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

November 6, 2006
Marco Chiarandini

DM63 - Heuristics for Combinatorial Optimization Problems

Exam Project, Fall 2006

Note 1 The project is carried out individually and it is not permitted to collaborate. The project comprises:

- algorithm design and implementation
- experimentation
- written Report

The evaluation of the project is done on the basis of the report alone. However contextually to the submission of the report a program that implements the best algorithm must also be submitted. The program will serve only to verify the correctness of the results presented. The report may be in English or Danish.

Note 2 Additional material to the project description is available on the web at: `http://www.imada.sdu.dk/Courses/DM63/project.php`. Please take vision of this link before starting the project.

Note 3 Corrections or updates to the project description will be announced on the course web-page and communicated via mailing list. It is students' responsibility to be registered in the course mailing list and to check for updates.

Note 4 *Submission.* Two printed copies of the written report must be handed in at the secretary office before 16.00 of **Tuesday, 19 December 2006**. Ask the secretary for a receipt showing that you have handed in the report in time. Contextually, the program code must be handed in by email to the lecturer. A reply will be sent as receipt.
Reports and codes handed in after the deadline will generally not be accepted. System failures, illness, etc. will not automatically give extra time.

---

1 Problem Description

The MAX-CUT problem consists in finding a partition of the vertices of a given edge-weighted undirected graph into two subsets such that the sum of the weights of the edges having endpoints in different subsets is maximized.
More formally, let $G(V, E)$ be an undirected graph and $\omega : E \to \Re$ an edge weight function. For any subset $S \subset V$ of vertices, the set of edges with one end in S and the other in $V \setminus S$ is called the *cut* induced by S, and is denoted by $\delta(S)$, i.e.,

$$\delta(S) = \{ij \in E | i \in S, j \notin S\}.$$

The weight of the cut is defined as the sum of the edge-weights in the cut:

$$\omega(\delta(S)) = \sum_{e \in \delta(S)} \omega(e) = \sum_{i \in S, j \notin S} \omega(ij).$$

The Max-Cut problem can then be defined as follows:

Max-Cut Problem
Input: An undirected graph G=(V,E) and a weight function $\omega : E \to \Re$.
Task: Find a cut $\delta(S)$ in G such that $\omega(\delta(S))$ is as large as possible.

A particular case of this optimization problem occurs when the weights of all edges are one, i.e., $\omega(e) = 1$ for all $e \in E$. This case is called the *Simple Max-Cut Problem* or *Maximum Bipartite Problem*.
The Max-Cut problem has several applications [SP95]. The two most famous are in statistical physics and in very-large-scale-integrated (VLSI) circuit design [BGJR88].
The decision version of Max-Cut is NP-complete [GJ79]. The optimization version is approximable within 1.1383 [GW95] but it is not approximable within a certain smaller ratio [sta01] unless P=NP.
State-of-art approximate algorithms for Max-Cut are reported in [BMZ01, FPRR02]. In particular, the algorithms studied in [FPRR02] are local search methods. More definitely, the methods considered are GRASP, Variable Neighborhood Search, Path Relinking and hybridizations thereof.

2 Project Content

The aim of the project is to study further heuristic algorithms beside those proposed in [FPRR02] to solve the Max-Cut problem.
Experimental analysis should be conducted on the set of four instance classes made available at `http://www.imada.sdu.dk/Courses/DM63/project.php`. A random generator is also available in case more instances with specific characteristics are needed.
All the three tasks below must be addressed in order to pass the exam.

Task 1

Devise, implement and empirically evaluate *at least one* construction heuristic and show that it produces cuts of better quality than a completely random construction.
The greedy algorithm given in [FPRR02] may be used as a reference algorithm. However, different alternatives for the order of vertices might be studied.
The consideration of other construction heuristics will contribute to increase the final mark. Heuristic ideas may be derived from the proximity of the Max-Cut problem with other problems such as the Graph Partitioning and the Max Independent Set.
The *Graph Partitioning Problem* asks, given an undirected weighted-edge graph, to determine a partitioning of V into two sets V_1, V_2 of equal or almost equal size such that the total weight of the cut is minimized.
The *Max Independent Set Problem* asks, given an undirected graph, to determine a set of vertices $I \subset V$ such that there is no edge connecting any two vertices in I and its cardinality is maximal.

Task 2

Devise, implement and empirically evaluate *at least two* local search schemes. In particular describe the solution representation, the initial solutions, the evaluation function, the neighborhood structures and the search strategy through the pivoting rules.
Provide details on the computation of the evaluation function and its delta value in the evaluation of neighbors.

Task 3

Devise, implement and empirically evaluate *at least two* metaheuristic algorithms (check the course content on the web-page for the list of methods described in the course).
The algorithms described in [FPRR02] might be useful for comparisons with new algorithms. The study of algorithms derived by methods not described in [FPRR02] will receive more credit in the final mark.

Remark 1 The final goal of the study is to indicate one or more algorithms as the best ones. The final algorithm must be the outcome of a comparison among alternatives and not because it was the only algorithm devised. In case the finally selected algorithms are more than one, indications must be provided about where each algorithm is better (e.g., in short or long run time, on different types of instances, etc.).

Remark 2 It is important to justify each choice in the configuration of the final algorithm. Alternatives for the main metaheuristic components must be considered and reasonable explanations or empirical results provided to justify the choice of one among them in the final algorithm.

Remark 3 There must be a comparison with the best known results from the literature. Specifically, a numerical table must be presented where for each instance it is reported the cost of the cut attained by the best algorithm devised and the best results in [FPRR02]. An indication of computational time is also requested.

Remark 4 It is recommended to give a clear structure to the report which reflects the component-wise vision maintained during the course. The description must be clear and detailed enough to allow the exact reimplementation of the algorithms described from the report only.

Remark 5 The total length of the report should not be less than 10 pages and not be more than 20 pages, appendix included (lengths apply to font size of 11pt and 3cm margins). Although these bounds are not strict, their violation is highly discouraged. In the description of algorithms, it is allowed to use short algorithmic sketches but not to include program codes.

Remark 6 The use of one or more of the following methods will receive higher credit in the final mark:

- variable depth search;
- ejection chains and dynasearch;
- very large scale neighborhoods.

An example which belongs to this class is the study on the adaptability to the Max-Cut problem of the Kernighan-Lin heuristic for Graph Partitioning [KL70].
The methods evolutionary algorithm and ant colony optimization will also receive higher credit than the other methods not mentioned here. Higher importance is however given to the three methods above.

Remark 7 Beside the points at Remark 6 the final mark will be determined by the level of depth of the study, the number of approaches considered and by the level in the following factors:

- complexity and originality of the approach chosen;
- originality of the experimental questions;
- organization of experiments which guarantee replicability and correctness of the conclusions;
- quality of the final results (however, showing that a promising approach does not work well in practice will also be considered equally well if there is the attempt to explain why).
- presence of complexity analysis for the procedures used;
- effective use of graphics in the presentation of experimental results.

References

[BGJR88] Francisco Barahona, Martin Grötschel, Michael Jünger, and Gerhard Reinelt. An application of combinatorial optimization to statistical physics and circuit layout design. *Operations Research*, 36(3):493–514, 1988.

[BMZ01] S. Burer, R.D.C. Monteiro, and Y. Zhang. Rank-two relaxation heuristics for max-cut and other binary quadratic programs. *SIAM Journal on Optimization*, 12:503–512, 2001.

[FPRR02] P. Festa, P.M. Pardalos, M.G.C. Resende, and C.C. Ribeiro. Randomized heuristics for the max-cut problem. *Optimization Methods & Software*, 17(6):1033–1058, 2002.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of $\mathcal{NP}$-Completeness*. Freeman, San Francisco, CA, USA, 1979.

[GW95] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the Association for Computing Machinery*, 42:1115–1145, 1995.

[KL70] Brian Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 49:291–307, February 1970.

[SP95] Zsolt Tuza Svatopluk Poljak. Maximum cuts and largest bipartite subgraphs. In Paul Seymour William Cook, Laszlo Lovasz, editor, *Combinatorial Optimization*, volume 20 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society, 1995.

[sta01] J. Håstad. Some optimal inapproximability results. *Journal of the Association for Computing Machinery*, 48(4):798–859, 2001.