# 17. Podstawowe problemy, metody i algorytmy optymalizacji dyskretnej.

Def: optymalizacja dyskretna:

Optymalizacja – mamy zmienną decyzyjną, ograniczenia na nią i funkcję celu. Należy wybrać taką wartość dla zmiennej decyzyjnej, żeby funkcja celu dla niej była jak najmniejsza (największa) i by zmienna decyzyjna spełniała ograniczenia.

Problem optymalizacyjny, gdzie zmienna decyzyjna jest dyskretna. Dwa warte uwagi rodzaje problemów dla optymalizacji dyskretnej:

1. Programowanie całkowitoliczbowe (integer programming)
2. Problemy kombinatoryczne

Ten podział nie jest zbyt ostry, gdyż wiele problemów kombinatorycznych da się zamodelować jako programowanie całkowitoliczbowe oraz wiele problemów z programowania całkowitoliczbowego może mieć interpretację kombinatoryczną.

## Programowanie całkowitoliczbowe

* Programowanie liniowe – programowanie, gdzie ograniczenia i funkcja celu ma postać liniową.
* Programowanie całkowitoliczbowe – programowanie liniowe, gdzie zmienne decyzyjne są dyskretne. Jest to problem NP-trudny.

Typowe problemy:

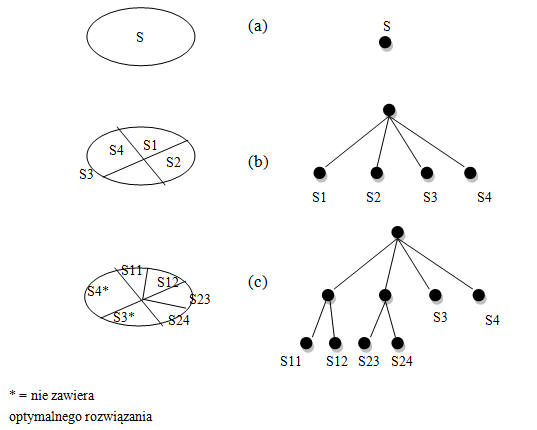
* Problem plecakowy (Knapsack) – trzeba zapakować do plecak przedmioty o jak największej wartości, przy ograniczeniu na miejsce.
* Minimalne pokrycie (Set cover) – mamy rodzinę zbiorów na uniwersum . Każdy ma koszt . Należy znaleźć taką rodzinę zbiorów , by . Optymalizacja polega na tym, by było jak najmniejsze.

Metody i algorytmy:

* Relaksacja – ściąga się ograniczenie na dyskretność zmiennej decyzyjnej. Zmienia to problem na programowanie liniowe, które można rozwiązać optymalnie w czasie wielomianowym. Wada tego rozwiązania jest taka, że uzyskane rozwiązanie może nie być rozwiązaniem poprawnym (nie da rozwiązania, które jest w dyskretnym zbiorze rozwiązań dopuszczalnych). Zaokrąglanie wartości nie musi (i zazwyczaj nie daje) optymalnego rozwiązania.
* Metody płaszczyzn tnących (Cutting Plane) – po relaksacji nadaje się dodatkowe ograniczenia, by rozwiązanie było poprawne.
* Brench&Bound – metoda przeglądu zupełnego. Działa szybciej niż zwykły przegląd zupełny, ponieważ potrafi odrzucać zbiory rozwiązań, bez ich sprawdzania, co do których jest pewna, że są gorsze niż optymalne.

Odbywa się to poprzez estymację ograniczeń zbiorów rozwiązań i porównywania ich do aktualnego najlepszego rozwiązania. W przypadku, kiedy estymowana wartość rozwiązania (niekoniecznie dopuszczalnego) jest gorsza od aktualnie najlepszego rozwiązania (dopuszczalnego) wtedy jest ono odrzucane wraz ze wszystkimi rozwiązaniami, które można na jego podstawie wygenerować.

Do zastosowania algorytmu do konkretnego problemu potrzeba wyspecyfikować a) metod podziału zbioru rozwiązań, b) funkcję ograniczenia górnego dla zbioru rozwiązań.



* Każdy problem ma też własne algorytmy aproksymacyjne, wymyślone specjalnie dla niego.

1. **Problemy kombinatoryczne**

Problem kombinatoryczny jest trójką , gdzie:

* jest skończonym zbiorem elementów.
* jest zbiorem dopuszczalnych rozwiązań.
* jest funkcją określającą koszt rozwiązania.

Celem jest wyznaczenie globalnego minimum, tj. rozwiązania , takiego, że .

Typowe problemy:

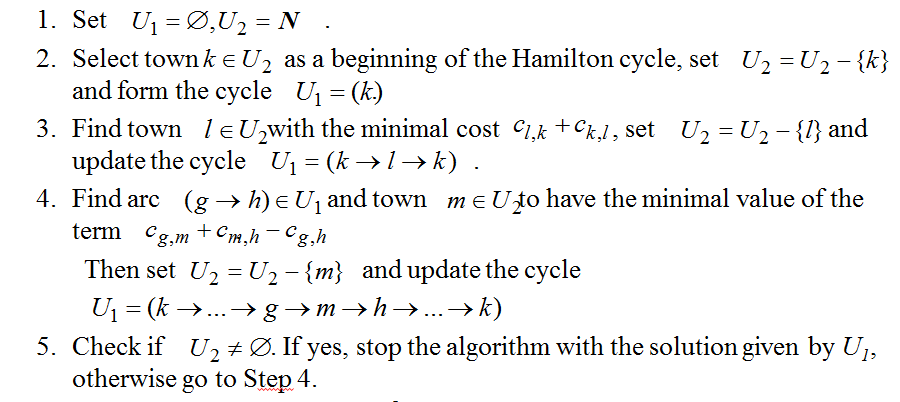
* Problemy dla kompleksu operacji
* Komiwojażera (TSP - Travelling Salesman Problem)

Def:

* + E jest zbiorem krawędzi gafu pełnego
  + jest zbiorem cykli Hamiltona w
  + jest długością trasy .

Algorytmy:

* + Algorytm włączania (Insertion)
    - Złożoność obliczeniowa
    - 2-aproksymacyjny
    - Algorytm według JJ:



* + -opt
    - Złożoność obliczeniowa
    - -opt jest 2-aproksymacyjny
    - Najpopularniejsze to 2-opt i 3-opt
    - Usuwamy niesąsiadujących krawędzi, a następnie dodajemy inne krawędzie do grafu tak by powstał cykl.



## Źródła

* wykłady JJ
* wykłady Kasperskiego
* <http://mat.gsia.cmu.edu/orclass/integer/integer.html>