Iterated Local Search with Random Restarts for the Mentorship and Teamwork Problem (Google Hash Code 2022)

Uran Lajçi[1], Elvir Misini[1], Kadri Sylejmani[2], Labeat Arbneshi[2]

[1] Faculty of Computer and Software Engineering, Prishtina University.
Agim Ramadani, 1000 Prishtinë, Kosovë
[uran.lajci, elvir.misini]@student.uni-pr.edu
[2] Faculty of Computer and Software Engineering, Prishtina University.
Agim Ramadani, 1000 Prishtinë, Kosovë
[kadri.sylejmani,
labeat.arbneshi]@uni-pr.edu

Keywords: Mentorship and Teamwork – Iterated Local Search – Google Hash Code 2022

1. Introduction and problem description

The field of optimization has been the focus of extensive research due to its wide-ranging applications, from logistics and supply chain management to workforce scheduling. This paper delves into a novel application of optimization techniques, specifically addressing the Mentorship and Teamwork Problem (MTP) proposed in the Google Hash Code 2022 competition [1]. The MTP is a complex task that involves the assignment of contributors to projects, subject to a variety of constraints and objectives. The overarching goal is to maximize the overall score of the assignments, thereby optimizing the allocation of resources and the productivity of the contributors. The MTP is characterized by a set of hard constraints, including the stipulation that a contributor can work on only one project at a time, and once a project commences, contributors work on it for the specified duration, becoming available for other projects only upon its completion. Furthermore, a contributor can be assigned to a project role if they meet the required skill level or have the skill at exactly one level below the required level and are mentored by another contributor on the same project who has the required skill level or higher. Each contributor can fill at most one role on a single project, and each role in a project can be filled by only one contributor. In addition to these hard constraints, there are also soft constraints that aim to maximize the total score for completed projects and assign contributors to project roles that fit their qualifications and enable opportunities. To address this problem, we propose an Iterated Local Search (ILS) algorithm with random restarts [2]. The ILS algorithm commences with an initial list of assignments and iteratively refines the solution by exploring the search space, aiming to optimize the given objective function. The algorithm incorporates perturbations and acceptance criteria to escape local optima and enhance search performance. The random restarts are introduced to further improve the algorithm's exploration capabilities. When a certain stopping criterion is met, the algorithm restarts from a new randomly generated solution, enabling the algorithm to explore different regions of the search space and avoid getting trapped in suboptimal solutions. In the context of related work, Ackah et al. developed an original algorithm for rich portfolio optimization (ARPO) that combines an iterated local search metaheuristic with quadratic programming to efficiently deal with complex variants of the mean-variance portfolio optimization problem [3]. They demonstrated that ARPO is competitive against existing state-of-the-art approaches in terms of the quality of the best solution generated and the computational times required to obtain it. Our paper, "Iterated Local Search with Random Restarts for the Mentorship and Teamwork Problem (Google Hash Code 2022)", also utilizes the Iterated Local Search algorithm for optimization problems. However, while our paper focuses on optimizing mentorship and teamwork, Ackah et al.'s paper applies the algorithm to financial portfolio optimization. Both papers underscore the versatility and effectiveness of the Iterated Local Search algorithm in diverse problem-solving contexts.

2. Solution method

Our solution to the Mentorship and Teamwork Problem (MTP) is encapsulated within the `IteratedLocalSearch` class, which employs an Iterative Local Search (ILS) algorithm with random restarts. This algorithm is designed to maximize the score for completed projects while adhering to the constraints of the MTP. It accepts an initial solution, a maximum runtime, and lists of projects and contributors as inputs. The ILS algorithm commences by initializing the search space (`currentSolution`), the home base (`currentHomeBase`), and the best solution found so far (`bestSolution`). It iterates until the allotted time is reached, applying a series of operators, namely `insertProjects`, `removeProject`, and `replaceContributors`, to tweak the current solution (`currentSolution`). If the tweaked solution (`tweakedSolution`) is valid and of higher quality, it replaces `currentSolution`. The algorithm updates the best solution (`bestSolution`) if `currentSolution` is superior.

The `IterativeOptimization` class includes several helper methods:

- deltaQuality: Calculates the difference in quality between the old and new solutions.
- copySolution: Creates a deep copy of the current solution.
- quality: Computes the quality of a given solution.
- newHomeBase: Selects a new home base for the next iteration.
- tweak: Applies a random operator to the current solution.
- insertProjects, removeProject, replaceContributors: These are the operators that modify the current solution.
- perturb: Creates a new starting point for the next iteration.

The algorithm returns the best valid solution found within the given time. If no valid solution is found, it returns the initial solution. The `IterativeOptimization` class offers an efficient approach to solving complex mentorship and teamwork problems using iterative local search techniques. This approach is particularly effective for large-scale problems where traditional optimization methods may be computationally expensive or infeasible.

Table 1. Considered neighborhoods

insertProjects(assignments, projects, contributors)	The `InsertProjects` method shuffles the first 10% of unassigned projects to the end of the list, promoting project rotation. If unassigned projects remain, it employs `InitialSolver.solveMentorshipAndTeamwork` to create additional assignments, ultimately returning an updated list inclusive of these new assignments.
removeProjects(assignments, contributors)	The `RemoveProject` function de-assigns the last 10% of assignments and reduces matching contributors' skill levels if the skill level is equal to or one higher than the project's requirement. The method is designed to rotate projects and adjust contributor skills, leading to a more balanced assignment list.
replaceContributors(assignmets, contributors)	The ReplaceContributors method replaces assigned contributors with unassigned contributors who have the same or higher skills, improving assignment quality. It iterates through assignments, searches for suitable replacements, and modifies the assignments accordingly.
perturb(assignmets)	The Perturb method introduces perturbations to the current solution by randomly swapping assignments. It performs a specified number of swaps, chosen based on the size of the solution. This random perturbation can help explore different configurations and escape local optima in the search space.

3. Preliminary experimental results

Table 2 shows the best results for the MTP and our best results. There are 5 instances that are used as input, where each instance has a specific way that is configured.

Table 2. Preliminary results. Best available solutions are from Hash Code - Problem Archive [4].

Instance	State of the art	Our Solutions
a	33	33
b	969,087	398,591
c	229,517	114,233
d	674,945	71,984
e	1,640,454	1,542,003
f	706,200	291,371
Total	4,220,236	2,418,215

Table 3. Experiments with all instances in the test set (with at least 5 minutes)

Test	a	b	c	d	e	f	Total
1	33	398,591	4,660	71,984	1,221,927	36,415	1,335,019
2	33	398,591	3,711	71,984	1,208,722	30,444	1,713,485
3	33	398,591	11,315	71,984	1,231,212	41,408	1,754,543
4	33	398,591	7,973	71,984	1,203,381	61,036	1,742,998
5	33	398,591	5,649	71,984	1,535,152	49,919	2,062,328
6	33	398,591	8,316	71,984	1,542,003	299,382	2,320,309
7	33	398,591	16,312	71,984	1,535,226	49,309	2,071,445
8	33	398,591	10,068	71,984	1,535,226	173,698	2,189,600
9	33	398,591	7,218	71,984	1,535,226	285,754	2,298,806
10	33	398,591	114,233	71,984	1,535,205	291,371	2,411,417

The complete solution for the Mentorship and Teamwork Problem, utilizing the Iterated Local Search with Random Restarts approach, can be found in the GitHub repository provided by Misini et al. [5]. The repository contains the source code, documentation, and relevant resources to better understand and implement the proposed algorithm. Readers are encouraged to visit the repository to explore the solution in detail and experiment with the algorithm for their specific problem instances.

4. Future work

The quest for enhanced solutions to the Mentorship and Teamwork Problem (MTP) opens up a plethora of promising avenues for future exploration. One such direction is the development of innovative operators. These operators could offer fresh ways to modify both the initial and candidate solutions, potentially leading to more effective search strategies. This concept is explored in the paper "Efficient Decision Makings for Dynamic Weapon-Target Assignment by Virtual Permutation and Tabu Search Heuristics" [6]. Another potential area of focus is the fine-tuning of operator usage percentages. By optimizing the proportion of changes made by operators, we could achieve a more efficient balance between exploration and exploitation of the search space, thereby enhancing the overall performance of the algorithm. The exploration of alternative local search algorithms not utilized in this paper also presents an intriguing opportunity. By applying different local search algorithms to the MTP, we could gain new perspectives and potentially discover more effective solutions. The paper "Effective metaheuristics for scheduling a hybrid flowshop with sequence-dependent setup times" [7] provides valuable insights into this approach. The use of hybrid methods, which combine different techniques into a single algorithm, could also prove beneficial. Such methods could leverage the strengths of multiple approaches, potentially improving both the quality of solutions and computational efficiency. Finally, the construction of heuristics for generating better initial solutions could provide a stronger starting point for the search process. This could lead to improved final solutions by enabling the algorithm to start from a more advantageous position in the search space. The paper "A Tutorial for Competent Memetic Algorithms: Model, Taxonomy, and Design Issues" [8] offers guidance on this topic. For a broader understanding of optimization problems and potential future directions, the surveys "Nurse rostering problems—a bibliographic survey" [9] and "A literature review on the vehicle routing problem with multiple depots" [10] provide comprehensive overviews. These future work directions and corresponding references offer a roadmap for continued research into the MTP, with the goal of developing increasingly effective and efficient solutions.

References

1. Google Coding Competitions, "Google Hash Code: Problem Statement," [Online]. Available: https://codingcompetitions.withgoogle.com/hashcode/round/00000000008caae7/0000 0000 0098afc8
2. H. R. Lourenço, O. C. Martin, and T. Stützle, "Iterated Local Search," [Online]. Available: https://repositori.upf.edu/bitstream/handle/10230/1243/513.pdf?sequence=1
3. R. Kizys , A. Juan, B. Sawik, and L. Calvet , "A Biased-Randomized Iterated Local Search Algorithm for Rich Portfolio Optimization," Applied Sciences, vol. 9, no. 17, p. 3509, Aug. 2019, doi: 10.3390/app9173509.
4. Google Coding Competitions, "Hash Code - Problem Archive," [Online]. Available: https://codingcompetitions.withgoogle.com/hashcode/round/00000000008caae7.
5. E. Misini, U. Lajçi, "Mentorship_Teamwork_ILS," GitHub, [Online]. Available: https://github.com/elvirmisini/Mentorship_Teamwork_ILS.
6. B. Xin, J. Chen, J. Zhang, L. Dou, and P. Zhihong, "Efficient Decision Makings for Dynamic Weapon-Target Assignment by Virtual Permutation and Tabu Search Heuristics," in IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews, vol. 41, no. 6, pp. 898-913, Nov. 2011. [Online]. Available: https://doi.org/10.1109/tsmcc.2010.2049261
7. Q. Pan, L. Gao, and K. Gao, "Effective metaheuristics for scheduling a hybrid flowshop with sequence-dependent setup times," in Applied Mathematics and Computation, vol. 301, pp. 12-32, June 2017. [Online]. Available: https://doi.org/10.1016/j.amc.2017.01.004
8. N. Krasnogor and J. G. Smith, "A Tutorial for Competent Memetic Algorithms: Model, Taxonomy, and Design Issues," in IEEE Transactions on Evolutionary Computation, vol. 9, no. 5, pp. 474-488, Oct. 2005. [Online]. Available: https://doi.org/10.1109/tevc.2005.850260
9. B. Cheang, H. Li, A. Lim, and B. Rodrigues, "Nurse rostering problems—a bibliographic survey," in European Journal of Operational Research, vol. 151, no. 3, pp. 447-460, Dec. 2003. [Online]. Available: https://doi.org/10.1016/s0377-2217(03)00021-3
10. J. R. Montoya-Torres, J. Benavides Franco, S. Nieto Isaza, H. Felizzola Jiménez, and N. Herazo-Padilla, "A literature review on the vehicle routing problem with multiple depots," in Computers & Industrial Engineering, vol. 79, pp. 115-129, Jan. 2015. [Online]. Available: https://doi.org/10.1016/j.cie.2014.10.029