Department of Mathematics and Computer Science
University of Southern Denmark, Odense

December 2, 2015
Marco Chiarandini

DM841 - Heuristics for Combinatorial Optimization

Exercises, Fall 2015

Gather in groups of two or three and carry out one of the following exercises per group. Make sure that all groups work on a different problem by first chooses, first decides. It is all right if a problem remains without group. Use the first part of the work to make sure that you have understood the problem, make eventually a small example or a drawing. Then address the points listed in the exercise.

Finally, if time advances, consider improvements in the efficiency (ie, in computational cost) of the operations for performing a first improvement or a best improvement local search with the model you have put forward. Such improvements can be obtained by:

A. fast incremental evaluation, ie, fast delta evaluation

B. neighborhood pruning, ie, avoiding to examine moves that are certainly not leading to an improvement

C. use of smart data structures

After 20 minutes of work, we will create new groups in such a way that all members come from different problems. In turn, each member has to present the problem and report the answer to the other members.

Exercise 1

Definition 1 Single Machine Total Weighted Tardiness Problem

Input: *A set J of jobs $\{1, \dots, n\}$ to be processed on a single machine and for each job $j \in J$ a processing time p_j, a weight w_j and a due date d_j.*

Task: *Find a schedule that minimizes the total weighted tardiness $\sum_{j=1}^{n} w_j \cdot T_j$, where $T_j = \{C_j - d_j, 0\}$ (C_j completion time of job j).*

- Model the problem in LS terms, ie:
  - define the variables and the solution representation
  - define a neighborhood
  - define implicit constraints
  - define soft constraints
  - define the evaluation function
- Make a computational analysis: give the neighborhood size and the computational cost of evaluating a neighbor.

Exercise 2

The Steiner tree problem is a generalization of the minimum spanning tree problem in that it asks for a spanning tree covering the vertices of a set U. Extra intermediate vertices and edges may be added to the graph in order to reduce the length of the spanning tree. These new vertices introduced to decrease the total length of the connection are called Steiner[1] vertices.

Definition 2 Steiner Tree Problem [2]

Input: *A graph $G = (V, E)$, a weight function $\omega : E \mapsto \mathbf{N}$, and a subset $U \subseteq V$.*

Task: *Find a Steiner tree, that is, a subtree $T = (V_T, E_T)$ of G that includes all the vertices of U and such that the sum of the weights of the edges in the subtree is minimal.*

The example in Figure 1 is an instance of the Euclidean Steiner problem showing that the use of Steiner vertices may help to obtain cheaper subtrees including all vertices from U.

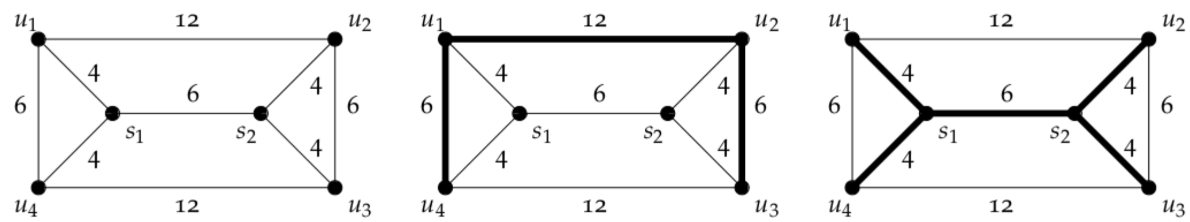

Figure 1: Vertices u_1, u_2, u_3, u_4 belong to the set U of special vertices to be covered and vertices s_1, s_2 belong to the set S of Steiner vertices. The Steiner tree in the second graph has cost 24 while the one in the third graph has cost 22.

- Model the problem in LS terms, ie:
  - define the variables and the solution representation
  - define a neighborhood
  - define implicit constraints
  - define soft constraints
  - define the evaluation function
- Make a computational analysis: give the neighborhood size and the computational cost of evaluating a neighbor.

Exercise 3

Definition 3 Total Weighted Completion Time on Unrelated Parallel Machines Problem

Input: *A set of jobs J to be processed on a set of parallel machines M. Each job $j \in J$ has a weight w_j and processing time p_{ij} that depends on the machine $i \in M$ on which it is processed.*

Task: *Find a schedule of the jobs on the machines such that the sum of weighted completion time of the jobs is minimal.*

[1] Jakob Steiner (18 March 1796 – April 1, 1863) was a Swiss mathematician.

[2] It is recommendable to search information on the problems posed, above all about the proof of their hardness. However, to maximize the positive effect of the exercises, it should be preferable to search information after you understood the problem and answered the questions.

- Model the problem in LS terms, ie:
  - define the variables and the solution representation
  - define a neighborhood
  - define implicit constraints
  - define soft constraints
  - define the evaluation function
- Make a computational analysis: give the neighborhood size and the computational cost of evaluating a neighbor.

Exercise 4

Definition 4 p-Median Problem

Input: *A set U of locations for n* users, *a set F of locations for m* facilities *and a distance matrix $D = [d_{ij}] \in \mathbf{R}^{n\times m}$.*

Task: *Select a set $J \subseteq F$ of p locations where to install facilities such that the sum of the distances of each user to its closest installed facility is minimized, i.e.,*

$$\min \Big\{ \sum_{i\in U} \min_{j\in J} d_{ij} \mid J \subseteq F \text{ and } |J| = p \Big\}$$

- Model the problem in LS terms, ie:
  - define the variables and the solution representation
  - define a neighborhood
  - define implicit constraints
  - define soft constraints
  - define the evaluation function
- Make a computational analysis: give the neighborhood size and the computational cost of evaluating a neighbor.

Exercise 5

Definition 5 Quadratic Assignment Problem

Input: *A set of n* locations *with a matrix $D = [d_{ij}] \in \mathbf{R}^{n\times n}$ of distances and a set of n* units *with a matrix $F = [f_{kl}] \in \mathbf{R}^{n\times n}$ of flows between them*

Task: *Find the assignment σ of units to locations that minimizes the sum of products between flows and distances, i.e.,*

$$\min_{\sigma\in\Sigma} \sum_{i,j} f_{ij} d_{\sigma(i)\sigma(j)}$$

- Model the problem in LS terms, ie:
  - define the variables and the solution representation

- define a neighborhood
- define implicit constraints
- define soft constraints
- define the evaluation function

- Make a computational analysis and show that a single neighbor can be evaluated in $O(n)$ when the cost of the current solution is known.

Exercise 6

Definition 6 ***Job Shop Scheduling***. *Given are m machines and a set of n jobs $J = \{1, 2, \dots, n\}$, where each job $j \in J$ consists of a set $\{o_{1j}, o_{2j}, \dots, o_{m_j j}\}$, of m_j operations. Furthermore, for each operation o_{ij} we are given a machine μ_{ij} on which it has to be processed and a processing requirement p_{ij}, where we have $\mu_{ij} \neq \mu_{i+1,j}$ for all $j \in J$ and $1 \leq i \leq m_j$. The problem is to find a schedule with minimum makespan. This means that we have to find a starting time σ_{ij} for each operation o_{ij}, such that for all $j \in J$ and $1 \leq i \leq m_j$ we have*

$$\sigma_{ij} + p_{ij} \leq \sigma_{i+1,j}$$

such that for all $j, j' \in J$, $1 \leq i \leq m_j$, and $1 \leq i' \leq m_{j'}$ with $o_{ij} \neq o_{i'j'}$ we have

$$\mu_{ij} = \mu_{i'j'} \wedge \sigma_{ij} \leq \sigma_{i'j'} \implies \sigma_{ij} + p_{ij} \leq \sigma_{i'j'}$$

and such that

$$\max_{j \in J}(\sigma_{m_j,j} + p_{m_j,j})$$

is minimal.

- Model the problem in LS terms, ie:
  - define the variables and the solution representation
  - define a neighborhood
  - define implicit constraints
  - define soft constraints
  - define the evaluation function
- Make a computational analysis and show that a single neighbor can be evaluated in $O(n)$ when the cost of the current solution is known.