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. 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 najdłuższe, na późniejsze wycinki itp.
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. Dopóki nie zostanie udowodnione $P = NP$, nie istnieje dla niego algorytm aproksymacyjny ze wspołczynnikiem mniejszym niż 3/2. 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. $OPT + O(log^2(d))$.[2]

1.2 Sformułowanie problemu liniowego całkowitoliczbowego

Przyjmijmy nastepujące oznaczenia: zbiór rodzajów elementów: $T = \{T_1, T_2, \ldots, T_n\}$, każdy rodzaj T_i z przypisaną pozytywną długością całkowitą p_i .

1.3 Algorytm OPT+1

1.3.1 Idea i działanie

1.3.2 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] K. Jansen, R. Solis-Oba.

Załącznik A

Zawartość płyty CD

W tym rozdziale należy krótko omówić zawartość dołączonej płyty CD.