Tabu Search based solution methods for scheduling log-trucks

Patrick Hirsch and Manfred Gronalt

BOKU – University of Natural Resources and Applied Life Sciences, Institute of Production Economics and Logistics, Feistmantelstrasse 4, 1180 Wien, Austria

Abstract. We present Tabu Search based solution heuristics for the Timber Transport Vehicle Routing Problem (TTVRP) that differ with respect to solution space. The TTVRP is characterized as follows: a fleet of *m* log-trucks which are situated at the respective homes of the truck drivers has to fulfil *n* transports of round timber between different wood storage locations and industrial sites. All transports are carried out as full truck loads. Since the full truck movements are predetermined our objective is to minimize the overall distance of empty truck movements. In addition to the standard VRP we have to consider weight constraints at the network, multi-depots, and time windows. Altogether four Tabu Search variants have been developed for the TTVRP and are presented here. The first one computes with the full neighborhood of a solution in each iteration step, whereas the second variant reduces the neighborhood and therefore accelerates the search process. The third variant uses an oscillating neighborhood size depending on the number of the iteration step. The fourth variant was developed to solve big problem instances within acceptable computing times. It is based on the second variant but uses an additional technique to broaden the search process. A post-optimization heuristic was developed and can be applied to all Tabu Search variants. The presented methods were verified with extensive numerical studies.

1 Introduction and problem description

An important challenge in wood flow planning is the optimization of transports between harvest areas and industrial sites. This problem is described in the literature as Timber Transport Vehicle Routing Problem (TTVRP) (see e.g. Karanta et al. (2000)) and Log-Truck Scheduling Problem (LTSP) (see e.g. Palmgren et al. (2004)). The TTVRP can be characterized as follows: a fleet of *m* trucks which are situated at the respective homes of truck drivers has to fulfill *n* transports of round timber between different wood storage locations and industrial sites, like pulp mills and sawmills, during a given time period. All transports are carried out as full truck loads; the vehicle is loaded at the wood storage location and unloaded at the industrial site. Each tour starts at the home of the truck driver who leaves with an empty truck to his first wood storage location to load round timber. After this he drives to the belonging industrial site and completes the transport. The truck driver can now finish his tour and return back home or start a new transport. Since the full truckload movements are predetermined they are considered as tasks in the rest of the paper. The following constraints must be taken into consideration: Some parts of the forest road networks are not suitable for bigger trucks due to weight restrictions. Therefore some wood storage locations can only be reached by trucks with a certain capacity. This is denoted as route weight constraints. There are time windows at the industrial sites which have predefined opening hours at their intakes. Time windows also occur at the truck starting points since truck drivers are only on duty at certain times. Additionally tour length constraints and capacity constraints have to be observed. The aim is to minimize the overall distance of empty truck movements.

The TTVRP is related to the Multi Depot Vehicle Routing Problem with Pickup and Delivery and Time Windows; supplementary one has to deal with specific route weight constraints and full truck loads. Gronalt et al. (2003) describe the TTVRP as a special application of full truck load scheduling problems. An overview of Vehicle Routing Problems can be found for example in Toth and Vigo (2002).

The modeling approach is based on the perception of the TTVRP as a special case of a Stacker Crane Problem (SCP). The SCP is a sequencing problem that consists of finding the minimum cost cycle on a mixed graph with arcs and edges $G = (V, E, A)$. The predefined arcs correspond to the tasks. A feasible solution must include all arcs. Since empty truck movements are also arcs all nodes are endpoint of at least one arc. Coja-Oghlan et al. (2004) give an example of a SCP which describes the scheduling of a delivery truck.

In Figure 1, a small example is presented to illustrate the TTVRP. Two log-trucks have to perform eight transport tasks. The log-trucks are situated at the home-locations *A* and *B*, respectively. Wood is provided at six different wood storage locations (*P1*, …, *P6*) and must be transported to three industrial sites (*I1*, …, *I3*). *I1* receives three loads: two from *P1* and one from *P2*. The first part of Figure 1 shows the required tasks and demonstrates the problem of linking these transports in a cost-efficient manner, taking into account the above-mentioned constraints. We can observe two kinds of tasks, so-called artificial tasks (*A'*, *B'*) and transport tasks (*1*, …, *8*). The artificial tasks are introduced in order to connect the starting point and endpoint of a cycle. The direct connection between two vertices is always the shortest one. It is impossible to transport directly from one wood storage location to another or from one industrial site to another. This is because we have to deal with full truckloads. The second part of Figure 1 shows the cost-optimal solution for this problem. *A1* is the first trip taken from the log-truck situated at *A*, *A2* is the second one, etc.; the same is true for *B1* to *B11*. Altogether, 8 transports and 10 empty truck movements are scheduled.

The respective model is presented in Gronalt and Hirsch (2006). It was validated for small instances, using the standard solver Xpress. For real-life problems we developed the Tabu Search based solution methods that are discussed in Section 2. We have made extensive numerical studies to test the algorithms; the results and a conclusion are presented in Section 3.

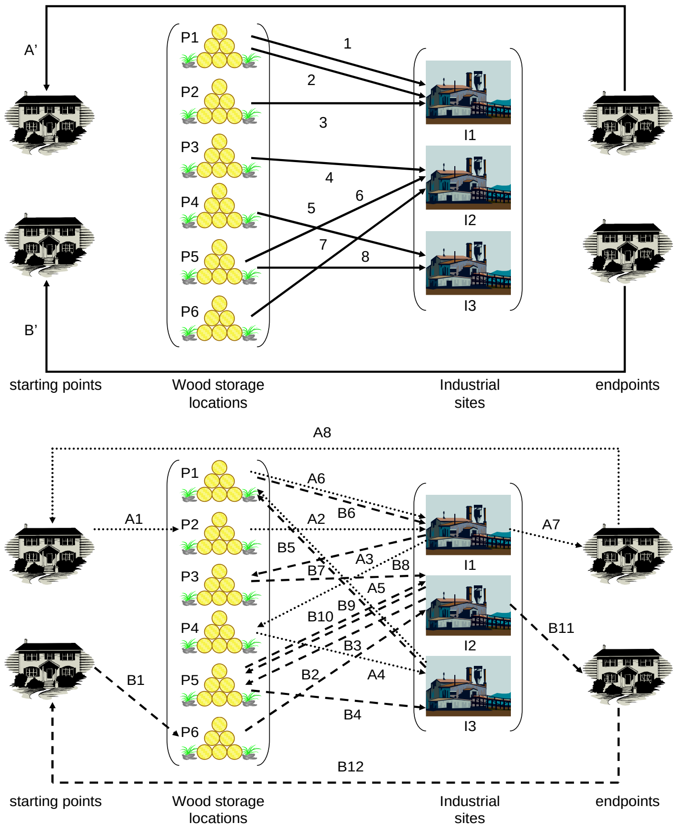

Fig. 1. Example of a TTVRP

2 Heuristic solution approach

The Tabu Search metaheuristic is applied successfully to many VRP problem instances. Cordeau et al. (2001) use a Unified Tabu Search method to solve PVRPTWs and MDVRPTWs, which have a similar structure as the TTVRP. Their method is adapted and modified in this work; it is combined with a greedy initial solution procedure and a post-optimization heuristic. Toth and Vigo (2003) proposed the Granular Tabu Search in order to restrict the neighborhood of solutions drastically and reduce computing times. Their approach was used as thought-provoking impulse for the different Tabu Search procedures.

The heuristic solution approach for the TTVRP consists of the following steps:

1. Restrict the solution space.
2. Find an initial solution with a greedy heuristic, namely a regret-heuristic.

3. Find an improved solution by applying one of the following Tabu Search procedures:
   a. Standard Tabu Search (TS)
   b. Tabu Search with a limited neighborhood (TSLN)
   c. Tabu Search with an alternating strategy (TSAS)
   d. Tabu Search for large problem instances (TSLP)
4. Apply a post-optimization heuristic based on 2opt.

The overall heuristic commences with a reduction of the solution space. In this first step, it is guaranteed that a truck r is only assigned to a task i that can be handled by this truck with respect to the truck capacity and the route weight limit. A regret-heuristic is used as a greedy method to construct an initial solution. Regret-values can be interpreted as costs that appear if the second-best option is taken instead of the best one. The gained solution may violate the duration- and time window constraints.

To describe a solution s the attributes B_{ir} which specify if task i is part of route r are used; if $B_{ir} = 0$ task i is not on route r; if $B_{ir} > 0$ the integer gives the position of task i on route r. Each solution s has cost $c(s)$. During each iteration the non-tabu neighborhood of the current solution is investigated. The neighborhood is defined in the following way: for every attribute $B_{ir} > 0$ in the current solution, task i is inserted into all possible routes $R_p \sim r$ on the position with the least additional costs and removed in route r; that means a move-operator is applied. The tabu duration θ depends on the problem size. The function $f(s)$ is equal to $c(s)$ plus the weighted violations of the constraints. The evaluation function $g(s)$ is used to determine the best neighbor solution; it is equal to $f(s)$ plus a penalty $p(s)$ if the value of $f(s)$ in the neighbor solution is bigger than the original value of $f(s)$ gained in the last iteration or equal to $f(s)$ otherwise. The value of $p(s)$ depends on a weighting parameter λ, the costs of the neighbor solution, the size of the problem, and the number of solutions the attributes of this neighbor solution were part of; this is counted with ρ_{ir}. The weighting factors are updated at the end of each iteration; they are increasing if the constraints were violated in the solution of this iteration and decreasing if they were met. A parameter δ is used to calculate these updates. If the solution of an iteration is feasible and better than the currently best found feasible solution, it is stored. This approach is denoted as TS.

The TSLN concentrates on a certain fraction of empty truck movements in the current solution s; only these links are to be removed in neighbor solutions. Other links can only be modified if a task from a removed link is inserted between their starting and ending points. The procedure functions in the following way: The empty truck movements are first sorted according to their duration in descending order. Then, a predefined number of links is chosen starting from the one with the longest duration. The number of used links is calculated as a fraction of all the existing empty truck movements; the divider D is set as a parameter. If $D = 4$ this means that one fourth of all the links of a solution s is taken away for being removed in neighborhood solutions. This strategy seems to be myopic since “shorter” links are unaffected directly. It leads to a drastic reduction of the computing time but as the numerical studies show it offers a worse solution quality than TS.

To overcome the myopia of TSLN, the TSAS was developed. The TSAS is a fusion of the TS and the TSLN. After a predefined number of iteration steps with a restricted neighborhood, an iteration step with a full neighborhood search is set. The parameter A is used to define which iteration steps will be computed with a full neighborhood search. For example, a setting of $A = 8$ means that in every eighth iteration step a full neighborhood search is performed. This new strategy leads to a drastic reduction of the computing time compared to the TS with no, or only minimal, losses in solution quality.

The TS, TSLN, and TSAS are capable to compute common real-life problem instances with 10 log-trucks and 30 transport tasks in reasonable computing times. A large problem instance with 80 log-trucks and 250 transport tasks was developed to test an extreme case. For this problem instance a full neighborhood search is extremely time-consuming and requires a lot of memory. Thus the TS and TSAS are unsuitable to solve these instances. The TSLN can handle these problem sizes, but seems to be too myopic. Therefore the TSLN was adapted to broaden its search strategy. The outcome is denoted as TSLP. The TSLP uses a slightly modified initial solution heuristic. It is not allowed to assign more than 4 transport tasks to a tour. This helps to meet the capacity constraints and time windows, but does not guarantee a feasible initial solution. Numerical experiments show that the traditional initial solution procedure generates tours with up to 20 transport tasks for the large problem instance. Thus, a lot of Tabu Search iteration steps are necessary to get to areas with feasible solutions. Like in the TSLN the empty truck movements are first sorted according to their duration in descending order. The number of links that are used to construct neighbor solutions is fixed with the parameter E. This parameter is similar to the divider D in the TSLN and TSAS. D defines a fraction of all empty truck movements, whereas E determines a certain number of them. E was introduced to enable a more flexible parameterization. In every fifth iteration step the TSLP uses E links with the shortest duration to be removed in neighbor solutions; in all other iteration steps the links with the longest duration are chosen, like in the TSLN. This strategy leads to a diversification of the search process.

A 2-opt based heuristic is applied to the modified tours as a local post-optimization procedure after each iteration step of the Tabu Search algorithm. The algorithm attempts to improve the single tours by changing the position of two transport tasks. If improvement is attained, the tour is rebuilt accordingly and the same procedure restarts until no further improvement can be found. Per definition, an improvement of a solution is only tolerated if the solution is feasible. The post-optimization procedure does not influence the Tabu Search algorithm; the input data for the next Tabu Search iteration step remains unchanged even if improvement is attained. Only the current best found solution and its costs are changed in this case.

3 Results and Conclusions

The presented algorithms were implemented in Visual Basic and tested in extensive numerical studies. Two sets of test cases were developed. Each set consists of 20 instances with 30 transport tasks and 10 trucks, which is a typical real life problem size. The first set of instances has weaker constraints than the second one in terms of the average task duration and the traveling times between the tasks. These test cases were used to evaluate TS, TSLN, and TSAS. An additional test case with 250 transport tasks and 80 trucks was developed to test the TSLP. The following conditions were varied in the test runs: Tabu Search specific parameters; total number of iteration steps; use of the post-optimization heuristic; the values of A, D, and E. The several Tabu Search variants were compared to each other and to the lower bound obtained by commercial solver software after a running time of 24 hours. We can clearly state that the TSLN, even though it offers the shortest computation times, is not recommendable with respect to solution quality. The TSAS offers an excellent solution quality in much shorter computing times than TS. But it is advantageous to use a full neighborhood search in more iteration steps if the constraints are tight; therefore the TS seems to be a good method if feasible solutions are very hard to find. The TSLP is capable to solve large problem instances within reasonable computing times. It can be observed that the improvements in solution quality have not been significantly compared to the additional computing times after 10,000 iteration steps. We can conclude that the heuristic solution approaches quickly converge to solutions with good quality. This fast speed of convergence may be an indication for being locked in local optima; but if we look at the intermediate solutions, we can notice that this is not the case, and the diversification strategies of TS, TSAS, and TSLP are working well.

The post-optimization strategy improves the best found solution mainly within the first iteration steps. This can be explained with the fact that there are generally not more than 5 transport tasks per tour. The functioning of the Tabu Search heuristics inserts transport tasks already on the cheapest positions in new tours and improves the tours therefore permanently, even though this does not guarantee a 2-optimal tour. The application of the post-optimization heuristic is cheap with respect to the computing time. Therefore, it can be concluded that it is recommendable to apply the post-optimization in all computations.

A comparison of the heuristic solution values and the lower bounds found by standard solver software after a computing time of 24h proved that the heuristics are working well.

Based on these numerical studies it can be concluded that the presented heuristic solution approaches are capable to solve real-life problem instances of the TTVRP in reasonable computing times with good solution quality. The TSLN can be seen as precursor of the TSAS and TSLP, which represent more sophisticated algorithms. It makes sense to put forth additional effort toward the enhancement of the TSAS and TSLP. Both algorithms could also bring forth benefits for other research areas that use a Tabu Search as solution method.

References

- Coja-Oghlan, A., Krumke, S. O., Nierhoff, T. (2004): A Heuristic for the Stacker Crane Problem on Trees which is Almost Surely Exact. Journal of Algorithms, In Press.
- Cordeau, J. F., Laporte, G., Mercier, A. (2001): A unified tabu search heuristic for vehicle routing problems with time windows. Journal of the Operational Research Society **52**, 928-936.
- Gronalt, M., Hartl, R., Reimann, M. (2003): New savings based algorithms for time constrained pickup and delivery of full truckloads. European Journal of Operational Research **151**, No. 3, 520-535.
- Gronalt, M. and Hirsch, P. (2006): Timber Transport Vehicle Routing Problems: Formulation and Heuristic Solution. Haasis, H. D., Kopfer, H., Schönberger, J. (eds.), Operations Research Proceedings 2005, Springer Berlin Heidelberg New York.
- Karanta, I., Jokinen, O., Mikkola, T., Savola, J., Bounsaythip, C. (2000): Requirements for a Vehicle Routing and Scheduling System in Timber Transport. Proceedings of the IUFRO International Conference: Logistics in the forest sector, 235-250.
- Palmgren, M., Rönnqvist, M., Värbrand, P. (2004): A near-exact method for solving the log-truck scheduling problem. International Transactions in Operational Research **11**, 447-464.
- Toth, P. and Vigo, D. (eds.) (2002): The Vehicle Routing Problem. SIAM, Philadelphia, USA.
- Toth, P. and Vigo, D. (2003): The Granular Tabu Search and Its Application to the Vehicle-Routing Problem. INFORMS Journal on Computing **15**, No. 4, 333-346.