Projekt

Optymalizacja we Wspomaganiu Decyzji

Daniel Petrykowski

279 115

15.11.2019

Spis treści

1. Treść zadania

OWD AK22

Rozważamy następujące uproszczone zagadnienia optymalnego planowania produkcji:

- Przedsiębiorstwo rozpatruje możliwość podjęcia dodatkowej działalności, polegającej na produkcji i sprzedaży 7 produktów P1, …, P7 w ciągu najbliższych 3 miesięcy.
- Produkty są wytwarzane na następujących maszynach: 4 szlifierkach, 2 wiertarkach pionowych, 3 wiertarkach poziomych, 1 frezarce i 1 tokarce. Z każdym produktem związana jest marża (w zł/sztukę). Marża oraz wymagane czasy produkcji 1 sztuki produktu (w godzinach) w danym procesie obróbki zostały przedstawione w poniższej tabeli:

	P1	P2	P3	P4	P5	P6	P7
Marża	10	6	8	4	11	9	3
Szlifowanie	0,5	0,7	---	---	0,3	0,2	0,5
Wiercenie pionowe	0,1	0,2	---	0,3	---	0,6	---
Wiercenie poziome	0,2	---	0,8	---	---	---	0,6
Frezowanie	0,05	0,03	---	0,07	0,1	---	0,08
Toczenie	---	---	0,01	---	0,05	---	0,05

- Istnieją ograniczenia rynkowe na liczbę sprzedawanych produktów w danym miesiącu. Są one następujące:

	P1	P2	P3	P4	P5	P6	P7
Styczeń	500	1000	300	300	800	200	100
Luty	600	500	200	0	400	300	150
Marzec	300	600	0	0	500	400	100

- Istnieje możliwość składowania do 100 sztuk każdego produktu w danym czasie w cenie 0,5 zł/sztukę za miesiąc. Pod koniec marca wymagane jest posiadanie zapasu 50 sztuk każdego produktu.
- Należy osiągnąć możliwie największy zysk przy jak najkrótszym czasie pracy przedsiębiorstwa. Można założyć, że maksymalny czas pracy przedsiębiorstwa to 24 dni w miesiącu (6 dni w tygodniu) w systemie dwóch zmian po 8 godzin każda.

Celem projektu jest symulacja pełnej interaktywnej analizy zagadnienia przeprowadzonej metodami ważenia ocen i ważonego programowania celowego. Forma sprawozdania to raport, który powinien zawierać:

1. Analityczne sformułowanie modelu. Specyfikację problemu decyzyjnego z dookreśleniem wszystkich elementów. Określenie zmiennych decyzyjnych, ograniczeń i kryteriów.
2. Sformułowanie modelu w postaci do rozwiązania z wykorzystaniem dowolnego narzędzia/środowiska implementacji. Wyniki symulacji pełnej interaktywnej analizy zagadnienia przeprowadzonej wskazanymi metodami. Interpretację wyników w terminach oryginalnego modelu
3. Porównanie wskazanych metod

2. Wstęp Teoretyczny

Zadanie przedstawione w punkcie 1 tyczy się problemu optymalizacji wielokryterialnej.

$$max\{(f_1(x), f_2(x), \ldots, f_m(x)) : x \in Q\}$$

Stanowi to uogólnienie pojęcia rozwiązania optymalnego i w przypadku optymalizacji jednokryterialnej te dwa pojęcia są tożsame. W optymalizacji wielokryterialnej różne rozwiązania efektywne generują różne i wzajemnie nieporównywalne wektory ocen. Niewątpliwie poszukiwania rozwiązania problemu decyzyjnego powinny być w takim wypadku ograniczone do zbioru rozwiązań efektywnych. W tym celu stosuje się techniki generacji rozwiązań. W niniejszym raporcie do tego celu zastosowano metodę ważenia ocen. Metoda ta generuje rozwiązania efektywne dla dowolnych dodatnich wag $w_i > 0$ rozwiązania optymalnego skalaryzacji ważonej:

$$max\,\{\sum_{i=1}^{m} w_i f_i(x) \quad : \quad x \in Q\,\}$$

W drugiej części raportu zostanie wykorzystana metoda ważonego programowania celowego, która szerzej została opisana w rozdziale 5.

Obydwie te metody pozwalają na przeprowadzenie interaktywnej analizy zagadnienia. Takie podejście nie traktuje zbioru rozwiązań efektywnych zadania jako jego ostateczny rezultat, ale umożliwia sterowany przegląd zbioru rozwiązań efektywnych. Na podstawie podawanych przez decydentów pewnych wartości parametrów sterujących systemem wspomagania decyzji.

3. Analityczne sformułowanie modelu

3.1. Model Matematyczny

3.1.1. Zbiory indeksowe

Zmienne	**Opis**
$P = P1, \ldots, P7$	Zbiór wytwarzanych produktów
$T = T1, \ldots, T5$	Zbiór dostępnych narzędzi
$M = M1, M2, M3$	Zbiór kolejnych miesięcy dla których przeprowadzana jest analiza

3.1.2. Parametry

Parametry	**Opis**
nr_t_t	Liczba narzędzi typu t [szt]
tpu_{tp}	Czas pracy maszyny typu t przy produkcji jednej sztuki produktu p [godz]

s_l_{pm}	Limit sprzedaży produktu typu p w miesiącu m [szt]
ma_p	Marża na produkt p [zł/szt]
vm	Pojemność magazynu [szt]
mp	Koszt magazynowania pojedynczej sztuki dowolnego produktu przez miesiąc [zł]
nr_sr	Wielkość wymaganego zapasu każdego produktu pod koniec trzeciego miesiąca
dpm	Liczba dni pracujących w miesiącu [d]
zpd	Liczba zmian w każdym dniu roboczym
t_z	Czas trwania zmiany [godz]
$tw = dpm \cdot zpd \cdot t\_z$	Liczba godzin pracujących w miesiącu [godz]
$ctwt_t = nr\_t_t \cdot tw$	Dostępny czas pracy narzędzia t w każdym miesiącu [godz]

3.1.3. Zmienne

Zmienne	**Opis**
p_{pm}	Liczba sztuk produktu p wyprodukowanych w miesiącu m [szt]
$d_{p1} = p_{p1}$	Liczba sztuk produktu p dostępna do sprzedaży w pierwszym miesiącu [szt]
$d_{pm} = p_{pm} + m_{p(m-1)}$ $dla \;\; m > 1$	Liczba sztuk produktu p dostępna do sprzedaży w miesiącu m (wyprodukowane w miesiącu m + zmagazynowane w miesiącu m-1)
s_{pm}	Liczba sztuk produktu p sprzedanych w miesiącu m [szt]
$cs_p = \sum_{m \in M} s_{pm}$	Całkowita liczba sprzedanych sztuk produktu p [szt]
$m_{pm} = d_{pm} - s_{pm}$	Liczba sztuk produktu p zmagazynowanych w miesiącu m [szt]
$cmp = mp \cdot \sum_{m \in M} \sum_{p \in P} m_{pm}$	Całkowity koszt wykorzystania magazynów [zł]
$twt_{tm} = \sum_{p \in P} p_{pm} \cdot tpu_{tp}$	Czas pracy narzędzia t w miesiącu m [godz]
$e = \left(\sum_{p \in P} cs_p \cdot ma_p \right) - cmp$	Zysk całkowity ze sprzedaży produktów [zł]
$tobr_{tm} = \frac{twt_{tm}}{nr\_t_t}$	Czas trwania poszczególnych faz obróbki przy wykorzystaniu narzędzia t w danym miesiącu m [godz]
$tpp_m = \max_{t \in T} tobr_{tm}$	Czas pracy przedsiębiorstwa w danym miesiącu m [godz]
$ctpp = \sum_{m \in M} tpp_m$	Całkowity czas pracy przedsiębiorstwa w przeciągu trzech miesięcy [godz]

Przyjęto, że wszystkie narzędzia są w stanie pracować nieprzerwanie i jednocześnie. Oznacza to, że czas pracy przedsiębiorstwa w danym miesiącu będzie równy najdłuższemu czasowi trwania pracy narzędzi w tym miesiącu podczas danej fazy obróbki.

Do zmiennych decyzyjnych należy zmienna p_{pm} określająca liczbę sztuk produktu p wyprodukowanych w miesiącu m oraz zmienna s_{pm} określająca liczbę sztuk produktu p sprzedanych w miesiącu m. Pośrednio od tych zmiennych zależy natomiast zmienna m_{pm} określająca ile produktów każdego typu będzie zmagazynowanych w danym miesiącu.

3.1.4. Ograniczenia

Ograniczenia	**Opis**
$s_{pm} \leq s\_l_{pm}, \quad \forall m \in M, \quad \forall p \in P$	Ograniczenie rynkowe sprzedawanych produktów
$s_{pm} \leq d_{pm}, \quad \forall m \in M, \quad \forall p \in P$	Ograniczenie sprzedaży produktów w kolejnych miesiącach w zależności od ilości dostępnego produktu
$m_{pm} \leq vm, \quad \forall m \in M, \quad \forall p \in P$	Ograniczenie pojemności magazynów
$twt_{tm} \leq ctwt_t, \quad \forall m \in M, \quad \forall t \in T$	Ograniczenie wykorzystania czasu pracy narzędzi w danym miesiącu
$m_{p3} = nr\_sr, \quad \forall t \in P$	Wymóg zgromadzenia odpowiedniej ilości sztuk każdego produktu p na koniec trzeciego miesiąca
$tpp_m \geq tobr_{tm}, \quad \forall m \in M, \quad \forall t \in T$	Czas pracy przedsiębiorstwa jest równy najdłuższemu czasowi prac narzędzi podczas danej fazy obróbki w danym miesiącu

3.1.5. Funkcja celu

Jako funkcje celu przyjęto maksymalizacje zysku e, przy jednoczesnym minimalizowaniu czasu pracy przedsiębiorstwa

$$max\ e - ctpp$$

3.2. Model AMPL

Plik modelu i danych z pełnym kodem stworzonym w AMPL zostały dołączone do tego sprawozdania. Poniżej natomiast zostały przedstawione główne fragmenty modelu.

Listing 1 Model - zdefiniowane zbiory

```
#--------------------------------------------------------------------------------
# Zbiory
#--------------------------------------------------------------------------------
# Zbiór produkowanych produktów
set PRODUCTS;
# Zbiór dostępnych narzędzi
set TOOLS;
# Zbiór miesięcy, dla których jest przeprowadzana analiza
set MONTHS ordered;
```

Listing 2 Model - zdefiniowane parametry

```
#-----------------------------------------------------------------------------
# Parametry
#-----------------------------------------------------------------------------

# Liczba każdego z narzędzi
param toolCount { TOOLS } >= 1;
# Marża na każdy z produktów [pln/szt]
param expectedProfitPerUnit { PRODUCTS } >= 0;
# Czas pracy narzędzia t podczas produkcji produktu p
param toolTimePerUnit {TOOLS , PRODUCTS } >= 0;
# Ograniczenie rynkowe produktu p w miesiącu m
param salesMarketLimit {MONTHS , PRODUCTS} >= 0;
# Limit pojemności magazunu na produt p
param storageLimit { PRODUCTS } >= 0;
# Koszt magazynowania jednego produktu [pln]
param storageUnitCost >= 0;
# Wymagana ilość zapasu produktów pod koniec 3 miesiąca
param storageRequirInThirdM >=  0;
# Dni w miesiącu
param daysPerMonth >= 1;
# Liczba zmian wciągu dnia
param shiftsPerDay >= 2;
# Czas trwania zmiany
param hoursPerShift >= 8;
# Robotogodziny w miesiącu
param workHoursPerMonth = daysPerMonth * shiftsPerDay * hoursPerShift ;
# Dostępny czas pracy narzędzia t w miesiącu m
param availableToolTime {t in TOOLS } = toolCount [t]* workHoursPerMonth;
```

Listing 3 Model - zdefiniowane zmienne

```
#---------------------------------------------------------------------------
# Zmienne
#---------------------------------------------------------------------------

# Liczba wyprodukowanych produktów p w miesiącu m
var produced {PRODUCTS, MONTHS } >= 0 integer ;
# Liczba sprzedanych produktów p w miesiącu m
var sold {PRODUCTS, MONTHS } >= 0 integer ;
# Liczba dostępnych do sprzedaży produktów p w miesiącu m
var availableProducts {p in PRODUCTS, m in MONTHS} integer >=0;
# Liczba produktów p zmagazynowanych w miesiącu m
var stored {p in PRODUCTS, m in MONTHS} integer >= 0;
# Całkowita liczba sprzedanych produktów p
var totalSold {p in PRODUCTS } = sum {m in MONTHS } sold [p, m];
# Koszt magazynowania wszystkich produktów w miesiącu m
var monthlyStorageCost {m in MONTHS} = (sum {p in PRODUCTS} stored[p, m])*storageUnitCost;
# Całkowity koszt magazynowania produktów
var totalCostStorege = sum {m in MONTHS} monthlyStorageCost[m];
# Czas pracy narzędzia t w miesiącu m
var monthlyTimeWorkTool {t in TOOLS, m in MONTHS} = sum {p in PRODUCTS}
produced[p,m]*toolTimePerUnit[t,p];
# Zysk ze sprzedaży - Przychody
var expectedSalesProfit = sum {p in PRODUCTS} totalSold[p]*expectedProfitPerUnit[p];
# Dochody
var profit = expectedSalesProfit - totalCostStorege;
# Czas trwania poszczególych faz obróbki w miesiącu magazunu
var prodProcessingTime {t in TOOLS, m in MONTHS} = monthlyTimeWorkTool[t, m] / toolCount[t];
# Czas pracy przedsiębiorstwa w danym miesiącu m
var factoryWorkTime {m in MONTHS} >= 0;
# Całkowity czas pracy przedsiębiorstwa w przeciągu trzech miesięcy
var complateFactoryWorkTime = sum {m in MONTHS} factoryWorkTime[m];
```

Listing 4 Model - ograniczenia

```
# ---------------------------------------------------------------------------
# Ograniczenia
#----------------------------------------------------------------------------

# Ograniczenie magazynowe na sprzedaż produktów
subject to SalesLimit {p in PRODUCTS, m in MONTHS}:
                              sold[p,m] <= availableProducts[p,m];

# Ograniczenie pojemności magazynowej
subject to StorageLimit {m in MONTHS, p in PRODUCTS}:
                              stored[p,m] <= storageLimit[p];

# Ograniczenie rynkowe na sprzedaż produktów
subject to SalesMarketLimit {m in MONTHS, p in PRODUCTS}:
                              sold[p,m] <= salesMarketLimit[m,p];

# Ograniczenie czasu pracy narzędzi w miesiącu
subject to ToolWorkTime {m in MONTHS, t in TOOLS}:
                              monthlyTimeWorkTool[t,m] <= availableToolTime[t];

#Wymaganie zgromadzenia odpowiedniej ilości sztuk każdego produktu p na koniec trzeciego miesiąca
subject to StoredRequirement {p in PRODUCTS}:
        stored[p, last(MONTHS)] = storageRequirInThirdM;

# Liczba dostępnych produktów w pierwszym miesiącu
subject to firstMounthAvailableProducts {p in PRODUCTS}:
                              availableProducts[p,first(MONTHS)] = produced[p, first(MONTHS)];
# Liczba dostępnych produktów po pierwszym miesiacu
subject to anotherMounthAvailableProducts { p in PRODUCTS, m in MONTHS: m!= first(MONTHS)}:
                              availableProducts[p,m] = produced[p, m] + stored[p,prev(m)];

# Liczba przechowywanych przedmiotów
subject to Stored {p in PRODUCTS, m in MONTHS}:
             stored[p,m] <= (availableProducts[p,m]-sold[p,m]);

# Czas produkcji maksyamlana wartość z prodProcessingTime
subject to FactoryWorkTime {m in MONTHS, t in TOOLS}:
    factoryWorkTime[m] >= prodProcessingTime[t,m];
```

4. Analiza zagadnienia przeprowadzona metodą ważenia ocen

4.1. Model Matematyczny

Metoda ważenia ocen:

$$max\{\sum_{i=1}^{m} w_i f_i(x) \quad : \quad x \in Q\}$$

Dla dowolnych wag w_i $(i = 1, \dots, m)$ rozwiązanie optymalne zadania ważonego jest rozwiązaniem efektywnym zadania wielokryterialnego.

4.1.1. Parametry

Zmienne	**Opis**
$w_i \quad (i = 1,2)$	Waga oceny

4.1.2. Ograniczenia

Ograniczenia	**Opis**
$w_i \geq 0$	Wagi muszą być dodatnie

4.1.3. Funkcja celu

W rozważanym zadaniu, aby spełnić wymagania odnośnie maksymalizacji zysków przy jednoczesnym minimalizowaniu czasu pracy fabryki. Możemy wyodrębnić dwie funkcje oceny: zysk osiągnięty na przestrzeni trzech miesięcy oraz całkowity czas pracy przedsiębiorstwa w przeciągu tych miesięcy. Funkcja celu będzie zatem posiadać postać:

$$max\{w_1 \cdot e - w_2 \cdot cttp\}$$

Dodatkowo z racji tego, że wartość $cttp$ jest o dwa rzędy wielkości mniejsza od e postanowiono przeskalować ją poprzez przemnożenie $cttp$ przez 100. Dzięki temu możliwe było sprawne skalowanie funkcji ocen przy mniejszych różnicach wag $w_i > 0$. Zatem ostateczna postać funkcji celu to:

$$max\{w_1 \cdot e - w_2 \cdot 100 \cdot cttp\}$$

4.2. Model Ampl

W opracowaniu tym przedstawiony kod AMPL prezentuje jedynie zmiany wprowadzone względem modelu. Poprzez wprowadzenie funkcji celu i skalaryzacji. Pliki z pełnym kodem modelu, danych oraz plik uruchomieniowy zawierający zdefiniowaną funkcje celu w języku AMPL zostały załączone do sprawozdania.

Listing 5 Plik uruchomieniowy ze zdefiniowaną funkcją celu

```
# Konfiguracja modelu
# ----------------------------------------------------
reset;
model prod.mod ;
data prod.dat ;


# Model
# ----------------------------------------------------------
param w1 >= 0;
param w2 >= 0;


# Funkcja celu
# ----------------------------------------------------------
maximize FC : w1 * profit - w2 * 100 * complateFactoryWorkTime;
objective FC;


# Inicjalizacja wag
# ----------------------------------------------------------
let w1 := 0.8;        # waga dla funkcji oceny zysku
let w2 := 0.2;        # waga dla funkcji oceny czasu pracy


# Obliczenia
# ----------------------------------------------------------
option solver cplex ;
solve;


# Wyniki
# ----------------------------------------------------------
printf "\n###################################################\n";
display profit;
display complateFactoryWorkTime;
display FC;
```

4.3.Rozwiązanie

Rozwiązanie przy wagach 0,8 i 0,2 dla odpowiednio funkcji oceny zysku i funkcji oceny czasu pracy wygląda następująco:

Listing 6 Rozwiązanie przy wadze 0.8 dla funkcji oceny zysku i 0.2 dla funkcji oceny czasu pracy

```
CPLEX 12.9.0.0: optimal integer solution; objective 31890
16 MIP simplex iterations
0 branch-and-bound nodes

##################################################
produced [*,*]
:   JAN   FEB   MAR   :=
P1   500   600   350
P2  1000   500   650
P3   300   200    50
P4   300     0    50
P5   800   400   550
P6   200   300   450
P7     0     0    50
;

sold [*,*]
:   JAN   FEB   MAR   :=
P1   500   600   300
P2  1000   500   600
P3   300   200     0
P4   300     0     0
P5   800   400   500
P6   200   300   400
P7     0     0     0
;

stored [*,*]
: JAN FEB  MAR   :=
P1   0   0   50
P2   0   0   50
P3   0   0   50
P4   0   0   50
P5   0   0   50
P6   0   0   50
P7   0   0   50
;

profit = 58425

complateFactoryWorkTime = 742.5

FC = 31890
```

Jak widać na załączonym listingu przy tak dobranych wagach i parametrach modelu jesteśmy w stanie osiągnąć zysk w przeciągu trzech miesięcy na poziomie 58 425 zł i będzie to wymagało pracy fabryki przez 742,5 h w przeciągu trzech miesięcy.

Przy czym warto zauważyć, że wszystkie założenia zostały spełnione. W ostatnim miesiącu zostało zmagazynowane 50 sztuk produktów każdego rodzaju. A jednocześnie udało się wyprodukować maksymalną ilość możliwych sztuk produktów narzuconą przez ograniczenia rynkowe na liczbę sprzedawanych produktów w miesiącu za wyjątkiem produktu P7. Okazuje się, że jego produkcja jest nie proporcjonalnie wysoce czasochłonna do możliwych do uzyskania zysków z jego sprzedaży. Oznacza to, że zostały jeszcze wolne moce przerobowe. Z dokładniejszej analizy czasu pracy wynika, że maksymalna liczba roboczogodzin w miesiącu wynosi 384. Podczas, gdy czas faktyczny czas pracy przedsiębiorstwa wynosi w kolejnych miesiącach poczynając od stycznia 307,5 godzin, 207,5 godzin w lutym i 227,5 godziny w marcu.

- Maksymalizacja zysku przy jednoczesnym minimalizowaniu czasu pracy

Listing 7 Rozwiązanie zysk przy wadze 1 dla funkcji oceny zysku i 1 dla funkcji oceny czasu pracy

```
CPLEX 12.9.0.0: optimal integer solution within mipgap or absmipgap;
objective 8929.5
1789 MIP simplex iterations
975 branch-and-bound nodes
absmipgap = 0.885438, relmipgap = 9.91587e-05

##################################################
produced [*,*]
:   JAN   FEB   MAR   :=
P1   407    1    47
P2    8    24    18
P3   348   152   50
P4   299    0    50
P5   800   400   550
P6   185   128   179
P7    0    1    49
;

sold [*,*]
:   JAN   FEB   MAR   :=
P1   405    0    0
P2    0    0    0
P3   300   200    0
P4   299    0    0
P5   800   400   500
P6   185   128   129
P7    0    0    0
;

stored [*,*]
:  JAN  FEB  MAR   :=
P1   2   3   50
P2   8   32   50
P3   48   0   50
P4   0   0   50
P5   0   0   50
P6   0   0   50
P7   0   1   50
;

profit = 31702

complateFactoryWorkTime = 227.725

FC = 8929.5
```

Jak widać na załączonym listingu przy tak dobranych wagach i parametrach modelu jesteśmy w stanie osiągnąć zysk w przeciągu trzech miesięcy na poziomie 31 702 zł i będzie to wymagało pracy fabryki przez 227,25 h w przeciągu trzech miesięcy.

- Maksymalizacja zysku przy jednoczesnym zaniedbaniu funkcji czasu pracy

Listing 8 Rozwiązanie maksymalizujące zysk przy wadze 1 dla funkcji oceny zysku i 0.01 dla funkcji oceny czasu pracy

```
CPLEX 12.8.0.0: optimal integer solution; objective 58688.75
16 MIP simplex iterations
0 branch-and-bound nodes

####################################################
produced [*,*]
:   JAN   FEB   MAR   :=
P1    500   600   350
P2   1000   500   650
P3    300   200    50
P4    300     0    50
P5    800   400   550
P6    200   300   450
P7    100   150   150
;

sold [*,*]
:   JAN   FEB   MAR   :=
P1    500   600   300
P2   1000   500   600
P3    300   200     0
P4    300     0     0
P5    800   400   500
P6    200   300   400
P7    100   150   100
;

stored [*,*]
:  JAN FEB  MAR   :=
P1   0   0   50
P2   0   0   50
P3   0   0   50
P4   0   0   50
P5   0   0   50
P6   0   0   50
P7   0   0   50
;

profit = 59475

complateFactoryWorkTime = 786.25

FC = 58688.8
```

Jak widać na załączonym listingu przy tak dobranych wagach i parametrach modelu jesteśmy w stanie osiągnąć zysk w przeciągu trzech miesięcy na poziomie 59 475 zł i będzie to wymagało pracy fabryki przez 786,25 h w przeciągu trzech miesięcy.

W porównaniu do wyników optymalizacji z poprzednimi wartościami parametrów tym razem udało się osiągnąć maksymalną wartość sprzedaży dla każdego rodzaju produktu, Jednak, że wzrost dochodów jest dość nieznaczny bo ok. 1000 zł natomiast czas pracy wzrósł aż o 43,75 godzin. W ujęciu na poszczególne miesiące wygląda to następująco 320 godzin pracy przedsiębiorstwa w styczniu, 226.25 godzin w lutym i 240 godzin w marcu.

- Minimalizacja czasu pracy przedsiębiorstwa przy jednoczesnym zaniedbaniu funkcji zysku

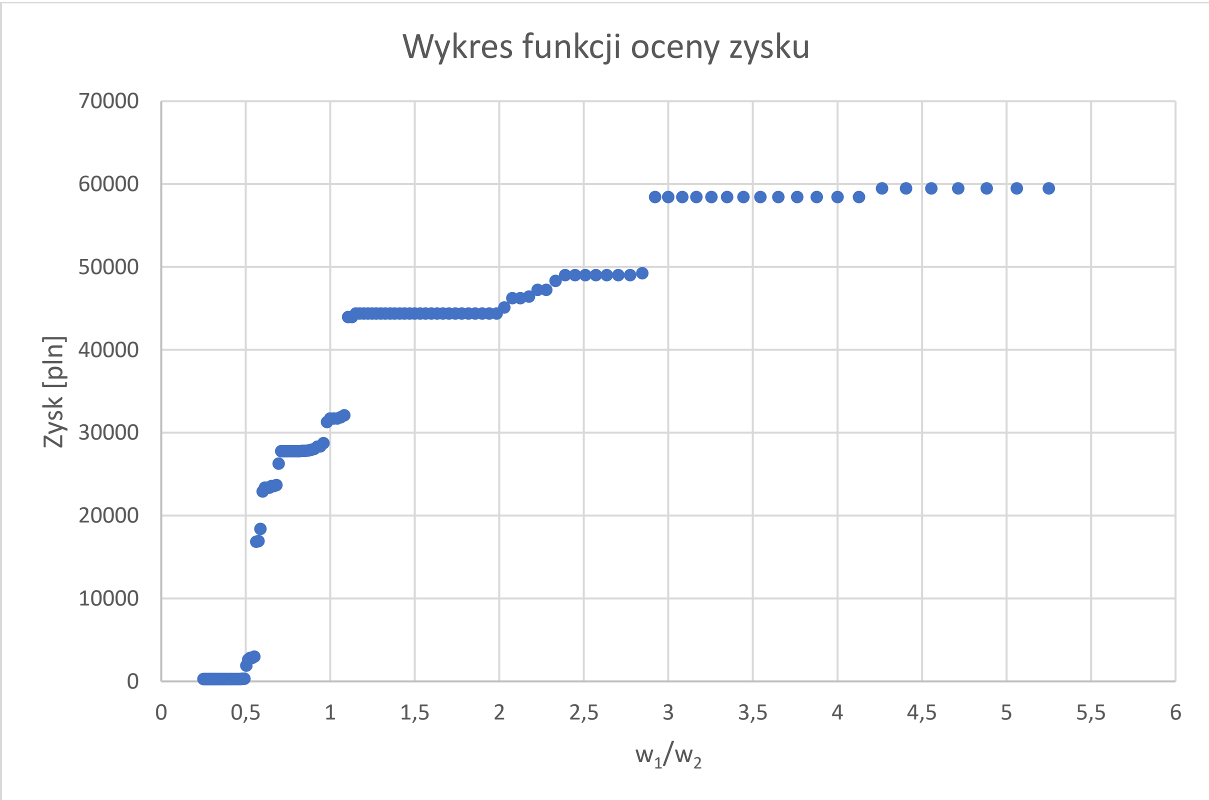
Listing 9 Rozwiązanie minimalizujące czas pracy przedsiębiorstwa przy wadze 0.01 dla funkcji oceny zysku i 1 dla funkcji oceny czasu pracy

```
CPLEX 12.8.0.0: optimal integer solution within mipgap or absmipgap; objective -
2997.225
43 MIP simplex iterations
0 branch-and-bound nodes
absmipgap = 0.0725051, relmipgap = 2.41907e-05

####################################################
produced [*,*]
:  JAN  FEB  MAR   :=
P1   0    0   50
P2   0    3   47
P3   0   12   50
P4   0    0   50
P5   0   30   53
P6   0   10   40
P7   0    0   50
;

sold [*,*]
:  JAN  FEB MAR   :=
P1   0    0   0
P2   0    0   0
P3   0   12   0
P4   0    0   0
P5   0   30   3
P6   0    0   0
P7   0    0   0
;

stored [*,*]
:  JAN  FEB  MAR   :=
P1   0    0   50
P2   0    3   50
P3   0    0   50
P4   0    0   50
P5   0    0   50
P6   0   10   50
P7   0    0   50
;

profit = 277.5

complateFactoryWorkTime = 30

FC = -2997.22
```

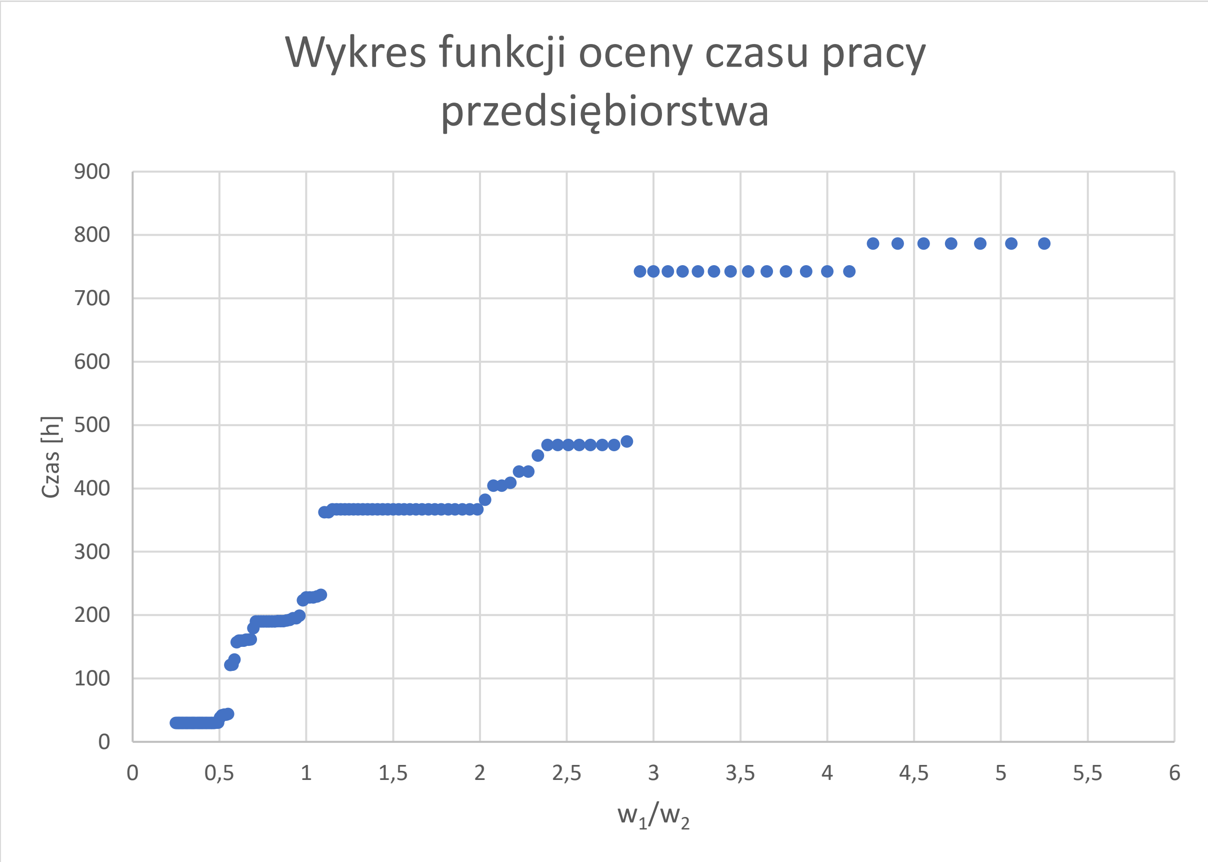
Dla tak dobranych parametrów fabryka pracuje 30 godzin w przeciągu których jest w stanie wyprodukować zysk 277,5 zł oraz produkuje 50 sztuk każdego produktu, który będzie mogła przeznaczyć na przechowanie w magazynie.

4.4. Obraz zbioru rozwiązań

Aby przedstawić wartości rozwiązań efektywnych dla różnych wartości wag wyznaczono wykres przedstawiający zmienność wartość funkcji oceny zysk oraz czas pracy przedsiębiorstwa w zależności od stosunku wagi w1 do wagi w2 . Dla każdej z wyznaczonej pary wag w1, w2 rozwiązano zadanie wielokryterialne za pomocą modelu i skryptu stworzonego w języku AMPL, który został przedstawiony w podrozdziale 4.1 i 4.2.



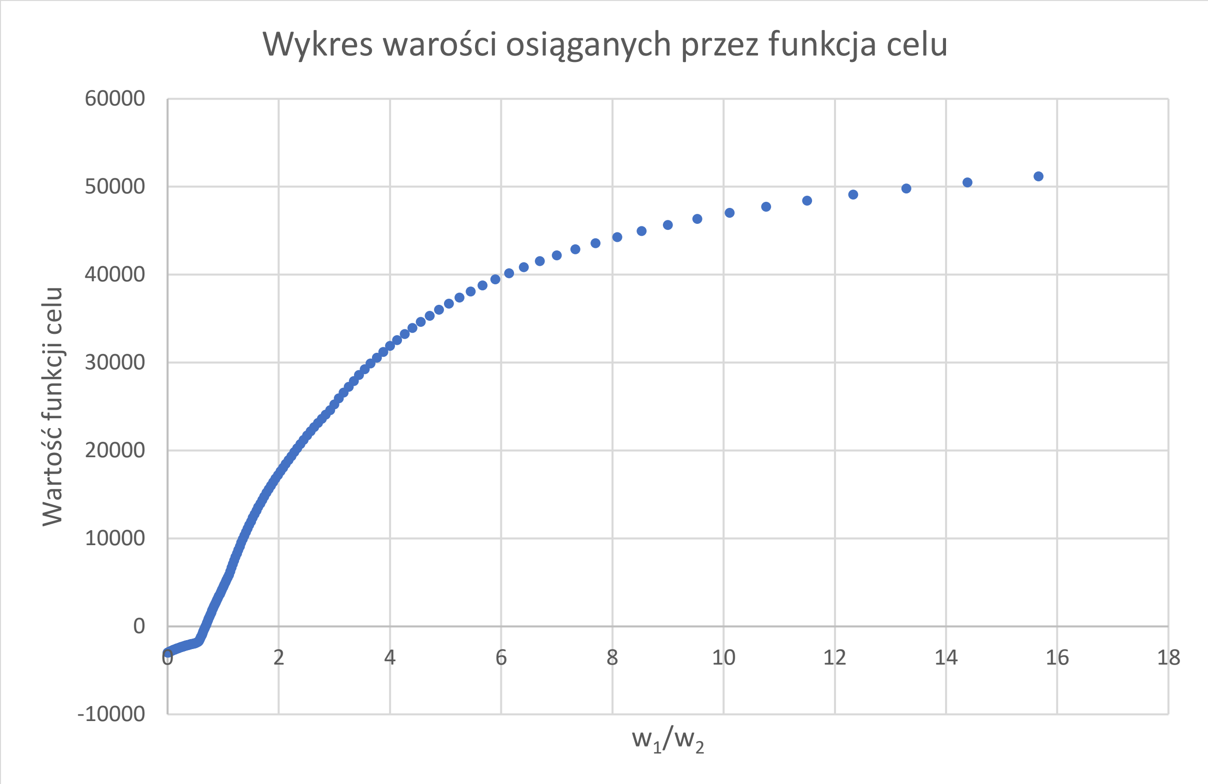
Rysunek 1 Wykres funkcji oceny zysku



Rysunek 2 Wykres funkcji oceny czasu pracy przedsiębiorstwa

Aby nie zaciemniać wyników przedstawione wykresy przedstawiają jedynie rozwiązania efektywne w zakresie od 0,25 do 5,25 stosunku w_1 do w_2. Zakres ten jest jednocześnie obszarem największej zmienności funkcji.

Na kolejnym wykresie przedstawiono wartości funkcji celu w zależności od stosunku w_1 do w_2.



Rysunek 3 Wykres wartości osiąganych przez funkcja celu

5. Analiza zagadnienia przeprowadzona metodą ważonego programowania celowego

5.1. Model Matematyczny

Dla rozwiązania zagadnienia optymalizacji wielokryterialnej przy wykorzystaniu metody ważonego programowania celowego funkcja celu przyjmuje postać:

$$g(d^-, d^+) = \sum_{i=1}^{m} (w_i^- d_i^- + w_i^+ d_i^+)$$

Przy czym w_i^- i w_i^+ są nieujemnymi wagami odpowiadającym poszczególnym odchyleniom definiowanym jako:

$$d_i^- = (a_i - f_i(x))_+ \ i \ d_i^+ = (f_i(x) - a_i)_+ \ dla\ i \ = 1,..m$$
$$d_i^- \geq 0, \qquad d_i^+ \geq 0 \ \ dla\ i \ = 1,..m$$

Zgodnie powyższym, aby uzyskać rozwiązanie które będzie najbardziej zbliżone do punktu aspiracji ustalonego przez a_i należy minimalizować funkcje celu.

$$min\ g(d^-, d^+)$$

Dodatkowo, aby zmienne d_i^- i d_i^+wyrażały odpowiednie odchylenie musi być spełniony warunek komplementarności

$$d_i^- d_i^+ = 0 \quad dla\ i = 1, .., m$$

Dla zdefiniowanego w rozdziale 3 modelu zapis powyższych zależności za pomocą programowania linowego będzie wyglądał następująco.

$$d_e^- \geq 0, \qquad d_e^+ \geq 0, \qquad d_{cttp}^- \geq 0, \qquad d_{cttp}^+ \geq 0$$

$$e + d_e^- - d_e^+ = a_e$$

$$cttp + d_{cttp}^- - d_{cttp}^+ = a_{cttp}$$

$$z = w_e^- d_e^- + w_e^+ d_e^+ + w_{cttp}^- d_{cttp}^- + w_{cttp}^+ d_{cttp}^+$$

$$min\ z$$

5.2. Model AMPL

Pełny kod AMPL rozszerzający model o metodę ważonego programowania celowego został dołączony do tego sprawozdania.

Listing 4 Model rozszerzony o metodę ważonego programowania celowego

```
# Parametry
# --------------------------------------------------------
param ae;
param acttp;
param wep;
param wem;
param wcttpp;
param wcttpm;


# Zmienne
# --------------------------------------------------------
var dem >= 0;
var dep >= 0;
var dcttpm >= 0;
var dcttpp >= 0;
var z = wep * dep + wem * dem + wcttpp * dcttpp + wcttpm * dcttpm;


# Ograniczenia
# --------------------------------------------------------
subject to PunktAspiracjiZysk:
		ae = profit + dem - dep;
subject to PunktAspiracjiCzas:
		acttp = complateFactoryWorkTime + dcttpm - dcttpp;


# Funkcja celu
# --------------------------------------------------------
minimize FC : z;
objective FC;

# Inicjalizacja wag
# --------------------------------------------------------

let ae := 60000;
let acttp := 100;
let wep := 1;		#
let wem := 1;		#
let wcttpp := 1;	#
let wcttpm := 1;	#

# Obliczenia
# --------------------------------------------------------
option solver cplex ;
solve;
```

5.3. Rozwiązanie

Rozwiązanie dla analizowanego zagadnienia przy wykorzystaniu metody ważonego programowania celowego, gdzie punkty aspiracji zostały ustawione na wartości:

a) punkt aspiracji zysk = 60000 (więcej niż jest możliwe do uzyskania przy bieżących parametrach wejściowych)
b) punkt aspiracji dla czasu produkcji = 100 (mniej niż jest potrzebne, aby zmaksymalizować zysk)

Natomiast wartość wag wynosiła 1.

Listing 11 Rozwiązanie dla metody ważonego programowania celowego (punkt aspiracji dla zysku = 60000; punkt aspiracji dla czasu produkcji = 100)

```
CPLEX 12.8.0.0: optimal integer solution; objective 1211.25
16 MIP simplex iterations
0 branch-and-bound nodes

##################################################
produced [*,*]
:   JAN   FEB   MAR   :=
P1   500   600   350
P2   1000   500   650
P3   300   200    50
P4   300     0    50
P5   800   400   550
P6   200   300   450
P7   100   150   150
;

sold [*,*]
:   JAN   FEB   MAR   :=
P1   500   600   300
P2   1000   500   600
P3   300   200     0
P4   300     0     0
P5   800   400   500
P6   200   300   400
P7   100   150   100
;

stored [*,*]
:  JAN FEB  MAR   :=
P1   0   0   50
P2   0   0   50
P3   0   0   50
P4   0   0   50
P5   0   0   50
P6   0   0   50
P7   0   0   50
;

profit = 59475

complateFactoryWorkTime = 786.25

FC = 1211.25
```

A zatem przy tak dobranych parametrach wyniki są identyczne do wyników otrzymanych przy wykorzystaniu metody ważenia ocen z maksymalizacją zysku i minimalizacją czasu pracy przy niewielkiej wadze.

Warto również rozważyć sytuacje kiedy to waga d_{cttp}^{-} i d_{cttp}^{+} będą znacznie większe od d_{e}^{-} i d_{e}^{+}.

$$d_{cttp}^{-} = 100, \qquad d_{cttp}^{+} = 100$$

$$d_{e}^{-} = 1, \qquad d_{e}^{+} = 1$$

Wtedy też:

Listing 12 Rozwiązanie dla metody ważonego programowania celowego (punkt aspiracji dla zysku = 60000; punkt aspiracji dla czasu produkcji = 100)

```
CPLEX 12.8.0.0: optimal integer solution within mipgap or absmipgap; objective 41071
125 MIP simplex iterations
35 branch-and-bound nodes
absmipgap = 2.17144, relmipgap = 5.28704e-05

###################################################
produced [*,*]
:   JAN  FEB  MAR   :=
P1  406    1   50
P2    9   25   16
P3  347  153   50
P4  300    0   50
P5  800  401  549
P6  185  128  179
P7    1    0   49
;

sold [*,*]
:   JAN  FEB  MAR   :=
P1  406    1    0
P2    0    0    0
P3  300  200    0
P4  300    0    0
P5  800  400  500
P6  185  128  129
P7    0    0    0
;

stored [*,*]
:  JAN  FEB  MAR   :=
P1   0   0  50
P2   9  34  50
P3  47   0  50
P4   0   0  50
P5   0   1  50
P6   0   0  50
P7   1   1  50
;

profit = 31726.5

complateFactoryWorkTime = 227.975

FC = 41071
```

Jak widać wyniki się w tym momencie zmieniły z racji tego, że wraz ze zwiększeniem wag dla funkcji oceny czasu pracy algorytm optymalizacji większą wagę przykłada do zminimalizowania funkcji czasu, a niżeli do maksymalizacji zysku.

Innym przypadkiem wartym analizy jest sytuacja kiedy obydwa punkty aspiracji będą znajdywały się w zakresie dopuszczalnych wartości funkcji ocen. Niech parametry zostaną ustawione na następujące wartości:

c) punkt aspiracji zysk = 40000
d) punkt aspiracji dla czasu produkcji = 400

Natomiast wartość wag wynosiła 1.

Listing 13 Rozwiązanie dla metody ważonego programowania celowego (punkt aspiracji dla zysku = 40000; punkt aspiracji dla czasu produkcji = 400)

```
CPLEX 12.8.0.0: optimal integer solution; objective 0
40 MIP simplex iterations
0 branch-and-bound nodes

####################################################
produced [*,*]
:   JAN   FEB   MAR    :=
P1   600    0   350
P2   386    0    0
P3   300   300    0
P4   300    50    0
P5   747   500   550
P6   200   242   271
P7   100    0    0
;

sold [*,*]
:   JAN   FEB   MAR    :=
P1   500   100   300
P2   336    0    0
P3   300   200    0
P4   300    0    0
P5   747   400   500
P6   200   142   321
P7    0    50    0
;

stored [*,*]
:   JAN   FEB  MAR    :=
P1   100    0   50
P2    50    50   50
P3     0    50   50
P4     0    50   50
P5     0     0   50
P6     0   100   50
P7   100    50   50
;

profit = 40000

complateFactoryWorkTime = 400

FC = 0
```

Jak widać w przypadku kiedy punkt aspiracji mieści się w zakresie możliwych do osiągnięcia punktów ocen algorytm stara się go osiągnąć i jeżeli nie stoi to w sprzeczności z innymi funkcjami ocen to zostaje on wyznaczony jako rozwiązanie. Może to prowadzić do nie oczekiwanych wyników które nie koniecznie maksymalizują zysk, a jedynie osiągają wartość punktu aspiracji. Skutkiem tego są dość nie oczekiwane działania.

6. Porównanie metody ważenia ocen z metodą ważonego programowania celowego

Zgodnie z przedstawionymi wcześniej wynikami metoda ważenia ocen jest sterowana jedynie poprzez wagi co prowadzi do wyznaczenia rozwiązania efektywnego. Jednak, że sterowanie wyborem rozwiązania jest znacznie trudniejsze i mniej intuicyjne dla decydenta, a niżeli przy wykorzystaniu metody ważonego programowania celowego.

Inną z zalet metody ważonego programowania celowego jest możliwość, przeglądu wszystkich dopuszczalnych wartości funkcji ocen. Jednak, że w przypadku nie znajomości wielkości przestrzeni ocen może to prowadzić do wyznaczenia zdominowanego wektora ocen.

Podsumowując, dzięki łatwości sterowania oraz większej liczbie możliwości kalibracji metody ważonego programowania celowego podczas wyznaczania rozwiązania jest ona znacznie potężniejszym narzędziem w rękach decydenta, a niżeli metoda ważenia ocen.

Załączniki

1. prod.mod – plik modelu w języku AMPL
2. prod.dat – plik z danymi dla modelu w języku AMPL
3. prod_m1.run – plik uruchomieniowy z zaimplementowaną metodą ważenia ocen
4. prod_m2.run – plik uruchomieniowy z zaimplementowaną metodą ważonego programowania celowego