Iterated Local Search with Random Restarts for the

Mentorship and Teamwork Problem (Google Hash Code 2022)

Elvir Misini1, Uran Lajçi1, Kadri Sylejmani2, Labeat Arbneshi2

1 Faculty of Computer and Software Engineering, Prishtina University.

Agim Ramadani, 1000 Prishtinë, Kosovë

[elvir.misini, uran.lajci]@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

In the paper "Iterated Local Search with Random Restarts for the Mentorship and Teamwork Problem (Google Hash Code 2022)," we focus on the Mentorship and Teamwork problem proposed in the Google Hash Code 2022 competition. The Mentorship and Teamwork Problem (MTP) involves creating assignments that are a combination of projects and contributors. We try to do the creation of assignments in the most optimal way. This is done by considering various constraints and objectives to maximize the overall score of the assignments. Hard constraints are: a contributor can work on only one project at a time, once a project starts, contributors work on it for the specified duration and become available for other projects only after its completion, 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, each role in a project can be filled by only one contributor. Soft constraints are: maximize the total score for completed projects, assign contributors to project roles that fit their qualifications and enable opportunities. The complete problem description can be found in the google hash code 2022 web page [1]. To solve this problem we propose an Iterated Local Search (ILS) [2] algorithm with random restarts. The algorithm starts with an initial list of assignments and iteratively refines the solution by exploring the search space, aiming to optimize the given objective function. The ILS 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. This helps the algorithm to explore different regions of the search space and avoid getting trapped in suboptimal solutions.In the paper "A Biased-Randomized Iterated Local Search Algorithm for Rich Portfolio Optimization", Ackah, Juan, Sawik, and Calvet [3] develop an original algorithm for rich portfolio optimization (ARPO). This algorithm considers more realistic constraints than those usually analyzed in the literature. The ARPO algorithm combines an iterated local search metaheuristic with quadratic programming to efficiently deal with complex variants of the mean-variance portfolio optimization problem. The authors demonstrate 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. In relation to my paper "Iterated Local Search with Random Restarts for the Mentorship and Teamwork Problem (Google Hash Code 2022)", both papers utilize the Iterated Local Search algorithm for optimization problems. However, while my paper focuses on optimizing mentorship and teamwork, Ackah et al.'s paper applies the algorithm to financial portfolio optimization. Both papers demonstrate the versatility and effectiveness of the Iterated Local Search algorithm in diverse problem-solving contexts.

1. Solution method

The IteratedLocalSearch class presents a utility for iterated local search with random restarts, applied to a mentorship and teamwork problem. The algorithm aims to maximize the score for completed projects while adhering to problem constraints. It accepts an initial solution, maximum runtime, and lists of projects and contributors as inputs.

The search algorithm initializes the search space (S), home base (H), and the best solution found so far (Best). It iterates until the allotted time is reached, applying operators such as Swap, InsertProjects, Inversion, and RemoveProject to tweak the current solution (S). If the tweaked solution (R) is valid and of higher quality, it replaces S. The algorithm updates the best solution (Best) if S is better.

Helper methods include deltaQuality, Copy, Quality, NewHomeBase, Tweak, Swap, InsertProjects, Inversion, RemoveProject, and Perturb. These methods calculate quality differences, create solution copies, compute solution quality, select new home bases, apply random operators, and create new starting points for iterations, respectively.

The algorithm returns the best valid solution found within the given time. If no valid solution is found, it returns the initial solution. The IteratedLocalSearch class offers an efficient approach to solving complex mentorship and teamwork problems using iterated local search techniques.

*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. |

1. Preliminary experimental results

The table 2 shows the best results of the solution for the MTP. The second row showes the state of the art solutions, they are not from one team but the best among all teams.

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

|  |  |  |
| --- | --- | --- |
|  | State of the art | Our Solutions |
| a | 33 | 33 |
| b | 969,087 |  |
| c | 229,517 |  |
| d | 674,945 |  |
| e | 1,640,454 |  |
| f | 706,200 |  |
| Total | 4,220,236 |  |

In the table 2 we show our results of the solution accros 10 tests, where each test is runing for at least 5 minutes.

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Minutes | a | b | c | d | e | f | class | Total |
| Test 1 | 33 |  |  |  |  |  |  |  |
| Test 2 | 33 |  |  |  |  |  |  |  |
| Test 3 | 33 |  |  |  |  |  |  |  |
| Test 4 | 33 |  |  |  |  |  |  |  |
| Test 5 | 33 |  |  |  |  |  |  |  |
| Test 6 | 33 |  |  |  |  |  |  |  |
| Test 7 | 33 |  |  |  |  |  |  |  |
| Test 8 | 33 |  |  |  |  |  |  |  |
| Test 9 | 33 |  |  |  |  |  |  |  |
| Test 10 | 33 |  |  |  |  |  |  |  |
| Test with heuristic |  |  |  |  |  |  |  |  |

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.

1. Future work

There are some different directions that the future work of finding better solutions for the mentorship and teamwork problem can be headed. These are some of our ideas that can be considered in the future:

1. The construction of different operators that can be used to change the initial solution and candidate solutions.
2. Finding a better percentage of usage to make changes from the operators.
3. Using different local search algorithms that are not used in this paper to solve this problem.
4. Using hybrid methods.
5. Construction of heuristics to get better initial solutions

Some papers that correspond to the above ideas for future work are show bellow:

1. For the construction of different operators and usage percentages, consider the following paper: "Efficient Decision Makings for Dynamic Weapon-Target Assignment by Virtual Permutation and Tabu Search Heuristics" [6]
2. For using different local search algorithms and hybrid methods, refer to: "Effective metaheuristics for scheduling a hybrid flowshop with sequence-dependent setup times," [7]
3. For the construction of heuristics to get better initial solutions, you can refer to: "A Tutorial for Competent Memetic Algorithms: Model, Taxonomy, and Design Issues," [8]
4. For additional insights into optimization problems and potential future directions, consider these surveys: "Nurse rostering problems—a bibliographic survey," [9], "A literature review on the vehicle routing problem with multiple depots," [10]

References

1. "Google Hash Code: Problem Statement," Google Coding Competitions. [Online]. Available: [https://codingcompetitions.withgoogle.com/hashcode/round/00000000008caae7/00000000](https://codingcompetitions.withgoogle.com/hashcode/round/00000000008caae7/000000000098afc8) [0098afc8](https://codingcompetitions.withgoogle.com/hashcode/round/00000000008caae7/000000000098afc8)
2. H. R. Lourenço, O. C. Martin, and T. Stützle, "Iterated Local Search," Universitat Pompeu Fabra, Barcelona, Spain, helena.ramalhinho@econ.upf.es; Université Paris-Sud, Orsay, France, martino@ipno.in2p3.fr; Darmstadt University of Technology, Darmstadt, Germany, [stuetzle@informatik.tu-darmstadt.de.](mailto:stuetzle@informatik.tu-darmstadt.de) [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," Google. [Online]. Available: <https://codingcompetitions.withgoogle.com/hashcode/round/00000000008caae7>. [Accessed: May 4, 2023].
5. E. Misini, U. Lajçi, "Mentorship\_Teamwork\_ILS," GitHub, [Online]. Available: [https://github.com/elvirmisini/Mentorship\_Teamwork\_ILS.](https://github.com/elvirmisini/Mentorship_Teamwork_ILS) [Accessed: May 4, 2023].
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](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](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](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](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](https://doi.org/10.1016/j.cie.2014.10.029)