Politechnika Warszawska
Instytut Automatyki i Informatyki Stosowanej

Sprawozdanie z projektu na przedmiot Wspomaganie Decyzji w Warunkach Ryzyka

Jan Adam Kumor, s234694

08 czerwca 2018

Spis treści

WDWR 18402

Rozważamy następujące zagadnienie planowania produkcji:

- Przedsiębiorstwo wytwarza 4 produkty P1,. . . ,P4 na następujących maszynach: 4 szlifierkach, 2 wiertarkach pionowych, 3 wiertarkach poziomych, 1 frezarce i 1 tokarce. Wymagane czasy produkcji 1 sztuki produktu (w godzinach) w danym procesie obróbki zostały przedstawione w poniższej tabeli:

	P1	P2	P3	P4
Szlifowanie	0,4	0,6	—	—
Wiercenie pionowe	0,2	0,1	—	0,6
Wiercenie poziome	0,1	—	0,7	—
Frezowanie	0,06	0,04	—	0,05
Toczenie	—	0,05	0,02	—

- Dochody ze sprzedaży produktów (w zł/sztukę) określają składowe wektora losowego $\mathbf{R} = (R_1, \dots, R_4)^T$. Wektor losowy $\mathbf{R}$ opisuje 4-wymiarowy rozkład t-Studenta z 5 stopniami swobody, którego wartości składowych zostały zawężone do przedziału $[5; 12]$. Parametry $\boldsymbol{\mu}$ oraz $\boldsymbol{\Sigma}$ niezawężonego rozkładu t-Studenta są następujące:

$$\boldsymbol{\mu} = \begin{pmatrix} 9 \\ 8 \\ 7 \\ 6 \end{pmatrix}, \qquad \boldsymbol{\Sigma} = \begin{pmatrix} 16 & -2 & -1 & -3 \\ -2 & 9 & -4 & -1 \\ -1 & -4 & 4 & 1 \\ -3 & -1 & 1 & 1 \end{pmatrix}.$$

- Istnieją ograniczenia rynkowe na liczbę sprzedawanych produktów w danym miesiącu:

	P1	P2	P3	P4
Styczeń	200	0	100	200
Luty	300	100	200	200
Marzec	0	300	100	200

- Jeżeli w danym miesiącu jest sprzedawany produkt P1 lub P2, to musi być również sprzedawany produkt P4 w ilości nie mniejszej niż suma sprzedawanych produktów P1 i P2.
- Istnieje możliwość składowania do 200 sztuk każdego produktu w danym czasie w cenie 1 zł/sztukę za miesiąc. Aktualnie firma nie posiada żadnych zapasów, ale jest pożądane mieć po 50 sztuk każdego produktu pod koniec marca.
- Przedsiębiorstwo pracuje 6 dni w tygodniu w systemie dwóch zmian. Każda zmiana trwa 8 godzin. Można założyć, że każdy miesiąc składa się z 24 dni roboczych.

1. Zaproponować jednokryterialny model wyboru w warunkach ryzyka z wartością oczekiwaną jako miarą zysku. Wyznaczyć rozwiązanie optymalne.
2. Jako rozszerzenie powyższego zaproponować dwukryterialny model zysku i ryzyka z wartością oczekiwaną jako miarą zysku i odchyleniem maksymalnym jako miarą ryzyka. Dla decyzji $\mathbf{x} \in Q$ odchylenie maksymalne jest definiowane jako $D(\mathbf{x}) = \max_{t=1,\dots,T} |\mu(\mathbf{x}) - r_t(\mathbf{x})|$, gdzie $\mu(\mathbf{x})$ oznacza wartość oczekiwaną, $r_t(\mathbf{x})$ realizację dla scenariusza t.
   a. Wyznaczyć obraz zbioru rozwiązań efektywnych w przestrzeni ryzyko–zysk.
   b. Wskazać rozwiązania efektywne minimalnego ryzyka i maksymalnego zysku. Jakie odpowiadają im wartości w przestrzeni ryzyko–zysk?
   c. Wybrać trzy dowolne rozwiązania efektywne. Sprawdzić czy zachodzi pomiędzy nimi relacja dominacji stochastycznej pierwszego rzędu. Wyniki skomentować, odnieść do ogólnego przypadku.

2 Jednokryterialny model wyboru w warunkach ryzyka z wartością oczekiwaną jako miarą zysku

W celu rozwiązania postawionego zadania dokonano sformułowania modelu programowania liniowego całkowitoliczbowego. Poniżej przedstawiono zapis matematyczny modelu.

2.1 Zbiory indeksowe

Zbiór	Opis
$P = P1, ..., P4$	Zbiór wytwarzanych produktów
$T = T1, ..., T5$	Zbiór typów narzędzi wykorzystywanych przy produkcji
$M = M1, M2, M3$	Zbiór kolejnych miesięcy produkcji

2.2 Parametry

Parametr	Opis
tc_t	Liczba narzędzi typu t [szt]
$eppu_p$	Oczekiwany zysk ze sprzedaży jednej sztuki produktu p [zł]
$ttpu_{tp}$	Czas wykorzystania maszyny typu t przy produkcji jednej sztuki produktu p [godz]
sml_{mp}	Limit sprzedaży produktu p w miesiącu m [szt]
stl_p	Limit pojemności magazynu na produkt p [szt]
$stcpu$	Koszt magazynowania jednej sztuki dowolnego produktu [zł]
$st0_p$	Początkowy stan magazynowy produktu p [szt]
dst_p	Porządany końcowy stan magazynowy produktu p [szt]
dpm	Liczba dni roboczych w każdym miesiącu [d]
spd	Liczba zmian w każdym dniu roboczym [j]
$whps$	Liczba godzin roboczych w ciągu każdej zmiany [godz]
$whpm = dpm \cdot spd \cdot hps$	Liczba godzin roboczych w ciągu każdego miesiąca [godz]
$att_t = tc_t * whpm$	Dostępna liczba godzin roboczych maszyn typu t w ciągu każdego miesiąca [godz]

2.3 Zmienne

Zmienna	Opis
p_{mp}	Liczba sztuk produktu p wyprodukowanych w miesiącu m [szt]
s_{mp}	Liczba sztuk produktu p sprzedanych w miesiącu m [szt]
$ts_p = \sum_{m \in M} s_{mp}$	Całkowita liczba sprzedanych sztuk produktu p
$std_{mp} = p_{mp} - s_{mp}$	Liczba sztuk produktu p zmagazynowanych w miesiącu m [szt]
$stg_{mp} = st0_p + \sum_{m_2=1}^{m} std_{m_2p}$	Stan magazynowy produktu p na koniec miesiąca m [szt]
$utt_{mt} = \sum_{p \in P} p_{mp} * ttpu_{tp}$	Wykorzystanie czasu pracy maszyny typu t w miesiącu m [godz]
$tstc = stcpu \cdot \sum_{m \in M} \sum_{p \in P} stg_{mp}$	Całkowity koszt wykorzystania magazynów [zł]
$ep = (\sum_{p \in P} ts_p \cdot eppu_p) - tstc$	Wartość zysku całkowitego dla wartości oczekiwanych zysku ze sprzedaży produktów [zł]

2.4 Ograniczenia

Ograniczenie rynkowe sprzedawanych produktów:

$$s_{mp} \leqslant sml_{mp}, \quad \forall m \in M, \quad \forall p \in P$$

Ograniczenie sprzedaży produktów w pierszym miesiącu:

$$s_{1p} \leqslant p_{1p}, \quad \forall p \in P$$

Ograniczenie sprzedaży produktów w kolejnych miesiącach:

$$s_{mp} \leqslant p_{mp} + stg_{mp}, \quad \forall m \in M \setminus \{1\}$$

Ograniczenie na powiązanie sprzedaży produktu 4 ze sprzedażą produktów 1 i 2:

$$s_{m4} \geqslant s_{m1} + s_{m2}, \quad \forall m \in M$$

Ograniczenie pojemności magazynów:

$$stg_{mp} \leqslant stl_p, \quad \forall p \in P$$

Ograniczenie na pożądany stan magazynowy na koniec miesiąca 3:

$$stg_{3p} \geqslant dst_p, \quad \forall p \in P$$

Ograniczenie wykorzystania czasu pracy narzędzi w danym miesiącu:

$$utt_{mt} \leqslant att_t, \quad \forall t \in T, \quad \forall m \in M$$

2.5 Funkcja celu

Jako funkcję celu przyjęto maksymalizację wartości oczekiwanej zysku: $maximize \quad ep$

3 Dwukryterialny model zysku i ryzyka z wartoscią oczekiwaną jako miarą zysku i odchyleniem maksymalnym jako miarą ryzyka

Model ten został zrealizowany jako rozszerzenie modelu jednokryterialnego o dodatkowe zbiory, parametry, zmienne, ograniczenia i nową funkcję celu.

3.1 Zbiory indeksowe

Zbiór	Opis
$S = S1, ..., S1000$	Zbiór scenariuszy wygenerowanych z rozkładu t-Studenta

3.2 Parametry

Parametr	Opis
$sppu_{ps}$	Zysk ze sprzedaży jednej sztuki produktu p w scenariuszu s [zł]

3.3 Zmienne

Zmienna	Opis
$sp_s = (\sum_{p \in P} ts_p \cdot sppu_p s) - tstc$	Wartość zysku całkowitego dla scenariusza s zysku ze sprzedaży produktów [zł]
$dev_s = \|ep - sp_s\|$	Odchylenie zysku w danym scenariuszu [zł]. Jako, że funkcja wartości bezwzględnej jest nieliniowa zmienna została poddana linearyzacji z użyciem zmiennych $ldev_s$, P_s, Q_s
$ldev_s = ep - sp_s$	Zmienna pomocnicza wykorzystana w linearyzacji odchylenia zysku w scenariuszu s
P_s	Zmienna pomocnicza wykorzystana w linearyzacji zmiennejk dev_s
Q_s	Zmienna pomocnicza wykorzystana w linearyzacji zmiennej dev_s
$mdev = \max_{s \in S} dev_s$	Maksymalne odchylenie zysu [zł]. Jako, że funkcja max jest nieliniowa, zmienna została poddana linearyzacji z użyciem zmiennych M, Z_s
M	Zmienna pomocnicza wykorzystana w linearyzacji zmiennej $mdev$
Z_s	Zmienna pomocnicza binarna wykorzystana w linearyzacji zmiennej $mdev$
$r = mdev$	Miara ryzyka, równa maksymalnemu odchyleniu zysku

3.4 Ograniczenia

Ograniczenie związane z linearyzacją zmiennej dev_s:

$$ldev_{s1} - ldev_{s2} + P_{s1} - Q_{s2} = 0, \quad \forall s_1, s_2 \in S$$

Ograniczenie związane z linearyzację zmiennej $mdev$:

$$mdev \geqslant dev_s, \quad \forall s \in S$$
$$mdev \leqslant dev_s + M(1 - Z_s), \quad \forall s \in S$$
$$\sum_{s \in S} Z_s = 1$$

3.5 Metoda punktu odniesienia

Jako model preferencji dla modelu dwukryterialnego została wybrana metoda punktu odniesienia. Wprowadza ona zestaw dodatkowych parametrów i zmiennych:

Parametr	Opis
asp_{ep}	Poziom aspiracji oczekiwanego zysku
asp_r	Poziom aspiracji ryzyka
λ_{ep}, λ_r	Współczynniki normalizujące, odpowiednio dla zysku i ryzyka. Ze względu na ogólne sformułowanie metody punktu odniesienia jako problemu maksymalizacji, λ_{ep} przyjmie wartość dodatnią, a λ_r ujemną.
β	Współczynnik pomniejszający wartość ocen wykraczających powyżej poziomu aspiracji
ε	Współczynnik składnika regularyzacyjnego
Zmienne	Opis
oc_{ep}, oc_r	Wartości indywidualnych funkcji osiągnięć dla zysku i ryzyka
v	Zmienna pomocnicza metody punktu odniesienia

Ograniczenia zmiennej v przez wartości indywidualnych funkcji osiągnięć:

$$v \leqslant oc_{ep} \quad \text{oraz} \quad v \leqslant oc_r$$

Ograniczenia indywidualnych funkcji osiągnięć:

$$oc_r \leqslant \lambda_r(r - asp_r)$$
$$oc_r \leqslant \beta\lambda_r(r - asp_r)$$
$$oc_{ep} \leqslant \lambda_p(ep - asp_{ep})$$
$$oc_{ep} \leqslant \beta\lambda_p(ep - asp_{ep})$$

Funkcja celu metody punktu odniesienia w postaci dla programowania liniowego:

$$\max \quad v + \varepsilon(oc_{ep} + oc_r)$$

4 Wyznaczenie parametrów zadania z rozkładu t-Studenta

W celu wyznaczenia wartości oczekiwanej wektora R (odpowiadającą parametrowi modelu $eppu_p$) wykorzystano następującą zależność:

$$E(R) = \mu + \sigma \cdot \frac{\Gamma(\frac{\nu-1}{2})((\nu + a^2)^{-\frac{\nu-1}{2}} - (\nu + b^2)^{\frac{\nu-1}{2}})\nu^{\frac{\nu}{2}}}{2(F_\nu(b) - F_\nu(a))\Gamma(\frac{\nu}{2})\Gamma(\frac{1}{2})}$$

gdzie:

- μ - wartość oczekiwana dla R,
- Γ - funkcja gamma Eulera,
- ν - liczba stopni swobody,
- F - dystrybuanta standardowego rozkładu t-Studenta $t(0,1;\nu)$ z ν stopniami swobody,
- $a = \frac{\alpha-\mu}{\sigma}$, gdzie α to lewy kraniec przedziału,
- $b = \frac{\beta-\mu}{\sigma}$, gdzie $\beta$to prawy kraniec przedziału.

Otrzymano wartości:

$$E(R)^T = [8.5094, 8.4710, 8.1319, 6.3944]$$

Do obliczenia wartości oczekiwanej oraz wyznaczenia scenariuszy wykorzystano skrypt napisany w języku R. Wygenerowano 1000 scenariuszy testowtych. Użyty skrypt przedstawia Listing 1.

Listing 1: Skrypt w języku R do obliczania wartości oczekiwanej wektora R i generowania scenariuszy z rozkładu t-Studenta.

```
library(tmvtnorm)

# t-Stutdet parameters
Mu = c(9, 8, 7, 6)
Sigma = matrix(c(16, -2, -1, -3,
                 -2,  9, -4, -1,
                 -1, -4,  4,  1,
                 -3, -1,  1,  1),
               nrow=4, ncol=4)
lower_bound = 5
upper_bound = 12

# Generate scenarios
data <- rtmvt(n=10000, mean=mu, sigma=sigma, df=5, lower=rep(lower_bound, 4), upper=rep(upper_bound, 4))
write.table(format(data, digits=15, drop0trailing=F), "data10000.txt", quote=F, sep="\t", eol="\n\t", col.names = F, row.names = T)
mean <- colMeans(data)

E <- function(idx, Mu, Sigma, v, alfa, beta) {
  mu = Mu[idx]
  sigma = Sigma[idx, idx]
  a = (alfa - mu)/sigma
  b = (beta - mu)/sigma
  nom = gamma((v-1)/2) *
        ((v+a^2)^(-1*(v-1)/2) -
         (v+b^2)^(-1*(v-1)/2)) *
        v^(v/2)
  den = 2 * (pt(b, v) - pt(a, v)) * gamma(v/2) * gamma(1/2)
  return (mu + sigma*(nom/den))
}

ER1 <- E(1, Mu, Sigma, 5, 5, 12)
ER2 <- E(2, Mu, Sigma, 5, 5, 12)
ER3 <- E(3, Mu, Sigma, 5, 5, 12)
ER4 <- E(4, Mu, Sigma, 5, 5, 12)
```

```
ER <- c(ER1, ER2, ER3, ER4)
write.table(ER, "ER.txt", sep="\t", col.names=F, row.names=F)
```

5 Model dla programu AMPL

5.1 Plik z modelem (.mod)

Listing 2: Model AMPL.

```
############################################################
# WDWR 18042                                               #
# Planowanie produkcj w warunkach ryzyka.                  #
# MODEL                                                    #
# Autor: Jan Kumor                                         #
############################################################

##########
# Zbiory #
##########
# Produkty
set PRODUCTS = {"P1", "P2", "P3", "P4"};
# Narzedzia
set TOOLS;
# Miesiace
set MONTHS ordered;
# Scenariusze
param scenarioCount = 1000;
set SCENARIOS = {1..scenarioCount};

#############
# Parametry #
#############

# Liczba kazdego z narzedzi
param toolCount {TOOLS} >= 1;

# Dochody ze sprzedazy [pln/szt]
param expectedProfitPerUnit {PRODUCTS} >= 0;

# Scenarios
param scenarioProfitPerUnit {SCENARIOS, PRODUCTS};

# Czasy produkcji [godz]
param toolTimePerUnit {TOOLS, PRODUCTS} >= 0;

# Ograniczenia rynkowe liczby sprzedawanych produktow [szt]
param salesMarketLimit {MONTHS, PRODUCTS} >= 0;

# Ograniczeine liczby magazynowanych produktow [szt]
param storageLimit {PRODUCTS} >= 0;

# Koszt magazynowania produktow [pln/szt per msc]
param storageUnitCost >= 0;
```

```

# Aktualny stan magazynowy [szt]
param startingStorage {PRODUCTS} >= 0;

# Pozadany stan magazynowy na koniec symulacji [szt]
param desiredEndStorage {PRODUCTS} >= 0;

# Liczba dni roboczych w miesiacu [d]
param daysPerMonth >= 1;

# Liczba zmian w ciagu jednego dnia roboczego
param shiftsPerDay >= 1;

# Dlugosc zmiany [godz]
param hoursPerShift >= 1;

# Liczba roboczogodzin w miesiacu [godz]
param workHoursPerMonth = daysPerMonth*shiftsPerDay*hoursPerShift;

# Czas pracy narzedzi w danym miesiacu
param availableToolTime {t in TOOLS} = toolCount[t]*workHoursPerMonth↩
    ;

###########
# Zmienne #
###########
# Produkcja produktow
var produced {MONTHS, PRODUCTS} >= 0 integer;

# Sprzedaz produktow w danym miesiacu
var sold {MONTHS,PRODUCTS} >= 0 integer;
var totalSold {p in PRODUCTS} = sum {m in MONTHS} sold[m, p];

# Ilosc produktow przekazanych do magazynu w danym miesiacu
var stored {m in MONTHS, p in PRODUCTS} = produced[m, p] - sold[m, p↩
    ];

# Stan magazynowy na koniec danego miesiaca
var storage {m in MONTHS, p in PRODUCTS} =
        startingStorage[p] + sum {m2 in MONTHS: ord(m2) <= ord(m)} ↩
            stored[m2, p];

# Wykorzystany czas pracy
var usedToolTime {m in MONTHS, t in TOOLS} =
        sum {p in PRODUCTS} produced[m,p]*toolTimePerUnit[t,p];

# Koszt magazynowania
var monthlyStorageCost {m in MONTHS} =
        (sum {p in PRODUCTS} storage[m, p])*storageUnitCost;
var totalStorageCost = sum {m in MONTHS} monthlyStorageCost[m];

# Zysk dla wartosci oczekiwanej
var expectedSalesProfit  =
        sum {p in PRODUCTS} totalSold[p]*expectedProfitPerUnit[p];
var expectedNetProfit =
        expectedSalesProfit - totalStorageCost;
```

```

# Zysk w danym scenariuszu
var scenarioSalesProfit {s in SCENARIOS} =
        sum {p in PRODUCTS} totalSold[p]*scenarioProfitPerUnit[s, p];
var scenarioNetProfit {s in SCENARIOS} =
        scenarioSalesProfit[s] - totalStorageCost;

# Odchylenie jako miara ryzyka - zlinearyzowana wartosc bezwzgledna
var deviation {s in SCENARIOS} =
        expectedNetProfit - scenarioNetProfit[s];
var P {SCENARIOS} >= 0;
var Q {SCENARIOS} >= 0;
subject to deviationLimit {s1 in SCENARIOS, s2 in SCENARIOS}:
        deviation[s1]-deviation[s2]+P[s1]-Q[s2] = 0;

#var maxDeviation = max {s in SCENARIOS} deviation[s];
var maxDeviation;
# Linearyzacja maksymalnego odchylenia jako miary ryzyka
param M = 10000;
var Z {SCENARIOS} binary;
subject to mdLimit {s in SCENARIOS}:
        maxDeviation >= deviation[s];
subject to mdWhere {s in SCENARIOS}:
        maxDeviation <= deviation[s] + M*(1-Z[s]);
subject to mdOS:
        sum{s in SCENARIOS} Z[s] = 1;

# Aliasy dla ocenianych wartosci
var profit = expectedNetProfit;
var risk = maxDeviation;

#######################
# Ograniczenia modelu #
#######################

# Ograniczenie rynkowe sprzedazy produktow
subject to SalesMarketLimit {m in MONTHS, p in PRODUCTS}:
        sold[m, p] <= salesMarketLimit[m, p];
# Ograniczenie magazynowe sprzedazy produktow
subject to SalesLimit1 {p in PRODUCTS}:
        sold[first(MONTHS), p] <= produced[first(MONTHS), p];
subject to SalesLimit2 {m in MONTHS, p in PRODUCTS: m != first(MONTHS)}:
        sold[m, p] <= produced [m, p] + storage[m, p];
# Powiazanie sprzedazy produktu P4 ze sprzedaza produktow P1 i P2
subject to P4SalesConstraint {m in MONTHS}:
        sold[m, "P4"] >= sold[m, "P1"] + sold[m, "P2"];
# Ograniczenie pojemnosci magazynowej
subject to StorageLimit {m in MONTHS, p in PRODUCTS}:
        storage[m, p] <= storageLimit[p];
# Ograniczenie na pozadany stan magazynowy na koniec marca
subject to DesiredStorage {p in PRODUCTS}:
        storage[last(MONTHS), p] >= desiredEndStorage[p];
#Ograniczenie czasu pracy narzedzi w miesiacu
subject to ToolWorkTime {m in MONTHS, t in TOOLS}:
        usedToolTime[m, t] <= availableToolTime[t];
```

```

##############################
# Metoda punktu odniesienia #
##############################
# Skladniki wektora oceny
set RATED = {"PROFIT", "RISK"};
# Wektor oceny
var value {r in RATED} =
        if r == "PROFIT" then profit
        else if r == "RISK" then risk;
# Wektor aspiracji
param aspiration {RATED};
# Wartosci utopii i nadiru
param utopia {RATED};
param nadir {RATED};
# Wspolczynniki normalizujace
param lambda {r in RATED} =
        1 / (utopia[r]-nadir[r]);
# Wspolczynnik skladnika regularyzacyjnego
param epsilon;
# Wspolczynnik pomniejszenia wartosci ocen ponad poziomem aspiracji
param beta;
# Indywidualne funkcje osiagniec
var individualRating {RATED};
# Zmienna pomocnicza metody punktu odniesienia
var v;
# Skalaryzujaca funkcja osiagniecia
var rating = v + epsilon * (sum {r in RATED} individualRating[r]);
# Odleglosc od punktu odniesienia
var distance {r in RATED} = value[r]-aspiration[r];
# Znormalizowana odleglosc od punktu odniesienia
var normalizedDistance {r in RATED} = lambda[r]*distance[r];
# Ograniczenia zmiennej v przez indywidualne funkcje osiagniec
subject to VSubject {r in RATED}:
        v <= individualRating[r];
# Ograniczenia indywidualnych funkcji osiagniec
subject to IndividualRatingSubjectBeta {r in RATED}:
        individualRating[r] <= beta*normalizedDistance[r];
subject to IndividualRatingSubject {r in RATED}:
        individualRating[r] <= normalizedDistance[r];

################
# Funkcje celu #
################
minimize MinimizeProfit: profit;
maximize MaximizeProfit: profit;
minimize MinimizeRisk: risk;
maximize MaximizeRisk: risk;
maximize RPM: rating;
```

5.2 Plik z danymi (.dat)

Listing 3: Dane dla modelu AMPL - pominięto scenariusze, pełny zestaw danych dostępny w załączniku.

```
##############################################################
# WDWR 18042                                                 #
# Planowanie produkcj w warunkach ryzyka.                    #
# DANE                                                       #
# Autor: Jan Kumor                                           #
##############################################################

# Narzedzia
set TOOLS := GRINDER VDRILL HDRILL MILLER LATHE;

# Miesiace
set MONTHS := JAN FEB MAR;

# Liczba narzedzi
param toolCount :=
        GRINDER 4
        VDRILL  2
        HDRILL  3
        MILLER  1
        LATHE   1
        ;

# Czasy produkcji h
param toolTimePerUnit:
                                P1              P2              P3 P4 :=
        GRINDER         0.4             0.6             0       0
        VDRILL          0.2             0.1             0       0.6
        HDRILL          0.1             0               0.7     0
        MILLER          0.06    0.04    0               0.05
        LATHE           0               0.05    0.02    0
        ;

# Ograniczenia rynkowe liczby sprzedawanych produktow pcs
param salesMarketLimit:
                        P1              P2              P3      P4      :=
        JAN             200             0               100     200
        FEB             300             100             200     200
        MAR             0               300             100     200
        ;

# Ograniczeine liczby magazynowanych produktow pcs
param storageLimit :=
        P1      200
        P2      200
        P3      200
    P4  200
```

```
        ;

# Koszt magazynowania produktow pln/pcs per month
param storageUnitCost := 1;

# Aktualny stan magazynowy pcs
param startingStorage :=
        P1      0
        P2      0
        P3      0
        P4      0
        ;

# Pozadany stan magazynowy na koniec marca pcs
param desiredEndStorage :=
        P1      50
        P2      50
        P3      50
        P4      50
        ;

# Liczba dni roboczych w miesiacu d
param daysPerMonth := 24;

# Liczba zmian w ciagu jednego dnia roboczego
param shiftsPerDay := 2;

# Dlugosc zmiany h
param hoursPerShift := 8;

# Zyski wartosc oczekiwana
param expectedProfitPerUnit :=
        P1      8.50944172786882
        P2      8.47100593224391
        P3      8.1319049712769
        P4      6.39446520538826
        ;

# Metoda punktu odniesienia
param epsilon = 0.000025;

param beta = 0.001;

param utopia :=
        PROFIT  11987
        RISK    1000
        ;

param nadir :=
        PROFIT  -2400
        RISK    2815
        ;

param           aspiration :=
        PROFIT  10000
        RISK    0
```

```
        ;

# Scenariusze
param scenarioProfitPerUnit:
                        P1                                       P2↩
                                                                  P3↩
                                                                  P4 ↩
                            :=
        1               6.78312108289149          5.79640238361981↩
                10.05787433056357        6.57331416435723
    ...
```

5.3 Skrypty uruchomieniowe (.run)

Listing 4: Skrypt wyznaczający wektory utopii i nadiru.

```
###########################################################
# WDWR 18042                                              #
# Planowanie produkcj w warunkach ryzyka.                 #
# SKRYPT URUCHAMIAJACY                                    #
# Autor: Jan Kumor                                        #
###########################################################

#######################
# Konfiguracja modelu #
#######################
model WDWR.mod;
data WDWR.dat;
option solver cplex;

###########################################################
# Rozwiazania optymalne dla wyznaczenia granic zmiennosci #
# tj wektorow utopii i nadiru                             #
###########################################################
# Minimalny zysk
printf "\n########################\n";
printf "### Minimizing profit ###\n";
printf "########################\n";
objective MinimizeProfit;
solve;
printf "Profit: %d\n", profit;
printf "Risk: %d\n", risk;

# Maksymalny zysk
printf "\n########################\n";
printf "### Maximizing profit ###\n";
printf "########################\n";
objective MaximizeProfit;
solve;
printf "Profit: %d\n", profit;
printf "Risk: %d\n", risk;

# Minimalny poziom ryzyka
printf "\n######################\n";
printf "### Minimizing risk ###\n";
```

```
printf "######################\n";
objective MinimizeRisk;
solve;
printf "Profit: %d\n", profit;
printf "Risk: %d\n", risk;

# Maksymalny poziom ryzyka
printf "\n#######################\n";
printf "### Maximizing risk k###\n";
printf "#######################\n";
objective MaximizeRisk;
solve;
printf "Profit: %d\n", profit;
printf "Risk: %d\n", risk;
```

Listing 5: Skrypt wyznaczający rozwiązania optymalne modelu jednokryterialengo.

```
##########################################################
# WDWR 18042                                             #
# Planowanie produkcj w warunkach ryzyka.                #
# SKRYPT URUCHAMIAJACY - Metoda punktu odniesienia       #
# Autor: Jan Kumor                                       #
##########################################################

#######################
# Konfiguracja modelu #
#######################
model WDWR.mod;
data WDWR.dat;
option solver cplex;

#############################
# Model jednokryterialny    #
#############################
printf "\n################################################\n";
printf "### Maximize profit for expected profit value ###\n";
printf "#################################################\n";
objective Profit;
solve;

display produced;
display sold;
display stored;
printf "Profit: %f\n", MaximizeProfit;
```

Listing 6: Skrypt wyznaczający rozwiązania optymalne modelu dwukryterialnego.

```
##########################################################
# WDWR 18042                                             #
# Planowanie produkcj w warunkach ryzyka.                #
# SKRYPT URUCHAMIAJACY - Metoda punktu odniesienia       #
# Autor: Jan Kumor                                       #
##########################################################

#######################
# Konfiguracja modelu #
#######################
```

```
model WDWR2.mod;
data WDWR2_trunc.dat;
option solver cplex;

##############################
# Metoda punktu odniesienia #
##############################
printf "\n######################################\n";
printf "### RPM solution space calculation ###\n";
printf "######################################\n";
objective RPM;

param steps = 10;
param stepSize {r in RATED} = (utopia[r] - nadir[r]) / (steps-1);
param iteration;
param iterationCount = steps*steps;
set RESULTS = {1..iterationCount};
set VALUES = {"PROFIT", "RISK", "RPM"};
param result {RESULTS, VALUES};

for {i in 0..steps-1} {
        for {j in 0..steps-1} {
                let iteration := 1 + steps*i + j;
                let aspiration["PROFIT"] := nadir["PROFIT"] + i * stepSize["PROFIT"];
                let aspiration["RISK"] := nadir["RISK"] + j * stepSize["RISK"];
                printf "### %d: Solving model for aspirations: %f, %f\n",
                        iteration, aspiration["PROFIT"], aspiration["RISK"];
                solve;
                let result[iteration, "PROFIT"] := profit;
                let result[iteration, "RISK"] := risk;
                let result[iteration, "RPM"] := RPM;
                printf "Profit: %f\n", profit;
                printf "Risk: %f\n", risk;
                printf "RPM: %f\n", RPM;
        }
}

display result;

printf { r in RESULTS } "%f\t%f\t%f\n", result[r,"PROFIT"], result[r,"RISK"], result[r,"RPM"] > ./result.csv ;
```

6 Rozwiązanie zadania optymalizacji

6.1 Wyniki dla modelu jednokryterialnego

Rozwiązanie optymalne modelu jednokryterialny zostało wyznaczone z użyciem solvera *CPLEX*. Stosowny skrypt uruchamiający dla programu *AMPL* przedstawia Listing 5. Wyniki wywołania przedstawia **??** zamieszczony poniżej. Na listingu widzimy wartości zmiennych decyzyjnych p_{mp},

$s_m p$ oraz wartość wyznaczonego rozwiązania optymalnego:

$$ep = 11987.42[z]$$

Listing 7: Wynik działania skryptu wyznaczającego rozwiązanie optymalne modelu jednokryterialnego.

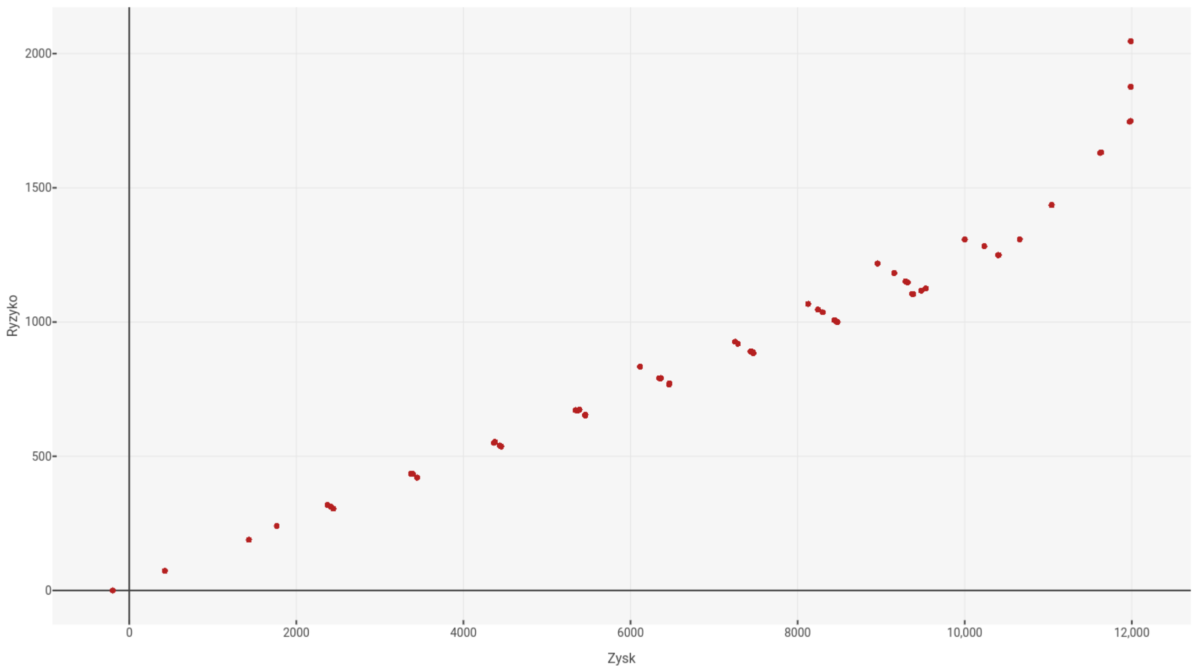
```
##################################################
### Maximize profit for expected profit value ###
##################################################
CPLEX 12.8.0.0: optimal integer solution; objective 11987.41899
11 MIP simplex iterations
0 branch-and-bound nodes
produced :=
JAN P1   200
JAN P2     0
JAN P3   100
JAN P4   200
FEB P1   200
FEB P2     0
FEB P3   200
FEB P4   200
MAR P1    50
MAR P2   250
MAR P3   150
MAR P4   250
;

sold :=
JAN P1   200
JAN P2     0
JAN P3   100
JAN P4   200
FEB P1   200
FEB P2     0
FEB P3   200
FEB P4   200
MAR P1     0
MAR P2   200
MAR P3   100
MAR P4   200
;

stored :=
JAN P1    0
JAN P2    0
JAN P3    0
JAN P4    0
FEB P1    0
FEB P2    0
FEB P3    0
FEB P4    0
MAR P1   50
MAR P2   50
MAR P3   50
MAR P4   50
```

```
;

Profit: 11987.418989
```



Rysunek 1: Obraz zbioru rozwiązań efektywnych w przestrzeni ryzyko-zysk

6.2 Wyniki dla modelu dwukryterialnego

6.2.1 Obraz zbioru rozwiązań efektywnych w przestrzeni ryzyko-zysk

Obraz zbioru rozwiązań efektywnch w przestrzeni ryzyko-zysk został uzyskany poprzez rozwiązanie zadania metody punktu odniesienia dla różnych wartości aspiracji dla zysku oraz ryzyka. Do wykonania obliczeń posłużono się skryptem przedstawionym na Listing 7. Obliczenia przeprowadzono ustalając poziomy aspiracji w wyznaczonych granicach zmienności zysku i ryzyka (wektory nadiru i utopii wyznaczone w kolejnej sekcji). Dla każdego poziomu aspiracji wykorzystano po 10 równoodległych wartości znajdujących się w przedziałach definiowanych przez wektory nadiru i utopii.

Ze względu na duży rozmiar zadania, a przez długi czas obliczeń przy 1000 scenariuszach, zdecydowano się ograniczyć ich liczbę do 50. Niestety nie jest to liczba wystarczająca do przeprowadzenia dokładnych obliczeń, jednak uzyskane wyniki powinny być wystarczające do przedstawienia działania metody.

Fragment wyników działania skryptu obliczeniowego przedstawia Listing 8. Obraz zbioru rozwiązań efektywnych w przestrzeni ryzyko-zysk pokazuje Rysunek 1.

Listing 8: Skrypt obliczający wartości do wyznaczenia obrazu zbioru rozwiązań efektywnych w przestrzeni ryzyko-zysk. Pełne wyniki dostępne w załączniku.

```
### 39: Solving model for aspirations: 2395.666667, 312.777778
CPLEX 12.8.0.0: optimal integer solution within mipgap or absmipgap; ↩
    objective 5.427722957e-07
39 MIP simplex iterations
0 branch-and-bound nodes
absmipgap = 7.81092e-07, relmipgap = 1.43881
```

```
Profit: 2417.827519
Risk: 311.250020
RPM: 0.000001
### 40: Solving model for aspirations: 2395.666667, 0.000000
CPLEX 12.8.0.0:
<BREAK> (cplex)
CPLEX solution status 13 with fixed integers:
        aborted in phase II
aborted, integer solution exists; objective -0.06697476417
116486 MIP simplex iterations
103313 branch-and-bound nodes
absmipgap = 6.93933e-05, relmipgap = 0.00103611
Profit: 1432.148909
Risk: 188.510726
RPM: -0.066975
```

Rozwiązania efektywne minimalnego ryzyka i maksymalnego zysku

Rozwiązania efektywne dla minimalnego ryzyka i maksymalnego zysku wyznaczono wykorzystując skrypt przedstawiony na listingu Listing 4. Na podstwaie wyników jego działania, które przedstawia Listing 9 można podać następujące rozwiązania:

- Minimalne ryzyko: $ep = -1000$, przy $r = 0$,
- Maksymalny zysk: $ep = 11987$, przy $r = 2569$

Dodatkowo poza zakresem zadania wyznaczonon pozostałe elementy potrzebne do wyznaczenia wektorów nadiru i utopii:

- Maksymalne ryzyko: $ep = 9193$, przy $r = 2815$,
- Minimalny zysk: $ep = -2400.00$, przy $r = 0.00$

Wektor nadiru: $(-2400, 2815)$

Wektor utopii: $(0, 11987)$

Listing 9: Skrypt wyznaczający rozwiązania optymalne modelu dwukryterialnego.

```
#########################
### Minimizing profit ###
#########################
CPLEX 12.8.0.0: optimal integer solution; objective -2400
7 MIP simplex iterations
0 branch-and-bound nodes
Profit: -2400
Risk: 0

#########################
### Maximizing profit ###
#########################
CPLEX 12.8.0.0: optimal integer solution; objective 11987.41899
32 MIP simplex iterations
0 branch-and-bound nodes
Profit: 11987
Risk: 2569

```

```
########################
### Minimizing risk ###
########################
CPLEX 12.8.0.0: optimal integer solution; objective 0
0 MIP simplex iterations
0 branch-and-bound nodes
Profit: -1000
Risk: 0

########################
### Maximizing risk k###
########################
CPLEX 12.8.0.0: optimal integer solution; objective 2815.995263
21837 MIP simplex iterations
705 branch-and-bound nodes
Profit: 9193
Risk: 2815
```

6.2.2 Analiza relacji dominacji stochastycznej dla trzech wybranych rozwiązań efektywnych

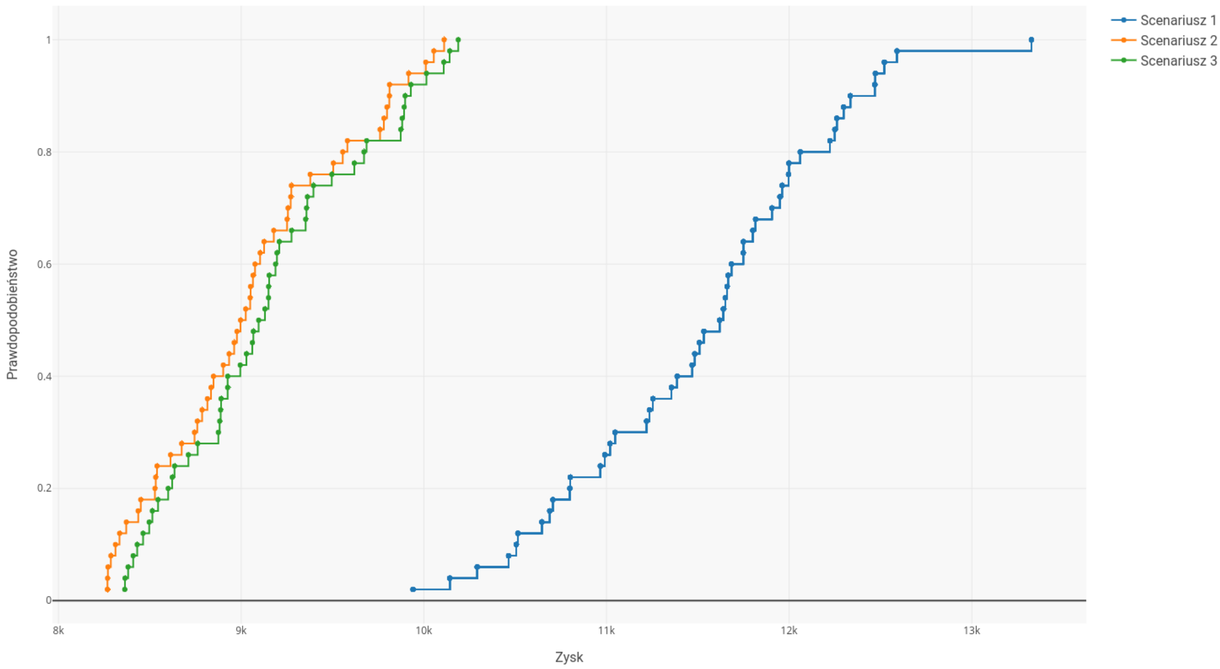
Do analizy wybrano następujące scenariusze:

1. Maksymalny zysk $ep = 11987.42$,
2. Poziomy aspiracji $asp_{ep} = 8789.89$ oraz $asp_r = 1876.67$,
3. Poziomy aspiracji $asp_{ep} = 10388.44$ oraz $asp_r = 938.33$.

Dane do analizy zostały wygenerowane w trakcie przeprowadzania obliczeń do poprzednich podpunktów i są dostępne w załącznikach.

Dystrybuanty zysku przedstawia Rysunek 2.

Na podstawie wykresów możemy stwierdzić, że rozwiązanie dla scenariusza z maksymalnym zyskiem dominuje w sensie FSD pozostałe rozwiązania. Dodatkowo widzimy, że rozwiązanie ze scenariusza 3 dominuje w sensie FSD rozwiązanie scenariusza 2.



Rysunek 2: Wykres dystrybuant zysku dla poszczególnych rozwiązań