SHORT TERM SCIENTIFIC MISSION (STSM)
— SCIENTIFIC REPORT —

Action number:	CA15140
STSM title:	Extension and Application of the Parameter-less Late Acceptance Hill-climbing
STSM start and end date:	06/11/2017 to 17/12/2017
Grantee name:	Mosab Bazargani

An update note: **(added on 10/05/2020)**	Results of this work are published in the *European Journal of Operational Research* [6] and the paper is Gold Open Access. It also has a companion website available at https://mbazargani.github.io/CCPcutoffTime/.

1 Purpose of the STSM

The purpose of the visit was to extend a research work, entitled "Parameter-less Late Acceptance Hill-climbing" (pLAHC), that I and Fernando Lobo published in GECCO'17. In that paper we combined a *parameter-less* method first introduced by Harik and Lobo [3] in 1999 in the context of genetic algorithms, with the *Late Acceptance Hill-climbing* (LAHC) algorithm, introduced by Burke and Bykov [2] in 2017. Our proposed technique simplifies the application of LAHC by not requiring the user to specify the history length parameter. It makes LAHC, a simple algorithm, even simpler to apply in practice, since it does not need any tuning. We went further and introduced pLAHC with seeding (pLAHC-s) to make pLAHC faster by using information obtained during run time. The method was successfully applied and evaluated on a number of instances of the Travelling Salesman Problem (TSP).

During the proposed STSM we wanted to examine if the results reported in [1] generalize to other problem classes beyond TSP. We planned to apply pLAHC to other popular permutation-based optimization problems that have real-world applications, i.e., the Linear Ordering Problem (LOP), the Permutation Flowshop Scheduling Problem (PFSP), and the Quadratic Assignment Problem (QAP). We used instances from widely adopted benchmark suites, LOLIB for LOP, Taillard benchmark suite for PFSP, and QAPLIB for QAP. In that context, we wanted to study the behaviour of pLAHC and pLAHC-s and analyse the results statistically.

We also planned to apply pLAHC and pLAHC-s to a real-world application, Project Scheduling Problem (PSP) [7]. This problem is related to the more general resource-constrained project scheduling problem, which has a large existing literature and benchmark problem library (PSPLIB).

COST Association AISBL | Avenue Louise 149 | 1050 Brussels, Belgium
T +32 (0)2 533 3800 | F +32 (0)2 533 3890 | office@cost.eu | www.cost.eu
Funded by the Horizon 2020 Framework Programme of the European Union

This research work was planned to carry out within 6 weeks at University of Algarve. The plan included implementing the above mentioned combinatorial problems within the existing C++ code (https://github.com/mbazargani/pLAHC), running experiments, and doing an analysis of the results obtained. The time needed for writing paper(s) was not included in the plan. We planned to do that after my return to London.

2 Description of Work Carried out During the STSMS

I started my research visit by following the plan described in my STSM proposal. We implemented and tested QAP and PFSP; two out of the three problems that we planned to tackle. Once finishing each problem, we ran experiments using instances of the known benchmark. This gave us a wider insight into the algorithm behaviour. Consequently, we extended the work while moving from one problem to another.

We also took time to analyse the behaviour of the original LAHC. We did so by reasoning about the algorithm's steps and complemented that by dumping more data from our runs and analysing them. Doing so lead us to have a better understanding of LAHC.

While discussing the results obtained, we got to the conclusion that we need a more germane way of deciding (automatically) when to restart a given LAHC run with a bigger history list size. So, we spent quite a chunk of my STSM time to formulate a way of doing so that is more sound from a theoretical perspective, with respect to the one used previously in [2, 1].

We derived and tested a method based on the Coupon Collector Problem [8]. We implemented and evaluated its applicability using a separate Python code. The idea looks promising and we embedded it into our C++ code for pLAHC.

Besides reading scientific papers related to the work that we were developing, following Fernando Lobo's suggestion, I read some chapters of the following books that are relevant to my STSM and discussed them with him:

- Stochastic Local Search: Foundations & Applications [4];
- Analyzing Evolutionary Algorithms — The Computer Science Perspective [5];
- Randomized Algorithms [8].

3 Description of the Main Results Obtained

Before applying pLAHC to new permutation-based optimization problems (i.e. QAP and PFSP), we first applied LAHC to them using fixed history list sizes of 1, 5000, and 50000. These sizes are those that were used by Burke and Bykov [2] to study the behaviour of LAHC. The results obtained show a similar behaviour as that reported by Burke and Bykov [2]: the longer the history length, the better solution quality obtained at the expense of time. This reinforces the justification for using the restart mechanism proposed for eliminating the only parameter of the LAHC [1].

The preliminary results that we obtained show that unlike TSP, in QAP, pLAHC without seeding performs better than pLAHC with seeding (pLAHC-s). Our understanding is that this behaviour is caused by the fact that local optimums are closer to each other in TSP than QAP [4]. However, we are still investigating this new founding.

As I explained in the previous section, in our GECCO'17 paper [1], we used the stopping criterion that was used by Burke and Bykov [2] to restart LAHC with doubling history length. They used the following stopping criterion to stop LAHC and called it *point of convergence*: the algorithm halts when the number of consecutive non-improving (idle) iterations reaches 2% over the total number of iterations, and at least 100000 iterations are performed to avoid early termination. The stopping criterion was in part tuned to the problems tackled by Burke and Bykov in the original paper of LAHC [2]. Here I need to stress out that a stopping criterion is usually not considered as an algorithm parameter. However, in the context of pLAHC and pLAHC-s the decision of when to restart with a bigger list size (i.e., the LAHC stopping criterion) can indeed be considered a parameter of the overall strategy. While the 2% of non-improving iterations seems a reasonable choice, the same cannot be said about the requirement of at least 100000 iterations.

Although it has been shown in the literature that those values work well for the problems used in the original paper [2, 1], they were derived on an empirical basis only. Consequently, we came to the conclusion that the restart mechanism of pLAHC and pLAHC-s needs a more sound theoretical basis.

We thought and discussed about this issue and among the various ideas that occurred to us we came up with a method that is based on the Coupon Collector's Problem. Using this method guarantees, with some degree of confidence, that we terminate/restart LAHC once we are in a local optimum. Our preliminary results show that this method is an appropriate replacement for the one that was used by Burke and Bykov, and by ourselves in the GECCO'17 paper.

The details of our finding will be presented in a paper that we are planning to publish.

4 Future Collaborations

As mentioned in the work plan of my STSM application, the time needed for writing paper(s) was not included in my visiting work plan. Since my return to London, I have been working on it, finishing experiments, and will be writing a paper with the results of our research in the near future. Once it's published we will inform the ImAppNIO/STSM evaluation panel.

Last but not least, in my work plan, I also mentioned that we are planning to apply pLAHC to a real-world application, Project Scheduling Problem (PSP) [7]. We could not go that far during my research visit because we ended spending a lot more time on figuring out a more appropriate restart decision for pLAHC. Nonetheless, we still plan to tackle the PSP problem with pLAHC-s in the near future.

References

[1] M. Bazargani and F. G. Lobo. Parameter-less late acceptance hill-climbing. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2017, Berlin, Germany, July 15-19, 2017*, pages 219–226. ACM, 2017.

[2] E. K. Burke and Y. Bykov. The late acceptance hill-climbing heuristic. *European Journal of Operational Research*, 258(1):70–78, 2017.

[3] G. R. Harik and F. G. Lobo. A parameter-less genetic algorithm. In W. Banzhaf et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, pages 258–265, San Francisco, CA, 1999. Morgan Kaufmann.

[4] H. H. Hoos and T. Stützle. *Stochastic Local Search: Foundations & Applications*. Elsevier/Morgan Kaufmann, 2004.

[5] T. Jansen. *Analyzing Evolutionary Algorithms - The Computer Science Perspective*. Natural Computing Series. Springer, 2013.

[6] F. G. Lobo, M. Bazargani, and E. K. Burke. A cutoff time strategy based on the coupon collector's problem. *European Journal of Operational Research*, mar 2020.

[7] L. L. Minku, D. Sudholt, and X. Yao. Evolutionary algorithms for the project scheduling problem: runtime analysis and improved design. In *Genetic and Evolutionary Computation Conference, GECCO '12, Philadelphia, PA, USA, July 7-11, 2012*, pages 1221–1228. ACM, 2012.

[8] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, New York, NY, USA, 1995.

COST Association AISBL | Avenue Louise 149 | 1050 Brussels, Belgium
T +32 (0)2 533 3800 | F +32 (0)2 533 3890 | office@cost.eu | www.cost.eu
Funded by the Horizon 2020 Framework Programme of the European Union