Computers & Operations Research 36 (2009) 1683 – 1692

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/cor

Solving the truck and trailer routing problem based on a simulated annealing heuristic

Shih-Wei Lin[a], Vincent F. Yu[b,*], Shuo-Yan Chou[b]

[a]Department of Information Management, Chang Gung University, No. 259, Wen-Hwa 1st Road, Kwei-Shan, Tao-Yuan 333, Taiwan
[b]Department of Industrial Management, National Taiwan University of Science and Technology, No. 43, Section 4, Keelung Road, Taipei 106, Taiwan

ARTICLE INFO

Available online 29 May 2008

Keywords:
Vehicle routing problem
Truck and trailer routing problem
Simulated annealing
Heuristic

ABSTRACT

In this study, we consider the application of a simulated annealing (SA) heuristic to the truck and trailer routing problem (TTRP), a variant of the vehicle routing problem (VRP). In the TTRP, some customers can be serviced by either a complete vehicle (that is, a truck pulling a trailer) or a single truck, while others can only be serviced by a single truck for various reasons. SA has seen widespread applications to various combinatorial optimization problems, including the VRP. However, to our best knowledge, it has not been applied to the TTRP. So far, all the best known results for benchmark TTRP instances were obtained using tabu search (TS). We applied SA to the TTRP and obtained 17 best solutions to the 21 benchmark TTRP benchmark problems, including 11 new best solutions. Moreover, the computational time required by the proposed SA heuristic is less than those reported in prior studies. The results suggest that SA is competitive with TS on solving the TTRP.

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

In this paper we consider the truck and trailer routing problem (TTRP), a variant of the vehicle routing problem (VRP). VRP has long been one of the most studied combinatorial optimization problems due to its complexity in nature and extensive applications in practice [1–9]. In the standard VRP, a set of customers with known demand is serviced by a fleet of homogeneous vehicles with known capacity. The goal is to design least-cost vehicle routes originating from and terminating at a central depot to fulfill individual customer demands without violating the capacity constraints of the vehicles in a way that each customer is serviced only once by exactly one vehicle. In practical applications, additional operational requirements and restrictions, such as in the case of TTRP, may be imposed on the VRP.

In the TTRP, the use of trailers (a commonly neglected feature in the VRP) is considered where customers are serviced by a truck pulling a trailer. However, due to practical constraints, including government regulations, limited maneuvering space at customer site, road conditions, etc., some customers may only be serviced by a truck. These constraints exist in many practical situations.

Gerdessen gave two real-world TTRP applications in [10]. The first one occurred in the distribution of dairy products by the Dutch dairy industry. In this case, many customers were located in cities with heavy traffic and limited parking spaces. Maneuvering a complete vehicle (that is, a truck pulling a trailer) was very difficult. Therefore, the trailer was often parked at some point while the truck delivered the products to the customers along a certain route. Another case arose in the delivery of compound animal feed to farmers. Because there were narrow roads and/or small bridges on the delivery routes, various types of vehicles were needed to make the deliveries. One type of vehicles, called double bottoms, consisting of a truck and a trailer, was commonly used. The trailer might be parked at a parking place while the truck serviced some farmers on the delivery route.

Another application related to the TTRP was given by Semet and Taillard [11]. It occurred in a major food chain store in Switzerland where 45 company owned chain stores were serviced by a fleet of 21 trucks and 7 trailers. The scheduling for the deliveries with a combination of trucks and trailers was therefore of great interest.

Hoff [12] considered another real-world problem occurred at a Norwegian dairy company which collects raw milk from farmers. The company used a fleet of heterogeneous trucks with tanks for the milk. A truck could either drive the route by itself or carry a trailer with an additional tank. Since most Norwegian farms were small and inaccessible for vehicles with trailers, these vehicles needed to leave its trailer at a parking place before driving a sub-tour to the farmers to collect milk. When the truck returned to the parking place, it could fill the milk over from the truck tank to the trailer tank and go on another sub-tour from there. It could also drive the trailer to a new parking place, fill the milk over and start a new sub-tour from the new parking place. The milk could be stored up to three days at the farms so the problem could be treated as a special type of the TTRP.

* Corresponding author. Tel.: +886 2 2730 3276; fax: +886 2 2737 6344.
E-mail address: vincent@mail.ntust.edu.tw (V.F. Yu).

0305-0548/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2008.04.005

The TTRP is computationally more difficult to solve compared with the VRP because the VRP can be regarded as a special case of the TTRP. Since the VRP itself is a very difficult combinatorial optimization problem and is usually tackled by heuristic approaches [13,14,8,15,9,16,17], it is only natural to develop heuristic approaches for the TTRP, such as in Chao [18] and Scheuerer [19]. Both of these studies solved the TTRP by first constructing an initial solution with heuristics, and then improving the initial solution with a tabu search (TS) algorithm. The purpose of this study is to demonstrate that a carefully designed simulated annealing (SA) heuristic is competitive with other meta-heuristics in solving the TTRP. We developed an SA heuristic for the TTRP and compared the results with those obtained from TS based heuristics in the literature. The computational results indicate that our SA heuristic performs as good as prior approaches on most benchmark problems. Moreover, it finds new best solutions to 11 of the 21 benchmark TTRP instances.

The remainder of this paper is organized as follows. Section 2 gives a detailed description of the TTRP and surveys related problems. Section 3 describes the main features of the proposed SA heuristic. Section 4 details the parameter values of the tested algorithm configurations and presents the results of a comprehensive computational study. Finally, in Section 5, conclusions are drawn and future research directions are given.

2. Problem definition and literature review

The VRP and some of its variants have attracted substantive attention from academia and industry alike during the past four decades. However, due to their intrinsic complexity, the TTRP and some closely related problems have seldom been studied. The TTRP model discussed in this article was first proposed by Chao in 2002 [18]. The TTRP can be formally defined on an undirected graph $G=(V,A)$, where $V=\{0,1,2,\ldots,n\}$ is the set of vertices and $A=\{(i,j):i,j\in V\}$ is the set of edges. Vertex 0 represents the central depot, while the remaining vertices in $V\backslash\{0\}$ correspond to customers. Each vertex i is associated with a non-negative demand d_i, and a customer type t_i, where $t_i=1$ indicates that customer$i$ is a truck customer (TC) that can be serviced by a truck only while $t_i=0$ means customer$i$ is a vehicle customer (VC) who can be serviced by either a single truck or a complete vehicle. Each edge (i,j) is associated with a non-negative cost, c_{ij}, which can be interpreted as the travel time required on the edge or simply the travel distance of the edge.

A fleet of m_k available trucks and m_r available trailers are given. However, the number of trucks and the number of trailers used in the routes are not determined *a priori*. It is possible that some trucks or trailers are not used in a TTRP solution. Without loss of generality, we assume $m_k \geqslant m_r$, as in Chao [18] and Scheuerer [19]. All trucks have identical capacity Q_k, and all trailers have identical capacity Q_r. If a trailer is assigned to a truck, it has to stay with the truck while the truck is on the main tour. The goal of the TTRP is to find a set of least cost vehicle routes that start and end at the central depot such that each customer is serviced exactly once and the total demand of any vehicle route does not exceed the total capacity of the vehicles used in that route.

There are three types of routes in a TTRP solution as illustrated in Fig. 1: (1) a pure truck route (PTR) traveled by a single truck; (2) a pure vehicle route (PVR) without any sub-tour traveled by a complete vehicle; and (3) a complete vehicle route (CVR) consisting of a main tour traveled by a complete vehicle, and at least one sub-tour traveled by the truck alone. A sub-tour begins and ends at the same vehicle customer site or the depot on the main tour. In other words, the trailer is dropped off at a vehicle customer site or the depot, called the root of the sub-tour, while the truck proceeds to service customers on the sub-tour. After all customers on the sub-tour are serviced, the truck returns to the root of the sub-tour, hooks up its trailer and continues to service remaining customers on the same route.

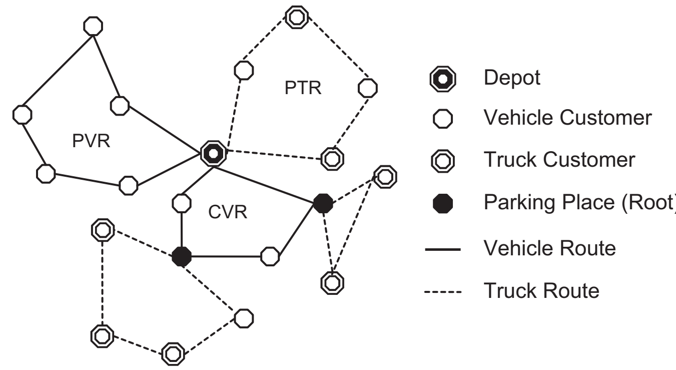

Fig. 1. An illustration of different type of vehicle routes.

Note that we follow Chao's definitions of vehicle route types which are slightly different from those of Scheuerer [19]. Scheuerer defined CVR to be a complete vehicle route with exactly one main tour and no, one, or more sub-tours, that is, the PVR and CVR used in Chao [18] and this paper. Despite the differences in terminology, Chao [18], Scheuerer [19] and this paper adopted the same TTRP model.

It's worth noting that Scheuerer [20] gave an 0–1 integer programming (0–1 IP) formulation of the TTRP which, to our best knowledge, is the first IP formulation for the TTRP, although earlier Semet [21] proposed a 0-1 IP model for a more restricted version of the TTRP. Drexl [22,23] gave a mixed integer programming model for a more general TTRP which included practical considerations such as variable costs for the trailers, fixed cost for the vehicles, parking places shared by different routes, and time windows. Drexl also developed a branch-and-price algorithm for the TTRP, which is the only exact approach for the TTRP in the literature. However, Drexl's algorithm had only been tested on relatively small instances of the TTRP that were randomly generated and structured to resemble real-world situations.

Both Chao [18] and Scheuerer [19] solved the TTRP by a 2-phase approach. In the first phase, construction heuristics were used to obtain an initial TTRP solution. The initial solution was then improved by a tabu search algorithm in the second phase.

Chao's construction heuristic was based on Fisher and Jaikumar's generalized assignment approach for vehicle routing [24]. The solution obtained in the construction phase was then improved by a TS improvement phase. It is worth noting that several features in Chao's heuristic may be considered for further improvement. First, the root of a sub-tour can only be changed during a descent phase of the tabu search, without worsening the objective function value; second, the number of sub-tours in each CVR is determined and fixed in the construction heuristic [19]. For a detailed description of the heuristic, see the original article by Chao [18].

Scheuerer [19] adopted Chao's TTRP model and developed two construction heuristics, T-Sweep and T-Cluster, along with a new tabu search improvement algorithm for the TTRP, and obtained better solution for each of the 21 benchmark TTRP instances. The T-Cluster heuristic is a cluster-based sequential insertion procedure in which routes are constructed one customer at a time up to full vehicle capacity. In the T-Sweep heuristic, feasible routes are constructed by rotating a ray centered at the depot and customers are gradually added to the current route, similar to the classical sweep algorithm by Gillett and Miller [2]. Both T-Cluster and T-Sweep are multi-start procedures. Scheuerer reported that T-Cluster heuristic outperforms T-Sweep and Chao's construction heuristic in terms of solution quality.

In Scheuerer's TS improvement procedure [19], intermediate infeasible solutions are allowed during the search and a shifting penalty approach by Gendreau et al. [25] is used to control infeasibility. Three types of neighborhoods are used in the TS: (1) Shift of consecutive nodes to existing tours or new sub-tours; (2) Swap of two subsets of nodes between two existing tours; (3) Sub-tour root refining.

The purpose of the sub-tour root refining is to find a better root node for CVR sub-tours. This procedure is applied to both the construction phase and the TS improvement phase. In the construction phase, whenever a VC is inserted into the main-tour, the procedure is applied to every sub-tour of the CVR to find the best possible root for all of its sub-tours. However, if the newly inserted customer is a TC, the procedure is only applied to the affected sub-tour. The sub-tour root refining procedure proceeds as follows: first, a sub-tour loop is formed by removing the two edges linking the sub-tour to the main-tour, and then connecting the resulting first and last customers on the sub-tour. Then the depot and every VC on the main-tour except for the former root are evaluated as the new root of the sub-tour. Cheapest insertion is used to determine the insertion position of the new root. The former root is replaced with the best alternative root found by the cheapest insertion provided that the total travel distance is reduced. The resulting sub-tour is then optimized by a 2-Opt and Or-Opt improvement heuristic. In the TS improvement phase, the same procedure is used to evaluate a reconnection of an existing sub-tour to all other root candidates on its main tour. However, new roots that deteriorate objective value may also be accepted in this phase.

To speed up the heuristic, the author restricted the evaluation process to be performed on the nearest h ($h \leqslant n$) nodes for all shift or swap moves. The author also reported sensitivity analyses for the parameter settings of his tabu search algorithm. For a more detailed description of the TS heuristic, readers are referred to the original article by Scheuerer [19].

There are few TTRP related problems in the literature. Semet and Taillard [11] discussed a practical VRP that allows limited use of trailers under accessibility restrictions. The problem differs from the TTRP in that a VC (called "trailer-store" in [11]) cannot be serviced in a sub-tour. Furthermore, other constraints such as time windows and vehicle-dependent variable costs are included in the problem. The authors developed a clustering-based construction heuristic and a TS heuristic for the problem.

Semet [21] considered a problem called the "partial accessibility constrained VRP" which is very similar to the TTRP. The problem is different from the TTRP as Semet made the following assumptions: two sub-tours cannot have the same root, all trucks must be used, number of trailers used must be determined *a priori*, and, as usual, the central depot cannot be visited in the middle of a vehicle route. Semet proposed a heuristic for the partial accessibility constrained VRP which was based on Fisher and Jaikumar's [24] generalized assignment method for the VRP. The trailer assignment problem was solved by a branch-and-bound algorithm with Lagrangian relaxation.

Gerdessen [10] studied a problem that is closely related to the TTRP, called the Vehicle Routing Problem with Trailers (VRPT). The model is different from the TTRP as all customers have unit demand, customers are assigned maneuvering costs instead of customer types, trailers can be parked at any customer site and each trailer is parked exactly once, vehicle speeds are different with and without trailers. The author proposed four construction heuristics for the problem. The author discussed two real-world applications, which we described in the previous section: the distribution of dairy products by the Dutch dairy industry and the delivery of compound animal feed to farmers.

Bodin and Levy [26] also studied a postal delivery problem which is very similar to the TTRP. In their problem, the postmen correspond to the trucks and their postal cars correspond to the trailers.

The site-dependent VRP (SDVRP) is also related to the TTRP. In the SDVRP, the fleet has many types of vehicles and there are vehicle-site compatibilities between customer sites and vehicle types (See [27–30]).

More recently, Scheuerer [20] gave two extensions of the TTRP, namely the multiple depot and the periodic TTRP. Drexl [31,23] studied a more general TTRP called the "vehicle routing problem with trailer and transshipments" in which the assignment of trailers to trucks are not fixed as in the TTRP. Drexl [22,23] also studied a TTRP related problem that includes more practical considerations.

3. Simulated annealing heuristic for the TTRP

Simulated annealing is a local search-based heuristic that is capable of escaping from being trapped into a local optimum by accepting, in small probability, worse solutions during its iterations. It has been applied successfully to a wide variety of highly complicated combinatorial optimization problems [32,7,33–37] as well as various real-world problems [38–40]. SA was introduced by Metropolis et al. in 1953 [41] and popularized by Kirkpatrick et al. [42]. The concept of the method is adopted from the "annealing" process used in the metallurgical industry. Annealing is the process by which slow cooling is applied to metals to produce better aligned, low energy-state crystallization. The optimization procedure of SA reaches a (near) global minimum mimicking the slow cooling procedure in the physical annealing process. It starts from a random initial solution. In each iteration, the algorithm takes a new solution from the predefined neighborhood of the current solution. The objective function value of this new solution is then compared with that of the current solution value in order to determine if an improvement has been attained. If indeed the objective function value of the new solution is better, that is, being smaller in the case of minimization, the new solution becomes the current solution from which the search continues by proceeding with a new iteration. A new solution with a larger objective function value may also be accepted as the current solution with a small probability determined by the Boltzmann function, $\exp(-\Delta/kT)$, where k is a predetermined constant and T is the current temperature. The essential idea is not to restrict the search moves only to those solutions that decrease the objective function value, but also allow moves that increase the objective function value. This mechanism may avoid the procedure being trapped prematurely in a local minimum.

In the following subsections, we discuss the proposed SA heuristic in detail, including the solution representation, the generation of the initial solution, the calculation of the objective function value, various types of neighborhood, the parameters used, and the SA procedure.

3.1. Solution representation and initial solution

Customers are classified as vehicle-serviced customers (VCs) and truck-serviced customers (TCs). A solution is represented by a string of numbers consisting of a permutation of n customers denoted by the set $\{1, 2, \ldots, n\}$ and N_{dummy} zeros (artificial depot or the root of a sub-tour), followed by the service vehicle types of individual VCs. The N_{dummy} zeros are used to separate routes or terminate a sub-tour. The parameter N_{dummy} is calculated by $\lfloor \sum_i d_i/Q_k \rfloor$, where $\lfloor \bullet \rfloor$ denotes the largest integer which is smaller than or equal to the enclosed number. The ith non-zero number in the first $n + N_{\text{dummy}}$ positions denotes the ith customer to be serviced.

The service vehicle type of a VC is either 0 or 1. If the VC is serviced by a complete vehicle, its service vehicle type is set to be 0. Otherwise, it is serviced by a truck alone, and its service vehicle type is set to be 1. Note that a TC must be serviced by a truck alone and thus does not need to be represented in the solution. The service

Table 1
A small TTRP example problem.

Customer i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Customer type	TC	VC	TC	VC	VC	VC	VC	VC	TC	TC	TC	VC	VC	VC	VC
Demand d_i	10	10	15	40	20	30	10	15	10	10	10	15	30	20	20

Truck capacity $Q_k = 100$; Trailer capacity $Q_r = 50$.

vehicle type of a VC determines the type of the vehicle used to service the VC so that each solution representation corresponds to exactly one TTRP solution.

The solution representation is further explained as follows. The first number in the solution indicates the first customer to be serviced in the first route. If the first customer on a route is to be serviced by a single truck, the route is set to be a PTR. Other customers are added to the route one by one from left to right to represent the sequence in which they are serviced, provided that the capacity of the vehicle in use is not violated. Note that, depending on the type of the service vehicle in use, the capacity of the vehicle may be (Q_k+Q_t) if it is a complete vehicle on a PVR or on the main tour of a CVR; or Q_t if it is a truck servicing customers alone, on a PTR or on a sub-tour of a CVR. If the next customer to be serviced in the solution representation is zero, the vehicle will either return to the root of a sub-tour or the depot. If it is on a sub-tour of a CVR, it will return to the root of the sub-tour where the trailer was parked and the sub-tour is terminated. Otherwise, it is on a PTR, on a PVR, or on a main tour of a CVR. In this case, the vehicle will return to the depot and the route is terminated.

Whenever a route is terminated and there are customers that haven't been serviced, a new route will be generated starting with the next customer in the solution representation. It can be verified that this solution representation always gives a TTRP solution without violating the capacity constraint of vehicles in use. However, the number of vehicle used may exceed the number of vehicles available using this solution representation. Therefore, a route combination procedure that tries to reduce the number of vehicles used is needed after the routes are generated from the solution representation. The route combination procedure simply checks if it is possible to combine two existing routes without violating vehicle's capacity constraint. If so, the routes are merged without any modification. The procedure continues until the number of vehicle used is no more than the number of available vehicle or there are no routes can be combined without violating the capacity constraint of the vehicle in use.

If the resulting solution still uses more vehicles than available, for each extra truck or trailer used, a penalty cost P is added to the objective function to make such solutions unattractive.

The initial solution is randomly generated. It is comprised of a randomly ordered sequence of the customers and the dummy zeros, and randomly set service vehicle types of VCs.

3.2. Illustration of solution representation

Table 1 gives a small TTRP example problem with 15 customers. A sample solution for this example is shown in Fig. 2, where two artificial depots are introduced. Customers 2, 4, 5, 6, 7, 8, 12, 13, 14 and 15 are vehicle customers (shown in boldface) and their service vehicle types are as shown in the solution representation. Shaded customers are to be serviced by a single truck. Among them, customer 2 and customer 8 are VCs that will be serviced by trucks since their service vehicle type is 1 in the solution representation. The remaining five customers are truck customers and thus must be serviced by trucks. A pictorial correspondence of this solution representation is given in Fig. 3, with which we now illustrate the routes one by one.

In this example, VC 4 will be serviced in the first route. Recall that the service vehicle type of every VC is predetermined in the solution representation. In this case, the service vehicle type of customer 4 equals 0, indicating that it is serviced by a complete vehicle. Since the next customer 11 is a truck customer, the trailer has to be parked at customer 4, and then the truck will go on to service customers 11, 8 (customer 8 is a VC whose service vehicle type is 1, thus it is serviced by the truck) and 1 sequentially. After finishing servicing customer 1, the route is supposed to continue on to customer 15. Since customer 15 is a VC whose service vehicle type in the solution representation is 0, indicating that it is serviced by a complete vehicle, the truck needs to return to customer 4 to pick up its trailer before continuing onto the route to service customer 15 and then customer 13. After finishing servicing customer 13, the next customer to be serviced in the solution representation is 0. Since the vehicle is currently on a main tour of a CVR, the 0 indicates that the next customer is the depot. Therefore, the vehicle will return to the depot, and the first route is terminated. The first route is a complete vehicle route, because it contains a main tour traveled by a complete vehicle and a sub-tour traveled by a truck alone.

The first customer to be serviced in the second route is customer 9, followed by customers 10, 2, and 3. Recall that we set a route to be a PTR whenever the first customer in the route is serviced by a single truck. Because customer 9 is a truck customer, this route is a pure truck route. Since customer 7 is a VC that is serviced by a complete vehicle and the current route is a PTR, after finishing servicing customer 3, this truck will return to the depot, and the route is terminated. Note that in this route, although customer 2 is a VC, its corresponding service vehicle type is 1, thus it is serviced by a truck. Also note that if VC 7's service vehicle type is 1 (serviced by a truck) and adding it to the route will not violate the truck's capacity, than a 0 must be placed right before VC 7 to represent the solution in Fig. 3. Keep in mind that when decoding a solution representation, the current route may be terminated when the entry in the next position is a 0 or when adding the next customer to the route violates the capacity constraint of the vehicle in use.

The third route starts by servicing customer 7, and then servicing other customers, in the sequence of 7, 14, 5, 6, and 12. A dummy zero appears immediately after customer 12 in the solution representation. Therefore the vehicle will go back to the depot after finishing servicing customer 12. The customers in the third route are all VCs. All of them are serviced by a complete vehicle because their service vehicle type is 0 in the solution representation. Thus, this route is a pure vehicle route.

At this point, if the number of vehicles used exceeds the number of available vehicles, the route combination procedure will be performed to reduce the number of vehicles used.

The solution representation has determined the customers on each route and the service vehicle type of each VC. Once this is done, it is easy to calculate the objective value (total distance traveled), obj(X, P), of a given solution X where P is a penalty cost associated with each extra vehicle used in the solution.

3.3. Neighborhood

We use a standard SA procedure with a random neighborhood structure that features various types of moves, including insertion, swap, and change of service vehicle type, to solve the TTRP. We define the set $N(X)$ to be the set of solutions neighboring a solution X. In each iteration, the next solution Y is generated from $N(X)$ either by insertion, swap, or change of service vehicle type of VCs as follows.

The insertion is carried out by randomly selecting the ith customer of X and inserting it into the position immediately preceding another randomly selected jth customer of X. The swap is performed by randomly selecting the ith and the jth customers of X, and then

4	11	**8**	1	**15**	**13**	0	9	10	**2**	3	**7**	**14**	**5**	**6**	**12**	0	1	0	0	0	0	1	0	0	0	0
Route 1							Route 2				Route 3						Service vehicle types of VCs for customer									
																	2	**4**	**5**	**6**	**7**	**8**	**12**	**13**	**14**	**15**

Fig. 2. A sample TTRP solution representation.

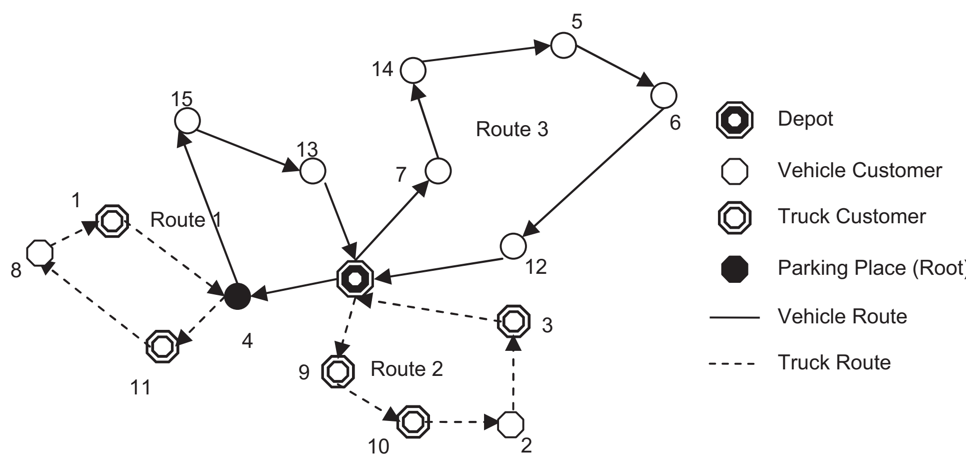

Fig. 3. An illustration of the TTRP solution corresponding to the solution presentation in Fig. 2.

swapping the positions of these two customers. The change of service vehicle type of VCs is performed by randomly selecting a VC from X, and then changing its service vehicle type from 1 to 0 or from 0 to 1, that is, if the chosen VC was serviced by a truck before the move, it will be serviced by a complete vehicle after the move, and vice versa. Take the solution representation in Fig. 2 for example. If customer 15 (a VC with service vehicle type 0, i.e. serviced by a complete vehicle) is randomly chosen to make the change of service vehicle type move, its service vehicle type will be changed from 0 to 1, indicating that it will be serviced by a truck after the move. Then after finishing servicing customer 1, the truck will go on to service customer 15 before returning to customer 4 to pick up its trailer, provided that adding customer 15 to the sub-tour will not violate the truck's capacity constraint.

The probabilities of choosing the swap, insertion, or the change of service vehicle type of VCs moves are set to be 0.2, 0.2, and 0.1, respectively.

In order to increase the chance of obtaining a better solution, besides randomly choosing one or two customers to undergo either one of the three operations, the best-of-N-trials moves are also performed, where N is a predetermined number of trails. The best solution among the N trial solutions is chosen as the next solution.

For swap and insertion, this number is set to be N_{trial}, calculated as $\lfloor (n + N_{\text{dummy}})/3 \rfloor$, where $\lfloor \bullet \rfloor$ denotes the largest integer which is smaller than or equal to the enclosed number. For the change of service vehicle type of VCs, each VC's service vehicle type is changed one at a time, thus the number of trials equals the number of VCs.

The probabilities of performing the best-of-N-trials moves are 0.2 and 0.2 for swap and insertion, respectively. With probability 0.1, the best-of-N-trials moves for the change of service vehicle type of VCs is performed.

The probabilities of performing swap, insertion, and change of service vehicle type of VCs on one or two randomly selected customers add up to 0.5, and the probability of performing the best-of-N-trials moves is also 0.5, making the total probability of neighborhood moves 1, as illustrated in the pseudo code in Fig. 4.

3.4. Parameters used

The SA begins with seven parameters I_{iter}, T_0, T_F, K, P, $N_{\text{non-improving}}$ and α. I_{iter} denotes the number of iterations the search proceeds at a particular temperature. While T_0 represents the initial temperature, T_F represents the final temperature below which the SA procedure is stopped. K is the Boltzmann constant used in the probability function to determine whether to accept a worse solution or not. P is the penalty cost associated with the number of extra vehicles used. $N_{\text{non-improving}}$ is the maximum allowable number of reductions in temperature during which the best objective function value is not improved. Finally, α is the coefficient controlling the cooling scheme.

3.5. The SA procedure

In the beginning, the current temperature T is set to be the same as T_0. Next, an initial solution X is randomly generated. The current best solution X_{best} and the best objective function value obtained so far are set to be X and $\text{obj}(X, P)$, respectively.

In each iteration, the next solution Y is generated from $N(X)$ and its objective function value is evaluated. Let Δ denote the difference between $\text{obj}(X, P)$ and $\text{obj}(Y, P)$, that is $\Delta = \text{obj}(Y, P) - \text{obj}(X, P)$. In each iteration, the probability of replacing X with Y, where X is the current solution and Y is the next solution, given that $\Delta > 0$, is $\exp(-\Delta/KT)$, is accomplished by generating a random number $r \in [0, 1]$ and replacing the solution X with Y if $r < \exp(-\Delta/KT)$. Meanwhile, if $\Delta \leqslant 0$, the probability of replacing

SA(T_0, T_F, α, K, P, $N_{non\text{-}improving}$, I_{iter})

Step 1: Let $N_{dummy} = \left\lfloor \sum_i \frac{d_i}{Q_k} \right\rfloor$. Let $N_{trial} = \left\lfloor \frac{(n + N_{dummy})}{3} \right\rfloor$.

Generate the initial solution X randomly.

Step 2: Let T=T_0; I=0; J=0; N=0; F_{best}=obj(X, P); X_{best}=X;

Step 3: I=I+1;

Step 4: (Generating a solution Y based on X)

Step 4.1: Generate r = random (0,1);

Step 4.2: Case $r \leq 0.2$

Generating a new solution Y from X by random swap operation;

Case $0.2 < r \leq 0.4$

Generating a new solution Y from X by choosing the best solution from N_{trial} random swap operations;

Case $0.4 < r \leq 0.6$

Generating a new solution Y from X by random insertion operation;

Case $0.6 < r \leq 0.8$

Generating a solution Y from X by choosing the best solution from N_{trial} random insertion operations;

Case $0.8 < r \leq 0.9$

Generating a solution Y from X by randomly selecting a VC in X and changing its service vehicle type from 0 to 1 or from 1 to 0;

Case $0.9 < r \leq 1.0$

Generating a solution Y from X by choosing the best solution from all possible service vehicle type change of VCs.

Step 5: If Δ=obj(Y, P) - obj(X, P) $\leq$0 {Let X=Y;}

Else {

Generate r = random (0,1);

If r<exp(-Δ/KT) {Let X=Y;}

}

Step 6: If (obj(X, P)<F_{best} and X is feasible){X_{best}=X; F_{best}=obj(X, P);N=0;}

Step 7: If I=I_{iter} {

T=αT; I=0;N=N+1;J=J+1;

If J=3 {

Perform 2-Opt local search on X_{best};

Perform Local search based on swap operation on X_{best};

Perform Local search based on insertion operation on X_{best};

Perform Local search based on change of service vehicle type operation on X_{best};

J=0;

}

}

Else {Go to Step 3;}

Step 8: If T<T_F or N=$N_{non\text{-}improving}$ {Terminate the SA heuristic;}

Else {Go to Step 3;}

Fig. 4. Pseudo-code of the proposed SA heuristic for the TTRP.

X with Y is 1. X_{best} records the best solution as the algorithm progresses.

The temperature T is decreased after running I_{iter} iterations since the previous decrease, according to the formula $T = \alpha T$, where $0 < \alpha < 1$. After every three temperature reductions, a local search procedure which sequentially performs 2-Opt, swap, insertion, and change of service vehicle types is used to improve the current best solution.

The algorithm is terminated if the current temperature T is lower than T_F or the current best solution X_{best} is not improved in $N_{\text{non-improving}}$ consecutive temperature reductions. Following the termination of SA procedure, the (near) optimal schedule can be derived from X_{best}. The proposed SA approach is summarized in Fig. 4.

4. Computational results

The SA heuristic was coded in C++ and compiled using Microsoft Visual C + + 6.0. It was then applied to the 21 TTRP benchmark instances created by Chao [18] on a Pentium IV 1.5 GHz PC with 1 GB RAM under Microsoft Windows XP operating system. To our best knowledge, this is the only set of TTRP benchmark problems in the OR literature.

These 21 TTRP benchmark problems are converted from seven basic VRP test problems given by Christofides, Mingozzi, and Toth (CMT) [43]. They are generated in the following manner. For each customer i in a CMT problem, the distance between i and its nearest neighbor customer is calculated and denoted by A_i. Each CMT problem was converted into three TTRP problems. In the first problem, 25% of the customers with the smallest A_i values are specified as truck customers. This percentage was increased to 50% and 75% in the second and third problem respectively. Table 2 shows the numbers and capacities of available trucks and trailers for all test problems and the ratio of total demand to total capacity reported by Chao [18].

Parameter selection may influence the quality of the computational results. In the initial experiments, the following combinations of parameters were tested.

$\alpha = 0.965, 0.975$;

$P = 25, 50, 75, 100, 125$;

$I_{\text{iter}} = 30000, 50000, 70000, 90000, 120000, 150000, 200000$;

$K = 1/1, 1/2, \ldots, 1/9$.

Setting $\alpha = 0.965$, $I_{\text{iter}} = 150000$, $P = 50$, and $K = 1/3$ seemed to give best results. Therefore they were used for further computational study. Other parameters used in the final analysis are: $T_0 = 100$, $T_F = 1$, and $N_{\text{non-improving}} = 30$. Since $T_0\alpha^{130} = 100 \times 0.965^{130} < 1 = T_F$, the current temperature will be below the final temperature after 130 temperature reductions. Thus, all the experiments were terminated after 130 iterations, or when X_{best} is not improved in 30 successive reductions in temperature. In order to show the convergence trend of the proposed approach, we took the 3rd TTRP benchmark problem and presented the relationship between the obtained objective function value and the number of temperature reduction in Fig. 5. As can be seen, the solution improvement rate decreases as the number of temperature reduction increases, and the obtained solution converges after a certain number of temperature reductions. Thus, a greater number of "reductions in temperature" may not enhance the solution quality.

In order to evaluate the performance of the SA heuristic, we compare the solutions obtained by the proposed SA heuristic with the solutions reported in Chao [18] and Scheuerer [19]. For each of the 21 test problems, results obtained by the TS method of Chao [18], the TS heuristic of Scheuerer [19], by the proposed SA heuristic, as well as the objective value $c(s^{**})$ of the best known solutions are presented in Table 3. The proposed SA heuristic obtained best solutions to 17 out of the 21 TTRP benchmark problems, 11 of them are new best solutions, while Scheuerer [19] remained the best known for 10 of the 21 problems (tied on 6 problems). Note that Scheuerer's best solutions were obtained while performing sensitivity analysis, so this is not a comparison on the performance of the two algorithms; instead we compare the average of the minimum objective values from 10 runs and the average of the average objective values from 10 runs of the proposed SA heuristic with those of Scheuerer's TS heuristic.

The average (Avg.) and the average relative percentage deviation (ARPD) are given in the last two rows of Table 3. It can be seen that the average of minimum objective values from 10 runs of the proposed SA heuristic is slightly better than that of Scheuerer's TS heuristic (958.06 and 962.40, respectively). The average of the average minimum objective values from 10 runs is also slightly lower than that of Scheuerer's TS heuristic (968.24 and 970.84, respectively). Moreover, on average, the SA heuristic took only about 5/6 of the time required by Scheuerer's TS (39.56 and 47.32, respectively) to obtain the best solutions.

Table 2
Dimensions of the TTRP test problems reported by Chao [18].

Problem ID	Original Problem	Number of Customers		Trucks		Trailers		Ratio of demand to capacity
		VC	TC	Number	Capacity	Number	Capacity	
1	CMT1	38	12	5	100	3	100	0.971
2		25	25					
3		13	37					
4	CMT2	57	18	9	100	5	100	0.974
5		38	37					
6		19	56					
7	CMT3	75	25	8	100	4	100	0.911
8		50	50					
9		25	75					
10	CMT4	113	37	12	100	6	100	0.931
11		150	75					
12		100	112					
13	CMT5	150	49	17	150	9	100	0.923
14		100	99					
15		50	149					
16	CMT11	90	30	7	150	4	100	0.948
17		60	60					
18		30	90					
19	CMT12	75	25	10	150	5	100	0.903
20		50	50					
21		25	75					

Table 3
A comparison of results obtained by the proposed SA heuristic and other approaches.

ID	Chao		Scheuerer ($\lambda = 15\,000$)			SA heuristic			$c(s^{**})$[k]
	Min $c(s^*)$[a]	T[b]	Min $c(s^*)$[c]	Avg $c(s^*)$[d]	T[e]	Min $c(s^*)$[f]	Avg $c(s^*)$[g]	T[h]	
1	565.02	4.19	566.80	567.98	9.51	566.82	568.86	6.80	564.68[i,j]
2	662.84	5.22	615.66	619.35	9.60	612.75	617.48	6.67	611.53[j]
3	664.73	6.50	620.78	629.59	11.24	618.04	620.50	5.59	618.04[i,j]
4	857.84	7.53	801.60	809.13	18.49	808.84	817.71	16.32	798.53[i,j]
5	949.98	7.06	839.62	858.98	15.16	839.62	858.95	14.42	839.62[i,j]
6	1084.82	7.96	936.01	949.89	18.62	934.11	942.60	13.65	930.64 [j]
7	837.80	16.43	830.48	832.91	33.60	830.48	838.50	24.96	830.48[i,j]
8	906.16	11.11	878.87	881.26	25.66	875.76	882.70	24.03	872.56[j]
9	1000.27	10.18	942.31	955.95	30.47	912.64	921.97	21.75	912.02[j]
10	1076.88	21.72	1039.23	1052.65	60.94	1053.90	1074.38	63.61	1039.07[i]
11	1170.17	17.10	1098.84	1107.47	56.17	1093.57	1108.88	60.33	1093.57[j]
12	1217.01	20.27	1175.23	1184.58	63.71	1155.44	1166.59	51.70	1154.73[j]
13	1364.50	42.34	1288.46	1296.33	165.41	1320.21	1340.98	119.56	1287.18[i]
14	1464.20	25.96	1371.42	1384.13	132.06	1351.54	1367.91	113.75	1347.40[j]
15	1544.21	24.62	1459.55	1488.71	154.10	1436.78	1454.91	93.87	1425.87[j]
16	1064.89	14.56	1002.49	1003.00	43.14	1004.47	1007.26	41.46	1002.49[i]
17	1104.67	13.74	1042.35	1042.79	33.73	1026.88	1035.23	38.81	1026.20[j]
18	1202.00	12.52	1129.16	1141.94	31.78	1099.09	1110.13	31.34	1098.15[j]
19	887.22	16.13	813.50	813.98	28.84	814.07	823.01	29.58	813.30[j]
20	963.06	10.09	848.93	852.89	24.57	855.14	859.06	28.47	848.93[i]
21	952.29	9.57	909.06	914.04	26.84	909.06	915.38	24.03	909.06[i,j]
Avg.	1025.74	14.51	962.40	970.84	47.32	958.06	968.24	39.56	953.53
ARPD	7.57%[l]		0.93%[l]	1.82%[m]		0.48%[l]	1.54%[m]		0.00

[a]Best solutions from 10 runs for Chao's best parameter set 5.

[b]Times in minutes on a Pentium II 350 MHz PC for Chao's best parameter set 5.

[c]Best solutions from 10 runs by Scheuerer ($\lambda = 15000$).

[d]Average solutions from 10 runs by Scheuerer ($\lambda = 15000$).

[e]Average times in minutes from 10 runs on a Pentium IV 1.5 GHz PC for Scheuerer ($\lambda = 15000$).

[f]Best solutions from 10 runs by the proposed SA heuristic.

[g]Average solutions from 10 runs by the proposed SA heuristic.

[h]Average times in minutes from 10 runs on a Pentium IV 1.5 GHz PC for the proposed SA heuristic.

[i]Best solutions obtained by Scheuerer.

[j]Best solutions obtained by the proposed SA heuristic.

[k]Best known solutions.

[l]Calculated by (Avg. Min $c(s^*)$ – Avg. $c(s^{**})$)/Avg. $c(s^{**})$.

[m]Calculated by (Avg. Avg $c(s^*)$ – Avg. $c(s^{**})$)/Avg. $c(s^{**})$.

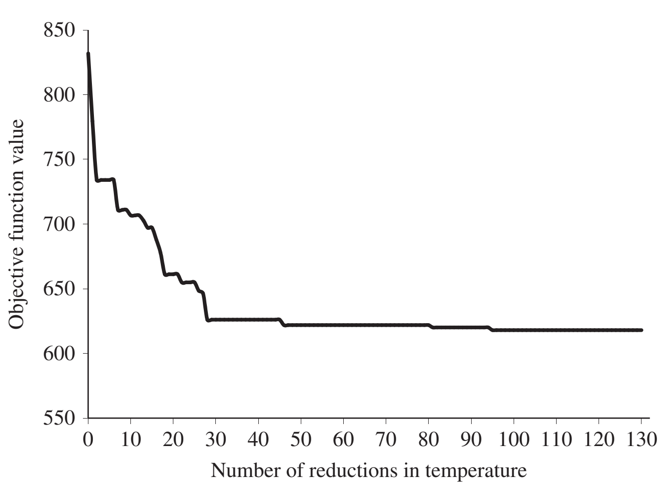

Fig. 5. Evolution of best solution obtained in the 3rd benchmark TTRP problem.

We further made an average relative gap comparison between the performance of Scheuerer's TS and the proposed SA. We computed the solution qualifications (in terms of solution quality and computing time) of the proposed SA heuristic relative to those of Scheuerer's TS approach [19]. The average relative gaps between the minimum objective values obtained in 10 runs, between the average objective values obtained in 10 runs, and between the average computing times are shown as percentages in Table 4. It can be seen that the performance of the proposed SA is slightly better than that of the TS.

Both of our SA heuristic and Scheuerer's TS heuristic seem to outperform Chao's TS method in terms of solution quality. However, we would like to point out that Chao's algorithm takes much less time than ours and Scheuerer's heuristic. Note that it remains unclear whether the computing times reported by Chao refer to the total time for all 10 runs or to the time for a representative single run [19]. Therefore, more experiments may be needed before reaching a conclusion on the performance of Chao's algorithm.

To summarize, the analyses indicate that to obtain quality solutions to the TTRP, the proposed SA heuristic is as effective as and seems to be slightly more efficient than Scheuerer's TS heuristic. However, we note that Scheuerer's TS heuristic was embedded into a solution-framework that allows for additional side constraints, such as additional cost components, multiple depots, and longer planning horizon. A stand-alone implementation of Scheuerer's algorithm is expected to reduce its runtimes significantly [19].

Table 4
Solution Qualifications of the Proposed SA Relative to the TS of Scheuerer.

Prob. ID	Comparison of minimum objective values of 10 runs (SA-TS)/TS[a]	Comparison of average objective values of 10 runs (SA-TS)/TS[b]	Comparison of average computing times (SA-TS)/TS[c]
1	0.00%	0.15%	−28.52%
2	−0.47%	−0.30%	−30.55%
3	−0.44%	−1.44%	−50.31%
4	0.90%	1.06%	−11.76%
5	0.00%	0.00%	−4.91%
6	−0.20%	−0.77%	−26.68%
7	0.00%	0.67%	−25.71%
8	−0.35%	0.16%	−6.35%
9	−3.15%	−3.55%	−28.62%
10	1.41%	2.06%	4.38%
11	−0.48%	0.13%	7.41%
12	−1.68%	−1.52%	−18.85%
13	2.46%	3.44%	−27.72%
14	−1.45%	−1.17%	−13.86%
15	−1.56%	−2.27%	−39.09%
16	0.20%	0.42%	−3.90%
17	−1.48%	−0.72%	15.07%
18	−2.66%	−2.79%	−1.39%
19	0.07%	1.11%	2.56%
20	0.73%	0.72%	15.89%
21	0.00%	0.15%	−10.49%
Average Relative Gap	−0.39%	−0.21%	−13.50%

[a]Relative gap between the minimum (best) objective values obtained in 10 runs of the algorithms (The lower, the better for the SA.).

[b]Relative gap between the average objective values obtained in 10 runs of the algorithms. (The lower, the better for the SA.).

[c]Relative gap between the CPU times (The lower, the faster for the SA.).

5. Conclusions and outlook for future research

In this paper, we proposed an SA heuristic for the TTRP and compared it with existing approaches using benchmark instances from the literature. The proposed SA heuristic found 17 best solutions to 21 TTRP benchmark problems, including 11 new best solutions and 6 previously reported best solutions [19]. Its main characteristics are the combination of a two-level solution representation with the use of dummy depots/roots, and the random neighborhood structure which utilizes three different types of moves. The computational study shows that our algorithm is competitive with other known solution approaches for the TTRP. Moreover, the algorithm is very efficient as it takes less time to obtain the best or near-best solutions. It would be interesting to test the effectiveness and efficiency of the proposed SA heuristic on various extensions of the TTRP, such as TTRP with time window constraints and TTRP with maximum tour duration constraints.

Acknowledgement

This work was partially funded by the New Faculty Start-up Grants, National Taiwan University of Science and Technology. This support is gratefully acknowledged. The authors are indebted to two anonymous referees for their thorough reviews and the insights they provided. Thanks are also due to Dr. Katta G. Murty and Dr. Yat-wah Wan for their valuable comments and suggestions.

References

[1] Dantzig GB, Ramser JH. The truck dispatching problem. Management Science 1959;6:80–91.

[2] Gillett B, Miller L. A heuristic algorithm for the vehicle dispatch problem. Operations Research 1974;22(2):340–9.

[3] Bodin L, Golden BL, Assad A, Ball MO. Routing and scheduling of vehicles and crews: the state of the art. Computers and Operations Research 1983;10: 62–212.

[4] Laporte G, Nobert Y. Exact algorithms for the vehicle routing problem. Surveys in Combinatorial Optimization 1987;31:147–84.

[5] In: Golden in:BL, Assad A, editors. Vehicle Routing: Methods and Studies. North-Holland: Amsterdam; 1988.

[6] Laporte G. The vehicle routing problem: an overview of exact and approximate algorithms. European Journal of Operational Research 1992;59:345–58.

[7] Van Breedam A. Improvement heuristics for the vehicle routing problem based on simulated annealing. European Journal of Operational Research 1995;86: 480–90.

[8] Laporte G, Gendreau M, Potvin JY, Semet F. Classical and modern heuristics for the vehicle routing problem. International Transactions in Operational Research 2000;7(4–5):285–300.

[9] Toth P, Vigo D. The Vehicle Routing Problem. Philadelphia, PA: SIAM Monographs on Discrete Mathematics and Applications; 2002.

[10] Gerdessen JC. Vehicle routing problem with trailers. European Journal of Operational Research 1996;93(1):135–47.

[11] Semet F, Taillard E. Solving real-life vehicle routing problems efficiently using tabu search. Annals of Operations Research 1993;41:469–88.

[12] Hoff A. Heuristics for Rich Vehicle Routing Problems. Ph.D. Thesis. Molde University College, 2006.

[13] Filipec M, Skrlec D, Krajcar S. An efficient implementation of genetic algorithms for constrained vehicle routing problem. IEEE International Conference on Systems, Man, and Cybernetics, October 11–14, 1998, San Diego, CA, USA: 2231-2236.

[14] Chin AJ, Kit HW, Lim A. A new GA approach for the vehicle routing problem. 11th IEEE International Conference on Tools with Artificial Intelligence, November 8–10, 1999, Chicago, Illinois, USA:307-311.

[15] Cordeau JF, Gendreau M, Laporte G, Potvin JY, Semet F. A guide to vehicle routing heuristics. Journal of the Operational Research Society 2002;53(5): 512–22.

[16] Baker BM, Ayechew MA. A genetic algorithm for the vehicle routing problem. Computers and Operations Research 2003;30:787–800.

[17] Mazzeo S, Loiseau I. An ant colony algorithm for the capacitated vehicle routing. Electronic Notes in Discrete Mathematics 2004;18:181–6.

[18] Chao IM. A tabu search method for the truck and trailer routing problem. Computers and Operations Research 2002;29(1):33–51.

[19] Scheuerer S. A tabu search heuristic for the truck and trailer routing problem. Computers and Operations Research 2006;33:894–909.

[20] Scheuerer S. Neue Tabusuche-Heuristiken für die logistische Tourenplanung bei restringierendem Anhängereinsatz, mehreren Depots und Planungsperioden. Ph.D. Thesis. School of Business, University of Regensburg, Germany; 2004 [in German]. ⟨http://www.opus-bayern.de/uni-regensburg/volltexte/2004/376/⟩.

[21] Semet F. A two-phase algorithm for the partial accessibility constrained vehicle routing problem. Annals of Operations Research 1995;61:45–65.

[22] Drexl M. A branch-and-price algorithm for the truck-and-trailer routing problem. Technical report. RWTH Aachen University, Germany; 2007. ⟨http://www.dpor.rwth-aachen.de/de/publikationen/pdf/or_2006-06.pdf⟩.

[23] Drexl M. On Some Generalized Routing Problems. Ph.D. Thesis. RWTH Aachen University, Germany; 2007. ⟨http://darwin.bth.rwth-aachen.de/opus3/volltexte/2007/2091/pdf/Drexl_Michael.pdf⟩.

[24] Fisher ML, Jaikumar R. Generalized assignment heuristic for vehicle routing. Networks 1981;11(2):109–24.

[25] Gendreau M, Hertz A, Laporte G. Tabu search heuristic for the vehicle routing problem. Management Science 1994;40(10):1276–90.
[26] Bodin L, Levy L. Scheduling of local delivery carrier routes for the United States postal service. In: Dror M, editor. Arc Routing: Theory, Solutions, and Applications. Boston: Kluwer; 2000. p. 419–42.
[27] Nag B, Golden BL, Assad A. Vehicle routing with site dependencies. In: Golden BL, Assad A, editors. Vehicle Routing: Methods and Studies. Studies in Management Science and Systems 16. The Netherlands: Amsterdam; 1988. p. 149–59.
[28] Rochat Y, Semet F. A tabu search approach for delivering pet food and flour in Switzerland. Journal of Operational Research Society 1994;45(11):1233–46.
[29] Chao IM, Golden BL, Wasil EA. A new algorithm for the site-dependent vehicle routing problem. In: Woodruff DL, editor. Advances in Computational and Stochastic Optimization, Logic Programming, and Heuristic Search. Norwell, MA: Kluwer Academic Publishers; 1998. p. 301–12.
[30] Chao IM, Golden BL, Wasil EA. A computational study of a new heuristic for the site-dependent vehicle routing problem. Information Systems and Operational Research 1999;37(3):319–36.
[31] Drexl M. The Vehicle Routing Problem with Trailers and Transshipments. Odysseus 2006, Third International Workshop on Freight Transportation and Logistics, May 23–26, 2006, Altea, Spain.
[32] Abramson D. Constructing school timetables using simulated annealing: sequential and parallel algorithms. Management Science 1991;37:98–113.
[33] Chwif L, Barretto MRP, Moscato LA. A solution to the facility layout problem using simulated annealing. Computers in Industry 1998;36:125–32.
[34] Jayaraman V, Ross A. A simulated annealing methodology to distribution network design and management. European Journal of Operational Research 2003;144:629–45.
[35] Lim A, Rodrigues B, Zhang X. A simulated annealing and hill-climbing algorithm for the traveling tournament problem. European Journal of Operational Research 2006;174:1459–78.
[36] McKendall Jr. AR, Shan J, Kuppusamy S. Simulated annealing heuristics for the dynamic facility layout problem. Computers and Operations Research 2006;33:2431–44.
[37] Lin SW, Chou SY, Chen SC. Meta-heuristic approaches for minimizing total earliness and tardiness penalties of single-machine scheduling with a common due date. Journal of Heuristics 2007;13:151–65.
[38] Kim KH, Moon KC. Berth scheduling by simulated annealing. Transportation Research Part B 2003;37:542–60.
[39] Candalino Jr. TJ, Kobza JE, Jacobson SH. Designing optimal aviation baggage screening strategies using simulated annealing. Computers and Operations Research 2004;31:1753–67.
[40] Lee DH, Cao Z, Meng Q. Scheduling of two-transtainer systems for loading outbound containers in port container terminals with simulated annealing algorithm. International Journal of Production Economics 2007;107:115–24.
[41] Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. Equations of state calculations by fast computing machines. Journal of Chemical Physics 1953;21:1087–92.
[42] Kirkpatrick S, Gelatt Jr. CD, Vecchi MP. Optimization by simulated annealing. Science 1983;220:671–80.
[43] Christofides N, Mingozzi A, Toth P. The vehicle routing problem. In: Christofides N, Mingozzi A, Toth P, Sandi C, editors. Combinatorial Optimization. Chichester, UK: Wiley; 1979. p. 315–38.