INFORMATIQUE THÉORIQUE ET APPLICATIONS

FOTO AFRATI
STAVROS COSMADAKIS
CHRISTOS H. PAPADIMITRIOU
GEORGE PAPAGEORGIOU
NADIA PAPAKOSTANTINOU

The complexity of the travelling repairman problem

Informatique théorique et applications, tome 20, nº 1 (1986), p. 79-87
<http://www.numdam.org/item?id=ITA_1986__20_1_79_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 20, n° 1, 1986, p. 79 à 87)

THE COMPLEXITY OF THE TRAVELLING REPAIRMAN PROBLEM (*)

by Foto AFRATI (1), Stavros COSMADAKIS (2),
Christos H. PAPADIMITRIOU (3), George PAPAGEORGIOU (1)
and Nadia PAPAKOSTANTINOU (1)

Communicated by J. E. PIN

Abstract. — *In the travelling repairman problem (TRP) we are given a finite set of points and the travel times between any two of them, and wish to find the route through them which minimizes the sum of the delays for reaching each point. We consider the TRP when all points are on the straight line, and give a polynomial-time algorithm for it. If we have deadlines (that is, bounds on the maximum delay for each point) the problem becomes NP-complete, but can be solved by a pseudo-polynomial time algorithm.*

Résumé. — Le problème du réparateur itinérant consiste en la donnée d'un ensemble fini de points, et des temps de parcours entre ces points. Le but est de trouver une trajectoire qui passe par tous ces points, et qui minimise la durée totale du trajet. Nous étudions ce problème dans le cas où tous les points sont alignés, et nous en présentons une solution polynômiale. Si le retard maximum pour chaque point est borné, le problème devient NP-complet, mais il peut être résolu avec un algorithme pseudo-polynôme.

1. INTRODUCTION

Consider the following situation: n machines located at different points of a map have to be repaired, and there is only one repairman. We are given the time $t\,[ij]$ required by the repairman in order to travel from machine i to machine j. We are interested in finding the route which minimizes the mean waiting time of the machines. We assume that the repair times are insignificant (or, equivalently, the same for all machines). We call this mean-flow variant of the travelling salesman problem the travelling repairman problem (TRP).

(*) Received September 1984, revised March 1985.
(1) National Technical University of Athens.
(2) National Technical University of Athens and M.I.T.
(2) National Technical University of Athens and Stanford University.

ISSN en cours 86/01 79 9/$ 2.90/© Gauthier-Villars.

Formally, an instance of the TRP can be described as follows: Given a distance matrix $t\,[ij]$ between n locations, and a distinguished starting location s, find a permutation $\pi(0)=s, \pi(1), \ldots, \pi(n-1)$ such that the following cost function is minimized:

$$c=\sum_{i=1}^{n-1}\sum_{j=1}^{i} t[\pi(j-1), \pi(j)]=\sum_{i=1}^{n-1}(n-i)\,t[\pi(i-1), \pi(i)].$$

Since this problem is so closely related with the travelling salesman problem, it is not surprisingly that the general version of the TRP is NP-complete [2]. Moreover it is straight-forward to show that the Euclidean version of the TRP (that is, the special case in which the locations are points on the plane, and the distances are the Euclidean metric) is also NP-complete (by observing that standard reductions to the Euclidians TSP also work for the TRP). Unlike the TSP, however, it is non-trivial to solve the one-dimensional further restriction of the problem, that is, the case in which the locations are all on a straight line. This is called the *line-TRP*. The problem becomes even more difficult, if we assign to each machine a deadline, that is, an upper bound on its delay which the repairman must not violate in his route.

In this paper we present certain complexity results regarding both versions of the line-TRP (with deadlines and without). In particular, in the next section we present a dynamic programming algorithm that solves the line-TRP without deadlines in $O(n^2)$ time. In section 3 we prove that the line-TRP with deadlines is NP-complete, by reducing the 0-1 KNAPSACK to it; though further on, we prove that we can tell whether there exists a route preserving the deadline constraints in $O(n^2)$ time, also by dynamic programming. Finally, in section 4 we present a pseudo-polynomial algorithm solving the line-TRP with deadlines in time $O(n^2 D)$, where D is the largest deadline of the instance.

2. THE LINE-TRP

The instance of the line-TRP is best described as in figure 1, where $s=x_0=y_0$ is the starting location, $x_1, \ldots, x_m$ (resp. $y_1, \ldots, y_n$) are the machines to the left (resp. right) of the origin. Let us start with an elementary, although crucial, observation.

LEMMA 1: *If in the optimum route the repairman has visited machine x_i, them it has also visited machine x_j, $j<i$. Similarly for the machines to the right of s.*

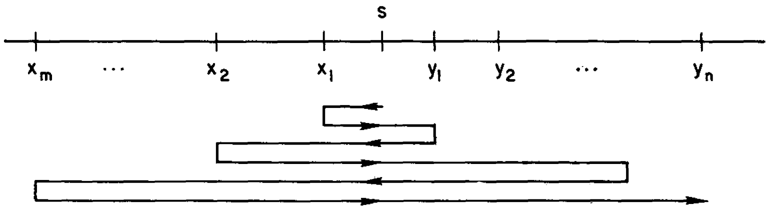

Fig. 1

Proof; Trivial. □

In other words, it is suboptimal for the repairman to "pass by" a machine without repairing it. Therefore, the optimum route looks like the one shown in figure 1.

Let us represent by $[x_i, y_j]$ the fact that the repairman is currently at x_i, which is also the leftmost location visited, while y_j is the rightmost location the repairman has visited. Similarly for $[y_j, x_i]$. By lemma 1, all locations between x_i and y_j are already visited. Thus, each such pair of locations represents a complete state of the route. The initial state is $[x_0, y_0]$, and the final is one of $[x_m, y_n]$, $[y_n, x_m]$. Finally, it follows from the lemma that state $[x_i, y_j]$ can be reached in an optimal route only from states $[x_{i-1}, y_j]$ and $[y_j, x_{i-1}]$.

We let $c\,[x_i, y_j]$ denote the minimum possible total delay accumulated by all locations during the time it took to visit all locations between x_i and y_j ending up in x_i. The above observations immediately lead to the following equations for computing this cost:

$$c\,[x_0, y_0]=c\,[y_0, x_0]=0,$$

$$c\,[x_i, y_j]=\min\{c\,[x_{i-1}, y_j]+(m+n+1-i-j)\,t\,[x_{i-1}, x_i],\; c\,[y_j, x_{i-1}]+(m+n+1-i-j)\,t\,[y_j, x_i]\},$$

(2.1)

$$c\,[y_i, x_j]=\min\{c\,[y_{i-1}, x_j]+(m+n+1-i-j)\,t\,[y_{i-1}, y_i],\; c\,[x_j, y_{i-1}]+(m+n+1-i-j)\,t\,[x_j, y_i]\},$$

where, in order to save two more equations, we let $c\,[x_i, y_{-1}]=c\,[y_j, x_{-1}]=\infty$. To justify (2.1), the second and third equations state that the minimum total delay in order to reach a particular state is the minimum total delay for reaching one of the two preceding states, plus the distance of the last locations in the two states multiplied by the number of unvisited locations at the previous state. Finally, the minimum cost is

(2.2) $$C=\min\{c\,[x_m, y_n],\; c\,[y_n, x_m]\}.$$

Thus we have:

THEOREM 1: *The line-TRP can be solved in $O(mn)$ time.*

Proof: We can compute all $c[x, y]$ using equations (2.1) inconstant time per value of $c[x, y]$ computed, and thus in $O(mn)$ time in total. In this estimate, our units of computation are arithmetic operations on integers of precision comparable to that of the largest integer in the cost matrix t (we shall later describe an algorithm whose *number of operations* depends exponentially on this precision). If we record at each computation which of the (at most two) candidate values was was the optimum, we can reconstruct the optimum path. □

3. THE LINE-TRP WITH DEADLINES

Suppose now that each machine i has a deadline d_i, and we impose the additional constraint that each machine must be repaired before its deadline expires. We show below that this problem is NP-complete. We redefine the problem for this purpose as usual, by also introducing another bound M, and asking whether there is a route such that all deadlines are met, while the sum of delays is at most M.

THEOREM 2: *The line-TRP with deadlines is NP-complete.*

THEOREM 2: *The line-TRP with deadlines is NP-complete.*

Proof: We reduce 0-1 KNAPSACK to line-TRP with deadlines. In this problem we are given a set [I] of n objects. Foo each $i \in I$ we are given its *size* s_i, and its *value* v_i, both integers. Finally, we are given two positive integers B and K. We are asked if there exist $x_1, \ldots, x_n \in \{0, 1\}$ such that:

$$\sum_{i=1}^{n} v_i x_i \geqq K, \tag{3.1}$$

$$\sum_{i=1}^{n} s_i x_i \leqq B. \tag{3.2}$$

In fact, we shall need a slightly modified version of the 0-1 KNAPSACK in which no two v_i are equal. It is trivial to prove that this version of the problem still remains NP-complete. We can also assume, without loss of generality, that $v_i < v_{i+1}$ for all i.

Given an instance of 0-1 KNAPSACK, we construct an instance of the line-TRP with deadlines as follows: For each element in $i \in I$ we add to our instance three locations. Two of these, z_i and z_i', are to the right of s at a distance of l_i and l_i' from s, respectively, and the third, Z_i, to the left at a distance of L_i (see *fig.* 2). Also, we add a point z_{n+1} to the right of s, at a

Fig. 2

distance l_{n+1}. All these lengths are defined below in terms of the parameters of the instance of 0-1 KNAPSACK:

(3.3) $$a_i = \frac{1}{2} s_i,$$

(3.4) $$a = \sum_{i=1}^{n} s_i,$$

(3.5) $$l_1 = a,$$

(3.6) $$l_i' = l_i + a_i,$$

(3.7) $$l_{i+1} = l_i' + a,$$

(3.8) $$L_i = Gv_i + 2a_i + 3(n-i)a_i - l_i.$$

Here G is a very large constant, whose role will be clear later. As for the deadlines:

(3.9) $$d_i = a_i,$$

(3.10) $$d_i' = d_i + a_i + 2l_i + 2L_i,$$

(3.11) $$D_i = d_i + 2a_i + l_i + L_i,$$

(3.12) $$d_{i+1} = d_i + 2a_i + l_i + 2L_i + l_{i+1},$$

(3.13) $$d_{n+1} = B + l_{n+1} + 2\sum_{i=1}^{n} (l_i + L_i),$$

and finally:

(3.14) $$M = N - 2GK,$$

where N stands for:

(3.15) $$N = 2\sum_{i=1}^{n} (n-i+1)(l_i + L_i) + 2a + 8\sum_{i=1}^{n} l_i + 5\sum_{i=1}^{n} L_i + l_{n+1}.$$

We claim that, there exists a legal route with cost $C \leqq M$ which achieves all deadlines if and only if there is a solution to the given instance of 0-1 KNAPSACK.

Suppose that $x_1, \ldots, x_n$ is the solution to the knapsack. We construct a route as follows: We first visit location z_1, then, if $x_1 = 1$ we repair machine z'_1, change direction, visit Z_1, change again, and visit z_2. On the other hand, if $x_1 = 0$, after z_1 we go to Z_1 first, and then to z'_1 and z_2. In general, we proceed in the same way, always changing direction at the locations on the left, while we change direction at the machine z_i (resp. z'_i) if $x_i = 0$ (resp. $x_i = 1$). Thus the route ends at z_{n+1}. We have to prove two things: (*a*) That the route is legal, in that it makes all the deadlines, and (*b*) that the corresponding cost is less than or equal to M. To prove (*a*), we have to show that no deadline is violated. For machines z_i, z'_i, and Z_i, $i = 1, \ldots, n$, this will be proved by induction on i. Denote by t_i, t'_i and T_i the time the repairman arrives at machine z_i, z'_i, and Z_i, respectively. For the basis step, it is obvious that $t_1 \leqq d_1$.

Assume that $t_i \leqq d_i$. Then, since z'_i will be visited first iff $x_i = 1$:

$$t'_i \leqq d_i + a_i + (2\, l_i + 2\, L_i)(1 - x_i)$$

and, by (3.10) $t'_i < d'_i$.

Next, for Z_i:

$$T_i = t_i + 2\, a_i\, x_i + l_i + L_i \leqq d_i + 2\, a_i + l_i + L_i$$

and by (3.11), $T_i \leqq D_i$.

Last, for z_{i+1}:

$$t_{i+1} = T_i + L_i + l_{i+1} \leqq D_i + L_i + l_{i+1}$$

and by (3.11), and (3.12) $t_{i+1} \leqq d_{i+1}$.

For the machine z_{n+1}, the time t_{n+1} can be recursively computed to:

$$t_{n+1} = 2 \sum_{i=1}^{n} (l_i + L_i) + l_{n+1} + 2 \sum_{i=1}^{n} a_j x_j. \tag{3.16}$$

By (3.3), (3.2) and (3.13) we obtain $t_{n+1} \leqq d_{n+1}$.

To prove part (*b*) we first compute the cost C. Notice that z_i is reached at time $t_i = l_i + 2 \sum_{j=1}^{i-1} (l_j + L_j + a_j x_j)$, z_i at time $t'_i = t_i + a_i + 2\,(l_i + L_i)\,(1 - x_i)$, and Z_i at time $T_i = t_i + l_i + L_i + 2\, a_i\, x_i$. The total contribution of the *i th* round of visits to the cost is then $C_i = a_i + 3t_i + 3(l_i + L_i) - 2(Gv_i + 3(n-i+1)a_i)x_i$. In adding the C_i's, notice that the $a_i x_i$ terms cancel, and thus:

$$C = N - 2G \sum_{i=1}^{n} v_i x_i. \tag{3.17}$$

Now, by (3.1) and (3.14), $C \leqq N - 2GK = M$, which complexes the proof in this direction.

Suppose now that we have a legal route with cost $C \leqq M$. First, we claim that the route has the form described above, that is, after z_i, we visit either z'_i, Z_i, z_{i+1} in this order, or Z_i, z'_i, z_{i+1}, in this order. Suppose not. Take the first time this form is violated. This can be done in four different ways:

(*a*) $z_i \to z'_i \to z_{i+1} \to Z_i$;

(*b*) $z_i \to z'_i \to Z_{i+1} \to z_{i+1}$;

(*c*) $z_i \to Z_i \to Z_{i+1} \to z'_i$;

(*d*) $z_i \to Z_i \to z'_i \to Z_{i+1}$.

Each one of the preceeding four cases leads to a contradiction, because the deadlines are defined in such a way, that they forbid exactly these kinds of routes. We shall prove only case (*b*) as an example:

$$\begin{aligned} t_{i+1} &= t_i + 2a_i + l_i + 2L_{i+1} + l_{i+1} \\ &\geqq d_i + 2a_i + l_i + 2L_{i+1} + l_{i+1} \\ &= d_{i+1} + 2(L_{i+1} - L_i) \end{aligned}$$

and thus

$$2(L_{i+1} - L_i) \leqq t_{i+1} - d_{i+1}.$$

In equation (3.8), however, the v_i's are strictly increasing and we can take take G to be as large as needed. Therefore, we can make the L_i's strictly increasing as well (say, by choosing $G > 4na$), and thus $t_{i+1} > d_{i+1}$. Thus, the deadline is violated.

Once we have established this "normal form" of the route, we can construct a solution of the 0-1 KNAPSACK by taking $x_i = 1$ whenever the route goes from z_i to z'_i, and $x_i = 0$ otherwise. It remains to be proved that such an assignment satisfies (3.1) and (3.2).

Since the route is a legal one, $t_{n+1} \leqq d_{n+1}$. However, by (3.16) and (3.13) we conclude that:

$$\sum_{i=1}^{n} s_i x_i \leqq B.$$

Finally, we know that $C \leqq M$, and by (3.17) and (3.14), we immediately obtain:

$$\sum_{i=1}^{n} v_i x_i \geqq K.$$

Thus the x_i's constructed constitute a solution of the knapsack problem, and the proof is complete. □

As an interesting aside, suppose that we wish to determine whether the given deadlines can be satisfied *at all*, no matter with how much total delay. The problem then is polynomial:

THEOREM 3: *Given an instance of the line-TRP with deadlines, we can tell whether there is some route that satisfies all the deadlines in* $O(mn)$ *time.*

Sketch: We can do this by a dynamic programming recurrence, as in theorem 1. One key observation is that a lemma analogous to lemma 1 holds here as well, in that it is suboptimal for the repairman to "pass by" machines, and thus pairs $[x, y]$ are adequate states. The recurrence now computes, for each pair $[x, y]$ as in theorem 1, the shortest time within which the repairman can visit all locations between x and y, ending up at y, *meeting all deadlines so far.* If no such route exists, this value is ∞. Our task is over once this time for $[x_m, y_n]$ or $[y_n, x_m]$ is finite. □

4. A PSEUDO-POLYNOMIAL ALGORITHM

We now attack the question of finding the optimum route, satisfying the deadlines (despite the fact that we proved it NP-complete in the previous section). Let us denote by $[x_i, y_j, t]$ the state at which the repairman at time t is currently at machine x_i, and has travelled at the opposite direction till machine y_j. Now we assign to each state the cost:

$$c[x_i, y_j, t] = \begin{cases} \infty \quad \text{if} \quad t > dx_i, \\ \min\{c[x_{i-1}, y_j, t - t[x_{i-1}, x_i]] + (m+n+1-i-j)\, t[x_{i-1}, x_i], \\ c[y_j, x_{i-1}, t - t[y_j, x_i]] + (m+n+1-i-j)\, t[y_j, x_i]\} \qquad \text{otherwise.} \end{cases}$$

The optimum total cost is given by:

$$C = \min\{c[x_m, y_n, t], c[y_n, x_m, t] : t = 0, 1, \ldots, D\},$$

where D is the longest deadline.

THEOREM 4: *The line-TRP with deadlines can be solved in $O\ (mnD)$.*

Sketch: By the straighforward implementation of the equations above. □

As usual, once we have a spseudo-polynomial algorithm, we can derive an algorithm which, given an instance of the TRP with deadlines and a "desired accuracy" ε, finds a route with relative error (that is, distance from the optimum, divided by the optimum) at most ε, in time polynomial in the size of the instance *and* $1/\varepsilon$. Such algorithms are called *fully polynomial-time approximation schemes* [*GJ*].

Corollary: *There is a fully polynomial-time approximation scheme for the TRP with deadlines, with time bound $O\ (mn\ (m+n)/\varepsilon)$.*

Sketch: We round off the deadlines to the next smaller multiple of $[D\ \varepsilon/(m+n)]$, and the travel times to the next larger multiple of $[D\ \varepsilon/(m+n)]$. The pseudo-polynomial algorithm then gives the desired accuracy and time bound. □

5. AN OPEN PROBLEM

When whe have, not deadlines, but *repair times*, the problem seems to become more complex. Naturally, it may now be better for the repairman to "pass by" a machine with a long repair time, in order to reach and repair some easy ones first. When the repair times are insignificant compared to the travel times, we have shown that the problem is in *P*. When the travel times are insignificant, then it is easy to see that it is optimum to visit the machines in order of increasing repair times. It is not clear, however, how to extend either of these ideas to the general case.

ACKNOWLEDGMENTS

This research was supported by the Hellenic Ministry of Research and Technology.

REFERENCES

1. M. R. GAREY and D. S. JOHNSON, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and co., San Francisco, 1979.
2. S. SAHNI and T. GONZALES, *P-Complete Approximation Problems*, J. Assoc. Comput. Mach., Vol. 23, 1976, pp. 555-565.