FOLIA POMERANAE UNIVERSITATIS TECHNOLOGIAE STETINENSIS
Folia Pomer. Univ. Technol. Stetin. 2010, Oeconomica 282 (60), 71–76

Anna Landowska

ZASTOSOWANIE PROGRAMOWANIA DYNAMICZNEGO DO ROZWIĄZANIA PROBLEMU OPTYMALNEGO PRZYDZIAŁU SUROWCA

APPLICATION OF DYNAMIC PROGRAMMING FOR SOLVING AN OPTIMAL ALLOTMENT PROBLEM

Katedra Zastosowań Matematyki w Ekonomii, Zachodniopomorski Uniwersytet Technologiczny w Szczecinie
ul. Klemensa Janickiego 31, 71-270 Szczecin, e-mail: alandowska@zut.edu.pl

Abstract. The article presents an application of dynamic programming method for solving an optimal allotment problem. On the basis of the presented example, authors showed that the dynamic programming method founded on Bellman's equation finds an optimal solution very quickly. Finding an optimal solution of any given problem using as few iterations as possible, is a very important problem in numerical calculation. The obtained results of optimization are compared with different methods.

Słowa kluczowe: metody optymalizacyjne, problem optymalnego przydziału, programowanie dynamiczne.
Key words: dynamic programming, optimal distribution problem, optimization methods.

WSTĘP

Programowanie dynamiczne (PD) jako metoda rozwiązywania pewnych zadań optymalizacji zostało opracowane w latach 50. ub. wieku przez Bellmana (1952). Metodę tę znacznie rozwinięto, w związku z czym można więcej problemów rozwiązać za jej pomocą.

W artykule przedstawiono zagadnienia w następującym porządku: Przedstawiono zasadę optymalności Bellmana oraz klasyfikację problemów rozwiązywanych za pomocą programowania dynamicznego. Następnie skupiono się na problemie optymalnego przydziału i opisano równanie funkcyjne programowania dynamicznego. W dalszej części przedstawiono przykład, który rozwiązano dwiema metodami, oraz podano wnioski końcowe.

W pracy wykazano, że metoda programowania dynamicznego generuje szybsze rozwiązanie oraz przy mniejszym nakładzie pamięci niż metoda, która przeszukuje całą przestrzeń rozwiązań. Istotną kwestią jest znalezienie metody, za pomocą której otrzymamy rozwiązanie w jak najkrótszym czasie.

PROBLEMY ROWZIĄZYWANE ZA POMOCĄ PROGRAMOWANIA DYNAMICZNEGO

Programowanie dynamiczne jest stosowane w wielu dyscyplinach nauki: w matematyce, inżynierii, ekonomii, medycynie, fizyce, a także w sztucznej inteligencji, systemach informacyjnych i wielu innych (Findeisen i in. 1972, Lew i Mauch 2007).

Teoretyczną podstawą, na której opiera się metoda PD, jest zasada optymalności: „Optymalna strategia sterowania ma tę własność, że jakikolwiek by był stan początkowy i decyzja początkowa, to następne decyzje muszą tworzyć optymalną strategię sterowania względem stanu wynikającego z pierwszej" (Bellman 1965, s. 504).

Fundamentalną cechą wszystkich problemów dających się rozwiązać za pomocą równań Bellmana jest rozdzielność funkcji celu w czasie. Zatem funkcja celu musi być sumą wkładów z kolejnych okresów. Przy czym wkład w każdym okresie zależy tylko od stanu, w jakim proces się znajduje i wybranego sterowania. Korzyścią wynikającą z zastosowania programowania dynamicznego jest to, że za jego pomocą można rozwiązać zadania o dużej liczbie zmiennych decyzyjnych, z nieciągłymi lub nieróżniczkowalnymi funkcjami celu oraz zadania niebędące zadaniami programowania wypukłego (Stadnicki 2006). Problematyka związana z programowaniem dynamicznym została szczegółowo opisana m.in. w pracach Findeisena i in. (1972) oraz Grabowskiego (1980).

Lew i Mauch (2007) w uporządkowany sposób przedstawiają grupy problemów, do rozwiązania których stosowane jest programowanie dynamiczne:

– problem drogi w grafie (ang. *graph routing problems*) – kojarzony ze znalezieniem najkrótszej (najdłuższej) ścieżki w grafie;

– problem porządku – kolejności (ang. *sequencing problems*) – wymagający znalezienia optymalnej kolejności lub permutacji obiektów w zbiorze;

– problem selekcji – wyboru (ang. *selection problems*) – polegający na wyborze optymalnego (zazwyczaj właściwego) podzbioru ze zbioru elementów;

– problem podziału lub grupowania (ang. *partitioning or clustering problems*) – gdy dany zbiór obiektów o mocy N jest dzielony na podzbiory, zwane partycjami, w sposób optymalny;

– problem dystrybucji (ang. *distribution problems*) – gdy dany zbiór obiektów jest rozprowadzany od dostawców do odbiorców w sposób optymalny; pod uwagę brany jest odpowiedni koszt przewozu (dostawy);

– problem produkcji, zapasów, zastępowania (ang. *production or inventory or replacement problems*) – w przypadku surowca, którego jednostki mogą być produkowane (dostawcy) lub na które jest zapotrzebowanie i które są konsumowane (odbiorcy), lub w przypadku każdego z etapów pośrednich; każdego z etapów pośrednich; każdy z tych etapów związany jest z kosztem, np. z kosztem produkcji lub kosztem zakupu maszyn do produkcji itp.;

– problem optymalnego drzewa binarnego (ang. *optimal binary tree problems*) – kojarzony z sytuacją, w której na podstawie decyzji oryginalny problem dzieli się na dwa podproblemy rozwiązywane oddzielnie w taki sposób, aby na podstawie rozwiązań podproblemów można było otrzymać rozwiązanie problemu głównego;

– problem probabilistyczny (ang. *probabilistic problem*) – każdy następny stan, ze zbioru rozwiązań alternatywnych, który obiekt może osiągnąć, jest zdeterminowany przez zaistnienie prawdopodobieństwa;

– problem probabilistyczny – w przypadku jednej klasy problemów, w których kolejnym etapom przypisane są wagi;

– problem dotyczący obróbki ciągów (ang. *problems involving string processing*);

– problemy nieoptymalizowane (ang. *nonopotimization problems*) – ujęcie w kontekście sekwencyjnych problemów decyzyjnych tam, gdzie nie są podjęte decyzje optymalizacyjne dotyczące pewnych funkcji obiektu.

PROBLEM OPTYMALNEGO PRZYDZIAŁU

Problem optymalnego przydziału (ang. *Optimal Allotment Problem*) polega na podjęciu decyzji dotyczącej sposobu dostarczenia danej ilości surowca jego użytkownikom, przy określonym koszcie lub zysku związanym z przydziałem jednostek produktu do użytkowników, odbiorców.

Załóżmy, że danych jest w sumie M jednostek surowca i niech $C(k,d)$ oznacza koszt lub zysk odnoszący się do przydziału d jednostek do użytkownika k, gdzie $d=0,...,M$, $k=1,...,N$. Przypuśćmy, że d_1 oznacza przydział jednostek dla użytkownika 1 w ilości d_1; dalej liczba jednostek d_2 została przydzielona użytkownikowi 2 itd. Zdefiniujmy stan (k, m) jako pozostałą liczbę m jednostek produktu w etapie k. Koszt decyzji o przydziale d jednostek w etapie k użytkownikowi k wynosi $C(k, d)$, natomiast następny stan to $(k+1, m-d)$ Równanie funkcyjne programowania dynamicznego (równanie Bellmana) ma postać:

$$f(k,m) = \min_{d \in (0,,...,m)} \{C(k,d) + f(k+1, m-d)\} \tag{1}$$

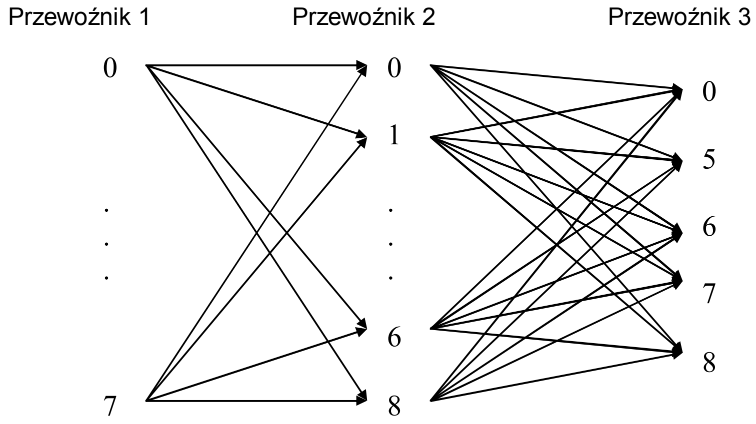
Celem jest znalezienie wartości $f(1,M)$ przy warunku początkowym $f(N+1,m)=0$, gdzie $m \geq 0$. Jeżeli $d > m$ (zamiast ograniczenia d przez M), możemy przyjąć dodatkowy warunek początkowy $f(N+1,m)=\infty$, gdzie $m<0$, aby zapobiec przydzieleniu większej ilości surowca niż jest dostępny (Lew i Mauch 2007).

W dalszej części artykułu rozwiążemy problem optymalnego wyboru przewoźników pewnego surowca. Usługa przewozu jest obarczona kosztem, natomiast koszt jest uzależniony od tonażu surowca; przewoźnicy oferują różne ceny. Problem ten zostanie rozwiązany za pomocą dwóch metod – metody wprost oraz programowania dynamicznego (problem optymalnego przydziału).

Przykład. Kopalnia chce dostarczyć 15 t surowca do fabryki. Obecnie na rynku znajduje się trzech przewoźników; każdy z nich proponuje koszt przewozu uzależniony od tonażu surowca (tab. 1). Kopalnia może złożyć tylko po jednym zamówieniu u każdego z przewoźników. Problem polega na wyborze takiego zamówienia, aby koszt przewozu był optymalny.

Najniższy koszt przewozu 15 t surowca wynosi 6600 zł, gdy przewoźnik 1 przewiezie 5 t surowca, przewoźnik 2 – 2 t, a przewoźnik 3 – 8 t.

Metoda wprost umożliwia poprawne rozwiązanie po rozpatrzeniu 240 przypadków. W tej metodzie analizowana jest każda możliwość złożenia zamówienia przewozu, nawet taka, w której suma przewożonych towarów jest mniejsza od 15 t. Poniżej przedstawiono sieci obrazujące sposób działania metodą wprost (rys. 1).



Rys. 1. Sieć przedstawiająca liczbę porównań w metodzie wprost

Metoda wykorzystująca programowanie dynamiczne umożliwia poprawne rozwiązanie po 152 porównaniach, co oznacza koszt transportu 6600 zł dla przewoźników 1, 2, 3 oraz przewóz surowca o wadze odpowiednio 5, 2, 8 t. W rozpatrywanym przykładzie dwie zmienne równania (1):

$$f(k,m) = \min_{d \in (0,,\ldots,m)} \{C(k,d) + f(k+1, m-d)\} \tag{1}$$

mają następującą interpretację:

k – numer przewoźnika, $k = 3,2,1$;

m – pozostała liczba ton surowca do przewozu;

$C(k,d)$ – koszt przewozu d ton surowca przez przewoźnika k w etapie k.

Optymalne rozwiązanie to wartość minimalna $f(1,15)$. Rozwiązywanie zaczynamy od obliczenia kosztu przewozu pozostałej ilości towaru przez przewoźnika 3. Zatem korzystając z równania (1) dla $k = 3$ oraz $k = 2$, otrzymujemy następujące funkcje:

$$f(3,m) = \begin{cases} 0 & \text{dla} \quad m = 0 \\ 25 & \text{dla} \quad 1 \le m \le 5 \\ 27 & \text{dla} \quad m = 6 \\ 30 & \text{dla} \quad m = 7 \\ 34 & \text{dla} \quad m = 8 \\ \infty & \text{dla} \quad m \ge 9 \end{cases}, \qquad f(2,m) = \begin{cases} 0 & \text{dla} \quad m = 0 \\ 5 & \text{dla} \quad m = 1 \\ 8 & \text{dla} \quad m = 2 \\ 15 & \text{dla} \quad m = 3 \\ 25 & \text{dla} \quad 4 \le m \le 5 \\ 27 & \text{dla} \quad m = 6 \\ 30 & \text{dla} \quad m = 7 \\ 34 & \text{dla} \quad m = 8 \\ 38 & \text{dla} \quad m = 9 \\ 42 & \text{dla} \quad m = 10 \\ 49 & \text{dla} \quad m = 11 \\ 57 & \text{dla} \quad m = 12 \\ 60 & \text{dla} \quad m = 13 \\ 64 & \text{dla} \quad m = 14 \\ 68 & \text{dla} \quad m = 15 \end{cases}$$

Przy czym wpisujemy symbol „∞”, jeżeli nasza decyzja prowadzi do łącznej ilości przewozu surowca powyżej wymaganych 15 t.

Zatem minimalna wartość funkcji $f(1,15)$ wynosi 66 – na podstawie równania Bellmana (1). Dla potwierdzenia obliczmy wartość:

$$\begin{aligned} f(1,15) = \min\{ & C(1,0) + f(2,15),\ C(1,1) + f(2,14),\ C(1,2) + f(2,13), \\ & C(1,3) + f(2,12),\ C(1,4) + f(2,11),\ C(1,5) + f(2,10),\ C(1,6) + f(2,9), \\ & C(1,7) + f(2,8),\ C(1,8) + f(2,7), \ldots,\ C(1,15) + f(2,0)\} = \\ & = \min\{68,70,69,71,68,66,67,67,\infty,\ldots,\infty\} = 66. \end{aligned}$$

Tabela 1. Problem transportowy

Przewoźnik k	Waga towaru d [t]	Koszt przewozu $C(k,d)$ [setki zł]
1	0 1 2 3 4 5 6 7	0 6 9 14 19 24 29 33
2	0 1 2 3 6 8	0 5 8 15 30 38
3	0 5 6 7 8	0 25 27 30 34

W tabeli przedstawiono dane umowne.

WNIOSKI

Z przedstawionego przykładu wynika, że programowanie dynamiczne daje optymalne rozwiązanie zadanego problemu przydziału. Programowanie dynamiczne poprzez wybór, w każdym z etapów procesu rozwiązywania optymalnych wartości, wymaga mniejszej liczby porównań (iteracji) niż metoda oparta na znalezieniu wszystkich możliwości i wybraniu rozwiązania optymalnego. Istotne jest, aby stosować takie metody, za pomocą których otrzymamy rozwiązanie optymalne, wykonując mniej obliczeń.

Na podstawie przeprowadzonych badań możemy stwierdzić, że PD jest szybką i dokładną metodą rozwiązywania poszczególnych zadań optymalizacji.

PIŚMIENNICTWO

Bellman R. 1952. On the theory of dynamic programming, in: Proceedings of the National Academy of Sciences. Santa Monica, California, USA, The RAND Corporation, 716–719.

Bellman R. 1967. Programowanie dynamiczne. Warszawa, PWN.

Findeisen W., Szymanowski J., Wierzbicki A. 1972. Teoria i metody obliczeniowe optymalizacji. Warszawa, PWN.

Grabowski W. 1980. Programowanie matematyczne. Warszawa, PWE.

Lew A., Mauch H. 2007. Dynamic programming. Berlin, Springer-Verlag.

Mauch H. 2004. A petri net representation for dynamic programming problems in management applications, in: Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04), vol. 3. Hawaii, Big Island, 30 072.

Robinett R.D., Wilson D.G. Eisler G.R., Hurtado J.E. 2005. Applied dynamic programming for optimizations of dynamical system. Philadelphia, SIAM.

Stadnicki J. 2006. Teoria i praktyka rozwiązywania zadań optymalizacji. Warszawa, WNT.

Szymczak C. 1998. Elementy teorii projektowania. Warszawa, PWN.