k-Interchange procedures for local search in a precedence-constrained routing problem

Harilaos N. PSARAFTIS
Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.

Received January 1981
Revised June 1982

We develop k-interchange procedures to perform local search in a precedence-constrained routing problem. The problem in question is known in the Transportation literature as the single vehicle many-to-many Dial-A-Ride Problem, or DARP. The DARP is the problem of minimizing the length of the tour traveled by a vehicle to service N customers, each of whom wishes to go from a distinct origin to a distinct destination. The vehicle departs from a specified point and returns to that point upon service of all customers. Precedence constraints in the DARP exist because the origin of each customer must precede his/her destination on the route. As in the interchange procedure of Lin for the Traveling Salesman Problem (TSP), a k-interchange is a substitution of k of the links of an initial feasible DARP tour with k other links, and a DARP tour is k-optimal if it is impossible to obtain a shorter tour by replacing any k of its links by k other links. However, in contrast to the TSP where each individual interchange takes O(1) time, checking whether each individual DARP interchange satisfies the origin-destination precedence constraints normally requires $O(N^2)$ time. In this paper we develop a method which still finds the best k-interchange that can be produced from an initial feasible DARP tour in $O(N^k)$ time, the same order of magnitude as in the Lin heuristic for the TSP. This method is then embedded in a breadth-first or a depth-first search procedure to produce a k-optimal DARP tour. The paper focuses on the $k = 2$ and $k = 3$ cases. Experience with the procedures is presented, in which k-optimal tours are produced by applying a 2-opt or 3-opt search to initial DARP tours produced either randomly or by a fast $O(N^2)$ heuristic. The breadth-first and depth-first search modes are compared. The heuristics are seen to produce very good or near-optimal DARP tours.

Work for this paper was supported in part by a grant from the U.S. Department of Transportation, Urban Mass Transportation Administration. I would like to thank Amedeo Odoni, Dick Larson and two anonymous reviewers for their comments.

North-Holland
European Journal of Operational Research 13 (1983) 391–402

1. Introduction

In its most generic form, the single vehicle many-to-many Dial-A-Ride Problem (DARP) is the problem of minimizing the length of the tour traveled by a vehicle which services N customers, each of whom wishes to go from a distinct origin to a distinct destination. The vehicle performs the tour by departing from a specified point and returning to the same point upon service of all customers. The DARP is a constrained version of the Traveling Salesman Problem (TSP) that is defined on its $2N + 1$ nodes, the constraints regarding the precedence relationships between the origin and destination of each customer on a feasible Dial-A-Ride tour. Several approaches for solving the above or other versions of the DARP have been presented during the past few years: For example, Wilson and others [13,14] have developed routing algorithms for Dial-A-Ride systems operating in Rochester, NY; Gavish and Srikanth [3] have developed mathematical formulations of the problem; Stein [11] has presented a probabilistic analysis of some heuristic algorithms on the problem; Psaraftis [8] has solved the above problem exactly using Dynamic Programming; Sexton and Bodin [2,10] have developed approximate algorithms based on Benders decomposition for subscriber Dial-A-Ride systems; Tharakan and Psaraftis [12] have solved the problem exactly in the case where the disutility of customers is exponential with time; and finally Psaraftis [9] has presented a worst case and an average performance analysis of some new polynomial-time heuristics for the problem. For a comprehensive review on the state of the art on this problem see Bodin, Golden and others [1].

Due to the fact that the generic version of the DARP described above is NP-complete, no efficient method for solving the problem *exactly* is known. The exact Dynamic Programming approach of Psaraftis requires $O(N^2 3^N)$ time, [8],

fact which limits the tractable problem size to no more than 9 or 10 customers, or about 20 nodes. Consequently, many researchers feel that the thrust on this problem should be based on fast, heuristic procedures, and this is reflected in the recent literature.

As in a heuristic approach to any optimization problem, a heuristic procedure for the DARP incorporates the following two general steps:

Step 1: Obtain an initial feasible Dial-A-Ride tour.
Step 2: Perform a 'local search' on that tour, to obtain an improved tour, or show that no improvement is possible.

Although the literature is relatively rich on how Step 1 of the above generic procedure can be tackled, little or nothing has been done to date to address Step 2 for the DARP. The purpose of this paper is to focus on this second step and to present a methodology for performing efficient local searches on Dial-A-Ride tours.

Our approach draws from the k-interchange procedure introduced by Lin [5] and Lin and Kernighan [6] for the TSP. As in the TSP, a *k-interchange* in the DARP is a substitution of k of the links of a DARP tour with k-other links. [1] A DARP tour is said to be *k-optimal* (or k-opt) if it is impossible to obtain another DARP tour of shorter length by replacing any k of its links by any other set of k links. Several other papers, such as Papadimitriou and Steiglitz [7], and Kanellakis and Papadimitriou [4], have examined issues related to the application of the above procedure to the TSP.

In contrast to the TSP where each individual interchange takes O(1) time, checking whether each individual DARP interchange satisfies the origin-destination precedence constraints normally requires $O(N^2)$ time. In this paper we develop a method which still finds the best k-interchange that can be produced from an initial feasible DARP tour in $O(N^k)$ time, the same order of magnitude as in the Lin heuristic for the TSP. This method is then embedded in a breadth-first or a depth-first search procedure to produce a k-optimal DARP tour. The paper focuses on the $k = 2$ and $k = 3$ cases. Experience with the procedures is presented, in which k-optimal tours are produced by applying a 2-opt or 3-opt search to initial DARP tours produced either randomly or by a fast $O(N^2)$ heuristic. The breadth-first and depth-first search modes are compared.

[1] Throughout this paper we will assume that a k-interchange is *strict*, that is, *all* of its k links will be substituted by k *other* links. Non-strict interchanges, where only *some* of the k links are substituted can be similarly considered. Notice that a non-strict k-interchange is a strict k'-interchange for some $k' < k$, hence our assumption causes no loss of generality.

2. The $k = 2$ case

Before we begin with the $k = 2$ case it is useful to introduce the notation we will be using throughout the paper.

A DARP tour of N customers (labeled $n = 1, \ldots, N$) can be described by a sequence $(S_0, S_1, \ldots, S_i, \ldots, S_{2N+1})$ where $i = 0, 1, \ldots, 2N + 1$ is a counter representing the ith stop of the tour and S_i is defined as follows:

$$S_i = \begin{cases} 0 & \text{if } i = 0 \text{ or } 2N+1 \text{ (starting and ending point of the tour),} \\ +n & \text{if vehicle picks up customer } n \text{ at stop } i, \\ -n & \text{if vehicle delivers customer } n \text{ at stop } i \end{cases}$$

$$(i = 0, 1, \ldots, 2N + 1).$$

It is clear that a DARP tour has $2N + 2$ links. We assume that the distance matrix $[C(S_i, S_j)]$ of any link that can be defined in our problem is known and symmetric.

An alternative representation of a DARP tour is via the matrix $[m(n, i)]$. $m(n, i)$ is the 'status' of customer n at the ith stop of the DARP tour. By convention:

$$m(n, i) = \begin{cases} 3 & \text{if customer } n \text{ has not been picked up so far,} \\ 2 & \text{if customer } n \text{ is on board the vehicle,} \\ 1 & \text{if customer } n \text{ has been delivered.} \end{cases}$$

We will find this representation very useful in our subsequent analysis. It is clear that matrix $[m]$ can be constructed from array (S) in $O(N^2)$ time.

A DARP tour is shown in Fig. 1(a). Only two links of that tour, (S_i, S_{i+1}) and $(S_{i'}, S_{i'+1})$ are shown in the figure. In general there will be more stops in the segments $(0, S_i)$, $(S_{i+1}, S_{i'})$ and $(S_{i'+1}, 0)$. We observe that there is *a unique direction* we can traverse a given DARP tour. Traversing the

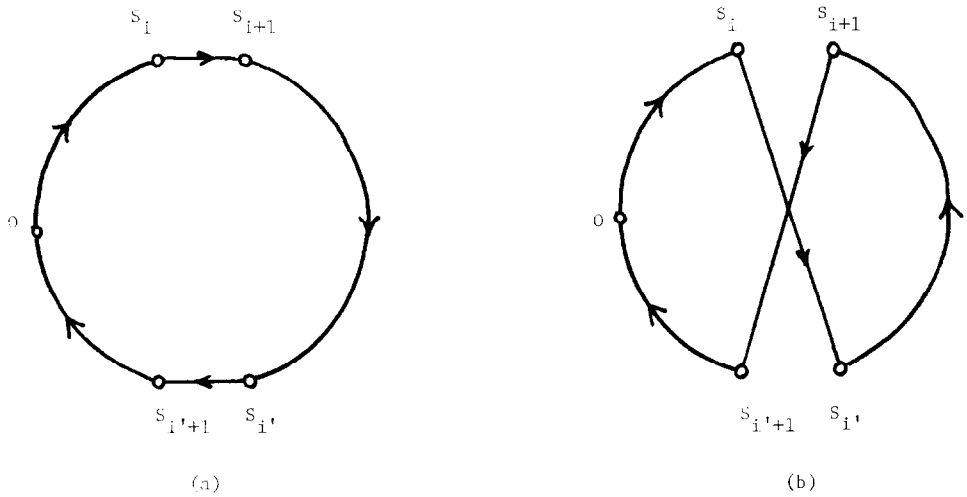

Fig. 1. A 2-interchange.

entire tour in the opposite direction would violate the origin-destination precedence constraints. This remark holds despite the fact that the distance matrix of the DARP is symmetric.

Performing an individual 2-interchange on a DARP tour involves the substitution of two of its links, say (S_i, S_{i+1}) and $(S_{i'}, S_{i'+1})$ in Fig. 1(a), with two other links, in this case $(S_i, S_{i'})$ and $(S_{i+1}, S_{i'+1})$ in Fig. 1(b). In the TSP such an interchange results in a local tour improvement if and only if $C(S_i, S_{i+1}) + C(S_{i'}, S_{i'+1}) > C(S_i, S_{i'}) + C(S_{i+1}, S_{i'+1})$ (favorable interchange). This condition is necessary for improving the tour in the DARP as well. However, a complication in the DARP is that the new tour resulting from the interchange may not be feasible. Notice that the direction by which the segment $(S_{i+1}, S_{i'})$ is traversed after the interchange (Fig. 1(b)) is the *opposite* of its original direction (Fig. 1(a)). Therefore it is conceivable that traversing that segment from $S_{i'}$ to S_{i+1} may violate the origin-destination precedence constraints, in which case the interchange in question is not feasible. It is thus clear that we should check whether an interchange is feasible before performing it.

For a given initial DARP tour, finding the best favorable 2-interchange (or proving that none exists) requires 'checking out' all possible ways to substitute links similar to (S_i, S_{i+1}) and $(S_{i'}, S_{i'+1})$ of Fig. 1(a) with links similar to $(S_i, S_{i'})$ and $(S_{i+1}, S_{i'+1})$ of Fig. 1(b). 'Checking out' means determining both whether the substitution is feasible and whether it results in a tour improvement. Since $0 \leqslant i \leqslant 2N-2$ and $i+2 \leqslant i' \leqslant 2N$, this means that the total number of possible 2-interchanges, and hence, 'check outs' as described above is equal to $\sum_{i=0}^{2N-2}(2N-i-1) = N(2N-1)$, or $O(N^2)$. The total computational effort will depend on the amount of time needed for each 'check out', as well as on how this verification procedure is structured within the algorithm.

An obvious but crude way to determine whether or not any proposed 2-interchange is feasible is to examine the interchange in question 'as we go' through the process of enumerating all 2-interchanges. The only way that the 2-interchange proposed in Fig. 1 is infeasible is if there is *at least one* customer who has his/her origin *and* destination in the segment $(S_{i+1}, S_{i'})$. For, if this were the case, inverting the segment would violate the origin-destination precedence constraints for that customer.

The most simple way to find out whether such a customer exists is to examine all pairs of nodes in the segment $(S_{i+1}, S_{i'})$ and look if there is at least one pair of the form $(+n, -n)$. This operation will take $O(N^2)$ time, and, if executed at every 2-interchange under consideration, will bring the total computational effort of the procedure to $O(N^4)$.

Although such a complexity is polynomial, it is by no means modest, particularly if the procedure is to be applied repeatedly to problems of considerable size (e.g. $N = 100$). An obvious question to

answer therefore is the following: Can we bring the complexity of this procedure to a rate lower than $O(N^4)$? The answer to this question turns out to be yes. A more daring question is: Can we bring the complexity of the procedure to a rate of $O(N^2)$, the same as in the TSP where no feasibility checks are needed? It turns out that the answer to the second question is also yes. The rest of this section will explain why.

A first step toward reducing the complexity of the algorithm can be made if we determine feasibility as follows:

Consider any particular 2-interchange similar to the one portrayed in Fig. 1, that is, any particular combination of i and i'. Then look at the ith and i'th columns of the $[m]$ matrix corresponding to the DARP tour in question (see beginning of this section). If there is a customer n for whom $m(n, i) = 3$ and $m(n, i') = 1$, then the 2-interchange in question is *infeasible*. If no such customer exists, this 2-interchange is feasible.

Performing the above procedure reduces the total computational complexity to $O(N^3)$, for at each proposed 2-interchange we have to perform $O(N)$ checks. The logic of the procedure is easy to follow, for only if $m(n, i) = 3$ and $m(n, i) = 1$ are customer n's origin *and* destination within the segment $(S_{i+1}, S_{i'})$.

The computational complexity can be reduced to $O(N^2)$ if we separate the feasibility checks from the improvement checks. This can be done by performing a 'screening' procedure in the beginning of the algorithm, the purpose of which is to determine the feasibility of every possible 2-interchange. Information from such a screening procedure is then stored in a matrix, to be used immediately afterwards in the 'optimization' part of the algorithm, where tour improvements are considered. In such a way, the latter part of the algorithm runs in $O(N^2)$ time, for each individual 2-interchange takes O(1) time. As it will be seen below, the 'screening' part of the algorithm also runs in $O(N^2)$ time, bringing the complexity of the whole procedure to $O(N^2)$.

To describe how the 'screening' part works, we need to define the following function:

For a given DARP tour and a given stop i ($0 \leqslant i \leqslant 2N - 2$) let FIRSTDEL($i$) be the position of the first of the deliveries remaining beyond stop $i + 1$, for which the corresponding customer has not been picked up to and including stop i. If no such delivery exists, set FIRSTDEL(i) $= 2N + 1$.

In mathematical terms, FIRSTDEL(i) $= x$ if x is the smallest number above $i + 1$ for which there exists a customer n so that $m(n, i) = 3$ and $m(n, x) = 1$. If no such customer exists, then $x = 2N + 1$.

We illustrate the above definition by an example: Table 1 shows a representation of a DARP tour for $N = 4$ customers, in terms of its sequence $(S_0, \ldots, S_9)$ and its matrix $[m]$. To find FIRSTDEL(0) we search for the first delivery *beyond* stop #1 for which the corresponding customer has not been picked up to and including stop #0. Clearly, this is the delivery of customer #1 which occurs at stop #4, hence FIRSTDEL(0) = 4. To find FIRSTDEL(1) we search for the first delivery beyond stop #2 for which the corresponding customer has not been picked up up to and including stop #1. Clearly, this is the delivery of customer 4 which occurs at stop #6 (customer 1 *has* been picked up by stop #1 so his/her delivery is no longer considered). Proceeding similarly we find FIRSTDEL(i) for $i = 2, \ldots, 6$, information which is summarized as follows:

FIRSTDEL(0) = 4,

FIRSTDEL(1) = 6,

FIRSTDEL(2) = 7,

FIRSTDEL(3) = 7,

FIRSTDEL(4) = 7,

Table 1
An illustrative example of a DARP tour ($N = 4$)

i	0	1	2	3	4	5	6	7	8	9
S_i	0	+1	+4	+2	−1	+3	−4	−3	−2	0
$m(1, i)$	3	2	2	2	1	1	1	1	1	1
$m(2, i)$	3	3	3	2	2	2	2	2	1	1
$m(3, i)$	3	3	3	3	3	2	2	1	1	1
$m(4, i)$	3	3	2	2	2	2	1	1	1	1

FIRSTDEL(5) = 9,

FIRSTDEL(6) = 9.

The function takes the value of 9 for $i = 5$ and 6 because no customers exist beyond $i = 5$ which have not been picked up.

Theorem 1 concerns the feasibility of a particular 2-interchange.

Theorem 1. *The substitution of links* (S_i, S_{i+1}) *and* $(S_{i'}, S_{i'+1})$ *with links* $(S_i, S_{i'})$ *and* $(S_{i+1}, S_{i'+1})$ *is feasible if and only if* $i' <$ FIRSTDEL(i).

The proof of this theorem is obvious and hence omitted.

The 'screening' part of the algorithm is therefore as follows:

Step 1: Calculate the values of FIRSTDEL(i). The algorithm for doing so is formalized in Fig. 2.

Step 2: Create a matrix FE(i, i') with values 'true' or 'false' depending on whether the substitution of links (S_i, S_{i+1}) and $(S_{i'}, S_{i'+1})$ with links $(S_i, S_{i'})$ and $(S_{i+1}, S_{i'+1})$ is feasible or not. The matrix is created as follows:

do $i = 0$ **to** $2N - 2$
  do $i' = i + 2$ **to** $2N$
  FE(i, i') = 'false'
  if $i' <$ FIRSTDEL(i), FE(i, i') = 'true'
  end loop i'
end loop i

Each of the above steps can be executed in $O(N^2)$ time.

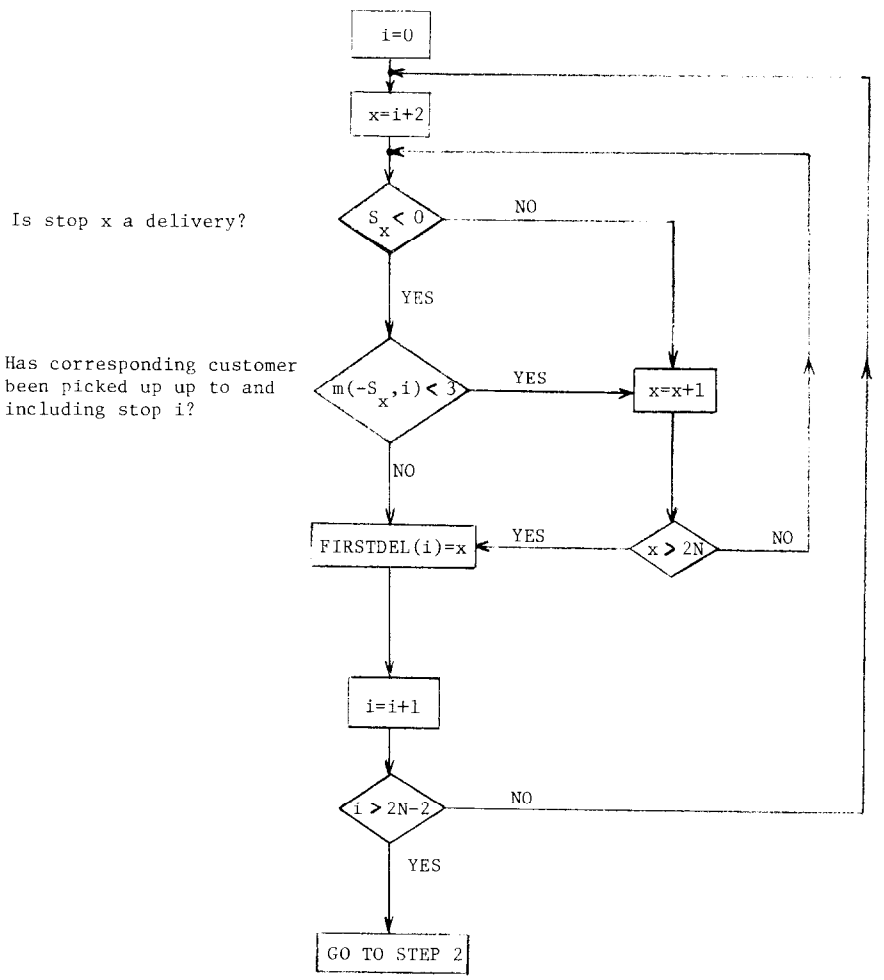

Fig. 2. $k = 2$: Step 1 of the 'screening' procedure (calculate the values of FIRSTDEL(i)).

Table 2
Feasibility matrix [FE(i, i')] for all possible 2-interchanges of the DARP tour of Table 1 (T: 'true', F: 'false'; circles represent positions where $i' = \text{FIRSTDEL}(i)$)

i	i'						
	2	3	4	5	6	7	8
0	T	T	Ⓕ	F	F	F	F
1	–	T	T	T	Ⓕ	F	F
2	–	–	T	T	T	Ⓕ	F
3	–	–	–	T	T	Ⓕ	F
4	–	–	–	–	T	Ⓕ	F
5	–	–	–	–	–	T	T
6	–	–	–	–	–	–	T

For our illustrative example, the matrix [FE] after the execution of Steps 1 and 2 of the screening procedure is summarized in Table 2 (T: 'true', F: 'false'). Circles represent positions where $i' = \text{FIRSTDEL}(i)$.

After the 'screening' part is executed, the procedure goes through the 'optimization' or tour improvement part in the same way as in the Lin heuristic for the TSP. Any particular 2-interchange for which FE(i, i') = 'false' is immediately discarded.

It should be emphasized here that the method described above only finds the best 2-interchange out of a given DARP tour, if such interchange exists. This by no means implies that the resulting tour is 2-optimal, because further improvement might be achieved if the procedure is applied again to the improved tour. Hence, to find a 2-optimal tour, the procedure has to be applied a number of times. There are two main ways for doing so (breadth-first and depth-first search) and they will be described in Section 4.

3. The $k = 3$ case

It is straightforward to extend the approach presented in the previous section to the $k > 2$ case. However, instead of presenting the general $k > 2$ case we would like to focus on the $k = 3$ case because some subtle points of that case merit special attention. The discussion of those points is expected to provide sufficient insight for the kinds of issues that are likely to be encountered in the general $k > 2$ case.

In contrast to the $k = 2$ case, where the two links (S_i, S_{i+1}) and $(S_{i'}, S_{i'+1})$ that will be dropped, uniquely identify the two links $(S_i, S_{i'})$ and $(S_{i+1}, S_{i'+1})$ that will substitute them, in the $k = 3$ case there are *two ways* of substituting any given triplet of links with a triplet of other links. Figures 3(b) and 3(c) show the two possible 3-interchanges that can be performed by dropping links (S_i, S_{i+1}), $(S_{i'}, S_{i'+1})$ and $(S_{i''}, S_{i''+1})$ of a given initial DARP tour (Fig. 3(a)).

It is straightforward to check that the DARP tour shown in Fig. 3(b) can be obtained from the one of Fig. 3(a) after the successive application of *two* 2-interchanges. Also, the tour of Fig. 3(c) can be derived after *three* 2-interchanges are applied to the initial tour.

Without loss of generality we will assume that all our 3-interchanges will be similar to the one of Fig. 3(c). Our analysis requires minor modifications if the 3-interchange is similar to that of Fig. 3(b). In other words, links (S_i, S_{i+1}), $(S_{i'}, S_{i'+1})$ and $(S_{i''}, S_{i''+1})$ will be substituted by links $(S_i, S_{i'+1})$, $(S_{i''}, S_{i''+1})$ and $(S_{i'}, S_{i+1})$. The condition

$$C(S_i, S_{i+1}) + C(S_{i'}, S_{i'+1}) + C(S_{i''}, S_{i''+1}) > C(S_i, S_{i'+1}) + C(S_{i''}, S_{i''+1}) + C(S_{i'}, S_{i+1})$$

is a necessary condition for the interchange in question to result in a local tour improvement.

Naturally, as in the $k = 2$ case, this condition is not sufficient, for the interchange might be infeasible. However, in contrast to the $k = 2$ case, the $k = 3$ case preserves *the direction* by which the segments $(S_{i+1}, S_{i'})$ and $(S_{i'+1}, S_{i''})$ will be traversed after the interchange. (One should note that this is true only for a 3-interchange of the type of Fig. 3(C)). Therefore, if these segments are feasible in the initial tour, they will be also feasible in the final tour, *except only* if there exists in the former segment an origin whose corresponding destination lies in the latter. This is true since *the order* by which these segments are traversed is reversed after the interchange (compare Figs. 3(a) and 3(c)). In that respect, checking whether the interchange in question is feasible would require a different kind of 'screening' test than the one applied on the $k = 2$ case. Other than the above difference, we can extend the methodology developed in the previous section to the $k = 3$ case in a straightforward manner as we will see below.

In order to find the best favorable interchange

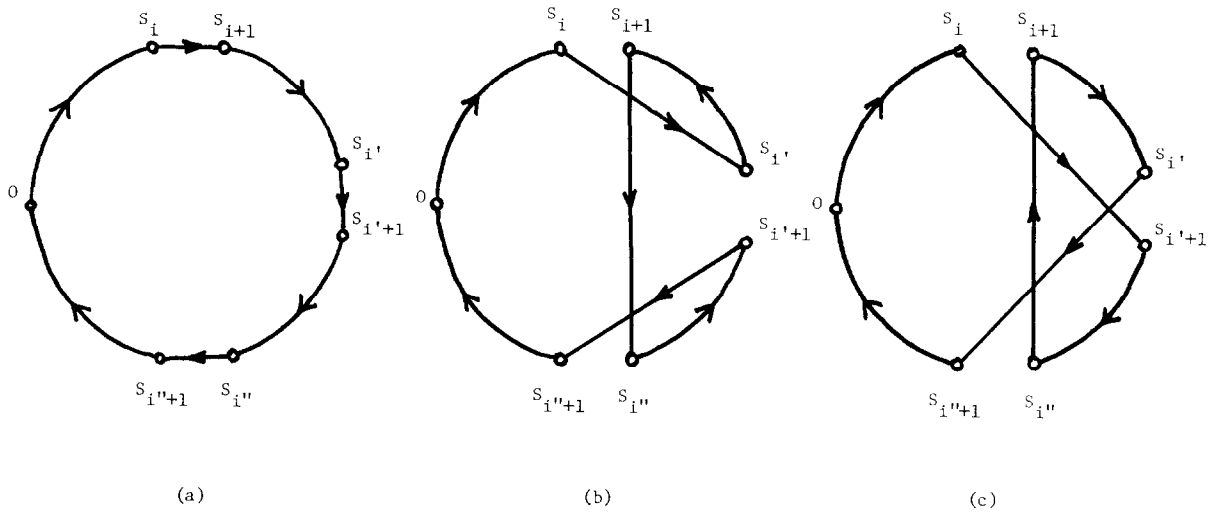

Fig. 3. Two ways to perform a 3-interchange. The one shown in (c) will be followed throughout this paper.

out of an initial DARP tour (or prove that none exists) we have to 'check out' all possible ways to substitute links (S_i, S_{i+1}), $(S_{i'}, S_{i'+1})$ and $(S_{i''}, S_{i''+1})$ with links $(S_i, S_{i'+1})$, $(S_{i'}, S_{i''+1})$ and $(S_{i''}, S_{i+1})$. Since $0 \leqslant i \leqslant 2N-2$, $i+1 \leqslant i' \leqslant 2N-1$ and $i'+1 \leqslant i'' \leqslant 2N$, this means that the total number of possible 3-interchanges is equal to

$$\sum_{i=0}^{2N-2} \sum_{i'=i+1}^{2N-1} (2N-i') = \tfrac{1}{3}N(2N-1)(2N+1),$$

or $O(N^3)$. Despite the fact that checking whether any individual 3-interchange is feasible can take $O(N^2)$ time, we can apply a method similar to the one developed in the previous section to keep the total computational effort to $O(N^3)$. As in the $k=2$ case, the 'screening' procedure is performed before the 'optimization' procedure and is based on a function that is defined as follows:

For a given DARP tour and two given stops i $(0 \leqslant i \leqslant 2N-2)$ and i' $(i+1 \leqslant i' \leqslant 2N-1)$, let FIRSTDEL$(i, i')$ be the position of the first of the deliveries remaining beyond stop i', for which the corresponding customer has not been picked up up to and including stop i, but is on board the vehicle at stop i'. If no such delivery exists, set FIRSTDEL$(i, i') = 2N+1$.

Mathematically, FIRSTDEL$(i, i') = x$ if x is the smallest number above i' for which there exists a customer n so that $m(n, i) = 3$, $m(n, i') = 2$ and $m(n, x) = 1$. If no such customer exists, then $x = 2N+1$.

Theorem 2 concerns the feasibility of a particular 3-interchange.

Theorem 2. *The substitution of links* (S_i, S_{i+1}), $(S_{i'}, S_{i'+1})$ *and* $(S_{i''}, S_{i''+1})$ *with links* $(S_i, S_{i'+1})$, $(S_{i'}, S_{i''+1})$ *and* $(S_{i''}, S_{i+1})$ *is feasible if and only if* $i'' <$ FIRSTDEL(i, i').

The proof of the theorem is obvious and hence omitted.

The algorithm for computing the values of FIRSTDEL(i, i'), as well as the 'screening' part of the procedure are similar to those presented in the previous section for $k=2$ and are omitted as well. Each of the above steps can be executed in $O(N^3)$ time.

As in the $k=2$ case, the method described above will find the best 3-interchange out of a given DARP tour, or show that none exists. The following section discusses, among other things, ways to arrive at a 2-optimal or a 3-optimal tour.

4. Computational experience

There is considerable flexibility on how the procedures described in the last two sections can be used so as to obtain a k-optimal DARP tour. The flexibility concerns the following:

(1) The *mode* of local search,

(2) The way the initial DARP tour is obtained.

Concerning the *mode* of local search, we will distinguish between breadth-first and depth-first search.

Breadth-first (or, steepest descent) search finds the *best* of all favorable k-interchanges that can result from a given DARP tour. This can be done as described in the previous two sections for $k = 2$ and $k = 3$ and involves $O(N^k)$ operations. Once such an interchange is found, the initial DARP tour is replaced and the procedure is applied again to the new tour. The process is repeated until no favorable k-interchange can be found.

By contrast, *depth-first* (or, greedy) search replaces the initial DARP tour upon the encounter of its *first* favorable k-interchange. Again, the process is repeated until no favorable k-interchange can be found.

In both cases, the *depth* of search is equal to the number of k-interchanges executed *plus one,* since the last iteration of the procedure is spent to prove that no further improvement is possible. In breadth-first search, the number of favorable interchanges that are encountered is in general greater than the number of those that are executed. Those two numbers are the same in depth-first search (by construction). The question on how deep can a search be is at this time open. The same holds for the question on which is the structure of the set of all solutions reachable from the initial solution by a finite number of interchanges, or, which is the quickest way to obtain the best local optimum. It is conjectured that there may be pathological cases where the depth of search is arbitrarily high. Therefore it should be emphasized that in both search modes the $O(N^k)$ running time established earlier is a time bound *for any single iteration* of the search procedure (improve a given tour or prove that no improvement is possible); it is *not* a time bound for the entire procedure (find a k-optimal tour.)

Whether any one of the above two modes of search is unambiguously superior to the other, is not a straightforward question to answer. For the TSP, Lin [5] reported that depth-first search produces k-optima in substantially less time than breadth-first search. However, he did not report on whether or not depth-first k-optimal tours were consistently shorter than the corresponding breadth-first k-optimal tours (quality of the solution). Furthermore, it is not clear that Lin's conclusion necessarily holds for DARP interchanges, since the structure of the DARP is different from that of the TSP.

To shed some light on the above issue we have programmed both search modes for the $k = 2$ and $k = 3$ cases and compared them via a series of simulation runs. In all runs, N origins and N destinations are sampled from a uniform distribution on the unit square. The vehicle departs from and finally returns to the center of the square. We have examined *two* different ways to obtain an initial DARP tour:

(1) Initial tour generated *randomly*. This means that at each stop the vehicle is equally likely to visit any of the remaining origins or any of the remaining destinations whose origins have been previously visited.

(2) Initial tour obtained via Psaraftis' $O(N^2)$ Minimum Spanning Tree (MST) heuristic. This heuristic was analyzed extensively in [9] and seen to produce very good DARP tours (more on that point later).

The results from those runs appear in Tables 3 to 6 for $N = 10$, 20 and 30 (20, 40 and 60-node DARP's) and for three random seeds for each N. Symbols appearing in those tables are explained as follows:

L_1: Length of initial DARP tour,
L_k: Length of k-opt DARP tour ($k = 2$, or 3),
ITOT_k: Total number of k-interchanges considered,
IFE_k: Total number of feasible k-interchanges encountered,
IFAV_k: Total number of favorable k-interchanges encountered,
INT_k: Total number of k-interchanges executed (= Depth of search $-$ 1).

L_1 and L_k are normalized by $\sqrt{2N}$ for reasons dealing with the asymptotic behavior of L^*, the optimal DARP tour length. Stein [1] has shown that for a unit area $L^*/\sqrt{2N}$ approaches 1.02 with probability 1.0 for $N \to \infty$. More on this point later.

In all tables, numbers in parentheses refer to depth-first search. The discussion of the computer runs goes as follows:

Table 3 refers to the $k = 2$ case, when the initial DARP tour is random. An immediate observation from that table is that, in contrast to the TSP,

Table 3
Comparison of breadth-first versus depth-first search 2-opt procedures when initial DARP tour is *random* (Numbers in parentheses refer to depth-first search; for further explanation see text)

Seed	$N = 10$			$N = 20$			$N = 30$		
	#427	#762	#833	#122	#347	#502	#55	#263	#910
$L_1/\sqrt{2N}$	2.71	2.57	2.69	2.77	3.48	4.26	4.13	3.82	4.02
$L_2/\sqrt{2N}$	1.31	1.19	1.63	1.84	1.71	2.02	1.80	1.75	1.63
	(1.14)	(1.23)	(1.62)	(1.57)	(1.77)	(1.88)	(1.83)	(1.44)	(1.15)
$ITOT_2$	3040	3800	1900	14040	21060	27300	83190	79650	92040
	(3918)	(4606)	(1940)	(22436)	(21361)	(29146)	(105542)	(134175)	(159292)
IFE_2	830	1193	424	2422	4441	4840	16160	15100	18745
	(922)	(1218)	(376)	(2292)	(3142)	(4024)	(23982)	(19470)	(26660)
$IFAV_2$	135	217	87	357	550	884	2095	1962	2413
	(43)	(41)	(20)	(46)	(58)	(60)	(115)	(137)	(168)
INT_2	14	18	8	17	26	34	45	44	50
	(43)	(41)	(20)	(46)	(58)	(60)	(115)	(137)	(168)

there seems to be no evidence that depth-first search is substantially faster than breadth-first search in producing a 2-optimum. Actually, in all nine cases, depth-first search arrives at a 2-optimum after having *considered* more 2-interchanges ($ITOT_2$) and after having *executed* more 2-interchanges (INT_2) than breadth-first search. However, as far as the quality of the solution is concerned, depth-first search wins in 6 out of 9 cases.

Table 4 is similar to Table 3 with the only difference that the initial DARP tour is now produced by Psaraftis' MST heuristic [9]. Again, there is no evidence that breadth-first search is computationally more burdensome than depth-first search. Notice however that in some cases the breadth-first executes more 2-interchanges than the depth-first search (in contrast to the results of Table 3). The quality of the solution is about the same, with each of the procedures winning twice and five runs resulting in a tie.

As it would be expected, we generally obtain better 2-optima (35% shorter on the average) if we

Table 4
Comparison of breadth-first versus depth-first search 2-opt procedures when initial DARP tour is produced by Psaraftis' MST heuristic [9]. (Numbers in parentheses refer to depth-first search; for further explanation see text)

Seed	$N = 10$			$N = 20$			$N = 30$		
	#427	#762	#833	#122	#347	#502	#55	#263	#910
$L_1/\sqrt{2N}$	1.43	1.28	1.24	1.16	1.19	1.42	1.20	1.21	1.29
$L_2/\sqrt{2N}$	1.31	1.22	1.03	1.08	1.10	1.37	1.13	1.11	1.25
	(1.11)	(1.21)	(1.12)	(1.10)	(1.10)	(1.37)	(1.13)	(1.13)	(1.25)
$ITOT_2$	570	760	950	6240	5460	6240	19470	19470	14160
	(980)	(795)	(1206)	(4579)	(4336)	(4349)	(21363)	(16844)	(10466)
IFE_2	214	271	322	1291	1612	1401	4120	4306	3242
	(286)	(281)	(366)	(894)	(1036)	(783)	(3761)	(2798)	(2058)
$IFAV_2$	6	14	11	39	35	35	69	79	36
	(8)	(4)	(6)	(6)	(6)	(9)	(6)	(16)	(13)
INT_2	2	3	4	7	6	7	10	10	7
	(8)	(4)	(6)	(6)	(6)	(9)	(6)	(16)	(13)

Table 5
Comparison of breadth-first versus depth-first search 3-opt procedures when initial DARP tour is produced *randomly* (Numbers in parentheses refer to depth-first search; for further explanation see text)

Seed	$N = 10$			$N = 20$			$N = 30$		
	#427	#762	#833	#122	#347	#502	#55	#263	#910
$L_1/\sqrt{2N}$	2.71	2.57	2.69	2.77	3.48	4.26	4.13	3.82	4.02
$L_3/\sqrt{2N}$	1.08	1.02	1.12	1.12	1.14	1.36	1.10	1.05	0.97
	(1.28)	(1.02)	(1.03)	(1.07)	(1.03)	(1.27)	(1.12)	(1.04)	(1.00)
$ITOT_3$	19950	(17290	18620	287820	319800	341120	1655540	1871480	2051430
	(16311)	(16187)	(21314)	(293882)	(425883)	(529871)	(2415244)	(2334677)	(2835179)
IFE_3	6666	6446	5350	63754	77779	61564	319740	336798	334904
	(4226)	(5093)	(5333)	(49674)	(70729)	(73535)	(361388)	(295705)	(338889)
$IFAV_3$	489	708	538	4102	6031	6298	19229	19137	22917
	(46)	(34)	(50)	(123)	(139)	(158)	(263)	(226)	(262)
INT_3	14	12	13	26	29	31	45	51	56
	(46)	(34)	(50)	(123)	(139)	(158)	(263)	(226)	(262)

Table 6
Comparison of breadth-first versus depth-first search 3-opt procedures when initial DARP tour is produced by Psaraftis' MST heuristic [9] (Numbers in parentheses refer to depth-first search; for further explanation see text)

Seed	$N = 10$			$N = 20$			$N = 30$		
	#427	#762	#833	#122	#347	#502	#55	#263	#910
$L_1/\sqrt{2N}$	1.43	1.28	1.24	1.16	1.19	1.42	1.20	1.21	1.29
$L_3/\sqrt{2N}$	1.31	1.22	1.03	1.02	1.00	1.19	1.09	1.02	1.08
	(1.31)	(1.21)	(1.03)	(1.06)	(1.00)	(1.26)	(1.10)	(1.05)	(1.08)
$ITOT_3$	11970	5320	3990	181220	117260	159900	647820	647820	791780
	(8483)	(11907)	(10462)	(104420)	(156147)	(119980)	(458025)	(471457)	(653996)
IFE_3	3849	1765	1070	39740	24208	42224	124599	113778	165728
	(2338)	(3516)	(2527)	(18846)	(23091)	(21041)	(74193)	(61958)	(110424)
$IFAV_3$	31	20	11	100	125	169	162	222	269
	(8)	(10)	(8)	(14)	(21)	(25)	(18)	(29)	(37)
INT_3	8	3	2	16	10	14	17	17	21
	(8)	(10)	(8)	(14)	(21)	(25)	(18)	(29)	(37)

use the MST heuristic to provide us with an initial DARP tour rather than obtaining the latter randomly. There are of course some exceptions (see Seed #910 for $N = 30$). In any event, a comparison of Tables 3 and 4 indicates that we definitely need more calculations (about one order of magnitude more) if we start with a random DARP tour.

Tables 5 and 6 are the equivalent of Tables 3, and 4 for the $k = 3$ case. As in the $k = 2$ case, there is no evidence that breadth-first search is computationally more burdensome than depth-first search. Actually, in all cases examined, the latter executes more interchanges than the former. The quality of the solution seems about the same. However, and in contrast to the $k = 2$ case, the use of the MST heuristic to generate initial DARP tours does not seem to produce better 3-optima than those produced if we start with a random tour. A comparison of Tables 5 and 6 indicates that the quality of the 3-optimum is about the same, independently of the method to construct the initial tour. In some cases actually (depth-first search), a random initial tour seems to produce slightly better 3-optima than a MST-generated initial tour. Of course, the computational effort of a 3-opt procedure applied on a random initial tour is greater (but not much greater) than the corresponding effort when the procedure is applied to an MST-generated tour.

Another observation is that the 3-opt procedures outperform the corresponding 2-opt algorithms by producing tours that are about 30% shorter on the average if the initial tour is random and about 6% shorter on the average if the initial tour is MST-generated. A fortiori, they produce tours that are about 9–10% shorter on the average than the ones produced by Psaraftis' MST $O(N^2)$ heuristic. Of course, the 3-opt procedures are significantly more burdensome computationally than the corresponding 2-opt algorithms (about one order of magnitude more).

It should be noted here that despite the superiority of the 3-opt procedures over the corresponding 2-opt ones in terms of quality of the solution, there are individual (rare) cases where a 2-opt procedure might produce a better tour (see for instance $N = 10$, seed #427, random initial tour and depth-first search). This phenomenon could very well occur even if our 3-interchanges were not strict, as they were assumed to be.

The above observations, interesting as they might be, cannot provide by themselves conclusive answers to what are perhaps the most important questions concerning the performance of the 2-opt and 3-opt procedures:

(1) How far from the optimal solution do those procedures deviate on the average?

(2) How do they compare with other heuristics for the sample problem?

Answering those questions requires far more analysis than it might seem at first glance. For instance, trying to answer the first question by comparing the procedures directly with an *exact* algorithm (e.g. the one presented in [8]) would involve severe computational difficulties if N is greater than 9 or 10. Trying to answer the second question is also difficult, because with the exception of the heuristics of Stein [11] and Psaraftis [9], no computational experience of other DARP heuristics lending itself to comparison has been reported to date.

Fortunately, an extensive investigation of this issue was made by the author in [9], in conjunction with the $O(N^2)$ Minimum Spanning Tree (MST) heuristic for the DARP. In that paper it was shown that, for $10 \leqslant N \leqslant 50$, the MST heuristic compares well, and in some cases, better than some heuristics proposed by Stein [11] for the same problem, despite the fact that one of those heuristics is asymptotically optimal (for $N \to \infty$). In [11] Stein showed that such an asymptotically optimal algorithm produces tours for which the ratio $L/\sqrt{2N}$, for a uniform distribution on a unit area and for $N \to \infty$, converges with probability 1.0 to 1.02. Stein's work actually established the use of $L/\sqrt{2N}$ as the 'yardstick' in evaluating a DARP algorithm's performance. However, in [9] it was shown that the *convergence* of the above asymptotically optimal heuristic to an optimal behavior is obtained for *very large* values of N (of the order of 10^3 to 10^6 customers if one is to maintain computational tractability). Test runs performed in [9] indicated that, for $N = 50$, Stein's asymptotically optimal heuristic generally produces *longer* DARP tours than those produced by the MST heuristic applied to the same problem instances and for which $L/\sqrt{2N}$ is in the neighborhood of 1.20 to 1.30. For $10 \leqslant N \leqslant 50$, it was shown that the MST heuristic yields a $L/\sqrt{2N}$ of about 1.15 to 1.20, and fairly independent of N. The reader is referred to [9] for this extensive analysis, the repetition of which is beyond the scope of this paper.

Results presented earlier in this section, showed that the 2-opt and 3-opt procedures produce DARP tours which are 5–10% shorter than those produced by the MST heuristic. In addition, another series of runs for *very small* problem sizes comparing these procedures with an *exact* DARP algorithm [8] showed that both procedures performed very well, especially the 3-opt, which, for those small problems, produced the exact optimum for the majority of cases.

In light of all of the above, we believe that the results of this section improve upon the results of [9], and hence, support the following conjecture: For the range of problem sizes examined ($N \leqslant 30$), for the kind of distance metric assumed (Euclidean), and for the form of probability distribution simulated (uniform on the unit square), the 2-opt and 3-opt algorithms produce very good or near-optimal DARP tours, even if the initial DARP tours are random. Whether or not this result can be generalized to higher values of N, different distance metrics and different probability distributions is not known at the moment.

5. Final remarks

The extension of the procedures developed in the previous cases to cases where $k > 3$ is straightforward. This would involve the modification of the 'screening' procedure described earlier. The computational effort associated with the general case is $O(N^k)$ per iteration of the search procedure.

The approach developed in this paper can find applications in other areas, not necessarily related to Dial-A-Ride systems, or routing for that matter. The general problem of sequencing a set of N jobs, each of which has *two tasks* that have to be executed in a prescribed order (although not necessarily the one immediately following the other), belongs to the same category. For instance, the scheduling of computer jobs can be considered as falling into this category (e.g. one task is the compilation of the program and the other its execution).

Of interest might be the case where the k-interchanges considered are not strict, in the sense that only *some* of the k links are substituted. It is conjectured that this modification is not likely to produce dramatic DARP tour improvements.

A final direction for possible research concerns the case where the initial DARP tour is not feasible, that is, violates the origin-to-destination precedence constraints. The important open question at this point is: Can one enhance the local search by producing a k-optimal solution through intermediate tours some of whom might be infeasible?

References

[1] L.D. Bodin, B.L. Golden, M. Ball and A. Assad, The state-of-the-art of routing and scheduling of vehicles and crews, Report, School of Business and Management, University of Maryland (1981).

[2] L.D. Bodin and T.R. Sexton, The subscriber Dial-A-Ride problem: The Baltimore, Maryland benchmark, ORSA/TIMS meeting, Colorado Springs, (1980).

[3] B. Gavish and Srikanth, Mathematical formulations of the Dial-A-Ride problem, Working paper, Graduate School of Management, University of Rochester.

[4] P.C. Kanellakis and C.H. Papadimitriou, 'Local search for the asymmetric traveling salesman problem, *Operations Res. 28* (1980) 1086–1099.

[5] S. Lin, Computer solutions to the traveling salesman problem, *Bell System Tech. J. 44* (1965) 2245–2269.

[6] S. Lin and B.W. Kernighan, An effective heuristic algorithm for the traveling salesman problem, *Operations Res. 21* (1973) 498–516.

[7] C.H. Papadimitriou and K. Steiglitz, Some Examples of difficult traveling salesman problems, *Operations Res. 26* (1978) 434–443.

[8] H.N. Psaraftis, A dynamic programming solution to the single-vehicle many-to-many immediate request Dial-A-Ride problem, *Transportation Sci. 14* (1980) 130–154.

[9] H.N. Psaraftis, Analysis of an $O(N^2)$ heuristic for the single vehicle many-to-many Euclidean Dial-A-Ride problem, *Transportation Res.*, to appear.

[10] T.R. Sexton, The single vehicle many-to-many routing and scheduling problem, PhD thesis, S.U.N.Y. at Stony Brook (1979).

[11] D.M. Stein, An asymptotic, probabilistic analysis of a routing problem, *Math. Operations Res. 3* (1978) 89–101.

[12] G.G. Tharakhan and H.N. Psaraftis, An exact algorithm for the exponential disutility Dial-A-Ride problem, Working paper, MIT (1981).

[13] N.H.M. Wilson and N.J. Colvin, Computer controil of the Rochester Dial-A-Ride system, Departmcnt of Civil Engineering, MIT, Report R77-31 (1977).

[14] N.H.M. Wilson and H. Weissberg, Advanced Dial-A-Ride algorithms research project: Final report, Department of Civil Engineering, MIT, Report R76-20 (1976).