Hybridizing Different Local Search Algorithms with Each Other and Evolutionary Computation: Better Performance on the Traveling Salesman Problem

Yuezhong Wu, Thomas Weise*, and Weichen Liu
UBRI, School of Computer Science and Technology, University of Science and Technology of China
Hefei, Anhui, China, 230027, tweise@mail.ustc.edu.cn. * Dr. Weise is the corresponding author.

ABSTRACT

We propose the new concept of hybridizing different local search algorithms with each other for the TSP. The new hybrids outperform their component algorithms. We then hybridize them with an Evolutionary Algorithm and Population-based ACO. The resulting EC-LS-LS hybrids perform even better.

Keywords

Local Search; Hybrid Algorithms; Traveling Salesman Problem; Memetic Algorithm; Ant Colony Optimization

1. INTRODUCTION

The Traveling Salesman Problem (TSP) [1] is defined as follows: Given n cities, a salesman departs from a start city, visits each city exactly once, and then returns back to the start city. The task is to find the city visiting order resulting in the minimal overall travel distance. Many algorithms have been applied to the TSP, from Evolutionary Computation (EC) [9] to exact methods like Branch and Bound [4]. We make the following contributions: We introduce the new concept of the LS-LS hybrids (for the TSP), i.e., hybrid algorithms combining two local search (LS) methods. We explore several LS-LS hybrids combining Multi-Neighborhood Search (MNS) [10], the Lin-Kernighan Heuristic (LK) [5], and FSM** [6]. We find that they almost always significantly outperform their LS components. We further hybridize our LS-LS hybrids with both Population-based ACO (PACO) [2] and Evolutionary Algorithms (EAs) [9]. We find that the resulting EC-LS-LS algorithms perform better than any other method in our experiments and in [6, 10, 12].

2. INVESTIGATED ALGORITHMS

LS algorithms for the TSP start at a random or heuristically-generated solution (tour). They remember the best solution discovered so far and try to improve it step by step. If a local optimum is reached, the LS applies a larger random modification. MNS is a LS that, in each iteration, scans several neighborhoods of the current solution and enqueues all possible improving moves. The best move is carried out. All invalidated intersecting moves are dropped and the remaining best move is applied. If the queue becomes empty, another scan is performed. If no improving moves can be found, a random sub-sequence of the current tour is randomly shuffled, which we refer to as soft restart. The LK heuristic dominates today's TSP research. We use the implementation from [12]. The Ejection Chain Method (ECM) FSM** [6] is an improvement of [7]. Although both LK and FSM** outperform MNS, the hybrids of MNS with Evolutionary Algorithms (EAs) and the Population-based Ant Colony Optimization (PACO) outperform similar hybrids based on them [6, 12].

Research on hybrid ("Memetic") algorithms is almost entirely focused on combining global and local search algorithms (EC-LS), as done in [6, 10, 12, 13]. However, LS algorithms can already exhibit different behaviors which might complement each other. MNS can find relatively good solutions quickly but often gets stuck in local optima. LK and FSM** initially are slower but find better final results [6, 12]. We pairwise hybridize LK, FSM**, and MNS with each other. We apply one LS approach until it cannot improve the solution anymore. The resulting tour is then passed as starting point to the other LS. Once this second LS gets trapped in a local optimum, we use its result as starting point again for the first LS. This is repeated until both LS methods cannot find an improvement, in which case we apply the same soft restart method as above. This is a generalization of Variable Neighborhood Search (VNS) [3], which explores the neighborhood of the current solution until it reaches a local optimum. It then uses another search operator to escape from it. With our new LS-LS hybrids, we extend the concept to whole LS algorithms instead of just search operators.

We investigate MA($\mu \overset{+}{,} \lambda$)-LS and MA($\mu \overset{+}{,} \lambda$)-LS-LS, which combine ($\mu \overset{+}{,} \lambda$) EAs with LS and LS-LS, respectively. The first populations of our MAs stem from the Edge-Greedy, Double Minimum Spanning Tree, Savings, Double-Ended Nearest Neighbor, and Nearest Neighbor Heuristic [10]. We apply Edge Crossover [11] at a crossover rate of 1. The LS component of the MA is applied to every solution generated, both by the initialization heuristics and crossover. We also hybridize PACO(k,l) with LS and LS-LS. Here, in each iteration, l tours are created in the same way as in the standard ACO. The "oldest" solution in the archive of size k is replaced by the best of the newly gener-

GECCO '16 July 20-24, 2016, Denver, CO, USA

ACM ISBN 978-1-4503-4323-7/16/07.
DOI: http://dx.doi.org/10.1145/2908961.2909001

PACO(3,10)-LK10-MNS *(rank 1)*, PACO(3,25)-LK10-MNS *(2)*, PACO(5,10)-LK10-MNS *(3)*, MA(16+64)-FSM**-LK10 *(4)*, PACO(5,10)-FSM**-LK10 *(5)*, PACO(3,25)-FSM**-LK10 *(6)*, MA(16,64)-FSM**-LK10 *(7)*, PACO(3,10)-FSM**-LK10 *(8)*, MA(16,64)-LK10-MNS *(9)*, MA(2,8)-LK10-MNS *(10)*, MA(16+64)-LK10-MNS *(11)*, MA(2,4)-LK10-MNS *(12.5)*, MA(2,8)-FSM**-LK10 *(12.5)*, MA(2+8)-FSM**-LK10 *(14)*, MA(2+4)-FSM**-LK10 *(15.5)*, PACO(5,10)-MNS *(15.5)*, MA(2+4)-LK10-MNS *(17.5)*, MA(2,4)-FSM**-LK10 *(17.5)*, PACO(3,10)-MNS *(19.5)*, PACO(3,25)-MNS *(19.5)*, MA(2+8)-LK10-MNS *(21)*, PACO(3,10)-FSM** *(22)*, PACO(5,10)-FSM** *(23)*, PACO(3,25)-FSM** *(24)*, MA(2+8)-FSM** *(25)*, MA(2+4)-FSM** *(26)*, MA(16,64)-FSM** *(27)*, MA(16+64)-FSM** *(28)*, MA(2,4)-FSM** *(29)*, MA(2,8)-FSM** *(30)*, MA(16+64)-MNS *(31)*, MA(2+4)-MNS *(32)*, MA(2+8)-MNS *(33)*, MA(16,64)-MNS *(34)*, MA(2,8)-MNS *(35)*, PACO(3,10)-LKn *(36)*, PACO(3,25)-LKn *(37)*, PACO(5,10)-LKn *(38)*, LK10-MNS *(39)*, FSM**-LK10 *(40)*, FSM**-LK5 *(41)*, FSM**-LK20 *(42)*, LK10-FSM** *(43)*, MA(2,4)-MNS *(44)*, LK5-FSM** *(45)*, FSM**-LK30 *(46)*, FSM**-MNS *(47)*, LK5-MNS *(48)*, FSM**-LK40 *(49)*, FSM**-LKn *(50)*, LK20-MNS *(51)*, LK20-FSM** *(52)*, LK20 *(53)*, LK30 *(54)*, LK40-MNS *(55)*, LK30-MNS *(56)*, LK30-FSM** *(57)*, LK10 *(58)*, LK40 *(59)*, FSM** *(60)*, LK40-FSM** *(61)*, LK5 *(62)*, MA(16+64)-LKn *(63)*, LKn-FSM** *(64)*, MA(16,64)-LKn *(65)*, LKn-MNS *(66)*, MA(2,8)-LK *(67)*, MA(2+4)-LK *(68)*, MA(2,4)-LK *(69)*, LKn *(70)*, MA(2+8)-LKn *(71)*, MNS-LK40 *(73)*, MNS-LK30 *(73)*, MNS-LK20 *(73)*, MNS-FSM** *(75)*, MNS-LK10 *(76)*, MNS-LK5 *(77)*, MNS-LKn *(78)*, and MNS *(79)*.

Figure 1: Algorithm ranking from best to worst, based on various performance measures and statistics (see [10] for details). The different algorithm types pure local search, LS-LS hybrid, EC-LS hybrid, and EC-LS-LS hybrid are highlighted.

ated solutions. The pheromone on an edge is proportional to the number of times the edge is contained in the archive. The initial populations are again obtained heuristically, in the same way as in the MAs.

3. RESULTS AND DISCUSSION

We perform 30 independent runs for 79 algorithm setups on all 110 symmetric *TSPLib* [8] benchmark cases. We define the following LS setups: FSM**, MNS, and 6 setups of the LK heuristic, differing in their candidate set size $s = \in \{5, 10, 20, 30, 40, n\}$ and named LKs, i.e., LK5, LK10, ..., LKn, respectively. We compare them to 26 LS-LS hybrids, whose name consists of the two component LS algorithms in the cyclically applied sequence. We construct EC-LS and EC-LS-LS hybrids which are named after the EC method followed by the applied LS. In Figure 1, we present an abridge algorithm performance ranking generated by the *TSP Suite* [10].Our experiments have led us to four major conclusions:

1. The new LS-LS hybrids are better than their pure LS algorithm components. This means that the new idea of combining the strengths of different LS algorithms is very promising.
2. The LS-LS hybrids are still slightly worse than the EC-LS algorithms. This means that a global search component is necessary in a good TSP solver.
3. The new EC-LS-LS hybrids outperform the LS-LS algorithms as well as EC-LS hybrids. The best overall algorithm, PACO(3,10)-LK10-MNS, unites the global search strength of PACO, the ability to find good solutions of the LK heuristic, and the fast exploitation speed of MNS.
4. In [6, 10, 12, 13] as well as the present study, PACO is the significantly better host EC method for hybridization than an EA. However, we also find an exception to this rule, as MAs with FSM**-LK10 are better than the corresponding PACO versions.

Acknowledgments We acknowledge support from the Fundamental Research Funds for the Central Universities and the National Natural Science Foundation of China under Grants 61150110488 and 71520107002.

References

[1] D. L. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook. *The Traveling Salesman Problem: A Computational Study*. Princeton Series in Applied Mathematics. Princeton, NJ, USA: Princeton University Press, 2007.

[2] M. Guntsch and M. Middendorf. Applying population based aco to dynamic optimization problems. In *3rd Intl. Workshop on Ant Colony Optimization (ANTS'02)*, pages 111–122, Brussels, Belgium, Sept. 12–14, 2002. Berlin, Germany: Springer-Verlag GmbH.

[3] P. Hansen and N. Mladenović. Variable neighborhood search: Principles and applications. *Europ. Journal of Operational Res. (EJOR)*, 130(3):449–467, 2001.

[4] Y. Jiang, T. Weise, J. Lässig, R. Chiong, and R. Athauda. Comparing a hybrid branch and bound algorithm with evolutionary computation methods, local search and their hybrids on the tsp. In *IEEE Symposium on Computational Intelligence in Production and Logistics Systems (CIPLS'14)*, Orlando, FL, USA, Dec. 9–12, 2014. Los Alamitos, CA, USA: IEEE Computer Society Press. doi:10.1109/CIPLS.2014.7007174.

[5] S. Lin and B. W. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21(2):498–516, Mar.–Apr. 1973.

[6] W. Liu, T. Weise, Y. Wu, and R. Chiong. Hybrid ejection chain methods for the traveling salesman problem. In *10th Intl. Conf. Bio-Inspired Computing – Theories and Applications (BIC-TA'15)*, volume 562 of *Communications in Computer and Information Science*, pages 268–282, Hefei, Anhui, China, Sept. 25–28, 2015. Berlin/Heidelberg: Springer-Verlag. doi:10.1007/978-3-662-49014-3_25.

[7] C. Rego. Relaxed tours and path ejections for the traveling salesman problem. *European Journal of Operational Research*, 106(2):522–538, 1998.

[8] G. Reinelt. Tsplib – a traveling salesman problem library. *ORSA Journal on Comp.*, 3(4):376–384, 1991.

[9] T. Weise. *Global Optimization Algorithms – Theory and Application*. Germany: it-weise.de (self-published), 2009. URL http://www.it-weise.de/projects/book.pdf.

[10] T. Weise, R. Chiong, K. Tang, J. Lässig, S. Tsutsui, W. Chen, Z. Michalewicz, and X. Yao. Benchmarking optimization algorithms: An open source framework for the traveling salesman problem. *IEEE Computational Intelligence Magazine (CIM)*, 9(3):40–52, Aug. 2014. doi:10.1109/MCI.2014.2326101.

[11] L. D. Whitley, T. Starkweather, and D. Fuquay. Scheduling problems and traveling salesman: The genetic edge recombination operator. In *3rd Intl. Conf. on Genetic Algorithms (ICGA'89)*, pages 133–140, Fairfax, VA, USA, June 4–7, 1989. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

[12] Y. Wu, T. Weise, and R. Chiong. Local search for the traveling salesman problem: A comparative study. In *Proc. of 14^{th} IEEE Conf. on Cognitive Informatics & Cognitive Computing (ICCI*CC'15)*, pages 213–220, Beijing, China, July 6–8, 2015. doi:10.1109/ICCI-CC.2015.7259388.

[13] D. Xu, T. Weise, Y. Wu, J. Lässig, and R. Chiong. An investigation of hybrid tabu search for the traveling salesman problem. In *10th Intl. Conf. Bio-Inspired Computing – Theories and Applications (BIC-TA'15)*, volume 562 of *Communications in Computer and Information Science*, pages 523–537. Berlin/Heidelberg: Springer-Verlag, Hefei, Anhui, China, Sept. 25–28, 2015. doi:10.1007/978-3-662-49014-3_47.