DM63
HEURISTICS FOR
COMBINATORIAL OPTIMIZATION

Lecture 14

Application Example and Further Notions in Optimization.

Marco Chiarandini

Outline

Outline

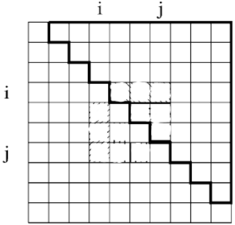
Linear Ordering Problem

Input: an $n \times n$ matrix C

Task: Find a permutation π of the column and row indexes $\{1, \ldots, n\}$ such that the value

$$f(\pi) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} c_{\pi_i \pi_j}$$

is maximized. In other terms, find a permutation of the columns and rows of C such that the elements in the upper triangle is maximized.



LOP Applications: Economics

Input-output analysis

The economy of a state is divided into n sectors, and an $n \times n$ input-output matrix C is constructed where the entry c_{ij} denotes the amount of deliveries from sector i to sector j in that year.

Triangulation (ie, solving associated LOP) allows identification of important inter-industry relations in an economy with respect to different countries.

An input-output analysis depicts and analyzes the dependence of one industry or sector on another.

LOP Applications: Graph Theory

Definition: A directed graph (or digraph) D consists of a non-empty finite set $V(D)$ of distinct vertices and a finite set A of ordered pairs of distinct vertices called arcs.

Feedback arc set problem (FASP)

Input: A directed graph $D = (V, A)$, where $V = \{1, 2, \ldots, n\}$, and arc weights c_{ij} for all $[ij] \in A$

Task: Find a permutation $\pi_1, \pi_2, \ldots \pi_n$ of V (that is, a linear ordering of V) such that the total costs of those arcs $[\pi_j \pi_i]$ where $j > i$ (that is, the arcs that point backwards in the ordering)

$$f(\pi) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} c_{\pi_j \pi_i}$$

is minimized.

LOP Applications: Graph Theory (2)

Definition: A linear ordering of a finite set of vertices $V = \{1, 2, \ldots, n\}$ is a bijective mapping (permutation) $\pi : \{1, 2, \ldots, n\} \rightarrow V$. For $u, v \in V$, we say that u is "before" v if $\pi^{-1}(u) < \pi^{-1}(v)$ ($\pi^{-1}(i) = pos_\pi(i)$).

Definition: A digraph D is complete if, for every pair x, y of distinct vertices of D both xy and yx arcs are in D.

Definition: An oriented graph is a digraph with no cycle of length two. A tournament is an oriented graph where every pair of distinct vertices are adjacent.

Remark: Given a digraph $D = (V, A)$ and a linear ordering of the vertices V, the arc set $E = \{[uv] | \pi^{-1}(u) < \pi^{-1}(v)\}$ forms an acyclic tournament on V. Similarly, an acyclic tournament $T = (V, E)$ induces a linear ordering of V.

LOP Applications: Graph Theory (3)

Definition: The cost of a linear ordering is expressed by

$$\sum_{\pi^{-1}(u)<\pi^{-1}(v)} c_{uv}$$

where the costs c_{uv} are the costs associated to the arcs.

Linear Ordering Problem

Input: Given a complete digraph $D = (V, A)$ with arc weights c_{ij} for all $ij \in A$

Task: Find an acyclic tournament $T = (V, T)$ in D such that

$$f(T) = \sum_{ij \in T} c_{ij}$$

is maximized.

Practical Application: rank players more fairly in sport tournaments. Eg: In football, vertices are teams and c_{ij} are goals scored by team i on team j. LOP yield different ranking than scheme points+goals.

Outline

1. A Case Study on the Linear Ordering Problem

2. Hybrid Metaheuristics

3. Optimization under Uncertainty

Hybrid Metaheuristics

- Synergies of components
- Component exchange between metaheuristics
- Cooperative search (parallel metaheuristics
- Integration metaheuristics systematic methods

Note: the approach of thinking seems to be superior to sticking too strongly to the frameworks behind the different metaheuristics paradigms!

Outline

Optimization Under Uncertainty

In many real life cases some problem

- parameters are unknown,
- the evaluation function is approximated or noisy
- data are dynamic

$\Rightarrow$ optimization under uncertainty.

Robust Optimization

Parameters are known only within certain bounds

The goal robust optimization is to find a solution which is feasible for all such data and optimal in some sense.

- *absolute robustness criterion*
- *robust deviation criterion*

Stochastic optimization

If probability distributions governing the data are known or can be estimated then it is possible to take advantage of this and use stochastic optimization.

In stochastic optimization the goal is to find some policy that is feasible for all (or almost all) the possible data instances and maximizes the expectation of some function of the decisions and the random variables.

Solutions to Stochastic Problems

- *A priori* solutions
- On line solutions
- Mixed strategy solutions (two-stage)

Two-stage solutions consists of:

- A first stage where some action is taken
- A second stage (made by recourse decisions) that compensates for any bad effects that might have been experienced as a result of the first-stage decision and the random outcome of events.

The optimal policy from such a model is a single first-stage policy and a collection of recourse decisions (a decision rule) defining which second-stage action should be taken in response to each random outcome.

Solution approaches

- The expected value of the objective function can be computed mathematically.

$$g(s) = E[f(\pi, s)] \qquad s \in \mathcal{S} \text{and} \pi \text{stochastic variables}$$

Then no difference with a deterministic problem. Although the evaluation function may become computationally prohibitive.

- The expected value of the objective function can only be estimated via sampling and simulation

$$g(s) = \overline{f}(\pi, s) = \frac{1}{N} \sum_{i=1}^{N} f(\pi_i, s) \qquad \text{(unbiased estimator)}$$