Pergamon

Location Science, Vol. 4, No. 1/2, pp. 1–19, 1996

0966-8349/96 $15.00 + 0.00

PII:S0966-8349(96)00008-3

LOCATING A TRANSIT LINE USING TABU SEARCH

HÉLÈNE DUFOURD*

Caisse de dépôt et placement du Québec, 1981 avenue McGill College, Montréal, Canada H3A 3C7

MICHEL GENDREAU

Département d'informatique et de recherche opérationnelle, Université de Montréal, C.P. 6128, succursale Centre-Ville, Montréal, Canada H3C 3J7

and

GILBERT LAPORTE*

École des Hautes Études Commerciales, 5255 avenue Decelles, Montréal, Canada H3T 1V6

(*Received for publication January* 1996)

Abstract—This article addresses the problem of locating a metro or rapid transit line. An alignment comprising n stations must be located in a territory, subject to minimum and maximum station interspacings. The objective is to maximize the total population covered by the alignment. A tabu search heuristic is developed and computational tests on randomly generated instances are presented. Sensitivity analyses are performed on a number of parameters used in the algorithm. Copyright © 1996 Elsevier Science Ltd

Keywords: Rapid transit line, metro, alignment, location, tabu search.

1. INTRODUCTION

Faced with urban sprawl and growing traffic congestion, several cities throughout the world have turned to metro, rapid rail transit (RRT) and light rail transit (LRT) systems. The number of metro networks now stands at around ninety (Gendreau *et al.*, 1995) while the number of RRTs and LRTs is growing at a frenetic rate (Schumann, 1992). For a taxonomy and classification of rapid transit systems, see Jiménez Solano (1993). For a comparison of different modes, see Stone *et al.* (1992). Rapid transit construction requires long term commitments and sizeable capital investments. What constitutes a good configuration is by no means obvious. Planners, engineers, users, environmentalists, and other interest groups do not, as a rule, agree on a common set of objectives and constraints. Data are often unreliable, uncertainties and cost overruns are common (Bonz, 1983), and non-quantifiable criteria or external attributes come into play, such as land use disruptions (Perrin and Benz, 1990), air and noise pollution (Blackledge and Humphreys, 1984), safety (Siegel, 1980; Straus, 1980), impact on real-estate value and on retail trade (Bay, 1985; Wulkan and Henry, 1985), impact on parking (Schabas, 1988), etc.

The core problem, that of designing a transit network capable of transporting a large number of people efficiently and effectively, is highly complex. Some authors have proposed

*Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale Centre-Ville, Montréal, Canada H3C 3J7.

various indices to assess the topological configurations of rapid transit systems (see, e.g. Musso and Vuchic, 1988; Laporte *et al.*, 1994) but such indices are more easily used as a way of measuring the quality of proposed or existing networks than as a means of generating good solutions. In operations research, the problem belongs to the class of network design problems, but due to their usually very large sizes, integrality restrictions and non-linearities, they can rarely be solved by means of standard network design methods. In fact, Magnanti and Wong (1984) point out the limitations of integer programming models for solving strategic transportation decision problems such as designing highways, airports and mass transit networks.

Rather than advocating the use of a large-scale mathematical model that would, in all likelihood, defy any known solution technique, we propose a practical and versatile heuristic approach for a simplified version of the problem, that of locating a single transit line, or determining an *alignment* to maximize the total population covered. The approach is based on tabu search, a technique that has proved highly appropriate for the solution of large-scale ill-structured optimization problems. Before discussing the details of our model and algorithm, we point out that our work is not restricted to the design of rapid transit lines. It applies equally well, for example, to the location of bus lines or expressways (see Current *et al.*, 1985, 1986). The proposed method can easily generate a number of good solutions within reasonable computing times. These solutions can then be assessed according to a variety of criteria and it is then up to the decision makers to choose among a set of interesting alternatives. In other words, what we propose is a tool to support a very complex planning process, rather than a substitute for human decision making. Multricriteria methods such as those described in Gomes (1989) and in Vincke (1992) should constitute a useful complement to the proposed approach.

The remainder of this paper is organized as follows. In the next section, we discuss the objective and constraints of our model, and we propose a grid structure for the representation and generation of feasible solutions. The tabu search algorithm is described in Section 3, followed by computational results in Section 4 and conclusions in Section 5.

2. THE MODEL

The primary objective when locating a transit line is to improve the population's mobility by providing shorter travel times (see, e.g. Blackledge and Humphreys, 1984; Bonz, 1983; Fukuyama, 1981). This is usually accomplished by maximizing the population covered by the line, but there is no general agreement on how this objective should be measured. For some authors (e.g. Chapleau *et al.*, 1986; Wirasinghe and Vandebona, 1987), this is interpreted as maximizing the catchment area or zone of influence of the line, i.e. the number of people living or working within a corridor around the line. This approach has some drawbacks as people who live near a transit line, but far from a station, are less likely to use it. Actual walking distances to a station seem to provide a better indication of the probability of using rapid transit. Thus the total population in the stations' catchment areas would seem to be a better measure of ridership. This is the objective used in several covering-location models (Gleason, 1975; Current *et al.*, 1982; Gendreau *et al.*, 1996), and it is also employed by transit planners (Halder and Majumder, 1981; Quqing, 1984; Lutin and Benz, 1992). There appear, however, to be two problems with this measure. The first is that population relocations invariably occur after the line has been built, a phenomenon noted by Wulkan and Henry (1985). The second problem is that what really counts is not so much the population living near the line, but the number of trips covered by it (Cohon, 1971; Huber and Church, 1985). The difficulty here is that estimating all O/D trips is rather costly and complex, and this type of information is hard to incorporate into an algorithm. Further, this measure does not do away with the relocation problem just mentioned. In any case, the population covered by all stations of a line should be strongly related with ridership, certainly if it includes not just people living near stations, but also people working in their vicinity. As a result, we have opted for this objective in our model.

The problem of locating a line providing good population coverage belongs to the family of covering-path problems, studied, for example, by Current *et al.* (1982, 1985, 1988), Current and Schilling (1994), Current *et al.* (1994), and Gendreau *et al.* (1996). However, what we are solving is not a pure covering path location problem as two side constraints are present: (1) the number of stations is given and equal to n (we used $n=15$ in our implementation); (2) the interstation spacing must lie between a lower bound $l_{\min}$ and an upper bound $l_{\max}$. We use the Manhattan distance metric to compute interstation spacing as this most accurately measures pedestrian walking distances in grid cities. Other metrics could, of course, be used where the situation warrants. The upper limit is imposed to ensure an adequate density of stations per unit distance; the lower limit helps provide adequate travel speeds and indirectly reduces construction costs by limiting the number of stations per mile. Note that these constraints coupled with the population coverage objective should favour the design of a high revenue/low cost line, even if construction and operating costs are not explicitly included in the objective (our method allows, however, for the incorporation of additional criteria).

As do other authors in the field (Dicesare, 1970; and Church and Clifford, 1979), we represent a city by a grid with a discretized popultion at the vertices. In our implementation, we use square grids of size 100×100. A transit line is defined by a shortest path through all stations, also located at vertices. The population cover $R(s)$ of station s is obtained by assigning non-increasing weights $\theta_d(d=0,\ldots,\bar{d})$ to the population living d units away from s using the Manhattan metric. We use $\bar{d}=3$, $\theta_0=1$, $\theta_1=1$, $\theta_2=0.5$ and $\theta_3=0.25$. Formally, the cover of s is defined as

$$R(s)=\sum_{d=0}^{\bar{d}} \sum_{(x,y):D[(x,y),s]=d} \theta_d p(x,y), \tag{1}$$

where $D[(x,y),s]$ is the Manhattan distance between (x,y) and s, and $p(x,y)$ is the population at (x,y). Graphically, $R(s)$ is obtained by adding the weighted populations of several concentric diamonds around station s (see Fig. 1). Since we use $l_{\min}=8$ and $l_{\max}=16$ in our implementation, all stations have disjoint catchment areas.

Formally, the problem can be stated as a non-linear integer mathematical program as follows. Let (x_s,y_s) denote the coordinates of station s, and let S be the set of candidate locations for the stations. Then the problem is to

$$\text{maximize} \sum_{s=1}^{n} R(s) \tag{2}$$

subject to

$$|x_s-x_{s+1}|+|y_s-y_{s+1}|\leqslant l_{\max} \quad (s=1,\ldots,n-1) \tag{3}$$

$$|x_s-x_{s'}|+|y_s-y_{s'}|\geqslant l_{\min} \quad (s,s'=1,\ldots,n;s\neq s') \tag{4}$$

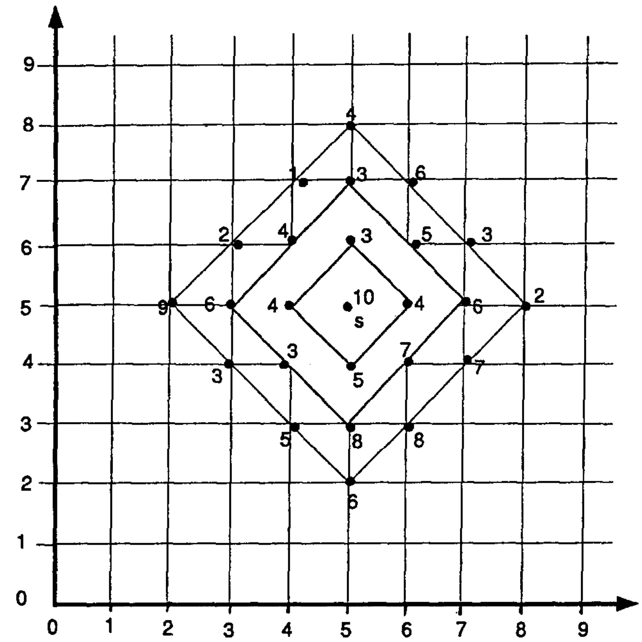

Fig. 1. Station s at (5,5) and its catchment area. Numbers represent vertex populations. $R(s)=10\theta_0+16\theta_1+42\theta_2+56\theta_3=61$.

$$(x_s,y_s)\in S \quad (s=1,\ldots,n). \tag{5}$$

Because of the nature and size of this model, we have opted for a heuristic approach.

3. A TABU SEARCH ALGORITHM

The scientific literature on the location of transit lines is very scarce. Some authors (Dicesare, 1970; Church and Clifford, 1979) have suggested an algorithm for locating linear structures such as pipelines, electricity transmission lines or transit lines. The method applies to multicriteria cost minimization problems. It first discretizes the region under study by superimposing a grid on it. Each cell is then assigned a score relative to each criterion. These scores are aggregated and a shortest path linking an origin and a destination is then computed. This method does not translate well to a population coverage objective as it would typically produce a line crisscrossing the entire region. Methods developed in the context of covering path problems (Current *et al.*, 1985, 1988; Current, 1988; Current and Schilling, 1994) are more suited to our problem, but they do not explicitly allow for side constraints. Further, these methods do not perform an extensive search of the solution space and could easily become trapped in a local optimum.

The methodology we propose is based on tabu search, a metaheuristic that provides a framework within which lower level heuristics must be implemented. The modern version of this method is rooted in the work of Glover (1986) and of Hansen (1986). Briefly, tabu search explores the solution space by moving at each iteration from a solution to the solution with the best value of the objective function in its neighbourhood, even if this causes the objective to deteriorate. To avoid cycling, solutions that were recently examined are declared forbidden or "tabu" for a number of iterations. It is customary not to store the actual tabu solutions, but only some of their attributes. The tabu status of a solution may be lifted it is corresponds to a new best solution. Several features such as diversification and intensification of the search process are now commonplace. For recent expository papers on tabu search, see Glover (1989, 1990), Glover and Laguna (1993), Pirlot (1992), and Hertz *et al.* (1996).

Applying tabu search to a particular problem requires embedding into the algorithm a fair amount of problem-specific knowledge. We now describe how a tabu search was applied to our problem. Before proceeding to a step by step description of the algorithm, we outline its main features.

3.1. *Solution representation*

At any iteration t of the algorithm, an alignment is represented by an ordered sequence $X^t=(s_1^t,s_2^t,\ldots,s_n^t)$, where s_i^t is the coordinate vector of station s_i. This solution must satisfy

$$D(s_i^t,s_{i+1}^t)\leqslant l_{\max} \quad (i=1,\ldots,n-1)$$

and

$$D(s_i^t,s_j^t)\geqslant l_{\min} \quad (i,j=1,\ldots,n;i\neq j). \tag{6}$$

The total cover of X^t is simply

$$z(X^t)=\sum_{i=1}^{n} R(s_i^t), \tag{7}$$

where $R(s_i^t)$ is defined by (1).

3.2. *Initial solutions*

The tabu search process is initiated from a solution obtained by performing a random walk along one of the two diagonals of the square grid. More specifically, a walk along the main diagonal consists of locating a station at the top left corner of the grid and moving one unit right or down at each step, with probability 1/2, as long as the edge of the grid is not reached; in such a case, the remaining portion of the walk follows the edge. A new station is located whenever 12 or 13 units of distance have been traveled (recall that we always locate 15 stations on a 100×100 grid). The inter-station distance (12 or 13) is selected randomly, using a probability of 1/2. A similar process is used to generate an initial alignment along the second diagonal (see Fig. 2). One advantage of this process is that it enables the generation of several feasible solutions for a given grid.

3.3. *Neighbourhood structure*

The neighbourhood of a solution X^t is another feasible solution obtained by moving one of the stations, say $s_i^t=(x,y)$ to one of the points $(x-1,y)$, $(x+1,y)$, $(x,y-1)$, $(x,y+1)$, as long as these points do not fall outside the grid boundary. Thus between two and four neighbours can be reached by moving any station belonging to a solution. For s_j to be an admissible neighbour of s_i^t, it must also satisfy

$$D[s_j,s_{i+1}]\leqslant l_{\max} \quad (\text{if } i<n) \tag{8}$$

$$D[s_j,s_{i-1}]\leqslant l_{\max} \quad (\text{if } i>1)$$

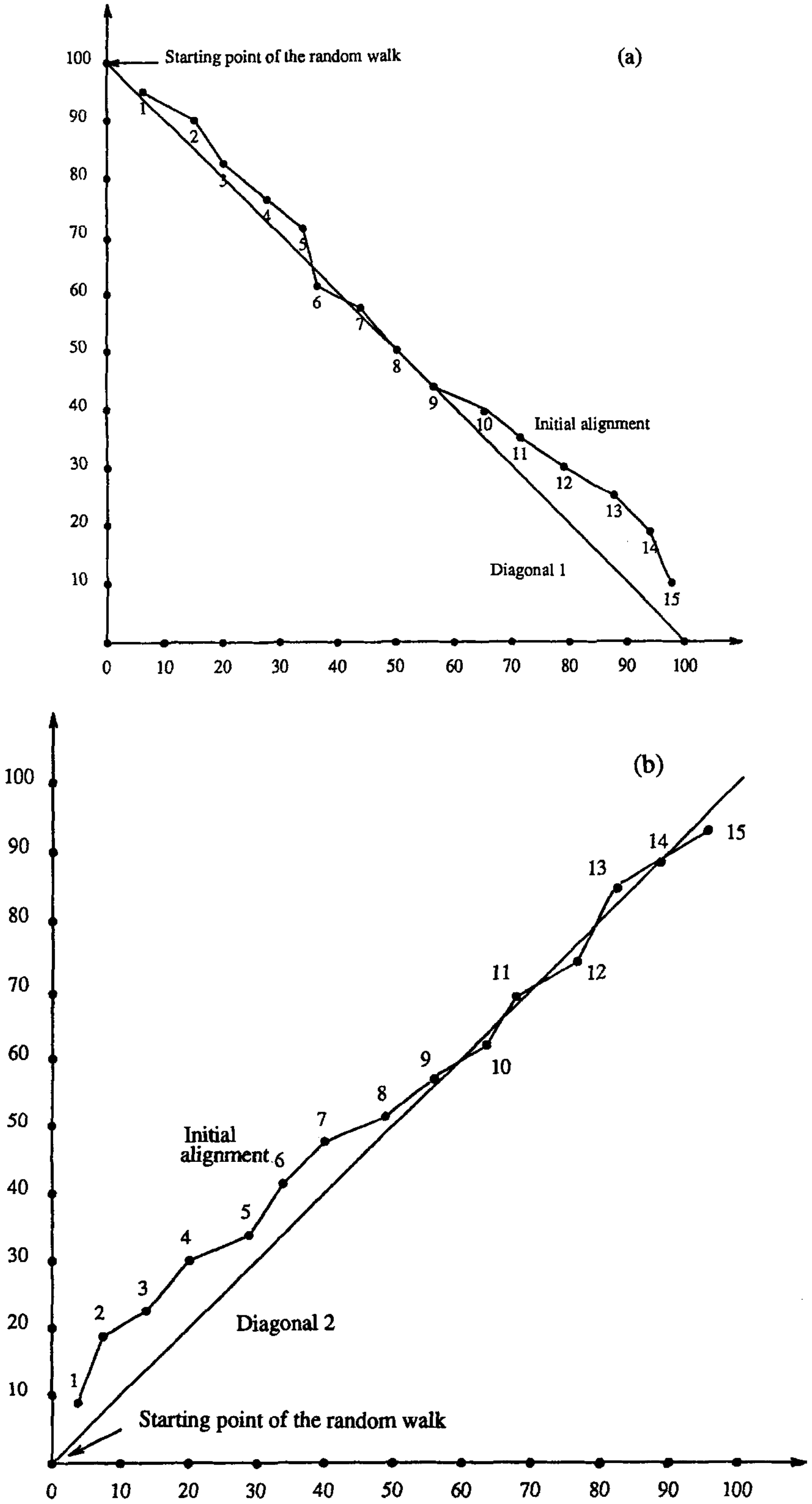

Fig. 2. (a) Initial alignment along the main diagonal. (b) Initial alignment along the second diagonal.

and

$$D[s_j,s_k]\geqslant l_{\min} \quad (k\in\{1,\ldots,n\}\backslash\{i\}).$$

The total cover of a solution X_{ij} obtained from X' by moving s_i' to one of its neighbours s_j is easily computed as

$$z(X_{ij})=z(X')-R(s_i')+R(s_j). \tag{9}$$

3.4. *Moves and tabu status*

Initially, no move is tabu. At a general iteration, the algorithm executes the best non-tabu move to a feasible neighbour of the current solution, but if a tabu move yields a better incumbent, it is implemented. Whenever a move is performed, the reverse move is declared tabu for m iterations, where m is randomly generated in an interval $[m_{\min},m_{\max}]$, according to a discrete uniform distribution, and $m_{\min}$, $m_{\max}$ are parameters of the algorithm. The concept of variable length tabu status was first introduced by Taillard (1991) in the context of the Quadratic Assignment Problem and was found to help avoid cycling. It has since been used successfully in a variety of other applications (Gendreau *et al.*, 1994 and Benati and Laporte, 1994).

3.5. *False starts*

To help prevent the process from starting on a wrong track, it is useful to carry out a limited search starting from several initial solutions before entering the core of the procedure. This strategy has also been applied successfully to other contexts (Gendreau *et al.*, 1994). In our implementations, we generate 30 alignments and select the best. To determine the 30 starting solutions, we proceed as follows: first, 60 initial alignments are generated as described in Section 3.2, 30 along each diagonal. Tabu search is performed from each initial alignment for 100 iterations, yielding 60 local optima. The 30 best local optima are then selected as a starting point for the entire procedure. The remaining 30 local optima are discarded. Figure 3 shows the alignment produced on a randomly generated grid after 100 iterations. (Note that the grids represented in Figs 3–5 are different from those used for the test problems in Section 4.)

3.6. *Diversification and intensification*

The operations described in Section 3.4 are applied iteratively, from a starting solution, until the objective has not improved for $\bar{t}$ consecutive iterations, where $\bar{t}$ is an input parameter. This defines the "Neighbourhood Search" procedure of the algorithm. Preliminary tests have shown, however, that applying Neighbourhood Search alone is not always sufficient to converge to a good solution. In the course of designing the algorithm, we observed that various starting solutions often produced final solutions that widely differed in quality. In other words, the algorithm often became trapped in a local maximum. To counter this we have created a second procedure called "Shake Up" using as a starting point a perturbation of the best known solution. This phase is inspired by two concepts commonly used in tabu search: diversification and intensification. Diversification consists of widening the search for better solutions by moving away from the current solution or the current best solution, sometimes by proceeding to a randomly selected solution. Intensification is the reverse operation: it consists of accentuating the search effort around a good solution. Our

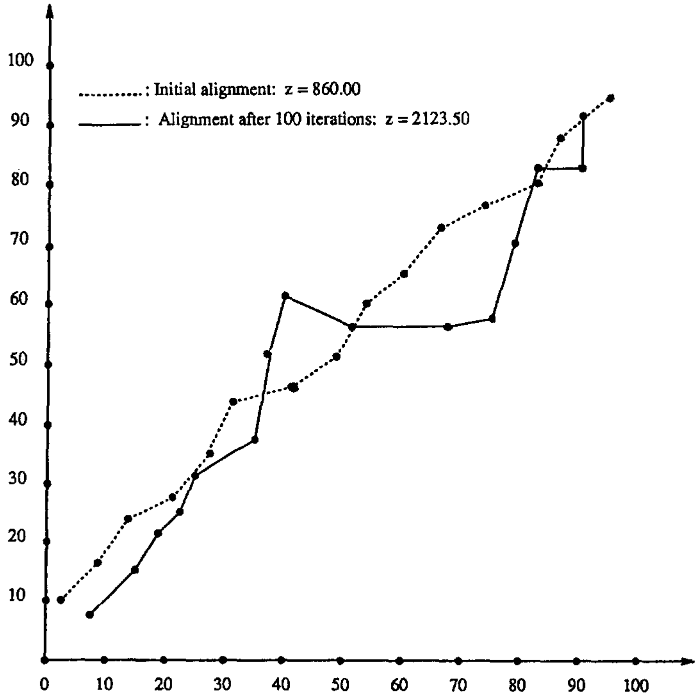

Fig. 3. Alignment obtained after 100 iterations.

Shake Up procedure combines intensification and diversification: it starts from the best known alignment and applies to it a major perturbation by allowing c stations ($c \geqslant 1$) to move up to e units away from their current location, as long as the bounds on inter-station spacing are respected. This procedure is never applied more than once during a given operation. We used $e=8$ as larger values of e tend to move adjacent stations too far apart (i.e. the $l_{\max}$ limit is exceeded). The Shake Up procedure is illustrated in Fig. 4. Tabu moves have to be adapted to this situation. Instead of preventing a station from going back to its former location for a number of iterations, the tabu status now applies to *directions*: a station cannot move back *towards* its former location. For example, a station that is moved in the northeast direction cannot be moved in the south or west direction for a number of iterations.

In practice, the Shake Up procedure is normally entered whenever Neighbourhood Search has produced at least one improvement in the objective function, but no improvement for $\bar{t}$ consecutive iterations. However, Shake Up is also applied after the first application of Neighbourhood Search, even if no improvement has been recorded.

3.7. *Summary of the algorithm*

Before describing the tabu search algorithm, we first explain its two main procedures: Neighbourhood Search and Shake Up.

3.7.1. Neighbourhood Search. At iteration t, let $X^t=(s_1^t,s_2^t,\ldots,s_n^t)$ be a feasible solution of value $z(X^t)$. Let $N(X^t)$ be the set of feasible neighbours of X^t, as defined in Section 3.3. The best neighbour of X^t is a solution $X_{i^*j^*} \in N(X^t)$ obtained by moving $s^t_{i^*}$ to its best neighbour s_{j^*}. Similarly define the best feasible non-tabu neighbour of X^t as $X_{\bar{i}\bar{j}} \in N(X^t)$. ($X_{i^*j^*}$ and $X_{\bar{i}\bar{j}}$ may coincide). Let X^* be the incumbent (the best known feasible solution) and let $z(X^*)$ be its value.

If $z(X_{i^*j^*}) > z(X^*)$, set $X^* := X^{t+1} := X_{i^*j^*}$ and $z(X^*) := z(X^{t+1}) := z(X_{i^*j^*})$. Declare the move of a station from s_{j^*} to s_{i^*} tabu for m iterations, where $m \in [m_{\min}, m_{\max}]$. If $z(X_{i^*j^*}) \leqslant z(X^*)$ and all moves defining the solutions of $N(X^t)$ are tabu, set $\delta := 1$ and return. Otherwise, set $X^{t+1} := X_{\bar{i}\bar{j}}$ and $z(X^{t+1}) := z(X_{\bar{i}\bar{j}})$. Declare the move of a station from $s_{\bar{j}}$ to $s_{\bar{i}}$ tabu for m iterations, where $m \in [m_{\min}, m_{\max}]$.

3.7.2. Shake Up. Proceed as in Neighbourhood Search but define $N(X^t)$ differently, this time by allowing c stations ($c \geqslant 1$) to move up to e units away from their current location. When a station is moved, moving it back in the direction of its former location is declared tabu for m iterations, where $m \in [m_{\min}, m_{\max}]$.

3.7.3. Tabu search algorithm.
Step 1. Let k_1 be the number of initial solutions first considered. For $l = 1, \ldots, k_1$, generate an

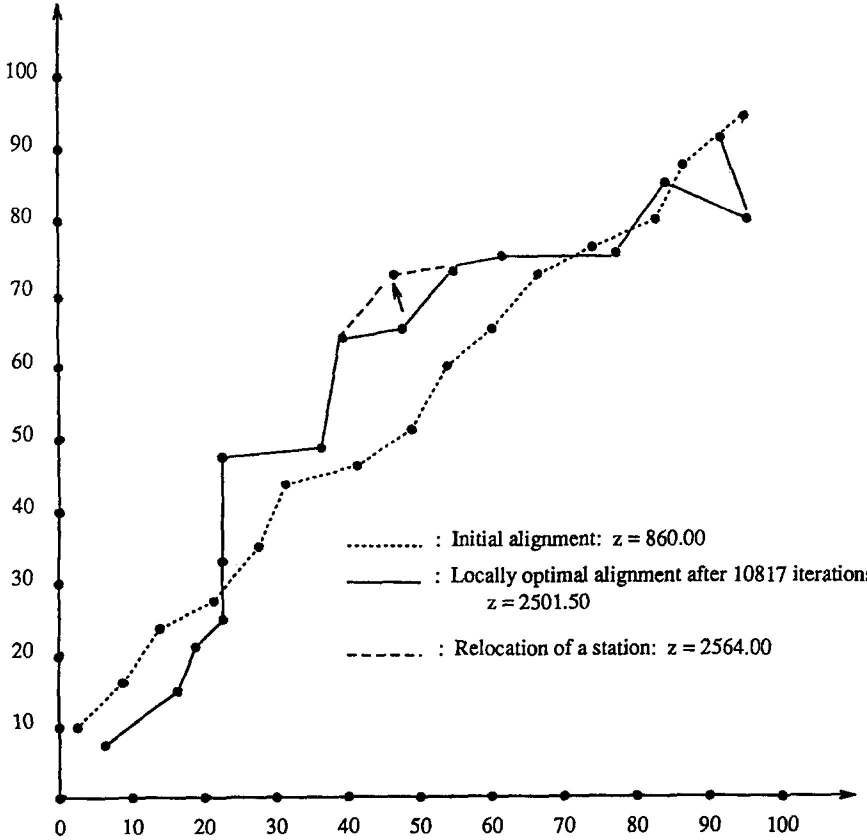

Fig. 4. Illustration of the Shake Up procedure.

initial solution by performing a random walk along one of the two diagonals of the grid. Apply the Neighbourhood Search procedure for 100 iterations. Let X_l^* be the best solution. Then select the $k_2(\leqslant k_1)$ best solutions X_l^* generated in this fashion as new starting points. Renumber these solutions from 1 to k_2. Execute Steps 2 and 3 for $l=1,\ldots,k_2$ and then stop

Step 2. Set $\delta:=0$, $t:=1$ and $X^*:=X_l^*$. Repeatedly apply Neighbourhood Search starting with $X^t:=X_l^*$, incrementing t by 1 after each application, until the procedure returns with $\delta:=1$ or until no improvement has been recorded for $\bar{t}$ consecutive iterations. If Neighbourhood Search has been entered more than once and no improvement has been recorded during its last application, proceed to the next l. Otherwise, proceed to Step 3.

Step 3. Apply the Shake Up procedure to X^*.

As the algorithm produces several solutions, planners can either choose the best (that with the largest objective function value), or any of them based on other criteria. There lies one of the advantages of the proposed approach. In Fig. 5 we depict the alignment with the largest population cover on a particular test problem.

4. COMPUTATIONAL RESULTS

Extensive computational experiments were performed on 10 randomly generated instances to assess the behavior of the algorithm and to study the impact of the parameters. As is common in tabu search, parameter setting is done experimentally. Initial values are first selected using common sense and the experience accumulated by other researchers in the

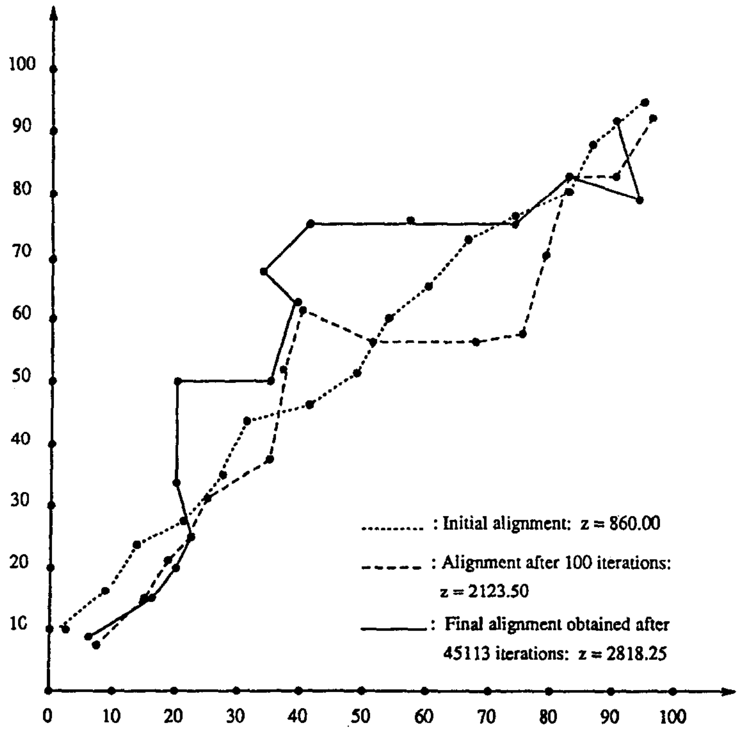

Fig. 5. Final alignment produced by the algorithm.

area of tabu search; for example, here we relied partly on the work of Gendreau *et al.* (1994) and of Taillard (1991). Sensitivity analyses are then performed on the most important parameters, the "best" values are identified and used for a final series of tests. These values partly depend on the characteristics of our test problems. Applying the same experimental methodology to other contexts may, of course, result in different parameter settings.

4.1. Problem generation

Each instance is characterized by the distribution of its population. To generate an instance or a "grid", we proceeded as follows.

Step 1. Assign each integer coordinate of 100×100 grid a population equal to zero.

Repeat Steps 2 and 3 25 times.

Step 2. Generate a 10×10 square, parallel to the axes, with an arbitrary center at a point with integer coordinates on the grid. Such a square covers at most 121 integer coordinates of the grid.

Step 3. Add to the population of each integer coordinate covered by the square a value of 1, 5 or 10, with equal probabilities.

This procedure is illustrated in Fig. 6. It tends to produce population distributions similar to what is witnessed in a number of cities—pockets of high density areas, irregularity and some continuity.

4.2. Basic tests

The algorithm was coded in *C* and run on a Sun Sparstation 10. We first present results for the basic tests, and then for the sensitivity analyses.

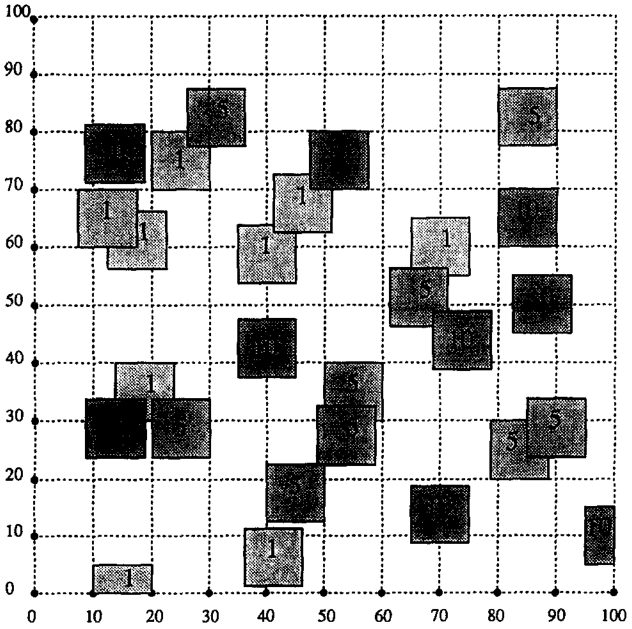

Fig. 6. Random generation of population on 100×100 grid.

Table 1. Parameter values for the basic tests

Number of stations (Section 2)	$n=15$
Limits on station interspacings (Section 2)	$l_{min}=8, l_{max}=16$
Factors used in the computation of $R(s)$, the population cover of station s (Section 2)	$\bar{d}=3, \theta_0=1, \theta_1=1, \theta_2=0.5, \theta_3=0.25$
Interval limits for tabu status (Section 3.4)	$m_{min}=5, m_{max}=10$
Number of lines initially generated (Sections 3.5 and 3.7)	$k_1=60$
Number of starting solutions retained after false starts (Sections 3.5 and 3.7)	$k_2=30$
Number of tabu iterations used for each false start (Section 3.5)	100
Maximum number of consecutive tabu iterations without improvement (Section 3.6)	$\bar{t}=10{,}000$
Number of stations moved in Shake Up procedure (Section 3.6)	$c=1$
Perimeter limit in the Shake Up procedure (Section 3.6)	$e=8$

All tests were executed on 10 basic grids (i.e. for 10 different population distributions), with the parameters shown in Table 1. In particular, $k_2=30$ alignments were produced for each grid. We present for each grid the best and average values over the 30 alignments. All computation times are expressed in minutes. The values of parameters used for the basic tests are not always the best. Sensitivity analyses have indeed revealed that different values are often preferable. At the end of this section, we present an unpdated series of test results obtained with the best set of parameters identified in our analyses.

We first present in Table 2 the best objective values obtained for each of the ten grids. This table gives the objective function value and the computing time to complete the search, using all k_2 starting solutions. All initial and final solutions were plotted and compared with a three-dimensional representation of population distribution. To illustrate this, in Fig. 7 we provide the diagram corresponding to Grid 9. This shows that the final solution can be very different from the starting solution. Also the best alignment produced by the tabu search heuristic seems to cover the most densely populated areas of the territory.

Table 2. Best values for the ten grids

Grid	Objective	Time
1	2921.50	74.3
2	2891.75	91.6
3	2601.25	91.6
4	2833.00	88.8
5	2901.50	89.3
6	2398.50	90.5
7	2744.75	97.0
8	2581.00	87.2
9	2409.25	76.7
10	2897.75	92.3

4.3. *Sensitivity analysis on the Shake Up parameter*

We have conducted sensitivity analyses on the values of the number of stations moved during the Shake Up procedure (c), the maximum number of consecutive iterations without improvement ($\bar{t}$), and the tabu interval ($[m_{min}, m_{max}]$).

When several stations are moved during a Shake Up, this is done sequentially, so as to respect the interspacing constraints (l_{min} and l_{max}). At each step, the feasible move yielding the best solution value is implemented even if this causes the objective to decrease. Our basic tests were executed with $c=1$. Increasing this value generally produces better solutions, but computing times can vary either way as this causes the tabu search process to follow a different path. Improvements, when they occur, are in the 2–6% range. Tests performed on all 10 grids with $c=1,\ldots,15$ indicate that *on average* the best value of c is 15, but the influence of this parameter on the average solution value is not critical. We can offer no convincing explanation for this lack of sensitivity.

4.4. *Sensitivity analysis on the maximum number of consecutive iterations without improvements*

One valuable piece of information provided by Table 2 is that more than half the total computation time is required to identify the local optimum. This indicates that most of the search is "productive" and the value of 10,000 for $\bar{t}$, the number of consecutive iterations without improvement, is probably not too high. An interesting question is whether is is worth using a significantly larger value for $\bar{t}$. We have thus redone all tests taking this time

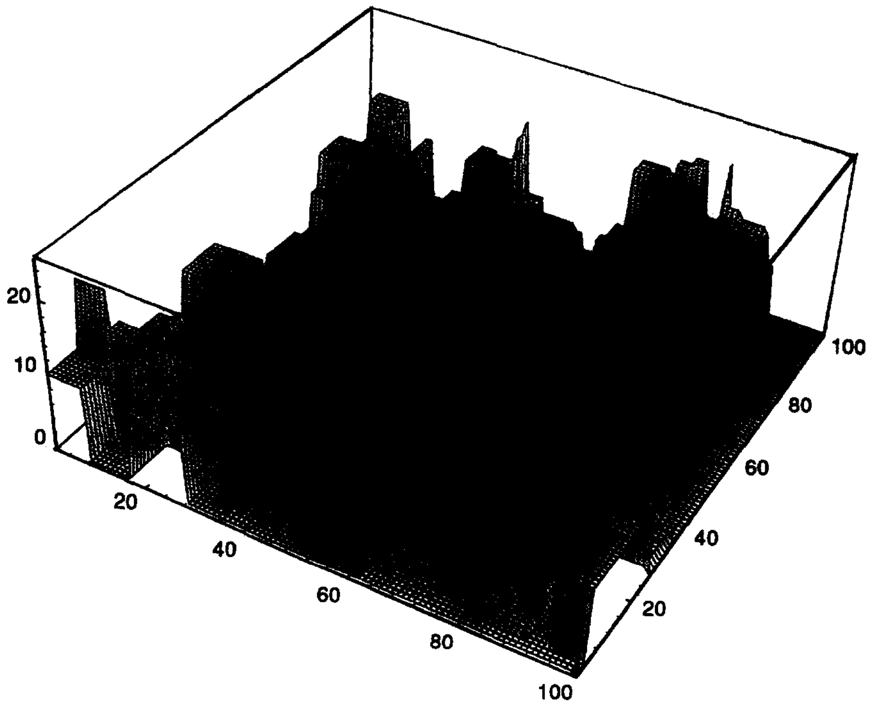

Fig. 7. Population distribution and alignment for Grid 9, using the parameters given in Table 1.

Table 3. Best objective values and computation times for $\bar{t}=10{,}000$ and $\bar{t}=20{,}000$

	$\bar{t}=10{,}000$ (from Table 2)		$\bar{t}=20{,}000$	
Grid	Objective	Time	Objective	Time
1	2921.50	74.3	3079.25	173.4
2	2891.75	91.6	2890.50	181.7
3	2601.25	91.6	2635.25	171.5
4	2833.00	88.8	2833.00	173.9
5	2901.50	89.3	2901.50	181.5
6	2398.50	90.5	2398.50	177.4
7	2744.75	97.0	2744.75	185.4
8	2581.00	87.2	2593.25	164.7
9	2409.25	76.7	2453.50	148.6
10	2897.75	92.3	2907.50	170.3
Average	2718.03	87.9	2743.70	172.8

$\bar{t}=20{,}000$. Results shown in Table 3 indicate that no significant gains are achieved by doubling $\bar{t}$. Moreover, computation times are typically much larger than when $\bar{t}=20{,}000$. We therefore conclude that using $\bar{t}=10{,}000$ is a reasonable choice.

4.5. *Sensitivity analysis on the tabu interval*

We have also analyzed the effect of the interval $[m_{\min}, m_{\max}]$ used to generate the length of the tabu status. Tests performed in the context of the Vehicle Routing Problem (Gendreau *et al.*, 1994) had revealed that the interval [5,10] was a very appropriate choice. For the transit line location problem, however, these values would appear to be too low to force the current alignment to sufficiently move away from its current location. In other words, with the current values, the search will often become stuck in a local maximum. Care must be taken, however, not to select too large values for $m_{\min}$ and $m_{\max}$ as all moves will soon become tabu and the search will end prematurely. We thus compare, in Table 5, four choices for $[m_{\min}, m_{\max}]$: [5,10], [20,40], [25,75] and [50,100]. Results show that increasing the bounds of the intervals from [5,10] has the double effect of improving the objective value and computing time. The best average solution values are obtained for $[m_{\min}, m_{\max}]=[20,40]$ and [25,75]. These two intervals give similar results in terms of solution quality, but [25,75] yields shorter computing times. With the interval [50,100], solution values deteriorate by about 4%.

With long tabu durations, the search process moves quickly to a good solution since much of the solution space is tabu. However, if $[m_{\min}, m_{\max}]$ becomes too large, too many moves are tabu and the search terminates prematurely. This explains the low computation times in the last column of Table 4. This analysis suggests that the interval [25,75] is a very good choice when $c=1$ given the tradeoff between solution quality and computing time.

4.6. *Final parameter values*

Combining the conclusions presented in Section 4.3 and the statistics of Table 4 would seem to indicate that taking $c=15$ and $[m_{\min}, m_{\max}]=[25,75]$ is a winning combination.

Table 4. Best objective values and computation times for $c=1$ and $[m_{min}, m_{max}]=[5,10]$, $[20,40]$, $[25,75]$ and $[50,100]$

	[5, 10]		[20, 40]		[25, 75]		[50, 100]	
Grid	Objective	Time	Objective	Time	Objective	Time	Objective	Time
1	2921.50	74.3	3672.00	47.2	3623.25	12.2	3589.75	5.1
2	2891.75	91.6	3627.25	54.9	3609.00	17.5	3415.75	5.5
3	2601.25	91.6	2644.50	76.7	2902.75	32.3	2653.00	8.9
4	2833.00	88.8	3308.50	41.4	3298.25	18.9	3205.50	5.6
5	2901.50	89.3	3878.25	55.9	3871.25	23.0	3764.50	6.5
6	2398.50	90.5	3034.75	54.7	3260.25	19.3	2735.00	4.4
7	2744.75	97.0	2848.00	64.2	2863.75	28.5	2785.75	7.3
8	2581.00	87.2	2954.75	66.6	2997.50	24.5	2926.50	5.6
9	2409.25	76.7	3140.00	65.0	3040.00	18.9	3018.00	6.7
10	2897.75	92.3	2903.25	52.7	2910.75	20.5	2912.25	8.2
Average	2718.03	87.9	3201.13	57.9	3237.70	21.6	3100.60	6.4

However, this is not the case as these parameters interact. When the number of stations moved during the Shake Up procedure is large and tabu restrictions are imposed for a large number of iterations, the search process becomes paralyzed as several interesting moves become tabu. As a result, good local optima are missed. We have attempted a mixed strategy: to use the [25,75] tabu interval in general and the [5,10] tabu interval for stations moved during Shake Up. This tended to produce about the same solutions as a pure strategy with $c=15$ and $m \in [25,75]$. On the whole, the best observed combination remains $c=1$ and $[m_{min}, m_{max}]=[25,75]$. The results of these analyses are presented in Table 5.

Table 5. Best solution values and computing times for some combinations of the parameters

	$c=1, m\in[5,10]$ (from Table 2)		$c=15, m\in[5,10]$		$c=1, m\in[25,75]$ (from Table 4)		$c=15, m\in[25,75]$		$c=15, m\in[25,75]$ in general; $m\in[5,10]$ in Shake Ups	
Grid	Objective	Time	Objective	Time	Objective	Time	Objective	Time	Objective	Time
1	2921.50	74.3	2595.45	49.4	3623.25	12.2	3139.14	9.2	3151.75	9.0
2	2891.75	91.6	2765.93	60.0	3609.00	17.4	3108.86	12.2	3192.94	11.3
3	2601.25	91.6	2168.56	45.5	2902.75	32.3	2417.92	14.5	2418.05	15.7
4	2833.00	88.8	2533.27	53.0	3298.25	18.9	2870.96	11.2	2835.93	8.7
5	2901.50	89.3	2573.21	59.4	3871.25	23.0	3033.97	11.6	2944.43	11.7
6	2398.50	90.5	2198.33	48.2	3260.25	19.2	2468.13	9.8	2506.95	10.2
7	2744.75	97.0	2265.05	46.8	2863.75	28.5	2554.01	17.3	2622.28	22.3
8	2581.00	87.2	2454.28	44.9	2997.50	24.4	2714.33	17.4	2723.96	15.1
9	2409.25	76.7	2267.01	45.8	3040.00	18.9	2831.21	17.9	2832.18	15.0
10	2879.75	92.3	2738.39	47.4	2910.75	20.5	2797.35	17.5	2807.42	19.2
Average	2718.03	87.9	2455.95	50.0	3237.70	21.6	2793.59	13.9	2803.59	13.8

4.7. Tests on instances with known optima

We have tested the final version of our algorithm (with $[m_{min}, m_{max}]=[25, 75]$ and $c=1$) of 10 grids with a known optimal alignment. To generate these instances we have modified each of our 10 grids as follows.

Step 1. Generate a grid as described in Section 4.1.

Step 2. For each point $s \in [0, 100]^2$ with integer coordinates, compute $R(s)$, as per equation (1) and set

$$R^* := \max_{s \in [1, 100]} \{R(s)\}.$$

Step 3. Randomly generate 15 stations in the restricted integer grid $[3, 97]^2$ as follows. Locate the first station at an arbitrarily chosen integer point on the east (left) boundary. Repeat the following two instructions 14 times:

(1) Determine λ, the number of feasible directions in which it is possible to move to locate the next station 12 units north, west or south of the last station located. A move is feasible if it does not locate a station outside the $[3, 96]^2$ grid and it does not locate a station on top of a previously located station. At any time, the number of feasible directions is equal to 2 or 3.

(2) Select a feasible direction with probability $1/\lambda$. Move 12 units in that direction to locate the next station.

Step 4. Remove all populations in the catchment area of each of the 15 stations just located. Recall that catchment area contains exactly 25 points (see Fig. 1). Assign each of these points a population of $R^*/12$ so that the cover of each station is equal to R^*.

Step 5. For each $s \in [1, 100]$, recompute $R(s)$ and set

$$R_1^* = \max_{s \in [1, 100]} \{R(s)\}.$$

If $R_1^* = R^*$ stop. Otherwise, set $R^* := R_1^*$ and go to Step 4.

Once this procedure is completed, the optimal solution value is $15R^*$. In Step 3, we use the sub-grid $[3, 97]^2$ to ensure that the catchment area of a station is never truncated by the boundary of the $[0, 100]^2$ grid. Step 5 is necessary to ensure that the previously computed R^* has not increased as a result of the new population assignments made in Step 4. Applying Step 5 thus guarantees that the solution value at the end of the procedure is indeed optimal.

We have then applied our algorithm to these ten modified grids to obtain a lower bound z on the value of the optimum. Using $k_2=30$, we obtained a ratio $z/15R^*$ varying from 0.88 to 1.00, with an average of 0.94. To analyze the impact of parameter k_2 (number of initial random alignments), we have also computed the best known ratio $z/15R^*$ using $k_2=5$, 10, 15, 20, 25, 30. The results are presented in Table 6. Finally, we have performed additional tests, using this time $l_{max}=24$ instead of $l_{max}=16$. From an algorithmic point of view, the effect of increasing l_{max} is to allow a wider neighbourhood exploration and better solutions. Using the same 10 grids, we have obtained optimal or near optimal solutions in all cases, and an average departure from optimality equal to only 1.15% when $k_2=30$.

These results must be interpreted with care. On the first set of problems ($l_{max}=16$), the algorithm produces solutions that are more than 90% optimal using only 10 starting configurations, but very little additional improvement is observed even with 30 starting

Table 6. Ratio $z/15R^*$ computed for the ten modified grids for k_2=5, 10, 15, 20, 25, 30

Grid	$k_2=5$	$k_2=10$	$k_2=15$	$k_2=20$	$k_2=25$	$k_2=30$
1	0.94	0.94	0.94	0.95	0.95	0.95
2	0.88	0.88	0.91	0.93	0.93	0.93
3	0.86	0.88	0.88	0.88	0.88	0.88
4	0.90	0.90	0.95	0.95	0.95	0.95
5	0.87	0.87	0.88	0.88	0.88	0.89
6	0.96	0.96	0.96	1.00	1.00	1.00
7	0.85	0.91	0.91	0.91	0.96	0.96
8	0.85	0.96	0.96	0.97	0.99	0.99
9	0.90	0.90	0.90	0.90	0.90	0.90
10	0.91	0.96	0.96	0.96	0.96	0.96
Average	0.89	0.92	0.93	0.93	0.94	0.94

solutions. On the other hand, when l_{max}=24 the deviation from optimality is always very small. This suggests that the performance of the tabu search algorithm is somewhat problem dependent. When the maximum spacing between two stations is large, the search process is less constrained and has the capacity to produce successive solutions that are significantly different from one another. As a result, the process will converge more rapidly towards good quality solutions.

5. CONCLUSIONS

We have considered the problem of locating a transit line in order to maximize population coverage, almost certainly the main criterion used in practice. The proposed heuristic generates a family of good alignments that can be assessed according to several criteria in order to help planners make a good choice. Our heuristic is based on tabu search, a very appropriate solution methodology for this type of unstructured problem. Depending on problem parameters, this algorithm will produce within reasonable computing times solutions whose quality varies from good to optimal. As our results indicate, selecting good parameters is critical to this type of approach, and this can only be achieved experimentally.

Other criteria such as the cost of the alignment could easily be incorporated into the objective. Also, transit lines are often not planned from scratch as there may already exist a partial network. The problem would then be a conditional location problem. This can easily be handled by simple modifications to our algorithm. Similarly, the method could also be extended to locate several lines simultaneously. Finally, we have chosen to run our tests on instances where the line can be located anywhere in a large territory. In practice, however, a broad corridor is first targeted for a transit line and the problem then becomes one of fine tuning an alignment within the restricted domain. This is precisely what happened when the Montreal metro was extended in the early eighties. Our approach is highly suited to this type of situation.

Acknowledgements—This work was partially funded by the Canadian Natural Sciences and Engineering Research Council (grants OGP0038816 and OGP0039682) and by the Quebec Government (FCAR grant 93ER0289). This support is gratefully acknowledged. Thanks are also due to two referees, to the associate editor and to John R. Current for their valuable comments.

REFERENCES

Bay, P. N. (1985) Determining cost-effectiveness of transit systems. Transportation Research Board State-of-the-Art Report 2—Light rail transit system design for cost-effectiveness, 9–12.

Benati, S. & Laporte, G. (1994) A tabu search algorithm for the $(r|X_p)$-medianoid and the $(r|p)$-centroid problems. *Location Science*, **2**, 193–204.

Blackledge, D. A. & Humphreys, E. M. H. (1984) The West Midland rapid transit study. *Proc. Planning and Transport Research & Computation Ltd.* Sussex, 71–84.

Bonz, M. (1983) Insertion et réalisation de l'infrastructure des métros légers dans le tissu urbain. *45e Congrès international de l'UITP*. Rio de Janeiro.

Chapleau, R., Lavigueur, P. & Baass, K. (1986) A posteriori impact analysis of a subway extension in Montreal. Publication #503, Centre de recherche sur les transports, Montreal.

Church, R. L. & Clifford, T. J. (1979) Discussion of environmental optimization of power lines by Economides and Sharifi. *J. Environ. Engineering Division, ASCE*. **105**, 438–439.

Condon, D. K. (1971) An extension of the Newark City subway system: its ridership, station location and effects. Urban mass transportation program, Newark College of Engineering, RWP 101.

Current, J. R. (1988) The design of a hierarchical transportation network with transshipment facilities. *Transportation Science* **22**, 270–277.

Current, J. R. & Schilling, D. A. (1994) The median tour and maximal covering tour problems: formulations and heuristics. *European Journal of Operational Research*, **73**, 114–126.

Current, J. R., Pirkul, H. & Rolland, E. (1994) Efficient algorithms for solving the shortest covering path problem. *Transportation Science*, **28**, 317–327.

Current, J. R., ReVelle, C. S. & Cohon, J. (1982) Multi-objective design of transportation networks. *Operations Research Group Report Series*, Report #82-02, Department of Geography and Environmental Engineering.

Current, J. R., ReVelle, C. S. & Cohon, J. (1985) Maximum covering/shortest path problems. *European Journal of Operational Research*, **21**, 189–199.

Current, J. R., ReVelle, C. S. & Cohon, J. (1986) The hierarchical network design problem. *European Journal of Operational Research*, **27**, 57–66.

Current, J. R., ReVelle, C. S. & Cohon, J. (1988) The minimum-covering/shortest path problem. *Decision Science*, **19**, 490–503.

Dicesare, F. (1970) A systems analysis approach to urban rapid transit guideway location. Ph.D. Dissertation, Department of Electrical Engineering, Carnegie-Mellon University.

Fukuyama, M. (1981) Optimal station location for a two hierarchy transit system. *Proc. Eight Int. Symp. Transport and Traffic Theory*, 264–291.

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

Gendreau, M., Laporte, G. & Mesa, J. A. (1995) Locating rapid transit lines. *Journal Advanced Transportation*, **29**, 145–162.

Gendreau, M., Laporte, G. & Semet, F. (1996) The covering tour problem. *Opers Res*. Forthcoming.

Gleason, J. M. (1975) A set covering approach to bus stop location. *Omega*, **3**, 605–608.

Glover, F. (1986) Future paths for integer programming and links to artificial intelligence. *Computer Operations Research*, **5**, 533–549.

Glover, F. (1989) Tabu search, part I. *ORSA Journal on Computing*, **1**, 190–206.

Glover, F. (1990) Tabu search, part II. *ORSA Journal on Computing*, **2**, 4–32.

Glover, F. & Laguna, M. (1993) Tabu search. In C. R. Reeves (ed.) *Modern Heuristics Techniques for Combinatorial Problems*. Blackwell, Oxford.

Gomes, L. (1989) Multicriteria ranking of urban transportation system alternatives. *Journal of Advanced Transportation*, **23**, 43–52.

Halder, D. K. & Majumder, A. (1981) A method for selecting optimum number of stations for a rapid transit system by network approach: an application in Calcutta tube rail. In N. K. Jaiswal (ed.) *Scientific Management of Transportation Systems*. Amsterdam, North-Holland.

Hansen, P. (1986) The steepest ascent mildest descent heuristic for combinatorial programming. Congress on Numerical Methods in Combinatorial Optimization, Capri.

Hertz, A., Taillard, E. & de Werra, D. (1996) Tabu search. In E. H. L. Aarts & J. K. Lenstra (eds). *Local Search in Combinatorial Optimization*. Wiley, Chichester.

Huber, D. & Church, R. L. (1985) Transmission corridor location modelling. *J. Transportation Engineering*, **111**, 114–130.

Jiménez Solano, J. (1993) Automated people movers. *Actas del I Symposium sobre Ingeniería de los Transportes*, Sevilla, 245–253.

Laporte, G., Mesa, J. A. & Ortega, F. (1994) Assessing topological configurations for rapid transit networks. *Studies in Locational Analysis*, **7**, 105–121.

Lutin, J. M. & Benz, G. P. (1992) Key issues in light rail transit station planning and design. *Transportation*

Research Record, **1361**, 117–124.
Magnanti, T. L. & Wong, R. T. (1984) Network design and transportation: models and algorithms. *Transportation Science*. **18**, 1–55.
Musso, A. & Vuchic, V. R. (1988) Characteristic of metro networks and methodology for their evaluation. *Transportation Research Record*, **1162**, 22–33.
Perrin, S. E. & Benz, G. P. (1990) Evaluating a large number of stations and alignment alternatives. *Transportation Research Record*, **1226**, 229–240.
Pirlot, M. (1992) General local search heuristics in combinatorial optimization: a tutorial. *Belgian Journal of Operations Research Statistics and Computer Science*, **32**, 8–67.
Quqing, G. (1984) Metro planning and construction in China. *Advanced Tunnelling Technology and Subsurface Use*, **4**, 67–69.
Schabas, M. (1988) Quantitative analysis of rapid transit alignment alternatives. *Transportation Quarterly*, **42**, 403–416.
Schumann, J. W. (1992) Status of North American LRT systems: 1992 update. *Transportation Research Record*, **1361**, 3–13.
Siegel, S. T. (1980) Major obstacles to effective L.R.T. surface operations. In *A Report on Light Rail Transit: Surface Operations*. Transportation Research Board, pp. 20–25.
Stone, J. R., Allen, J. D., Moerz, A. & Gardner, B. (1992) Transit system evaluation: guideway bus vs light rail transit. *Journal of Advanced Transportation*, **26**, 213–240.
Straus, P. (1980) Issues relating to effective L.R.T. surface operations. In *A Report on Light Rail Transit: Surface Operations*. Transportation Research Board, pp. 7–15.
Taillard, É. (1991) Robust taboo search for the quadratic assignment problem. *Parallel Computing*, **17**, 443–455.
Vincke, P. (1992) *Multicriteria Decision Aid*. Wiley, Chichester.
Wirasinghe, S. C. & Vandebona, U. (1987) Some aspects of the location of subway stations and routes. Presented at the *Fourth International Symposium on Locational Decisions*, Namur, Belgium.
Wulkan, A. & Henry, L. (1985) Evaluation of light rail transit for Austin, Texas. Transportation Research Board State-of-the-Art report 2—Light Rail Transit System Design for Cost-Effectiveness, 82–90.