Politechnika Wrocławska
Wydział Informatyki i Telekomunikacji

Kierunek: **INA**
Specjalność: **-**

Praca Dyplomowa Inżynierska

Algorytm OPT + 1 dla problemu cięcia belek

Adam Niezgoda
NR INDEKSU: 254623

Opiekun pracy
dr Maciej Gębala

problem optymalizacyjny, solver liniowy, algorytm aproksymacyjny

Wrocław 2022

Streszczenie

Abstract

Tutaj treść streszczenia po angielsku.

Spis treści

Spis rysunków

Spis tabel

Wstęp

Rozdział 1

Analiza problemu

W tym rozdziale scharakteryzowany zostanie problem cięcia belek (ang. Cutting Stock Problem), rozważany przez autora, w dalszej części oznaczony jako PCB. Zarysowane też zostaną podstawy algorytmów używanych do jego rozwiazania.

1.1 Problem cięcia belek

Jest to problem znalezienia takiego rozkładu elementów na belkach, z których owe elementy będa wycinane, tak aby zminimalizować straty materiału. W niniejszej pracy autor skupia się na problemie jednowymiarowym, minmalizowana jest liczba belek, z których są wycinane elementy i są one tej samej długości (β),a liczba rodzajów elementów (d) jest stała. Istnieja też inne jego warianty. Można rozpatrywać problem dwu, trzy - wymiarowy, przyjąć różne długości belek, jak również skupić sie na tym, aby resztki na pojedyńczych belkach były, jak najmniejsze.
Jest to problem optymalizacyjny - liczbę żużytych belek można wyrazić za pomocą funkcji celu (całkowitej w przypadku, gdy instancja problemu nie przewiduje możliwości dzielenia elementów), i pragniemy ją zminimalizować. Wynik optymalny, w tym wypadku, to taka liczba zużytych w rozwiązaniu belek, że już niemożliwe byłoby wycięcie wszystkich elementów z liczby o jeden mniejszej. Z punktu widzenia złożoności obliczeniowej, problem należy do klasy problemów silnie NP-trudnych. W przeszłości konstruowano algorytmy dające wynik optymalny, które działy w czasie mniejszym bądź równym wielomianowemu, ale działo się to dla przypadku małego d, bądź dawały wynik optymalny powiększony o funkcję od d tj. np. $OPT + O(log^2(d))$.[4]

1.2 Sformułowanie problemu liniowego całkowitoliczbowego

Przyjmijmy nastepujące oznaczenia: zbiór rodzajów elementów: $\mathcal{T} = \{T_1, T_2, \ldots, T_n\}$, każdy rodzaj T_i z przypisaną pozytywną długością całkowitą $p_i \leqslant \beta$. Zbiór wszystkich elementów: $|\sigma| = n$, w którym występuje n_i elementów typu T_i, tj. $\sum_{T_i \in \mathcal{T}} n_i = n$. Konfiguracja C_i jest zbiorem elementów o sumie długości, co najwyżej β (długość belki). C_i może mieć postać d-wymiarowego wektora $C_i = \langle a(C_i, 1), a(C_i,2), \ldots, a(C_i,d)\rangle$, w którym każda j-ta pozycja, $a(C_i)$ mówi o liczbie elementów długości p_j w C_i. Niech $\mathcal{C}$ będzie zbiorem wszystkich konfiguracji, którego moc wyniesie maksymalnie n^d, co oczywiście nie zdarzy sie często, gdyż niektóre elementy są na tyle duże, że kolejne już nie zmieszczą się do tej samej belki. Wtedy problem cięcia belek może zostać sformułowany w następujący sposób:

$$
\begin{aligned}
minm = \sum_{C_i \in \mathcal{C}} x_{C_i},& \\
\text{tak, że} \sum_{C_i \in \mathcal{C}} a(C_i,j) x_{C_i} \geqslant n_j, dla j = 1, \ldots d,& \\
x_{C_i} \in \mathbb{Z}_{\geqslant 0}, \text{dla każdego } C_i \in \mathcal{C}&
\end{aligned}
$$

W tym sformułowaniu n_j oznacza liczbę wszystkich elementów długości p_j. Zmienna x_{C_i} mówi o liczbie belek przechowujących obiekty zgodnie z konfiguracją C_i.

1.3 Bin Packing

Analogicznym problemem do PCB jest problem pakowania koszy (ang. Bin Packing) dalej wspominany jako BP. Może być rozpatrywany w kontekście optymalizacji. W tym przypadku, elementy różnych rozmiarów musza być upakowane w **skończoną** (tutaj różnica w stosunku do PCB) liczbę koszy, każdy zadanej, stałej długości, w taki sposób który zminimalizuje ich zużytą liczbę. Patrząc na niego w kategorii problemu decyzyjnego, pytaniem jest czy zadane elementy zmieszczą się w podaną na wejściu problemu liczbę koszy. W przypadku zdefiniowania BP w programowaniu liniowym całkowitoliczbowym, w przeciwieństwie do PCB, zmienne całkowitoliczbowego nie reprezentują ile razy dana konfiguracja została uzyta, ale: 1. czy dany kosz został użyty oraz 2. czy element o indeksie odpowiadającemu zmiennej został włożony do kosza o odpowadajacym indeksie.

Tu brakuje opisu zmiennych, ale czy ja w ogóle mam to opisywać, skoro to paste z wiki?

$$
\begin{aligned}
min\ K = & \sum_{j=1}^{n} y_j,\ K \geqslant 1 \\
& \sum_{i \in I}^{n} s(i) x_{ij} \leqslant B y_j, && \forall j \in \{1, \ldots, n\} \\
& \sum_{j=1}^{n} x_{ij} = 1, && \forall i \in I \\
& y_j \in \{0,1\}, && \forall j \in \{1, \ldots, n\} \\
& x_{ij} \in \{0,1\}, && \forall i \in I \forall j \in \{1, \ldots, n\}
\end{aligned}
$$

Można też na problem popatrzeć poprzez pryzmat liczb wymiernych: $B = 1 \wedge \forall i \in I : s(i) \in \mathbb{Q} \cap (0, 1]$

1.4 Algorytm aproksymacyjny

Jednym ze posobów na poradzenie sobie z brakiem algorytmów wielomianowych dla problemów NP-zupełnych, obok stosowania algorytmów z wykładniczym czasem działania dla małych danych wejściowych lub izolowania specjalnych przypadków dla których znamy algorytmy wielomianowe, jest próba znalezenia takich algorytmów, które, w czasie wielomianowym, dzadzą rozwiązania bliskie optymalnego. Często takie rozwiązanie powinno dawać satysfakcjonujące wyniki. Algorytm, który zwraca rozwiązania bliskie optymalnego, nazywamy ***algorytmem aproksymacyjnym***.
Do określenia dokładności wyników takich algorytmów w rozwiązywaniu problemów optymalizacyjnych, używa się pojęcia współczynnika aproksymacji. Mówimy, że algorytm ma współczynnik aproksymacji $\rho(n)$, gdy dla każdych danych wejściowych rozmiaru n, koszt C rozwiązania uzyskanego przez algorytm mieści się we współczynniku $\rho(n)$ kosztu C^* optymalnego rozwiązania. Tak dla problemu minimalizacji zależność wygląda następująco: $0 < C^* \leqslant C$. C/C^* jest współczynnikiem razy ile koszt rozwiązania przybliżonego jest większy od kosztu rozwiązania optymalnego. Analogicznie się to ma do przypadku problemu maksymalizacji: $0 < C \leqslant C^*$. Współczynnik to: C^*/C.[2, Rozdział 35]

Dla BP istnieje kilka algorytmów aproksymacyjnych:
W 1980 pokazano, że dla uogólnionego d-wymiarowego Bin Packing, każdy algorytm o złożoności $o(nlogn)$ musi mieć współczynnik aproksymacji $\geqslant d$.[7]
W 1994 wykazano, że heurystyki FFD i BFD charakteryzują się absolutnym współczynnikiem 1.5, który zarazem jest najlepszym możliwym dla PCB, dopóki nie zostanie udowodnione $P = NP$.[5]

Skrót	Nazwa angielska	Współczynnik przybliżenia
NF	Next Fit	2
FF	First Fit	1.7
BF	Best Fit	1.7
NFD	Next Fit Decreasing	1.691
FFD	First Fit Decreasing	11/9
BFD	Best Fit Decreasing	11/9
H_M	Harmonic	1.691

Tablica 1.1: Tabela algorytmów aproksymacyjnych - na podstawie artykułu "A note on the approximability of cutting stock problems"

Według autorów pracy, "A note on the approximability of cutting stock problems", owe algorytmy po pewnej konwersji mogą zostać użyte do rozwiązywania instancji Problemu Cięcia Belek. Stawiają oni tezę, że przy zastosowaniu zaprezentwanych konwersji, również istnieje Asymptotic Polynomial Time Approximation Scheme (APTAS) dla jednowymiarowego PCB.[3] Najbardziej znanym takim schematem jest ten zaprezentowany przez Fernandeza de la Vegę i Luekera, który zwaraca rozwiązania o koszcie mniejszym lub równym: $(1+\epsilon)OPT(L)+1$, gdzie $OPT(L)$ to koszt optymalny.[6]

W niniejszej pracy do dalszych rozważań i implementacji wybrany został First Fit Decreasing.

1.4.1 First Fit Decreasing

Wycinanie elementów z belki, będzie tu traktowane jako dokładanie elementów do kosza o pojemności równej długości belki. Algorytm jest opisany następująco:

Algorithm 1.1: First Fit Decreasing - pseudokod

Data: Lista elementów różnej długości
Result: Upakowanie - ułożenie elementów w koszach tak, aby suma rozmiarów elementów w każdym koszy była równa, co najwyżej pojemności kosza

```
1 Posortowanie elementów w kolejności nierosnącej;
2 Otworzenie nowego kosza;
3 foreach elementu z posortowanej listy, znajdź pierwszy kosz do którego mieści się aktualny element do
4     if znalazł się taki kosz then
5         dołóż element do tego kosza
6     else
7         Otwór nowy kosz i dołóż do niego element
```

1.5 Algorytm OPT+1

Tutaj opiszę, że dzieli on te elementy na małe i duże, zajmuje sie duzymi tak jak wyżej opisane, potem pakuje małe i te które mieszczą się frakcyjnie są trakowane first-fitem. Nie zaimplementowałem tego algorytmu, bo używa algorytmu Lenstry, który do rozwiązania DLP powstałego przy uzyciu tego algorytmu, wykorzystuje pewną wersję algorytmu Ellipsoid.

1.5.1 Idea i działanie

1.5.2 Modyfikacje

Rozdział 2

Projekt systemu

2.1 Parametry wejściowe

2.2 Diagram przepływu

Rozdział 3

Implementacja systemu

3.1 Opis technologii

Do implementacji systemu użyto języka C w wersji C17 / python w wersji 3.9 i możliwego do wywołania z poziomu tych języków *callable library* [1]. Napisał bym więcej, ale wciąż pracuję nad kodem.

3.2 Solvery liniowe - APIs

3.3 Omówienie kodów źródłowych

Rozdział 4

Instalacja i wdrożenie

Tu opiszę wymagania jakie wersje języka, jak zbudować kod źródłowy, zainstalować solvery itd.

Rozdział 5

Analiza wyników

5.1 Dokładność rozwiązań

5.2 Czas działania

Podsumowanie

Możliwe, że algorytm aproksymacyjny dla pewnych przypadków nie będzie, aż tak źle wyglądał na tle tego OPT+1, więc będzie odpowiedź na ile warto męczyć się z implementacją tego drugiego plus właśnie czy brutefore kiedyś skończy działanie ...

Bibliografia

[1] Glpk callable libraries.

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. *Introduction to Algorithms.* Massachusetts Institute of Technology, 2009.

[3] Y. W. E. X. G.F. Cintra, F.K. Miyazawa. A note on the approximability of cutting stock problems. 2011.

[4] K. Jansen, R. Solis-Oba. A polynomial time opt + 1 algorithm for the cutting stock problem with a constant number of object lengths. 2011.

[5] D. Simchi-Levi. New worst-case results for the bin-packing problem. 1994.

[6] G. S. L. W. Fernandez de la Vega. Bin packing can be solved within $1 + \epsilon$ in linear time. 1981.

[7] A. C.-C. Yao. New algorithms for bin packing. 1980.

Załącznik A

Zawartość płyty CD

W tym rozdziale należy krótko omówić zawartość dołączonej płyty CD.