

# DM86 Local Search Methods – Weekly Notes

Week 7, Spring 2006

---

## 1 Lecture March 16

In the lecture we focused on few other theoretical problems that are at the core of several practical applications. The need for solving large scale instances in these problems pushes toward the use of local search methods. The problems and some of their practical applications are:

Set Partitioning	↔	Crew Scheduling	
Set Covering	↔	Crew Scheduling	
Set Packing	↔	Single-Unit Combinatorial Auctions	
Knapsack	↔	Multidimensional Knapsack	↔ Multi-Unit Combinatorial Auctions
Bin Packing			
Generalized Assignment			
Quadratic Assignment			

With the only exception of the quadratic assignment problem, the common trait of these problems is of being strongly constrained problems. For the Knapsack there exists a very efficient dynamic programming algorithm but the multidimensional knapsack is computationally much more complex. On the set partitioning local search methods are less efficient than integer programming.

After the presentation and formalization of the problems, groups of 2 students were formed and to each group was posed one of the Exercises 1, 2 and 4 (see Section 2 of this Note). After about 40 minutes each group presented to the class the approach that they conceived for the problem at hand.

At the oral exam, for the three Exercises seen in class (Ex. 1,2,4) a student may be asked to state the problem and give least one local search solution approach. The Section 3 of this Note provides few pointers where to retrieve further information on these problems.

## 2 Exercises

The first three exercises assigned in class are here re-proposed. Students are invited to take vision of the solution methods developed for these problem by using the literature given above.

### Exercise 1: Set Covering Problem

In the Set Covering Problem (SCP) we are given a  $m \times n$  matrix  $A = (a_{ij})$  where all elements are either 1 or 0. A column  $j$  covers a row  $i$  if  $a_{ij} = 1$  and has a cost  $c_j$ . The SCP asks to find a minimum cost subset of columns that cover all the rows. In mathematical programming the problem can be formulated as follows:

$$\begin{aligned} \min \quad & \sum_{j=1}^n c_j x_j \\ \text{such that} \quad & \sum_{j=1}^n a_{ij} x_j \geq 1 \quad i = 1, \dots, m \\ & x_j \in \{0, 1\} \quad j = 1, \dots, n \end{aligned}$$

- Design a greedy construction heuristic for the SCP.
- Design Perturbative Search procedures. In this regards address the following issues:
  - Define the search space by deciding the solution representation. Address the issue of feasible and infeasible candidate solutions.
  - Define an evaluation function in the case of a search space which includes also infeasible candidate solutions.
  - Define one or more appropriate neighborhood relations in both cases of a search space restricted or not to feasible candidate solutions.
  - Conjecture which should be the best approach and why.
  - Is it possible to improve the complexity of iterative best/first improvement in one of the devised neighborhoods by maintaining one or more auxiliary data structures to speed up the computation of the delta values of a local move? If yes, then sketch the initialization and update of such auxiliary data structures.

### Exercise 2: Set Packing Problem

Given a collection of sets  $V$ , each of which has an associated real cost  $c_j$  and contains elements drawn from a finite base set  $I = \{1, \dots, m\}$ , the Set Packing Problem (SPP) consists in finding a collection of disjoint sets of maximum total cost. In mathematical programming the problem can be formulated as follows:

$$\begin{aligned} \max \quad & \sum_{j \in V} c_j x_j \\ \text{such that} \quad & \sum_{j \in V} a_{ij} x_j \leq 1 \quad i = 1, \dots, m \\ & x_j \in \{0, 1\} \quad j \in V \end{aligned}$$

- Design one or more greedy construction heuristic for the SPP.
- Design Perturbative Search procedures. In this regards address the following issues:
  - Define the search space.
  - Define the evaluation function.
  - Define one or more appropriate neighborhood relations.
  - Is it possible to improve the complexity of iterative best/first improvement in one of the devised neighborhoods by maintaining one or more auxiliary data structures to speed up the computation of the delta values of a local move?
  - Define for each of the neighborhood relations a promising search strategy (*i.e.*, an implementation of the “step” component of a local search method).

### Exercise 3: Bin Packing Problem

Given a set of  $n$  items with weights  $w_i$ ,  $i = 1, \dots, n$  associated with each of them, the bin packing problem (BPP) consists in finding the minimum number of bins of capacity  $b$  necessary to pack the items.

- Design a greedy construction heuristic for the GAP.
- Design Perturbative Search procedures. In this regards address the following issues:
  - Define the search space by deciding the solution representation. Address the issue of feasible and infeasible candidate solutions.
  - Define an evaluation function in the case of a search space which includes also infeasible candidate solutions.

- Define one or more appropriate neighborhood relations in both cases of a search space restricted or not to feasible candidate solutions.
- Conjecture which should be the best approach and why.
- Is it possible to improve the complexity of iterative best/first improvement in one of the devised neighborhoods by maintaining one or more auxiliary data structures to speed up the computation of the delta values of a local move?

#### Exercise 4: Generalized Assignment Problem

In the Generalized Assignment Problem (GAP) we are given  $n$  jobs  $J = \{1, \dots, n\}$ ,  $m$  agents  $I = \{1, \dots, m\}$ , the costs  $c_{ij}$  and the resource requirements  $a_{ij}$  of job  $j$  on agent  $i$ , and the amount  $b_i$  of the resource available to agent  $i$ . An assignment is a mapping  $\sigma : J \rightarrow I$ . The problem consists in determining a minimum cost assignment such that every job is assigned to exactly one agent and the resource constraint is satisfied. In mathematical programming the problem can be formulated as follows:

$$\begin{aligned} \min \quad & \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ \text{such that} \quad & \sum_{i=1}^m x_{ij} = 1 \quad j = 1, \dots, n \\ & \sum_{j=1}^n a_{ij} x_{ij} \leq b_i \quad i = 1, \dots, m \\ & x_{ij} \in \{0, 1\} \end{aligned}$$

- Design a greedy construction heuristic for the GAP.
- Design Perturbative Search procedures. In this regards address the following issues:
  - Define the search space by deciding the solution representation. Address the issue of feasible and infeasible candidate solutions.
  - Define an evaluation function in the case of a search space which includes also infeasible candidate solutions.
  - Define one or more appropriate neighborhood relations in both cases of a search space restricted or not to feasible candidate solutions.
  - Conjecture which should be the best approach and why.
  - Is it possible to improve the complexity of iterative best/first improvement in one of the devised neighborhoods by maintaining one or more auxiliary data structures to speed up the computation of the delta values of a local move?

#### Exercise 5: Quadratic Assignment Problem

In the Quadratic Assignment Problem (QAP) we are given  $n$  objects and  $n$  locations  $I = \{1, \dots, n\}$ , a distance matrix  $A$  where  $a_{ij}$  represents the distance between the locations  $i$  and  $j$ , and a flow matrix  $B$  where  $b_{rs}$  represents the flow between objects  $r$  and  $s$ . An assignment is a mapping  $\sigma : I \rightarrow I$ . The problem consists in determining an assignment of exactly one object to exactly one location such the objective function:

$$\sum_{i=1}^n \sum_{j=1}^n b_{ij} a_{\sigma(i)\sigma(j)}$$

is minimized.

- Most 'simple' LS methods for QAP are based on the 2-exchange neighborhood. In an Iterative Improvement procedure it is profitable to record the contribution of the local changes in a table and after each local change update the table by recomputing only a part of it.
  - Explain how the table must be constructed.

- Sketch the initialization and update procedures of such table. Which elements of the table must be updated after a local move?
- Compare the complexity of an iterative *best* improvement step by using or not such table.
- Find another speed up technique which might be profitably applied in the exploration of the neighborhood (recall those seen for the TSP).

### 3 Literature

Applications of Local Search Methods to the problems listed above can be found in the following publications (the list is far from being exhaustive!). This material is not necessary for the exam, yet the students can check whether their envisioned approaches to the exercises resemble some of those proposed in the past.

On the Set Covering Problem:

Text book Chapter 10.

M. Yagiura, M. Kishida, and T. Ibaraki. A 3-flip neighborhood local search for the set covering problem. *European Journal of Operational Research*, 172(2):472–499, 2006.

On the Set Packing Problem:

Text book Chapter 10 (treated as Combinatorial Auction).

Xavier Delorme, Xavier Gandibleux, and Joaquin Rodriguez. GRASP for set packing problems. *European Journal of Operational Research*, 153(3):564–580, 2004.

An Ant Colony Optimisation Algorithm for the Set Packing Problem. Xavier Gandibleux and xavier Delorme and Vincent T'Kindt. In Marco Dorigo, Mauro Birattari, Christian Blum, Luca M.Gambardella, Francesco Mondada, and Thomas Stützle, editors, *Ant Colony, Optimization and Swarm Intelligence: 4th International Workshop, ANTS 2004, Brussels, Belgium*, volume 3172, 2004.

On the Multidimensional Knapsack Problem:

Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack problems*. springer-berl, 2004.

On the Bin Packing Problem:

A. Alvim, D. Aloise, F. Glover, and C. Ribeiro. Local search for the bin packing problem. In *3rd Metaheuristics International Conference*, 1999.

On the Generalized Assignment Problem:

M. Yagiura, T. Ibaraki, and F. Glover. A path relinking approach with ejection chains for the generalized assignment problem. *European Journal of Operational Research*, pages 548–569, 2006.

M. Yagiura, T. Ibaraki, and F. Glover. An ejection chain approach for the generalized assignment problem. *INFORMS Journal on Computing*, 16(2):133–151, 2004.

T. Yamaguchi M. Yagiura and T. Ibaraki. A variable depth search algorithm for the generalized assignment problem. In *Meta-heuristics: Advances and Trends in Local Search Paradigms for Optimization*, pages 459–471. Kluwer Academic Publishers, 1998.

On the Quadratic Assignment Problem:

Text book Chapter 10.

T. Stützle. Quadratic Assignment Problem. Slides from the course “Stochastisch lokale Suche”, 2005 <http://www.imada.sdu.dk/~marco/Teaching/DMP86/Articles/folienQAP.pdf>.

T. Stützle and M. Dorigo. Local search and metaheuristics for the quadratic assignment problem. Technical Report AIDA-01-01, FG Intellektik, FB Informatik, TU Darmstadt, Germany, 2001.

Éric D. Taillard. Comparison of iterative searches for the quadratic assignment problem. *Location Science*, 3(2):87–105, 1995.