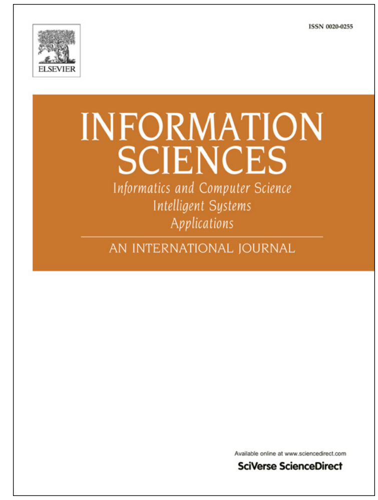

A Variable Iterated Greedy Algorithm for the Traveling Salesman Problem with Time Windows

Korhan Karabulut, M. Fatih Tasgetiren

PII: S0020-0255(14)00424-1
DOI: http://dx.doi.org/10.1016/j.ins.2014.03.127
Reference: INS 10805

To appear in: *Information Sciences*

Received Date: 10 April 2012
Revised Date: 24 February 2014
Accepted Date: 29 March 2014

Please cite this article as: K. Karabulut, M. Fatih Tasgetiren, A Variable Iterated Greedy Algorithm for the Traveling Salesman Problem with Time Windows, *Information Sciences* (2014), doi: http://dx.doi.org/10.1016/j.ins.2014.03.127

A Variable Iterated Greedy Algorithm for the Traveling Salesman Problem with Time Windows

Korhan Karabulut
Software Engineering Department
Yasar University
Selcuk Yasar Campus, Izmir, Turkey
korhan.karabulut@yasar.edu.tr

M. Fatih Tasgetiren
Industrial Engineering Department
Yasar University
Selcuk Yasar Campus, Izmir, Turkey
fatih.tasgetiren@yasar.edu.tr

Abstract—This paper presents a variable iterated greedy algorithm for solving the traveling salesman problem with time windows (TSPTW) to identify a tour minimizing the total travel cost or the makespan, separately. The TSPTW has several practical applications in both production scheduling and logistic operations. The proposed algorithm basically relies on a greedy algorithm generating an increasing number of neighboring solutions through the use of the idea of neighborhood change in variable neighborhood search (VNS) algorithms. In other words, neighboring solutions are generated by destructing a solution component and re-constructing the solution again with variable destruction sizes. In addition, the proposed algorithm is hybridized with a VNS algorithm employing backward and forward 1_Opt local searches to further enhance the solution quality. The performance of the proposed algorithm was tested on several benchmark suites from the literature. Experimental results confirm that the proposed algorithm is either competitive to or even better than the best performing algorithms from the literature. Ultimately, new best-known solutions are obtained for 38 out of 125 problem instances of the recently proposed benchmark suite, whereas 15 out of 31 problem instances are also further improved for the makespan criterion.

Keywords-traveling salesman problem with time windows, iterated greedy algorithm, variable neighborhood search, heuristic optimization

1. Introduction

The traveling salesman problem with time windows (TSPTW) addresses finding a minimum cost-tour that starts and ends at a given depot and features exactly one visit to a given set of customers. Each customer has a service time and a time window defining its ready time and due date. Each customer must be visited before its due date minus service time for a feasible tour. Otherwise, the tour is considered infeasible. On the other hand, if the vehicle arrives before the customer ready time, it must wait. The TSPTW can be modeled as a routing or a scheduling problem. For routing tasks, the objective is to find a route to visit a number of customers, starting and ending at a depot with the constraint that each customer must be visited in a time window. Typically, in this case, the objective function is the cost of a tour, which is, in fact, the total distance traveled. In addition to above, the TSPTW is equivalent to modeling the problem of scheduling jobs on a single machine where setup times are sequence dependent, and each job has a release and due date. In this case, the objective function is to minimize the tour-completion time, the so-called makespan.

The TSPTW has been demonstrated to be NP-Hard. In addition, even finding a feasible solution is an *NP*-complete problem **[1].** Solution methods for the TSPTW can be categorized into exact and heuristic approaches. The first exact algorithms for the TSPTW, developed by Christofides **[2]** and Baker **[3]**, were branch-and-bound algorithms, and the main focus was on the makespan minimization for solving instances up to 50 nodes. However, they were restricted to tight time windows or mostly overlapping windows. Both makespan and travel-cost optimization were considered in Langevin et al. **[4]**, where a two-commodity flow formulation was presented within a branch-and-bound procedure that was able to solve instances up to 40 nodes. Dumas et al. **[5]** extended earlier dynamic programming approaches through the use of a state-space-reduction technique that was able to solve instances up to 200 customers. In addition to the above, Ascheuer et al. **[6]** considered a branch-and-cut algorithm for solving the asymmetric TSPTW, and Balas and Simonetti **[7]** developed a dynamic programming algorithm for various TSP variants with precedence constraints, including the TSPTW. Constraint programming