PERGAMON

Intl. Trans. in Op. Res. 6 (1999) 665–678

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

www.elsevier.com/locate/orms

An adaptive tabu search algorithm for the capacitated clustering problem

Paulo M. França[a,*], Nelida M. Sosa[b], Vitoria Pureza[c]

[a]*Faculdade de Engenharia Elétrica, Universidade Estadual de Campinas, C.P. 6101, Campinas, SP 13081-970, Brazil*
[b]*Universidad Nacional del Altiplano, Peru*
[c]*Universidade Federal de S Carlos, Brazil*

Received 23 July 1998; received in revised form 10 February 1999; accepted 28 May 1999

Abstract

In the Capacitated Clustering Problem, a given set of customers with distinct demands must be partitioned into p clusters with limited capacities. The objective is to find p customers, called medians, from which the sum of the distances to all other customers in the cluster is minimized. In this article, a new adaptive tabu search approach is applied to solve the problem. Initial solutions are obtained by four constructive heuristics that use weights and distances as optimization criteria. Two neighborhood generation mechanisms are used by the local search heuristic: *pairwise interchange* and *insertion*. Computational results from 20 instances found in the literature indicate that the proposed method outperforms alternative metaheuristics developed for solving this problem.

Keywords: Capacitated clustering problems; Metaheuristics; Tabu search

1. Introduction

In the Capacitated Clustering Problem (CCP), a given set of customers with corresponding weights or demands must be partitioned into p clusters with limited capacities. Each customer must be assigned to exactly one cluster. The objective is to find p customers, called medians, from which the sum of the distances to all other customers in the cluster is minimized.

* Corresponding author. Tel.: +55-19-788-3774; fax: +55-19-289-1395.
E-mail address: franca@densis.fee.unicamp.br (P.M. França)

PII: S0969-6016(99)00017-9

The CCP can be viewed as a special case of the Capacitated Facility Location Problem with a single source constraint, as well as the Capacitated *p*-Median Problem. Such clustering problems arise in many practical situations including the following:

- Consolidation of customer orders for vehicle shipment (Koskosidis and Powell, 1992; Bramel and Simchi-Levi, 1995).
- Assignment of customers to vehicles for multi-vehicle Dial-A-Ride transit systems (Jaw et al., 1986).
- Grouping of constituencies to define boundaries of political districts (Bourjolly et al., 1981).
- Design of sales force territories (Mulvey and Beck, 1984).

Comprehensive surveys of clustering models and their applications can be found in Refs. (Kaufman and Rousseeuw, 1990; Everitt, 1993).

The CCP is a complex combinatorial problem which has been proved to be NP-complete (Garey and Johnson, 1979). Due to its complexity, exact algorithms are not expected to be able to handle problems of the dimensions found in real-world applications, and most methods proposed to solve it are heuristic-based. These can be broadly classified into constructive heuristics and improvement heuristics. In constructive methods, a feasible solution is constructed through the successive addition of new elements according to some criterion. The well known *greedy* algorithms are examples of this class of heuristics. In improvement heuristics, a mechanism that iteratively interchanges elements of a solution is used to improve the objective function. Local search or neighborhood search are examples of improvement methods. The main disadvantage in using a local search method is that it may result in unsatisfactory local optima.

Metaheuristics are global optimization procedures that are superimposed on local search methods so that local optima can be overcome. In the past decade, a number of metaheuristics have been proposed and successfully applied in solving a variety of combinatorial problems. The most relevant are Tabu Search, TS (Glover and Laguna, 1997), Simulated Annealing, SA (Kirkpatrick et al., 1983; Cerny, 1985), GRASP (Feo and Resende, 1989) and Genetic Algorithms (Holland, 1975). Most metaheuristics are provided with a mechanism that allows the objective function to deteriorate, so they are capable of escaping from local optima.

TS starts from an initial solution and moves to the next solution by selecting the best of its neighbors. If the current solution is a local minimum, this means accepting a non-improving movement. However, since the search always looks for the best movement it may return to the local minimum from which it has just emerged. To prevent such cycling, the reversal of a move that has just been performed is inserted in a constantly updated *tabu list* and is forbidden (tabu) during a certain number of iterations. In practice, such restrictions are applied to move constituents or attributes rather than the move itself. A move will then be forbidden if some (or all) current tabu conditions of its attributes are satisfied. In this case, we say the tabu rule has been activated.

Various other elements and strategies are included in the TS methodology. Two of the most popular and powerful of these TS strategies for enhancing the performance of the method are diversification and intensification. Intensification modifies rules related to choice of movements so that historically good solutions are favored, while diversification modifies rules in order to incorporate features which have not been encountered in the solutions up to that time.

Diversification aims to direct the search to as yet unexplored regions. A vast literature has been devoted to successful TS applications for solving a variety of combinatorial optimization problems (Osman and Christofides, 1994; Gendreau et al., 1994; França et al., 1995; França et al., 1996; Glover et al., 1999); for an annotated bibliography on metaheuristics, the reader may refer to Laporte and Osman (1994) or to the book of Aarts and Lenstra (1997).

The present paper proposes a TS-based heuristic to solve the CCP. Our TS implementation incorporates a new adaptive scheme, which allows the search to integrate automatically the diversification and intensification phases. Another feature of this adaptive TS approach is its ability to attain high quality solutions using simple neighborhood structures. Moreover, the proposed method reduces considerably the tedious computational tests necessary to set adequate TS parameters such as tabu tenure, or the number of iterations during which a recent movement will be considered tabu. This paper is organized as follows. In Section 2, the CCP is defined and the most relevant methods of solution which have been proposed are surveyed. Constructive methods to generate feasible initial solutions are presented in Section 3 and the local search heuristic and adaptive TS approach are described in Section 4. Section 5 reports the results obtained with computational tests of a set of 20 instances from the literature, and a comparison of these results with those obtained with two other approaches. Concluding remarks are presented in Section 6.

2. The CCP formulation

A CCP mixed-integer formulation very similar to the one given by Mulvey and Beck (1984) will be adopted. Consider a set of customers, $I = \{1,2,\ldots,n\}$, with an $n \times n$ symmetric matrix $[d_{ij}]$ representing the Euclidean distances between pairs of customers. Assume that $d_{ij} > 0$ and $d_{ii} = 0$, $\forall i,j \in I$. A positive weight w_i is assigned to each customer i. Let $J = \{1,2,\ldots,m\}$ denote the set of clusters and W_j their capacities. A median of cluster j is defined as the customer from which the sum of the distances to all other customers in cluster j is minimised. For the sake of simplicity in notation, the capacity of each cluster, W, is assumed to be identical.

The binary variable y_j assumes the value 1 if customer j is assigned as the median for cluster $j \epsilon J$; otherwise, its value is zero. The binary variable x_{ij} indicates whether or not customer i is assigned to cluster j. If it is, this variable equals 1; otherwise, it equals zero.

Constraint set (2) guarantees that only p clusters will be selected. The constraints defined in Eq. (3) ensure that all customers are assigned, while the constraints in (4) prevent assignments to customers that have not been selected as medians. The constraints (5) ensure that the sum of the weights assigned to a cluster does not violate its capacity.

$$F(P) = \mathrm{Min} \sum_{i \in I} \sum_{j \in J} d_{ij} x_{ij} \tag{1}$$

subject to

$$\sum_{j \in J} y_j = p \tag{2}$$

$$\sum_{j \in J} x_{ij} = 1 \quad i \in I \tag{3}$$

$$x_{ij} \leq y_j \quad i \in I,\ j \in J \tag{4}$$

$$\sum_{i \in I} w_i x_{ij} \leq W \quad j \in J \tag{5}$$

$$x_{ij}, y_j \in \{1,0\} \quad i \in I,\ j \in J \tag{6}$$

Many heuristic methods have been proposed to solve the CCP. Mulvey and Beck (1984) present two algorithms. The first proposes a primal heuristic based on the greedy assignment of customers to prespecified seeds. In this approach, a random set of p medians is chosen and customers are assigned to the nearest median such that the cluster capacity constraints (5) are not exceeded. This assignment is made in decreasing order of the regret value, defined as the absolute value of the difference in distance between a customer's first and second nearest median. This strategy guarantees that the possibility of assignment of a customer to a very distant median is minimized. After all assignments, the median of each cluster is re-computed. If new medians are found, the assignment process is repeated. When the final set of medians is found, a pairwise interchange procedure is adopted to improve the solution. A new random set of seeds is then generated, and the entire process is repeated for a predetermined number of steps. These authors also propose a hybrid heuristic-subgradient method where the primal heuristic is embedded within a subgradient algorithm. Computational results of 18 randomly generated problems of up to 100 customers and 20 medians show that the method is capable of finding solution values with a relative deviation from the lower bound of the subgradient ranging from 0.37 to 3.6%.

The method proposed by Koskosidis and Powell (1992) is an improvement on the algorithm of Mulvey and Beck. They suggest various algorithms to find initial solutions based on the Lagrangian relaxation of the assignment constraints (3), which leads to the decomposition of a single CCP problem into a series of knapsack problems. Bramel and Simchi-Levi (1995) also use Lagrangian relaxation to decompose the problem into knapsack problems and then calculate a lower bound through a subgradient procedure. Their problem considers that a fixed cost is incurred for locating a median at site $j, j = 1, \ldots, m$, and do not specify a fixed number of medians to be located. As the solution provided by the subgradient method may be infeasible, they propose a heuristic procedure to find a feasible solution (upper bound). The heuristic is based on the observation that the knapsack solutions can be used for ordering the medians according to the benefit of setting up a median at a site. At each iteration of the subgradient method, a sequence of the optimal solution values found in each knapsack problem is constructed in a non-decreasing order. Then, a bin packing problem which seeks for the minimum possible number of medians is solved using W as bin capacities. A bin packing solution is found by taking the first median of the sequence and assigning the customers in its optimal knapsack to this median. The next median is taken and the process follows until the minimum number of medians is found. If some customers remain unconnected, they are assigned to medians where they fit with minimum additional cost. If the relative error between the best upper bound and the best lower bound is less than a threshold value, the method applies a local improvement heuristic and terminates. Otherwise, a branch-and-bound algorithm is used to reduce the gap.

The method presented by Osman and Christofides (1994) applies a hybrid SA and TS algorithm, which utilizes a three stage procedure to obtain an initial feasible solution. In the first stage, a set of initial seeds is determined such that a good seed spread is guaranteed; the two farthest customers are chosen and assigned as medians; the next median is selected to maximize the product of the distances from this median to the medians already selected. In the second stage, the non-medians customers are assigned to their nearest medians as long as the capacities of the clusters allow this. Whenever an assignment causes a capacity overflow, the corresponding customer is assigned to the immediately available nearest median. Finally, in the third stage, the median of each cluster is re-computed according to the final assignments. If new medians are found, the assignment process is repeated. This initial solution is then improved by a metaheuristic. They propose a hybrid SA/TS scheme which employs a shift and a pairwise interchange neighborhoods. Osman and Christofides report computational results of 20 problems with deviations from the best known solutions in the range of 0.0–2.94%.

Another heuristic approach was developed by Fisher and Jaikumar (1981) to solve the Generalized Assignment Problem (GAP), which is closely related to the CCP, except that the cluster seeds are given and fixed. The method was applied to solve a version of the vehicle routing problem.

3. Initial solution

Tabu search implementations require an initial feasible solution to initiate the search process. Such a feasible solution can be obtained by many algorithms. We chose the iterative three-stage procedure proposed by Osman and Christofides (1994) and described in the Section 2. The procedure may fail to produce a feasible solution, however, the implantation proposed here introduces minor modifications in the first and second stages in order to minimise this occurrence. The improvement in the first stage of Osman and Christofides' algorithm involves the choice of the last median that should be the customer that minimizes the product of the distances between this customer and all the other $p-1$ medians. The original criterion used in the second stage is to assign customers to its nearest medians, with this assignment being made in increasing order of distance. In our second-stage implementation, four different criteria are used: (1) identical to Osman and Christofides' criterion; (2) assign customers as a function of the quotient d_{ij}/w_i, with the non-medians customers being assigned in increasing order of this quotient; (3) regret function is the third way to assign customers to medians, with non-medians customers being ordered in decreasing order of regret; (4) simply uses a sequence of customers arranged in decreasing order of weight.

4. TS implementation

Some of the key features that are responsible for effective TS implementation are the following:

- neighborhood generation mechanism;
- definition of the tabu activation rule;
- tabu tenure, i.e., length of time in which a previous move will be maintained tabu-active.

In this TS approach for the CCP, two neighborhood generation mechanisms are considered. Given that P is the current solution, the first neighborhood $\mathrm{N}_1(P)$, called *pairwise interchange*, consists of all solutions generated by interchanging any two customers that belong to different clusters. The second neighborhood $\mathrm{N}_2(P)$ is called *insertion* or *shift* and consists in inserting any customer that belongs to one cluster into another.

A local search step will be performed when a move to one of the neighbors of $\mathrm{N}_1(P)$ or $\mathrm{N}_2(P)$ yields an improvement in the objective function. For any given pair of clusters, the customers are first searched sequentially and systematically by pairwise interchange; only afterwards does the search look for improving solutions in the insertion neighborhood. The next solution P' can be either the neighbor that results in the best improvement or the one that first causes an improvement. Whenever a movement is performed, the medians of the clusters involved in it have to be recalculated.

The central idea underlying TS is the use of a flexible memory to guide the search process. Once a movement is performed, its reverse is prohibited for a certain number of iterations θ. This means that at a certain iteration, the effective neighborhood used to select the next solution will be a subset of $\mathrm{N}_1(P) \cup \mathrm{N}_2(P)$. Characterized in this way, TS may be viewed as a dynamic neighborhood search. Since the method allows non-improving movements, the prohibition of the reversal of the most recent movements is the way TS prevents cycles with a length less than or equal to $|\theta|$ from occurring in the search trajectory. Actually TS does not store the complete solutions of the previous θ movements, since such a practice would be space and time-consuming. A more elegant and equally effective procedure is to store solution *attributes* that have changed during the recent past. This kind of memory is called *attributive recency-based memory*. Selected attributes that occur in recently visited solutions are labeled *tabu-active*. When a move attempts to reach some solution which contains or exceeds a pre-specified number of *tabu-active* elements, it will be declared tabu, and prohibited to be performed.

Regarding a pairwise interchange movement, the attributes chosen to control the search are the four edges involved when two customers from different clusters are exchanged. To perform a movement observe that two edges have to be deleted and two others added. After every move, four θ numbers are randomly chosen within a tabu tenure range defined by parameters $\theta_{\min}$ and $\theta_{\max}$ and assigned to each edge. These define the number of iterations for which the edges should remain tabu-active. During the next θ iterations, any candidate move comprising one or more of these edges may be subject to restrictions. The status of a move involving one or more of these tabu-active edges is determined by the *tolerance* parameter T_1. This tolerance expresses the tabu activation rule by defining the maximum number of tabu-active edges which can be present in a move. Hence, T_1 can assume the values 0, 1, 2, 3 or 4. If T_1, is set to zero, the search will be limited to moves that do not involve any tabu-active edge, whereas if $T_1 = 4$, any candidate move will be accepted, regardless of the status of the edges involved.

The same concepts are used to manipulate insertion movements. Observe that when a customer is inserted into a cluster, only two edges are involved. In this case, the tabu activation rule depends on the definition of a new tolerance parameter T_2, which can assume the values 0, 1 or 2. The present TS implementation also considers an aspiration criterion, which states that the tabu activation rule is to be overridden if the move yields a solution better than the best obtained so far (incumbent solution).

4.1. Adaptive TS algorithm

This adaptive TS version also incorporates a specific feature which gives the search the ability to set proper values to T_1, and T_2 for intensification and diversification purposes automatically. Such a TS approach has been proposed by Pureza and França (1996), and it has been successfully applied to solve other combinatorial problems. The main goal of the adaptive TS approach is to alter restrictiveness levels in order to intensify the exploration when indicators identify promising regions, and promote diversification, if improvements seem to be minimal. Many recent TS implementations aiming the integration of intensification and diversification strategies have been proposed. The concept of *moving gaps* (Hübscher and Glover, 1992), where the tabu list consists of both a static and a dynamic part, is one of these attempts. Chakrapani and Skorin-Kapov (1993) utilize a similar concept for the quadratic assignment problem by dynamically varying the tabu list size through eight configurations.

The level of restrictiveness of the search can be governed by controlling the values assigned to T_1 and T_2, as well as the range of the tabu tenures. When the search process encounters

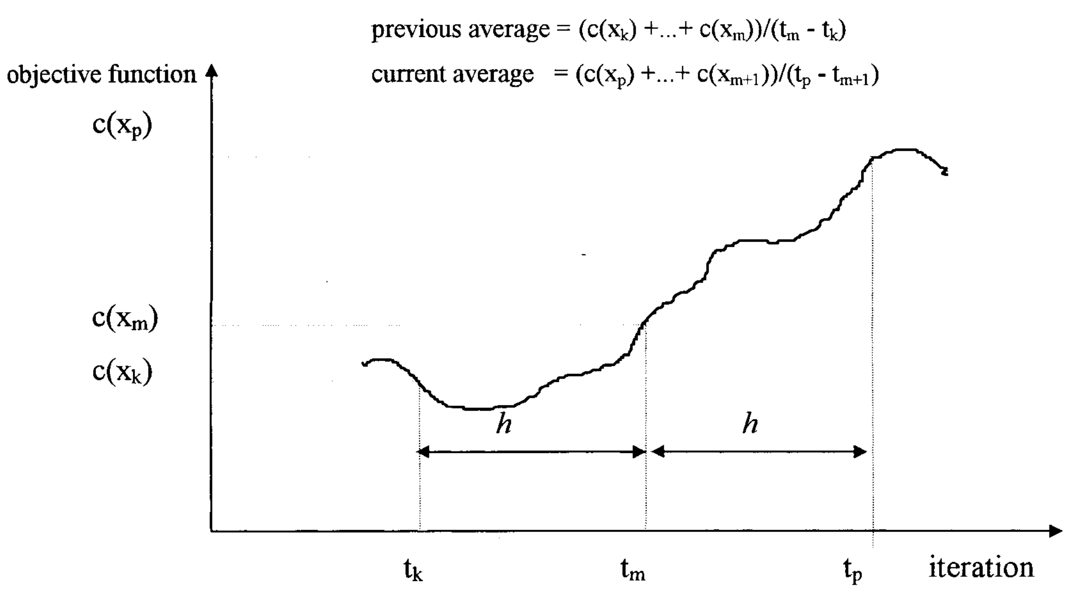

Fig. 1. Identification of trajectories.

high restrictiveness levels ($T_1 = T_2 = 0$, for instance), it is likely to fail in finding high quality solutions because the most attractive moves of the current neighborhood will be forbidden. Under these circumstances, a search diversification should occur. Analogously, the opposite situation — low restrictiveness levels — should promote search intensification.

The control mechanism which determines when to shift from diversification to intensification phases, or vice versa, is based on *trajectory patterns*, i.e., the behavior of the objective function value over the last h iterations. This h parameter indicates the length of the horizon used to compute and evaluate the trajectories. Three different cost behaviors have been investigated: *stagnation, ascent trajectories* and *descent trajectories*. When search stagnation is observed, the adaptive mechanism responds by imposing increasing restrictiveness levels aiming at promoting a diversification phase. Under an ascent trajectory, the method relaxes tabu restrictions to stop diversification. If the solution sequence describes a descent cost trajectory, the adaptive approach establishes mild restrictiveness levels in order to stimulate a more extensive exploration of the current region.

The identification of these three trajectory patterns involves calculating the average of the objective function costs observed during the last two *search stages*, defined as the current stage and previous stage. In both stages, the average costs are computed over h iterations. Fig. 1 illustrates the trajectory identification process. Assume that $c(x_i)$ is the objective function value calculated at iteration t_i.

The relation between the values of the previous and current averages is used as a means of identifying the trajectory pattern. If the average values are approximately the same, we say that a stagnation has taken place, but if the current average is greater than the previous one, the search is describing an ascent trajectory. Since the use of these simple statistics to infer search behaviors faces certain limitations, a suitable choice of h is mandatory to minimize possible misinterpretations.

In addition to its role in search pattern identification, h also defines the period during which the perturbation selected in the previous analysis or evaluation is to be applied. The horizon should be long enough to ensure that improvement possibilities are not overlooked during promising stages and to overcome stagnation. However, excessively long horizons generally result in an excessive increase of solution costs during the application of high restrictiveness levels, which may direct the search to lower quality local optima and delay further improvement.

Preliminary computational experiments suggested the adoption of random dynamic horizons. Hence, for every evaluation, a random integer h is generated within the range $[h_{\min}, h_{\max}]$. Despite the fact that the search process can be controlled not only by the parameters T_1 and T_2 but also by the tabu tenure θ, in this implementation only the parameters T_1 and T_2 are considered, since the search response is much slower to alterations in the θ parameter, when compared to changes in T_1 or T_2. The use of parameter θ for control purposes would require a more complex control system, such as the determination of the period during which a perturbation remains influential on the search process.

The perturbation of T_1 and T_2 is performed at predetermined intervals, so that the iteration of the next evaluation is defined in advance. This iteration is calculated by adding $k * h$ to the current iteration, where k can assume any real value. Actually, the k value is a tuning factor to adjust the period during which a perturbation is applied. A perturbation intended to lead the search to a high restrictiveness level — $T_1 = T_2 = 0$, for instance — requires small k values, generally lower than 0, so that the diversification phase period will be limited. Descent trajectories, on the other hand, impose k values greater than 1 so that the search time lasts long enough to benefit from the promising possibilities of the region. Another important element adopted in this implementation is an aspiration criterion, which keeps the search operating under standard parameter values when an improvement is verified during the last search stage, regardless of the trajectory patterns. In this case k, is also greater than 1. Based on previous computations, standard values of $T_1 = 3$ and $T_2 = 1$ were selected.

4.2. Adaptive TS algorithm steps

Step 1. Let t be the current iteration. Set $t = 0$. From a starting solution, set $T_1 = 3$, $T_2 = 1$ and proceed with the heuristic search until the first local optimum is found. Meanwhile, store the costs of the generated solutions in a list. Until the stopping criterion is not satisfied, repeat the following steps.

Step 2. Make $previous_{mean}$ equal to the mean of the costs stored in the list and reset the list. Draw a random evaluation horizon h within the adopted range $[h_{\min}, h_{\max}]$, and resume the search under the same operating conditions for h iterations. The next evaluation ($next_{eval}$) will occur at iteration $t + h$. Meanwhile, complete the cost list. When $t = next_{eval}$, make $current_{mean}$ equal to the mean of the costs in the list. Select a new h.

Step 3. If the search succeeds in improving the best solution, set $T_1 = 3$, $T_2 = 1$ and maintain the search under these conditions for h iterations ($next_{eval} = t + h$) and go to Step 5. Otherwise, go to Step 4.

Step 4. Compute $diff = (previous_{mean} - current_{mean})/previous_{mean}$.

4.1. If abs($diff$) ≤ 0.01 then the search is stagnated. Apply maximum restrictiveness levels ($T_1 = T_2 = 0$), and maintain the search under these conditions for $h/2$ iterations (($next_{eval} = t + h/2$).

4.2. Otherwise, and if $diff < -0.01$, an ascent trajectory has been identified; in this case, restrictiveness levels are altered as a function of this difference, according to the additional set of rules presented below. Maintain the search under these operational conditions for $2 * h$ iterations ($next_{eval} = t + 2 * h$) if $T_1 = 4$ and $T_2 = 2$, or for h iterations if other T_1 and T_2 values have been used ($next_{eval} = t + h$).

4.3. Otherwise, and if $diff > 0.01$, an descent trajectory has been verified. If $T_1 = 4$ and $T_2 = 2$, set $T_1 = 3$ and $T_2 = 1$ in order to increase restrictiveness, and maintain the search under these operational conditions for h iterations ($next_{eval} = t + h$).

Step 5. When $t = next_{eval}$, go to Step 3.

4.3. Additional rules for ascent trajectories

In Step 4.2, new tolerance values are selected through a set of rules, where each rule comprises a specific range of *diff* values:

1. If $-0.035 \leq diff$, then set $T_1 = 4$ and $T_2 = 1$. The purpose of this choice is to reach a new

valley. High restrictiveness levels reduce the chances to return to the previous valley, which would imply null restrictiveness.
2. If $-0.035 \leq diff \leq -0.025$ then set $T_1 = 3$ and $T_2 = 1$. When less diversification is verified, higher restrictiveness levels should be applied to prevent the return to the valley from which the search has emerged, while exploring the possibilities of the current region.
3. If $-0.025 < diff \leq -0.015$, then set $T_1 = 2$ and $T_2 = 0$.
4. If $-0.015 < diff \leq -0.01$, then set $T_1 = 1$ and $T_2 = 0$.

5. Computational experiments

The effectiveness of the adaptive TS approach in solving the CCP has been tested with 20 problems found in the literature (Beasley, 1990). Two sets of problems with $n \in \{50,100\}$ and $p \in \{5,10\}$ were used. Each set contains ten different instances of size 50×5 and ten of 100×10. It is assumed that the customers are located on the plane and that their coordinates are randomly generated from a uniform distribution in the range [1, 100]. The Euclidean distances d_{ij} are rounded down to the nearest integer. The weights w_i are generated from the uniform distribution [1, 20]. The generation of cluster capacity is controlled by the *tightness factor* defined as

$$\tau = \sum_{i \in I} \frac{w_i}{m.W}$$

The tightness values adopted in the experiments vary in the range [0.82, 0.96], which are the same used by Osman and Christofides.

Two strategies for selecting a new neighbor were tested: the *best-improve* strategy, which examines all solutions within the neighborhood and selects the one which results in the best objective function improvement, and the *first-improve* strategy, which accepts the first solution that improves the objective function. Results have shown that in general the *best-improve* strategy performs better when $n = 50$, whereas the *first-improve* yields the best results when $n = 100$.

Regarding the constructive heuristics proposed to find an initial feasible solution, all foul heuristics succeed in achieving feasible solutions. The only exception is the heuristic which uses the first criterion that assigns customers to the nearest medians, which failed in one instance of the problem set. The experiments indicated that the best choice is the heuristic using the quotient d_{ij}/w_i as the assignment criterion, followed by the one using the regret function criterion.

A well known result reported by various authors (Glover and Laguna, 1997) is that the number of iterations θ for which a move is declared tabu-active should not be a constant, but rather a random variable uniformly distributed in an interval. Computational experiments showed that the interval $[n/4, n/2]$ provided the best results. The values adopted for the interval $[h_{\max}, h_{\max}]$ were [5, 10] and [10, 20] for problems with $n = 50$ and $n = 100$, respectively. The stopping criterion adopted was the number of iterations during which no improvement in the incumbent solution is observed. It was set 2000 iterations for problems with 50 customers and

5000 iterations for problems with 100 customers. Tests were performed on a SUN Sparc20 workstation using the C language to code the algorithm.

In the first computational experiment, we compare the performance of the proposed method relative to the optimal solutions obtained by a branch and bound procedure (Maniezzo et al., 1998). Table 1 shows the results obtained in different stages of the solution process. The second column shows the values of the tightness factor τ. The third column shows the relative percent deviation from the optimal solution attained by the local search stage (LS). The next column shows the deviations obtained by the tabu search implementation without the adaptive mechanism (TS) followed by the results attained by the adaptive TS algorithm (ATS). In all TS implementations, the initial feasible solution was obtained by the heuristic that uses the quotient d_{ij}/w_i as the assignment criterion. The final column shows the deviations obtained by the previous adaptive TS algorithm when the search is re-started from a new initial solution obtained now by the heuristic that uses the regret function as the assignment criterion (ATS$^+$).

The results presented in Table 1 show that the simple TS implementation improves considerably over the solutions obtained by local search, leading to an average cost reduction of 52%. The cost reduction observed by the adaptive TS algorithm compared to its non-adaptive version averages 98.6%. The adaptive TS implementation with a re-start feature was capable of achieving optimal solutions for all but one of the 20 test problems.

Table 1
Comparisons to optimal solutions

Problem	τ	LS	TS	ATS	ATS$^+$
1	0.82	5.18	0	0	0
2	0.84	0	0	0	0
3	0.85	3.72	3.20	0	0
4	0.86	5.22	0	0	0
5	0.90	4.75	0.30	0	0
6	0.92	8.89	0.26	0	0
7	0.92	2.43	2.92	0	0
8	0.92	0.41	0.61	0	0
9	0.93	1.93	5.45	0	0
10	0.96	12.80	0.97	0	0
11	0.85	7.05	0.70	0	0
12	0.85	8.69	0.41	0	0
13	0.86	10.33	10.33	0	0
14	0.88	2.64	4.68	0.30	0
15	0.88	12.64	7.97	0.27	0
16	0.88	18.97	4.93	0	0
17	0.89	7.35	1.35	0	0
18	0.89	5.36	6.04	0.19	0
19	0.90	2.61	3.49	0.19	0.09
20	0.94	7.01	11.94	0	0
Average		6.89	3.27	0.047	0.004

Our second computational experiment compares the proposed metaheuristic approach with two others also proposed to solve the CCP: the hybrid simulated annealing/tabu search algorithm (OC) with the non-monotonic cooling schedule presented by Osman and Christofides (1994), and the simulated annealing approach (CO) proposed by Connolly (1990), (1992). Table 2 shows the comparisons in terms of the relative percent deviation (RPD) from the optimal solutions and also the objective function values attained by each algorithm ($F(P)$), with F_{opt} representing the objective function values of the optimal solutions. The results from Connolly's algorithm were extracted from Ref. (Osman and Christofides, 1994). The analysis of the results in Table 2 shows that our method outperformed the two other metaheuristics.

Table 3 shows the average CPU time in seconds to run the 10 instances with 50 customers and 100 customers, respectively. It is worth mentioning that the results due to Connolly and Osman/Christofides was obtained with a VAX 8600 and their algorithms were coded in FORTRAN 77. From Tables 2 and 3, it can be seen that the ATS algorithm performs slightly better than the OC algorithm at a savings of almost 50% in CPU time. The ATS$^+$ implementation ranks first when compared to all other methods, although at the expense of more computational effort.

Table 2
Comparisons with other metaheuristics

Problem	F_{opt}	OC		CO		ATS		ATS$^+$	
		$F(P)$	RPD	$F(P)$	RPD	$F(P)$	RPD	$F(P)$	RPD
1	713	713	0	734	2.95	713	0	713	0
2	740	740	0	740	0	740	0	740	0
3	751	751	0	751	0	751	0	751	0
4	651	651	0	651	0	651	0	651	0
5	664	664	0	664	0	664	0	664	0
6	778	778	0	778	0	778	0	778	0
7	787	787	0	805	2.29	787	0	787	0
8	820	820	0	820	0	820	0	820	0
9	715	715	0	715	0	715	0	715	0
10	829	829	0	829	0	829	0	829	0
11	1006	1006	0	1006	0	1006	0	1006	0
12	966	966	0	966	0	966	0	966	0
13	1026	1026	0	1026	0	1026	0	1026	0
14	982	985	0.30	982	0	985	0.30	982	0
15	1091	1091	0	1091	0	1094	0.27	1091	0
16	954	954	0	954	0	954	0	954	0
17	1034	1039	0.48	1037	0.29	1034	0	1034	0
18	1043	1045	0.19	1045	0.19	1045	0.19	1043	0
19	1031	1031	0	1032	0.10	1033	0.19	1032	0.09
20	1005	1005	0	1019	1.39	1005	0	1005	0
Average			0.049		0.360		0.047		0.004

Table 3
Average CPU time comparisons (in seconds)

n	CO	OC	ATS	ATS$^+$
50	20.33	23.23	13.89	41.44
100	373.29	338.19	304.67	500.46

6. Concluding remarks

In this article a new adaptive tabu search-based heuristic approach for solving the CCP was presented. Four constructive heuristics based on weight and/or distance criteria were proposed for ascertaining initial feasible solutions. A local search scheme makes use of two neighborhood generation mechanisms (pairwise interchange and insertion) for the selection of a new improving solution. Two different versions of a basic tabu search algorithm incorporating an adaptive mechanism for systematically perturbing selected tabu elements were developed. The main idea underlying this adaptive TS approach is controlling the restrictiveness of the search in order to promote intensification of the search when some indicators identify promising regions, and diversification if improvements seem to be minimal. The main advantage of this adaptive TS approach is its ability to attain high quality solutions using simple neighborhood structures.

Computational tests of 20 problems obtained from the literature of sizes varying from 50 to 100 customers have shown that the new method out-performs two other metaheuristic approaches recently proposed for solving the CCP. When compared to optimal solutions, the new approach was able to attain the optimum for all but one of the problems.

Acknowledgements

This work was partially supported by the Conselho Nacional de Desenvolvimento Cientifíco e Tecnológico, CNPq, and Fundação Aperfeiçoamento de Pessoal de Nível Superior, CAPES. The authors are grateful to the valuable suggestions of an anonymous referee.

References

Aarts, E.H.L., Lenstra, J.K., 1997. Local Search in Combinatorial Optimization. Wiley, Chichester.

Beasley, J.E., 1990. OR-library: distributing test problems by electronic mail. Journal of the Operational Research Society 41 (11), 1069–1072.

Bourjolly, J.M., Laporte, G., Rousseau, J.M., 1981. Political districting by computer application to the island of Montreal. Infor. 19 (2), 113–124.

Bramel, J., Simchi-Levi, D., 1995. A location based heuristic for general routing problem. Operations Research 43(4), 649–660.

Cerny, V., 1985. A thermodynamical approach to the travelling salesman problem: an efficient simulated annealing algorithm. Journal of Optimization Theory and Applications 45, 41–51.

Chakrapani, J., Skorin-Kapov, J., 1993. Massively parallel tabu search for the quadratic assignment problem. Annals of Operations Research 41, 327–341.

Connolly, D., 1990. An improved annealing scheme for the QAP. European Journal of Operational Research 46, 93–100.

Connolly, D., 1992. General purpose simulated annealing. Journal of the Operational Research Society 43, 494–505.

Everitt, B.S., 1993. Cluster Analysis. Wiley, New York.

Feo, T.A., Resende, M.G.C., 1989. A probabilistic heuristic for a computationally difficult set covering problem. Operations Research Letters 8, 67–71.

Fisher, M.L., Jaikumar, R., 1981. A generalized assignment heuristic for the vehicle routing problem. Networks 11(2), 109–124.

França, P.M., Gendreau, M., Laporte, G., Müller, F.M., 1995. The *m*-traveling salesman problem with minmax objective. Transportation Science 29 (3), 267–275.

França, P.M., Gendreau, M., Laporte, G., Müller, F.M., 1996. A tabu search heuristic for the multiprocessor scheduling problem with sequence dependent setup times. International Journal of Production Economics 43, 79–89.

Garey, M.R., Johnson, D.S., 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco.

Gendreau, M., Hertz, A., Laporte, G., 1994. A tabu search heuristic for the vehicle routing problem. Management Science 40, 1276–1290.

Glover, F., Kochenberger, G., Alidaee, B., Amini, M., 1999. Tabu search with critical event memory: an enhanced application for binary quadratic programs. In: Voss, S., Martello, S., Osman, I.H., Roucairol, C. (Eds.), Metaheuristics: Advances and Trends in Local Search Paradigms for Optimization. Kluwer Academic Publishers, Boston, pp. 93–110.

Glover, F., Laguna, M., 1997. Tabu Search. Kluwer Academic Publishers, Boston.

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor.

Hübscher, R., Glover, F., 1992. Applying tabu search with influential diversification to multiprocessor scheduling, private communication.

Jaw, J.-J., Odoni, A.R., Psaraftis, H.N., Wilson, N.H.M., 1986. A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. Transportation Research B 20, 243–257.

Kaufman, L., Rousseeuw, P., 1990. Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York.

Koskosidis, Y.A., Powell, W.B., 1992. Clustering algorithms for consolidation of customer orders into vehicle shipments. Transportation Research B 26, 365–379.

Kirkpatrick, S., Gelatt, C.D., Vecchi, P.M., 1983. Optimization by simulated annealing. Science 220, 671–680.

Laporte, G., Osman, I.H., 1994. Metaheuristics: a bibliography. In: Laporte, G., Osman, I.H. (Eds.), Metaheuristic in Combinatorial Optimization, Annals of Operations Research 63.

Maniezzo, V., Mingozzi, A., Baldacci, R., 1998. A bionomic approach to the capacitated *p*-median problem. Journal of Heuristics 4 (3), 263–280.

Mulvey, J.M., Beck, M.P., 1984. Solving capacitated clustering problems. European Journal of Operational Research 18, 339–348.

Osman, I.H., Christofides, N., 1994. Capacitated clustering problems by hybrid simulated annealing and tabu search. International Transactions in Operational Research 1 (3), 317–336.

Pureza, V.M., França, P.M., 1996. An adaptive tabu search approach based on the topology of the solution space. In: Proceedings of the II ALIO/EURO Workshop on Practical Combinatorial Optimization, Valparaiso, Chile, 233–249.