

Optymalizacja. Algorytmy dokładne

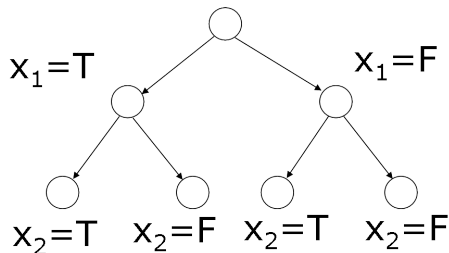
dr hab. inż. Maciej Komosiński, prof. PP

Instytut Informatyki
Politechnika Poznańska
www.cs.put.poznan.pl/mkomosinski

Organizacja zbioru rozwiązań w problemie SAT

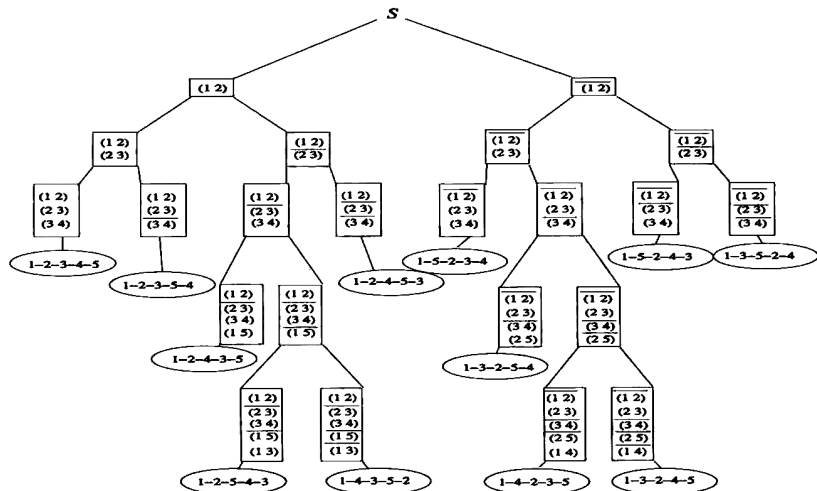
Organizacja zbioru rozwiązań w problemie SAT

Wielokrotny podział na dwa podzbiory: $x_1 = T, x_1 = F, \dots$

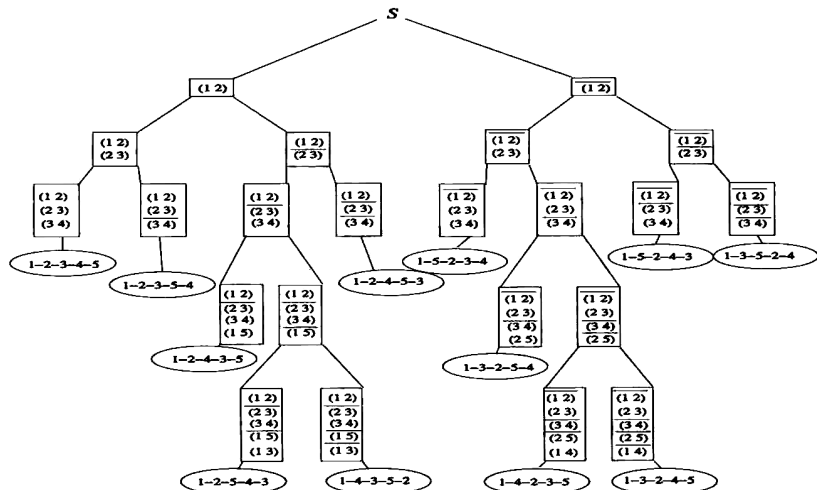


Organizacja zbioru rozwiązań w problemie TSP

Organizacja zbioru rozwiązań w problemie TSP



Organizacja zbioru rozwiązań w problemie TSP



(Dla B&B lepsza by była bardziej naturalna organizacja zbioru: drzewo niebinarne, węzły = kolejne miasta...)

Jeśli w drzewie umieścimy rozwiązania niekompletne
(częściowe)...

Czy można jakoś ograniczyć przeglądanie takiego drzewa?

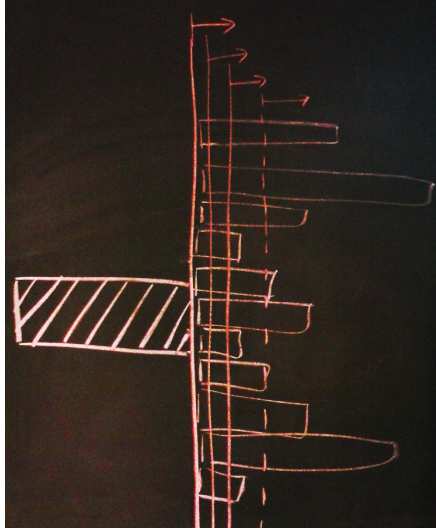
Jeśli w drzewie umieścimy rozwiązania niekompletne (częściowe)...

Czy można jakoś ograniczyć przeglądanie takiego drzewa?

- Tak – można nie przeglądać gałęzi prowadzących tylko do rozwiązań niedopuszczalnych
- Tak – można nie przeglądać gałęzi, w których na pewno nie znajdziemy rozwiązania lepszego od już znalezionego

Ilustracja z wykładu – coraz lepsze ograniczenie

Czerwona przerywana linia to złe ograniczenie – trasy mogą być krótsze



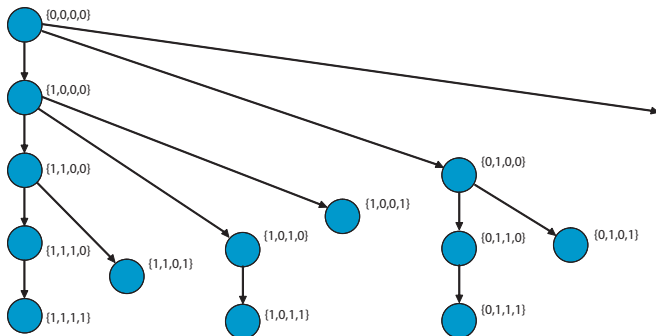
Problem plecakowy – zapis matematyczny

Problem plecakowy – zapis matematyczny

- Zmienne
 - $x_i \in \{0, 1\}$ – 1 jeżeli element i jest umieszczany w plecaku
- Maksymalizuj
 - $\sum_i x_i c_i$
- Przy ograniczeniu:
 - $\sum_i x_i w_i \leq W$
- Problem plecakowy jest NP-trudny

Metoda podziału i ograniczeń – Branch & Bound

Konstruowanie rozwiązania jest widziane jak przeglądanie drzewa.
Np. dla problemu plecakowego:



Górna (dolna) granica – upper (lower) bound

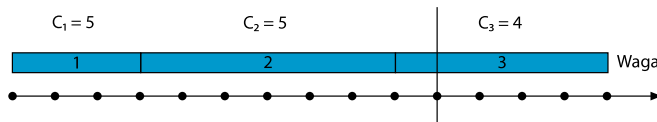
- Górna granica – wartość, której na pewno nie przekroczy funkcja celu dla żadnego z rozwiązań, w których ustalono pewne elementy (np. pewne zmienne decyzyjne)
- Np. jaka jest górna granica dla rozwiązań, w których $x_1 = 1$ i $x_5 = 1$?
- Górna granica nie musi być, i z reguły nie jest, osiągalna przez żadne rozwiązanie
- Znajdowanie górnej granicy jest zależne od problemu

Znajdowanie górnej granicy dla problemu plecakowego

- Załóżmy, że do plecaka możemy wkładać części elementów.
- Posortuj elementy według stosunku c_i/w_i .
- Wkładaj do plecaka elementy rozpoczynając od elementu o największym stosunku c_i/w_i aż do zapełnienia plecaka.
- Ostatni element może zostać włożony częściowo.

Przykład. Pojemność plecaka = 10

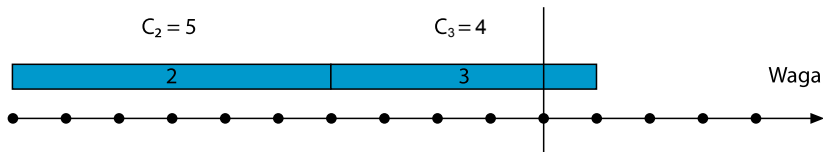
Element i	w_i	c_i	Stosunek c_i/w_i
1	3	5	1.666667
2	6	5	0.833333
3	5	4	0.8
4	2	1.5	0.75
	3	1	0.333333
	4	1	0.25



$$\text{Górna granica} = 5 + 5 + \frac{1}{5} \cdot 4 = 10.8$$

Znajdowanie górnej granicy dla problemu plecakowego

- Jeżeli pewne części rozwiązania są już ustalone (elementy są wybrane – 1, lub nie – 0), to rozpatrujemy tylko pozostałe elementy
- Np. jeśli zdecydowaliśmy, że nie umieszczamy elementu 1, a element 2 jest już w plecaku i mamy 4 jednostki wolne:

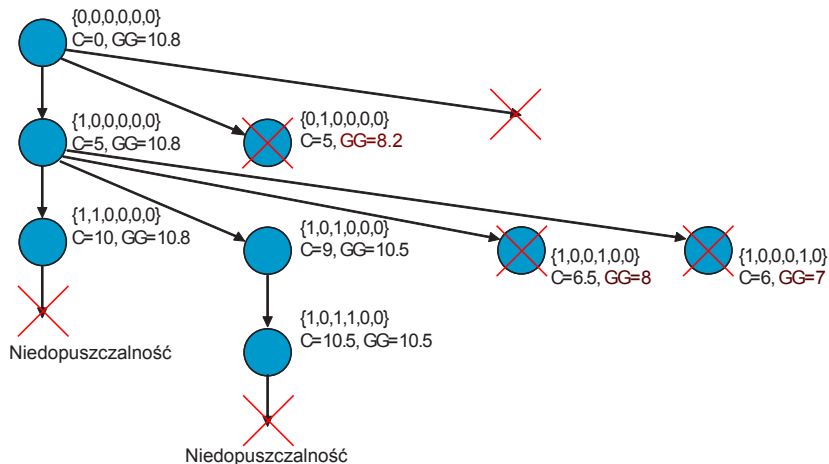


$$\text{Górna granica} = 5 + \frac{4}{5} \cdot 4 = 8.2$$

Metoda podziału i ograniczeń przy maksymalizacji

Jeżeli dla danej gałęzi drzewa górna granica („w tej gałęzi będzie co najwyżej tak dobrze” czyli „nie będzie lepiej niż”) jest niższa od znanego już rozwiązania, to gałąź tę można pominąć.

B&B – przykład (problem plecakowy)



B&B – rozwiązywanie całkowitoliczbowego PM

- Jeśli z problemu całkowitoliczbowego PM usuniemy ograniczenie na to, że zmienne muszą być całkowitoliczbowe, możemy polepszyć rozwiązanie optymalne lub go nie zmienić
- Zatem rozwiązanie optymalne PM ciągłego jest niegorsze od rozwiązania optymalnego PM dyskretnego
- Zatem rozwiązanie optymalne PM ciągłego jest ograniczeniem („nie będzie lepiej niż”) rozwiązania optymalnego PM dyskretnego
- W korzeniu rozwiązujemy PM ciągłe, a w kolejnych poziomach drzewa dodajemy sukcesywnie ograniczenia na kolejne zmienne decyzyjne (*) tak, by wykluczyć przedział zawierający wartość niecałkowitą zmiennej
- W każdym węźle rozwiązujemy PM ciągłe z dodanymi ograniczeniami (*) po to, by wyznaczyć granicę, i ewentualnie pominąć gałąź jeśli ta granica jest gorsza niż najlepsze znalezione już rozwiązanie całkowitoliczbowe
- Przykład: <https://www.youtube.com/watch?v=WNRrmXZkRi0>

Branch & Bound – produkcja

Znajdź przydział, który maksymalizuje łączną wydajność.

Wydajność c_{ij} :

Stanowisko \ Wyrób	1	2	3	4
	1	2	3	4
A	2	10	9	7
B	15	4	14	8
C	13	14	16	11
D	4	15	13	9

- 1 Sformułuj ogólne zadanie jako PM
- 2 Zaproponuj algorytm jego rozwiązania metodą B&B
- 3 Zastosuj go do powyższej instancji problemu.

Programowanie dynamiczne

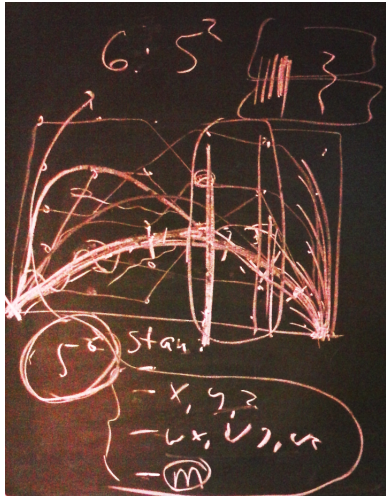
- PD jest stosowane do wieloetapowych procesów decyzyjnych z własnością Markowa
- Katarzyna Jakowska-Suwalska. Programowanie dynamiczne, przykłady i zadania (online doc).
- WPD: $n + 1$ stanów (s_i), n decyzji (x_i), $s_{i+1} = T_i(s_i, x_i)$
- WPD: $s_1 \rightarrow d_1 \rightarrow s_2 \rightarrow d_2 \rightarrow s_3 \rightarrow \dots \rightarrow s_n \rightarrow d_n \rightarrow s_{n+1}$
- Na każdym etapie procesu, będąc w stanie $s_i \in S_i$, podejmujemy decyzję d_i ze zbioru decyzji D_i
- Strategia to ciąg decyzji $d_1, d_2, \dots, d_n$

Programowanie dynamiczne

- PD jest stosowane do wieloetapowych procesów decyzyjnych z własnością Markowa
- Katarzyna Jakowska-Suwalska. Programowanie dynamiczne, przykłady i zadania (online doc).
- WPD: $n + 1$ stanów (s_i), n decyzji (x_i), $s_{i+1} = T_i(s_i, x_i)$
- WPD: $s_1 \rightarrow d_1 \rightarrow s_2 \rightarrow d_2 \rightarrow s_3 \rightarrow \dots \rightarrow s_n \rightarrow d_n \rightarrow s_{n+1}$
- Na każdym etapie procesu, będąc w stanie $s_i \in S_i$, podejmujemy decyzję d_i ze zbioru decyzji D_i
- Strategia to ciąg decyzji $d_1, d_2, \dots, d_n$
- Własność Markowa: w dowolnym stanie procesu, wpływ dalszych decyzji na funkcję celu zależy tylko od bieżącego stanu i tych decyzji (historia nie wpływa)
- Wiele zadań optymalizacji da się sprowadzić do (potraktować jak) WPD z własnością Markowa
- Na przykład $z = f_1(s_1, x_1) + \dots + f_n(s_n, x_n)$
- Z własności Markowa wynika zasada optymalności Richarda Bellmana którą wykorzystuje PD: „Strategia optymalna ma tę własność, że niezależnie od tego jaki był stan początkowy i jakie były decyzje początkowe, pozostałe decyzje muszą tworzyć strategię optymalną ze względu na stan będący efektem tych początkowych decyzji.”

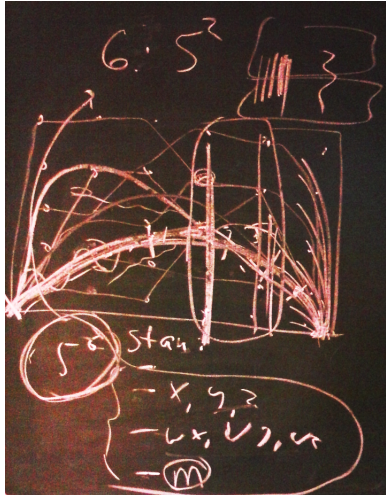
Programowanie dynamiczne – ilustracja z wykładu

Trajektorie lotu rakiety



Programowanie dynamiczne – ilustracja z wykładu

Trajektorie lotu rakiety



Zauważ, że to zastosowanie jest niemal identyczne z TSP.

Programowanie dynamiczne – ogólniejsze ujęcie

- Zamiast etapów przebiegających w czasie, wyobrażamy sobie problemy zagnieżdżone w sobie (żeby rozwiązać dany problem trzeba rozwiązać wszystkie jego podproblemy)
- Tworzy się w ten sposób drzewo zależności, w korzeniu mamy nasz pierwotny problem („początek czasu”)
- Problemy mają zwykle taką samą naturę, więc całość można zwięźle zapisać rekurencyjnie
- Rekurencję można ewentualnie rozwinąć w iterację, to kwestia techniczna i wybór
- Żeby uniknąć wielokrotnego rozwiązywania identycznych podproblemów podczas przechodzenia przez drzewo, „kieszujemy” ich wyniki (*memoization*) i to pozwala zredukować złożoność obliczeniową.

Programowanie dynamiczne – przykład

Zadanie: jednowymiarowy proces alokacyjny. Rozdziel $M = 8$ jednostek zasobu na $N = 3$ działalności (zyski podane w tabeli) tak, aby zmaksymalizować całkowity zysk.

x	0	1	2	3	4	5	6	7	8
$f_1(x)$	0	5	15	40	80	90	95	98	100
$f_2(x)$	0	5	15	40	60	70	73	74	75
$f_3(x)$	0	4	26	40	45	50	51	52	53

- 1 Sformułuj ogólne zadanie jako PM
- 2 Zaproponuj sposób jego dekompozycji dla PD
- 3 Rozwiąż powyższą instancję problemu.

Programowanie dynamiczne [obliczenia]

x	0	1	2	3	4	5	6	7	8
$f_1(x)$	0	5	15	40	80	90	95	98	100
$f_2(x)$	0	5	15	40	60	70	73	74	75
$f_3(x)$	0	4	26	40	45	50	51	52	53

[illegible]

Programowanie dynamiczne – TSP

Zaproponuj, jak zastosować PD dla TSP.

Niech kolejne etapy procesu decyzyjnego odpowiadają wyborowi kolejnych miast do odwiedzenia.