

Algorytmy optymalizacji dyskretnej

Sprawozdanie z listy 2

Marek Świergoń (261750)

1 kwietnia 2023, PWr, WiIT INA

1 Zadanie 1

1.1 Treść zadania (autor: dr inż. Karol Gotfryd)

Pewne przedsiębiorstwo lotnicze musi podjąć decyzję o zakupie paliwa do samolotów odrzutowych, mając do wyboru trzech dostawców. Samoloty tankują paliwo regularnie na czterech lotniskach, które obsługują.

Firmy paliwowe poinformowały, że mogą dostarczyć następujące ilości paliwa w nadchodzącym miesiącu: Firma 1 – 275 000 galonów, Firma 2 – 550 000 galonów i Firma 3 – 660 000 galonów. Niezbędne ilości paliwa do odrzutowców na poszczególnych lotniskach są odpowiednio równe: na lotnisku 1 – 110 000 galonów, na lotnisku 2 – 220 000 galonów, na lotnisku 3 – 330 000 galonów i na lotnisku 4 – 440 000 galonów.

Koszt jednego galonu paliwa (w \$) z uwzględnieniem kosztów transportu dostarczonego przez poszczególnych dostawców kształtuje się na każdym z lotnisk następująco:

	Firma 1	Firma 2	Firma 2
Lotnisko 1	10	7	8
Lotnisko 2	10	11	14
Lotnisko 3	9	12	4
Lotnisko 4	11	13	9

Wyznacz plan zakupu i dostaw paliwa na lotniska, który minimalizuje koszty. Następnie na jego podstawie odpowiedz na poniższe pytania.

- (a) Jaki jest minimalny łączny koszt dostaw wymaganych ilości paliwa na wszystkie lotniska?
- (b) Czy wszystkie firmy dostarczają paliwo?
- (c) Czy możliwości dostaw paliwa przez firmy są wyczerpane?

1.2 Opis modelu

Niech

- $N_s = \{1, 2, \dots, n\}$ – zbiór firm paliwowych (dostawców),
- $N_a = \{1, 2, \dots, m\}$ – zbiór lotnisk (odbiorców),
- s_i – maksymalna liczba galonów paliwa, jaką może dostarczyć lotniskom i -ta firma,

- d_j – zapotrzebowanie j-tego lotniska na paliwo,
- c_{ij} – koszt zakupu jednego galonu paliwa od i-tej firmy przez j-te lotnisko.

Graf $G = (N, A)$, $N = N_s \cup N_a$, $A = N_s \times N_a$ jest grafem rozwiązywanego problemu, będącego instancją problemu transportowego (brak węzłów pośredniczących, sami dostawcy i odbiorcy).

Wprowadzamy macierz zmiennych decyzyjnych $[x_{ij}]$ – ile paliwa będzie dostarczone z firmy i-tej do j-tego lotniska.

Funkcją celu, którą będziemy minimalizować, jest łączny koszt wszystkich dostaw:

$$\min c(x), \quad c(x) = \sum_{(i,j) \in A} c_{ij} \cdot x_{ij}.$$

Ograniczenia modelu:

1. (firma nie może dostarczać więcej paliwa niż wynika z jej podaży)

$$\sum_{j:(i,j) \in A} x_{ij} \leq s_i, \quad i \in N_s,$$

2. (zapotrzebowanie lotnisk na paliwo musi być zaspokojone)

$$\sum_{i:(i,j) \in A} x_{ij} = d_j, \quad j \in N_a,$$

3. (nie wysyłamy „ujemnej ilości paliwa”)

$$x_{ij} \geq 0.$$

W zadaniu każda firma ma możliwość dostawy do każdego z lotnisk (czyli w modelu zbiór krawędzi $A = N_s \times N_a$); gdyby tak nie było, to trzeba by wymusić $x_{ij} = 0$, jeśli firma i-ta nie dostarcza paliwa do j-tego lotniska. Taką implikację w teoretycznym modelu należałoby przerobić na zbiór ograniczeń liniowych, ale w samej implementacji w JuMP nie jest to konieczne, ponieważ łatwo można wymusić wartość na zmiennej metodą `fix(zmienna, wartość; force=true)`.

1.3 Rozwiązanie: wyniki i ich interpretacja

Powyższy model został zapisany w języku Julia zgodnie z dokumentacją modułu JuMP. Kod źródłowy znajduje się w pliku `zadanie1.jl`. Dane wczytywane są z pliku tekstowego, a do rozwiązania modelu zastosowany został solver HiGHS. Plik tekstowy z danymi z treści zadania zobrazowany został poniżej.

```
.;L1;L2;L3;L4;SUPPLY
F1;10;10;9;11;275000
F2;7;11;12;13;550000
F3;8;14;4;9;660000
DEMAND;110000;220000;330000;440000
```

Rysunek 1: Plik tekstowy z danymi z treści zadania.

Wyniki dla powyższych danych (macierz dostaw) przedstawia poniższa tabela.

	Lotnisko 1	Lotnisko 2	Lotnisko 3	Lotnisko 4	RAZEM
Firma 1	–	165 000	–	110 000	275 000
Firma 2	110 000	55 000	–	–	165 000
Firma 3	–	–	330 000	330 000	660 000
RAZEM	110 000	220 000	330 000	440 000	

Tabela 1: Optymalna macierz dostaw (w galonach) dla danych z treści zadania.

Z Tabeli 1 i danych z treści zadania wynika, że:

- (a) Minimalny łączny koszt dostaw wymaganych ilości paliwa na wszystkie lotniska wynosi 8 525 000.
- (b) Każda z firm dostarcza paliwo.
- (c) Tylko firmy 1 i 3 mają wyczerpane możliwości dostaw.

2 Zadanie 2

2.1 Treść zadania (autor: dr inż. Karol Gotfryd)

Dana jest sieć połączeń między n miastami reprezentowana za pomocą skierowanego grafu $G = (N, A)$, gdzie N jest zbiorem miast (wierzchołków), $|N| = n$, A jest zbiorem połączeń między miastami (łuków), $|A| = m$. Dla każdego połączenia z miasta i do miasta j , $(i, j) \in A$, dane są koszt przejazdu c_{ij} oraz czas przejazdu t_{ij} (im mniejszy koszt, tym dłuższy czas przejazdu). Dane są również dwa miasta $i^\circ, j^\circ \in N$.

1

Celem jest znalezienie połączenia (ścieżki) między zadanymi dwoma miastami, którego całkowity koszt jest najmniejszy i całkowity czas przejazdu nie przekracza z góry zadanego czasu przejazdu T .

- (a) Zapisz model programowania całkowitoliczbowego w wybranym języku. Rozwiąż własny egzemplarz problemu ($n \geq 10$) za pomocą solvera GLPK (lub np. Cbc).
- (b) Czy ograniczenie na całkowitoliczbowość zmiennych decyzyjnych jest potrzebne? Sprawdź, jakie będą wartości zmiennych decyzyjnych, jeśli usuniemy ograniczenie na ich całkowitoliczbowość (tj. mamy przypadek, w którym model jest modelem programowania liniowego).
- (c) Czy po usunięciu ograniczenia na czasy przejazdu w modelu bez ograniczeń na całkowitoliczbowość zmiennych decyzyjnych i rozwiązaniu problemu otrzymane połączenie jest akceptowalnym rozwiązaniem?

2.2 Opis modelu

Niech

- $N = \{1, 2, \dots, n\}$ – zbiór miast,
- A – zbiór połączeń między miastami (graf $G = (N, A)$ nie musi być kliką),
- $c_{ij}, (i, j) \in A$ – koszt przejazdu z miasta i -tego do miasta j -tego (zakładamy, że $c_{ij} > 0$),
- $t_{ij}, (i, j) \in A$ – czas przejazdu z miasta i -tego do miasta j -tego (zakładamy, że $t_{ij} > 0$).

Chcemy wyznaczyć ścieżkę pomiędzy miastami $s \in N$ i $t \in N$, której całkowity koszt będzie najmniejszy a całkowity czas nie przekracza zadanej wartości T . Jest to instancja problemu najkrótszej ścieżki z ograniczeniem (Shortest Weight-Constrained Path), gdzie minimalizujemy koszt, a czas przejazdu zalicza się do zbioru ograniczeń.

Każdej krawędzi $(i, j) \in A$ przypisujemy zmienną decyzyjną x_{ij} , określającą to, czy dana krawędź znajduje się w optymalnej ścieżce (0 – nie, 1 – tak). Funkcją celu, którą będziemy minimalizować, jest sumaryczny koszt trasy, czyli

$$\min c(x), \quad c(x) = \sum_{(i,j) \in A} c_{ij} \cdot x_{ij}$$

Ograniczenia dla modelu:

1. (poprawność ścieżki z s do t)

$$\sum_{j:(i,j) \in A} x_{ij} - \sum_{j:(j,i) \in A} x_{ji} = \begin{cases} 1, & \text{gdy } i = s \\ -1, & \text{gdy } i = t \\ 0, & \text{w p.p.} \end{cases}$$

2. (ograniczenie na łączny czas ścieżki)

$$\sum_{(i,j) \in A} t_{ij} \cdot x_{ij} \leq T$$

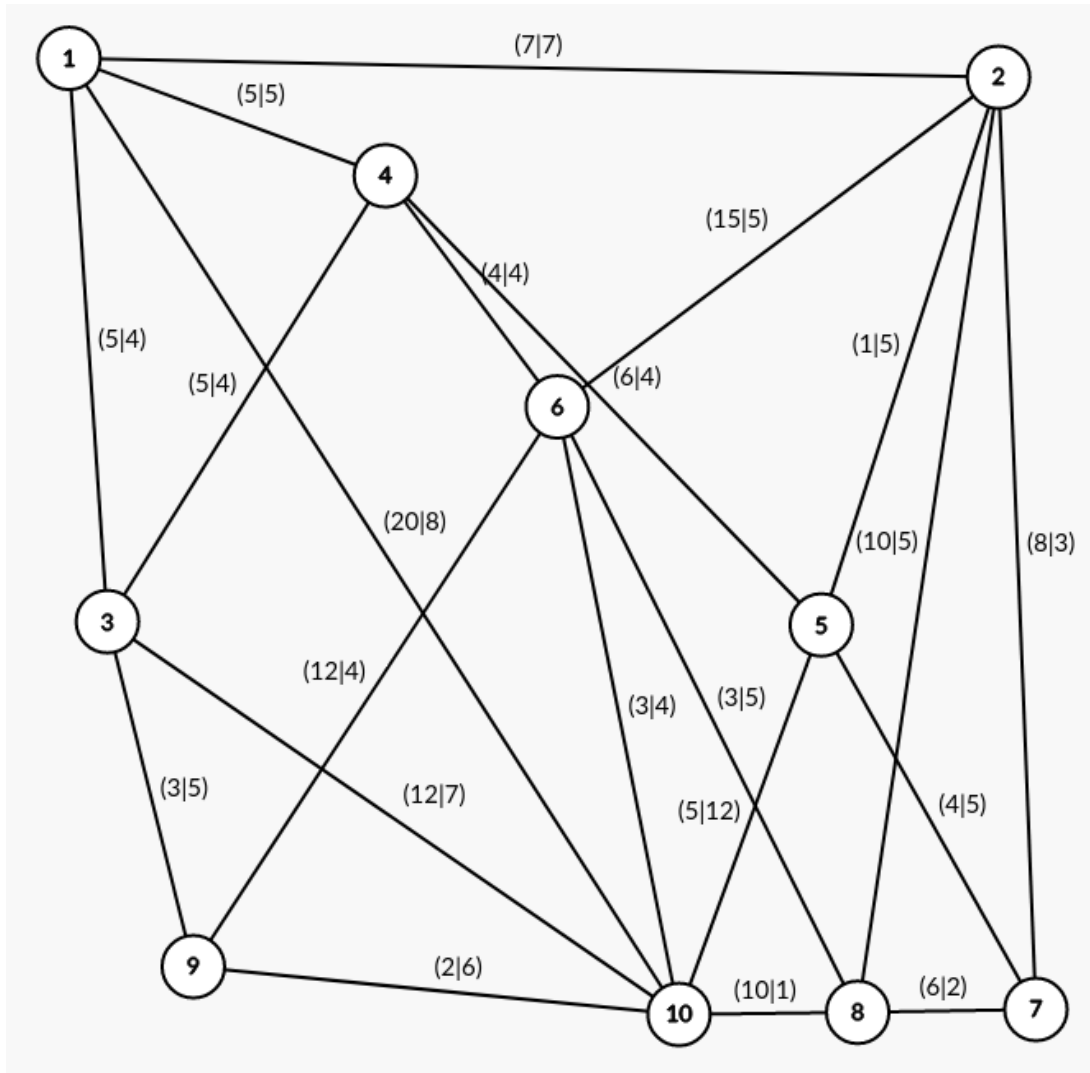
3. (wymuszenie całkowitoliczbowości)

$$x_{ij} \in \{0, 1\}.$$

2.3 Rozwiązanie: wyniki i ich interpretacja

Powyższy model został zapisany w języku Julia zgodnie z dokumentacją modułu JuMP. Kod źródłowy znajduje się w pliku `zadanie2.jl`. Dane wczytywane są z dwóch plików tekstowych: w jednym pliku znajdują się koszty krawędzi, a w drugim czasy. Zapis danych jest w formacie macierzy $n \times n$; gdy jakaś krawędź nie istnieje, to uznajemy, że ma ona czas i koszt równy zero. W związku z tym tworzona jest również analogiczna macierz zmiennych decyzyjnych $[x_{ij}]$, przy czym dodatkowymi warunkami wymuszone zostaje to, że $x_{ij} = 0$, gdy krawędź $(i, j) \notin A$.

Przy rozwiązywaniu modelu zastosowany został solver HiGHS. Do celów testowych powstała instancja problemu, której szczegóły zostały podane na poniższych rysunkach.



Rysunek 2: Graf własnego egzemplarza problemu. Format znaczników krawędzi: $(c_{ij} \mid t_{ij})$

0;7;5;5;0;0;0;0;0;20	0;7;4;5;0;0;0;0;0;8
7;0;0;0;1;15;8;10;0;0	7;0;0;0;5;5;3;5;0;0
5;0;0;5;0;0;0;0;3;12	4;0;0;4;0;0;0;0;5;7
5;0;5;0;6;4;0;0;0;0	5;0;4;0;4;4;0;0;0;0
0;1;0;6;0;0;4;0;0;5	0;5;0;4;0;0;5;0;0;12
0;15;0;4;0;0;0;3;12;3	0;5;0;4;0;0;0;5;4;4
0;8;0;0;4;0;0;6;0;0	0;3;0;0;5;0;0;2;0;0
0;10;0;0;0;3;6;0;0;10	0;5;0;0;0;5;2;0;0;1
0;0;3;0;0;12;0;0;0;2	0;0;5;0;0;4;0;0;0;6
20;0;12;0;5;3;0;10;2;0	8;0;7;0;12;4;0;1;6;0

Rysunek 3: Pliki tekstowe zawierające odpowiednio koszty (po lewej) i czasy (po prawej) połączeń między miastami dla własnego egzemplarza problemu.

Poniższy rysunek przedstawia najkrótszą ścieżkę z $s = 9$ do $t = 2$ dla dwóch różnych czasów maksymalnych.

```

Koszt najtańszej ścieżki 9->...->2 o jej czasie wynoszącym maksymalnie 23: 8.0
Czas ścieżki: 23.0
Przebieg ścieżki:
9->10->5->2
*****
Koszt najtańszej ścieżki 9->...->2 o jej czasie wynoszącym maksymalnie 22: 15.0
Czas ścieżki: 16.0
Przebieg ścieżki:
9->3->1->2

```

Rysunek 4: Wynik działania programu dla własnego egzemplarza problemu, $s = 9$, $t = 2$ i maksymalnych czasów $T \in \{22, 23\}$.

Usunięcie ograniczenia na całkowitoliczbowość zmiennych decyzyjnych niestety może powodować niecałkowite wartości zmiennych. Poniższy rysunek pokazuje wartości zmiennych decyzyjnych dla tych samych danych co w Rysunku 4.

```

Koszt najtańszej ścieżki 9->...->2 o jej czasie wynoszącym maksymalnie 23: 8.0
Czas ścieżki: 23.0
Macierz zmiennych decyzyjnych:
  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0
  0.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0

*****
Koszt najtańszej ścieżki 9->...->2 o jej czasie wynoszącym maksymalnie 22: 9.0
Czas ścieżki: 22.0
Macierz zmiennych decyzyjnych:
  0.0  0.143  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.143  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.0  0.857  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.0  0.0  0.143  0.0  0.0  0.0  0.0  0.0  0.0  0.857
  0.0  0.0  0.0  0.0  0.857  0.0  0.0  0.0  0.0  0.0

```

Rysunek 5: Wynik działania programu po usunięciu ograniczenia na całkowitoliczbowość dla własnego egzemplarza problemu, $s = 9$, $t = 2$ i maksymalnych czasów $T \in \{22, 23\}$.

Widzimy, że w przypadku gdy dozwolona ścieżka o minimalnym koszcie nie „wykorzystuje” całego dostępnego czasu, to solver bierze „ułamki” tańszych krawędzi. Dzięki temu zmniejsza koszt, w pełni wykorzystując dostępny czas. Oczywiście nie jest to zjawiskiem pożądanym, więc ograniczenie na całkowitoliczbowość jest tu konieczne. Można z tego również wywnioskować, że gdyby nie było ograniczenia na łączny czas połączenia, to solver rzeczywiście zwróciłby poprawne rozwiązanie.

3 Zadanie 3

3.1 Treść zadania (autor: dr inż. Karol Gotfryd)

Zapisz model dla zadania 3. z Listy 2 na ćwiczenia w wybranym języku i rozwiąż go dla podanych tam danych za pomocą solvera GLPK (lub np. Cbc).

W opisie rozwiązania przedstaw optymalny przydział radiowozów dla każdej zmiany i dzielnicy oraz podaj całkowitą liczbę wykorzystywanych radiowozów.

Treść zadania 3. z Listy 2 na ćwiczenia:

Zadanie 3. Policja w małym miasteczku posiada w swoim zasięgu trzy dzielnice oznaczone jako p_1 , p_2 i p_3 . Każda dzielnica ma przypisaną pewną liczbę radiowozów wyposażonych w radiotelefony i sprzęt pierwszej pomocy. Policja pracuje w systemie trzymianowym. W tabelach 1 i 2 podane są minimalne i maksymalne liczby radiowozów dla każdej zmiany.

Tabela 1: Minimalne liczby radiowozów dla każdej zmiany i dzielnicy

	zmiana 1	zmiana 2	zmiana 3
p_1	2	4	3
p_2	3	6	5
p_3	5	7	6

Tabela 2: Maksymalne liczby radiowozów dla każdej zmiany i dzielnicy

	zmiana 1	zmiana 2	zmiana 3
p_1	3	7	5
p_2	5	7	10
p_3	8	12	10

Aktualne przepisy wymuszają, że dla zmiany 1, 2 i 3 powinno być dostępnych, odpowiednio, co najmniej 10, 20 i 18 radiowozów. Ponadto dzielnice p_1 , p_2 i p_3 powinny mieć przypisane, odpowiednio, co najmniej 10, 14 i 13 radiowozów. Policja chce wyznaczyć przydział radiowozów spełniający powyższe wymagania i minimalizujący ich całkowitą liczbę. Sformułuj ten problem jako problem cyrkulacji.

3.2 Opis modelu

Do tego zadania można podejść dwojako: albo zakładamy, że radiowóz może odbyć tylko jedną zmianę w ciągu dnia, albo zakładamy, że radiowóz może uczestniczyć we wszystkich trzech zmianach. Pierwsze podejście ma sens np. gdybyśmy z radiowozem ściśle wiązali jeden zespół personelu go obsługujący, który danego dnia pracuje tylko na jednej zmianie. Drugie podejście traktuje radiowóz dosłownie jako pojazd, który jako taki może być raczej eksploatowany przez całą dobę.

Mamy dane:

- $S = \{1, 2, \dots, n\}$ – zbiór zmian,
- $P = \{1, 2, \dots, m\}$ – zbiór dzielnic,
- S_{MIN_i} – minimalna liczba radiowozów łącznie dla i -tej zmiany,

- P_{MIN_j} – minimalna liczba radiowozów łącznie na wszystkich zmianach dla j-tej dzielnicy,
- m_{ij} – minimalna liczba radiowozów dla i-tej zmiany i j-tej dzielnicy,
- M_{ij} – maksymalna liczba radiowozów dla i-tej zmiany i j-tej dzielnicy.

3.2.1 Model przy wykorzystaniu radiowozu na jednej zmianie

Określmy zmienne decyzyjne: x_{ij} – liczba radiowozów przypisanych na i-tej zmianie do j-tej dzielnicy. Funkcją celu będzie całkowita liczba radiowozów ze wszystkich zmian i dzielnic.

$$\min f(x), \quad f(x) = \sum_{i=1}^n \sum_{j=1}^m x_{ij}$$

Ograniczenia dla modelu:

1. (łączna liczba radiowozów na zmianie nie jest za mała)

$$\sum_{j=1}^m x_{ij} \geq S_{MIN_i}, \quad i = 1, \dots, n$$

2. (łączna liczba radiowozów przypisana do dzielnicy nie jest za mała)

$$\sum_{i=1}^n x_{ij} \geq P_{MIN_j}, \quad j = 1, \dots, m$$

3. (liczba radiowozów na danej zmianie i dzielnicy mieści się w zakresie)

$$m_{ij} \leq x_{ij} \leq M_{ij} \quad \text{i jest całkowita}$$

W ograniczeniu 3. w praktyce nie jest konieczna całkowitoliczbowość, gdyż ograniczenia podane ograniczenia na liczby radiowozów są liczbami całkowitymi, a minimalizując liczbę radiowozów będziemy właśnie zrównywać liczby radiowozów z tymi ograniczeniami.

3.2.2 Model przy wykorzystaniu radiowozu na wszystkich zmianach

Dla tego modelu istotnie różni się funkcja celu do zminimalizowania.

$$\min f(x), \quad f(x) = \max_{j \in \{1, \dots, m\}} \sum_{i=1}^n x_{ij}$$

Funkcja ta jest niepoprawna dla modeli programowania liniowego, musimy ją zapisać inaczej, wprowadzając dodatkowe, pomocnicze zmienne decyzyjne.

Tak jak w poprzednim modelu, wprowadzamy zmienne decyzyjne x_{ij} – liczba radiowozów przypisanych na i-tej zmianie do j-tej dzielnicy. Dodajemy pomocnicze zmienne: s_j – łączna liczba radiowozów na j-tej zmianie, t – największa łączna liczba radiowozów na zmianie. Aby zmienne pomocnicze rzeczywiście reprezentowały wspomniane wartości, musimy wprowadzić dodatkowe ograniczenia do modelu.

Ograniczenia dla modelu:

1. (łączna liczba radiowozów na zmianie nie jest za mała)

$$\sum_{j=1}^m x_{ij} \geq S_{MIN_i}, \quad i = 1, \dots, n$$

2. (łączna liczba radiowozów przypisana do dzielnic nie jest za mała)

$$\sum_{i=1}^n x_{ij} \geq P_{MIN_j}, \quad j = 1, \dots, m$$

3. (liczba radiowozów na danej zmianie i dzielnic mieści się w zakresie)

$$m_{ij} \leq x_{ij} \leq M_{ij} \quad \text{i jest całkowita}$$

4. (s_j pokazuje łączną liczbę radiowozów na zmianie)

$$s_j = \sum_{i=1}^n x_{ij}$$

5. (t musi być co najmniej równe najliczniejszej zmianie)

$$t \geq s_j, \quad j = 1, \dots, m$$

Dzięki takim ograniczeniom zmienne pomocnicze przyjmują wartości zgodne z ich definicjami. Funkcja celu do minimalizacji:

$$\min \tilde{f}(x), \quad \tilde{f}(x) = t$$

Ponieważ minimalizujemy t , to rzeczywiście będzie ono równe maksymalnej łącznej liczbie radiowozów spośród wszystkich ze zmian.

3.3 Rozwiązanie: wyniki i ich interpretacja

Powyższe modele zostały zapisane w języku Julia zgodnie z dokumentacją modułu JuMP. Kod źródłowy znajduje się w plikach `zadanie3.jl` (pierwszy model) i `zadanie3.minmax.jl` (drugi model). Dane wczytywane są z dwóch plików tekstowych: w jednym pliku znajdują się minimalne liczby radiowozów, a w drugim maksymalne. Poniższy rysunek obrazuje pliki tekstowe dla danych z treści zadania.

<code>.;S1;S2;S3;MIN_PRECINT</code>	<code>.;S1;S2;S3</code>
<code>P1;2;4;3;10</code>	<code>P1;3;7;5</code>
<code>P2;3;6;5;14</code>	<code>P2;5;7;10</code>
<code>P3;5;7;6;13</code>	<code>P3;8;12;10</code>
<code>MIN_SHIFT;10;20;18</code>	

Rysunek 6: Pliki tekstowe zawierające odpowiednio minimalne (po lewej) i maksymalne (po prawej) liczby radiowozów, zgodnie z treścią zadania.

Poniższy rysunek przedstawia rozwiązanie dla modelu z radiowozem będącym tylko na jednej zmianie i danym z treści zadania.

```
Minimalna całkowita liczba radiowozów: 48.0
Macierz radiowozów:
      S1      S2      S3
P1    2.0    7.0    5.0
P2    3.0    6.0    5.0
P3    5.0    7.0    8.0
```

Rysunek 7: Plik tekstowy z rozwiązaniem dla danych z treści zadania i modelu z radiowozami wykorzystywanymi w ciągu dnia tylko na jednej zmianie.

Rozwiązanie dla modelu zakładającego możliwość wykorzystania tego samego radiowozu na wszystkich trzech zmianach obrazuje Rysunek 8.

Minimalna całkowita liczba radiowozów: 20.0			
Macierz radiowozów:			
	S1	S2	S3
P1	3.0	4.0	3.0
P2	3.0	6.0	5.0
P3	5.0	10.0	10.0

Rysunek 8: Plik tekstowy z rozwiązaniem dla danych z treści zadania i modelu z radiowozami wykorzystywanymi w ciągu dnia tylko na jednej zmianie.

Rzeczywiście nie jest możliwe wykorzystanie mniej niż 20 radiowozów, ponieważ minimalna liczba radiowozów na zmianie 2. wynosi 20. Co ciekawe, rozwiązanie z pierwszego modelu jest również rozwiązaniem optymalnym dla drugiego modelu (również wykorzystujemy maksymalnie 20 radiowozów, gdy mogą one pracować przez całą dobę). Z kolei optymalne rozwiązanie podane przez solver dla drugiego modelu nie jest optymalne z perspektywy funkcji celu pierwszego modelu (suma liczby radiowozów po zmianach i dzielnicach wynosi 49 zamiast 48). Może to wynikać z faktu, że w drugim modelu nie przejmujemy się nadmiarowymi radiowozami dla zmian mniej licznych, gdyż nie wpłyną one na wartość funkcji celu (o której wprost decyduje najliczniejsza ze zmian).

4 Zadanie 4

4.1 Treść zadania (autor: dr inż. Karol Gotfryd)

Pewna firma przeładunkowa posiada teren, na którym składowane kontenery z cennym ładunkiem. Teren podzielony jest na $m \times n$ kwadratów. Kontenery składowane są w wybranych kwadratach. Zakłada się, że kwadrat może być zajmowany przez co najwyżej jeden kontener. Firma musi rozmieścić kamery, żeby monitorować kontenery. Każda kamera może obserwować k kwadratów na lewo, k kwadratów na prawo, k kwadratów w górę i k kwadratów w dół. Kamera nie może być umieszczona w kwadracie zajmowanym przez kontener.

Zaplanuj rozmieszczenie kamer w kwadratach tak, aby każdy kontener był monitorowany przez co najmniej jedną kamerę oraz liczba użytych kamer była jak najmniejsza.

Zapisz model programowania całkowitoliczbowego w wybranym języku. Rozwiąż własny egzemplarz problemu ($m, n \geq 5$; rozwiązania dla ≥ 2 różnych wartości parametru k) za pomocą solvera GLPK (lub np. Cbc).

4.2 Opis modelu

Podobnie jak w Zadaniu 3, treść zadania można zamodelować na różne sposoby. Wśród nich można wyróżnić dwa, różniące się działaniem kamer: albo kamery obserwują kwadrat o długości boku $2k + 1$ i środkiem będącym w polu kamery, albo kamery obserwują dosłownie k kwadratów w każdą ze stron, czyli mają pole widzenia w kształcie krzyża.

Dla obu rozważanych wersji dane są wspólne, jedyna różnica wystąpi w ograniczeniach dla modeli.

Mamy dane:

- m – szerokość terenu w zadanych jednostkach („kwadratach”)

- n – wysokość terenu w zadanych jednostkach („kwadratach”)
- $[c_{ij}]$ – macierz kontenerów mająca rozmiar $m \times n$: $c_{ij} = 1$, gdy na kwadracie o współrzędnych (i, j) znajduje się kontener; w przeciwnym przypadku $c_{ij} = 0$,
- k – jedna z danych, definiuje rozmiar pola widzenia kamery

Wprowadzamy macierz zmiennych decyzyjnych $[x_{ij}]$ rozmiaru $m \times n$, gdzie $x_{ij} = 1$, gdy na kwadracie o współrzędnych (i, j) znajduje się kamera; w przeciwnym przypadku $x_{ij} = 0$.

W obu modelach będziemy chcieli minimalizować całkowitą liczbę kamer użytą do obserwacji kontenerów.

$$\min f(x), f(x) = \sum_{i=1}^m \sum_{j=1}^n x_{ij}$$

Część ograniczeń dla modeli będzie się różnić ze względu na inne pole widzenia kamer.

4.2.1 Ograniczenia dla modelu z kamerą o polu widzenia będącym kwadratem

1. (tam, gdzie są kontenery, nie mogą być kamery)

$$c_{ij} + x_{ij} \leq 1, \quad i \in \{1, \dots, m\}, j \in \{1, \dots, n\}$$

2. (każdy kontener musi być obserwowany)

$$\sum_{l=i-k}^{i+k} \sum_{o=j-k}^{j+k} x_{lo} - c_{ij} \geq 0, \quad i \in \{1, \dots, m\}, j \in \{1, \dots, n\}$$

Możemy zauważyć, że jeśli $c_{ij} = 0$ to nie potrzeba żadnej kamery, zatem suma z ograniczenia może być równa zero i ograniczenie jest spełnione. Jeżeli zaś $c_{ij} = 1$, to suma musi dać co najmniej 1, czyli musi istnieć kamera obserwująca kontener (mającą go w swoim polu widzenia). Oczywiście x_{ij} i c_{ij} są równe 0, gdy $i \notin \{1, 2, \dots, m\}$ lub $j \notin \{1, 2, \dots, n\}$.

3. (na każdym polu albo jest tylko jedna kamera, albo nie ma kamer)

$$x_{ij} \in \{0, 1\}, \quad i \in \{1, \dots, m\}, j \in \{1, \dots, n\}$$

4.2.2 Ograniczenia dla modelu z kamerą o polu widzenia będącym krzyżem

1. (tam, gdzie są kontenery, nie mogą być kamery)

$$c_{ij} + x_{ij} \leq 1, \quad i \in \{1, \dots, m\}, j \in \{1, \dots, n\}$$

2. (każdy kontener musi być obserwowany)

$$\left(\sum_{l=i-k}^{i+k} x_{lj} + \sum_{o=j-k}^{j+k} x_{io} \right) - c_{ij} \geq 0, \quad i \in \{1, \dots, m\}, j \in \{1, \dots, n\}$$

Działanie tego ograniczenia jest analogiczne do tego z poprzedniego modelu; różni się ono polem widzenia kamery.

3. (na każdym polu albo jest tylko jedna kamera, albo nie ma kamer)

$$x_{ij} \in \{0, 1\}, \quad i \in \{1, \dots, m\}, j \in \{1, \dots, n\}$$

4.3 Rozwiązanie: wyniki i ich interpretacja

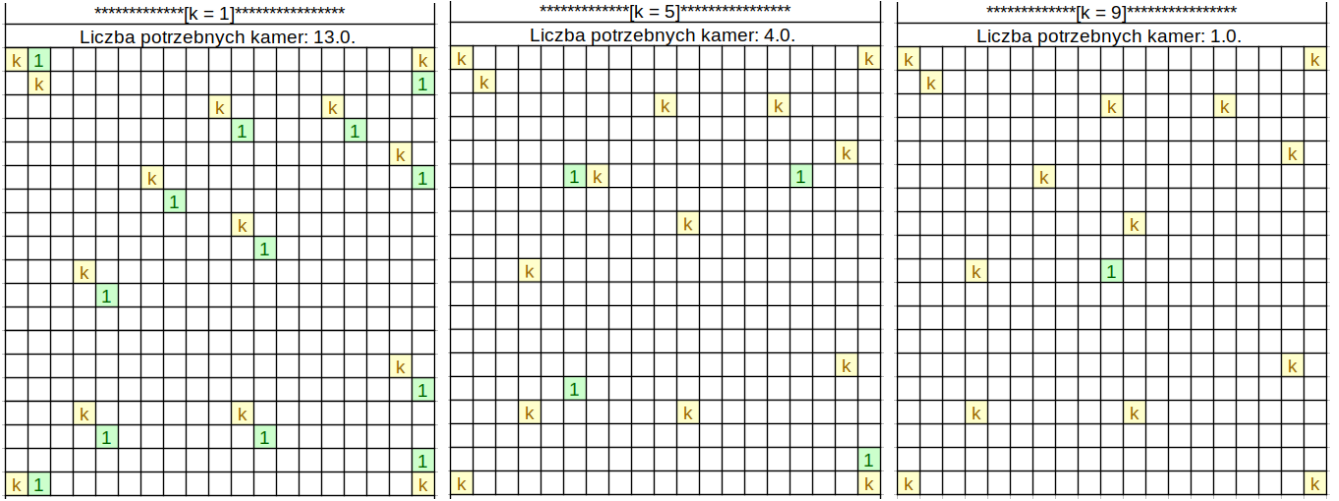
Powyższe modele zostały zapisane w języku Julia zgodnie z dokumentacją modułu JuMP. Kod źródłowy znajduje się w plikach `zadanie4_square.jl` (pierwszy model) i `zadanie4_cross.jl` (drugi model). Dane wczytywane są z pliku tekstowego następującym formacie: kwadraty w rzędzie oddzielone są spacjami, znak nowej linii wyznacza początek kolejnego rzędu. Cyfra 1 oznacza, że na danym polu znajduje się kontener; 0 oznacza brak kontenera. Do przetestowania implementacji modeli zostały utworzone dwa egzemplarze problemu, których pliki tekstowe zobrazowane są poniżej.

[illegible]

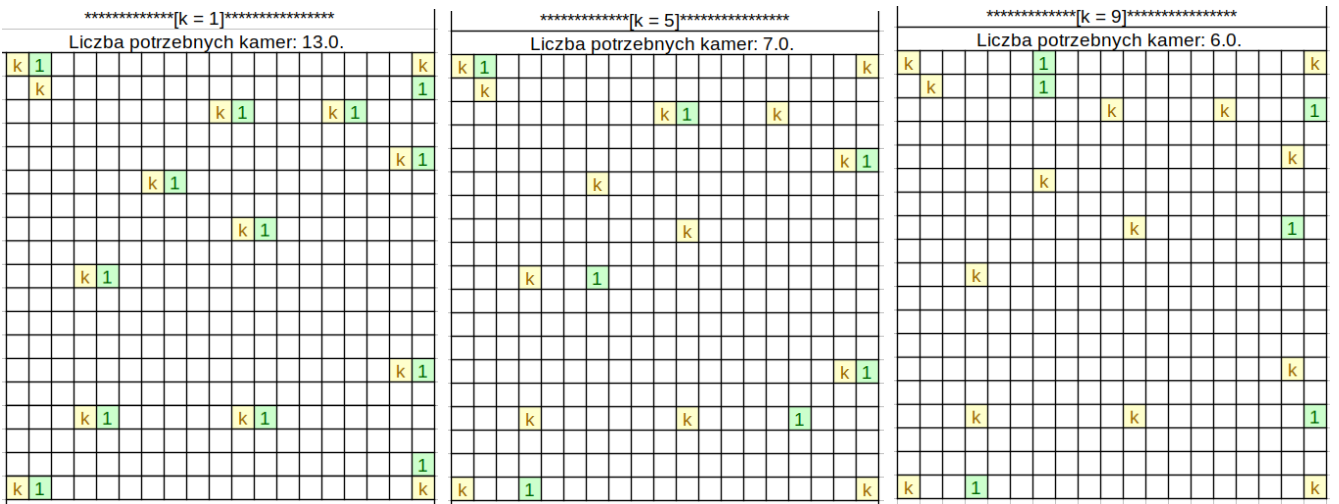
Rysunek 9: Pliki tekstowe zawierające dane na temat ułożenia kontenerów w dwóch różnych egzemplarzach problemu.

Implementacja w JuMP różni się trochę od modelu teoretycznego. Po pierwsze, możemy łatwo zrezygnować z ograniczenia 1., narzucając wartość $x_{ij} = 0$ dla pól z kontenerami ($c_{ij} = 1$) metodą `fix(x[i, j], 0.0; force = true)`. Ponadto ograniczenie 2. można uprościć i zastosować je tylko dla pól, w których są kontenery, dzięki możliwości zapisania instrukcji warunkowej `if` przy definiowaniu ograniczenia.

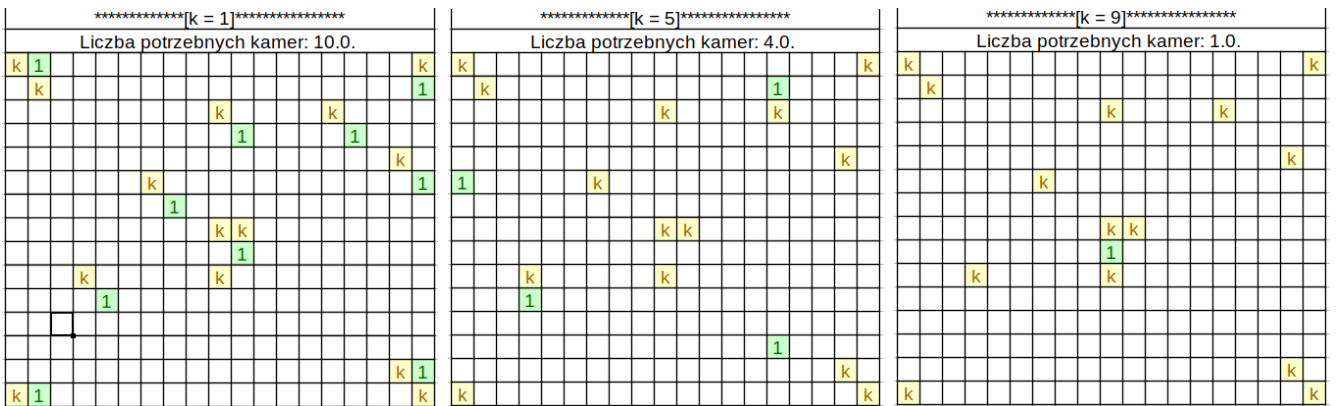
Poniższe rysunki przedstawiają rozwiązania dwóch egzemplarzy problemu, których dane zostały pokazane na Rysunku 9. Dla każdej z instancji wybrano kilka wartości parametru pola widzenia kamer k . Zauważyć na nich można, że kamery o polu widzenia w kształcie kwadratu są w stanie obserwować znacznie większą liczbę kontenerów.



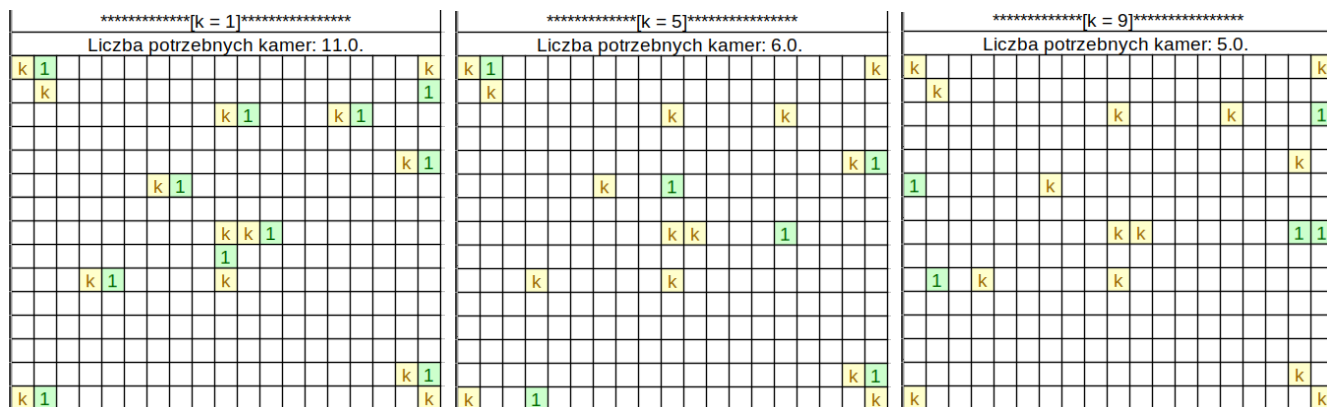
Rysunek 10: Rozwiązania pierwszego egzemplarza problemu dla modelu o polu widzenia kamer w kształcie kwadratu i parametru $k \in \{1, 5, 9\}$. Kolorem żółtym oznaczone są kontenery, zaś kolorem zielonym zaznaczono optymalne rozmieszczenie kamer.



Rysunek 11: Rozwiązania pierwszego egzemplarza problemu dla modelu o polu widzenia kamer w kształcie krzyża i parametru $k \in \{1, 5, 9\}$. Kolorem żółtym oznaczone są kontenery, zaś kolorem zielonym zaznaczono optymalne rozmieszczenie kamer.



Rysunek 12: Rozwiązania drugiego egzemplarza problemu dla modelu o polu widzenia kamer w kształcie kwadratu i parametru $k \in \{1, 5, 9\}$. Kolorem żółtym oznaczone są kontenery, zaś kolorem zielonym zaznaczono optymalne rozmieszczenie kamer.



Rysunek 13: Rozwiązania drugiego egzemplarza problemu dla modelu o polu widzenia kamer w kształcie krzyża i parametru $k \in \{1, 5, 9\}$. Kolorem żółtym oznaczone są kontenery, zaś kolorem zielonym zaznaczono optymalne rozmieszczenie kamer.

5 Zadanie 5

5.1 Treść zadania (autor: dr inż. Karol Gotfryd)

Zakład może produkować cztery różne wyroby P_i , $i \in 1, 2, 3, 4$, w różnych kombinacjach. Każdy z wyrobów wymaga pewnego czasu obróbki na każdej z trzech maszyn. Czasy te są podane w poniższej tabeli (w minutach na kilogram wyrobu). Każda z maszyn jest dostępna przez 60 godzin w tygodniu. Produkty P_1 , P_2 , P_3 i P_4 mogą być sprzedane po cenie, odpowiednio, 9, 7, 6 i 5 \$ za kilogram. Koszty zmienne (koszty pracy maszyn) wynoszą, odpowiednio, 2 \$ za godzinę dla maszyn M_1 i M_2 oraz 3 \$ za godzinę dla maszyny M_3 . Koszty materiałowe wynoszą 4 \$ na każdy kilogram wyrobu P_1 i 1 \$ na każdy kilogram wyrobu P_2 , P_3 i P_4 . W tabeli podany jest także maksymalny tygodniowy popyt na każdy z wyrobów (w kilogramach).

Produkt	Maszyna			Maksymalny popyt tygodniowy
	M_1	M_2	M_3	
P_1	5	10	6	400
P_2	3	6	4	100
P_3	4	5	3	150
P_4	4	2	1	500

Wyznacz optymalny tygodniowy plan produkcji poszczególnych wyrobów i oblicz zysk z ich sprzedaży. Zapisz model programowania liniowego w wybranym języku i rozwiąż go za pomocą solvera GLPK (lub np. Cbc).

5.2 Opis modelu

Niech

- $M = \{1, 2, \dots, m\}$ – zbiór maszyn,
- $P = \{1, 2, \dots, n\}$ – zbiór produktów,
- u_j – maksymalny łączny czas pracy j-tej maszyny w ciągu tygodnia (w minutach dla wygody obliczeń),
- p_i – cena sprzedaży kilograma i-tego produktu,

- d_i – maksymalny tygodniowy popyt na i -ty produkt [kg],
- $mc(j)$ – koszt pracy j -tej maszyny przez jedną godzinę,
- $uc(i)$ – koszt materiałowy produkcji kilograma i -tego produktu,
- t_{ij} – czas na j -tej maszynie konieczny (choć niewystarczający, potrzebna obróbka produktu przez jakiś czas na wszystkich maszynach) do wyprodukowania i -tego produktu (w minutach).

Wprowadźmy n zmiennych decyzyjnych x_i – ilość i -tego produktu do wyprodukowania [kg]. Zmienne te są wystarczające, gdyż nie mamy określonej kolejności obróbki produktu na maszynach, zatem sama ilość produktu do wyprodukowania jest wystarczającą informacją (wystarczy zachłannie kolejować na maszynach produkty, które nie zostały jeszcze przez obróbite).

Funkcją celu jest dochód z całego tygodnia, który będziemy chcieli maksymalizować. Uwzględnia on przychody ze sprzedaży produktów i wszystkie koszty poniesione przy ich produkcji.

$$\max f(x), \quad f(x) = \sum_{i \in P} \left[(p_i - \sum_{j \in M} \left(\frac{mc(j)}{60} \cdot t_{ij} \right) - uc(i)) \cdot x_i \right]$$

Ograniczenia modelu:

1. (ilość wyprodukowanych produktów nie przekracza zapotrzebowania na nie)

$$x_i \leq d_i, \quad i \in P,$$

2. (maszyny nie przekraczają czasu pracy)

$$\sum_{i \in P} t_{ij} \cdot x_i \leq u_j, \quad j \in M,$$

- 3.

$$x_{ij} \geq 0.$$

5.3 Rozwiązanie: wyniki i ich interpretacja

Powyższy model został zapisany w języku Julia zgodnie z dokumentacją modułu JuMP. Kod źródłowy znajduje się w pliku `zadanie5.jl`. Dane wczytywane są z dwóch plików tekstowych. W jednym pliku znajdują się dane dotyczące produktów: tygodniowy popyt, potrzebny czas obróbki na maszynach, cena sprzedaży za kilogram i koszty materiałowe. W drugim pliku zapisane zostały dane dotyczące maszyn: koszt pracy przez godzinę i maksymalny czas pracy w tygodniu (w minutach).

Poniższe rysunki ilustrują format danych w plikach, na przykładzie danych z treści zadania.

<code>.;MAX_DEMAND;PRICE;MATERIAL_COSTS;M1;M2;M3</code>	<code>.;MACHINE_COST;UPTIME</code>
<code>P1;400;9;4;5;10;6</code>	<code>M1;2;3600</code>
<code>P2;100;7;1;3;6;4</code>	<code>M2;2;3600</code>
<code>P3;150;6;1;4;5;3</code>	<code>M3;3;3600</code>
<code>P4;500;5;1;4;2;1</code>	

Rysunek 14: Pliki tekstowe zawierające odpowiednio dane dotyczące produktów (po lewej) i maszyn (po prawej) dla przykładu z treści zadania.

Dla danych z zadania, największy zysk wynosi 3632,5. Zostaje on uzyskany przy całkowitym spełnieniu popytu na produkty P2, P3 i P4 oraz dopełnieniu pozostałego czasu pracy maszyn mniej opłacalnym produktem P1.

Największy zysk: 3632.5

Tabela produkcji:

	P1	P2	P3	P4
	125.0	100.0	150.0	500.0

Rysunek 15: Optymalny tygodniowy plan produkcji produktów dla danych z treści zadania.