Badania operacyjne: Wykład
Zastosowanie kolorowania grafów w planowaniu produkcji typu no-idle

Paweł Szołtysek

12 czerwca 2008

Streszczenie

Planowanie produkcji jest jednym z problemów optymalizacji dyskretnej, należącym do dziedziny badań operacyjnych. Ta praca formułuje problem dla produkcji typu no-idle, gdzie niedopuszczalne są przestoje, przedstawia go w formie sformalizowanej, prezentuje autorskie podejście do problemu, które poszerza zastosowanie kolorowania grafów w planowaniu, oraz na podstawie implementacji na prostym przypadku przedstawia działanie tego algorytmu. Wyniki uzyskane dla omównionego przypadku okazały się być rozwiązaniem optymalnym.

1 Sformułowanie problemu

Weźmy pod uwagę zakład produkcyjny, który wytwarza pewne produkty za pomocą różnych maszyn.
O produktach tych będziemy zakładać, że:

- na ich wytworzenie składa się szereg niezależnych operacji, wykonywanych przez różne maszyny;
- wszystkie te operacje trwają tyle samo (lub mogą być podzielone na takie niezależne operacje, które będą trwały tyle samo);
- tak maszyny jak i produkty są systemami typu no-idle, ich wyłączenie lub spowodowanie przestoju jest niemożliwe (nieopłacalne).

Dopuszczamy więc fakt, że przedmioty będą się różniły między sobą w zakresie potrzebnych do ich wytworzenia maszyn.

2 Opis formalny problemu

Dany mamy pewien zbiór maszyn M i zbiór produktów P, które chcemy wytworzyć. Daną mamy także relację $\alpha : P \times M \to \mathbb{N}$ pomiędzy tymi zbiorami, określającą ile razy maszyna M będzie używana do wytworzenia produktu P. Naszym zadaniem jest zaproponowanie takiej organizacji pracy, która pozwoli na przeprowadzenie procesu produkcji zgodnie z ograniczeniami danymi w sekcji 1. Funkcją celu będzie więc funkcja zliczająca pojawienie się przestojów w produkcji.

$$F = min \sum_{v=0}^{V} \sum_{Cmin}^{Cmax} \begin{cases} 0 & \text{jesli kolor jest wykorzystany} \\ 1 & \text{wpp} \end{cases} \tag{1}$$

gdzie $V = M \cup P$ reprezentuje sumę produktów i maszyn, $Cmin$ oraz $Cmax$ to odpowiednio oznaczony minimalny i maksymalny kolor danego obiektu.
Poza tym, będziemy dążyli też do minimalizacji ilości użytych kolorów podczas całego procesu.
Problem należy do klasy NP-trudnych.

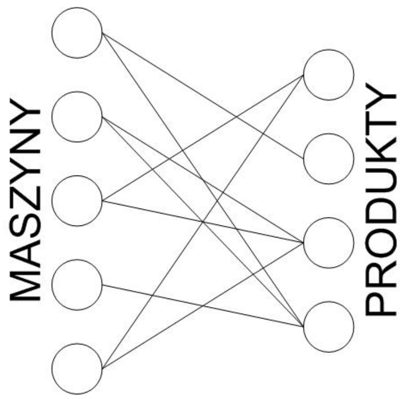
3 Proponowane algorytmy rozwiązania

3.1 Przegląd zupełny

Metodą, która z pewnością zwraca optymalne rozwiązanie takiego problemu, jest przegląd zupełny. Polega on na skonstruowaniu wszystkich możliwych uszeregowań, wyliczeniu dla nich funkcji celu i wybraniu najlepszej. Jest to jednak algorytm bardzo złożony obliczeniowo. Wyznaczenie ogromnej ilości rozwiązań i wyliczenie i porównanie funkcji celu jest czasochłonne. Może on być wykorzystywany tylko do planowania produkcji przy niewielkiej ilości maszyn i produktów (rzędu kilka).

3.2 Zwarte kolorowanie krawędziowe grafu

Problem można przedstawić w postaci grafu, gdzie wierzchołkami będą odpowiednio produkty i maszyny, a krawędzie (może być ich wiele) będą zgodne z funkcją α. Taki graf będzie dwudzielny, nieincydentymi ze sobą wierzchołkami będą odpowiednio te reprezentujące produkty i maszyny.
Postawione zadanie wymaga nadania etykiet na krawędzie, które będą odpowiednio określały kiedy dane zadanie ma być wykonane, tj. kiedy pewna operacja na maszynie i produkcie ma zostać wykonana.



Rysunek 1: Grafowa reprezentacja zadania planowania produkcji.

Przez wzgląd na funkcję celu, naturalnym zastosowaniem wydaje się być pomysł dokonania tego kolorowania w taki sposób, aby zachować zwartość, czyli ciągłość kolorów przy jednym wierzchołku w danym zakresie.
Należy jednak pamiętać, że nie wszystkie sytuacje można pokolorować w ten sposób.

3.2.1 Algorytm zwartego kolorowania grafu

Algorytm przedstawiony poniżej jest autorską interpretacją i rozszerzeniem algorytmu kolorowania zwartego drzew przedstawionego w [1]. Różnica polega na zastosowaniu omówionego tam podejścia do zwartego kolorowania struktur drzewowych z cyklami. Ponadto poniższy algorytm akceptuje i rozwiązuje też problem dla wielokrotnych wystąpień połączeń między wierzchołkami.

1. Utwórz tymczasowy graf $G'(V', E') = G(V, E)$.

2. Jeśli jest cokolwiek w kolejce, dodaj to do G', chyba, że było wrzucone na stos w ostatniej iteracji.

3. Usuń z grafu G' wolne krawędzie.

4. Jeśli E' jest pusty, to sprowadziliśmy zadanie do kolorowania drzewa. Wystarczy pokolorować kolejne ścieżki w dowolnej kolejności przy zachowaniu odpowiednich ograniczeń nałożonych przez zwartość.
   Jeśli E' jest niepusty, to w grafie są cykle. Znajdź cykl.

5. Dokonaj analizy już pokolorowanych krawędzi i zaproponuj zwarte pokolorowanie do obecnie już istniejącego.

Jeśli ograniczenie jest na conajwyżej jednym wierzchołku, koloruj z uwzględnieniem istniejącego koloru tego wierzchołka.
Jeśli ograniczenie jest na większej ilości wierzchołków, podziel cykl na ścieżki pomiędzy nimi i koloruj je zwarcie z istniejącym pokolorowaniem. Jeśli jest to niemożliwe, przekaż ścieżkę do kolejki.

6. Usuń z grafu G' krawędzie które zostały pokolorowane (do innej struktury, wynikowej) lub przekazane do kolejki.

7. Wróć do kroku 2.

Jeśli przez kilka kolejnych iteracji nie będzie dokonywane przyporządkowanie, oznaczać to będzie, że nie znaleziono rozwiązania optymalnego - co jednak nie oznacza, że takiego rozwiązania nie ma. Jest to bowiem algorytm heurystyczny.
Znalezione rozwiązanie można polepszać zmieniając kolejność kolorowania grafów, lub, jeśli jest to akceptowalne, dodać pozostałe krawędzie kolorując je optymalnie, ale nie zwarcie.

4 Przykład

Przedstawione podejście przeanalizujemy na przykładzie.
Załóżmy, że mamy sześć maszyn ($M = 6$), cztery produkty do wytworzenia ($P = 4$), a zapotrzebowanie maszyn na poszczególne produkty (α) przedstawia się następująco:

α	0	1	2	3
0	1	0	0	1
1	0	0	1	0
2	1	1	0	0
3	1	0	0	1
4	0	0	1	0
5	0	0	0	1

W celu rozwiązania tego problemu zaimplementowany w języku C++ został algorytm przedstawiony w sekcji 3.2.1.
Kolorowanie według niego przebiega następująco.

	0	1	2	3
0	4	X	X	5
1	X	X	NA	X
2	NA	NA	X	X
3	5	X	X	4
4	X	X	NA	X
5	X	X	X	NA

	0	1	2	3
0	4	X	X	5
1	X	X	4	X
2	NA	NA	X	X
3	5	X	X	4
4	X	X	5	X
5	X	X	X	NA

	0	1	2	3
0	4	X	X	5
1	X	X	4	X
2	NA	NA	X	X
3	5	X	X	4
4	X	X	5	X
5	X	X	X	6

I w ostatniej iteracji otrzymujemy ostateczne pokolorowanie:

	0	1	2	3
0	4	X	X	5
1	X	X	4	X
2	6	5	X	X
3	5	X	X	4
4	X	X	5	X
5	X	X	X	6

Otrzymany wynik spełnia założenia i ograniczenia, które zostały postawione na początku pracy. Funkcja celu wynosi 0. Ponadto wykorzystaliśmy optymalną liczbę kolorów (trzy).

5 Podsumowanie

Przedstawiony został prosty przykład w celu zobrazowania procesu działania autorskiego algorytmu. Wprawdzie znalezienie rozwiązania zadania wyżej

przedstawionego nie jest większym problemem nawet na kartce, to w rzeczywistości przy nieco większym rozmiarze algorytm radzi sobie bez większego problemu, podczas gdy powyższa metoda znacznie ten proces strukturalizuje i upraszcza, dzięki swoim logicznym podstawom.
W znacznej większości przypadków algorytm znajduje rozwiązanie bliskie optymalnemu.
Przedstawiony powyżej algorytm można rozwinąć o elementy bardziej trafnego wybierania cykli czy ścieżek które pozostały do pokolorowania. Przełoży się to bezpośrednio na jakość wyniku zwracanego przez algorytm.

Literatura

[1] Marek Kubale: *Optymalizacja dyskretna: modele i metody kolorowania grafów. Zwarte kolorowanie grafów*, Wydawnictwa Naukowo-Techniczne, Warszawa 2002, 167-179