



POLITECHNIKA ŚLĄSKA  
WYDZIAŁ AUTOMATYKI, ELEKTRONIKI I INFORMATYKI  
KIERUNEK AUTOMATYKA I ROBOTYKA

Projekt inżynierski

Strategia Just in Time w systemach produkcyjnych  
- analiza struktury gniazdowej dla heurystyk FIFO i LIFO.

Autor: Kamil Strzempowicz  
Kierujący pracą: dr inż. Waldemar Grzechca

Gliwice, styczeń 2013



*Rodzicom*  
*bez Was nie udałoby mi się*



# Spis treści

<b>1. Wstęp</b>	<b>3</b>
<b>2. Metody szeregowania zadań</b>	
<b>w systemach wytwarzania gniazdowego</b>	<b>5</b>
2.1. Metody heurystyczne . . . . .	5
2.2. Inne metody . . . . .	7
<b>3. Heurystyki FIFO i LIFO w JSSP</b>	<b>9</b>
3.1. First In First Out . . . . .	11
3.2. Last In First Out . . . . .	12
<b>4. Program kSzereg</b>	<b>13</b>
<b>5. Zastosowanie programu kSzereg</b>	<b>17</b>
5.1. Badanie zlecenia . . . . .	17
5.1.1. Heurystyka FIFO . . . . .	17
5.1.2. Heurystyka LIFO . . . . .	19
5.2. Zmodyfikowane zlecenie ze zmienioną marszrutą . . . . .	21
5.2.1. Heurystyka FIFO . . . . .	21
5.2.2. Heurystyka LIFO . . . . .	23
5.3. Zmodyfikowane zlecenie bez zmian w marszrucie . . . . .	25
5.3.1. Heurystyka FIFO . . . . .	25
5.3.2. Heurystyka LIFO . . . . .	27
5.4. Zmodyfikowane zlecenie z naciskiem na $e_j$ . . . . .	29
5.4.1. Heurystyka FIFO . . . . .	29
5.4.2. Heurystyka LIFO . . . . .	31
<b>6. Wnioski</b>	<b>33</b>
<b>7. Literatura</b>	<b>35</b>



# 1 Wstęp

Celem tej pracy jest przedstawienie strategii *Just in Time* (JIT) dla szeregowania zadań w systemach produkcyjnych o strukturze gniazdowej. Szeregowanie przeprowadzono na podstawie heurystyk *First In First Out* (FIFO) i *Last In First Out* (LIFO). Na potrzeby tej pracy powstał program *kSzereg* szeregujący zadania wykorzystując obie te heurystyki, oraz obliczający wyznaczniki jakości uszeregowania w kontekście strategii JIT.

System produkcyjny jest to celowo zaprojektowany i zorganizowany układ materialny, energetyczny i informacyjny eksploatowany przez człowieka i służący produkowaniu określonych produktów (wyrobów lub usług) w celu zaspokajania różnorodnych potrzeb konsumentów [1].

Systemy produkcyjne można podzielić ze względu na strukturę na:

- Systemy gniazdowe - najczęściej wykorzystywane do pojedynczych zamówień, bądź krótkich serii. Takie systemy zwykle zmieniają swoje zastosowanie po zakończeniu każdego zlecenia. Zlecenie składa się ze skończonej liczby zadań, a każde z nich wymaga przeprowadzenia zestawu operacji na maszynach w ustalonym porządku, innym dla każdego zadania.
- System przepływowy - kierunek przepływu wszystkich zleceń przez maszyny jest ten sam. Dla idealnego systemu przepływowego liczba zadań jest równa liczbie maszyn dla każdego zlecenia. [2]
- Pojedyncza maszyna - system składający się z jednej tylko maszyny. Zakłada się, że procesy wytwarzania produktów są jednostadialne, a zlecenia jednostkowe. Stąd zlecenia są tożsame z zadaniami, przy czym nie istnieją ograniczenia kolejnościowe [2].
- Maszyny równoległe - system składający się z  $m$  identycznych maszyn. Podobnie jak w problemie jednomaszynowym zakłada się, że procesy wytwarzania produktów są jednostadialne, a zlecenia jednostkowe. Stąd, podobnie jak tam, zlecenia są tożsame z zadaniami, przy czym nie istnieją ograniczenia kolejnościowe [2].

- Linia montażowa - zespół stanowisk roboczych (maszynowych, ręcznych lub mieszanych) ugrupowanych według kolejności operacji procesu technologicznego [3].

Szeregowanie może zostać zdefiniowane jako przydział zasobów w czasie do zadań w celu optymalizacji kryterium. Z punktu widzenia szeregowania w systemach wytwarzania zasoby często są nazywane maszynami, natomiast jako kryterium optymalności uszeregowania często przyjmowany jest czas potrzebny do zakończenia wszystkich zadań, czy maksymalne spóźnienie zadania [4].

W strategii *Just in Time* zadanie powinno być ukończone możliwie blisko swojego terminu zakończenia (due date) jak to tylko możliwe. Zbyt wczesne zakończenie zadania pociąga za sobą koszty utrzymania, takie jak magazynowania czy ubezpieczenia. Z drugiej jednak strony spóźnione zlecenie często skutkuje karami umownymi czy nadszarpnięciem reputacji przedsiębiorstwa [5]. Mimo to często przedsiębiorstwu bardziej opłaca się zakończyć zadanie krótko po terminie, niż ponosić koszty związane ze zbyt wczesnym zakończeniem pracy. Należy się więc zastanowić jak matematycznie opisać opłacalność zarówno spóźnienia jak i przedwczesnego ukończenia zadania. Taką rolę pełnią tzw. funkcje kosztu.

Na potrzeby tej pracy wybrano następujące funkcje (wyznaczniki):

$$\sqrt{\sum e_j^2 + \sum l_j^2} \quad (1)$$

$$\alpha * \sum e_j + \beta * \sum l_j \quad (2)$$

gdzie:

$e_j$  - przedwczesność j-tego zadania,

$l_j$  - spóźnienie j-tego zadania,

$\alpha, \beta$  - wagi przedwczesności i spóźnienia zakładane przy formułowaniu problemu

Zastosowanie strategii *Just in Time* w systemach produkcyjnych o strukturze gniazdowej można więc sprowadzić do poszukiwania takiego rozwiązania problemu harmonogramowania, aby minimalizować funkcję 1 lub 2.



## 2 Metody szeregowania zadań w systemach wytwarzania gniazdowego

Teoretycznie optymalne rozwiązanie problem szeregowania zadań w systemie wytwarzania gniazdowego (JSSP) można dokonać na podstawie przeglądu zupełnego, ponieważ zawsze ilość możliwych rozwiązań zawsze jest skończona. Jednak, jak wynika m. in. z [6][7], problem ten jest silnie NP-trudny, czyli pozostaje NP-trudny nawet przy ograniczeniu maksymalnej wartości występujących w jego opisie liczb. Natomiast problem NP-trudny jest co najmniej tak trudny, jak każdy problem klasy NP (nondeterministic polynomial), a często trudniejszy. Oznacza to, że już przy niewielkiej ilości zadania problem przestaje być rozwiązywalny w rozsądnym czasie nawet przy użyciu superkomputerów. W związku z tym wykorzystuje się metody pozwalające uzyskać nieoptymalne, ale akceptowalne rozwiązanie bez potrzeby rozpatrywania każdego możliwego rozwiązania [8].

### 2.1 Metody heurystyczne

Metody heurystyczne pozwalają na stosunkowo szybkie i łatwe znalezienie dopuszczalnego rozwiązania. Zwykle jednak nie jest to rozwiązanie optymalne, lecz niejednokrotnie bardziej opłacalne jest wdrożenie takiego rozwiązania, niż żmudne poszukiwanie lepszego rozwiązania, ze względu na czas i zasoby wymagane do jego odnalezienia. W szczególności w sytuacji awaryjnej, gdy nie ma czasu na wykorzystanie bardziej zaawansowanych metod [9].

Metody te zwykle polegają na nadawaniu priorytetu zadaniom w momencie wystąpienia konfliktu na podstawie danych zlecenia i chwili czasu. Konflikt jest to sytuacja, gdy w tym samym momencie na dany wolny zasób (maszynę) czeka więcej niż jedno zadanie. Trzeba wtedy zdecydować które z zadań zostanie obsłużone jako pierwsze.

Najpopularniejsze heurystyki[10]:

- First In First Out - Pierwsze jest przetwarzane zadanie, które wpłynęło najwcześniej.

- Last In First Out - Pierwsze jest przetwarzane zadanie, które wpłynęło najpóźniej.
- Earliest Due Date - Pierwsze jest przetwarzane zadanie, które ma zostać wyprodukowane najwcześniej.
- Least Work Remaining - Pierwsze jest przetwarzane zadanie, do ukończenia którego pozostało najmniej pracy.
- Shortest Processing Time - Pierwsze jest przetwarzane zadanie, którego całkowity czas przetwarzania jest najkrótszy.
- Longest Processing Time - Pierwsze jest przetwarzane zadanie, którego całkowity czas przetwarzania jest najdłuższy.
- Shortest Imminent Processing Time - Pierwsze jest przetwarzane zadanie, którego czas przetwarzania na tej maszynie jest najkrótszy.
- Longest Imminent Processing Time - Pierwsze jest przetwarzane zadanie, którego czas przetwarzania na tej maszynie jest najdłuższy.
- Fewest Operations Remaining - Pierwsze jest przetwarzane zadanie, do ukończenia którego potrzeba najmniejszej ilości operacji.
- Most Operations Remaining - Pierwsze jest przetwarzane zadanie, do ukończenia którego potrzeba największej ilości operacji.
- Least Slack Time - Pierwsze jest przetwarzane zadanie, którego różnica między czasem potrzebnym do jego ukończenia a czasem pozostałym do terminu zakończenia pracy jest najmniejsza.
- Least Slack Time per Operation - Pierwsze jest przetwarzane zadanie, którego stosunek czasu 'Slack' w stosunku do ilości pozostałych operacji jest najmniejsza.
- Critical Ratio - Pierwsze jest przetwarzane zadanie, którego stosunek czasu pozostałego do terminu ukończenia pracy i czasu potrzebnego do jego ukończenia jest najmniejszy.
- Random - Losowe zadanie jest przetwarzane jako pierwsze.

## 2.2 Inne metody

Istnieją też nieheurystyczne metody szeregowania zadań w systemach wytwarzania gniazdowego. Zwykle wymagają one więcej czasu i zasobów niż metody heurystyczne, jednocześnie nie gwarantują uzyskania lepszego rozwiązania. Na przykład metoda podziałów i oszacowań, która pozwala 'odciąć' nieoptymalne gałęzie z grafu wszystkich rozwiązań i pozwala na znalezienie rozwiązania optymalnego, jednak przy wielu zadaniach trwa to bardzo długo [11]. Często po znalezieniu satysfakcjonującego rozwiązania algorytm jest przerywany.

Do znalezienia rozwiązania JSSP można wykorzystać także algorytmy genetyczne, które są adaptacyjnym narzędziem ukierunkowanego poszukiwania optimum funkcji celu (np. (1), czy (2)), które pozwala na równoległe przeszukiwanie zakodowanej przestrzeni alternatywnych rozwiązań. Procedura ta wykorzystuje procesy losowe do określenia kierunku przeszukiwania. Idea algorytmów genetycznych naśladuje zasadę działania rzeczywistego mechanizmu ewolucyjnego, czyli darwinowską strategię przetrwania osobników biologicznie najlepiej przystosowanych. Możemy zasymulować ten proces i zastosować go do rozwiązywania problemów trudnych obliczeniowo, dla których nie jesteśmy w stanie skonstruować algorytmu dającego dokładne rozwiązanie w rozsądnym czasie (problemy klasy NP), przy czym w ogólnym przypadku uzyskujemy rozwiązania suboptymalne, za to stosunkowo szybko [2]. Dodatkowo wyniki działania algorytmu genetycznego można poprawić, np. w [5] zastosowano algorytm lokalnego przeszukiwania.



### 3 Heurystyki FIFO i LIFO w JSSP

Algorytm rozwiązywania problemu szeregowania zadań w systemie wytwarzania gniazdowego za pomocą heurystyk FIFO i LIFO ilustruje poniższy przykład. Dane zlecenia zostały przedstawione w formie tabeli, gdzie:

$j$  - numer zadania

$r_j$  - czas pojawienia się zadania w systemie

$d_j$  - termin ukończenia zadania

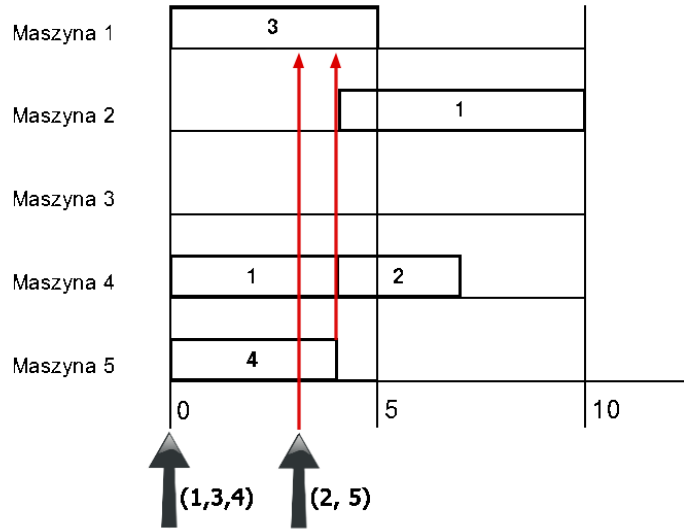
m1-m5 - numer maszyny na której zadanie ma być przetwarzane w czasie zapisanym w nawiasie.

Tablica 1: Struktura zlecenia

$j$	$r_j$	$d_j$	Marszruta technologiczna
1	0	20	m4 (4) - m2 (6) - m4 (5)
2	3	29	m4 (3) - m3 (6) - m1 (5) - m4 (5)
3	0	20	m1 (5) - m5 (4)
4	0	20	m5 (4) - m1 (4) - m2 (6)
5	3	20	m1 (3) - m5 (4)

Tablica 2: Algorytm szeregowania zadań

t	Zdarzenie
0	Do systemu wchodzi zadania 1, 3 i 4, które zostają przydzielone zgodnie ze swoimi marszrutami technologicznymi odpowiednio do maszyn m4, m1 i m5.
3	Do systemu wchodzi zadania 2 i 5, lecz maszyny, na których powinno zacząć się ich przetwarzanie są zajęte, czekają więc na zwolnienie maszyny.
4	Na maszynie 5 zakończyło się przetwarzanie zadania 4. Na tę maszynę żadne zadanie nie oczekiwało. Na maszynie 4 zakończyło się przetwarzanie zadania 1. Na tę maszynę oczekuje tylko zadanie 2, które zostaje od razu na niej uruchomione. Zadanie 1 natomiast czeka na zwolnienie się maszyny 1.
5	Przetwarzanie zadanie 3 na maszynie 1 dobiegło końca. Na tę maszynę oczekiwały dwa zadania: 1 i 5, więc nastąpił tzw. konflikt.

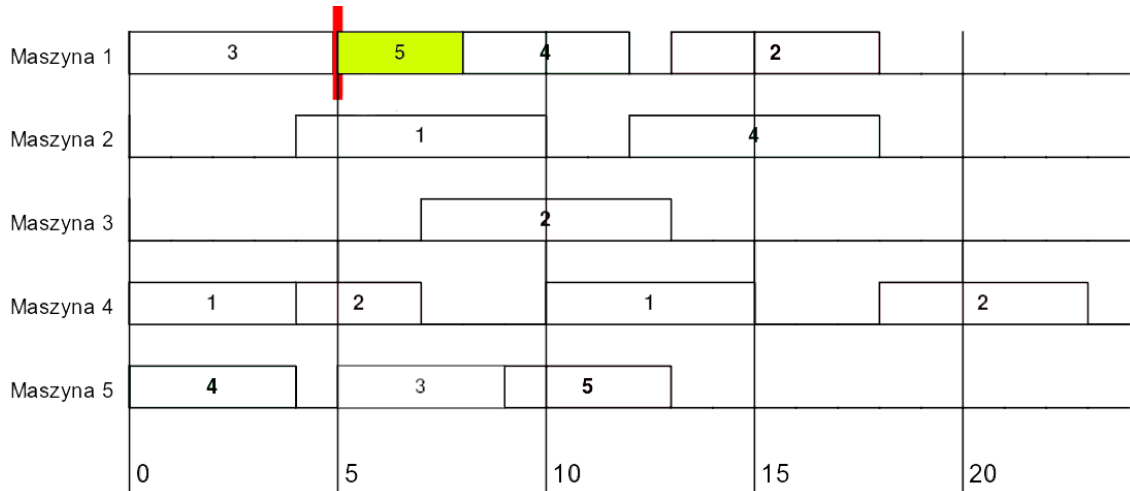


Rysunek 1: Uszeregowanie zadań w chwili wystąpienia konfliktu

Heurystyki służą do rozwiązywania takich właśnie konfliktów. Jeżeli zdecydujemy się na heurystykę FIFO dostęp do maszyny uzyska to zadanie, które oczekuje najdłużej (jako pierwsze zajęło miejsce w kolejce”), w tym przypadku jest to zadanie 5. Heurystyka LIFO faworyzuje zadania najkrócej oczekujące na zasób, w tym przypadku więc będzie to zadanie 4. Jeżeli nastąpiłby konflikt między dwoma równoprawnymi pod względem wybranej heurystyki zadaniami, należy wtedy zastosować dowolną inną ”pomocniczą” heurystykę, choćby Random, czyli wybranie losowego zadania.

Dalsze rozwiązanie tego przykładu krok po kroku nie zostanie tu zaprezentowane, natomiast końcowe uszeregowanie w zależności od wybranej heurystyki przedstawiają poniższe wykresy Gantt’a oraz wyznaczniki jakości uszeregowania. Różnice między tymi dwoma uszeregowaniami wydają się być kosmetyczne, wynika to jednak z trywialności omawianego przykładu. Podczas analizy tych wyznaczników widać różnice między uszeregowaniami. W tym konkretnym przypadku wybór heurystyki nie wpłynął na  $C_{max}$ , czyli czas ukończenia całego zlecenia. Jednak trochę lepsze rozwiązanie, ze względu na  $\bar{F}$  oraz  $\sqrt{\sum e_j^2 + \sum l_j^2}$ , dała heurystyka FIFO.

### 3.1 First In First Out



Rysunek 2: Uszeregowanie przy użyciu heurystyki FIFO

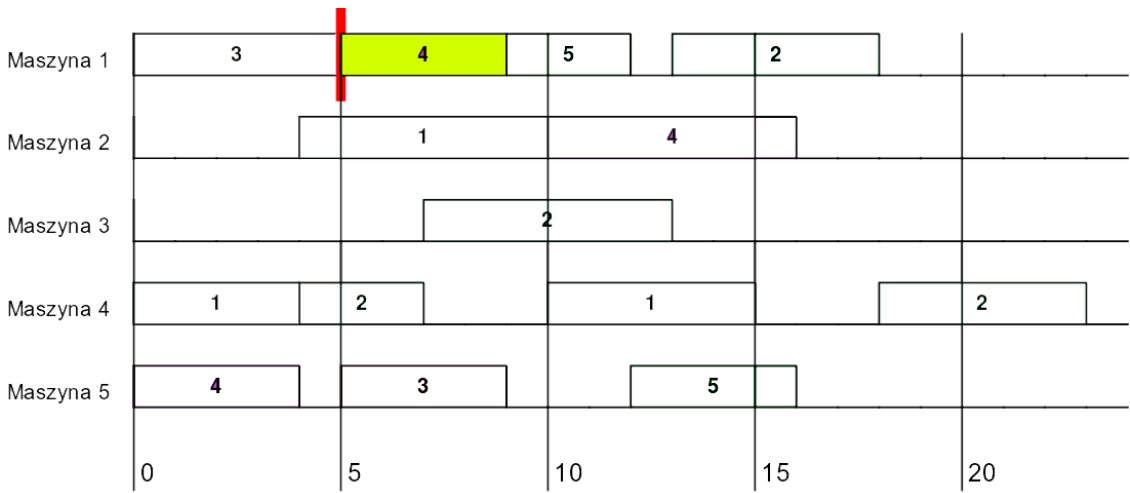
Tablica 3: Parametry wykonanych zadań

j	$c_j$	$f_j$	$l_j$	$e_j$
1	15	15	0	5
2	23	20	0	6
3	9	9	0	11
4	18	18	0	2
5	13	10	0	7

$$C_{max} = 23 \quad \sqrt{\sum e_j^2 + \sum l_j^2} = 15.3297$$

$$\bar{F} = 14.4 \quad \alpha * \sum e_j + \beta * \sum l_j \Big|_{\substack{\alpha=0.4 \\ \beta=0.6}} = 12.4$$

3.2 Last In First Out



Rysunek 3: Uszeregowanie przy użyciu heurystyki LIFO

Tablica 4: Parametry wykonanych zadań

j	$c_j$	$f_j$	$l_j$	$e_j$
1	15	15	0	5
2	23	20	0	6
3	9	9	0	11
4	16	16	0	4
5	16	13	0	4

$$C_{max} = 23 \quad \sqrt{\sum e_j^2 + \sum l_j^2} = 14.6287$$
$$\bar{F} = 14.6 \quad \alpha * \sum e_j + \beta * \sum l_j \Big|_{\substack{\alpha=0.4 \\ \beta=0.6}} = 12$$



## 4 Program kSzereg

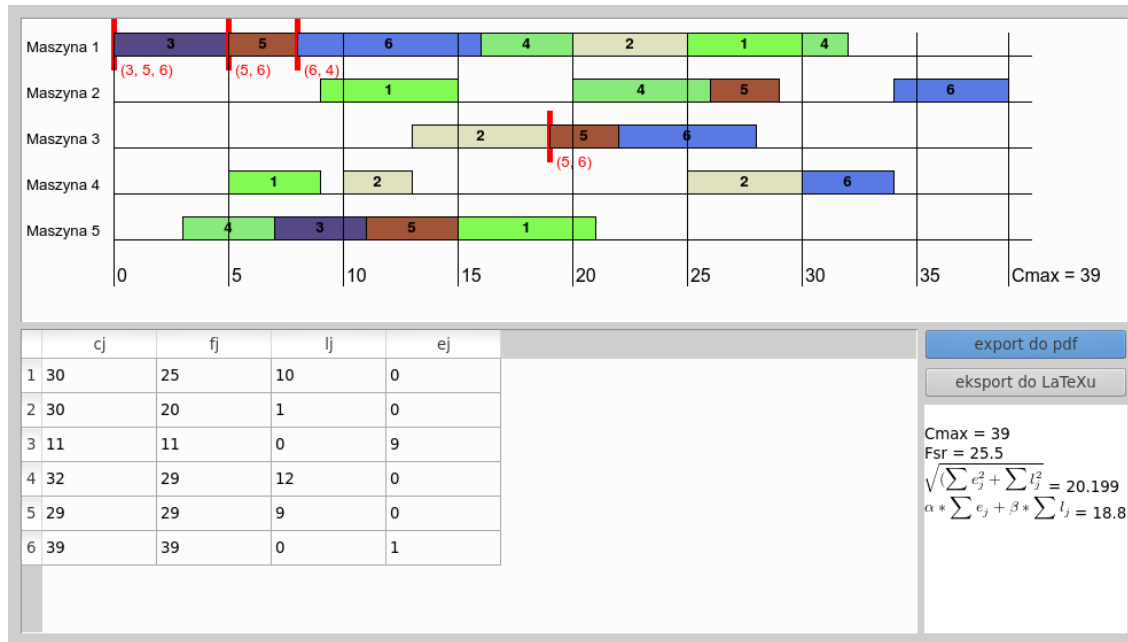
Program kSzereg został napisany w C++ na potrzeby tej pracy dyplomowej. Umożliwia on przeprowadzenie szeregowania zadań w systemie wytwarzania gniazdowego na podstawie heurystyki FIFO bądź LIFO. Wprowadzanie danych odbywa się za pośrednictwem graficznego interfejsu opartego o framework Qt. Na górnej belce programu po prawej stronie znajdują się kontrolki umożliwiające wybór heurystyki używanej do rozwiązania zadania oraz parametrów  $\alpha, \beta$  podstawianych do wzoru (2). Znajdują się tam też przyciski umożliwiające zapisanie zlecenia do pliku binarnego z rozszerzeniem .mar (eksport) oraz wczytanie zlecenia z takiego pliku (import).

Przed dodaniem zadań do zlecenia należy ustalić maksymalną długość marszruty. W zadaniach z krótszą marszrutą technologiczną należy pozostawić operacje z zerowym czasem, które nie będą brane pod uwagę podczas rozwiązywania JSSP. Po dodaniu zadania możliwość zmiany długości marszruty jest blokowana. Każda operacja jest definiowana przez numer maszyny, na której ma zostać wykonana oraz czas jej wykonywania. Ilość maszyn możliwych do wyboru może być zmieniana w czasie edycji zlecenia (po dodaniu zadań). Poza marszrutą technologiczną zadanie definiują czasy rozpoczęcia ( $r_j$ ) i zakończenia ( $d_j$ ) zadania. Możliwe jest też nadanie zadaniu nazwy.

Ilość maszyn: 5		Długość marszruty: 4		Dodaj zlecenie		Dane: Import Export		β: 0.30 α: 0.00		Metoda: FIFO		Rozwiąż	
1	punto	5	20	Maszyna m4	Czas 4	Maszyna m2	Czas 6	Maszyna m4	Czas 5	Maszyna m1	Czas 0		
2	126p	10	29	Maszyna m4	Czas 3	Maszyna m3	Czas 6	Maszyna m1	Czas 5	Maszyna m4	Czas 5		
3	brava	0	20	Maszyna m1	Czas 5	Maszyna m5	Czas 4	Maszyna m1	Czas 0	Maszyna m1	Czas 0		
4	golf	3	20	Maszyna m5	Czas 4	Maszyna m1	Czas 4	Maszyna m2	Czas 6	Maszyna m1	Czas 0		
5	pasat	0	20	Maszyna m1	Czas 3	Maszyna m5	Czas 4	Maszyna m1	Czas 0	Maszyna m1	Czas 0		

Rysunek 4: Główne okno programu

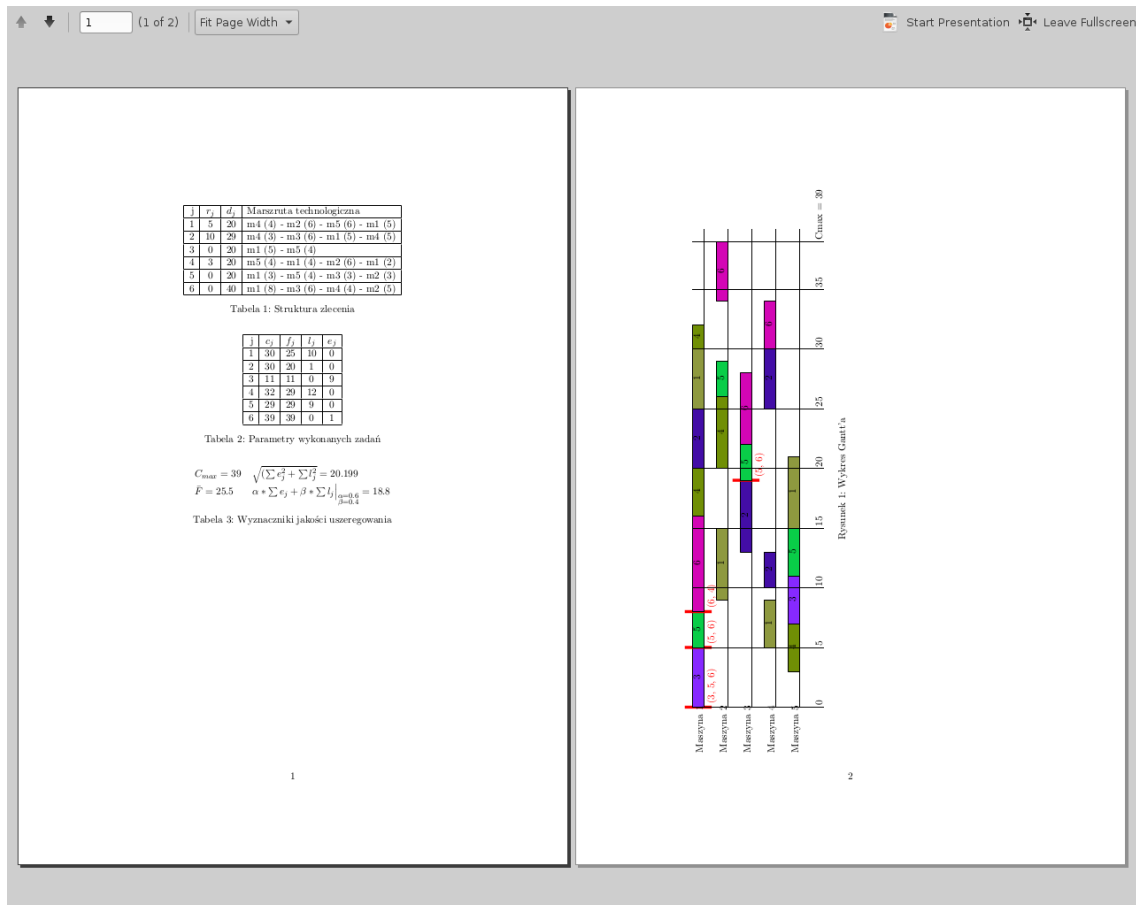
Program kSzereg rezentuje wyniki działania w postaci wykresu Gantt'a wraz z tabelą zawierającą czasy wykonania ( $C_j$ ), przepływu ( $f_j$ ), spóźnienia ( $l_j$ ) i przedwczesności (earliness,  $e_j$ ) każdego zadania zlecenia. Dodatkowo prezentowane są wyznaczniki jakości uszeregowania zlecenia jako całości: czas ukończenia całego zlecenia ( $C_{max}$ ), średni czas przepływu ( $\bar{F}$ ), maksymalne spóźnienie ( $T_{max}$ ), średnie spóźnienie ( $\bar{T}$ ) oraz wartości funkcji kosztów (1) oraz (2).



Rysunek 5: Prezentacja wyznaczonego uszeregowania

Z poziomu okna wyznaczonego uszeregowania istnieje możliwość zapisania wyników działania programu w postaci pliku .tex z kodem w  $\text{\LaTeX}$ -u lub .pdf. W obu przypadkach sam wykres Gantt'a jest eksportowany do pliku .svg w folderze output. Funkcje te korzystają z zewnętrznych programów inkscape oraz rubber (wraz z odpowiednimi paczkami texlive), które muszą być zainstalowane w systemie.

Dodatkowo istnieje możliwość obsługi programu kSzereg z poziomu terminalu tekstowego korzystając z wcześniej zapisanych plików .mar, nie jest bowiem możliwa konfiguracja zadania jedynie, rozwiązanie problemu i eksport wyników do pliku .tex lub .pdf. Zaletą obsługi programu z terminalu jest możliwość podania jako parametr nazwy pliku tekstowego z zapisanymi ścieżkami dostępu do wielu plików .mar, a nawet wybranie obu heurystyk do rozwiązania tych zleceń, co spowoduje rozwiązanie JSSP dla każdego z tych zadań. Szczegółowy opis intreface'u linii poleceń jest dostępny po uruchomieniu programu z przełącznikiem '-help'.



Rysunek 6: Wyeksportowany plik .pdf

```
File Edit Tabs Help
konserw@kon... x konserw@kon... x
konserw@konserw-bodhi:~/git/inzynierka$ ./kSzereg --help
usage: kSzereg [-h <heuristics>] [-l <list>] [-p] [-m] [--help]
or kSzereg without arguments for GUI operation [spectratio=true] {./obrazki/wykres}
Strategy: Just in Time in manufacturing systems, FIFO and LIFO heuristics analysis
for job shop problem.

OPTIONS:
  -h, --help            Display this help text and exit
  -m, --maximize        GUI operation with maximized window
  -h, --heuristic <type> Set heuristic used to resolve conflicts. FIFO is default
  -l, --list <list>     Import list of .mar files to process from <list>
  -p, --pdf             Compile LaTeX output to .pdf format

Available heuristics:
  FIFO                First In First Out
  LIFO                Last In First Out
  all                 Apply all heuristics subsequently

konserw@konserw-bodhi:~/git/inzynierka$
```

Rysunek 7: Pomoc interface'u linii poleceń



## 5 Zastosowanie programu kSzereg

### 5.1 Badanie zlecenia

#### 5.1.1 Heurystyka FIFO

Tablica 5: Struktura zlecenia

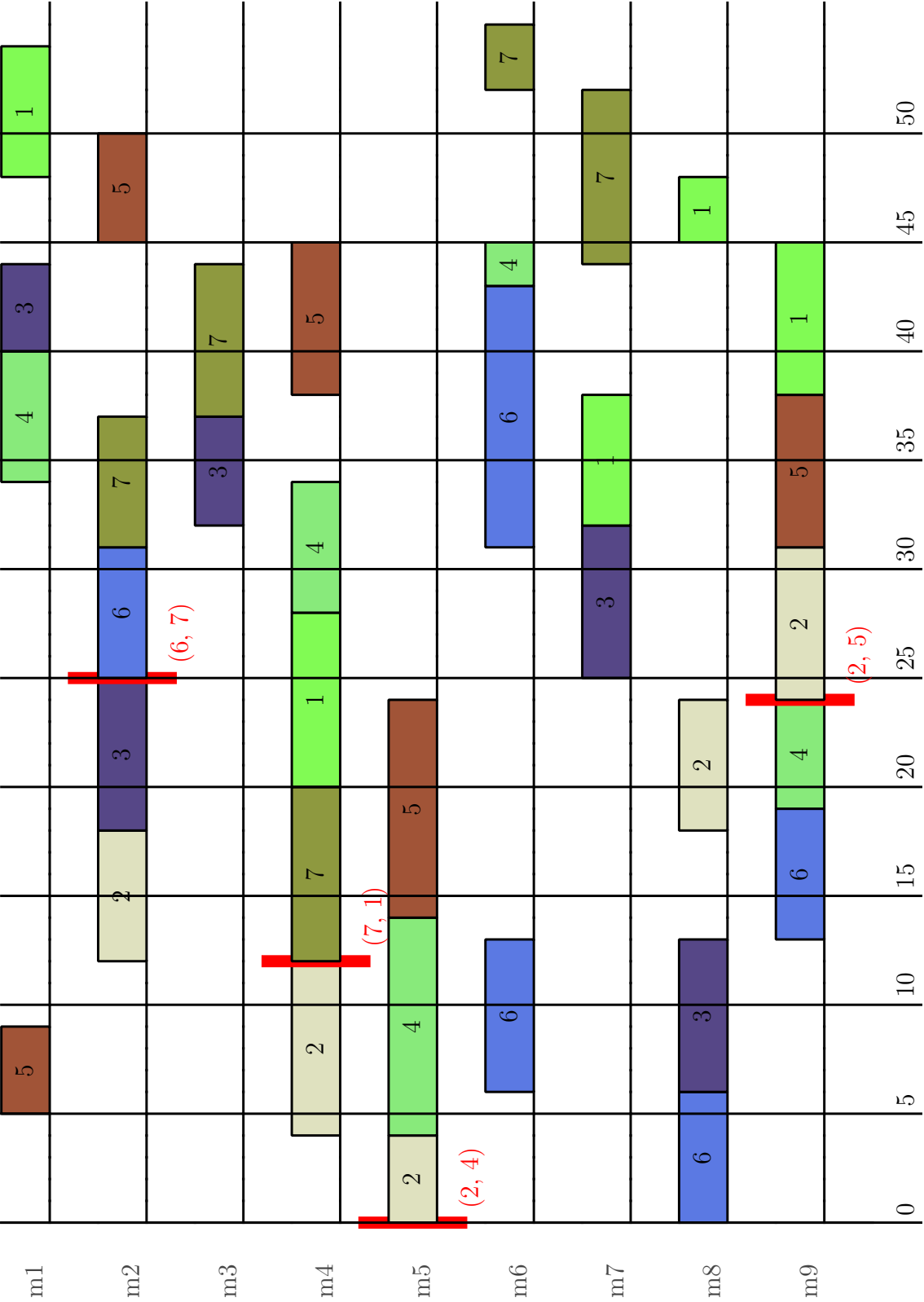
j	$r_j$	$d_j$	Marszruta technologiczna
1	10	55	m4 (8) - m7 (6) - m9 (7) - m8 (3) - m1 (6)
2	0	45	m5 (4) - m4 (8) - m2 (6) - m8 (6) - m9 (7)
3	5	50	m8 (7) - m2 (7) - m7 (7) - m3 (5) - m1 (4)
4	0	40	m5 (10) - m9 (5) - m4 (6) - m1 (6) - m6 (2)
5	5	45	m1 (4) - m5 (10) - m9 (7) - m4 (7) - m2 (5)
6	0	40	m8 (6) - m6 (7) - m9 (6) - m2 (6) - m6 (12)
7	5	45	m4 (8) - m2 (6) - m3 (7) - m7 (8) - m6 (3)

Tablica 6: Parametry wykonanych zadań

j	$c_j$	$f_j$	$l_j$	$e_j$
1	54	44	0	1
2	31	31	0	14
3	44	39	0	6
4	45	45	5	0
5	50	45	5	0
6	43	43	3	0
7	55	50	10	0

$$C_{max} = 55 \quad T_{max} = 10 \quad \sqrt{\sum e_j^2 + \sum l_j^2} = 19.799$$

$$\bar{F} = 42.4286 \quad \bar{T} = 3.28571 \quad \alpha * \sum e_j + \beta * \sum l_j \Big|_{\substack{\alpha=0.2 \\ \beta=0.8}} = 22.6$$



Rysunek 8: Wykres Gantt'a

## 5.1.2 Heurystyka LIFO

Tablica 7: Struktura zlecenia

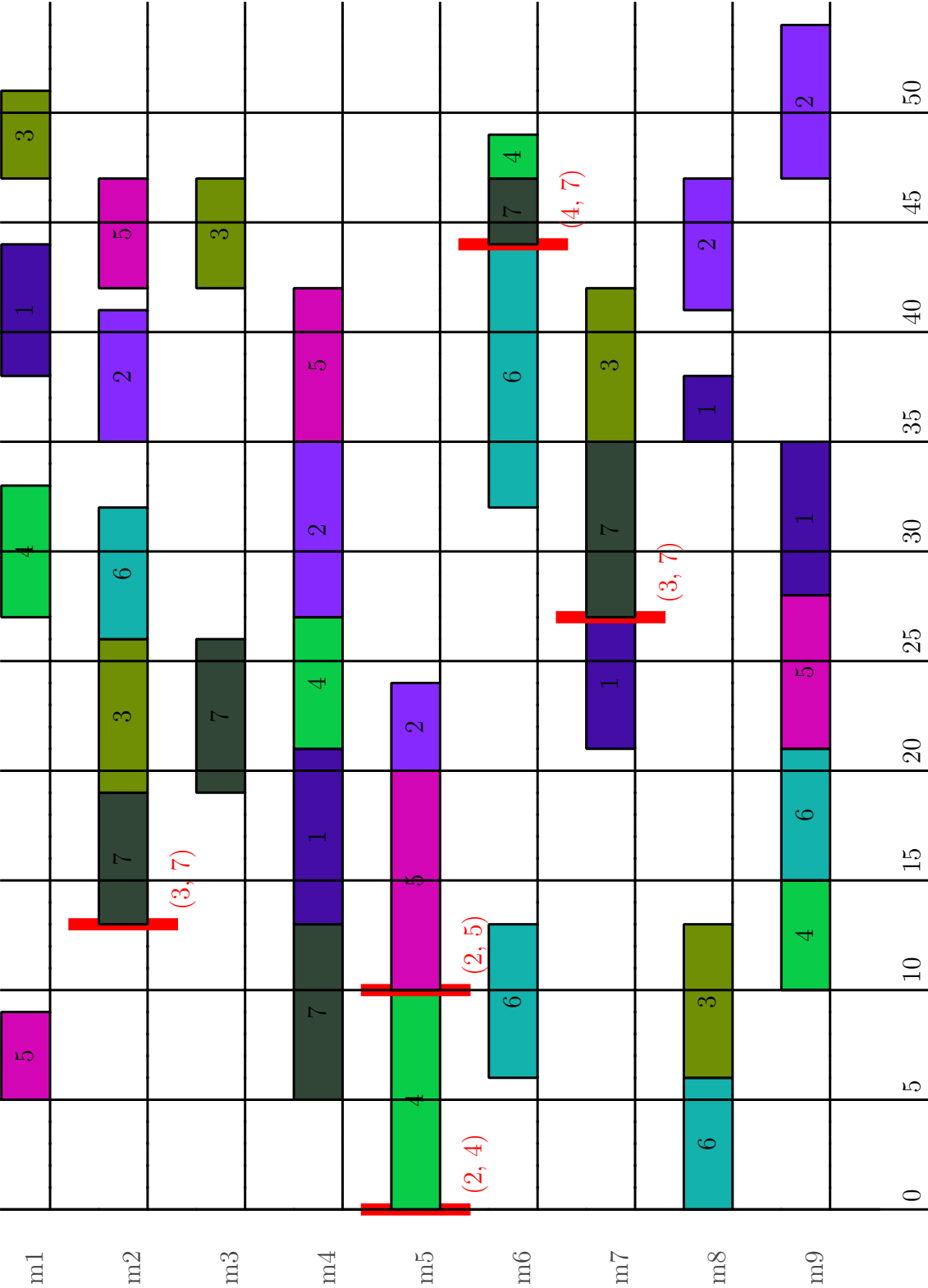
j	$r_j$	$d_j$	Marszruta technologiczna
1	10	55	m4 (8) - m7 (6) - m9 (7) - m8 (3) - m1 (6)
2	0	45	m5 (4) - m4 (8) - m2 (6) - m8 (6) - m9 (7)
3	5	50	m8 (7) - m2 (7) - m7 (7) - m3 (5) - m1 (4)
4	0	40	m5 (10) - m9 (5) - m4 (6) - m1 (6) - m6 (2)
5	5	45	m1 (4) - m5 (10) - m9 (7) - m4 (7) - m2 (5)
6	0	40	m8 (6) - m6 (7) - m9 (6) - m2 (6) - m6 (12)
7	5	45	m4 (8) - m2 (6) - m3 (7) - m7 (8) - m6 (3)

Tablica 8: Parametry wykonanych zadań

j	$c_j$	$f_j$	$l_j$	$e_j$
1	44	34	0	11
2	54	54	9	0
3	51	46	1	0
4	49	49	9	0
5	47	42	2	0
6	44	44	4	0
7	47	42	2	0

$$C_{max} = 54 \quad T_{max} = 9 \quad \sqrt{\sum e_j^2 + \sum l_j^2} = 17.5499$$

$$\bar{F} = 44.4286 \quad \bar{T} = 3.85714 \quad \alpha * \sum e_j + \beta * \sum l_j \Big|_{\substack{\alpha=0.2 \\ \beta=0.8}} = 23.8$$



Rysunek 9: Wykres Gantt’a



## 5.2 Zmodyfikowane zlecenie ze zmienioną marszrutą

### 5.2.1 Heurystyka FIFO

Tablica 9: Struktura zlecenia

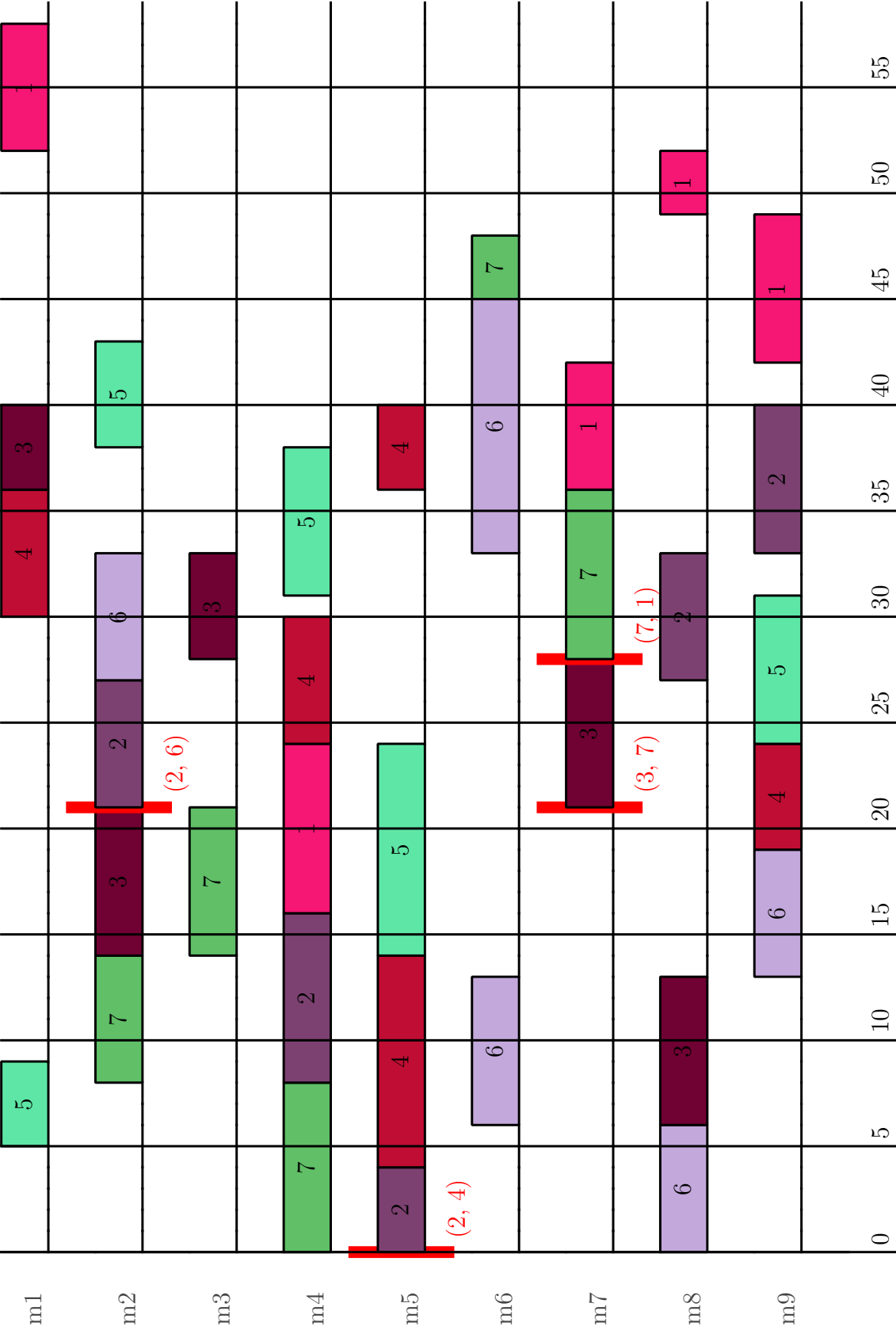
j	$r_j$	$d_j$	Marszruta technologiczna
1	10	55	m4 (8) - m7 (6) - m9 (7) - m8 (3) - m1 (6)
2	0	45	m5 (4) - m4 (8) - m2 (6) - m8 (6) - m9 (7)
3	5	50	m8 (7) - m2 (7) - m7 (7) - m3 (5) - m1 (4)
4	0	40	m5 (10) - m9 (5) - m4 (6) - m1 (6) - m5 (4)
5	5	45	m1 (4) - m5 (10) - m9 (7) - m4 (7) - m2 (5)
6	0	40	m8 (6) - m6 (7) - m9 (6) - m2 (6) - m6 (12)
7	0	45	m4 (8) - m2 (6) - m3 (7) - m7 (8) - m6 (3)

Tablica 10: Parametry wykonanych zadań

j	$c_j$	$f_j$	$l_j$	$e_j$
1	58	48	3	0
2	40	40	0	5
3	40	35	0	10
4	40	40	0	0
5	43	38	0	2
6	45	45	5	0
7	48	48	3	0

$$C_{max} = 58 \quad T_{max} = 5 \quad \sqrt{\sum e_j^2 + \sum l_j^2} = 13.1149$$

$$\bar{F} = 42 \quad \bar{T} = 1.57143 \quad \alpha * \sum e_j + \beta * \sum l_j \Big|_{\substack{\alpha=0.2 \\ \beta=0.8}} = 12.2$$



Rysunek 10: Wykres Gantt'a

## 5.2.2 Heurystyka LIFO

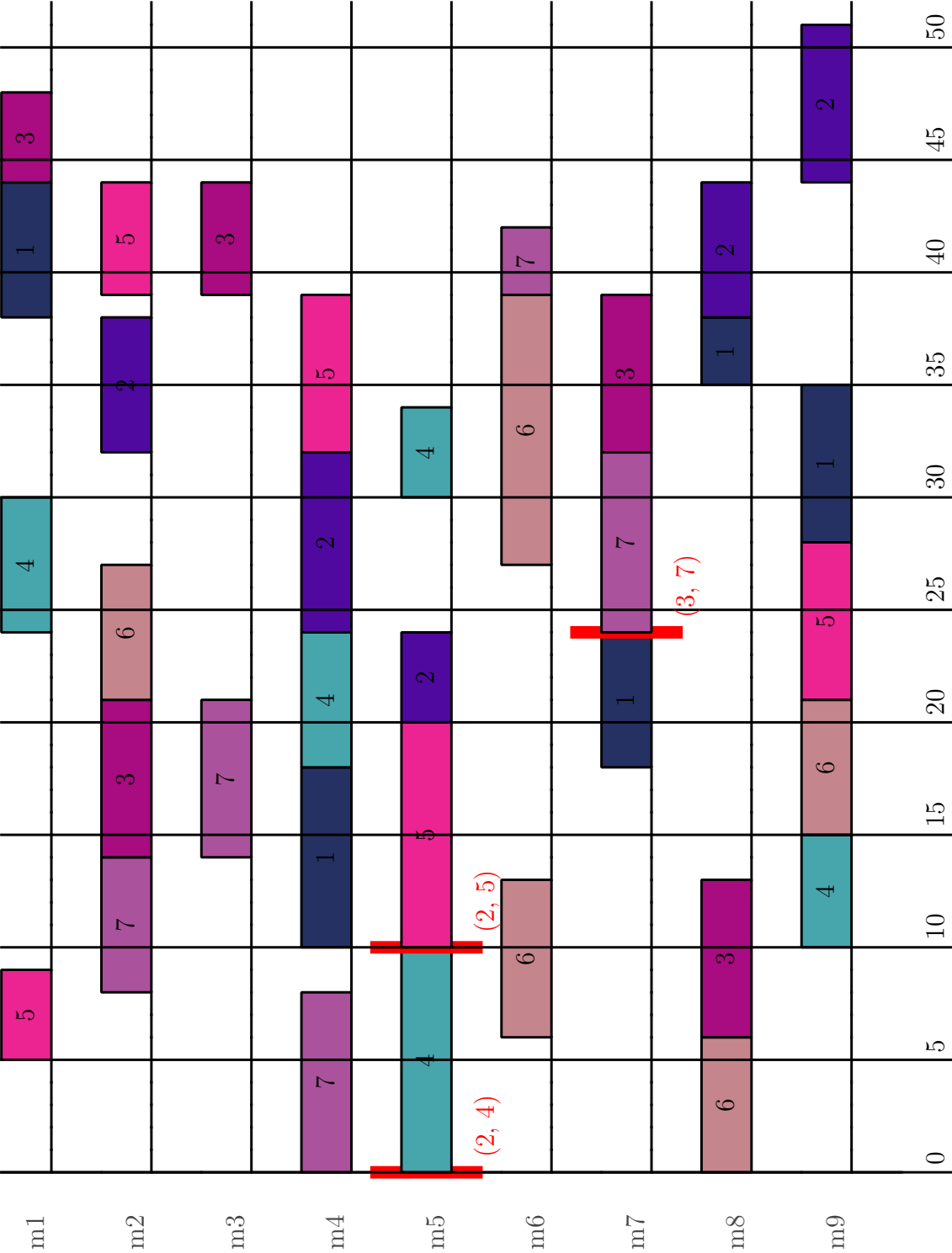
Tablica 11: Struktura zlecenia

j	$r_j$	$d_j$	Marszruta technologiczna
1	10	55	m4 (8) - m7 (6) - m9 (7) - m8 (3) - m1 (6)
2	0	45	m5 (4) - m4 (8) - m2 (6) - m8 (6) - m9 (7)
3	5	50	m8 (7) - m2 (7) - m7 (7) - m3 (5) - m1 (4)
4	0	40	m5 (10) - m9 (5) - m4 (6) - m1 (6) - m5 (4)
5	5	45	m1 (4) - m5 (10) - m9 (7) - m4 (7) - m2 (5)
6	0	40	m8 (6) - m6 (7) - m9 (6) - m2 (6) - m6 (12)
7	0	45	m4 (8) - m2 (6) - m3 (7) - m7 (8) - m6 (3)

Tablica 12: Parametry wykonanych zadań

j	$c_j$	$f_j$	$l_j$	$e_j$
1	44	34	0	11
2	51	51	6	0
3	48	43	0	2
4	34	34	0	6
5	44	39	0	1
6	39	39	0	1
7	42	42	0	3

$$\begin{aligned}
C_{max} &= 51 & T_{max} &= 6 & \sqrt{\sum e_j^2 + \sum l_j^2} &= 14.4222 \\
\bar{F} &= 40.2857 & \bar{T} &= 0.857143 & \alpha * \sum e_j + \beta * \sum l_j \Big|_{\substack{\alpha=0.2 \\ \beta=0.8}} &= 9.6
\end{aligned}$$



Rysunek 11: Wykres Gantt'a

### 5.3 Zmodyfikowane zlecenie bez zmian w marszrucie

#### 5.3.1 Heurystyka FIFO

Tablica 13: Struktura zlecenia

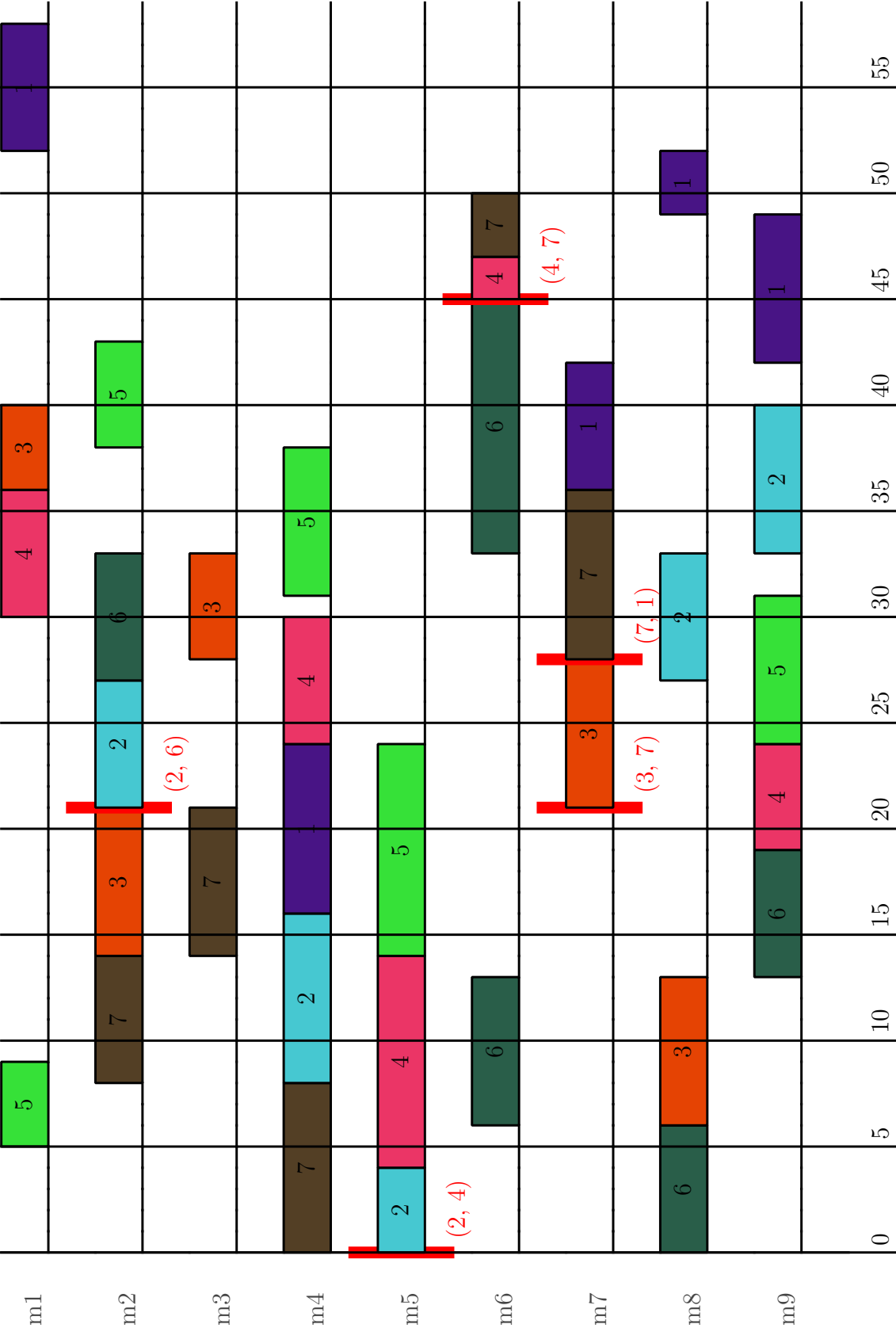
j	$r_j$	$d_j$	Marszruta technologiczna
1	10	55	m4 (8) - m7 (6) - m9 (7) - m8 (3) - m1 (6)
2	0	45	m5 (4) - m4 (8) - m2 (6) - m8 (6) - m9 (7)
3	5	50	m8 (7) - m2 (7) - m7 (7) - m3 (5) - m1 (4)
4	0	40	m5 (10) - m9 (5) - m4 (6) - m1 (6) - m6 (2)
5	5	45	m1 (4) - m5 (10) - m9 (7) - m4 (7) - m2 (5)
6	0	40	m8 (6) - m6 (7) - m9 (6) - m2 (6) - m6 (12)
7	0	45	m4 (8) - m2 (6) - m3 (7) - m7 (8) - m6 (3)

Tablica 14: Parametry wykonanych zadań

j	$c_j$	$f_j$	$l_j$	$e_j$
1	58	48	3	0
2	40	40	0	5
3	40	35	0	10
4	47	47	7	0
5	43	38	0	2
6	45	45	5	0
7	50	50	5	0

$$C_{max} = 58 \quad T_{max} = 7 \quad \sqrt{\sum e_j^2 + \sum l_j^2} = 15.3948$$

$$\bar{F} = 43.2857 \quad \bar{T} = 2.85714 \quad \alpha * \sum e_j + \beta * \sum l_j \Big|_{\substack{\alpha=0.2 \\ \beta=0.8}} = 19.4$$



Rysunek 12: Wykres Gantt'a

## 5.3.2 Heurystyka LIFO

Tablica 15: Struktura zlecenia

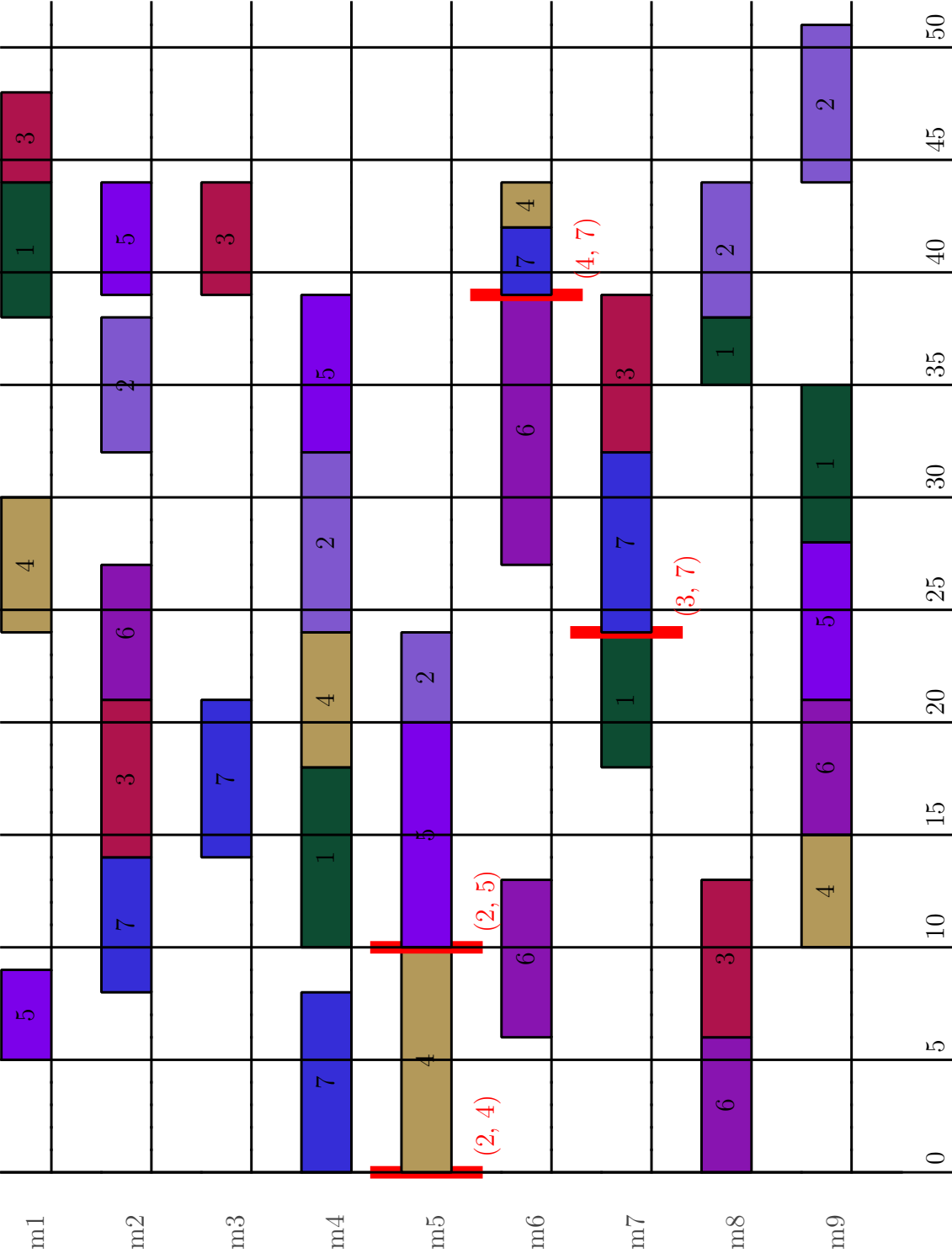
j	$r_j$	$d_j$	Marszruta technologiczna
1	10	55	m4 (8) - m7 (6) - m9 (7) - m8 (3) - m1 (6)
2	0	45	m5 (4) - m4 (8) - m2 (6) - m8 (6) - m9 (7)
3	5	50	m8 (7) - m2 (7) - m7 (7) - m3 (5) - m1 (4)
4	0	40	m5 (10) - m9 (5) - m4 (6) - m1 (6) - m6 (2)
5	5	45	m1 (4) - m5 (10) - m9 (7) - m4 (7) - m2 (5)
6	0	40	m8 (6) - m6 (7) - m9 (6) - m2 (6) - m6 (12)
7	0	45	m4 (8) - m2 (6) - m3 (7) - m7 (8) - m6 (3)

Tablica 16: Parametry wykonanych zadań

j	$c_j$	$f_j$	$l_j$	$e_j$
1	44	34	0	11
2	51	51	6	0
3	48	43	0	2
4	44	44	4	0
5	44	39	0	1
6	39	39	0	1
7	42	42	0	3

$$C_{max} = 51 \quad T_{max} = 6 \quad \sqrt{\sum e_j^2 + \sum l_j^2} = 13.7113$$

$$\bar{F} = 41.7143 \quad \bar{T} = 1.42857 \quad \alpha * \sum e_j + \beta * \sum l_j \Big|_{\substack{\alpha=0.2 \\ \beta=0.8}} = 11.6$$



Rysunek 13: Wykres Gantt'a



## 5.4 Zmodyfikowane zlecenie z naciskiem na $e_j$

### 5.4.1 Heurystyka FIFO

Tablica 17: Struktura zlecenia

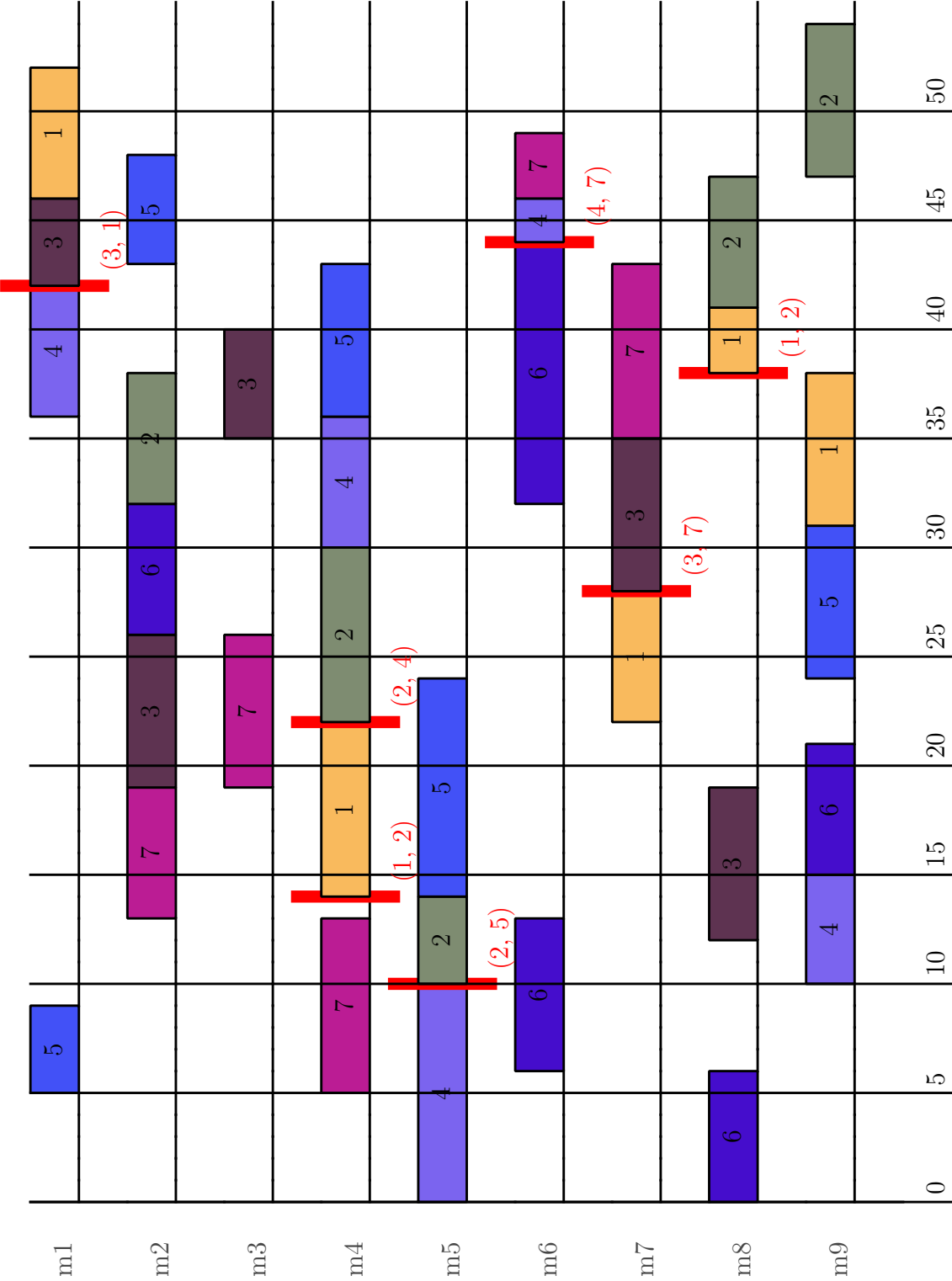
j	$r_j$	$d_j$	Marszruta technologiczna
1	14	55	m4 (8) - m7 (6) - m9 (7) - m8 (3) - m1 (6)
2	3	45	m5 (4) - m4 (8) - m2 (6) - m8 (6) - m9 (7)
3	12	50	m8 (7) - m2 (7) - m7 (7) - m3 (5) - m1 (4)
4	0	40	m5 (10) - m9 (5) - m4 (6) - m1 (6) - m6 (2)
5	5	45	m1 (4) - m5 (10) - m9 (7) - m4 (7) - m2 (5)
6	0	40	m8 (6) - m6 (7) - m9 (6) - m2 (6) - m6 (12)
7	5	45	m4 (8) - m2 (6) - m3 (7) - m7 (8) - m6 (3)

Tablica 18: Parametry wykonanych zadań

j	$c_j$	$f_j$	$l_j$	$e_j$
1	52	38	0	3
2	54	51	9	0
3	46	34	0	4
4	46	46	6	0
5	48	43	3	0
6	44	44	4	0
7	49	44	4	0

$$C_{max} = 54 \quad T_{max} = 9 \quad \sqrt{\sum e_j^2 + \sum l_j^2} = 13.5277$$

$$\bar{F} = 42.8571 \quad \bar{T} = 3.71429 \quad \alpha * \sum e_j + \beta * \sum l_j \Big|_{\substack{\alpha=0.2 \\ \beta=0.8}} = 22.2$$



Rysunek 14: Wykres Gantt'a

## 5.4.2 Heurystyka LIFO

Tablica 19: Struktura zlecenia

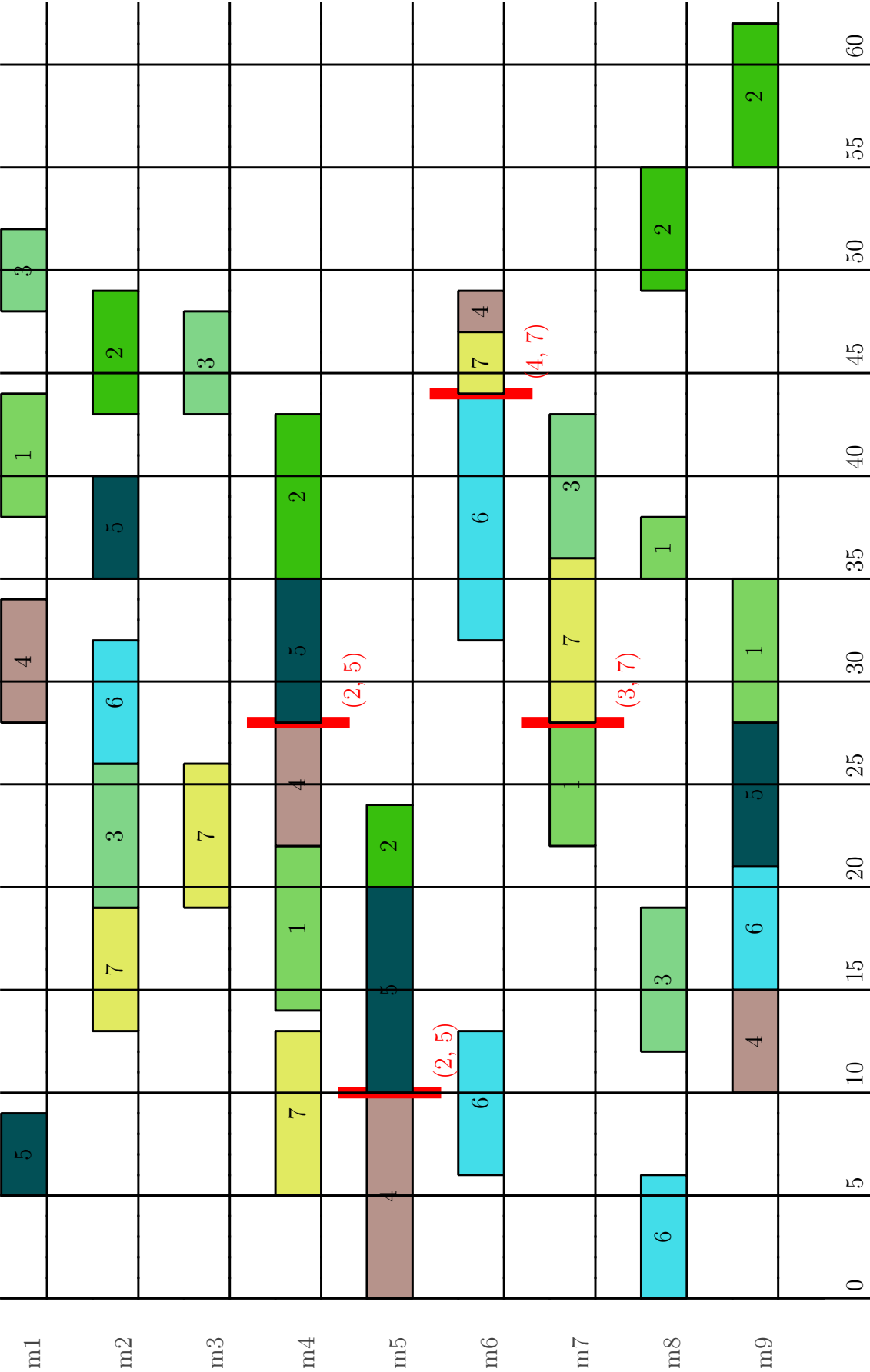
j	$r_j$	$d_j$	Marszruta technologiczna
1	14	55	m4 (8) - m7 (6) - m9 (7) - m8 (3) - m1 (6)
2	3	45	m5 (4) - m4 (8) - m2 (6) - m8 (6) - m9 (7)
3	12	50	m8 (7) - m2 (7) - m7 (7) - m3 (5) - m1 (4)
4	0	40	m5 (10) - m9 (5) - m4 (6) - m1 (6) - m6 (2)
5	5	45	m1 (4) - m5 (10) - m9 (7) - m4 (7) - m2 (5)
6	0	40	m8 (6) - m6 (7) - m9 (6) - m2 (6) - m6 (12)
7	5	45	m4 (8) - m2 (6) - m3 (7) - m7 (8) - m6 (3)

Tablica 20: Parametry wykonanych zadań

j	$c_j$	$f_j$	$l_j$	$e_j$
1	44	30	0	11
2	62	59	17	0
3	52	40	2	0
4	49	49	9	0
5	40	35	0	5
6	44	44	4	0
7	47	42	2	0

$$C_{max} = 62 \quad T_{max} = 17 \quad \sqrt{\sum e_j^2 + \sum l_j^2} = 23.2379$$

$$\bar{F} = 42.7143 \quad \bar{T} = 4.85714 \quad \alpha * \sum e_j + \beta * \sum l_j \Big|_{\substack{\alpha=0.2 \\ \beta=0.8}} = 30.4$$



Rysunek 15: Wykres Gantt'a

## 6 Wnioski

W sekcji 5.1 przedstawiono przykład zlecenia w systemie produkcyjnym o strukturze gniazdowej, który został rozwiązany przez program kSzereg z użyciem heurystyki FIFO oraz LIFO. Oba uzyskane uszeregowania są dopuszczalne i niezależnie od przyjętego kryterium oceny nieznacznie różnią się od siebie. Stanowią one punkt odniesienia dla kolejnych trzech zleceń, które powstały poprzez modyfikacje tego zlecenia w myśl strategii *Just in Time*, a więc minimalizacji zarówno spóźnienia zadań jak i czasu magazynowania gotowych produktów przed terminem ich dostarczenia.

Tablica 21: Porównanie wyznaczników jakości uszeregowania

sekcja	$C_{max}$	$\bar{F}$	$T_{max}$	$\bar{T}$	$\sqrt{\sum e_j^2 + \sum l_j^2}$	$\alpha * \sum e_j + \beta * \sum l_j \Big _{\substack{\alpha=0.2 \\ \beta=0.8}}$
5.1.1	55	42,4286	10	3,28571	19,7999	22,6
5.1.2	54	44,4286	9	3,85714	17,5499	23,8
5.2.1	58	42	5	1,57143	13,1149	12,2
5.2.2	51	40,2857	6	0,857143	14,4222	9,6
5.3.1	58	43,2857	7	2,85714	15,3948	19,4
5.3.2	51	41,7143	6	1,42857	13,7113	11,6
5.4.1	54	42,8571	9	3,71429	13,5277	22,2
5.4.2	62	42,7143	17	4,85714	23,2379	30,4

W sekcji 5.2 zmodyfikowano wyjściowy problem tak, aby rozwiązanie z użyciem heurystyki LIFO pozwoliło na zmniejszenie spóźnień zadań, a więc i wyznacznika (2), który jest w tym przypadku czterokrotnie bardziej wrażliwy na spóźnienia niż wczesność ( $\alpha = 0.2$ ,  $\beta = 0.8$ ). Było to możliwe dzięki przesunięciu chwili rozpoczęcia zadania nr 7 do początku czasu zlecenia oraz zmianie ostatniej operacji w marszrucie zadania nr 4, z 2 jednostek czasu na maszynie m6, na 4 jednostki czasu na maszynie m5. To wydłużenie czasu wykonywania operacji wynikło z założenia, że maszyna m5 najprawdopodobniej jest mniej odpowiednia do wykonania tej operacji od maszyny m6, jednak zamiana jest w ogóle możliwa. Te dwie zmiany spowodowały znaczną poprawę wszystkich rozpatrywanych wyznaczników jakości uszeregowania, ponieważ tylko jedno zadanie pozostało spóźnione, szczególnie funkcji (2). Pod względem tego właśnie współczynnika, jak i  $C_{max}$ ,  $\bar{F}$ , oraz ( $\bar{T}$ ) jest to najlepsze spośród prezentowanych rozwiązań.

Niestety założenie możliwości zmiany nawet jednego punktu marszruty technologicznej, nawet z dwukrotnym wydłużeniem czasu na wykonanie tej operacji, jest bardzo optymistyczne. Może się okazać, że wprowadzenie takich zmian wymaga dodatkowych nakładów finansowych, np. na dodatkowe narzędzia, czy szkolenie pracownika, lub jest wręcz niemożliwe bez przebudowy całego gniazda. Okazuje się jednak, że nawet nie zmieniając żadnej marszruty technologicznej, jedynie poprzez przyspieszenie realizacji zadania nr 7 jest możliwe uzyskanie niewiele gorszych wyników. Jak pokazano w sekcji 5.3.2 jedynie średnie spóźnienie zadań znacząco się zwiększyło, gdyż oprócz zadania 2 również zadanie 4 jest spóźnione, jednak wpłynęło to na poprawę wyznacznika (1), ponieważ jest gotowe bliżej terminu.

Nie zawsze istnieje możliwość wcześniejszego rozpoczęcia przetwarzania zadania, jednak nawet wtedy można zastosować strategię *Just in Time*, gdyż prawie zawsze można zdecydować o późniejszym rozpoczęciu przetwarzania zadania. Jak dowodzi uszeregowanie z sekcji 5.4.1 takie przesunięcie w czasie rozpoczęcia trzech zadań o kilka jednostek czasowych może doprowadzić do skrócenia czasu magazynowania gotowych produktów nie wydłużając czasu przetwarzania całego zlecenia. W tym przypadku największej poprawie uległ wyznacznik (1), ponieważ nie przykłada on większej wagi do spóźnień, jak wyznacznik (2), który poprawił się jedynie nieznacznie.

Podsumowując, strategia *Just in Time* w systemach produkcyjnych o strukturze gniazdowej pozwala na duże oszczędności poprzez obniżenie kosztów zbyt wczesnego wyprodukowania towaru, jak magazynowanie, czy ubezpieczenie, oraz spóźnionej produkcji, co może się wiązać między innymi z karami umownymi. Jak przedstawiono na przykładzie, wprowadzenie takiej strategii może być przeprowadzone kompleksowo - poprzez wymuszenie wcześniejszego ukończenia zadań na innych stanowiskach i zmiany w samej strukturze gniazda, aby umożliwić zmiany w marszrucie technologicznej - lub poprzez mniejsze przesunięcia w grafiku przetwarzania zadań. Zawsze warto więc przeanalizować rozpatrywany problem szeregowania zadań pod kątem strategii *Just in Time*.

## 7 Literatura

- [1] dr Zbigniew Pastuszak, *Zarządzanie logistyczne Podstawowe definicje*, Źródło: [umcs.net.pl](http://umcs.net.pl) [dostęp: 8.01.2013]
- [2] dr inż. Waldemar Grzechca na podstawie materiałów prof. dr hab. inż. Mirosława Zaborowskiego, *Wykłady z przedmiotu Zautomatyzowane Systemy Wytwarzania*, Źródło: <http://platforma.polsl.pl/rau1> [dostęp: 20.12.12]
- [3] Linia produkcyjna  
Źródło: [http://pl.wikipedia.org/wiki/Linia\\_produkcyjna](http://pl.wikipedia.org/wiki/Linia_produkcyjna) [dostęp: 20.12.12]
- [4] Fatima El Khoukhi, Tarik Lamoudan, Jaouad Boukachour, Ahmed El Hilali Alaoui, *Ant Colony Algorithm for Just-in-Time Job Shop Scheduling with Transportation Times and Multirobots*, ISRN Applied Mathematics, Article ID 165620, 2011,
- [5] Rodolfo Pereira Araujo, André Gustavo dos Santos, José Elias Cláudio Arroyo, *Genetic Algorithm and Local Search for Just-in-Time Job-Shop Scheduling*, 2009 IEEE Congress on Evolutionary Computation (CEC 2009)
- [6] M.R. Garey and D.S. Johnson, *Computers and intractability: A guide to the theory of NP-complete-ness*, Freeman, New York, 1979.
- [7] E.L. Lawrence, J.K. Lenstra, A.H.G. Rinnooy, *Recent developments in deterministic sequencing and scheduling: A survey, Deterministic and stochastic scheduling*, Riedel, Dordrecht, 1982.
- [8] Algorithms and Theory of Computation Handbook, CRC Press LLC, 1999, "strongly NP-hard", w Dictionary of Algorithms and Data Structures [online], Paul E. Black, ed., U.S. National Institute of Standards and Technology. 17.12.2004. Dostępny pod adresem <http://www.nist.gov/dads/HTML/stronglyNP.html> [dostęp: 7.01.2013]

- [9] Ramesh Bollapragada, Norman M. Sadeh *Proactive Release Procedures for Just-in-Time Job Shop Environments, Subject to Machine Failures* ©2004 Wiley Periodicals, Inc. Naval Research Logistics 51: 1018 –1044, 2004.
- [10] M. G. C. Resende, *Shop Floor Scheduling of Semiconductor Wafer Manufacturing*, PhD thesis Department of Industrial Engineering and Operations Research, University of California, Berkeley, 1987
- [11] Philippe Baptiste, Marta Flamini, Francis Sourd, *Lagrangian Bounds for Just-In-Time Job-Shop Scheduling*, Computers & Operations Research, doi: 10.1016/j.cor.2006.05.009.