Hyper-heuristics for the Time-dependent ATSP with Time Windows and Precedence Constraints applied to Air Travel

Matheus Simões, Laura Bahiense and Celina Figueiredo

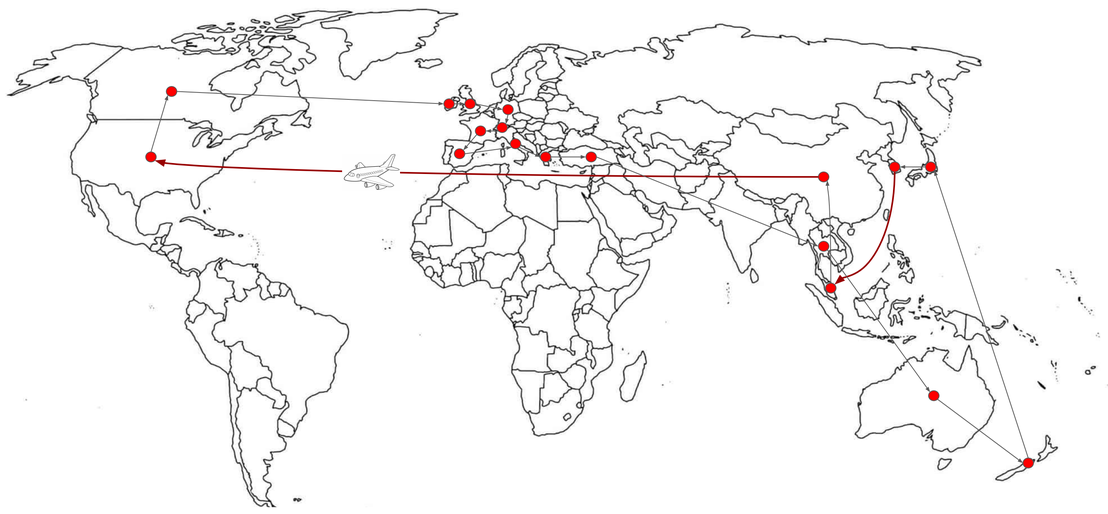
Federal University of Rio de Janeiro (UFRJ), Brazil
{simoesmc, laura, celina}@cos.ufrj.br

Motivation:

A traveler wants to visit N countries in N weeks, one country per week with some special restrictions:

- Time Window, i. e., some countries must be visited in a specific week;
- Precedence Constraint, i. e., there are pair of countries with a precedence relationship, where a country must be visited in the week immediately following the other.

The goal it to find a tour that minimizes the cost of flight between these countries. This problem can be modeled as a variant of the TSP. In this context, going from one vertex to the other in different weeks has an impact on the cost, generating a dependence on the time in which a given connection takes place. Furthermore, the existence of a connection from one country to another does not guarantee that the opposite occurs. So, this problem is summarized as a Time-dependent ATSP with time window and precedence constraint in the context of air travel.



Example of a tour with two precedence constraints where it is necessary to visit China immediately before USA, and South Korea immediately before Malaysia.

Methodology:

The problem and the dataset were introduced by Saradatta and Pongchairerks in 2017. They generated random initial solutions that were improved by two local search algorithms, using SWAP and INSERT heuristics respectively, and compared the result with a modified nearest neighbor algorithm to solve the Time-dependent ATSP with Time Windows and Precedence Constraints (TD-ATSP-TWPC).

In this work, we implemented the local search algorithms used by Saradatta and Pongchairerks and embedded them into a hyper-heuristic framework, to show that the combined use of heuristics can produce better solutions than their use separately. A hyper-heuristic is a general-purpose problem-independent heuristic search framework which operates a set of low level heuristics to solve computationally hard problems, using only limited information such as the objective function and the direction of the optimization. In each iteration the hyper-heuristic selects a low level heuristic to be applied in the current solution. The new solution is tested in a move acceptance criterion, if approved, the solution becomes the new one and we proceed to the new iteration, otherwise, the new solution is discarded.

The set of low level heuristics was composed by the heuristics used by Saradatta and Pongchairerks in their local search algorithms:

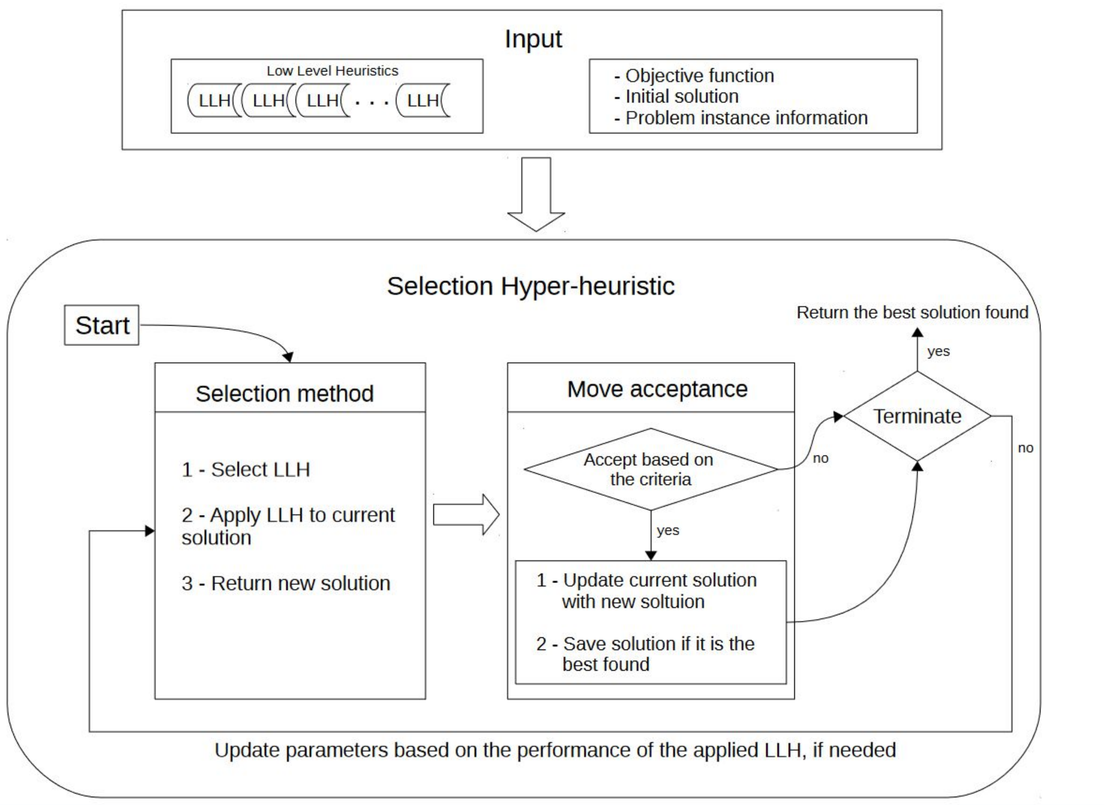
- SWAP: it randomly selects two vertices in the solution and swaps them; and
- INSERT: it randomly selects a vertex and a position in the solution. Then the vertex is removed from the current position and inserted into the selected position. Consequently, all vertices between the old position and the new one are moved.

The heuristic selection methods applied within the hyper-heuristics were:

- **Simple Random (SR)**: it uses a uniform probability distribution to randomly select a low level heuristic at each step;
- **Random Descent (RD):** it selects a low level heuristic randomly and applies it repeatedly as long as an improvement is found;
- **Random Permutation (RP):** it generates a random ordering of the low level heuristics and, at each step, successively applies a low-level heuristic in the provided order;
- **Random Permutation Descent (RPD):** it generates a random ordering of the low-level heuristics and applies each of them repeatedly as long as an improvement is found, respecting the provided order; and
- **Reinforcement Learning (RL):** it assigns an initial score to each low level heuristic in the beginning of the algorithm and adjusts the scores while learning through the iterations. When a low level heuristic improves a solution, its score is updated positively, while a worsening move decreases the score of a low level heuristic.

Finally, we used an **adapted Improve or Equal** deterministic acceptance criterion which only accepts two kinds of solutions - the ones with better or equal cost, or the ones with a slightly worse cost provided they have the same number of artificial edges. For a better comparison, the stopping criterion was based on reaching N*(N-1) iterations without improvement, as done in the local search.

Hyper-heuristic diagram:



Results:

All methods were coded in C++ language and all experiments were executed on a personal computer with an i7-7500U 2.7 GHz Intel processor and 8 GB of RAM memory. Moreover, each method was executed 10 times aiming to get a wider view of its performance.

TABLE 1. Results for the TD-ATSP-TWPC when reproducing the local search methods.

Instance	Method	Best	Worst	Avg.	Std.Dev.(%)†	Avg.Time(s)	Best Known
1	INSERT	**102280**	176970	128559	15.83	0.0037	**102280**
2	SWAP	127209	174103	157778	9.01	0.0039	**115122**
3	INSERT	137722	230970	175672	16.67	0.0036	**133329**
4	SWAP	219975	300723	262077	10.02	0.0087	**196620**
5	INSERT	205049	322688	258241	13.55	0.0081	**194682**
6	SWAP	176333	243099	206925	9.60	0.0084	**149065**

† The average standard deviation is 12.44.

TABLE 2. Results for the TD-ATSP-TWPC when applying our hyper-heuristic framework.

Instance	Method	Best	Worst	Avg.	Std.Dev.(%)†	Avg.Time(s)	Best Known
1	SR	**102280**	139186	121859	9.71	0.0042	**102280**
2	RD	**115122**	164564	138332	11.93	0.0039	**115122**
3	RL	**129547**	174690	150226	10.19	0.0038	133329
4	RP	**191324**	277484	245733	11.18	0.0087	196620
5	RL	**191637**	243495	213667	7.23	0.0084	194682
6	RD	**147192**	213858	176065	12.15	0.0089	149065

† The average standard deviation is 10.39.

The results depicted in Table 2 prove that the combined use of the low level heuristics in our hyper-heuristic framework was able to provide best results and better average solution costs for all instances compared to the local search algorithms used in Table 1. Regarding the high standard deviation values in Table 2, we observed that the results reproducing Saradatta and Pongchairerks experiments (Table 1) also include high values for the standard deviation. Moreover, our hyper-heuristic framework was able to produce a slight improvement, as shown by the average standard deviations 12.44 in Table 1 and 10.39 in Table 2. In conclusion, for solving the TD-ATSP-TWPC, we can affirm that combining low level heuristics within a hyper-heuristic framework leads to much better results.

Conclusion and future work:

We implemented the algorithms used by Saradatta and Pongchairerks and embedded them into a new hyper-heuristic framework to solve the Time-dependent Asymmetric Traveling Salesman Problem with Time Windows and Precedence Constraints, showing that the combined use of heuristics can produce better solutions than their use separately. We were able to find better results in all instances, both in terms of solution quality and computational time.

In our hyper-heuristic framework we used an adapted Improve or Equal deterministic acceptance criterion. Based on the evidences of meeting local minima, we suggest for future work the application of non-deterministic strategies, such as the Monte Carlo-based move acceptance strategies which accept all improving moves and some non-improving moves with a certain probability.

References:

- Thanaboon Saradatta and Pisut Pongchairerks. A time-dependent ATSP with time window and precedence constraints in air travel. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 9(2-3):149–153, 2017.
- Ahmed Kheiri and Edward Keedwell. Selection hyper-heuristics. In Proceedings of the Genetic and Evolutionary Computation Conference Companion, pages 983–996, 2022
- Yaroslav Pylyavskyy, Ahmed Kheiri, and Leena Ahmed. A reinforcement learning hyper-heuristic for the optimisation of flight connections. In 2020 IEEE Congress on Evolutionary Computation (CEC), pages 1–8. IEEE, 2020
- Edmund K. Burke, Michel Gendreau, Matthew Hyde, Graham Kendall, Gabriela Ochoa, Ender ̈Ozcan, and Rong Qu. Hyper-heuristics: a survey of the state of the art. pages 449–468, 2010.