Construct, Merge, Solve & Adapt versus Large Neighborhood Search: Which one works better when?*

Maria J. Blesa, Evelia Lizárraga†
Universitat Politècnica de Catalunya - BarcelonaTech
Barcelona, Spain
{mjblesa,evelial}@cs.upc.edu

Christian Blum
Artificial Intelligence Research Institute (IIIA-CSIC)
Bellaterra, Spain
christian.blum@iiia.csic.es

ABSTRACT

The development and the application of hybrid metaheuristics has enjoyed an increasing popularity in recent years because they often allow to combine the strengths of different ways of solving optimization problems in a single algorithm. Especially the combination of heuristic search with exact techniques has been quite fruitful. One of the most well known algorithms from this field is called *Large Neighborhood Search* (LNS) [2], which is based on the following general idea. Given a valid solution to the tackled problem instance, first, destroy selected parts of it, resulting in a partial solution. Then apply some other, possibly exact, technique to find the best valid solution on the basis of the given partial solution, that is, the best valid solution that contains the given partial solution. Thus, the destruction step defines a *large neighborhood*, from which a best (or nearly best) solution is determined, not by naive enumeration but by the application of a more effective alternative technique.

One of the latest algorithmic developments in this line is labelled *Construct, Merge, Solve and Adapt* (CMSA) [1]. Just like LNS, the main idea of CMSA is to iteratively apply a suitable exact technique to *reduced problem instances* or *sub-instances*, i.e. to a subset of the set of solutions for the tackled problem instance, which is obtained by a reduction of the search space. The idea of both, LNS and CMSA, is to identify substantially reduced sub-instances of a given problem instance such that the sub-instances contain high-quality solutions to the original problem instance. This might allow the application, for example, of an exact technique with reasonable computational effort to the sub-instance in order to obtain a high-quality solution to the original problem instance. In other words, both algorithms employ techniques for reducing the search space of the tackled problem instances.

Although both LNS and CMSA are based on the same general idea, the way in which the search space is reduced differs from one to the other. Based on this difference we had the intuition that LNS would (generally) work better than CMSA for problems for which solutions are rather large, and the opposite would be the case in the context of problems for which solutions are rather small. The size of solutions is hereby measured by the number of solution components (in comparison to the total number) of which they are composed. For example, in the case of the travelling salesman problem, the complete set of solution components is composed of the edges of the input graph. Moreover, solutions consist of exactly n components, where n is the number of vertices of the input graph. The above-mentioned intuition is based on the consideration that, for ending up in some high-quality solution, LNS needs to find a path of over-lapping solutions from the starting solution to the mentioned high-quality solution. The smaller the solutions are, the more difficult it should be to find such a path. A theoretical validation of our intuition seems, a priori, rather difficult to achieve. Therefore, we decided to study empirical evidence that would support (or refute) our intuition. For this purpose, we used the multi-dimensional knapsack problem (MDKP). For this problem it is possible to generate both, problem instances for which solutions are small and problem instances for which solutions are large. We implemented both LNS and CMSA for the MDKP and performed an empirical study of the results of both algorithms for problem instances over the whole range between small and large solutions. The outcome of the presented study is empirical evidence for the validity of our intuition.

CCS CONCEPTS

• **Theory of computation → Discrete optimization**; **Optimization with randomized search heuristics**; **Algorithm design techniques**;

KEYWORDS

Hybrid algorithms, CMSA, LNS, Multi-dimensional Knapsack

ACM Reference format:
Maria J. Blesa, Evelia Lizárraga and Christian Blum. 2017. Construct, Merge, Solve & Adapt versus Large Neighborhood Search: Which one works better when?. In *Proceedings of ACM Celebration of Women in Computing, Barcelona, Spain, September 6-8, 2017 (womENcourage 2017),* 1 pages.
https://doi.org/10.475/123_4

*This work was funded by project TIN2012-37930-C02-02 (Spanish Ministry for Economy and Competitiveness, FEDER funds from the European Union) and project SGR 2014-1034 (AGAUR, Generalitat de Catalunya). An extended version of this work has been published in the 17th European Conference on Evolutionary Computation in Combinatorial Optimisation (EvoCOP 2017). LNCS 10197, pages 60-74. Springer.
†E. Lizárraga acknowledges funding from the Mexican National Council for Science and Technology (CONACYT, doctoral grant number 253787).

womENcourage 2017, September 6-8, 2017, Barcelona, Spain

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

REFERENCES

[1] Christian Blum, Pedro Pinacho, Manuel López-Ibáñez, and José A. Lozano. 2016. Construct, Merge, Solve & Adapt: A new general algorithm for combinatorial optimization. *Computers & Operations Research* 68 (2016), 75–88.

[2] D. Pisinger and S. Ropke. 2010. Large Neighborhood Search. In *Handbook of Metaheuristics*, International Series in Operations Research & Management Science, Vol. 146. Springer US, 399–419.