Institut for Matematik & Datalogi
Syddansk Universitet

August 19, 2005
JBJ

DM63 Meta-heuristics — Ugeseddel 1

"Weekly notes"

The weekly notes will only be published in the first part of the course until the project is handed out. Look at my home page for new information. The notes are in English, because we have (a) non-Danish participant(s). If this causes problems, do not hesitate to ask for clarification if you do not understand what I write!

Literature

1. Mainly "Noter til DM63" which will be available in the bookstore shortly.
2. A few research papers which will be handed out. This will depend on which project will be chosen for the course.
3. Z. Michalewicz and D.B. Fogel, How to solve it: Modern Heuristics. This will only be supplementary literature which you can use to read more about the various techniques we cover. You can find it in the library on the shelf next to Tove and Bente.
4. Further supplementary literature: C. R. Reeves (editor) Modern heuristic techniques for combinatorial problems. This book can be found in the library on the shelf next to Tove and Bente. The book contains a large amount of useful information, among others a large number of references for further study.

Evaluation

Project which is evaluated using an external censor and graded according to the 13 point scale. Group work is not permitted! The project will be handed out early November and must be returned around middle December. However you will already start working on the problem for the project in the exercises during the course.

Format:

The course is mostly experimental. There will be about 8 lectures in the beginning, where I introduce various (meta)heuristics and discuss their application to different optimization problems. We will also spend some time discussing the results you have obtained by experimenting with the heuristics. After this you will start working on the final project. The project, which is handed out early November, will involve a large amount of programming work (mostly due to many experiments, so the amount of code should not be very large). By solving the exercises as they are posed (rather than waiting until November!!) you will save a lot of time in the final project.

Lecture on September 1, 2005:

I will introduce various optimization problems: TSP, QAP, Graph-partioning, vertex-cover, feedback arc set, max clique, minimum strong spanning subdigraph, minimum maximum degree spanning trees. Based on these, I discuss local search and various 2-opt type algorithms.

Relevant literature in the notes : pages 6-11 (Non danish readers: see Reeves Chapter 1) I will cover much more than what is discussed on those pages, so show up and become inspired!

Exercises:

Start working on them right away. They will give you experience which will be useful later on. Several of the programs you write will become relevant for the project in some way. It is also important that some of you make the experiments, so that we have something to discuss when we meet.

1. Make various procedures for generating test data for the graph partitioning problem. This involves choosing a good representation of the graph and the of the partition, such that you can calculate the new objective function after swapping two vertices (one form each part) very fast. In a typical application of a heuristic like simulated annealing on the graph partitioning problem, the algorithm will perform hundreds of thousands such swaps.

2. You should make a procedure to generate random graphs with a fixed edge probability p (that is for every pair of distinct vertices i and j the edge ij is in the graph with probability p). You can do this easily using a random number generator. Think about various other ways of generating good test data e.g. by deleting edges at random or constructing a fixed good partition and then permuting the vertices.

3. Implement several heuristics for the graph partitioning problem and compare these with respect to the quality of the solutions that you find.Feel free to devise your own heuristics but you should also try the following ones (we always use A, B to denote the current partition of the vertices):

   (a) **Descent:** find a pair $x \in A, y \in B$ such that moving x to B and y to A (called swapping x and y) will decrease the number of edges between the two sets.

   (b) **Descent***: as above but also allow and interchange that preserves the number of edges across the cut.

   (c) **Steepest Descent:** Find the best pair x, y to interchange and swap x and y if the objective function decreases when doing so.

   (d) **Steepest Descent*:** as above but also allow and interchange that preserves the number of edges across the cut.

(e) **Lin-Kernighan:** Find the best pair x_1, y_1 to swap. Swap these and fix the position of these vertices (may not be swapped again in this round). In the k'th iteration ($1 \leq k \leq n = |A|$) find the best pair x_k, y_k among the remaining $(n-k)^2$ pairs, swap this pair and fix the position of x_k, y_k (may not be swapped again). When n swaps have been made, we are back at the starting partition with the roles of A and B interchanged. Now let j be chosen such that the partition $A' = (A - \{x_1, \ldots, x_j\}) \cup \{y_1, \ldots, y_j\}, B' = (B - \{y_1, \ldots, y_j\}) \cup \{x_1, \ldots, x_j\}$ has the minimum number of edges across the cut among the n such partitions. Start again from A', B' and continue until a certain stop criterion is reached (user specified, could be until no improvement for 5 rounds or similar).

Industrial projects

As some of you may know, IMADA has recently started a collaboration with industrial partners in order to increase the knowledge transfer in both directions. The leader of this project is your humble teacher, Jrgen Bang-Jensen. This collaboration will result in much better possibilities for students to work on projects which are relevant for industry and thereby preparing themselves well for working on similar problems in a company. There will be projects at all levels, from normal (course like) projects via bachelor projects and master thesis (specialer) to phd projects. DM63 is extremely relevant in this connection, since many of the projects will involve solving some optimization problem via metaheuristics or methods from mathematical programming. The latter is be the topic of the course DM85 which I will rteach again next spring.

Those of you who are interested in such projects are hereby encouraged to contact me!