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

Yuezhong Wu, Thomas Weise, Weichen Liu

yuezhong@mail.ustc.edu.cn tweise@ustc.edu.cn lwc@mail.ustc.edu.cn

GECCO JULY 2016

Benchmarking of Optimization Algorithm: The TSP Suite

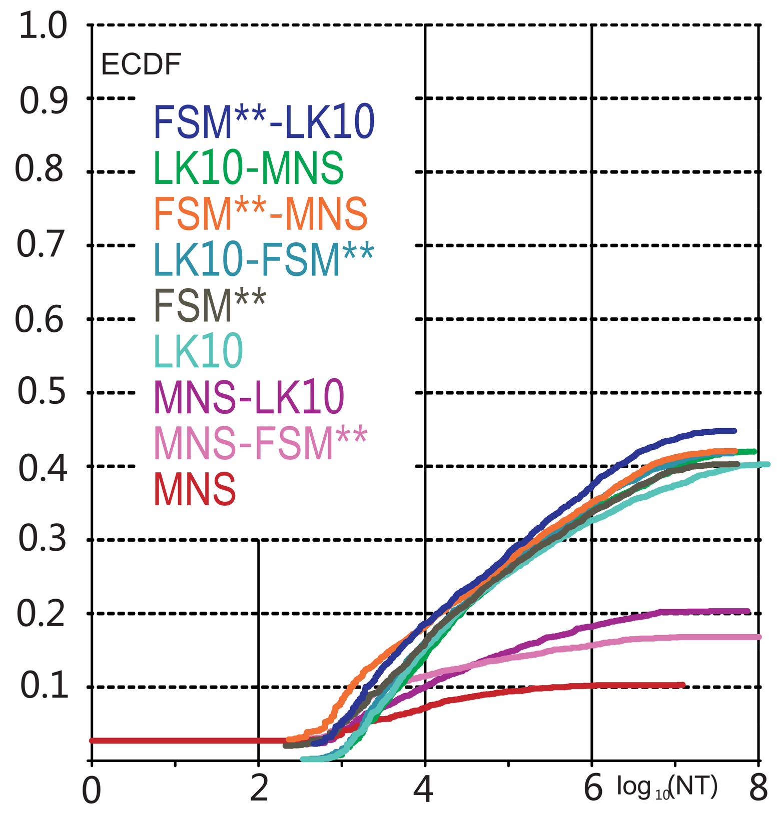
~ The TSP Suite [1] is a framework developed at UBRI for assessing the performance of TSP solvers.

~ Download: http://github.com/optimizationBenchmarking/tspSuite/, http://www.logisticplanning.org/

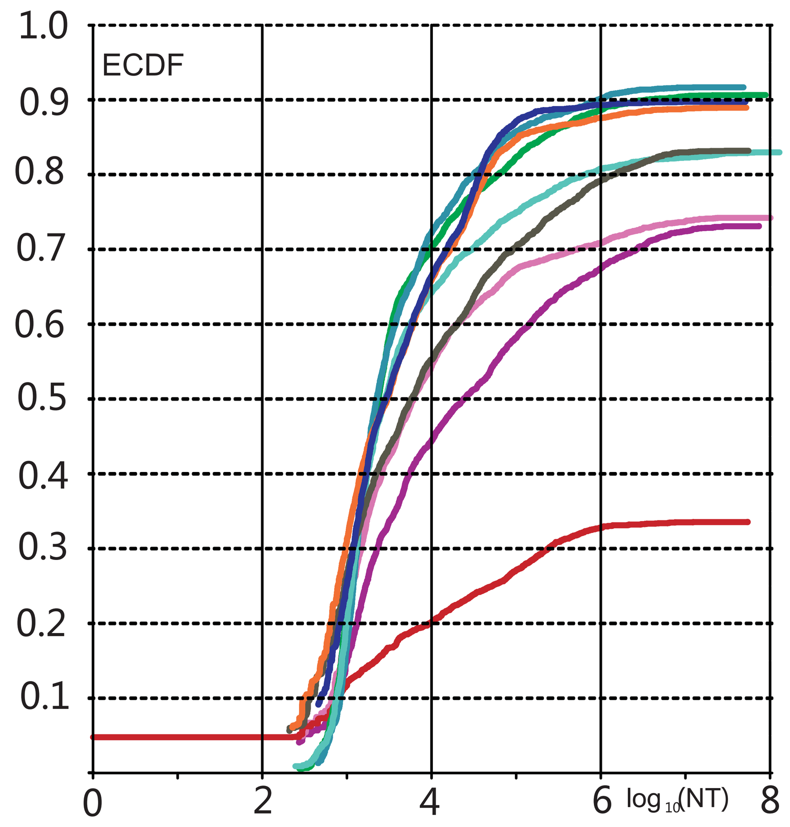
~ Traditional experimentation only focusing on final results is not suitable to benchmark metaheuristics or anytime algorithms: Analysis on progress over time is needed.

TSP Suite is a holistic framework implementing this idea:

~ automated execution of experiments and gathering of benchmark data
~ automated parallelization and distribution on a cluster
~ automated evaluation of gathered data and comparison and ranking of runtime behavior algorithms according to several different statistics.
~ evaluation generates reports similar to small theses or journal articles with performance figures and tables
~ written in Java, platform independent
~ JUnit test for debugging TSP solvers
~ flexible in building new algorithms, many builtin algorithms

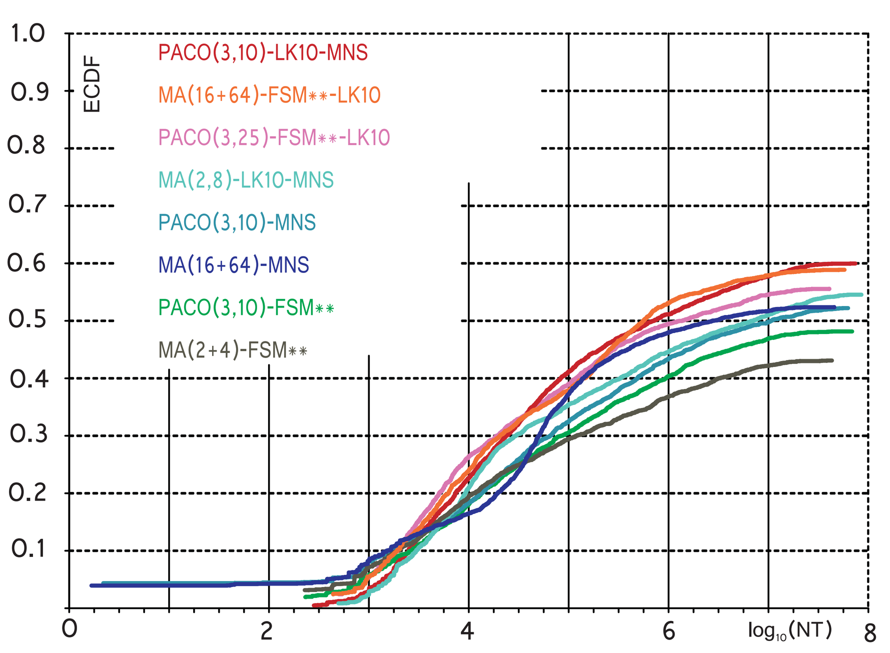


(a) this figure shows the ecdf over the logarithmic normalized time with goal error F = 0

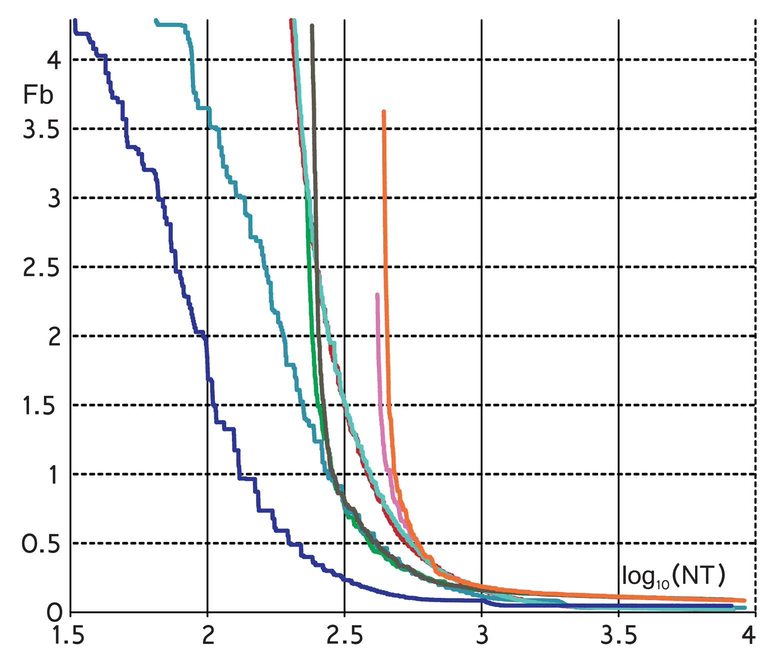


(b) this figure shows the ecdf over the logarithmic normalized time with goal error F = 0.025

Figure 1: figures show the ecdf over the logarithmic normalized time with different goal errors F of LS-LS hybrids and LS approaches



(a) this figure shows the ecdf over the logarithmic normalized time with goal error F = 0



(b) this figure shows the quality of the best solution discovered so far of instances with 16384 < n < 32768

Figure 2: figures show the ecdf change and progress over the logarithmic normalized time of the EC-LS and EC-LS-LS hybrids

Example Investigation[3]: Hybrid Local Search-Local Search algorithms and Hybrid Evolutionary Computation-Local Search-Local Search algorithms

Local Search applied on TSP

~ keep improving one current solution by applying modification until it can not
~ local search investigated here
  ~ a modified Lin-Kernighan (LK) heuristic from [2]
  ~ Multi-Neighborhood Search (MNS) from [1]
  ~ FSM** (an improved ejection chain method from [4])

Local Search (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 heuristic [2] and FSM** [4] initially are slower but find better final results
~ FSM** is considered to have the ability to reach parts of search space which cannot be reached by LK heuristic

New hybrid LS-LS Algorithms

~ LK heuristic, MNS and FSM** are used as LS components in hybrid LS-LS algorithms
~ pairwise hybridize the three LS with each other
~ one LS approach is applied 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, its result is served as starting point again for the first LS. This is repeated until both LS methods cannot find an improvement, in which case the same soft restart method described in [2] is applied.
~ is a generalization of variable neighborhood search (VNS). LS-LS hybrids extends the concept of search operators to whole LS algorithms.

Evolutionary Algorithms as host methods in new EC-LS-LS hybrids:

~ Evolutionary Algorithm (EA) which uses edge crossover.
~ Population-based Ant Colony Optimization (PACO)
  ~ maintain a set of K solutions
  ~ edges in those solutions define the pheromones
  ~ in each iteration, m solutions are generated and replaced the oldest ones in the population

New hybrid EC-LS-LS Algorithms

~ In [3], some EC-LS hybrids are investigated, now new hybrid EC-LS-LS algorithms is investigated by just replacing LS as LS-LS in the hybrids.
~ MA-LS-LS, is a hybrid of EA and LS-LS algorithm. LS-LS algorithm is applied on each newly generated solution in EA.
~ PACO-LS-LS, is a hybrid of PACO and LS-LS algorithm. LS-LS algorithm is applied on each newly generated solution in PACO.
~ The first population of both MA-LS-LS and PACO-LS-LS is generated heuristically by several tour construction algorithms [1].

Detailed analysis and comparison are performed on all methods mentioned above

~ The new LS-LS hybrids are better than their LS components which means the new idea of combining different strengths of different LS algorithms is very promising
~ LS-LS hybrids are still worse than EC-LS hybrids which shows a good global search algorithm is necessary for a good hybrid
~ The PACO-LS-LS hybrids outperform all other algorithms mentioned above as well as in the paper
~ The best overall algorithm is PACO(3,10)-LK10-MNS which unites the global search strength of PACO, the ability to find good solutions of the LK heuristic, and the fast exploitation speed of MNS
~ In general, PACO is a better host EC method than an EA [1, 2]. However, we find an exception to this rule, as MAs with FSM**-LK10 are better than the corresponding PACO versions.
~ For more specific analyses, patterns shown in these algorithms, please refer to the paper

Overall ranking from best to worst (left to right and up to bottom, grouped by pure local search, LS-LS hybrid, EC-LS hybrid and EC-LS-LS hybrid)

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), **ACO(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)

[1] Thomas Weise, Raymond Chiong, Ke Tang, Jörg Lässig, Shigeyoshi Tsutsui, Wenxiang Chen, Zbigniew Michalewicz, and Xin Yao. Benchmarking Optimization Algorithms: An Open Source Framework for the Traveling Salesman Problem. IEEE Computational Intelligence Magazine (CIM), 9(3):40-52, August 2014. Featured article and selected paper at the website of the IEEE Computational Intelligence Society (http://cis.ieee.org/)

[2] Yuezhong Wu, Thomas Weise, and Raymond Chiong. Local Search for the Traveling Salesman Problem: A Comparative Study. In Proceedings of the 14th IEEE Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC'15), July 6-8, 2015, Beijing, China, pages 213-220, ISBN: 978-1-4673-7289-3.

[3] Yuezhong Wu, Thomas Weise, and Weichen Liu. Hybridizing Different Local Search Algorithms with Each Other and Evolutionary Computation: Better Performance on the Traveling Salesman Problem. In Proceedings of the 18th Genetic and Evolutionary Computation Conference (GECCO'16), Denver, Colorado, USA, July 20-24, 2016, New York, NY, USA: Association for Computing Machinery (ACM). accepted for publication as short paper

[4] Weichen Liu, Thomas Weise, Yuezhong Wu, and Raymond Chiong. Hybrid Ejection Chain Methods for the Traveling Salesman Problem. In Proceedings of the 10th International Conference on Bio-Inspired Computing – Theories and Applications (BIC-TA'15), Maoguo Gong, Linqiang Pan, Tao Song, Ke Tang, and Xingyi Zhang, editors, September 25-28, 2015, Hefei, Anhui, China, volume 562 of Communications in Computer and Information Science. Berlin/Heidelberg: Springer-Verlag, pages 268-282, ISBN 978-3-662-49013-6.

Thomas Weise is a Assoc. Prof. working in USTC supervising Yuezhong Wu and Weichen Liu. The logo of USTC is displayed here.

Open Source Software for comparing arbitrary optimization algorithms on arbitrary optimization problems. The power of TSP Suite, for your research. Java and Dockerized: Should run on any system. Produces LaTeX reports.

http://optimizationBenchmarking.github.io or http://www.optimizationBenchmarking.org