

ZASTOSOWANIE ŚRODOWISKA MATHEMATICA DO ROZWIĄZANIA PROBLEMU TRANSPORTOWEGO

W artykule zaprezentowano przykład rozwiązania zamkniętego zagadnienia transportowego z wykorzystaniem środowiska programu Mathematica firmy Wolfram Research. Integralną część artykułu stanowi zapis kodu programu umożliwiającego dokonanie optymalizacji planu przewozu na przykładzie transportu kruszywa na place budowy. Opisana w artykule aplikacja pozwala na zamodelowanie zadania transportowego oraz uzyskanie jego optymalnego rozwiązania, minimalizującego koszt dostawy towaru od punktów nadania do punktów odbioru.

WSTĘP

Właściwe zaplanowanie procesu realizacji usługi przewozu powinno uwzględniać aspekt kosztowy. Efektywna realizacja tych zadań jest możliwa tylko poprzez właściwe zarządzanie tym procesem na całej trasie dostawy od dostawcy do odbiorcy. W racjonalizacji kosztów kluczową rolę odgrywa umiejętność optymalizacji warunków realizacji procesu transportowego, czyli innymi słowy rozwiązania problemu transportowego. Rozwiązanie problemu transportowego polega zatem na zaplanowaniu przewozu towaru między dostawcami a odbiorcami, tak aby koszt był jak najniższy, co stanowi rozwiązanie optymalne [2].

W roku 1941 angielski badacz F.L. Hitchcock opracował po raz pierwszy zagadnienie transportowe, określił on problem i opracował algorytm dla planu transportu jednorodnego wyrobu z kilku różnorodnych miejsc zaopatrzenia do kilku punktów zgłaszających popyt na ten wyrób. Celem stawianym w trakcie optymalizacji planu przewozu produktów z punktu A do punktu B, jest najczęściej zmniejszenie kosztów transportu. Ponadto spotyka się także założenia w postaci zmniejszenia odległości, jaką wykonują środki transportowe w sieci przewozowej bądź optymalizowania łącznego czasu transportu produktów [1].

W literaturze jest wiele przykładów metod rozwiązania zagadnienia transportowego. Zagadnienie transportowe można rozwiązywać za pomocą metod analitycznych takich jak metoda simpleks, kąta północno-zachodniego, najmniejszego elementu, VAM oraz metod z wykorzystaniem algorytmów genetycznych [3], metody potencjałów, algorytmu węgierskiego [4].

Celem artykułu jest przedstawienie zastosowania środowiska programu Mathematica do rozwiązania zagadnienia transportowego z wykorzystaniem programowania liniowego. Integralną częścią publikacji jest opracowany skrypt programu (rys.2), który pozwala na rozwiązanie zamkniętego zadania transportowego. Po wprowadzeniu swoich danych użytkownik wykorzystując zaprezentowany kod programu może w szybki sposób otrzymać gotowe rozwiązanie w wygodnej tabelarycznej formie. Przykład zastosowania pakietu Mathematica do rozwiązania zamkniętego zadania transportowego zostanie przedstawiony na przykładzie transportu piasku na place budowy.

1. PRZYKŁAD

Trzy piaskownie zajmujące się wydobyciem i sprzedażą kruszyw zaopatrują w piasek okoliczne inwestycje budowlane. W przykładzie piaskownie zlokalizowane są w trzech różnych miejscach, oznaczone zostały symbolicznie P_1 , P_2 , P_3 , natomiast lokalizacje miejsc, do których należy dostarczyć piasek oznaczono symbolami U_1 , U_2 , U_3 , U_4 , U_5 .

Koszty transportu a_{ij} wyrażają koszt za transport jednej wywrotki piasku (w zł/ wywrotka) z i -tej piaskowni do j -tego placu budowy. Koszt ten jest związany przede wszystkim z odległością pomiędzy poszczególnymi lokalizacjami punktów nadania (P_i) oraz dostawy (U_j). W przykładzie założono, że transport będzie się odbywać wywrotką do przewozu kruszyw o ładowności 15 Mg.

Natomiast podaż piasku, czyli ilość dostępnego piasku wyrażona wielokrotnością ładowności jednej wywrotki, oznaczono symbolem A_i , gdzie indeks i dotyczy lokalizacji poszczególnych piaskowni. Natomiast popyt, czyli zapotrzebowanie na piasek zgłoszone przez kierowników poszczególnych inwestycji budowlanych, w analizowanym przykładzie oznaczono symbolem B_j , gdzie j oznacza lokalizację poszczególnych placów budowy. Analogicznie jest ono wyrażane wielokrotnością ładowności jednej wywrotki. W analizowanym przypadku rozpatruje się zamknięte zadanie transportowe, a zatem sumaryczny popyt jest równy sumarycznej podaży piasku. Tabelaryczne zestawienie powyższych danych wejściowych przedstawiono poniżej (Tab.1).

Tab. 1. Dane do przykładu

a_{ij}		Place budowy					A_i
		U_1	U_2	U_3	U_4	U_5	
Pia- skow- nie	P_1	125	130	150	200	260	19
	P_2	240	180	300	350	120	17
	P_3	250	230	280	370	280	14
B_j		8	13	10	9	10	50

Celem jest opracować plan przewozu piasku z punktów sprzedaży na poszczególne place budowy, tak aby łączny koszt transportu był jak najmniejszy.

Należy zwrócić uwagę, że w analizowanym przypadku sumaryczna podaż (dostępność piasku w piaskowniach) równa jest sumarycznemu popytowi (całkowite zapotrzebowanie na piasek), co można wyrazić za pomocą równości:

$$\sum_{i=1}^3 A_i = \sum_{j=1}^5 B_j \quad (1)$$

Biorąc pod uwagę dane wejściowe, charakteryzujące podaż i popyt zestawione w tabeli 1 oraz wprowadzając zmienną decyzyjną x_{ij} , która oznacza ilość wywrotek piasku, które można przewieźć z poszczególnych piaskowni na place budowy, można zapisać warunki, które muszą zostać spełnione:

$$x_{1,1} + x_{1,2} + x_{1,3} + x_{1,4} + x_{1,5} = 19 \quad (2)$$

$$x_{2,1} + x_{2,2} + x_{2,3} + x_{2,4} + x_{2,5} = 17 \quad (3)$$

$$x_{3,1} + x_{3,2} + x_{3,3} + x_{3,4} + x_{3,5} = 14 \quad (4)$$

oraz:

$$x_{1,1} + x_{2,1} + x_{3,1} = 8 \quad (5)$$

$$x_{1,2} + x_{2,2} + x_{3,2} = 13 \quad (6)$$

$$x_{1,3} + x_{2,3} + x_{3,3} = 10 \quad (7)$$

$$x_{1,4} + x_{2,4} + x_{3,4} = 9 \quad (8)$$

$$x_{1,5} + x_{2,5} + x_{3,5} = 10 \quad (9)$$

Dodatkowo, należy wprowadzić warunek brzegowy w postaci:

$$x_{i,j} \geq 0 \quad (10)$$

Dla tak określonych warunków można zapisać funkcję celu:

$$K(x_{ij}) = \sum_{i=1}^3 \sum_{j=1}^5 a_{i,j} x_{i,j} \rightarrow \min \quad (11)$$

2. ROZWIĄZANIE

Poniżej na rysunku 3 przedstawiono zapis kodu programu umożliwiającego realizację celu, którym było zaplanowanie przewozu piasku z punktów sprzedaży (piaskowni) do punktów odbioru (place budowy).

W celu lepszego zrozumienia proponowanego rozwiązania, każdy wiersz kodu opisano dodatkowym komentarzem, który wyjaśnia znaczenie poszczególnych linii.

Dane wejściowe (podaż piasku w poszczególnych piaskowniach A_i , popyt na piasek w poszczególnych placach budowy B_j oraz koszt przewozu między poszczególnymi punktami a_{ij} , zgodne z tab.1 przedstawia rysunek 1.

$$A_i = \begin{pmatrix} 19 \\ 17 \\ 14 \end{pmatrix}; B_j = \begin{pmatrix} 8 \\ 13 \\ 10 \\ 9 \\ 10 \end{pmatrix}; a_{ij} = \begin{pmatrix} 125 & 130 & 150 & 200 & 260 \\ 240 & 180 & 300 & 350 & 120 \\ 250 & 230 & 280 & 370 & 280 \end{pmatrix}$$

Rys.1. Wektory podaży, popytu oraz jednostkowe koszty transportu między poszczególnymi piaskowniami i placami budowy.

Po uruchomieniu programu wygenerowano rozwiązanie, które przedstawia tabela 2.

Tab.2 Rozwiązanie wygenerowane w wyniku uruchomienia opracowanego skryptu

	Budowa Nr1	Budowa Nr2	Budowa Nr3	Budowa Nr4	Budowa Nr5	Podaż piasku w Piaskowni [liczba wywrotek]
Piaskownia Nr1	0	0	10	9	0	19
Piaskownia Nr2	0	7	0	0	10	17
Piaskownia Nr3	8	6	0	0	0	14
Zapotrzebowanie piasku [liczba wywrotek]	8	13	10	9	10	

Na podstawie otrzymanego rozwiązania plan przewozu powinien kształtować się następująco:

- Z piaskowni nr 1 należy przewieźć 10 wywrotek piasku na plac budowy nr 3 oraz 9 wywrotek piasku na plac budowy nr 4.
- Natomiast z piaskowni nr 2 należy pokryć zapotrzebowanie na piasek zgłoszone przez plac budowy nr 2 w ilości 7 wywrotek oraz przez plac budowy nr 5 w ilości 10 wywrotek.
- Z piaskowni nr 3 należy zaopatrzyć dwa place budowy, nr 1 i 2, odpowiednio w 8 i 6 wywrotek piasku.

Program pozwala także na określenie minimalnego kosztu transportu, który zgodnie z powyższym planem przewozu wynosi łącznie 9140 zł (rys.1).

Solution of TP	Min function
$\sum_{i=1}^m \sum_{j=1}^n a_{ij} x_{ij} = 9140$	

Rys. 1 Wygenerowane rozwiązanie w postaci minimalnego kosztu transportu.

PODSUMOWANIE

W pracy przedstawiono metodykę rozwiązania zbilansowanego zagadnienia transportowego w oparciu o programowanie liniowe z wykorzystaniem oprogramowania Mathematica. Głównym celem publikacji było opracowanie skryptu programu w tym środowisku. Opracowany skrypt może być zaimplementowany także przez innych użytkowników, po wprowadzeniu własnych danych może służyć do znalezienia rozwiązania optymalnego dla innych przykładów praktycznych typowych dla zbilansowanego zadania transportowego.

BIBLIOGRAFIA

1. Filipowicz P., Łucki Z., Badania Operacyjne, AGH, Kraków, 2008
2. Krystek J., Grabalski T., Komputerowe wspomaganie procesu modelowania transportu, MECHANIK 7/2013, s. 323-330
3. Prońko R., Zastosowanie klasycznego algorytmu genetycznego do rozwiązania zbilansowanego zagadnienia transportowego, Studia i Materiały. Miscellanea Oeconomicae Rok 16, Nr 2/2012, s.305-314
4. Józefowska J., Modele i narzędzia optymalizacji w systemach informatycznych zarządzania, Politechnika Poznańska 2010/2011

```

Print[Style["-----", {Bold, 14}]];
(*Dane*)
(*Ilość dostępnych wywoitek w Piaskowni*)
Ai = {19, 17, 14};
(*Ilość wywoitek piasku zamówionych przez poszczególne Place budowy*)
Bj = {8, 13, 10, 9, 10};
(* Macierz → Koszt przewiezienia piasku z Piaskowni na Place budowy*)
aij = {{125, 130, 150, 200, 260}, {240, 180, 300, 350, 120}, {250, 230, 280, 370, 280}};
(*Rozwiązanie zadania*)
(*Długość wektora Ai*)
m = Length[Ai];
(*Długość wektora Bj*)
n = Length[Bj];
(*Drukowanie danych*)
Print["Ai = ", MatrixForm[Ai], "; ", "Bj = ", MatrixForm[Bj], "; ", "aij = ", MatrixForm[aij]]
Print[Style["-----", {Bold, 14}]];
(*Budowa macierzy*)
w1 = Table[If[(i - 1) n + 1 ≤ j ≤ i n, 1, 0], {i, m}, {j, 1, m n}];
macierzS1 = {w1};
w2 = Table[If[Mod[j, n] === Which[k ≤ n - 1, k, k === n, 0], 1, 0], {k, 1, n}, {j, 1, m n}];
macierzS2 = {w2};
A = Partition[Flatten[Append[macierzS1, macierzS2]], 15];
Button["Macierz Zagadnienia Transportowego", CreateDocument[{MatrixForm[A]}]]
Print[Style["-----", {Bold, 14}]];
(*Budowa wektora vc, stworzonego z wektora aij danych*)
vc = Flatten[aij];
vb = Flatten[{Ai, Bj};

vb = Table[{vb[[k]], 0}, {k, Length[vb]}];
(*Rozwiązanie ZT*)
solTP = LinearProgramming[vc, A, vb];
(*Podział rozwiązania ZT (x) na n kawałków*)
solTPM = Partition[solTP, n];
(*Budowa tabeli*)
(*Tworzenie etykiety dla Piaskowni*)
ct = Table[Style[StringForm["Piaskownia Nr`", p], Bold], {p, m}];
(*Tworzenie etykiety dla placów budowy*)
pt = Table[Style[StringForm["Plac budowy`nNr`", w], Bold], {w, n}];
(*Dodanie do powyższej etykiety dwóch etykiet: "Ilość wywoitek piasku w Piaskowni" oraz "Ilość wywoitek piasku Ai" oraz wolnej komórki*)
pt = AppendTo[pt, Text["Podaż \npiasku \n w Piaskowni [liczba wywoitek]"];
pt = AppendTo[pt, Text["Ilość \nwywoitek piasku\n Ai"];
pt = Prepend[pt, ""];
(*Tworzenie wiersza wartości zapotrzebowania budowy wraz z etykietą "Zapotrzebowanie na piasek")
zapotrzebowanie1 = Prepend[Table[Text[Style[Sum[solTPM[[i, j]], {i, m}], Italic, Bold, Red]], {j, n}], Text["Zapotrzebowanie na piasek [liczba wywoitek]"];
(*Tworzenie wiersza wartości zapotrzebowania budowy wraz z etykietą "Zapotrzebowanie Bj")
zapotrzebowanie2 = Prepend[Table[Text[Style[bl[[j]], Italic, Bold, Black]], {j, n}], Text["Zapotrzebowanie Bj"];
(*Tworzenie tabeli wartości ilości wywoitek piasku*)
iloscl = Table[Text[Style[Sum[solTPM[[i, j]], {j, n}], Italic, Bold, Blue]], {i, m}];
(*Tworzenie tabeli wartości ilości wywoitek piasku ai*)
iloscl2 = Table[Text[Style[a[[i]], Italic, Bold, Black]], {i, m}];
(*Przekształcanie wartości rozwiązania x do formy tabeli*)
tabela = {};
Do[tabela = AppendTo[tabela, Part[solTPM, k]], {k, m}];
tabela = Delete[tabela, {1}];
(*Dodawanie do powstałej tabeli kolumny ilość wywoitek piasku w Piaskowni*)
Do[AppendTo[tabela[[k]], ilosc1[[k]], {k, m}];
(*Dodawanie do powstałej tabeli kolumny ilość wywoitek piasku Ai*)
Do[AppendTo[tabela[[k]], ilosc2[[k]], {k, m}];
(*Dodawanie do tabeli kolumny etykiet dla Piaskowni*)
tabela = MapThread[Prepend, {tabela, ct}];
(*Dodatkowo do tabeli wiersza zapotrzebowania budowy na końcu*)
tabela = Insert[tabela, zapotrzebowanie1, -1];
(*Dodatkowo do tabeli wiersza zapotrzebowania Bj na końcu*)
tabela = Insert[tabela, zapotrzebowanie2, -1];
(*Dodawanie do tabeli etykiet Budowa oraz ilości wywoitek piasku w Piaskowni i ilości wywoitek piasku Ai*)
tabela = Insert[tabela, pt, 1];
(*Drukowanie rozwiązania w postaci tabeli*)
TabView[{"Solution of TP" → Grid[tabela, Frame → All, Background → {None, {LightGray}}, ItemStyle → {{Blue, Automatic}, {Blue}}, Alignment → Center],

"Min function " → StringForm["∑i=1m ∑j=1n aijxij = `", solTP.vc]]];

Print[Style["-----", {Bold, 14}]];
sa = Sum[Part[a, q], {q, 1, m}];
sb = Sum[Part[b1, q], {q, 1, n}];
Print["∑i=1m Ai = ", sa, "; ", "∑j=1n Bj = ", sb]
Print[Style["-----", {Bold, 14}]];

```

Rys. 2. Skrypt programu umożliwiającego rozwiązanie zbilansowanego zadania transportowego w środowisku Mathematica

Mathematica environment application to solve the problem of transport

This article presents an example of solution the closed transportation problem with the using of the Mathematica from Wolfram Research. An integral part of the article is a program code in Mathematica to permit an optimization plan for transport on the example of the transport of aggregates for construction sites.

Autorzy:

dr inż. Barbara Miłaszewicz- Politechnika Opolska, Wydział Inżynierii Produkcji i Logistyki

mgr inż. Marcin Maleska – Politechnika Opolska, Wydział Elektrotechniki, Informatyki i Automatyki