

Optimalizacja kosztów w łańcuchu dostaw – implementacja modelu matematycznego

Paweł Sitek, Jarosław Wikarek

Katedra Systemów Sterowania i Zarządzania, Politechnika Świętokrzyska

Streszczenie: W artykule przedstawiono implementację autorzkiego modelu optymalizacji kosztów w łańcuchu dostaw. Model został sformułowany w postaci zagadnienia programowania liniowego całkowitoliczbowego z funkcją celu określającą koszty dystrybutora, producenta oraz transportu. Implementacji dokonano w środowisku pakietu optymalizacji LINGO firmy LINDO Systems Inc. Po dokonaniu implementacji zostały przeprowadzone eksperymenty obliczeniowe dla przykładowych zbiorów danych.

Słowa kluczowe: łańcuch dostaw, programowanie całkowitoliczbowe, optymalizacja

1. Wprowadzenie

W [1] przedstawiono model matematyczny optymalizacji kosztów w łańcuchu dostaw jako zadanie programowania liniowego całkowitoliczbowego (MILP – ang. *Mixed Integer Linear Programming*) [2, 3]. W modelu funkcja celu określa koszty wytwarzania produktów, ich dostawy od producenta do dystrybutora oraz od dystrybutora do odbiorcy, jak również koszty samego dystrybutora. Ograniczenia modelu związane są z czasami realizacji dostaw, zdolnościami produkcyjnymi, pojemnościami, którymi dysponują dystrybutorzy oraz z realizacją ilościową zamówień. Szczegółową dyskusję modelu przeprowadzono w [1].

2. Implementacja modelu

Implementacji modelu dokonano w środowisku pakietu „LINGO” firmy LINDO [4]. Oprogramowanie „LINGO” to wszechstronne narzędzie przeznaczone do budowy i rozwiązywania matematycznych modeli optymalizacji. Pakiet „LINGO” dostarcza w jednym środowisku: języka do budowy modeli optymalizacyjnych, edytora zawierającego wszystkie niezbędne funkcje oraz wbudowanych „solverów” do rozwiązywania modeli optymalizacji. W środowisku „LINGO” można modelować i rozwiązywać liniowe, nieliniowe, kwadratowe, całkowitoliczbowe i stochastyczne problemy optymalizacji. Implementacja w środowisku LINGO wykonywana jest na dwa sposoby.

Pierwszy sposób polega na wpisaniu do edytora „LINGO” modelu w postaci jawnej, czyli pełnej funkcji celu, wszystkich ograniczeń, wartości parametrów itd. Choć jest to sposób intuicyjny i zgodny z postacią standardową programowania liniowego [2], to w praktyce mało przydatny. Wynika to z rozmiarów modeli implementowanych w praktyce. Dla niedużego przykładu przedstawionego

w punkcie 3, liczba zmiennych decyzyjnych wyniosła 196 a ograniczeń 335.

Drugi sposób to wykorzystanie języka modelowania matematycznego systemu „LINGO” stanowiącego integralną część pakietu „LINGO”, którego podstawowe elementy składni przedstawiono w tab. 1. Dla przykładów rzeczywistych, których rozmiary przekraczają kilka zmiennych decyzyjnych, możliwa jest jedynie budowa i implementacja modelu za pomocą języka modelowania (tab. 1, rys. 1). Podstawowe elementy składni języka modelowania matematycznego systemu „LINGO” przedstawiono w tab.1.

Tab. 1. Podstawowe elementy składni języka modelowania matematycznego systemu „LINGO”

Tab. 1. The basic syntax of mathematical modeling language of „LINGO”

Oznaczenie matematyczne	Składnia języka LINGO
<i>Minimum</i>	MIN =
\sum_{jk}	@sum(POTRZEBY(J,K))
<i>Dla każdego j w zbiorze miast</i>	@FOR(miasta(J))
•	*
=	=
<i>X całkowitoliczbowy</i>	@gin(X)
$X \in \{0,1\}$	@bin(X)
<i>Wczytanie wartości współczynników p z pliku dane.ltd</i>	p=@file(dane.ltd)

Model w języku systemu „LINGO” może być zapisany w pliku tekstowym, przy pomocy dowolnego edytora tekstów i powinien mieć standardowe rozszerzenie *.lng, a plik danych *.ltd.

Strukturę modelu tworzą sekcje. Główną sekcją jest sekcja MODEL, która rozpoczyna się słowem MODEL: a kończy słowem END. W obrębie tej sekcji mogą znajdować się inne sekcje. Do najważniejszych, które zostały wyróżnione odpowiednimi słowami kluczowymi należą: sekcja SETS (SET: ENDSETS) oraz DATA (DATA: ENDDATA). W sekcji SETS można definiować typy obiektów prostych, złożonych oraz ich wzajemne związki. W implementowanym modelu przykładami obiektów prostych są typy Produkty, Fabryki itd. oraz złożonych Produkcja, Dystrybucja itd. W tej sekcji przypisywane są poszczególnym typom parametry i zmienne modelu. Sekcja DATA umożliwia inicjowanie lub przypisanie wartości do poszczególnych parametrów modelu. W pakiecie „LINGO” można to robić na dwa sposoby. Bezpośrednio w sekcji umieszczać dane liczbowe albo umieszczać odwołania do plików, gdzie te dane są zawarte. Taki sposób budowy

modelu zapewnia separację danych od modelu właściwego, co jest bardzo ważne, ponieważ zmiana wartości danych, czy nawet ich rozmiaru nie powoduje konieczności wprowadzania zmian do funkcji celu czy ograniczeń. Tę cechę ma wyłącznie model implementowany w postaci niejawniej za pomocą języka programowania matematycznego pakietu „LINGO”.

3. Przykłady liczbowe

Model optymalizacji kosztów w łańcuchu dostaw przedstawiony w [1] został zaimplementowany w środowisku pakietu „LINGO”. Postać niejawną modelu przedstawiono na rys. 1, natomiast obszerne fragmenty modelu w postaci jawnej, automatycznie wygenerowanego przy wykorzystaniu pakietu „LINGO”, są widoczne na rys. 2. Indeksy, parametry oraz zmienne decyzyjne modelu przedstawiono w tab. 2.

Tab. 2. Zestawienie indeksów, parametrów oraz zmiennych decyzyjnych modelu matematycznego optymalizacji

Tab. 2. Summary indices, parameters and decision variables of the mathematical model optimization

Symbol	Opis
Indeksy modelu	
k	indeks produktu (k=1..O)
j	indeks odbiorcy (punktu dostawy, klienta, sklepu) (j=1..M)
i	indeks producenta (fabryki) (i=1..N)
s	indeks dystrybutora (centrum dystrybucji, magazyn logistyczny) (s=1..E)
N	liczba producentów
M	liczba odbiorców
E	liczba dystrybutorów
O	liczba produktów
Parametry modelu	
F _s	koszt stały dystrybutora s (s=1..E)
P _k	przeźren/objętość zajmowana przez produkt k (k=1..O)
V _s	maksymalna pojemność/objętość/przeźren dystrybutora s (s=1..E)
W _{ik}	zdolności produkcyjne fabryki i dla produktu k (i=1..N). (k=1..O)
C _{ik}	koszt wytworzenia produktu k w fabryce i (i=1..N). (k=1..O)
R _{sk}	jeśli dystrybutor s (s=1..E) może dostarczać produkt k (k=1..O) to R _{sk} =1 w przeciwnym przypadku R _{sk} =0
T _{Psk}	czas potrzebny na przygotowanie wysyłki przez dystrybutora s (s=1..E) produktu k (k=1..O)
T _{Cjk}	nieprzekraczalny czas realizacji dostawy do odbiorcy j (j=1..M) produktu k (k=1..O)
Z _{jk}	zapotrzebowanie odbiorcy j (j=1..M) na produkt k (k=1..O)
T _{fi}	czas dostawy od producenta i do dystrybutora s (i=1..N) (s=1..E)
A _{isk}	koszt dostawy od producenta i do dystrybutora s produktu k (i=1..N) (s=1..E) (k=1..O)

T _{m_{sj}}	czas dostawy od dystrybutora s do odbiorcy j (s=1..E) (j=1..M)
G _{s_{jk}}	koszt dostawy od dystrybutora s do odbiorcy j produktu k (s=1..E) (j=1..M) (k=1..O)
Zmienne decyzyjne	
X _{isk}	wielkość dostawy od producenta i do dystrybutora s produktu k
X _{b_{isk}}	jeśli od producenta i do dystrybutora s jest dostarczany produkt k to X _{b_{isk}} =1 w przeciwnym przypadku X _{b_{isk}} =0
Y _{s_{jk}}	wielkość dostawy od dystrybutora s do odbiorcy j produktu k
Y _{b_{s_{jk}}}	jeśli od dystrybutora s do odbiorcy j jest dostarczany produkt k to Y _{b_{s_{jk}}} =1 w przeciwnym przypadku Y _{b_{s_{jk}}} =0
T _s	jeśli dystrybutor s bierze udział w dostawach to T _s =1 jeśli nie to T _s =0
W _{X_{ik}}	wielkość produkcji fabryki i produktu k (i=1..N). (k=1..O)
V _{X_s}	zajęta przestrzeń dystrybutora s (s=1..E)

Optymalizacji dokonano dla dwóch przykładów P1 i P2. W obu przykładach problem dotyczy łańcucha dostaw, w którym jest dwóch producentów (i=1..2), trzech dystrybutorów (s=1..3), czterech odbiorców (j=1..4) oraz pięć typów produktów (k=1..5). Oba przykłady różnią się pojemnością, jaka jest w dyspozycji dystrybutorów. Dane liczbowe dla wszystkich parametrów modelu z tab. 2 zostały przedstawione w tab. 3.

Tab. 3. Zbiór fragmentów tabel z danymi liczbowymi dla przykładów P1 i P2

Tab. 3. The set of parts of data tables for examples P1 and P2

s	F _s	P1 - V _s	P2 - V _s
1	4000	800	1000
2	3000	1000	2000
3	2000	800	1000

j	k	Z _{jk}	T _{Cjk}	j	k	Z _{jk}	T _{Cjk}
1	1	10	10	3	1	0	0
1	2	30	8	3	2	30	8
1	3	20	10	3	3	30	10
1	4	0	0	3	4	30	8
1	5	10	10	3	5	40	10
2	1	30	10	4	1	40	10
2	2	40	8	4	2	0	0
2	3	0	0	4	3	20	10
2	4	30	8	4	4	20	8
2	5	40	10	4	5	20	10

s	k	R _{sk}	T _{Dsk}	s	k	R _{sk}	T _{Dsk}
1	1	1	1	3	1	0	---
1	2	1	1	3	2	1	1
1	3	1	1	3	3	1	1
1	4	1	1	3	4	1	1
1	5	0	---	3	5	1	1
2	1	1	2	2	4	1	2
2	2	1	2	2	5	1	2
2	3	1	2				

i	s	T _{fi_s}
1	1	1
1	2	2
1	3	3
2	1	3
2	2	2
2	3	1

s	j	T _{m_sj}
1	1	1
1	2	1
1	3	2
1	4	2
2	1	1
2	2	1
2	3	1
2	4	1
3	1	2
3	2	2
3	3	1
3	4	1

i	s	k	A _{isk}	i	s	k	A _{isk}
1	1	1	300	2	1	1	---
1	1	2	300	2	1	2	500
1	1	3	400	2	1	3	600
1	1	4	300	2	1	4	500
1	1	5	---	2	1	5	---
1	2	1	400	2	2	1	---
1	2	2	400	2	2	2	400
1	2	3	500	2	2	3	500
1	2	4	400	2	2	4	400
1	2	5	---	2	2	5	400
1	3	1	---	2	3	1	---
1	3	2	500	2	3	2	300
1	3	3	600	2	3	3	400
1	3	4	500	2	3	4	300
1	3	5	---	2	3	5	300

i	k	W _{ik}	C _{ik}	i	k	W _{ik}	C _{ik}
1	1	200	10	2	1	0	---
1	2	200	20	2	2	200	25
1	3	200	30	2	3	200	25
1	4	300	10	2	4	300	10
1	5	0	---	2	5	200	20

k	P _k
1	3
2	2
3	4
4	5
5	6

s	j	k	G _{sjk}	s	j	k	G _{sjk}
1	1	1	200	2	1	1	300
1	1	2	200	2	1	2	300
1	1	3	200	2	1	3	300
1	1	4	200	2	1	4	300
1	1	5	---	2	1	5	300
1	2	1	200	2	2	1	300
1	2	2	200	2	2	2	300
1	2	3	200	2	2	3	300
1	2	4	200	2	2	4	300
1	2	5	---	2	2	5	300
1	3	1	200	2	3	1	300
1	3	2	200	2	3	2	300
1	3	3	200	2	3	3	300
1	3	4	200	2	3	4	300
1	3	5	---	2	3	5	300
1	4	1	200	2	4	1	300
1	4	2	200	2	4	2	300
1	4	3	200	2	4	3	300
1	4	4	200	2	4	4	300
1	4	5	---	2	4	5	300
3	1	1	---	3	3	1	---
3	1	2	200	3	3	2	200
3	1	3	200	3	3	3	200
3	1	4	200	3	3	4	200
3	1	5	200	3	3	5	200
3	2	1	---	3	4	1	---
3	2	2	200	3	4	2	200
3	2	3	200	3	4	3	200
3	2	4	200	3	4	4	200
3	2	5	200	3	4	5	200

```

Model:
Sets:
Produkty /1..@file(rozmiary.ldt)/:P;
Fabryki /1..@file(rozmiary.ldt)/;
Miasta /1..@file(rozmiary.ldt)/;
Punkty /1..@file(rozmiary.ldt)/:F,V,Vx,T;
Produkcja (fabryki,produkty):W,Wx,C;
Potrzeby (miasta,produkty):Z,Tc;
Dystrybucja (punkty,produkty):R,Tp;
dostawy_1 (fabryki,punkty,produkty):X,Xb,A;
dostawy_2 (punkty,miasta,produkty):Y,Yb,G;
dos_to_pun (fabryki,punkty):Tf;
dos_to_mia (punkty,miasta):Tm;
EndSets
Data:
P = @file(dane.ldt);
F = @file(dane.ldt);
V = @file(dane.ldt);
W = @file(dane.ldt);
C = @file(dane.ldt);
Z = @file(dane.ldt);
Tc = @file(dane.ldt);
R = @file(dane.ldt);
Tp= @file(dane.ldt);
Tf= @file(dane.ldt);
A = @file(dane.ldt);
Tm= @file(dane.ldt);
G = @file(dane.ldt);
EndData
Min= @sum(punkty(s):F(s)*T(s)) +
@sum(dostawy_1(i,s,k):A(i,s,k)*X(i,s,k)) +
@sum(dostawy_2(s,j,k):G(s,j,k)*Y(s,j,k)) +
@sum(produkcja(i,k):C(i,k)*
@sum(punkty(s):X(i,s,k)));
@for(produkcja(i,k):
@sum(punkty(s):X(i,s,k)) <=W(i,k);
@sum(punkty(s):X(i,s,k)) =Wx(i,k));
@for(potrzeby(j,k):
@sum(punkty(s):R(s,k)*Y(s,j,k)) >=Z(j,k));
@for(punkty(s): @for(produkty(k):
@sum(fabryki(i):X(i,s,k)) =
@sum(miasta(j):Y(s,j,k))));
@for(punkty(s):
@sum(produkty(k):
p(k)*@sum(fabryki(i):X(i,s,k)))
<=T(s)*V(s);
@sum(produkty(k):
p(k)*@sum(fabryki(i):X(i,s,k))) =Vx(s);
@for(potrzeby(j,k): @for(punkty(s):
@for(fabryki(i)|Z(j,k)#gt#0:
Xb(i,s,k)*Tf(i,s)+Xb(i,s,k)*Tp(s,k)+
Yb(s,j,k)*Tm(s,j)<=Tc(j,k)));
@for(punkty(s):@bin(T(s));
@for(dostawy_1(i,s,k):
@bin(Xb(i,s,k));

X(i,s,k)<=1000*Xb(i,s,k);X(i,s,k)>=Xb(i,s,k));
@for(dostawy_2(s,j,k):
@bin(Yb(s,j,k));

Y(s,j,k)<=1000*Yb(s,j,k);Y(s,j,k)>=Yb(s,j,k));
End

```

Rys. 1. Zawartość pliku scm.lng z modelem optymalizacji kosztów w łańcuchu dostaw [1]

Fig. 1. The contents of file scm.lng (the supply chain cost optimization model [1])

Po implementacji modelu w języku modelowania matematycznego (rys. 1) została uruchomiona optymalizacja. Wyniki optymalizacji są widoczne w okienku wynikowym systemu „LINGO” (rys. 3, 4), gdzie również można odczytać parametry procesu znajdowania rozwiązania optymalnego tzn. liczbę iteracji, wykorzystany algorytm optymalizacji (*Branch-and-Bound*) [3], liczbę zmiennych decyzyj-

nych, w tym całkowitoliczbowych, ograniczeń itd. W procesie optymalizacji znajdowane jest globalne rozwiązanie problemu dla konkretnych danych (tab. 3), które w rozpatrywanym przypadku oznacza najniższy koszt realizacji zapotrzebowań klientów poprzez łańcuch dostaw i wynosi odpowiednio dla P1=255 900, a dla P2=251 900.

```

[ _1] MIN= 4000 * T_1 +...+ 2000 * T_3 + 300 *
X_1_1_1 + ...+ 400 * X_2_3_5 + 200 * Y_1_1_1 +
...+ 200 * Y_3_4_5 +10 * X_1_1_1 +... + 20 *
X_2_3_5
Ograniczenie 1
[ _2] X_1_1_1 + X_1_2_1 + X_1_3_1 <= 200;
...
[ _21] X_2_1_5 + X_2_2_5 + X_2_3_5 - WX_2_5 = 0;
Ograniczenie 2
[ _22] Y_1_1_1 + Y_2_1_1 >= 10;
...
[ _41] Y_2_4_5 + Y_3_4_5 >= 20;
Ograniczenie 3
[ _41] Y_2_4_5 + Y_3_4_5 >= 20;
...
[ _56] - Y_3_1_5 - Y_3_2_5 - Y_3_3_5 - Y_3_4_5 +
X_1_3_5 + X_2_3_5 = 0;
Ograniczenie 4
[ _57] 3 * X_1_1_1 + 2 * X_1_1_2 + 4 * X_1_1_3 +
5 * X_1_1_4 + 6 * X_1_1_5 + 3 *
X_2_1_1 + 2 * X_2_1_2 + 4 * X_2_1_3 + 5 *
X_2_1_4 + 6 * X_2_1_5 - 800 * T_1 <= 0;
...
[ _59] 3 * X_1_3_1 + 2 * X_1_3_2 + 4 * X_1_3_3 +
5 * X_1_3_4 + 6 * X_1_3_5 + 3 *
X_2_3_1 + 2 * X_2_3_2 + 4 * X_2_3_3 + 5 *
X_2_3_4 + 6 * X_2_3_5 - 800 * T_3 <= 0;
Ograniczenie 5
[ _60] YB_1_1_1 + 2 * XB_1_1_1 <= 10;
...
[ _155] YB_3_4_5 + 2 * XB_2_3_5 <= 10;
Ograniczenie 6 - Binarność
[ _156] X_1_1_1 - 10000 * XB_1_1_1 <= 0;
[ _157] X_1_1_1 - XB_1_1_1 >= 0;
[ _158] X_1_1_2 - 10000 * XB_1_1_2 <= 0;
[ _159] X_1_1_2 - XB_1_1_2 >= 0;

[ _334] Y_3_4_5 - 10000 * YB_3_4_5 <= 0;
[ _335] Y_3_4_5 - YB_3_4_5 >= 0;
@BIN( YB_1_1_1); @BIN( YB_1_1_2);
...
@BIN( XB_1_1_1); @BIN( XB_1_1_2);
@BIN( XB_2_3_4); @BIN( XB_2_3_5)
@BIN( T_1); @BIN( T_2); @BIN( T_3);

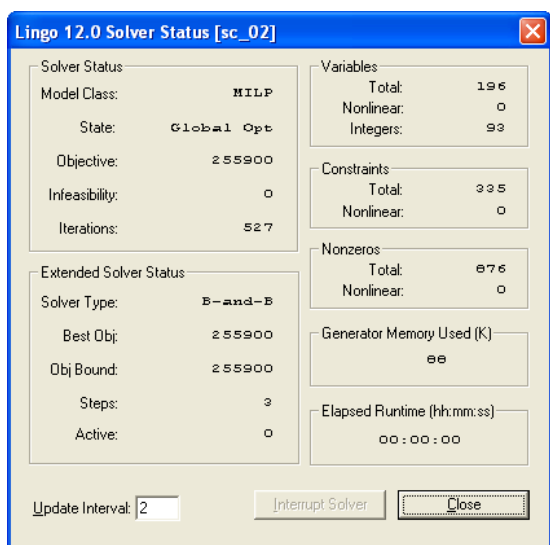
```

Rys. 2. Postać kanoniczna modelu optymalizacji kosztów w łańcuchu dostaw [1]

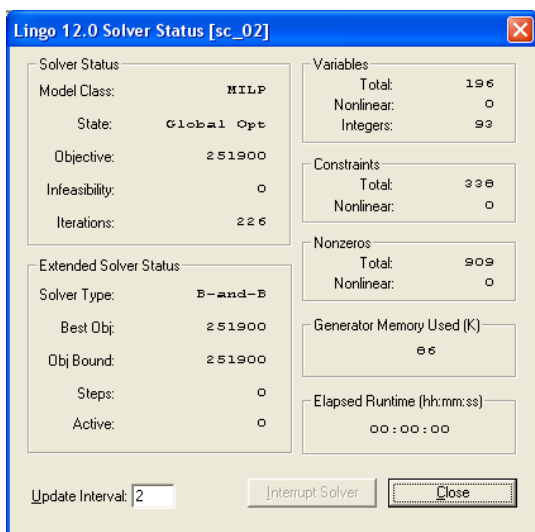
Fig. 2. The canonical form of the supply chain cost optimization model [1]

Jednocześnie wyznaczane są konkretne wartości zmiennych decyzyjnych, które minimalizują ten koszt (tab. 4). Wartości te oznaczają m.in. wielkości dostaw od producenta do dystrybutora wybranych produktów (X_{isk})

oraz wielkości dostaw produktów od konkretnych dystrybutorów do wybranych odbiorców (Y_{sjk}). Innym sposobem zastosowania zaimplementowanego modelu jest określenie wpływu na koszt zmiany parametrów modelu. Można szczegółowo analizować wrażliwość rozwiązania w zależności od parametrów A, G, C, T, V itd. W artykule skupiono się na wpływie parametru V, czyli pojemności/przestrzeni (P1: $V_1=800, V_2=1000, V_3=800$; P2: $V_1=V_2=V_3=1000$), jaką dysponuje dystrybutor na rozwiązanie. Okazało się, że zmiana tego parametru zmniejsza koszt realizacji dostaw P2 ale również wyłącza z dostaw jednego dystrybutora (V_2). Okazało się również, że dalsza zmiana pojemności dystrybutora $s=3$ nie wpływa na wartość funkcji celu. Takich analiz może być wykonanych bardzo wiele. Do dalszych analiz rozszerzono model optymalizacyjny już w fazie implementacji. Wprowadzono pomocnicze zmienne V_{X_s} (wartość oznacza poziom wykorzystania pojemności dystrybutora s) oraz $W_{X_{ik}}$ (poziom wykorzystania zdolności produkcyjnej producenta i dla produktu k).



Rys. 3. Wynik optymalizacji dla P1
Fig. 3. Result of optimization for P1



Rys. 4. Wynik optymalizacji dla P2
Fig. 4. Result of optimization for P2

Tab. 3. Zbiór fragmentów tabel z wynikami dla przykładów P1 i P2

Tab. 3. The set of parts of tables with results for examples P1 and P2

Przykład P1 $F_c = 255\ 900.0$

i	s	k	X_{isk}
1	1	1	80
1	1	2	70
1	1	3	5
1	1	4	80
2	2	5	30
2	3	2	30
2	3	3	65
2	3	5	80

s	V_s	V_{X_s}
1	800	800
2	1000	180
3	800	800

s	j	k	Y_{sjk}
1	1	1	10
1	2	1	30
1	2	2	40
1	2	4	30
1	3	2	30
1	3	4	30
1	4	1	40
1	4	3	5
1	4	4	20
2	1	5	10
2	2	5	20
3	1	2	30
3	1	3	20
3	2	5	20
3	3	3	30
3	3	5	40
3	4	3	15
3	4	5	20

Przykład P2 $F_c = 251\ 900.0$

i	s	k	X_{isk}
1	1	1	80
1	1	2	70
1	1	4	80
2	3	2	30
2	3	3	70
2	3	5	110

s	V_s	V_{X_s}
1	1000	780
2	1000	0
3	1000	1000

s	j	k	Y_{sjk}
1	1	1	10
1	2	1	30
1	2	2	40
1	2	4	30
1	3	2	30
1	3	4	30
1	4	1	40
1	4	4	20
3	1	2	30
3	1	3	20
3	1	5	10
3	2	5	40
3	3	3	30
3	3	5	40
3	4	3	20
3	4	5	20

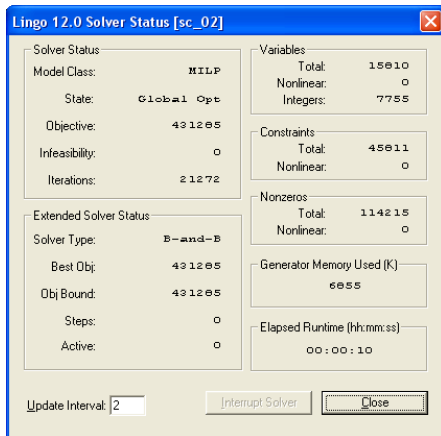
Przykład P1

i	k	W_{ik}	W_{Xik}
1	1	200	80
1	2	200	70
1	3	200	5
1	4	300	80
1	5	0	0
2	1	0	0
2	2	200	30
2	3	200	65
2	4	300	0
2	5	200	110

Przykład P2

i	k	W_{ik}	W_{Xik}
1	1	200	80
1	2	200	70
1	3	200	0
1	4	300	80
1	5	0	0
2	1	0	0
2	2	200	30
2	3	200	70
2	4	300	0
2	5	200	110

Dla przetestowania możliwości wydajnościowych dokonanej implementacji oraz samego modelu dokonano dalszych licznych eksperymentów. Na rys. 5 przedstawiono wynik optymalizacji dla przykładu znacznie większego niż P1 i P2. W przykładzie P3 problem dotyczy łańcucha dostaw, w którym jest sześciu producentów ($i=1..6$), pięciu dystrybutorów ($s=1..5$), dwudziestu pięciu odbiorców ($j=1..25$) oraz pięćdziesiąt typów produktów ($k=1..50$). W tym przykładzie było 15810 zmiennych decyzyjnych, w tym 7755 całkowitoliczbowych oraz 45811 ograniczeń.



Rys.5. Wynik optymalizacji dla P3
Fig.5. Result of optimization for P3

4. WNIOSKI

W pracy przedstawiono implementację modelu optymalizacji kosztów łańcucha dostaw. Sformułowanie modelu w postaci zadania MILP umożliwiło jego rozwiązanie przy wykorzystaniu narzędzia programowania matematycznego "LINGO"¹ [4]. Po implementacji za pomocą języka modelowania matematycznego pakietu „LINGO”, przeprowadzono liczne eksperymenty obliczeniowe. Dwa z nich w postaci przykładów P1 i P2 zostały opisane w artykule. Na podstawie eksperymentów, analiz oraz poprzednich doświadczeń autorów można stwierdzić, że zaproponowany model oraz jego implementacja dają bardzo duże możliwości zastosowań. Po pierwsze, umożliwiają znajdowanie rozkładu przepływów (wartości zmiennych decyzyjnych) dla zamodelowanego łańcucha dostaw, które to minimalizują globalny koszt realizacji zapotrzebowań odbiorców.

¹ Oprogramowanie sfinansowane z projektu MOLAB Program Operacyjny Innowacyjna Gospodarka (Oś 2. Działanie 2.2).

a nawet na strukturę łańcucha dostaw. Przedstawiona w artykule analiza pod kątem jedynie pojemności, jaką dysponują dystrybutorzy, w pełni potwierdza tę tezę.

Bibliografia

1. Sitek P., J.Wikarek *Model matematyczny optymalizacji kosztów w łańcuchu dostaw*, "Pomiary Automatyka Robotyka", Nr 2/2012, xx-xx.
2. Schrijver A.: *Theory of Linear and Integer Programming*. ISBN 0-471-98232-6, John Wiley & sons. 1998.
3. Williams H.P.: *Logic and Integer Programming*. Springer 2009.
4. [www.lindo.com]. ■

Optimization of costs in supply chain – mathematical model implementation

Abstract: The paper presents the implementation of the supply chain cost optimization model. The model was formulated as a linear integer programming problem with objective function specifies the cost of distribution, manufacturing and transportation. Implementation took place in an environment optimization package "LINGO". After the implementation the computational experiments were carried out for sample data sets.

Keywords: optimization, supply chain, mixed integer linear programming

dr inż. Paweł Sitek

Ukończył studia na Wydziale Elektrotechniki i Automatyki Politechniki Świętokrzyskiej w 1991 r. W 2000 r. uzyskał stopień doktora na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej w Gliwicach. Jest autorem lub współautorem ponad 90 artykułów. Główne kierunki badań obejmują optymalizację oraz wspomaganie decyzji dla procesów produkcji, logistyki i dystrybucji przy wykorzystaniu klasycznych MIP (Mixed Integer Programming) oraz deklaratywnych CLP (Constraint Logic Programming) środowisk programowania.

e-mail: sitek@tu.kielce.pl

dr inż. Jarosław Wikarek

Ukończył studia na Wydziale Elektrotechniki i Automatyki Politechniki Świętokrzyskiej w 1994 r. W 2002 r. uzyskał stopień doktora na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej w Gliwicach. Jest autorem lub współautorem ok. 70 artykułów. Główne kierunki badań obejmują optymalizację oraz wspomaganie decyzji dla procesów produkcji, logistyki i dystrybucji przy wykorzystaniu klasycznych MIP (Mixed Integer Programming) oraz deklaratywnych CLP (Constraint Logic Programming) środowisk programowania.

e-mail: j.wikarek@tu.kielce.pl