Information Processing Letters 41 (1992) 333–339
North-Holland 17 April 1992

An approximation algorithm for the general routing problem

Klaus Jansen
FB IV – Informatik, Universität Trier, 5500 Trier, Germany

Communicated by T. Lengauer
Received 18 September 1991

Abstract

Jansen, K., An approximation algorithm for the general routing problem, Information Processing Letters 41 (1992) 333–339.

In this paper a generalization of the TSP which is given by Orloff and called the general routing problem (GRP) is considered. The goal of the GRP is to find a minimum cost cycle in a graph $G=(V, E)$ which visits vertices in a required subset $V' \subset V$ exactly once and covers edges in a required subset $E' \subset E$ at least once. We generalize the known heuristic of Christofides for the TSP with triangle inequality and approximate ratio 3/2 to the GRP.

Keywords: Analysis of algorithms, combinatorial problems

1. Introduction

A difficult combinatorial optimization problem is to find an optimal route for a single vehicle on a given network. A network is given as a connected graph G consisting of a set V of vertices, a set E of edges and given positive weights (or costs, distances) $c: E \to \mathbb{R}^+$ on the edges. The problem of visiting all vertices in such a given network with minimum cost is the classical traveling salesman problem (TSP). The problem of covering all edges of a network with minimum total cost is the chinese postman problem (CPP). If only a subset E' of all edges must be traversed, we get the rural postman problem (RPP). The goal of the general routing problem (GRP), first described by Orloff [11], is to find a minimum cost cycle which visits each vertex in a required subset V' exactly once and traverses each edge in a required subset E' of the network.

Most of these problems cannot be solved in polynomial time, unless P = NP. For the theory of complexity we refer to the classical NP-bible of Garey and Johnson [7]. Only the chinese postman problem can be solved in polynomial time $O(|V|^3)$ by an algorithm by Edmonds [5]. Karp [8] has shown, that the TSP even with triangle inequality is NP-hard and the other NP-proofs for the RPP and GRP are given by Lenstra and Rinnooy Kan [10].

For these NP-hard problems different approximative algorithms are analysed. The performance of these heuristics is measured by the maximum ratio of the approximative solution value to the optimum value. The best-known polynomial algorithm for TSP with triangle inequality is due to Christofides [3] with ratio $\frac{3}{2}$. For the general TSP where the triangle inequality does not hold, an even stronger result is known. Sahni and Gonzales [12] have shown that the problem of finding a

Correspondence to: Dr. K. Jansen, FB IV – Informatik, Universität Trier, Postfach 3825, 5500 Trier, Germany. Email: jansen@dm3.uni-trier.de.

solution with value at most k times the optimum value is NP-hard for any finite k.

A similar approach as the algorithm of Christofides for the TSP can be applied to the RPP with the same ratio $\frac{3}{2}$, see [6]. In this paper we generalize the heuristic of Christofides to the GRP with triangle inequality.

Besides these results many other approximative algorithms are proposed for the solution of vehicle routing and scheduling problem. We refer for an overview and a classification of these problems and heuristics to a survey of Bodin, Golden, Assad and Ball [2].

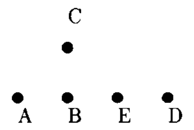

Fig. 1. Points in the plane.

2. Definitions and the TSP

Let us briefly describe the algorithm of Christofides [3] for the TSP. In the next section we can show a way to extend this method. The instance of the TSP is given by a complete graph and a cost function $c: E \to \mathbb{R}^+$ which satisfies the triangle inequality.

The goal is to find a tour, i.e. a sequence $S = v_0, v_1, \ldots, v_{|V|-1}, v_0$ with $v_i \in V$, $v_i \neq v_j$ for $0 \leqslant i < j \leqslant |V| - 1$ and with minimum cost

$$c(S) = \sum_{i=0}^{|V|-1} c_s(v_i, v_{(i+1) \bmod |V|}).$$

In the first step of the algorithm a minimum spanning tree $T = (V, \bar{E})$ will be generated. A spanning tree is a subgraph which includes all the vertices, is connected and has no circuits. A minimum spanning tree is a tree where the sum of the edge costs is as small as possible. In comparison to the TSP there are a variety of polynomial-time algorithms for the generation of a minimum spanning tree; see for example [1]. The length C_T of a minimum spanning tree for the graph obtained for the instance above is less than the length C_{opt} of the minimum tour, because an optimal TSP tour minus an edge is a spanning tree.

The second step has the goal to generate an eulerian multi-graph. An eulerian graph is a connected graph in which each vertex has even degree and a multi-graph is a graph with allowed parallel edges. For that consider the $2 \cdot k$ vertices $\bar{V} \subset V$ with odd degree in the spanning tree T and produce a minimum weighted matching for these vertices. A matching for the graph induced by $\bar{V}$ is a collection of k mutually nonadjacent edges and the weight of a matching is the sum of the costs of these edges. Any matching for $\bar{V}$ gives us k edges and converts the spanning tree into an eulerian graph, and a minimum weighted matching M is one matching with minimum weight C_M. Such a minimum weight matching can be found in time $O(|V|^2)$ with a standard algorithm of Lawler [9]. It was shown that the weight C_M of this optimal matching is at most half the length of the minimum TSP tour.

The edges of the optimal matching are added to the spanning tree and then we get an eulerian multi-graph for which we can generate an euler-tour $S = v_0, \ldots, v_m, v_0$ using depth-first search in time $O(|V|)$; see [1]. If there is one vertex $v \in V$ at least twice in the tour ($v = v_i = v_j$ with $i < j \leqslant m$), delete the multiple vertex v_j in the tour. This can be done iteratively until the tour visits only vertices exactly once. By using the triangle inequality for the edge costs the length C_{tour} of the generated TSP tour is less than the length C_{EUL} of the eulerian tour. Therefore we get the length $C_{tour} \leqslant \frac{3}{2} C_{opt}$. Cornuejols and Nemhauser [4] have shown examples that this bound $\frac{3}{2}$ for the algorithm of Christofides is tight. Currently this algorithm is the best polynomial-time approximative algorithm for the TSP with triangle inequality.

	A	B	C	D	E
A					
B	1				
C	$\sqrt{2}$	1			
D	3	2	$\sqrt{5}$		
E	2	1	$\sqrt{2}$	1	

Fig. 2. Distances between them.

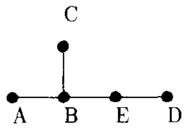

Fig. 3. Spanning tree.

As example we have five vertices, $V = \{A, B, C, D, E\}$, which lie in the plane as described by Fig. 1. The distances between these vertices are given in Fig. 2.

As tree T we get the graph in Fig. 3 and as graph with matching edges the one in Fig. 4. In the example we take the odd-degree vertices $\bar{V} = \{A, B, C, D\}$ and choose the matching $M = \{\{A, B\}, \{C, D\}\}$. As eulertour we generate A, B, C, D, E, B, A and after using short cuts we get the TSP tour A, B, C, D, E, A.

3. Heuristic for the GRP

Let us consider the general routing problem and the following graph

$$G|_{V',E'} = (\{v \mid v \in e \in E'\} \cup V', E').$$

At first we assume that this graph is connected. A GRP tour can only exists if there is no vertex $v \in V'$ with degree greater than 2 in the graph $G|_{V',E'}$. Therefore we consider the case that all vertices $v \in V'$ have degree $0 < d(v) \leqslant 2$ and that the other have degree $d(v) > 0$.

For a GRP tour we need k additional edges between vertices $\bar{V}$ with odd degree. Such an edge can also be a path with intermediate vertices in $V \setminus V'$. At first we can generate for each pair of vertices with odd degree the shortest path over these edges. This can be done by a modification of a classical shortest-path algorithm. By solving the weighted matching problem with these costs we get k different pairs of vertices with odd degree and therefore k paths with intermediate vertices in $V \setminus V'$. Such a matching must not exist; for example, consider a path of length three where both edges belong to E' and where the middle vertex belongs to V'. For the connected case we can consider the following algorithm:

Algorithm 3.1.
(1) Let $d(v)$ be the degree of the vertices in $G' = G|_{V',E'}$.
(2) If $d(v) > 2$ for one vertex, stop: a GRP tour cannot exist.
(3) Let $\bar{V}$ be the vertices of odd degree.
(4) Compute for each pair $v,w \in \bar{V}$ with $v \neq w$ the shortest path (if existing) in G with intermediate vertices in $V \setminus V'$.
(5) Solve the matching problem for $\bar{V}$ with these shortest paths as weights.
(6) If no matching exists, stop: a GRP tour cannot exist.
(7) Otherwise get k pairs of vertices and add the corresponding paths to G'.
(8) Compute an eulertour using depth-first search.

Theorem 3.2. *Let $G = (V, E)$ be a graph with weights $c: E \to \mathbb{R}^+$, let $V' \subset V$ and $E' \subset E$ and let $G|_{V',E'}$ be the connected graph as described above. Let $\bar{V}$ be the vertices of odd degree and let $\bar{G} = (\bar{V}, \bar{E})$ be a graph with an edge $\{v, w\} \in \bar{E}$ iff $(v \neq w)$ and if there is a path from v to w in G with intermediate vertices in $V \setminus V'$.*

(1) *If one vertex $v \in V'$ has degree $d(v) > 2$ or if there exists no matching in $\bar{G}$, a GRP tour cannot exist.*

(2) *If all vertices $v \in V'$ have degree $1 \leqslant d(v) \leqslant 2$ and if there is a matching in $\bar{G}$, Algorithm 3.1 generates an optimal solution.*

Proof. (1) If we have a vertex $v \in V'$ with $d(v) > 2$, we cannot generate a GRP tour which visits v exactly once. Now assume that $1 \leqslant d(v) \leqslant 2$ for each $v \in V'$. For the GRP tour we must enlarge the graph $G|_{V',E'}$ by adding edges $e \in A$ into an eulerian multi-graph G_A with degree equal to 2 for each vertex $v \in V'$. To show that a matching in $\bar{G}$ exists, take one vertex v of odd degree in $G|_{V',E'}$ and generate a path to another vertex

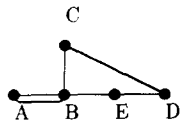

Fig. 4. Eulerian graph.

$w \neq v$ with $w \in \bar{V}$ using only edges from A. Since the vertex v has odd degree, an incident edge $e \in A$ exist. If the other endpoint v' has odd degree, we have found such a vertex. Since in the other case the vertex v' has another incident edge from A, we can iterate this argument and find a path between vertices of $\bar{V}$ using edges of A with intermediate vertices of $V \setminus V'$. Deleting this path in the graph G_A, only the endpoints of the path have odd degree. Therefore, we can construct a path between another pair of vertices in $\bar{V}$ and can iterate this argument to get a matching in $\bar{G}$.

(2) In the first part we have shown that the GRP tour can be generated by adding a matching between the vertices of $\bar{V}$. Therefore, an optimal solution is given by a minimum weighted matching in $\bar{G}$. □

This result is also an extension of the algorithm for the CPP given by Edmonds [5]. Now we look at the general case where the graph $G|_{V',E'}$ is not connected. Let us assume that the triangle inequality is satisfied. At first we must compute the connected components of $G|_{V',E'}$. This can be done by depth-first search in time $O(|E| + |V|)$.

Lemma 3.3. *Let $G = (V, E)$ be a complete graph where the distances $c: V \to \mathbb{R}^+$ satisfy the triangle inequality; let $V' \subset V$ and $E' \subset E$. Let $K_1, \ldots, K_k$ be the connected components of $G|_{V',E'}$ with $k > 1$.*

(1) *If there is one vertex $v \in V'$ with degree $d(v) > 2$ in one of the components K_i, a GRP tour cannot exist.*

(2) *If for one component K_i all vertices are in V' and have degree $d(v) = 2$, a GRP tour cannot exist.*

(3) *If both conditions are not satisfied, there is a GRP tour.*

Proof. If condition (1) or (2) is satisfied, at least one of the vertices must be visited more than once. In the third case we have that in each component there is at least one vertex $v \notin V'$ or there is at least one vertex $v \in V'$ with degree $d(v) < 2$. If the vertex has degree 1, then there is at least one other vertex v' with $v' \notin V'$ or with $d(v') = 1$. Using this fact we can enlarge $G|_{V',E'}$ into a connected graph where all vertices $v \in V'$ have degree less than or equal to 2. Since G is complete, there are edges between each pair of odd-degree vertices. Therefore, we can obtain a matching in $\bar{G}$ and also a GRP tour for G. □

We denote C_{opt} as the length of the optimal GRP tour and $C_{E'}$ as the length of the given edges in E'. Now we describe a method to generate such a GRP tour. We consider the components and compute the shortest distances between each pair of components. But we consider only links between vertices in $V \setminus V'$ or with degree $d(v) \in \{0, 1\}$. Then we generate a minimum spanning tree for the graph where the vertices are these connected components and where the costs are given by the link distances. The corresponding edges are added to the whole graph $G|_{V',E'}$. This graph is now connected and we have only new edges between vertices in $V \setminus V'$ or with old degree $d(v) \in \{0, 1\}$. In the next step we compute a minimum weighted matching for the vertices with odd degree described as for the connected case. At last we must delete the vertices in V' with degree greater than 2. This can be done by short-cuts described later. The whole algorithm can be described as follows:

Algorithm 3.4.

(1) Compute the connected components $K_1, \ldots, K_k$ of $G|_{V',E'}$.

(2) If $k = 1$, apply Algorithm 3.1.

(3) If there is a vertex $v \in V'$ with degree $d(v) > 2$, stop: a GRP tour cannot exist.

(4) If in at least one component K_i each vertex $v \in K_i$ is in V' and has degree $d(v) = 2$, stop: a GRP tour cannot exist.

(5) Let U be the set of vertices v with $v \notin V'$ or with degree $d(v) \leqslant 1$. Define a complete graph $G_k = (\{1, \ldots, k\}, E_k)$ with the cost $c(e)$ of edge $e = \{i, j\}$ with $i \neq j$ equal to the shortest link between a vertex in $K_i \cap U$ and a vertex in $K_j \cap U$.

(6) Compute a minimum spanning tree for G_k and add the corresponding edges to $G_{V',E'}$.

(7) Determine the odd-degree vertices $\bar{V}$, compute a minimum weighted matching for them

and add the edges of this matching to the graph.

(8) Generate an euler tour for the generated graph using depth-first search.

(9) Apply the short-cut Algorithm 3.6.

Lemma 3.5. (1) *The cost of the tour after step* (8) *of Algorithm* 3.4 *is at most* $\frac{3}{2}$ C_{opt}, *where* C_{opt} *is the length of an optimal GRP tour.*

(2) *After step* (8) *each vertex* $v \in V'$ *with degree* $d(v) = 2$ *in* $G|_{V',E'}$ *has the same degree and the other vertices have even degree greater than or equal to* 2.

Proof. (1) Consider the minimum spanning tree T and the edges in E'. In an optimum solution we must have additionally to the edges E' a cycle C to connect the components $K_1, \ldots, K_k$. In this cycle there is no edge with an endpoint $v \in V'$ and degree $d(v) = 2$ in $G|_{V',E'}$; otherwise the vertex is visited more than once. If we delete enough edges from the cycle, we get a spanning tree. Since the cost C_T is less than the cost of the cycle, we get $C_T + C_{E'} \leqslant C_{opt}$. Also we see that the matching has the cost $C_M \leqslant \frac{1}{2} C_{opt}$. This can be proved as follows. Let us convert a minimum GRP tour in one tour $v_0, \ldots, v_{2k-1}, v_0$ which visits only the vertices in $\bar{V}$. Using the triangle inequality the length of this tour is at most as large as the length of the GRP tour. In addition this tour gives us two matchings

$$M_j = \{\{v_{2i+j}, v_{(2i+j+1) \bmod (2k)}\} \mid 0 \leqslant i \leqslant k-1\}$$

with $j \in \{0, 1\}$. Therefore we have $C_M \leqslant \frac{1}{2}(C_{M_0} + C_{M_1}) \leqslant C_{opt}$. Therefore, we can generate an eulertour with cost at most $\frac{3}{2} C_{opt}$ which visits all vertices in V' and covers all edges in E'.

(2) Consider a vertex $v \in V'$ with degree $d(v) = 2$ in $G|_{V',E'}$. In step (6) we add no incident edge to v and therefore after (6) the degree is the same. Since the degree is even, also in step (8) no incident edge will be added. For the vertices with $d(v) = 0$ at least one incident edge will be added in step (6) and after step (7) using the matching all vertices have even degree greater than or equal to 2. □

Now we describe the short-cut procedure. Given the eulertour $S = v_0, v_1, \ldots, v_m, v_0$ we compute at first a feasible starting point. This is a vertex which must not be deleted in the tour. If $E' = \emptyset$, we have the TSP problem and can choose each vertex v_i. But if $E' \neq \emptyset$, we take a vertex in $v_i \in V \setminus V'$ or a vertex v_i with $\{v_i, v_{i+1}\} \in E'$. Using the renumbered eulertour $S' = v_i, \ldots, v_m, v_0, \ldots, v_i$, we delete vertices of V' if they appear more than once. If the old degree of a vertex $v \in V'$ was 2, we have seen that after step (8) the degree of v is still the same. Therefore such a vertex must not be deleted. But if the degree $d(v)$ was 1 before, it is possible that the degree is greater than 2 after step (8). Then we must search for the first occurrence of the corresponding incident edge of v in E' and must delete the other occurrences of v in the whole tour. If the degree $d(v)$ was 0 and after step (8) greater than 2, we must delete v after his first visit.

Algorithm 3.6. Let $S = v_0, \ldots, v_m, v_0$ be the eulertour where v_0 is a feasible starting point.

(1) Let $i = 0$.

(2) For each $v \in V$ set *visit*$(v) = 0$.

(3) If $i = m + 1$, stop.

(4) Set *visit*$(v_i) = 1$ and consider $e_i = \{v_i, v_{(i+1) \bmod (m)}\}$. If $e_i \in E'$, then set $i := i + 1$ and goto (3).

(5) Choose the smallest index $j > i$
- with $j = m + 1$ or
- with $v_j \notin V'$ or
- with $v_j \in V'$, $d(v_j) = 0$, *visit*$(v_j) = 0$ or
- with $v_j \in V'$, $d(v_j) = 1$,
  $e_j = \{v_j, v_{(j+1) \bmod (m)}\} \in E'$, *visit*$(v_j) = 0$.

(6) If $j > i + 1$, then delete the vertices $v_{i+1}, \ldots, v_{j-1}$, set $i = j$ and goto (3).

Lemma 3.7. *After the short-cut procedure all vertices* $v \in V'$ *have degree* 2.

Proof. By a case study about the degree $d(v)$ of the vertices $v \in V'$ in the graph $G|_{V',E'}$. If the degree $d(v) = 2$, we have the same degree after step (8) of Algorithm 3.4. Such a vertex will not be deleted in the short-cut procedure. If the degree $d(v)$ was 0 and if v occurs more than

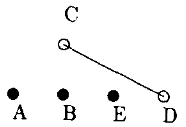

Fig. 5. GRP-instance.

once, after the first visit of v we set $visit(v) = 1$ and then v will be deleted in the rest tour. In the last case we have $d(v) = 1$. If v occurs in the tour with two incident edges from $E \setminus E'$, this occurrence will be deleted. After we have found the first occurrence with an incident edge in E', we set $visit(v) = 1$ and after that each occurrence of v in the rest tour will be deleted. □

Together we have found the following result:

Theorem 3.8. *Let $G = (V, E)$ be a complete graph where the distances $c: E \to \mathbb{R}^+$ satisfy the triangle inequality, and let $V' \subset V$ and $E' \subset E$. Then Algorithm* 3.4 *generates a tour which visits each vertex of V' exactly once and covers each edge $e \in E'$ at least once. The length of the generated tour is at most $\frac{3}{2}C_{opt}$ is the length of an optimum GRP tour.* □

Now we want to discuss an example for this problem. We take the same vertices and distances as in the example for the TSP. But we take as edges in E' the drawn line and as V' the dark labelled vertices in Fig. 5. In the first step the edges $\{A, B\}$, $\{B, C\}$ and $\{B, E\}$ with length 3 for the spanning tree are generated to connect the components; see Fig. 6.

After that we solve the matching problem with the odd-degree vertices $\overline{V} = \{A, B, D, E\}$. The best solution takes the edges $\{A, B\}$, $\{D, E\}$. The generated graph can be seen in Fig. 7. An eulertour is for example given by C, D, E, B, A, B, C. This tour must be shortened, because the vertex

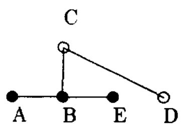

Fig. 6. Spanning tree.

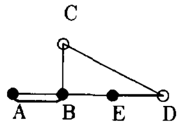

Fig. 7. Eulerian graph.

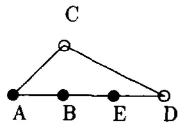

Fig. 8. A GRP tour.

$B \in V'$ appeares twice. The reduced tour is C, D, E, B, A, C which is also given as graph in Fig. 8.

We note that the optimal tour can also be generated in polynomial time, if we have only a constant number k of connected components. One possibility is to take all spanning trees with $k - 1$ edges to connect these components. Since k is constant, there are only polynomial many such trees. But we must take only trees so that the resulting graph has only vertices in V' with degree 1 or 2. With a matching we can generate an eulerian graph so that all vertices in V' have degree 2 and the other even degree. The minimum cost eulerian graph is chosen and we get an eulertour which satisfies also the condition that all vertices in V' are visited exactly once.

References

[1] A.V. Aho, J.E. Hopcroft and J.D. Ullman, *The Design and Analysis of Computer Algorithms* (Addison-Wesley, Reading, MA, 1974).

[2] L. Bodin, B.L. Golden, A. Assad and M. Ball, The state of the art in the routing and scheduling of vehicles and crews, *Comput. Oper. Res.* **10** (1983) 63–212.

[3] N. Christofides, Worst case analysis of a new heuristic for the traveling salesman problem, Management Science Research Rept. 388, Carnegie-Mellon University, Pittsburgh, PA, 1976.

[4] G. Cornuejols and G.L. Nemhauser, Tight bounds for Christofides Traveling Salesman Heuristic, *Math. Programming* **14** (1978) 116–121.

[5] J. Edmonds, The chinese postman problem, *Oper. Res.* **13** (1965) B73–B77.

[6] G.N. Fredericksón, Approximation algorithms for some postman problems, *J. ACM* **26** (1979) 538–554.

[7] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, San Francisco, CA, 1979).

[8] R.M. Karp, Reducibility among combinatorial problems, in: R.E. Miller and J.W. Thatcher, eds., *Complexity of Computer Computations* (Plenum Press, New York, 1972) 85–103.

[9] E.L. Lawler, *Combinatorial Optimization: Networks and Matroids* (Holt, Rinehart and Winston, New York, 1976).

[10] J.K. Lenstra and A.H.G. Rinnooy Kan, On general routing problems, *Networks* **6** (1976) 273–280.

[11] C.S. Orloff, A fundamental problem in vehicle routing, *Networks* **4** (1974) 35–64.

[12] S. Sahni and T. Gonzales, P-complete approximation problems, *J. ACM* **23** (1976) 555–565.