On the Exact Solution of the Euclidean Three-Matching Problem

Gábor Magyar, Mika Johnsson, and Olli Nevalainen

The *Euclidean 3-matching problem* (E3MP) is to form l disjoint triplets from $n = 3l$ points in the plane and connect the three points of each triplet by two line segments so that the total length of the line segments is minimal, see Figure 1.

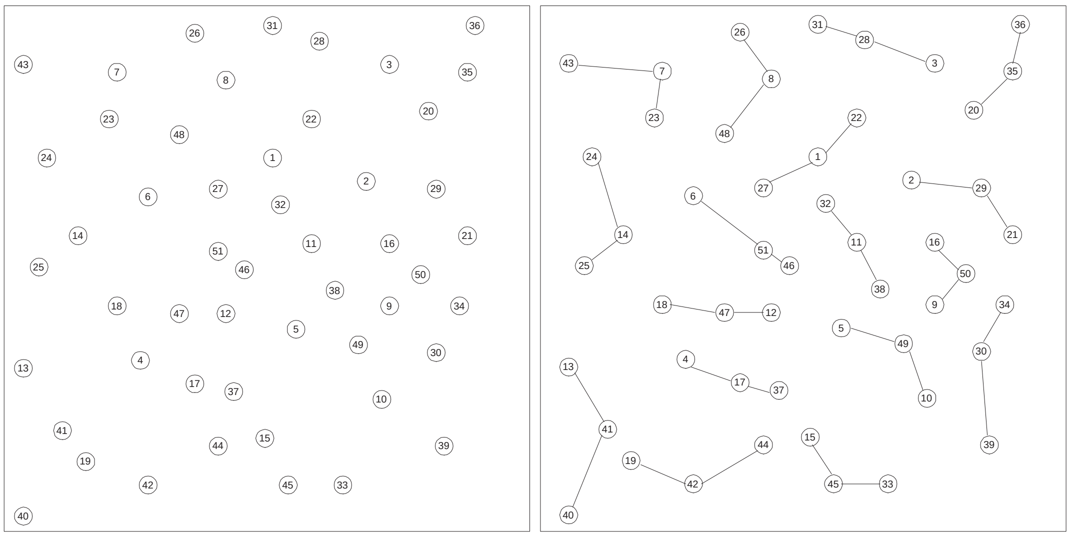

Figure 1. A sample 51-point 3MP problem instance ('eil51' from TSPLIB) along with its optimal solution. The vertices are numbered as they appear in the problem file.

The E3MP occurs in some industrial applications; in manual insertion of electronic components on a printed circuit board, the operations are arranged into close triplets in order to aid the worker in his task. Certain flexible machines (General Surface Mounter) for the automated insertion of electronic components have three to eight insertion heads and operate in cycles of component pickups and insertions with the heads. The throughput of the machine can be maximized by minimizing the length of the inter-board head movements. A similar problem occurs in the scheduling of an automated assay analysis instrument (AutoDelphia).

The problem is a special case of the general 3MP, which is NP-complete. The Euclidean problem can be solved heuristically by standard approaches, like local search heuristics using pairwise interchanges, simulated annealing, tabu search and genetic algorithm. Several lower bounds can be given for the problem. By knowing the optimal solution to the problem we can evaluate the quality of the lower bounds and the upper bound solutions of the heuristic approaches.

The 3MP can be formulated by the following *0-1 Integer Programming Problem*:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} c_{ij} \ \text{ subject to } \ \frac{1}{2} \sum_{i=1}^{n} x_{ij} + \sum_{k=1}^{n} x_{jk} = 1, (j = 1, \dots, n), x_{ij} \in \{0, 1\}$$

where the decision variable $x_{ij} = 1$ if and only if there is an $i \rightarrow j$ edge present in the solution, and c_{ij} is the cost of the $i \rightarrow j$ edge.

We find the optimal solution for the E3MP by the well-known *branch-and-bound* (B&B) technique. The branch-and-bound method implicitly enumerates all the feasible solutions of the particular combinatorial optimization problem under consideration. This is achieved by maintaining an enumeration tree which has partitions of the solution space in its nodes. The process starts with one node, namely the root, which represents all the feasible solutions to the problem. During this intelligent search, the *branching* *operation* is used to explore a selected node and to partition the solutions represented by it into its descendant nodes. The effectiveness of the B&B search method lies in the *bounding operation*, where we calculate lower bounds (in case of a minimization problem) for the solutions represented in the nodes. If the lower bound for the solutions of a certain node exceeds a known upper bound of the problem, the node and thus its whole subtree can be discarded as it cannot contain any better solution than already known. A good combination of a clever branching operation and a sharp lower bounding operation, augmented with some heuristic method for providing tight upper bounds can give reasonable performance for hard optimization problems. The task of writing an efficient branch-and-bound algorithm for our problem is challenging in itself and we discuss a number of design alternatives and their trade-offs in the study.

We introduce two branching strategies and three bounding operations for the E3MP. The latter include a lower bound based on the Lagrangian relaxation of the above integer programming formulation, a sharp problem-specific and a rather trivial lower bound. Both of the branching operations perform the partitioning by fixing and/or deleting certain edges (variables) in the *partial solutions* stored along with the particular node. These operations can use various edge selection rules for the branching and they can also incorporate automatic fixing and/or deleting of certain type of edges depending on the previous decisions leading to the partial solution in the particular node. We build several different B&B procedures from the above mentioned components and complete them with various node and edge selection rules and upper bound heuristics.

The role and performance of the different components of the B&B procedures is evaluated by empirical comparisons. The test set includes various problem instances, either coming directly from industry, or picked from TSPLIB, or generated randomly. The best variant is also compared to a public domain LP-solver and the results show that our procedure has a far better performance. We have also performed empirical tests on problems with restrictions on the available edges for the solution. These experiments provide useful information for a better understanding of the structure of the optimal solutions.

Summarizing our work, we have solved a restricted version of a practically relevant NP-complete graph problem (which is probably also NP-complete). The results show that our method is applicable for relatively small problems ($n < 54$) and its performance is superior to existing solving methods. The optimal solution of the problem is very useful at the evaluation of the lower bounds and the heuristic approaches to the problem. Furthermore, we suppose, that many of the ideas of our algorithms could be applied successfully to solve other graph problems.