN]	Czas przedsięwzięcia $C_{max}$ [dni robocze]	Permutacja dla najlepszej wartości $C_{max}$
1	2	3	4	5
1	1795,30	<b>1795,30</b>	<b>572</b>	((12,9,5,3,4,11),(7,8,2,10),(1,6), ((12,10,5,6),(9,11,8,2,3),(7,4,1)), (11),(5,6),(12,9,3,4),(2,10,1,8),(7)), ((11,1,12),(5,2,9,7,8,4,10,6,3)), (8,1),(9),(5,7,11,3),0,(12,2,10,6,4)), (5,2,12,8,4,10,1),(3),(11,9),(7,6)), (12,7,2,3,4,5,1,8,6,11,10,9),0,0,0), (7,5),(2,3),(4,12,8,1,6,11,10,9)), (7,3,2,4,12,5,8,6,11,1,10,9),0,0,0), (8,3,5,1,11,9),(2,6,4,12,7,10),0,0,0,0), (3,2),(8),(7,12,4,11),(5,6,1,10,9)), (4,5,12,3,11,7),(2,8,1,6,10,9),0,0,0), (7,3,2,5,12,4,8,6,11,10,1,9),0,0,0), (5,2,4,3,12,7,11,6,8,10,1,9),0,0,0), (2,5,12,7,11,1),(8),(4,3,6),(10,9))
2	2200	<b>2192,40</b>	<b>504</b>	((7,8,9,1,12,5),(10,4,11,2,3),(6), (6,5,10),(3,2,9,8),(12,1,11,7,4)), (4),(10,5),(6,12),(2,1,8),(11,9,3,7)), (11,8,2,9,5),(4,7,12,1,3,10,6)), (7,12,3,6),0,(10,9),0,(8,2,11,5,4,1)), (12,2,3,6,9),(7,1),(8,10,11,4),(5)), (6,11,2,3,10,5,8,9,7),(12,4),0,(1)), (6,12),(11,2,9),(4,1,3,10,5,8,7)), (12,6,2,4,11,1,10,3,5,8,9),(7),0,0), (10,5,6,12,3),(2,1,4,11,8,7,9),0,0,0,0), (0,(4,10,1,8,3),(12,2,5,9),(6,11,7)), (12,4,6,11,2,3,1,10,8),(5,9,7),0,0,0), (12,6,4,2,10,11,1,5,3,9),(8),0,(7)), (4,1,2,10,8,5,3),(12,11,9),(6,7)), (11,8,3),(5,6),(4,10,7),(2,1,12,9))
3	2600	<b>2585,60</b>	<b>451</b>	((8),(4,7,5,10,9,11,6,1,12),(3,2)), (8,9,6,1,10,12),(3,2,5,11),(4,7)), (2),(8,9,11,1,6,5),(3),(10,12),(4,7)), (11,3,8),(7,1,6,2,4,9,10,5,12)), (4,11,3,10),(2),(9,6,7,1),(8,5,12),0), (11),(1,6),(2),(3,9,12,10,4,5,7,8)), (8,4,2,7,6,5,10,9),0,0,(3,11,12,1)), (12,1,6),(3,8),(4,2,11,7,5,10,9)), (3,8,4,2,12,1,6,7,11,5),(10),(9),0), (2,4,11,7,1,6,12,10),(3,8,5,9),0,0,0,0), (3),(2,6,11,7,5,10,9),(8,1,12),(4)), (3,4,11,7,12,9),(2,8,6,1,10,5),0,0,0), (3,8,4,1,2,11,6,5,9),(7),(12,10),0), (1,4,8,2,3,7,5),(11,6,12,10),(9)), (12),(4,7,5),(3,2,11),(8,6,10,9))
4	3000	<b>2983,39</b>	<b>385</b>	((6,7,1),(4,10,11,5,9),(12,8,3,2)), (11,10),(6,4,2,9,8,5),(12,3,1,7)), (6,3,7,2),(10,9,5),(12),(1,8,11),(4)), (12,1,11,4,9,10,8,3),(6,5,7,2)), (12,8,3),(9,11,10,5,2),0,(4,7,1),(6)), (4),(3,11,12,8,9,5),(7,10),(1,6,2)), (6,10,7,5,8,9),(12),(4,11),(3,1,2)), (12,4,1,7),0,(6,3,10,2,11,5,8,9)), (6,12,3,10,4,1,2,7,11,5),0,(9),(8)), (4,6,10,7,5,12,8,9),(3,11,2,1),0,0,0,0), (4,2,8),(3,1,5,7),(12,6,10),(11,9)), (10,1,11,7),(4,12,6,3,2,5,8,9),0,0,0,0), (12,3,1,2,4,5,8),(10,9),(6,11,7),0), (12,6,3,10,11,1,2),(4,7),(5,8,9)), (3,6,4,12,1),(10,2,5),(11,7,8,9),0)
5	3400	<b>3394,30</b>	<b>335</b>	((8,9,7,3,11,10),(2,5,12,1),(4,6)), (8,6,3,11,12,5,10,1,9),(4),(2,7)), (7),(6,3),(4,12,10,9),(2,11,5,1),(8)), (11,10,8,9),(2,4,7,12,3,5,1,6)), (0,(12),(8,7,11,1),(6,2,9,5),(3,4,10)), (4,11),(2,12,8,5,1),(3,6,10,9,7),0), (8,7,10,11,9),(4,12,5),(3),(6,2,1)), (8,12,7,5,10,6),0,(4,3,2,1,11,9)), (8,12,7,2,10,6,1,11,9),(4),(5),(3)), (12,8,2,7,6,3,11,9),(4,10,5,1),0,0,0,0), (8,10),(3,5,11,1,9),(4,7),(12,2,6)), (3,7,2,1,10,5,11),(4,12,6,8),0,(9),0), (4,3,5,6,7,10),(12,8,2,9),(11,1),0), (3,12,6,5,11),(4,2,8,7,10),(1,9)), (8,3,1,9),(2,7,10),(12,6,11),(4,5))

Tabela Z.15. cd.

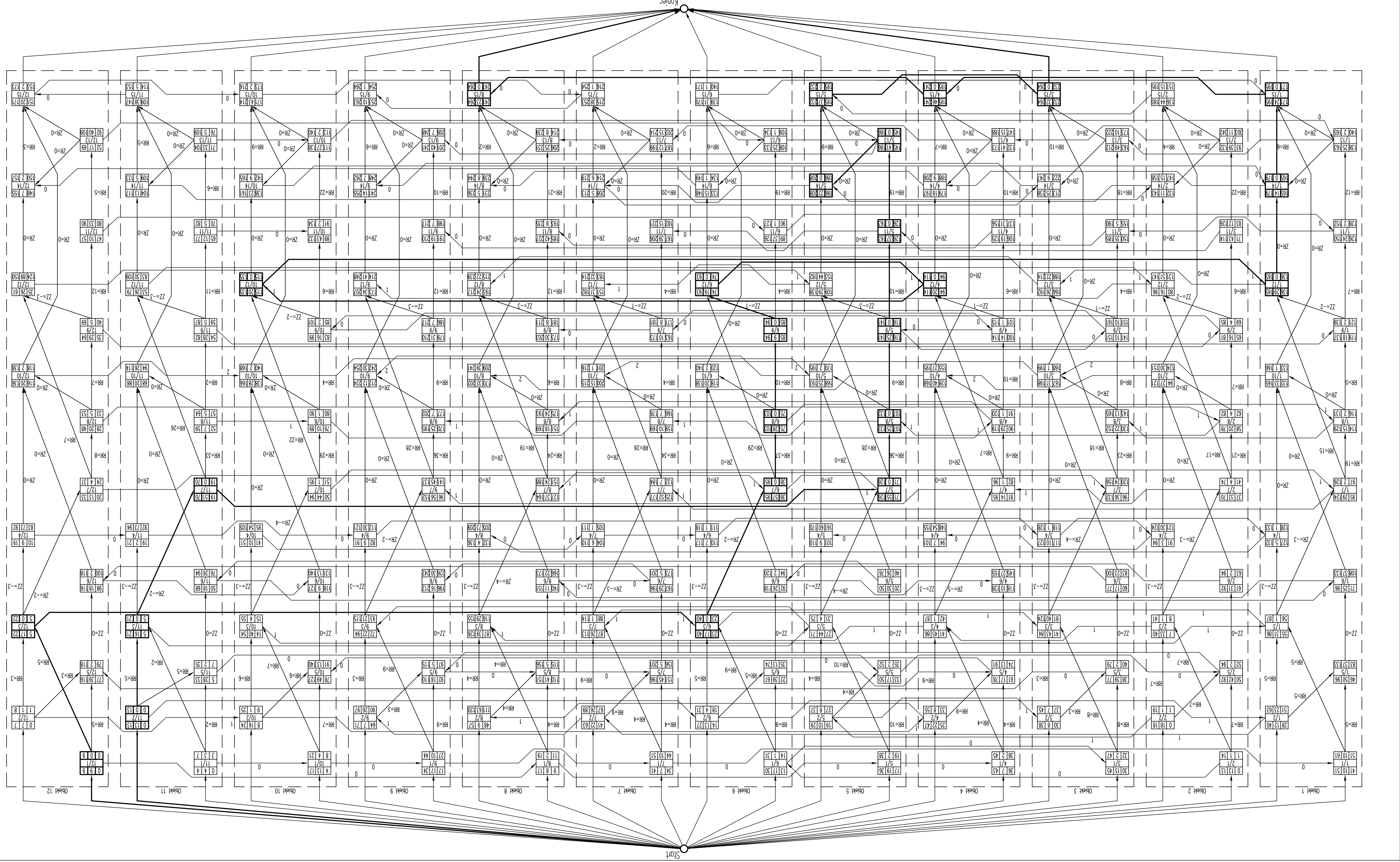
1	2	3	4	5
6	3800	3755,20	318	((6,8,3,1),(9),(4,12,5,11,7,2,10)), (9,6,3),(12,8,4,11,5),(10,1,2,7)), (0),(2,11,5),(12,4),(9,8,10),(6,3,1,7)), ((4,6,3,12,7,10,11,2),(5,9,1,8)), ((11,1,9),(4,10,8,5),(3,6,2),(12,7),(0)), (12,3,10),(4,7,11,8,5),(1),(6,9,2)), (9,3,10,5),(12,4,2,8),(6,11),(1,7)), (6,11,10,8,7),(1),(12,9,4,3,2,5)), (12,9,1,2,8,7,5),(4,3,10),(6),(11)), ((6,3,9,11,2,7,8),(12,4,1,10,5),(0,0,0,0)), (0),(9,12,10,1,5,7),(4,6,2,8),(3,11)), (3,12,11),(9,4,6,1,8,7,5),(0,0),(10,2)), (9,12,4,2),(6,11,10,5),(1,8,7),(3)), (9,3,11,10,5,8),(6,12,2,1,7),(4)), (9,11,2,8),(3,5),(4,10,1),(6,12,7))
7	4200	4178,20	298	((1,11,3,10,7,5),(8,2,6),(4,9,12)), (1,12,2,10,6),(4),(8,9,11,3,5,7)), (3),(4,10,6),(1,12,9),(8),(11,2,5,7)), (9,5,7,10,4),(1,11,2,8,12,6,3)), (3),(7,10,6),(9,5,11,2),(1),(4,8,12)), (9,3,12,11,7,4,5),(6),(1,8,2),(10)), (1,2,10,7),(4,12,3,6),(11),(8,9,5)), (8,2,9,6),(4,3,10,5,7),(1,11,12)), (11,8),(1,4,3,9,6,7),(12),(2,10,5)), ((1,3,10,4,6),(8,12,11,2,7,9,5),(0,0,0,0)), (1,8,9,3),(4,11,2,10,7),(0),(12,6,5)), (11,2,1,4,10,9),(8,6,12,7,5),(3),(0,0)), (11,2,6,3),(4,1,10,7),(12,8),(9,5)), (11,2,10,6,9,3),(12,4,7,5),(8,1)), (1,11,4),(2,5),(12,7,3),(8,10,9,6))
8	4600	4542,60	290	((7,8,3,5,10),(4,9,12,6),(1,11,2)), (7,3,5,8),(12,1,4,9,6,10),(11,2)), (7,9,5),(2),(1,8),(11,4,12,6),(3,10)), (1,3,7,4,11),(9,10,12,8,2,6,5)), (0),(2,4,8,10,5),(3,12,7),(1),(11,9,6)), (3,2,5),(9,1,12,4,11,10),(8,7,6,0)), (1,4,12,10,6),(7,8,5),(2,3),(11,9)), (12,9),(4,8,10,5,6),(1,2,3,11,7)), (1,2,11),(4,12,7,9,5),(10,8),(3,6)), (3,4,8,6,9),(1,11,12,10,7,5),(0,0,0,2)), (1,10),(2),(4,12,7,5),(3,11,9,8,6)), (1,3,11,8,9),(4,12,2,7,10),(6),(5,0)), (2,11,10,5),(3,7,9),(0),(1,4,12,8,6)), (3,9,6),(12,7,1,10,8,5),(2,4,11)), (12,7,8),(2,3,5),(11,1,6),(4,10,9))
9	5000	4962,30	283	((2,9,10),(6,1,3),(11,4,8,12,7,5)), (4,1,8,3),(11,2),(5,6,12,9,10,7)), (4,7),(2),(6,10),(12,9,5,3),(11,1,8)), (6,7,8,3,9,11),(4,10,12,5,1,2)), (11,2,5,1,8),(9,7,3),(0),(4,6),(10,12)), (12,1),(2,4,8),(6,9),(10,11,5,7,3)), (12,2,1,5),(11,8,3),(6,10,7),(4,9)), (11,2,1,8,7,3),(12,6,10,9,5),(4)), (11,12,8,3),(4,2,6),(1,9,7),(10,5)), (4,2,10,8,5),(6,12,11,1),(9,7),(3),(0,0)), (7),(2,8,3),(11,6,9),(4,12,1,10,5)), (11,6,4,9),(2,10,12,7,5),(1),(3),(8)), (4,6,12,5),(1,7),(11,10,9),(2,8,3)), (1,12,2,11,5),(10),(4,8,7,6,9,3)), (2,12,5),(4,6,3),(11,8,1,9),(7,10))
10	5400	5396,40	275	((10,12,8),(3,1,4,5),(11,7,6,9,2)), (11),(10,12,9,7,5,4,2),(3,8,1,6)), (3),(12,7,5),(10,9,2),(11,1),(6,4,8)), (1,7,5,10,3,12,9,6,8),(2,4,11)), (6,5,4),(1,7,2),(10),(11,9),(3,12,8)), (1,8,12),(9,6,11,3,10,2),(7),(4,5)), (11,10,5),(12,1,6),(3,9,8),(7,4,2)), (12,6,5,8),(11,1,10,4,2),(7,3,9)), (12,1,10,6),(9,5),(11,3,2,8),(7,4)), (1,7,12,6,2,8),(11,10,4),(5),(0),(9),(3)), (1,4,8),(10,5),(7,2),(11,12,3,9,6)), (3,6,12,2),(7,10),(8),(1,5,9),(11,4)), (7,3),(12,1,9,5,8),(11,10,6,2),(4)), (7,10,5),(11,12,3),(1,6,4,2,9,8)), (7,11,5),(1,6,9,8),(10,4),(12,3,2))
11	5800	5705,10	282	((6,9,4),(11,12,2,5),(3,7,10,8,1)), (11),(12,5,10,9,2,8,1),(3,6,7,4)), (10,8,1),(11,7),(3,6,2),(12,5),(9,4)), (11,12,3,2,9,7,6,8,1,10),(5,4)), (6,11,2),(5,7),(9,3,1),(12,10,8),(4)), (9,7,11,8,1,5),(6),(12,4,3,2),(10)), (11,5,8,9),(12,7,4,2),(3,6,1),(10)), (6,8,9),(11,12,7,10,5),(3,4,2,1)), (11,3,4,8),(12,10,5,2),(6,1,9),(7)), (3,4,5,1),(7,11,12),(8,2,9),(0),(6),(10)), (3,7,8),(11,10,5,1,9),(12),(4,6,2)), (7,4),(6,10,2),(11,3,12,5),(8),(1,9)), (12,7,4,1,9),(3,2),(10,6,8),(11,5)), (11,3,6,12),(10,4,5,9),(7,8,2,1)), (11,12,5,1),(6,9),(3,8,4),(7,10,2))

Tabela Z.15. cd.

1	2	3	4	5
12	6200	<b>6150,40</b>	<b>268</b>	((7,1,6,5,8),(11,9,4,2),(12,3,10), (2,6,3,8),(11,12,5,10),(9,7,4,1), ((11,2,5,8),(12,4,1),(9,3),(6),(7,10)), (12,7,4,11,6,10),(5,9,1,2,3,8)), (9,11,6,5,7),(1),(12,3),(2,10),(8,4), (6,2,4),(11,7),(3,5,9,8),(12,10,1), (11,2,3,8),(12,6,5),(7,4,1),(9,10)), ((11,7,6,1,3,8),(2,10,5),(12,9,4), (11,12),(7,6,4,1),(9,2,10,3,5),(8)), ((1,2),(11,7,9,5),(10,8),(12,3),(6),(4)), (6,1),(9,12,4,2,5),(7),(11,10,3,8)), (9,4,10),(7,1,2,3,8),(12),(11,6,5,0), (12,7,4),(10,3),(11,6,2,8),(9,1,5), (12,11,7,3),(6,9,4,1,2,5,8),(10), (12,10,3),(6,9,1,8),(7,2),(11,4,5)))
13	6600	<b>6563,90</b>	<b>276</b>	((10,4,9,1),(2,6,11,3,5),(7,8,12), (2,12,3,9,10,11,5,1),(7,6),(8,4), (6,3),(7,2),(4),(8,9,5,1),(12,10,11)), (9,2,7,5,6,8,3,1),(12,4,10,11)), (6,5,1),(4,9,10,11),(2,7),(8),(3,12)), (8),(2,9,12,5),(10,4,7,3,1),(6,11)), (7,3,11),(6,10,1),(4,2),(12,8,9,5), (12,8,2),(4,9),(7,6,3,10,11,5,1)), (7,8),(4,2),(9,5,1),(12,6,3,10,11)), (8,7,9,5,1),(10),(11),(3),(4,2),(12,6), (6,9,11),(7,2,10,5,1),(12,4),(8,3)), (3,11),(9),(12,10,5,1),(4,7,2),(8,6)), (12,8,3,11,1),(4,9,5),(2,10),(7,6), (8,4),(3,9,5),(6,2,12,7,10,11,1), (8,12,10,1),(4,3,5),(7,9),(6,2,11)))
14	7000	<b>6884,10</b>	<b>276</b>	((12,1,11,9,10,6,3),(4,2),(5,7,8), (12,2,7),(5,4),(1,8,11,10,3,6,9), (12,7,6,9),(2),(1,11,8),(5,10,3),(4), (10,3,8),(5,1,12,7,11,2,4,6,9)), (0),(1,4,9,7),(11,2,5),(10,8),(12,3,6)), (5,4,6,3),(11,10),(7,1,8,9),(2,12)), (2,11,10),(4,6),(1,5,8,3),(12,7,9), (4,5,8,10,3),(12,1,2,11,6),(7,9), (6,8),(4,7,5,9,3),(2,11,10),(12,1)), (2),(11,6,1),(5,12,3),(0),(4,8),(7,10,9), (4,12,9,8),(1,5,11),(7,10),(2,6,3)), (1,3,10),(7,4,11),(5,6,8),(0),(12,2,9), (5,9,8),(7,1,10),(4,1),(12,2,6,3)), (4,12,7,10,8),(5,9,3),(2,1,11,6), (5,12,10),(2,7,3),(4,6,9,1),(11,8)))
15	7400	<b>6528,60</b>	<b>280</b>	((9,3,12),(7,1,8,4,5,10),(6,11,2), (6,1,2,5,3,10),(7,8,4),(12,9,11)), (11,2,10),(9,12,8),(4,3),(7),(1,6,5), (1),(6,11,8,2,5,7,4,3,9,12,10)), (4,5),(1,3,6),(11,8),(2,10),(7,9,12)), (1,4,11,9),(5,12,7,10,8,2),(0),(3,6)), (1,6),(7,12,2,3),(11,5),(9,4,10,8)), (9,6,5,8),(7,11,12,2,10,3),(1,4), (11,2,10,3,8),(6),(7,9,12),(1,4,5)), (7,10,5),(6,2,8),(9,12,3),(1),(11,4),(0), (2,3),(12,6,10),(11,9,5,8),(7,1,4), (0),(9,11),(2,6,3,5,8),(10),(7,12,1,4), (2,3),(1,9),(11,12,7,6,5,8),(4,10)), (9,10,5,8),(6,2,3),(11,1,7,12,4), (12,6,2),(1,4,8),(7,5),(11,9,10,3)))
16	7800	<b>7573,70</b>	<b>268</b>	((6,1,10,3),(7,11,12,4,5,2),(8,9), (9,6,7,3),(5,11,10,2),(8,12,1,4), (9,10,2),(12,7,3),(6,1),(8,4),(11,5)), (9,8,11,7,5,3,1),(6,2,12,4,10)), (11,5,3,2),(0),(8,4,9),(6,1,10),(12,7)), (0),(5,1,12,4,10,9),(6),(11,8,2,3,7)), (6,4,3,7),(12,1,11),(8,10),(9,5,2)), (12,1,4,11,3,2,7),(6),(8,9,10,5)), (2),(6,4),(12,8,10,11,7),(1,9,5,3)), (8),(5),(1,12,2,7),(4),(6,10),(11,3,9)), (1),(12,6,4,10,2),(5,7),(8,9,11,3)), (8),(0),(9,5,3),(4,6,10,2,7),(12,1,11)), (4),(6,12,11,3),(8,10,2,7),(1,9,5)), (6,8,4,9,3),(1,10,5),(11,12,2,7)), (6,4,5),(1,12,9,2),(11,3),(8,10,7)))
17	8200	<b>6790,30</b>	<b>275</b>	((11,8,1,2,5,4),(7,3,9),(12,10,6), (11,1,3,5,6),(8,2,9,4),(7,12,10)), (3,5),(2),(11,10,6),(7,1,4),(8,12,9)), (11,12,4,2,5,8,3),(10,9,7,6,1)), (0),(8,2,4,9),(12,11),(7,6,5,10),(3,1)), (11,6,4,5),(12,8,3,10,9),(7,2),(1)), (11,3,1,6),(12,2,4,10),(7),(8,9,5)), (11,3,2,10,5),(12,9,1,6),(8,7,4)), (0),(7,3,4,2),(11,12,9,10,5),(8,1,6)), (4),(3,1),(0),(11,12,7,10,6),(0),(8,2,9,5)), (11,5),(8,12,2,1),(4,7,6),(3,9,10)), (2,5),(9,8,10,1),(7,11),(6),(12,4,3)), (12,9,1),(3,7,5),(8,4),(11,2,10,6)), (11,12,4,7,1,5),(3),(8,2,9,10,6)), (3,1,6),(12,8,10,9),(11,4,5),(7,2)))

Tabela Z.15. cd.

1	2	3	4	5
18	8600	7696,10	266	((12,11,12,4,7,9),(8,10,5,3),(6), ((6,11,7,12),(8,2,4),(3,1,10,5,9)), ((6,7,10),(1,2),(8,4),(12,3,9),(11,5)), ((4,5,11,7,6,3,8,1,9,10,12),(2)), (4),(11,3,10),(1,9,2),(12,7),(6,5,8)), (5,11,2),(7,10,12,3,9),(1,6,4),(8)), (1,2,3),(8,4,10),(11,5),(6,12,7,9)), (8,12,7,5,9),(6,4,3),(11,1,2,10)), (0,(11,2,7,3,10),(8,12),(6,4,1,5,9)), (0,(4),(6,8),(10,5,12),(7),(11,2,1,3,9)), (11,4,1,5),(7,8),(6,3,9),(2,12,10)), (2),(1,10),(11,5),(12),(6,4,7,3,8,9)), (6,4,12,10,9),(11,1,3),(7,8),(2,5)), (7,6,3),(11,1,8,12,5,9),(2,4,10)), (4,3),(11,6,8,12,9),(7,1,5),(2,10)))
19	9000	6686,50	273	((8,12,4,2),(9,10,1),(3,5,11,6,7)), ((8,12,4,6,10),(9,12,5,11,7),(3)), ((5,6,10),(8,7,1,1),(3),(12,4),(9,2)), ((8,10,7,1,11),(4,3,2,5,12,9,6)), ((11,7,9),(12,10),(3,1,5,6),(4,8),(2)), (1,10),(2,7,9,11),(12),(4,3,8,5,6)), (12,3,6),(8,2,4,1),(5,10),(9,7,11)), (3,2,7),(9,4,1,1),(12,8,5,6,10)), (3,5,6,10,1),(8),(12,4,7,11),(9,2)), (12,7,5),(10,11),(3,4),(9),(8,6),(2)), (3,5),(9,4,7),(10),(12,8,2,6,11,1)), (10,4),(3,2,5,9),(8,12,7,11),(6,1)), (3,5),(2,6,11),(12,8,4,10,1),(7,9)), (8,3,4,10),(12,2,7,6,11),(5,9)), (7,5,1),(4,6),(8,2,9,1),(12,3,10)), ((8,12,3,5),(11,6,1),(10,4,7,9,2)), ((11,3,5,4),(10,6),(8,9,12,7,1,2)), ((8,2),(12,4),(10),(6,7,1,5),(11,9,3)), ((7,11,6,1,8),(9,3,2,4,5,12,10)), ((8,1,4),(3,9,2,7),(5),(10,6),(12,11)), ((11,9,4),(6,12,10),(8,3),(5,1,2,7)), ((10,1,3,6),(8,4),(11,7,5),(12,9,2)), ((12,4,7,1,5,6),(11,9,2,3),(8,10)), (10,3,6),(12,4),(8,7,2,1,5),(11,9)), (12,8),(9,11,10,3,6),(7,1),(4),(2,5)), (9,7,5),(4,1),(12,8),(11,10,2,3,6)), ((11,9,7,2,1),(8,3,6),(4),(10),(12,5)), (10,9,1),(12,5),(4,8,2),(11,7,3,6)), (4,7,8,2,1,9,5),(10,11),(12,3,6)), ((4,2,5),(9,10),(11,12,8,3),(7,1,6))
20	9400	5929,30	286	((8,12,11,1,5,2),(7,4,6),(10,3,9)), ((8,7,9,1,11,5),(6,12,2),(3,10,4)), ((8),(12,4,7),(6,2,1),(10,11),(3,5,9)), ((10,11,7,9,2,4),(12,5,6,11,3,8)), ((5,2,11,4,3),(1,9,8),(7,6),(10,12)), ((2,10,7),(4,2,1,9),(6,5),(8,3,11)), ((10,3,7,11),(12,4),(8,6,2,1,5,9)), ((10,12,11),(8),(6,3,1),(4,2,7,5,9)), (4),(9),(6,7),(10,8),(12,2,5,1),(1,3)), (4,6,7),(10),(3,1,9),(8,2,12,11,5)), ((8,4,12,11),(2,3,6,9),(10,7,5),(1)), ((10,7,11),(2,1,9),(8,4,6,5),(12,3)), ((10,12,2,7,1,11),(4,6,3,5,9),(8)), ((2,5),(10,4,8,9),(12,7,11),(6,3,1))
21	9897,00	7319,30	275	((8,12,11,1,5,2),(7,4,6),(10,3,9)), ((8,7,9,1,11,5),(6,12,2),(3,10,4)), ((8),(12,4,7),(6,2,1),(10,11),(3,5,9)), ((10,11,7,9,2,4),(12,5,6,11,3,8)), ((5,2,11,4,3),(1,9,8),(7,6),(10,12)), ((2,10,7),(4,2,1,9),(6,5),(8,3,11)), ((10,3,7,11),(12,4),(8,6,2,1,5,9)), ((10,12,11),(8),(6,3,1),(4,2,7,5,9)), (4),(9),(6,7),(10,8),(12,2,5,1),(1,3)), (4,6,7),(10),(3,1,9),(8,2,12,11,5)), ((8,4,12,11),(2,3,6,9),(10,7,5),(1)), ((10,7,11),(2,1,9),(8,4,6,5),(12,3)), ((10,12,2,7,1,11),(4,6,3,5,9),(8)), ((2,5),(10,4,8,9),(12,7,11),(6,3,1))



Rys. Z.4. Przykład z p. 6.2.2 – sieć

zależności dla permutacji

- $\pi = (((11,10,9,7,1),(12,8,5,4),(2,6,3)),$
- $(12,10,4,8),(11,6,1,7,9),(2,5,3)),$
- $(12,6,5),(2,4,8),(10,1,7),(11,5,9)),$
- $(12,11,9,2,4,5),(10,7,6,3,1,8)),$
- $(6,4,10),(3,12),(1),(11,5,2,9,8,7)),$
- $(3,12,10),(2,6,4),(5,11,1,8,7,9)),$
- $(2,4,9),(12,6,3),(11,5,8),(10,1,1,7)),$
- $(12,2,5,7,9),(6,1,3),(11,10,4,8)),$
- $(6),(11,1),(12,2,10,4,5,3,7,8),(9)),$
- $(2,4,8),(6),(1,3,7),(11),(12,10,5,9)),$
- $(12,6,1,3,8),(11,10),(4,9),(2,5,7)),$
- $(6,4,10,1,3,8),(11,2,9),(12,5,7)),$
- $(2,4,7,8),(12,6,5,9),(11,10,1,3)),$
- $(10,1,4,8),(6,12),(11,2,5,7,3,9)),$
- $((11,12,10,7,9),(2,5,3),(6,1,4,8)))$

Wynik uzyskany z zastosowaniem algorytmu TS.

Legenda:

- $\square$  L - czas wykonania roboty ! w obiekcie !
- $\square$  L<sup>1</sup> - najwcześniejszy termin rozpoczęcia wykonania wk. roboty
- $\square$  L<sup>2</sup> - najpóźniejszy termin rozpoczęcia wykonania wk. roboty
- $\square$  L<sup>3</sup> - najwcześniejszy termin zakończenia wykonania wk. roboty
- $\square$  L<sup>4</sup> - najpóźniejszy termin zakończenia wykonania wk. roboty
- L - zapas czasu wk. roboty
- ! - numer obiektu
- ! - numer roboty
- ! - numer obiektu
- ! - czas wykonania roboty ! w obiekcie !

! - czas wykonania roboty ! w obiekcie !

! - numer obiektu

! - numer roboty

! - najwcześniejszy termin rozpoczęcia wykonania wk. roboty

! - najpóźniejszy termin rozpoczęcia wykonania wk. roboty

! - najwcześniejszy termin zakończenia wykonania wk. roboty

! - najpóźniejszy termin zakończenia wykonania wk. roboty

L - zapas czasu wk. roboty

! - numer obiektu

! - czas wykonania roboty ! w obiekcie !

! - numer roboty

! - najwcześniejszy termin rozpoczęcia wykonania wk. roboty

! - najpóźniejszy termin rozpoczęcia wykonania wk. roboty

! - najwcześniejszy termin zakończenia wykonania wk. roboty

! - najpóźniejszy termin zakończenia wykonania wk. roboty

L - zapas czasu wk. roboty

! - numer obiektu

! - czas wykonania roboty ! w obiekcie !

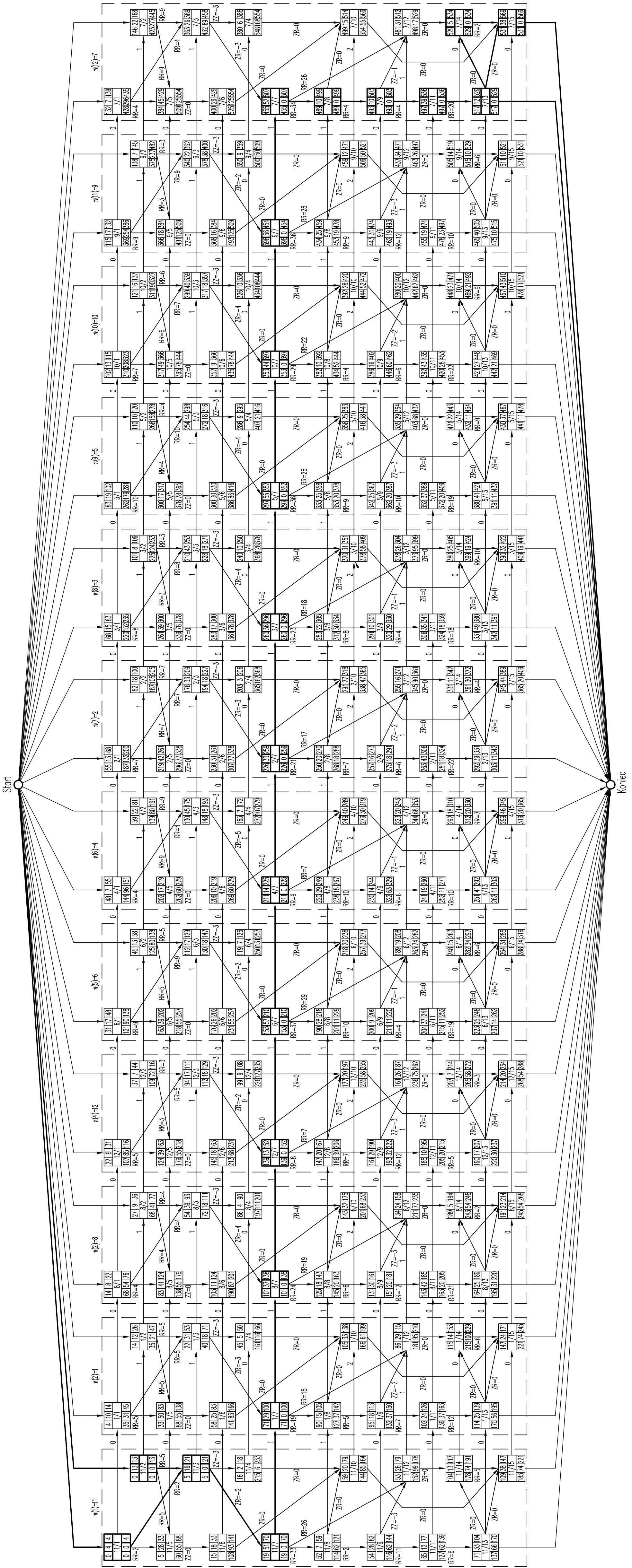
! - numer roboty

! - najwcześniejszy termin rozpoczęcia wykonania wk. roboty

! - najpóźniejszy termin rozpoczęcia wykonania wk. roboty

! - najwcześniejszy termin zakończenia wykonania wk. roboty

! - najpóźniejszy termin zakończenia wykonania wk. roboty



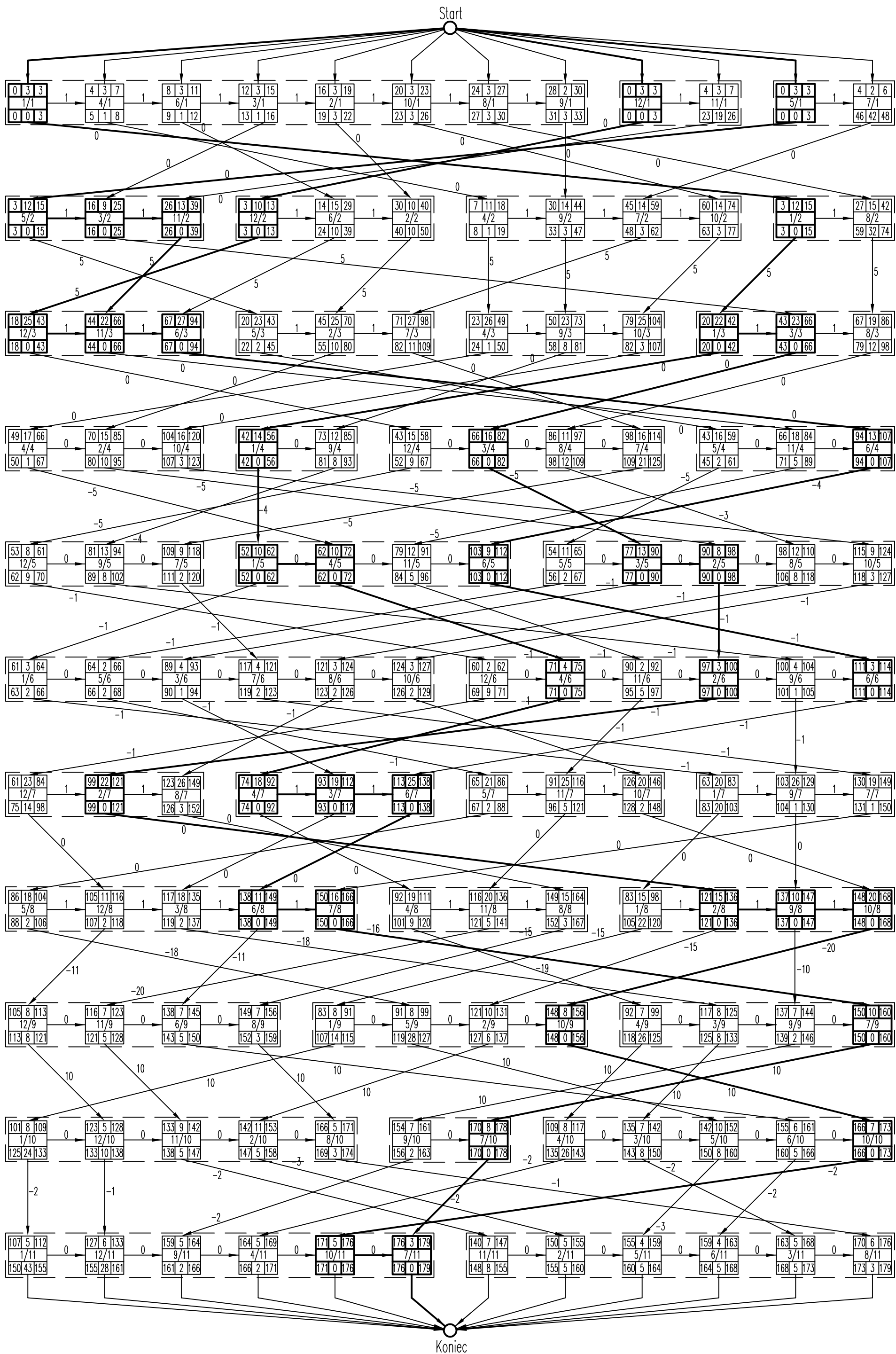
**Legenda:**

$t_j$	$t_j^*$
$t_j^*$	$t_j$
$t_j^*$	$t_j$
$t_j^*$	$t_j$

- $t_j$  - czas wykonywania roboty i w obiekcie j
- $t_j^*$  - najwcześniejszy termin rozpoczęcia wykonywania ww. roboty
- $t_j^*$  - najpóźniejszy termin zakończenia wykonywania ww. roboty
- $t_j^*$  - najwcześniejszy termin rozpoczęcia wykonywania ww. roboty
- $t_j^*$  - najpóźniejszy termin zakończenia wykonywania ww. roboty
- L - zapas czasu ww. roboty
- i - numer rodzaju roboty
- j - numer obiektu

Liniami pogrubionymi oznaczono przebieg sieci krytycznej.  
 Relacje kolejnościowe, jeśli nie oznaczono inaczej, są wyrażone jako relacje typu ZR (zakonczenie - rozpoczęcie).

**Rys. Z.3. Przykład z p. 6.2.1 – sieć zależności dla permutacji  $\pi=(11,1,8,12,6,4,2,3,5,10,9,7)$ .  
 Wynik uzyskany z zastosowaniem algorytmu TS.**



**Legenda:**

$t_{ij}$	$t_{ij}^{wz}$
$i/j$	
$t_{ij}^{pz}$	$t_{ij}^{pz}$

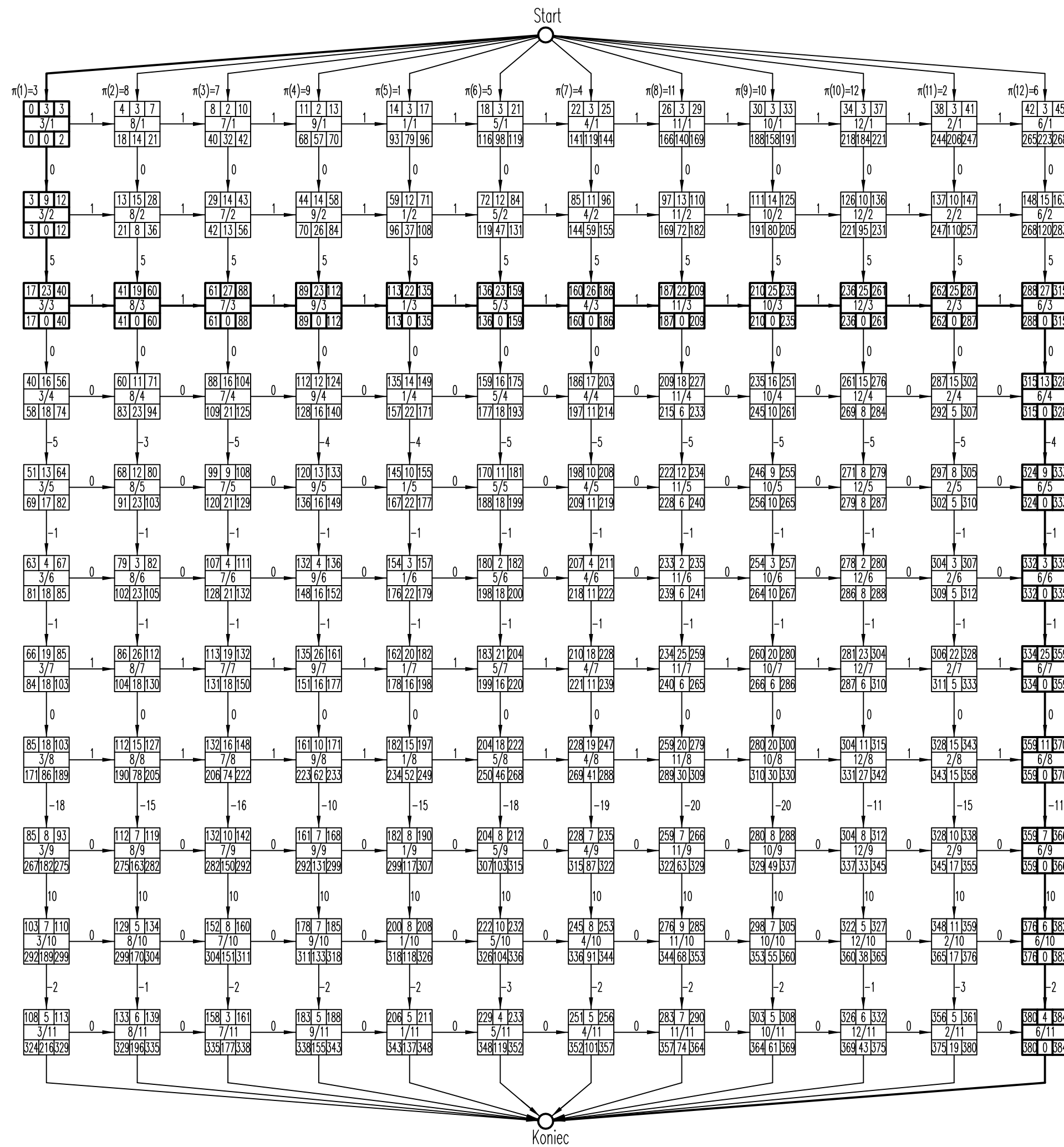
- $t_{ij}$  – czas wykonywania roboty  $i$  w obiekcie  $j$
- $t_{ij}^{wz}$  – najwcześniejszy termin rozpoczęcia wykonywania ww. roboty
- $t_{ij}^{wz}$  – najwcześniejszy termin zakończenia wykonywania ww. roboty
- $t_{ij}^{pz}$  – najpóźniejszy termin rozpoczęcia wykonywania ww. roboty
- $t_{ij}^{pz}$  – najpóźniejszy termin zakończenia wykonywania ww. roboty
- $L$  – zapas czasu ww. roboty
- $i$  – numer rodzaju roboty
- $j$  – numer obiektu

Liniami pogrubionymi oznaczono przebieg ścieżki krytycznej.  
 Relacje kolejnościowe, jeśli nie oznaczono inaczej, są wyrażone jako relacje typu ZR (zakończenie – rozpoczęcie).

Rys. Z.2. Przykład z p. 6.1.2 – sieć zależności dla permutacji  $\pi = (((1,4,6,3,2,10,8,9),(12,11),(5,7)), ((5,3,11),(12,6,2),(4,9,7,10),(1,8)), ((12,11,6),(5,2,7),(4,9,10),(1,3,8)), ((4,2,10),(1,9),(12,3,8,7),(5,11,6)), ((12,9,7),(1,4,11,6),(5,3,2,8,10)), ((1,5,3,7,8,10),(12,4,11,2,9,6)), ((12,2,8),(4,3,6),(5,11,10),(1,9,7)), ((5,12,3,6,7),(4,11,8),(1,2,9,10)), ((12,11,6,8),(1,5,2,10),(4,3,9,7)), ((1,12,11,2,8),(9,7),(4,3,5,6,10)), ((1,12,9,4,10,7),(11,2,5,6,3,8))))).$

Wynik uzyskany z zastosowaniem algorytmu TS.





**Legenda:**

$t_{ij}^w$	$t_{ij}^z$	$t_{ij}^w$
$t_{ij}^w$	$i/j$	$t_{ij}^z$
$t_{ij}^w$	$L$	$t_{ij}^z$

- $t_{ij}$  – czas wykonywania roboty  $i$  w obiekcie  $j$
- $t^w$  – najwcześniejszy termin rozpoczęcia wykonywania ww. roboty
- $t^z$  – najwcześniejszy termin zakończenia wykonywania ww. roboty
- $t^w$  – najpóźniejszy termin rozpoczęcia wykonywania ww. roboty
- $t^z$  – najpóźniejszy termin zakończenia wykonywania ww. roboty
- $L$  – zapas czasu ww. roboty
- $i$  – numer rodzaju roboty
- $j$  – numer obiektu

Liniami pogrubionymi oznaczono przebieg ścieżki krytycznej.  
 Relacje kolejnościowe, jeśli nie oznaczono inaczej, są wyrażone jako relacje typu ZR (zakończenie – rozpoczęcie).

Rys. Z.1. Przykład z p. 6.1.1 – sieć zależności dla permutacji  $\pi=(3,8,7,9,1,5,4,11,10,12,2,6)$ . Wynik uzyskany z zastosowaniem algorytmu TS.