

Politechnika Warszawska

Instytut Automatyki i Informatyki Stosowanej

Sprawozdanie z projektu na przedmiot Optymalizacja we Wspomaganiu Decyzji

Jan Adam Kumor, s234694

22 stycznia 2018

Spis treści

1	Treść zadania	2
2	Model programowania mieszanego liniowego-całkowitoliczbowego	3
2.1	Parametry	3
2.2	Zmienne	3
2.3	Ograniczenia	3
2.4	Metoda punktu odniesienia	5
3	Model dla programu AMPL	5
3.1	Plik z modelem (.mod)	5
3.2	Plik z danymi (.mod)	8
3.3	Skrypty uruchomieniowe (.run)	10
4	Symulacja procesu podejmowania decyzji	12

1 Treść zadania

Dane nr 7

Pewne miasto jest zasilane w wodę z trzech ujęć wodnych A, B, C. Badania statystyczne pozwoliły oszacować zapotrzebowanie na wodę na najbliższe 4 godziny. Kształtuje się ono w kolejnych godzinach następująco 1377 ton, 1429 ton, 445 ton i 1712 ton.

Ujęcie wody A może pobrać maksymalnie 896 ton wody w ciągu godziny. Koszt pobrania jednej tony wody wynosi 87zł, jeśli ujęcie pobiera nie więcej niż 716 ton w ciągu godziny i 125zł za każdą tonę powyżej tego progu.

Ujęcie wody B może pobrać maksymalnie 570 ton wody w ciągu godziny. Koszt pobrania jednej tony wody wynosi 93zł, jeśli ujęcie pobiera nie więcej niż 456 ton w ciągu godziny i 193zł za każdą tonę powyżej tego progu.

Ujęcie wody C może pobrać maksymalnie 771 ton wody w ciągu godziny. Koszt pobrania jednej tony wody wynosi 78zł, jeśli ujęcie pobiera nie więcej niż 539 ton w ciągu godziny i 120zł za każdą tonę powyżej tego progu.

W trakcie uzdatniania wody powstają odpady chemiczne. Ilość odpadów uzależniona jest od jakości wody czerpanej przez dane ujęcie. Dla ujęcia pierwszego jest to 300g/tonę, dla ujęcia drugiego 310g/tonę, dla ujęcia trzeciego 270g/tonę odpadów.

Każde ujęcie może być czynne lub nieczynne. Ujęcie nieczynne nie pobiera wody i nie ponosi żadnych kosztów stałych. Ujęcie czynne może pobierać wodę w przewidzianym dla siebie zakresie lub może nie pobierać wody w ogóle. Ujęcie czynne ponosi stałe koszty niezależne od ilości pobieranej wody. Koszty te są różna dla poszczególnych ujęć i wynoszą odpowiednio 9800zł/h, 6600zł/h i 6000zł/h.

Zasady bezpieczeństwa wymagają, aby w każdej godzinie przynajmniej 2 ujęcia były czynne. Ilość pobieranej wody jest ustalana na początku każdej godziny.

Należy opracować plan pracy poszczególnych ujęć wody w kolejnych czterech godzinach horyzontu planowania zapewniający pokrycie programowanego zapotrzebowania na wodę, przy jednoczesnej minimalizacji kosztów oraz ilości wytwarzanych odpadów. Należy stworzyć system wykorzystujący metodę punktu odniesienia do wspomagania podejmowania decyzji w tym zakresie.

1. Sformułować model programowania mieszanego liniowego-całkowitoliczbowego. Model w zapisie matematycznym powinien zostać zawarty w sprawozdaniu z wykonania projektu. Należy zdefiniować i opisać wszystkie zmienne występujące w modelu. Funkcja celu oraz ograniczenia (grupy ograniczeń) muszą zostać dokładnie opisane: funkcja każdego z nich, rola poszczególnych jego składników, itp. Opis modelu musi być czytelny, wyczerpujący i wskazujący na zrozumienie zagadnienia. Sprawdzający powinien na jego podstawie móc ocenić intencje autora.
2. Przygotować model dla programu AMPL i załączyć do sprawozdania.
3. Przeprowadzić symulację procesu podejmowania decyzji. Wyniki kolejnych kroków (przynajmniej 10 kroków) zapisać w sprawozdaniu.

2 Model programowania mieszanego liniowego-całkowitoliczbowego

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

2.1 Parametry

Parametr	Opis
mp_i	Maksymalna produkcja i-tego ujęcia [t]
ks_i	Godzinowy koszt stały pracy i-tego ujęcia wody [zł]
kz_{ij}	Koszt zmienny wydobycia tony wody przez i-te ujęcie, w j-tym przypadku (przypadek zależy od ilości wody pobieranej przez ujęcie) [zł/t]
wz_i	Ilość zanieczyszczeń wytwarzanych przez i-te ujęcie przy pobieraniu wody [g/t]
pr_{ij}	Próg produkcji i-tego ujęcia po przekroczeniu, którego koszt wydobycia odpowiada przypadkowi j+1, dla najwyższego progu arbitralnie wysoka liczba (999999) [t]
zap_k	Zaporzebowanie miasta na wodę w k-tej godzinie [t]
$T = \{1, \dots, n_T\}$	Zbiór reprezentujący kolejne godziny horyzontu planowania produkcji, gdzie $n_T = 4$ - liczba godzin
$I = \{1, \dots, n_I\}$	Zbiór reprezentujący ujęcia wody, gdzie $n_I = 3$ - liczba ujęć
$J = \{1, \dots, n_J\}$	Zbiór reprezentujący zakresy kosztów pracy ujęć wody, gdzie $n_J = 2$ - liczba zakresów

2.2 Zmienne

Zmienna	Opis
a_{ik}	Zmienne binarne określające aktywność i-tego ujęcia w k-tej godzinie. Jeśli ujęcie jest aktywne przyjmują wartość 1, w przeciwnym wypadku 0
p_{ik}	Produkcja i-tego ujęcia w k-tej godzinie (główna zmienna decyzyjna) [t]
uz_{ijk}	Zmienne binarne określające czy i-te ujęcie w k-tej godzinie produkuje wodę w j-tym zakresie kosztów. Jeśli j-ty zakres jest wykorzystywany przyjmują wartość 1, w przeciwnym wypadku 0
pz_{ijk}	Produkcja i-tego ujęcia w k-tej godzinie przypadająca na j-ty zakres kosztów [t]
pg_k	Sumaryczna produkcja wody w k-tej godzinie [t]
ksg_k	Koszt stały pracy ujęć w k-tej godzinie [zł]
kzg_k	Koszt zmienny pracy ujęć w k-tej godzinie [zł]
kg_k	Sumaryczny koszt pracy ujęć w k-tej godzinie [zł]
k	Całkowity koszt pracy ujęć [zł]
z	Całkowita ilość zanieczyszczeń wytworzonych przy pobieraniu wody [g]

2.3 Ograniczenia

Obliczenie sumarycznej produkcji wody w k-tej godzinie:

$$pg_k = \sum_{i \in I} p_{ik}, \quad \forall k \in T$$

Obliczenie kosztu stałego pracy ujęć w k-tej godzinie:

$$ksg_k = \sum_{i \in I} a_{ik} ks_i, \quad \forall k \in T$$

Obliczenie kosztu zmiennego pracy ujęć w k-tej godzinie:

$$kzg_k = \sum_{i \in I} \sum_{j \in J} pz_{ijk} kz_{ij}, \quad \forall k \in T$$

Obliczenie sumarycznego kosztu pracy ujęć w k-tej godzinie:

$$kg_k = ksg_k + kzg_k, \quad \forall k \in T$$

Obliczenie całkowitego kosztu pracy ujęć:

$$k = \sum_{k \in T} kg_k$$

Obliczenie ilości wytworzonych zanieczyszczeń:

$$z = \sum_{k \in T} \sum_{i \in I} p_{ik} wz_i$$

Ograniczenie na minimalną ilość aktywnych ujęć (co najmniej 2 aktywne w każdej godzinie):

$$\sum_{i \in I} a_{ik} \geq 2, \quad \forall k \in T$$

Ograniczenie na produkcję nieaktywnych ujęć (tylko aktywne ujęcia mogą produkować wodę):

$$a_{ik} = 0 \implies p_{ik} = 0$$

Ograniczenie maksymalnej godzinowej produkcji każdego z ujęć:

$$p_{ik} \leq mp_i, \forall i \in I, \quad \forall k \in T$$

Ograniczenie wolumenu produkcji każdego z ujęć przypadającego na dany zakres kosztów produkcji:

$$pz_{ijk} \leq \begin{cases} pr_{ij}, & j = 1 \\ pr_i - pr_{i(j-1)}, & j \in J \setminus \{1\} \end{cases}$$

Ograniczenie odpowiedzialne za ustawienie zmiennych binarnych mówiących o tym czy dany zakres kosztów produkcji w k-tej godzinie jest wykorzystywany:

$$pz_{ijk} \geq \begin{cases} pr_{ij} uz_{ijk}, & j = 1 \\ (pr_{ij} - pr_{i(j-1)}) uz_{ijk}, & j \in J \setminus \{1\} \end{cases}$$

2.4 Metoda punktu odniesienia

Metoda punktu odniesienia wprowadza zestaw dodatkowych parametrów i zmiennych:

Parametr	Opis
asp_k	Poziom aspiracji kosztu produkcji wody
asp_z	Poziom aspiracji ilości wyprodukowanych zanieczyszczeń
λ_k, λ_z	Współczynniki normalizujące, odpowiednio dla kosztu i zanieczyszczeń. W omawianym przypadku preferowany koszt i poziom zanieczyszczeń jest zgodny lub mniejszy od odpowiednich poziomów aspiracji, stąd ze względu na ogólne sformułowanie metody punktu odniesienia jako problemy maksymalizacji, przyjmą one wartości ujemne.
β	Współczynnik pomniejszający wartość ocen wykraczających powyżej poziomu aspiracji
ϵ	Współczynnik składnika regularyzacyjnego
Zmienne	Opis
oc_k, oc_z	Wartości indywidualnych funkcji osiągnięć dla kosztu i zanieczyszczeń
v	Zmienna pomocnicza metody punktu odniesienia

Ograniczenia zmiennej v przez wartości indywidualnych funkcji osiągnięć:

$$v \leq oc_k \quad \text{oraz} \quad v \leq oc_z$$

Ograniczenia indywidualnych funkcji osiągnięć:

$$oc_k \leq \lambda_k(k - asp_k)$$

$$oc_k \leq \beta \lambda_k(k - asp_k)$$

$$oc_z \leq \lambda_z(z - asp_z)$$

$$oc_z \leq \beta \lambda_z(z - asp_z)$$

Funkcja celu metody punktu odniesienia w postaci dla programowania liniowego:

$$\max \quad v + \epsilon(oc_k + oc_z)$$

3 Model dla programu AMPL

3.1 Plik z modelem (.mod)

Listing 1: Model AMPL.

```

1 #####
2 # System wspomagający decyzje w zakresie planowania pracy #
3 # ujec wody. #
4 # MODEL #
5 # Autor: Jan Kumor #
6 #####
7
8 #####
9 # Zbiory #
10 #####

```

```

11
12 # Ujecia wody
13 set INTAKES;
14
15 # Zakresy wolumenu produkcji ujec o roznych kosztach
16 set RANGES ordered;
17
18 #####
19 # Parametry #
20 #####
21
22 # Liczba godzin
23 param T >= 1;
24
25 # Maksymalna wydajnoc ujec wody [t/h]
26 param efficiency {INTAKES} >= 0;
27
28 # Koszta zmienne pracy ujec [zl/t]
29 param variableCost {INTAKES, RANGES} >= 0;
30
31 # Progi zakresow wolumenu produkcji o roznych kosztach [t/h]
32 param threshold {INTAKES, RANGES} >= 0;
33
34 # Koszta stale pracy ujec [zl/h]
35 param fixedCost {INTAKES} >= 0;
36
37 # Iloc wytwarzanych zanieczyszcze [g/t]
38 param pollution {INTAKES} >= 0;
39
40 # Zapotrzebowanie na wode w kolejnych godzinach [t]
41 param hourlyDemand {1..T} >= 0;
42
43 #####
44 # Zmienne #
45 #####
46
47 # Aktywnosc danego ujecia w danej godzinie
48 var active {1..T, INTAKES} binary;
49 # Wolumen produkcji wody danego ujecia w danej godzinie
50 var intakeProduction {1..T, i in INTAKES} >=0, <= efficiency[i];
51 # Godzinowa produkcja
52 var hourlyProduction {1..T} >= 0;
53 # Wolumen produkcji wody przypadajacy na dany zakres kosztow dla ↔
   danego ujecia w danej godzinie
54 var intakeProductionInRange {1..T, i in INTAKES, r in RANGES} >=0;
55 var thresholdExceeded {t in 1..T, i in INTAKES, r in RANGES} binary;
56 # Koszt zmienny poboru wody przez dane ujecie w danej godzinie
57 var intakeVariableCost {1..T, INTAKES} >= 0;
58 # Koszta stale pracy ujec dla danej godziny
59 var hourlyFixedCost {1..T} >= 0;
60 # Koszta zmienne pracy ujec w danej godzinie
61 var hourlyVariableCost {1..T} >= 0;
62 # Koszt pracy ujec dla danej godziny
63 var hourlyCost {1..T} >= 0;
64 # Calkowity koszt pracy ujec wody
65 var totalCost >= 0;

```

```

66 # Calkowita ilosc wytwarzanych zanieczyszcze przy zalozonym ←
    wolumenie produkcji
67 var totalPollution >= 0;
68
69 #####
70 # Ograniczenia modelu #
71 #####
72 # Obliczenie godzinowej produkcji
73 subject to CalculateHourlyProduction {t in 1..T}:
74     hourlyProduction[t] = sum {i in INTAKES} intakeProduction[t,i←
        ];
75 # Ograniczenia zakresow kosztow produkcji
76 subject to Range1 {t in 1..T, i in INTAKES}:
77     intakeProductionInRange[t,i,first(RANGES)] <= threshold[i,←
        first(RANGES)];
78 subject to Range2 {t in 1..T, i in INTAKES, r in RANGES: r != first(←
        RANGES)}:
79     intakeProductionInRange[t,i,r] <= (threshold[i,r]-threshold[i←
        ,prev(r)]) * thresholdExceeded[t,i,prev(r)];
80 subject to Range3 {t in 1..T, i in INTAKES}:
81     threshold[i,first(RANGES)]*thresholdExceeded[t,i,first(RANGES←
        )] <= intakeProductionInRange[t,i,first(RANGES)];
82 subject to Range4 {t in 1..T, i in INTAKES, r in RANGES: r != first(←
        RANGES)}:
83     (threshold[i,r]-threshold[i,prev(r)])*thresholdExceeded[t,i,r←
        ] <= intakeProductionInRange[t,i,r];
84 # Obliczenie wolumenu produkcji dla danego ujecia
85 subject to CalculateIntakeProduction {t in 1..T, i in INTAKES}:
86     intakeProduction [t,i] = sum {r in RANGES} ←
        intakeProductionInRange[t,i,r];
87 # Obliczenie kosztow pracy ujec
88 subject to CalculateHourlyFixedCost {t in 1..T}:
89     hourlyFixedCost[t] = sum {i in INTAKES} fixedCost[i] * active←
        [t,i];
90 subject to CalculateIntakeVariableCost {t in 1..T, i in INTAKES}:
91     intakeVariableCost[t,i] = sum {r in RANGES} ←
        intakeProductionInRange[t,i,r]*variableCost[i,r];
92 subject to CalculateHourlyVariableCost {t in 1..T}:
93     hourlyVariableCost[t] = sum {i in INTAKES} intakeVariableCost←
        [t,i];
94 subject to CalculateHourlyCost {t in 1..T}:
95     hourlyCost[t] = hourlyFixedCost[t] + hourlyVariableCost[t];
96 subject to CalculateTotalCost:
97     totalCost = sum {t in 1..T} hourlyCost[t];
98 # Obliczenie calkowitej liczby wyprodukowanych zanieczyszcze
99 subject to CalculateTotalPollution:
100     totalPollution = sum {t in 1..T, i in INTAKES} ←
        intakeProduction[t,i] * pollution[i];
101 # Ograniczenia wolumenu planowanej produkcji aktywnych ujec
102 subject to InactiveIntakesProduction {t in 1..T, i in INTAKES}:
103     active[t,i] = 0 ==> intakeProduction[t,i] = 0;
104 # Zapewnienie aktywnosci co najmniej dwoch ujec w kazdej godzinie
105 subject to NumberOfActive {t in 1..T}:
106     sum {i in INTAKES} active[t,i] >= 2;
107 # Pokrycie zapotrzebowania na wode w kazdej godzinie
108 subject to HourlyDemand {t in 1..T}:

```

```

109         hourlyProduction[t] = hourlyDemand[t];
110
111     #####
112     # Metoda punktu odniesienia #
113     #####
114     # Skladniki wektora oceny
115     set RATED = {"COST", "PLTN"};
116     # Wektor oceny
117     var value {r in RATED} =
118         if r == "COST" then totalCost
119         else if r == "PLTN" then totalPollution;
120     # Wektor aspiracji
121     param aspiration {RATED};
122     # Wspolczynniki normalizujace - ujemna lambda gdy minimalizujemy ↔
123     # skladnik
124     param lambda {RATED};
125     # Wspolczynnik skladnika regularyzacyjnego
126     param epsilon;
127     # Wspolczynnik pomniejszenia wartosci ocen ponad poziomem aspiracji
128     param beta;
129     # Indywidualne funkcje osiagniec
130     var individualRating {RATED};
131     # Zmienna pomocnicza metody punktu odniesienia
132     var v;
133     # Skalaryzujaca funkcja osiagniecia
134     var rating = v + epsilon * (sum {r in RATED} individualRating[r]);
135     # Odlegloc od punktu odniesienia
136     var distance {r in RATED} = value[r]-aspiration[r];
137     # Znormalizowana odlegloc od punktu odniesienia
138     var normalizedDistance {r in RATED} = lambda[r]*distance[r];
139
140     # Ograniczenia zmiennej v przez indywidualne funkcje osiagniec
141     subject to VSubject {r in RATED}:
142         v <= individualRating[r];
143     # Ograniczenia indywidualnych funkcji osiagniec
144     subject to IndividualRatingSubjectBeta {r in RATED}:
145         individualRating[r] <= beta*normalizedDistance[r];
146     subject to IndividualRatingSubject {r in RATED}:
147         individualRating[r] <= normalizedDistance[r];
148
149     #####
150     # Funkcje celu #
151     #####
152     maximize MaximizeRating: rating;
153     minimize MinimizeCost: totalCost;
154     minimize MinimizePollution: totalPollution;
155     maximize MaximizeCost: totalCost;
156     maximize MaximizePollution: totalPollution

```

3.2 Plik z danymi (.mod)

Listing 2: Dane dla modelu AMPL.

```

1 #####
2 # System wspomagajacy decyzje w zakresie planowania pracy #

```



```

3  # ujec wody. #
4  # DANE #
5  # Autor: Jan Kumor #
6  #####
7
8  # Zbiory indeksowe
9  # Ujecia wody
10 set INTAKES := A B C;
11
12 # Zakresy wolumenu produkcji ujec o roznych kosztach
13 set RANGES := NORMAL HIGH;
14
15 # Liczba godzin
16 param T := 4;
17
18 # Maksymalna wydajnosc ujec wody [t/h]
19 param efficiency :=
20     A      896
21     B      570
22     C      771
23     ;
24 # Koszta zmienne pracy ujec [zl/t]
25 param variableCost:
26     NORMAL  HIGH :=
27     A      87      125
28     B      93      193
29     C      78      120
30     ;
31 # Progi zakresow wolumenu produkcji o roznych kosztach [t/h]
32 param threshold default 999999:
33     NORMAL :=
34     A      716
35     B      456
36     C      539
37     ;
38 # Koszta stale pracy ujec [zl/h]
39 param fixedCost :=
40     A      9800
41     B      6600
42     C      6000
43     ;
44 # Ilosc wytwarzanych zanieczyszcze [g/t]
45 param pollution :=
46     A      300
47     B      310
48     C      270
49     ;
50
51 # Zapotrzebowanie na wode w kolejnych godzinach [t]
52 param hourlyDemand :=
53     1      1377
54     2      1429
55     3      445
56     4      1712
57     ;
58

```

```

59 # Metoda punktu odniesienia
60
61 param epsilon = 0.000025;
62
63 param beta = 0.001;
64
65 param          lambda :=
66     COST      -0.00001115
67     PLTN      -0.00001037
68     ;
69
70 param          aspiration :=
71     COST      582029
72     PLTN      1503070
73     ;

```

3.3 Skrypty uruchomieniowe (.run)

Listing 3: Skrypt wyznaczający wektory utopii i nadiru.

```

1 #####
2 # System wspomagający decyzje w zakresie planowania pracy #
3 # ujec wody. #
4 # SKRYPT URUCHAMIAJACY - Wyznaczenie utopii i nadiru #
5 # Autor: Jan Kumor #
6 #####
7
8 #####
9 # Konfiguracja modelu #
10 #####
11 model OWD.mod;
12 data OWD.dat;
13 option solver cplex;
14
15 #####
16 # Rozwiązania optymalne dla wyznaczenia granic zmiennoci #
17 # tj wektorow zenitu i nadiru #
18 #####
19 # Minimalny koszt
20 printf "#####\n";
21 printf "### Minimizing cost ###\n";
22 printf "#####\n";
23 objective MinimizeCost;
24 solve;
25 printf "Cost: %d\n", totalCost;
26 printf "Pollution: %d\n", totalPollution;
27
28 # Maksymalny koszt
29 printf "#####\n";
30 printf "### Maximizing cost ###\n";
31 printf "#####\n";
32 objective MaximizeCost;
33 solve;
34 printf "Cost: %d\n", totalCost;
35 printf "Pollution: %d\n", totalPollution;

```

```

36
37
38 # Minimalny poziom zanieczyszcze
39 printf "\n#####\n";
40 printf "### Minimizing pollution ###\n";
41 printf "#####\n";
42 objective MinimizePollution;
43 solve;
44 printf "Cost: %d\n", totalCost;
45 printf "Pollution: %d\n", totalPollution;
46
47 # Maksymalny poziom zanieczyszcze
48 printf "\n#####\n";
49 printf "### Maximizing pollution ###\n";
50 printf "#####\n";
51 objective MaximizePollution;
52 solve;
53 printf "Cost: %d\n", totalCost;
54 printf "Pollution: %d\n", totalPollution;

```

Listing 4: Skrypt dokonujący kroku symulacji z wykorzystaniem metody punktu referencyjnego.

```

1 #####
2 # System wspomagający decyzje w zakresie planowania pracy #
3 # ujec wody. #
4 # SKRYPT URUCHAMIAJACY - Metoda punktu odniesienia #
5 # Autor: Jan Kumor #
6 #####
7
8 #####
9 # Konfiguracja modelu #
10 #####
11 model OWD.mod;
12 data OWD.dat;
13 option solver cplex;
14
15 #####
16 # Metoda punktu odniesienia #
17 #####
18 printf "\n#####\n";
19 printf "### Reference point method ###\n";
20 printf "#####\n";
21 objective MaximizeRating;
22 solve;
23 for {t in 1..T} {
24     printf "\n#####\n";
25     printf "### Hour %d ###\n", t;
26     printf "#####\n";
27     printf "Intakes production:\n";
28     for {i in INTAKEs} {
29         printf "- %s ", i;
30         if active[t,i]
31         then printf "(ACTIVE) ";
32         else printf "(INACTIVE) ";
33         printf "%dt (", intakeProduction[t,i];
34     for {r in RANGEs} {

```

```

35         printf "%s: %d ", r, intakeProductionInRange[←
            t,i,r];
36     }
37     printf ")\n";
38 }
39 printf "Demand: %d\n", hourlyDemand[t];
40 printf "Production: %d\n", hourlyProduction[t];
41 printf "Cost:\n - Fixed: %d\n - Variable: %d\n - Total: %d\n"←
    ,
42     hourlyFixedCost[t], hourlyVariableCost[t], hourlyCost←
        [t];
43 }
44 printf "\n#####\n";
45 printf "Total cost: %d\n", totalCost;
46 printf "Pollution produced: %d\n", totalPollution;
47 printf "\n\tValue\tAspiration\tDistance\tLambda\n";
48 for {r in RATEd} {
49     printf "%s\t%d\t%d\t\t%d\t\t%f\n", r, value[r], aspiration[r←
        ], distance[r], lambda[r];
50 }
51
52 printf "Rating: %f\n", MaximizeRating;

```

4 Symulacja procesu podejmowania decyzji

Przed przeprowadzeniem właściwej symulacji wyznaczono maksymalne i minimalne osiągalne wartości kosztu oraz ilości wyprodukowanych zanieczyszczeń (utopia i nadir), które posłużyły do wyznaczenia parametrów λ_k oraz $lambda_z$. W tym celu wykorzystano skrypt przedstawiony na Listing 3. Wyniki działania skryptu przedstawia Listing 5

Listing 5: Wynik działania skryptu wyznaczającego wektory utopii i nadiru.

```

1  #####
2  ### Minimizing cost ###
3  #####
4  CPLEX 12.8.0.0: optimal integer solution; objective 492360
5  4 MIP simplex iterations
6  0 branch-and-bound nodes
7  Cost: 492360
8  Pollution: 1422690
9  #####
10 ### Maximizing cost ###
11 #####
12 CPLEX 12.8.0.0: optimal integer solution; objective 582029
13 8 MIP simplex iterations
14 0 branch-and-bound nodes
15 Cost: 582029
16 Pollution: 1503070
17
18 #####
19 ### Minimizing pollution ###
20 #####
21 CPLEX 12.8.0.0: optimal integer solution; objective 1406610
22 0 MIP simplex iterations

```

```

23 0 branch-and-bound nodes
24 Cost: 532901
25 Pollution: 1406610
26
27 #####
28 ### Maximizing pollution ###
29 #####
30 CPLEX 12.8.0.0: optimal integer solution; objective 1503070
31 1 MIP simplex iterations
32 0 branch-and-bound nodes
33 Cost: 582029
34 Pollution: 1503070

```

Wartości wektora utopii:

$$k^u = 492360 \quad \text{oraz} \quad z^u = 1406610$$

Wartości wektora nadiru:

$$k^n = 582029 \quad \text{oraz} \quad z^n = 1503070$$

Na tej podstawie, wykorzystując zależność $\lambda_i = 1/(y_i^u - y_i^n)$, wyznaczono następujące wartości parametrów:

$$\lambda_k = 1,115 \cdot 10^{-5} \quad \text{oraz} \quad \lambda_z = 1,037 \cdot 10^{-5}$$

Następnie przeprowadzono 10 kroków symulacji dla różnych wartości poziomów aspiracji:

1. Punkt odniesienia: koszt i ilość zanieczyszczeń optymalne:

	Poziom aspiracji	Wartość	Różnica
Koszt	492360	498664	6304
Ilość zanieczyszczeń	1406610	1413388	6778

Wartość funkcji celu: $-0,070298$

Ilość iteracji: 35

2. Punkt odniesienia: koszt optymalny, ilość zanieczyszczeń suboptymalna, niska:

	Poziom aspiracji	Wartość	Różnica
Koszt	492360	493412	1052
Ilość zanieczyszczeń	1420000	1421131	1131

Wartość funkcji celu: $-0,011732$

Ilość iteracji: 38

3. Punkt odniesienia: koszt optymalny, ilość zanieczyszczeń suboptymalna, wysoka:

	Poziom aspiracji	Wartość	Różnica
Koszt	492360	492360	0
Ilość zanieczyszczeń	1470000	1422690	-47310

Wartość funkcji celu: 0.000000

Ilość iteracji: 53

4. Punkt odniesienia: koszt suboptymalny, niski, ilość zanieczyszczeń optymalna:

	Poziom aspiracji	Wartość	Różnica
Koszt	500000	502804	2804
Ilość zanieczyszczeń	1406610	1409625	3015

Wartość funkcji celu: -0.031269

Ilość iteracji: 27

5. Punkt odniesienia: koszt suboptymalny, wysoki, ilość zanieczyszczeń optymalna:

	Poziom aspiracji	Wartość	Różnica
Koszt	550000	509901	-40099
Ilość zanieczyszczeń	1406610	1406610	0

Wartość funkcji celu: 0.000000

Ilość iteracji: 21

6. Punkt odniesienia: koszt i ilość zanieczyszczeń lepsze od stanu utopii:

	Poziom aspiracji	Wartość	Różnica
Koszt	480000	500337	20337
Ilość zanieczyszczeń	1390000	1411867	21867

Wartość funkcji celu: -0.226777

Ilość iteracji: 28

7. Punkt odniesienia: koszt lepszy od utopii, ilość zanieczyszczeń optymalna:

	Poziom aspiracji	Wartość	Różnica
Koszt	480000	493451	13451
Ilość zanieczyszczeń	1406610	1421073	14463

Wartość funkcji celu: -0.149990

Ilość iteracji: 37

8. Punkt odniesienia: koszt optymalny, ilość zanieczyszczeń lepsza od utopii:

	Poziom aspiracji	Wartość	Różnica
Koszt	492360	501995	9635
Ilość zanieczyszczeń	1400000	1410360	10360

Wartość funkcji celu: -0.107442

Ilość iteracji: 27

9. Punkt odniesienia: koszt i ilość zanieczyszczeń suboptymalne, niskie:

	Poziom aspiracji	Wartość	Różnica
Koszt	500500	500862	362
Ilość zanieczyszczeń	1411000	1411390	390

Wartość funkcji celu: -0.004046

Ilość iteracji: 30

10. Punkt odniesienia: koszt i ilość zanieczyszczeń równe wektorowi nadiru:

	Poziom aspiracji	Wartość	Różnica
Koszt	582029	495954	−86075
Ilość zanieczyszczeń	1503070	1432470	−70600

Wartość funkcji celu: 0.000732

Ilość iteracji: 35