European Journal of Operational Research 116 (1999) 87–99

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

Theory and Methodology

A two-phase tabu search approach to the location routing problem

Dilek Tuzun *, Laura I. Burke

Department of Industrial and Manufacturing Systems Engineering, Mohler Lab. #200, Lehigh University, Bethlehem, PA 18015-1582, USA

Received 17 March 1997; accepted 9 February 1998

Abstract

In many distribution systems, the location of the distribution facilities and the routing of the vehicles from these facilities are interdependent. Although this interdependence has been recognized by academics and practitioners alike, attempts to integrate these two decisions have been limited. The location routing problem (LRP), which combines the facility location and the vehicle routing decisions, is NP-hard. Due to the problem complexity, simultaneous solution methods are limited to heuristics. This paper presents a two-phase tabu search architecture for the solution of the LRP. First introduced in this paper, the two-phase approach offers a computationally efficient strategy that integrates facility location and routing decisions. This two-phase architecture makes it possible to search the solution space efficiently, thus producing good solutions without excessive computation. An extensive computational study shows that the TS algorithm achieves significant improvement over a recent effective LRP heuristic.

Keywords: Tabu search; Location routing problem; Heuristics; Routing; Location

1. Introduction

Distribution of products to customers is one of the most important activities of a manufacturing company. According to Srivastava and Benton (1990), the overall cost of transportation and warehousing accounts for over 20% of the GNP, therefore substantial savings can be achieved by improving distribution systems even by only a small amount. The location of distribution facilities and the distribution of products from these facilities to customers are two key components of a distribution system. In various different settings, these two components are interdependent, therefore it is necessary to consider the facility location and the distribution decisions simultaneously.

In practice, there are two main ways of distributing products from facilities to customers. (1) Each delivery vehicle serves only one customer on a straight-and-back basis on a given route. This is the case when the customer demand is a full truckload (TL). (2) A vehicle stops at more than one customer on its route, which is the case if each

* Corresponding author. Present address: 1320 E. Algonquin Road Apt. 21, Schaumburg, IL 60173, USA. Tel.: +1 847 700 2443; fax: +1 847 364 1009; e-mail: Dilek.Tuzun@ual.com

PII: S 0 3 7 7 - 2 2 1 7 (9 8) 0 0 1 0 7 - 6

customer's requirement is less than a truckload (LTL). For the first case, it is appropriate to assume that the delivery cost can be represented by the moment sum cost function where unit shipment cost from a facility to a customer is assumed to be independent of the route taken to visit the customer. Then the total delivery cost is the sum, overall customers and facilities, of the product of the unit shipment cost from the facility to the customer and the number of units delivered to that customer.

However, when the customer demands are less than a truckload, the delivery cost to a customer depends on the routing of the delivery vehicles. Using the moment sum function ignores this interdependence between routing and location decisions. For LTL distribution systems, the routing decisions should be incorporated in the location models to represent them realistically. Integrated location routing models are used to solve the facility location problem (FLP) and the vehicle routing problem (VRP) simultaneously so that the interactions between the two decisions are reflected. Location routing models are especially necessary for systems where the time horizon for the facility location decisions are not too long, and location costs are comparable to the routing costs. Some areas where the integrated approach is applicable include the distribution in food and drink industries, delivery to retail shops, delivery of newspapers, and distribution of various consumer goods.

This paper introduces a novel two-phase architecture that integrates the location and routing decisions of the LRP. The two-phase approach coordinates two tabu search (TS) mechanisms – one seeking a good facility configuration, the other a good routing that corresponds to this configuration – in a computationally efficient algorithm. The same concept can be applied to other problems that include more than one level of decision making. There are many combinatorial problems in the area of logistics (and in many other areas of operations research) where a "multiple phase" TS algorithm can be applied. The promising computational results for the LRP suggest that the same approach can be taken for other problems with multiple levels of decision making.

This paper is organized as follows. We define the LRP and summarize the related literature in Section 2. Section 3 introduces the two-phase TS algorithm for the LRP. Then, the experimental design for the computational analysis is described in Section 4. Next, the computational results are discussed in Section 5. Finally, conclusions and suggestions for future work are given in Section 6.

2. Problem definition and related literature

The location routing problem (LRP) can be defined as follows: A feasible set of potential facility sites and locations and expected demands of each customer are given. Each customer is to be assigned to a facility which will supply its demand. The shipments of customer demand are carried out by vehicles which are dispatched from the facilities, and operate on routes that include multiple customers. There is a fixed cost associated with opening a facility at each potential site, and a distribution cost associated with any routing of vehicles that includes the cost of acquiring the vehicles used in the routing, and the cost of delivery operations. The cost of delivery operations is linear in the total distance traveled by the vehicles. The LRP is to determine the location of the facilities and the vehicle routes from the facilities to the customers to minimize the sum of the location and distribution costs such that the vehicle capacities are not exceeded.

In practice there may be several different constraints on the vehicle routes, such as limitations on the total time elapsed, or the total distance traveled. In this paper, we adopt one of the most common constraints: the total demand of customers served on a given route should not exceed the vehicle capacity. A mathematical formulation of the LRP we study – similar to Perl and Daskin (1985) three index formulation – is given in Appendix A.

Early works that address an integrated location routing problem concentrate on the location of a single facility. Among these is a study that analyzes the error resulting from using the moment sum cost function to represent the routing cost (Webb, 1968). By comparing the solutions of an approach that uses a moment sum cost function, and one that computes the actual multiple stop route costs for each potential location, Webb concludes that the moment sum approximation is not an accurate representation of the routing cost. Eilon et al. (1971) also addresses the inadequacy of using the moment sum approach for the routing cost for the single facility LRP. Using the asymptotic formula for the traveling salesman problem by Bearwood et al. (1959), they approximate the route length for the case where the customers are uniformly distributed in a square. Watson-Gandy and Dohrn (1973) use this route length estimator to solve a multi-facility LRP.

Once the locations of the facilities are determined, the LRP reduces to a multiple depot VRP. Both of these subproblems of the LRP, namely the FLP and the VRP, have been shown to be NP-hard (Cornuejols et al. (1977) and Karp (1972), Lenstra and Rinnooy Kan (1981)), thus the LRP also belongs to the class of NP-hard problems. Due to its complexity, exact solution approaches to the LRP have been very limited. One of the earliest was an exact algorithm for the single facility LRP without tour length restrictions due to Laporte and Nobert (1981). They formulated the problem as an integer linear program and used a constraint relaxation technique to solve it. Integrality is then achieved by branch and bound. Following this paper, Laporte et al. (1983, 1986) used a similar approach to solve the uncapacitated and capacitated multi-facility LRP. Finally, Bookbinder and Reece (1988) formulated a three layer (plants – distribution centers (DC's) – customers) multi-commodity, capacitated distribution system as a nonlinear mixed integer program, and applied Benders' decomposition to decompose the problem into its location and routing components. Due to the exponential growth in the problem size, these exact methods for the LRP have been limited to small and medium size instances (up to 20–50 customers). As the problem size gets larger, heuristic procedures prove to be the only viable alternative.

One of the earlier attempts to solve a practical LRP is a study due to Or and Pierskalla (1979) where the problem involves locating regional blood banks to serve hospitals. This study ignores the location part of the problem, and concentrates on the allocation and routing decisions. Another practical LRP model for a newspaper delivery system is described in Jacobsen and Madsen (1980) and Madsen (1983). They introduce three heuristics (ALA-SAV, SAV-DROP, and TREE-TOUR) to solve the three layer LRP for the system, and obtain a slight improvement over the current distribution strategy. Perl and Daskin (1985) propose a three-phase algorithm to solve a complex LRP which accounts for variable DC throughput costs and DC throughput capacities. Srivastava (1993) also suggests three heuristics for the LRP, and explores the effects of several environmental factors on the algorithm performance. The first heuristic, SAV1 assumes all facilities to be open initially, and uses approximate routing costs for open facilities to determine the facility to be closed. A modified version of the savings algorithm for the multiple depot case (Tillman, 1969) is used to approximate the routing costs. The algorithm iterates between the routing and facility closing phases until a desired number of facilities remain open. The second heuristic (SAV2) takes the opposite approach, and adds facilities one by one. The third heuristic, CLUST identifies clusters by generating the minimal spanning tree of customers and then separating it into a desired number of clusters using a density search technique. According to the computational experiment, all of these algorithms perform significantly better than the sequential approach implemented in the same paper. The sequential approach, which is commonly used in practice, first determines the facility locations using a moment sum approximation, and then solves the multi-depot routing problem applying the modified savings algorithm. One recent study by Chien (1993) attempts to estimate the routing cost using two different estimators based on the spatial characteristics of the problem, instead of performing the routing for every allocation. Finally, there are a few surveys in the literature that address the LRP. (See Madsen, 1981; Balakrishnan et al., 1987; Laporte, 1988).

As summarized above, few studies attempt to solve LRP's of realistic size. There remains a need for effective computational approaches to solve large LRP's. One other shortcoming of the literature is the lack of comparative studies that evaluate the relative performance of the heuristics. In this paper we compare the new two-phase TS approach with one of the recent heuristics by Srivastava, SAV1. Since most of the heuristics discussed above solve slightly different versions of the LRP, it is difficult to compare the effectiveness of Srivastava's algorithm with others. However, it yields substantial improvement over the sequential approach which solves the location and routing problems sequentially, and also over one other heuristic introduced in the same paper. (Performance of the CLUST algorithm is also comparable to SAV1, but we preferred SAV1 since its implementation is more straightforward.) Yet, its computational time requirement is reasonable for practical size problems. These results suggest that the algorithm is fairly effective in solving LRP's of realistic size. One important contribution of this paper is initiating a basis for evaluating the performance of LRP heuristics by comparing our heuristic with the SAV1 algorithm which demonstrates capability of solving realistic LRP's.

3. Two-phase tabu search algorithm for the LRP

One recent development in the solution of combinatorial problems is the introduction of metaheuristics such as tabu search, genetic algorithms, simulated annealing, and neural networks. All of these metaheuristics aim to search the solution space more effectively than conventional approaches using different strategies. They show great promise in solution of difficult combinatorial problems such as the LRP. Among these, TS explores the solution space by moving from a solution to its best neighbor, even if this results in a deterioration of the objective function value. This strategy allows the search to move out of the local optima and explore other regions of the solution space. (See Glover, 1995; Glover et al., 1993, for overviews of TS.) TS has been applied to both the facility location type problems and various forms of the VRP, and the results are encouraging. This suggests that TS can be used to solve the LRP which combines the two problems in an integrated model. In this paper, we propose a two-phase TS algorithm for the LRP. Fig. 1 presents a general diagram of this algorithm. The two-phase approach, introduced for the first time in this paper, aims to integrate two levels of decision making (location and routing) in a computationally efficient manner. In the location phase of the algorithm, a TS is performed on the location variables to determine a good configuration of facilities to be used in the distribution. For each of the location configurations visited during the location phase, another TS is run on the routing variables in order to obtain a good routing for the given configuration. This two-phase approach offers a simple and natural representation, since the LRP is decomposed into two subproblems in terms of two different types of decision variables. On the other hand, the two searches are coordinated so that an efficient exploration of the solution space is performed. Each time a move is performed on the location phase, the routing phase is started in order to update the routing according to the new configuration. However, since only a certain section of the routing is affected by the change in the depot configuration, it is possible to restrict the search to this section. Thus, the routing phase is a localized search, as opposed to a global exploration of all routing moves. This eliminates a lot of unnecessary computation, and allows the two-phase algorithm to find good solutions within reasonable computation time. Now, we define the elements of the TS algorithm.

(0) *Initialization*: The TS algorithm is initialized with one randomly selected facility open, and all of the other candidate facilities closed. We have chosen this initialization because of it is fast and simple, however, any initialization procedure may be used to initiate the TS algorithm. In particular, if the lower bound on the number of facilities to open is known in advance, it is possible to initialize the algorithm with this lower bound. If the algorithm is initiated with one open facility, all customers are simply assigned to this facility. For any other initialization, each customer is assigned to the nearest open facility. Given the allocation of the customers to the facilities, a separate routing problem is solved for each open facility. We use the original savings algorithm (Clarke and Wright, 1964) followed by a simple 2-opt procedure

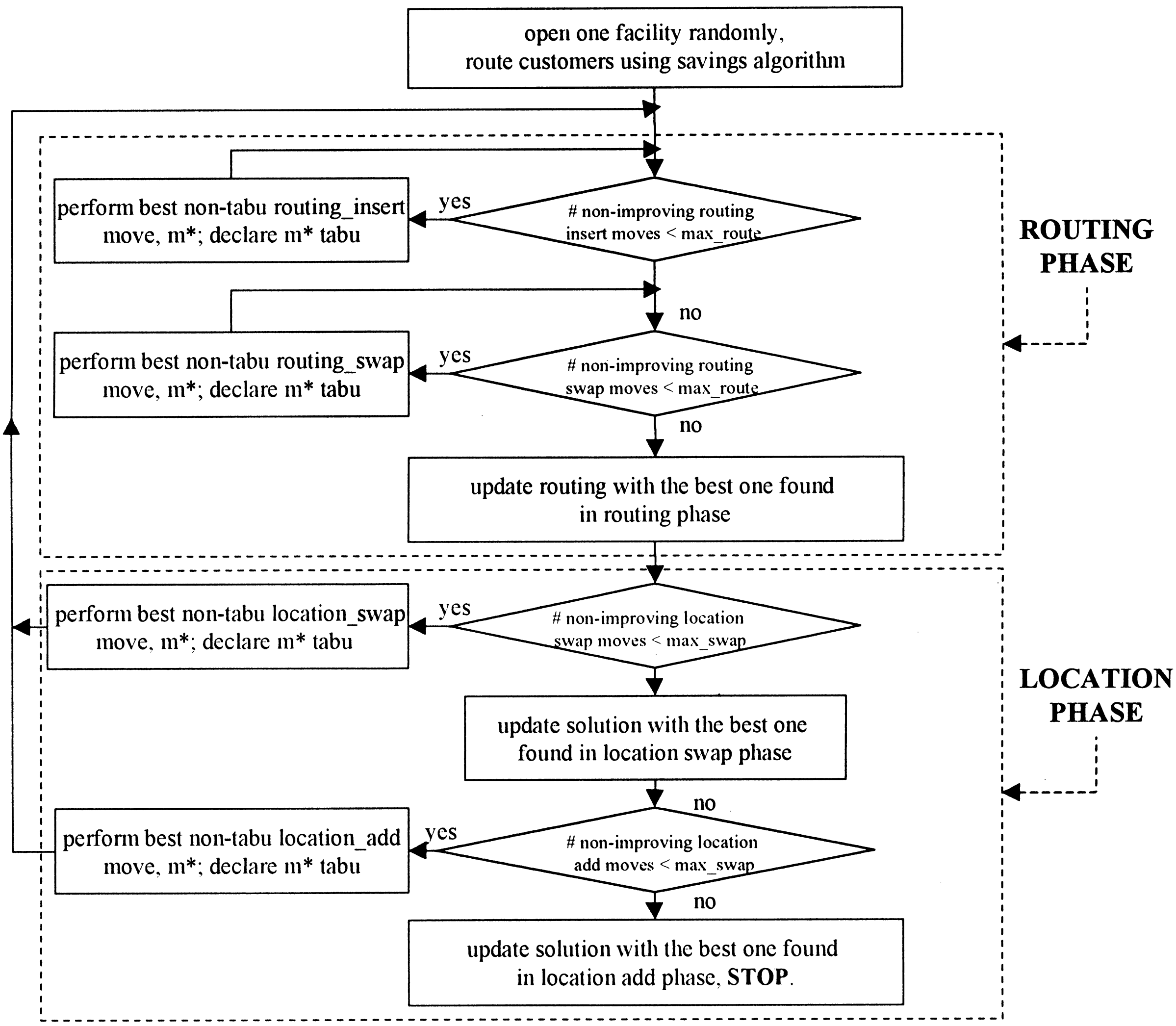

Fig. 1. Two-phase tabu search alogrithm for the LRP.

(Lin, 1965) to obtain the initial routing for the open facility.

(1) *Location phase*: In this phase, we apply two different type of moves to go from one facility configuration to the other: swap moves and add moves.

(1.1) *Swap moves*: For a given number of facilities, the location phase first performs swap moves which close one of the open facilities, and open one that is currently closed simultaneously. Swap moves keep the number of open facilities in the solution constant, and search for a good configuration for a certain number of facilities. At each iteration, the location phase searches for the best swap move to perform. In order to select the best swap move, it is necessary to evaluate the cost of a swap move. The difference in the fixed cost when we open one facility and close another is straightforward (fixed cost of the facility to open – fixed cost of the facility to close). However, the difference in the routing cost is difficult to estimate. To do this, we take a simplistic approach, and assume that each customer is assigned to the open facility that is closest to it. The difference in routing cost is then estimated using the difference in the direct distance between the customer and the facility according to the new and old assignments. The swap move evaluation is the sum of this routing cost estimate and the difference in the fixed cost. The swap move which yields the smallest evaluation is then performed, and both the move and its reverse are declared tabu for a number of iterations. After the swap move is performed, the search resumes to the routing phase to update the routing according to the swap move.

(1.2) *Add moves*: Swap moves are applied until a *max_swap* number of nonprofitable moves (moves with positive cost) are completed. Having explored the configurations with the current number of facilities, the search mechanism then increases the number of facilities by applying an add move. An add move opens one of the currently closed facilities, and therefore increases the number of facilities by one. The facility to be added is the one whose addition yields the minimum estimated cost. Here, the routing cost is again estimated using the difference in direct distances for the customer assignments before and after the add move. Since opening a facility can only improve the routing cost estimate, this cost is always negative. The fixed cost of the facility to be opened is then added to the routing estimate in order to calculate the overall cost estimate. As in the swap move, the search again returns to the routing phase in order to update the routing after the add move. After one add move, the search continues with a series of swap moves until the termination criterion is satisfied.

The TS is terminated when a *max_add* number of add moves are performed without any improvement over the best objective function value. This termination criterion enables the search to terminate without exploring configurations with more facilities than necessary. Since the number of facilities to be opened is not known a priori, without such termination criterion, the search would continue until solutions with all candidate facilities open are explored. Assuming that the number of candidate locations is much larger than the number of facilities to be opened, starting with one facility, and using the max_add threshold for termination reduces the exploration of many unfavorable configurations.

An important feature of the location phase is the separation of the swap and add moves. Since the cost of a location move (swap or add) is only an estimate, it does not reflect the tradeoff between the fixed cost of opening facilities, and the routing cost from those facilities to the customers. Therefore, if moves adding, dropping or swapping facilities are allowed at each iteration, without a precise estimate of the costs, the TS may lead to too few or too many facilities. Evaluating the moves that change the number of facilities separately from the rest reduces the error that is caused by cost estimation.

(2) *Routing phase*: After each swap or add move is performed, the routing phase is started from the best routing found for the previous facility configuration in order to modify the routing according to the current facility configuration. First, the customers are reassigned to the closest open facility. For each open facility, the number of changes to the customer allocation (Δc) is recorded. Δc includes the customers that are reassigned to this facility, as well as the customers reassigned to another facility from this facility after the location move. Δc is a measure of how much a facility is affected from the location move. If Δc is larger than a threshold Δc_max, then the facility should be included in the routing phase; i.e., its customers are considered for the routing moves. Moreover, since the facility's routing will be largely affected by the changes to its customer allocation, the customers assigned to this facility are rerouted using the savings algorithm at the beginning of the routing phase. On the other hand, if Δc does not exceed the threshold, customers newly assigned to this facility are simply inserted at the best position available, and the ones that are reassigned to other facilities are deleted from its routes. Apart from this, the routing for this facility remains unchanged during the routing phase. This threshold eliminates many unnecessary routing moves from consideration, and reduces computation.

Two set of moves are performed sequentially in the routing phase: insert moves, and swap moves.

(2.1) *Insert moves*: This type of move inserts a customer, which is assigned to a facility that exceeds the threshold Δc_max, to a new position on a route originating from its current facility, or any other open facility that is close enough to the customer. For a facility to be considered for insertion, it should be one of the *f_max* open facilities closest to the customer. Using this threshold, we avoid trying many nonprofitable moves which try to insert the customer to facilities that are far away. Unlike the location moves, routing moves (both insert and swap) are very straightforward to evaluate. Thus, in order to select the best move, the actual cost of each move is calculated using the difference in the route length. All eligible moves that insert customers, which are currently assigned to the facilities exceeding the threshold, to their *f_max* closest facilities are evaluated in the order they appear on the routes. During the evaluation process, as soon as a profitable move is found, it is performed, and applying an insert move to this customer is declared tabu for a number of iterations. If no profitable move is found by the time all of the moves are evaluated, the nontabu move with the least cost is performed. This hill climbing approach is usually as effective as the steepest descent for TS applications, and it is computationally more efficient. (Osman (1993) reports that the hill climbing approach is better than the steepest descent for his TS implementation on a single depot VRP.) Insert moves are terminated after a *max_route* number of iterations are performed without improvement over the best solution found for the current facility configuration. This is followed by a set of swap moves.

(2.2) *Swap moves*: This type of move swaps the position of any two customers that are currently assigned to a facility that exceeds the threshold Δc_max. Similar to the insert moves, a customer can only be swapped with one of the *c_max* customers closest to it to eliminate unnecessary computation. Since these closest customers may be assigned to the same, as well as different facilities, swap moves can be between customers that are assigned to the same facility, or between those assigned to different facilities. All moves that swap customers (belonging to the eligible facilities) with their *c_max* closest customers are evaluated in the order they appear on the routes. The same hill climbing approach is taken for these moves, i.e. a profitable move is performed as soon as it is encountered. After the swap move is done, swapping these two customers is declared tabu for a number of iterations. Swap moves are also terminated after a *max_route* number of routing iterations are performed without improvement.

Aspiration criterion: A tabu routing move is applied if it is a profitable move, i.e. if the cost of the move is negative. We do not use an aspiration criterion for the location moves since the move evaluation value is only an estimate, and does not reflect the exact cost of the move.

Tabu attributes and tabu duration: For add moves in the location phase and insert moves in the routing phase, the tabu attribute is respectively the facility being added and the customer being inserted. For the swap moves in the location (routing) phase, the two facilities (customers) being swapped are the two tabu attributes that are recorded. For the attribute(s) that are declared tabu, we use a probabilistic tabu duration similar to the one proposed by Taillard (1991) to reduce the possibility of cycling. Both the location and the routing attributes are declared tabu for a tabu duration that is randomly selected from a uniform interval. The bounds of the intervals for the location and routing moves are set separately. In most location routing problems, the number of facility candidates is much smaller than the number of customers, therefore it is appropriate to set the tabu durations for the location attributes relatively short.

The TS algorithm described above aims to explore the solution space of the LRP intelligently. First, it eliminates exploration of facility configurations that are not promising by quitting the search after a number of nonprofitable location moves. Second, it starts the routing phase from a good routing solution for the previous facility configuration. Since the routing is only partially affected by one location move, this routing solution is not very far away from a good solution for the new facility configuration. Therefore, the routing phase does not require excessive computation each time it is restarted. This is also linked to a concept called the "proximate optimality principle" discussed in Glover and Laguna (1993), and Glover et al. (1993). This concept stipulates that good solutions at one level are likely to be found close to good solutions at another level. In TS, the proximate optimality principle can be exploited by remaining at each successive level for a chosen number of iterations, and then incorporating the best solution found to initiate a move to the next level. Each configuration of the facility variables can be viewed as a level in the two-phase algorithm. Following the principle, the procedure should remain at each configuration of the location variables for a chosen number of iterations, which in fact corresponds to the routing phase of the algorithm. Then, the best routing found in the routing phase is used to initiate a move to the next level or the next configuration of the location variables.

4. Experimental design

In order to evaluate the performance of our two-phase TS algorithm, we compare it to one of the most recent LRP heuristics in the literature: SAV1 algorithm by Srivastava (1993). This algorithm compares favorably to the other two heuristics proposed in the same paper (SAV2 and CLUST), and also to the sequential. The comparison of the two-phase TS algorithm with the SAV1 heuristic initiates a basis for evaluating the performance of LRP heuristics which is lacking in the LRP literature.

As also mentioned in Srivastava (1993), performance of heuristics may vary significantly depending on the characteristics of the problem instance. In order to explore how the performance of the TS algorithm is affected by problem characteristics, we generated a test set that consists of a wide variety of problems. *Problem size*, *spatial distribution of customers*, and *route structure* are three LRP characteristics that may affect algorithm performance. As for any other problem, the size of an LRP instance may affect both the solution quality and the computational requirements of a heuristic. In order to control the size of the problem, we use two factors in our experimental design: number of customers (n) and number of candidate facilities (m). In the experiment n was set at levels 100, 150 and 200, whereas m was set at 10 and 20. Spatial distribution of customers may also affect the performance of an LRP heuristic (see Srivastava (1993) for a similar analysis). While some algorithms perform better on problems where customers are uniformly distributed within the service area, others do well on problems where customers appear in clusters at certain locations. Spatial distribution was also controlled using two factors: number of clusters (*cl*), and clustering ratio (*cl_ratio*). *cl* is simply the number of areas where the customer density is high. It is set at 3 levels: 0, 3, and 5, where level 0 refers to uniformly distributed customers. *cl_ratio* is the ratio of number of customers that belong to a cluster to the total number of customers. This factor is at 2 levels: 75% and 100%. The last factor, vehicle capacity (*vc*), is related to the structure of the routes. For all of the instances, we generated customer demand from a uniform distribution in the range [10, 20]. Since the average demand is the same for all problem instances, *vc* is the only factor that controls number of customers to be served on a single tour. As we increase the value of *vc*, longer routes will be formed by the algorithm, making the routing component of the LRP more challenging. Two levels were chosen for *vc*: 150 and 300. Since the average demand is 15 for all problems, these two levels correspond to an average of 10 and 20 customers on a route respectively.

Five problems were generated for each cell of the full factorial design with the five factors described, resulting in 360 problems total. In all of these instances, fixed cost was set equal to 1.0 for all facilities. Table 1 gives a summary of the factors used in the experimental design.

Table 1
Factors and their levels used in the experimental design

Factor	Levels
Algorithm used	TS, SAV1
Number of customers (n)	100, 150, 200
Number of clusters (*cl*)	0, 3, 5
Clustering ratio (*cl_ratio*)	75%, 100%
Number of candidate facilities (m)	10, 20
Vehicle capacity (*vc*)	150, 300

5. Computational results

In this section, we compare the performance of the two-phase TS algorithm to that of SAV1 algorithm using the test set of 360 problems described in the previous section. Both of the algorithms were coded in C and were run on a Gateway 2000 PC Model G6-266M with 266 MHz Pentium II processor. The TS algorithm was run using the following threshold parameters: $max\_add = 3$, $max\_swap = 7$, $max\_route = 10$, $f\_max = 4$, $c\_max = 8$, and $\Delta c\_max = 0.5(avg\_nc)$ where *avg_nc* (average number of customers/route) = *vc/average demand*. As for the tabu durations, these were generated uniformly from intervals [5, 8] and [10, 13] for the location and routing attributes respectively. Here we should note that we have experimented with lower tabu duration ranges (such as [2, 5] for location and [4, 7] for routing) and the solution quality was slightly

Table 2
Sample computational results of the two-phase TS algorithm vs. SAV1

Problem	*n*	*cl*	*cl_ratio*	*m*	*vc*	SAV1		Tabu search	
						cost	cpu	cost	cpu
P111112	100	0	0.75	10	300	1596.02	1	1556.64	5
P111122	100	0	0.75	20	300	1557.07	3	1531.88	3
P111212	100	0	1	10	300	1457.01	1	1443.43	3
P111222	100	0	1	20	300	1555.89	3	1511.39	4
P112112	100	3	0.75	10	300	1245.75	1	1231.11	4
P112122	100	3	0.75	20	300	1140.13	3	1132.02	2
P112212	100	3	1	10	300	830.46	1	825.12	3
P112222	100	3	1	20	300	744.67	3	740.64	3
P113112	100	5	0.75	10	300	1326.09	1	1316.98	3
P113122	100	5	0.75	20	300	1316.54	3	1274.5	4
P113212	100	5	1	10	300	927.17	1	920.75	4
P113222	100	5	1	20	300	1109.31	3	1042.21	3
P131112	150	0	0.75	10	300	2066.22	3	2000.97	12
P131122	150	0	0.75	20	300	1977.41	8	1892.84	12
P131212	150	0	1	10	300	2113.57	2	2022.11	14
P131222	150	0	1	20	300	1931.63	8	1854.97	13
P132112	150	3	0.75	10	300	1653.43	4	1555.82	9
P132122	150	3	0.75	20	300	1554.52	9	1478.8	12
P132212	150	3	1	10	300	1237.96	2	1231.34	9
P132222	150	3	1	20	300	953.26	9	948.28	9
P133112	150	5	0.75	10	300	1848.43	3	1762.45	9
P133122	150	5	0.75	20	300	1496.37	7	1488.34	9
P133212	150	5	1	10	300	1247.28	3	1264.63	10
P133222	150	5	1	20	300	1192.58	8	1182.28	9
P121112	200	0	0.75	10	300	2463.19	5	2379.47	22
P121122	200	0	0.75	20	300	2289.35	15	2211.74	22
P121212	200	0	1	10	300	2395.86	4	2288.17	23
P121222	200	0	1	20	300	2417.32	19	2355.81	26
P122112	200	3	0.75	10	300	2203.37	9	2158.6	20
P122122	200	3	0.75	20	300	1805.91	20	1787.02	18
P122212	200	3	1	10	300	1560.77	8	1549.79	18
P122222	200	3	1	20	300	1122.85	19	1112.96	18
P123112	200	5	0.75	10	300	2312.78	5	2056.11	23
P123122	200	5	0.75	20	300	2046.46	15	2002.42	20
P123212	200	5	1	10	300	1849.8	6	1877.3	20
P123222	200	5	1	20	300	1423.77	15	1414.83	17
Average						1610.284	6.39	1566.77	11.5

inferior for lower tabu durations, thus we used the higher ranges for the computational experiment. We also experimented with a higher value of *max_route* (20). For this value, the cpu time increased 40% whereas the solution quality improved only by 0.03%, therefore *max_route* parameter was set to 10 for the experiment.

We report our computational results in terms of both solution quality (measured by cost) and cpu time. In Table 2, a 36 problem sample of the computational results is given. These problems are half of the first set of 72 problems (out of the five sets with all combinations of problem characteristics) with vehicle capacity equal to 300. Since interaction of vehicle capacity and algorithm is insignificant, we display the results for different levels the remaining factors.

Comparison of solution quality: As a result of the ANOVA, we find that all six factors (five problem characteristics and algorithm used) affect the total cost significantly at level 0.005. The TS algorithm is found to be significantly better than the SAV1 algorithm: average total cost is 1975.676 for TS, and 2028.703 for SAV1. This translates into 2.61% savings over the SAV1 algorithm. In terms of the solution quality, none of the interaction effects of the algorithm and a problem characteristic is significant.

Comparison of cpu times: As a result of the ANOVA performed on the cpu times, we see that all factors are significant at level 0.001. All interaction effects of algorithm except for the one with vehicle capacity are also significant at level 0.001. In Tables 3–6, we tabulate the average cpu times for each level of these interactions. In Table 3, we see that cpu time requirement of TS is about twice that of SAV1 for both levels of n, i.e., the rate of increase in cpu time as n increases is about the same for both algorithms. Compared to a one pass heuristic, TS is expected to require more time, but the fact that it significantly outperforms SAV1 and its cpu time scales similarly is especially encouraging.

Table 3
Interaction effect of algorithm and n on cpu times in seconds

Algorithm	Number of customers (n)		
	100	150	200
TS	3.550	10.400	20.025
SAV1	2.050	5.058	10.908

Table 4
Interaction effect of algorithm and cl on cpu times in seconds

Algorithm	Number of clusters (cl)		
	0	3	5
TS	12.816	10.416	10.741
SAV1	5.583	6.625	5.808

Table 5
Interaction effect of algorithm and *cl_ratio* on cpu times in seconds

Algorithm	Clustering ratio (*cl_ratio*)	
	75%	100%
TS	11.966	10.683
SAV1	5.955	6.055

Table 6
Interaction effect of algorithm and m on cpu times in seconds

Algorithm	Number of facilities (m)	
	10	20
TS	11.233	11.416
SAV1	3.055	8.955

On the other hand, the two algorithms respond differently to cl (see Table 4). While TS spends more time on uniformly distributed problems, average cpu time for SAV1 is highest for clustered problems. (In a five-cluster problem, the variance of distances between the customers is less than a three-cluster one. Therefore we can say that the three-cluster problem is the "most clustered" of all.) The decrease in cpu time for the TS algorithm when the problem is clustered can be attributed to the reduced number of viable customer to facility assignments. When customers are grouped at a few locations, it is obvious that opening certain facilities (the ones that are not close to any clusters) is not profitable. Therefore, the TS algorithm converges faster. On the other hand, SAV1 requires more time when the customers are clustered. As we described in Section 2, this algorithm starts with all facilities opened, and drops a selected facility at each iteration. It stops when it is not profitable to drop another facility anymore. Since clustered problems usually require fewer facilities to be opened, SAV1 requires more iterations to find a solution. The interaction effect of algorithm and *cl_ratio* also displays a similar behavior for the two algorithms (see Table 5). While TS runs faster for more clustered problems ($cl\_ratio = 100\%$), SAV1 requires more time for such instances.

Finally, in Table 6 the behavior of the two algorithms varies with the number of candidate facilities (m). While TS takes almost the same amount of time as m is doubled, cpu requirement of SAV1 increases almost three times. Since the solution quality of TS is significantly better than SAV1, and the difference in cpu times decreases as m increases, TS becomes more attractive when there are many candidate facilities to choose from.

To summarize, the two-phase TS algorithm provides substantial savings over the SAV1 algorithm but it requires more computational time. Considering that the SAV1 heuristic is a one pass greedy heuristic, the additional computation required by the TS algorithm is quite reasonable. As we stated earlier, distribution costs account for a large portion of the product cost in most industries. Therefore, in many distribution systems it is readily justified to invest more computational time in order to improve the solution quality even by a small amount.

6. Conclusions and suggestions for future work

This paper introduces a new solution approach to the LRP. Previous studies have been limited to a few conventional heuristics, and exact solution techniques which are limited to small size problems. In this study, we present a two-phase TS algorithm that iterates between location and routing phases in order to search for better solutions. The two-phase approach introduced in this paper aims to explore the solution space efficiently. The TS algorithm was tested on a large test set that contains a wide variety of problems. The paper also provides a comparison of the two-phase algorithm with a recent LRP heuristic. Since this is the first paper that compares two different LRP heuristics, its contribution to forming a basis for performance evaluation of heuristics is noteworthy. The results of the comparative study are also promising: the two-phase TS algorithm not only performs significantly better than the other heuristic, but its computational time requirement is also quite reasonable for realistic size problems. One other important finding is that the computation times required by both algorithms increase by approximately the same factor as the number of customers increase, moreover, the difference in computation time of the two algorithms reduces as number of candidate facilities increases. Therefore, the two-phase TS algorithm becomes a more attractive solution method as the problem size increases.

The two-phase TS algorithm described in this paper utilizes only short term memory. Although it provides substantial improvement in solution quality, it may be possible to further improve performance by implementing longer term memory of TS. As mentioned in Glover (1995), incorporating long term strategies into TS is crucial to fully utilizing its capabilities. We leave long term implementation for future work.

The concept of "multiple phase" TS is introduced here for the first time. The encouraging results suggest that it can be applied to other combinatorial problems that contain multiple levels of decision making. In this application, there are two levels of decision making (location and routing) where the decisions made at one level affect the other. A similar approach may be used for any combinatorial problem with more than one type of decision variable that are interdependent. Multi-level location problems with more than one type of facility to be located, design of hub and spoke type of networks, shipment planning problems with multiple modes of transportation are possible application areas in the area of logistics.

Acknowledgements

This study is partially sponsored by a National Science Foundation (NSF) National Young Investigator Award to Dr. Laura I. Burke.

Appendix A: Mixed integer programming formulation

$$\text{Minimize} \sum_{i\in S}\sum_{j\in S}\sum_{k\in V} C_{ij}X_{ijk} + \sum_{k\in V}\left(C_k\sum_{r\in G}\sum_{j\in H}X_{rjk}\right) + \sum_{r\in G}F_rZ_r$$

subject to:

$$\sum_{k\in V}\sum_{i\in S}X_{ijk} = 1 \quad \forall j\in H, \tag{A.1}$$

$$\sum_{j\in H}\sum_{i\in S}q_jX_{ijk} \leqslant Q_k \quad \forall k\in V, \tag{A.2}$$

$$\sum_{i\in S}X_{ipk} - \sum_{j\in S}X_{pjk} = 0 \quad \forall k\in V,\ p\in S, \tag{A.3}$$

$$\sum_{r\in G}\sum_{j\in H}X_{rjk} \leqslant 1 \quad \forall k\in V, \tag{A.4}$$

$$\sum_{k\in V}X_{rmk} + Z_r + Z_m \leqslant 2 \quad \forall m = 1,\ldots,R, \quad r\in G, \tag{A.5}$$

$$\sum_{k\in V}\sum_{j\in H}X_{rjk} - Z_r \geqslant 0 \quad \forall r\in G, \tag{A.6}$$

$$\sum_{j\in H}X_{rjk} - Z_r \leqslant 0 \quad \forall k\in V,\ r\in G, \tag{A.7}$$

$$R_i - R_j + (R+N)\sum_{k\in V}X_{ijk} \leqslant R+N-1 \quad \forall i,j\in H, \quad i\neq j, \tag{A.8}$$

$$X_{ijk} = 0 \text{ or } 1 \quad \forall i,j\in S,\ k\in V, \tag{A.9}$$

$$Z_r = 0 \text{ or } 1 \quad \forall r\in G. \tag{A.10}$$

Model parameters

- $\boldsymbol{G}$ — $\{r \mid r = 1,\ldots,R\}$ is the set of R feasible sites of candidate facilities;
- $\boldsymbol{H}$ — $\{i \mid i = R+1,\ldots,R+N\}$ is the set of N customers to be served;
- $\boldsymbol{S}$ — $\{G\}\cup\{H\}$ is the set of all feasible sites and customers (also referred to as nodes);
- $\boldsymbol{V}$ — $\{v_k \mid k = 1,\ldots,K\}$ is the set of K vehicles available for routing from the facilities;
- $\boldsymbol{C_{ij}}$ — average annual cost of travelling from node i to node j, $i\in S$, $j\in S$.
- $\boldsymbol{C_k}$ — annual cost of acquiring vehicle k $(k = 1,\ldots,K)$;
- $\boldsymbol{F_r}$ — annual cost of establishing and operating a facility at site r $(r = 1,\ldots,R)$;
- $\boldsymbol{q_j}$ — average number of units demanded by customer j, $j\in H$;
- $\boldsymbol{Q_k}$ — capacity of vehicle k $(k = 1,\ldots,K)$;
- $\boldsymbol{d_{ij}}$ — distance from node i to node j;

Decision variables

$$x_{ijk} = \begin{cases} 1 & \text{if vehicle } k \text{ goes from node } i \text{ to node } j,\ i\in S,\ j\in S,\ k\in V,\ i\neq j, \\ 0 & \text{otherwise,} \end{cases}$$

$$Z_r = \begin{cases} 1 & \text{if a facility is established at site } r,\ r\in G, \\ 0 & \text{otherwise,} \end{cases}$$

$\boldsymbol{R_i}$ are continuous variables used in the subtour breaking constraints.

In this formulation, the objective function minimizes the total cost of routing and acquisition of the vehicles, and locating and operating the depots. The first set of constraints ensures that each customer is served by one and only one vehicle. The second set ensures that the vehicle capacity constraint is not exceeded for any of the vehicles used in routing. The third set of constraints are the route continuity constraints which imply that every point that is entered by a vehicle should be left by the same vehicle. Constraint set 4 guarantees that each vehicle is routed from at most one depot. Constraint set 5 ensures that there are no links between any two depots. Constraint sets 6 and 7 requires that a vehicle is routed from a depot if and only if that depot is opened. Constraint set 8 consists of the subtour elimination constraints which guarantee that each tour must contain a depot from which it originates, i.e. none of the tours consists of customers only. The last two sets of constraints are the integrality constraints.

References

Balakrishnan, A., Ward, J.E., Wong, R.T., 1987. Integrated facility location and vehicle routing models: Recent work and future prospects. American Journal of Mathematical and Management Sciences 7, 35–61.

Bearwood, J., Halton, J.H., Hammersley, J.M., 1959. The shortest path through many points. Proceedings of the Cambridge Philosophical Society 55, 299–327.

Bookbinder, J.H., Reece, K.E., 1988. Vehicle routing considerations in distribution system design. European Journal of Operational Research 37, 204–213.

Chien, W.T., 1993. Heuristic procedures for the practical-sized uncapacitated location – capacitated routing problems. Decision Sciences 24 (5), 995–1021.

Clarke, G., Wright, J., 1964. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 12 (4), 568–581.

Cornuejols, G., Fisher, M.L., Nemheuser, G.L., 1977. Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. Management Science 23 (8), 1977.

Eilon, S., Watson-Gandy, C.D.T., Christofides, N., 1971. Distribution Management, Mathematical Modeling and Practical Analysis, Griffin, London.

Glover, F., 1995. Tabu search fundamentals and uses, Technical Paper. University of Colorado, Boulder, CO.

Glover, F., Laguna, M., 1993. Tabu Search, in: Reeves, C. (Ed.), Modern Heuristic Techniques for Combinatorial Problems. Blackwell Scientific Publications, Oxford.

Glover, F., Taillard, , É.de Werra, D., 1993. A user's guide to tabu search. Annals of Operations Research 41, 3–28.

Jacobsen, S.K., Madsen, O.B.G., 1980. A comparative study of heuristics for a two-level routing-location problem. European Journal of Operational Research 5, 378–387.

Karp, R., 1972. Reducibility among combinatorial problems, in: Miller, R., Thatcher, J. (Eds.), Complexity of Computer Computations. Plenum Press, New York, pp. 85–104.

Laporte, G., 1988. Location routing problems, in: Golden, B.L., Assad, A.A. (Eds.), Vehicle Routing: Methods and Studies. North-Holland, Amsterdam.

Laporte, G., Nobert, Y., 1981. An exact algorithm for minimizing routing and operating costs in depot location. European Journal of Operational Research 6, 224–226.

Laporte, G., Nobert, Y., Arpin, D., 1986. An exact algorithm for solving a capacitated location-routing problem. Annals of Operations Research 6, 293–310.

Laporte, G., Nobert, Y., Pelletier, P., 1983. Hamiltonian location problems. European Journal of Operational Research 12, 80–87.

Lenstra, J.K., Rinnooy Kan, A.H.G., 1981. Complexity of vehicle routing and scheduling problems. Networks 11, 221–227.

Lin, S., 1965. Computer solutions of the TSP. Bell Systems Technical Journal 44, 2245–2269.

Madsen, O.B.G., 1981. A survey of methods for solving combined location-routing problems, in: Jaiswal, N.K. (Ed.), Scientific Management of Transport Systems. North-Holland, Amsterdam.

Madsen, O.B.G., 1983. Methods for solving combined two level location routing problems of realistic dimensions. European Journal of Operational Research 12, 295–301.

Or, I., Pierskalla, W.P., 1979. A transportation location-allocation model for regional blood banking. AIIE Transactions 11, 86–95.

Osman, I.H., 1993. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. Annals of Operations Research 41, 421–451.

Perl, J., Daskin, M., 1985. A warehouse location-routing problem, Transportation Research 19B (5), 381-396.

Srivastava, R., 1993. Alternate solution procedures for the location-routing problem. Omega International Journal of Management Science 21 (4), 497–506.

Srivastava, R., Benton, W.C., 1990. The location-routing problem: Considerations in physical distribution system. Computers and Operations Research 17 (5), 427–435.

Taillard, E., 1991. Robust taboo search for the quadratic assignment problem. Parallel Computing 17, 443–455.

Tillman, F., 1969. The multiple terminal delivery problem with stochastic demands. Transportation Science 3, 192–204.

Watson-Gandy, C.T.D., Dohrn, P.J., 1973. Depot location with van salesman – A practical approach. Omega 1, 321–329.

Webb, M.H.T., 1968. Cost functions in the location of depots for multiple-delivery Journeys. Operational Research Quarterly 19, 311–328.