PACE Solver Description: CIMAT_Team

Carlos Segura
Area de Computación, Centro de Investigación en Matemáticas (CIMAT), Guanajuato, Mexico

Lázaro Lugo
Area de Computación, Centro de Investigación en Matemáticas (CIMAT), Guanajuato, Mexico

Gara Miranda
Departamento de Ingeniería Informática y de Sistemas, Universidad de La Laguna, Spain

Edison David Serrano Cárdenas
Area de Matemáticas Aplicadas, Centro de Investigación en Matemáticas (CIMAT), Guanajuato, Mexico

Abstract

This document describes MAEDM-OCM, a first generation memetic algorithm for the one-sided crossing minimization problem (OCM), which obtained the first position at the heuristic track of the Parameterized Algorithms and Computational Experiments Challenge 2024. In this variant of OCM, given a bipartite graph with vertices $V = A \cup B$, only the nodes of the layer B can be moved. The main features of MAEDM-OCM are the following: the diversity is managed explicitly through the Best-Non-Penalized (BNP) survivor strategy, the intensification is based on Iterated Local Search (ILS), and the cycle crossover is applied. Regarding the intensification step, the neighborhood is based on shifts and only a subset of the neighbors in the local search are explored. The use of the BNP replacement was key to attain a robust optimizer. It was also important to incorporate low-level optimizations to efficiently calculate the number of crossings and to reduce the requirements of memory. In the case of the longest instances ($|B| > 17000$) the memetic approach is not applicable with the time constraints established in the challenge. In such cases, ILS is applied. The optimizer is not always applied to the original graph. In particular, twin nodes in B are grouped in a single node.

2012 ACM Subject Classification Computing methodologies → Discrete space search

Keywords and phrases Memetic Algorithms, Diversity Management, One-sided Crossing Minimization

Digital Object Identifier 10.4230/LIPIcs.IPEC.2024.31

Supplementary Material *Software*: `https://github.com/carlosségurag/PaceChallenge24`
archived at `swh:1:dir:c7ae637013531bdb241615553d8bbcb6b25ab469`
Software: `https://zenodo.org/doi/10.5281/zenodo.12512615`

Acknowledgements We would like to thank to the High-Performance Computing (HPC) team of CIMAT and to CONAHCYT for the funding of the HPC laboratories.

1 Preliminaries

The One-sided Crossing Minimization problem (OCM) involves arranging the nodes of a bipartite graph in two layers, so that the crossing of edges is minimized when a straight-line drawing is performed. In the Parameterized Algorithms and Computational Experiments (PACE) Challenge, the vertices associated to each of the layers (A and B) as well as the order of the vertices in A are given. Thus, the problem seeks an order of B so as to minimize the number of crossings. Since any order of the nodes in B is valid, a natural encoding is a permutation of the vertices in B, which has been the encoding selected for our method.

19th International Symposium on Parameterized and Exact Computation (IPEC 2024).
Editors: Édouard Bonnet and Paweł Rzążewski; Article No. 31; pp. 31:1–31:4
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Algorithm 1 Memetic Algorithm with Explicit Diversity Management for OCM.

Require: *InitFactor*, N(size of population), *Stopping criterion Phase 1 (time1)*, *Global Stopping criterion (time2)*

1: **Initialization:** Generate an initial population P_0 with N individuals. Assign $i = 0$.

2: **Iterated Local Search:** Apply Iterated Local Search to every individual in P_0.

3: **Diversity Initialization:** Calculate the initial desired minimum distance (D_0) as the mean distance among individuals in P_0 multiplied by $InitFactor$.

4: **while** the execution time of Phase 1 (time1) has not been reached **do**

5:     **Mating Selection:** Perform binary tournament selection on P_i in order to fill the mating pool with N parents.

6:     **Variation:** Apply the cycle-based crossover (CX) in the mating pool to create the set O_i with N offspring.

7:     **Iterated Local Search:** Apply Iterated Local Search to every individual in O_i.

8:     **Survivor Selection:** Apply Best-Non Penalized survivor selection strategy (BNP) to create P_{i+1} by considering P_i and O_i as input.

9:     $i = i + 1$

10: **end while**

11: **Iterated Local Search:** Apply Iterated Local Search to the best evaluated permutation so far, until time2 is exhausted

12: **Return** best evaluated permutation.

Regarding the solving strategies, Memetic Algorithms (MAs) are one of the most effective solvers for NP-hard problems. In fact, in several problems where solutions are encoded as permutations, MAs are the leading methods. This is the case of the Job-Shop Scheduling Problem [1] and the Linear Ordering Problem [3], among others. In these cases, the explicit management of diversity was key to develop robust methods that are able to reach high-quality solutions with a high probability. Thus, our team decided to adapt some of the principles that were successful in those problems to the OCM.

2 MAEDM-OCM: a first generation memetic algorithm for the one-sided crossing minimization problem

The Memetic Algorithm with Explicit Diversity Management for the One-sided Crossing Minimization problem (MAEDM-OCM) is a first-generation MA. MAEDM-OCM applies a set of operators that have already proven to be effective for permutation encoding. Given that short-term executions are performed, a first-generation MA is applied. Thus, a population-based approach is combined with a non-adaptive intensification scheme which in this case is Iterated Local Search (ILS). Algorithm 1 shows the general working operation of our proposal. Differently to most MAs, the method is divided in two phases. In the first phase (lines 1-10) a traditional MA is considered. In the second phase (line 11), ILS is applied to the best solution found so far. The reason to incorporate this second phase is that for medium and large instances, the stopping criterion used in the challenge is not enough to evolve a large number of generations. Thus, there might be opportunity for further improvements by applying ILS. In spite of this additional change, the most important decisions that affect the performance of MAEDM-OCM are related to the specific components that were used in the first phase. In the following, the working operation of each component of the first phase is described.

Our approach starts by initializing a population with N individuals, where each permutation is equiprobable (Line 1). Then, each solution is improved with ILS (Line 2). ILS uses a first-improvement stochastic hill-climber that considers the shift neighborhood [2]. This neighborhood is selected because it can be explored efficiently by storing the crossing number matrix. Neighbors are generated by moving a vertex in the given permutation to an alternative position and shifting all the vertices in the intermediate positions. However, not all the moves are taken into account. Each number is moved to the left and right until its worsening is larger than a threshold value or until it reaches the first or last position. Then, the best move is accepted. The worsening threshold is equal to the median value of the crossing number matrix multiplied by $cuttingMult$, which is a parameter of the optimizer. Regarding the perturbations performed by ILS, three different alternatives were used equiprobably: swap a set of $SwapSize$ pairs of positions, do a random shuffle of a block with size $permBlock$ or move a block with a size that is selected randomly between 1 and 10 to a random position of the permutation.

An important feature of our approach is that it considers diversity explicitly. This is done with the replacement strategy, which works by setting a minimum desired distance that is updated during the run. Similarly to [1], the distances that appear in the initial population are used to set the initial desired distance (D_0) (Line 3). In particular, D_0 is calculated as the mean distance among all the individuals in the initial population multiplied by $InitFactor$, which is a parameter of MAEDM-OCM. In order to calculate D_0, the Spearman's footrule distance [4] is employed.

MAEDM-OCM evolves a set of generations until a given stopping criterion is reached (Lines 4-10). At each generation, a set of N parents is selected using binary tournaments (Line 5). Then, the cycle-based crossover is applied (Line 6), and ILS is used to intensify (Line 7). Finally, the diversity-aware replacement strategy called BNP is applied (Line 8). BNP is an elitist survivor selection strategy that avoids the survival of too close solutions. The meaning of too close is defined dynamically. At the initial stages it forces larger distances between solutions with the aim of promoting exploration, whereas at the final stages closer solutions are accepted with the aim of promoting exploitation. The details are given in [1].

2.1 Other Improvements and Treatment of Large Instances

There were several low-level optimizations that were important to the efficiency:

- The performance of the local search was improved by storing the crossing number matrix and an improvement matrix that contains the gain of swapping two consecutive nodes of a solution.
- The crossing number matrix is calculated efficiently by using data structures such as balanced search trees or the two-pointer technique, depending on the size of the instance.
- The data types for storing the matrices is adapted depending on the requirements of the instance.
- Twin vertices in B are grouped for creating a shorter graph with parallel edges that can be used to solve the original problem with a reduced search space.

In spite of the efforts for efficiency, the stopping time established for the challenge was not large-enough for using the two phases of MAEDM-OCM in the longest instances. In instances with $|B| > 17000$, only the second phase is used, i.e. it directly applies ILS. Moreover, in this case the solution is not created randomly. Instead, for each edge (a_i, b_i) an score is assigned which is equal to the amount of existing edges with its A-endpoint lower than a_i and its B-endpoint different to b_i. Then, each vertex of B is assigned an score equal to the mean of its adjacent edges. The initial solution greedily sorts the vertices in B by increasing score.

References

1 Oscar Hernández Constantino and Carlos Segura. A parallel memetic algorithm with explicit management of diversity for the job shop scheduling problem. *Appl Intell*, pages 1–13, April 2021.

2 Manuel Laguna, Rafael Martí, and Vicente Campos. Intensification and diversification with elite tabu search solutions for the linear ordering problem. *Comput Oper Res*, 26(12):1217–1230, 1999.

3 Lázaro Lugo, Carlos Segura, and Gara Miranda. A diversity-aware memetic algorithm for the linear ordering problem. *Memetic Computing*, 14(4):395–409, 2022. doi:10.1007/s12293-022-00378-5.

4 Marc Sevaux, Kenneth Sörensen, et al. Permutation distance measures for memetic algorithms with population management. In *Proceedings of 6th Metaheuristics International Conference*, MIC'05, pages 832–838, 2005.