UNIVERSITEIT ANTWERPEN

RUCA
UNIVERSITAIR CENTRUM ANTWERPEN

Departement Toegepaste Economische Wetenschappen

A DESCENT HEURISTIC FOR THE VEHICLE ROUTING PROBLEM BASED ON CHAIN PARTITIONING

Gerrit K. JANSSENS

WORKING PAPER

95/10

UNIVERSITY OF ANTWERP (RUCA) - Department of Applied Economics

Middelheimlaan 1 - 2020 Antwerpen - België - Tel. +32(0)3/218.07.72 - Fax +32(0)3/218.06.52

A DESCENT HEURISTIC FOR THE VEHICLE ROUTING PROBLEM BASED ON CHAIN PARTITIONING

Gerrit K. JANSSENS

95/10

A DESCENT HEURISTIC FOR THE VEHICLE ROUTING PROBLEM BASED ON CHAIN PARTITIONING

Gerrit K. JANSSENS
Computer Science and Operations Management
University of Antwerp (RUCA)
Middelheimlaan 1
B-2020 Antwerp, Belgium

0. Introduction

Optimal solution methods for the vehicle routing and scheduling problem (VRSP) are of practical use only for small problems (see e.g. LAPORTE and NOBERT [1987], MAGNANTI [1981] and CHRISTOFIDES et al. [1981]). Scientific literature has focused on heuristics to solve problems of moderate and of large size. Further improvements to the obtained sub-optimum can be reached by the use of local or global search methods.

Local search methods have been used successfully in the Travelling Salesman Problem (TSP). Methods, generally known as the *r*-opt methods exchange *r* edges in the graph representation of a TSP (e.g. LIN and KERNIGHAN [1973]). In a feasible tour *r* edges are exchanged for *r* edges not in that solution as long as the result remains a tour and the length of that tour is less than the length of the previous tour (GOLDEN and STEWART [1985]). The value of *r* is chosen to be two or three. From each move a feasible tour is generated. If one extends this idea towards exchange of arcs for the VRSP, as OSMAN [1993] does, an additional problem appears.

Arcs exchanges between two distinct routes can cause unfeasibility in the generated solutions. The vehicle capacity constraint in one of the routes can be violated. This means that the neighbourhood search terrain is very rugged because it contains many infeasible moves, while it is claimed that the terrain should be smooth.

As a solution to this indesirable effect, several solutions can be proposed. Firstly, a check on the feasibility can be done after each move. This extensive checking increases however the run-time of the algorithm. By using a *range limiter* the neighbourhood size is diminished, i.e. the number of moves is limited by a problem characteristic. By this the number of infeasible moves which are generated should be limited. The idea originiates from CASOTTO et al. [1987] and is used for the VRSP in JANSSENS and VAN BREEDAM [1995].

A second approach includes the overcapacity as a penalty in the objective function. Penalty terms are added to the cost to make unfeasible solutions less attractive. This method surely smooths the neighbourhood terrain but endangers the convergence.

As a third method, we propose to enforce the feasibility by modifying the neighbourhood moving strategy. Neighbourhood solutions are not generated from the solution but from an inverse transformation of the solution. If from an inverse transformation always a feasible solution of the original problem can be found, the problem of infeasible solutions is avoided. This technique is applied by ALFA et al. [1991] for the VRSP. They first obtain a giant tour. The tour is split in a left to right direction in such a way that a new route starts whenever the capacity of the previous tour is exceeded. Their results turn out to be good for very small problems (which also could be solved optimally) but are poor for larger problems. Our method fits in this class, but is based on totally different principles. It not only generates a feasible solution at all times but also guarantees a smooth neighbourhood. The drawback of the procedure is its increased complexity of the move increasing the run-time.

1. A chain partition-formulation of the VRSP

The basic step in the heuristic is an *optimal chain partition* of a tree. This step will be dealt with in detail before embedding it into the heuristic.

The formulation of the VRSP is based on the definition of the *chain* of nodes of a tree, as introduced by MISRA and TARJAN [1975]. Let T = (V,E) be a tree with vertex set V and edge set E. A rooted tree (T,**r**) is a tree with a distinguished vertex **r** called the root. If **v** and **w** are vertices in a

rooted tree (T,r) and if v is contained in the path from r to w, it is said that v is an *ancestor* of w and w is a *descendant* of v
(notation: v => w).

If v → w and {v,w} is an edge of T, it is said that v is the father of w and w is the son of v (notation: v → w).

As a graphical example can serve the tree in fig. 1.

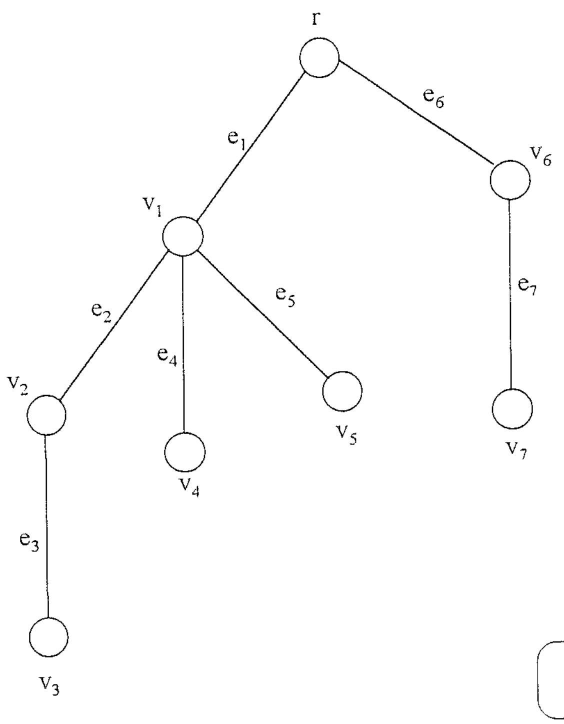

Figure 1

$V = \{r, v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$

$E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$

Father-son relationships: $v_1 \rightarrow v_2$, $v_1 \rightarrow v_4$, $v_6 \rightarrow v_7$, $r \rightarrow v_6$, amongst others.

Ancestor-descendant relationships: $r \Rightarrow v_2$, $v_1 \Rightarrow v_3$, $r \Rightarrow v_7$, $v_6 \Rightarrow v_7$, amongst others.

Definition (MISRA and TARJAN [1975]) : A chain partition of a rooted tree (T,r) is a collection E' of edges such that, for any vertex v, E' contains at most one edge {v,w} with $v \to w$.

This means that, as for any vertex in a chain only one descendant can appear in the same chain, a chain is a set of edges which define a simple path in T. (notation: P_i, where i is the index identifying the path). By this:

$$E' = \bigcup_{i=1}^{k} P_i,$$

where the edges of P_i have no vertices in common with those of P_j $(i \neq j)$.

Making abstraction of the depot location, i.e. the place where all routes in the schedule start and end, a relation exists between a tree and its chain partition, and a set of routes. The set of vertices corresponds to the set of customers to be visited. The set of edges corresponds to a possible set of two-customer sequences, i.e. if {v,w} is an edge in T, then customer v is visited just before or just after customer w. A chain relates to two VRP-concepts: (1) the edges of a chain determine the sequence of customers to be visited in one tour, or (2) the vertices included in the chain determine the set of customers in the tour without specifying a sequence. In both cases the chain is interpreted as a vehicle route.

Given the definition of a chain, MISRA and TARJAN [1975] define the *optimal chain partition problem*. A non-negative cost $c_1(v)$ is associated with each vertex $v \in V$, and there exists a maximum cost

$$m \geq \max_{v \in V} c_1(v)$$

An unrestricted real-valued cost $c_2(v,w)$ is associated with each edge $(v,w) \in E$. The optimal chain partition problem is the problem of finding a chain partition

$$C = \bigcup_{i=1}^{k} P_i$$

of maximum total edge cost, satisfying

$$\sum_{v \text{ on } P_i} c_1(v) \leq m, \qquad (i = 1, \dots, k)$$

The relation between the optimal chain partition problem and the VRSP can be extended as follows. Let $c_1(\mathbf{v})$ be the demand at each customer's site **v**, being less than the vehicle's capacity **m**. The constraint

$$\sum_{v \text{ on } P_i} c_1(v) \leq m$$

assures that the total demand to be collected on route i does not exceed the vehicle capacity. The edge cost corresponds to the distance (or time) traveled. By this, the total edge cost corresponds nearly to the total distance traveled.

Some small changes have to be made in order to use the optimal chain partition (OCP) algorithm to generate feasible VRSP solutions:

(1) the OCP-algorithm obtains a chain partition with maximal cost. The VRP requires a minimal total distance or time traveled. If d_{vw} is the distance between customers **v** and **w**, the edge cost used in the OCP-algorithm should be

$$c_2(v,w) = M - d_{vw}$$

where M is a large number,

$$M \rangle\rangle \max_{v,w \in E} c_2(v,w)$$

(2) Only one chain contains the root node **r** , while as being interpreted as a route starting from the depot all chains should do. The solution to this problem depends on the interpretation of the chain. If the chain is interpreted as a *sequence* of customers to be visited in a route, the least cost edge is added connecting the root node with one of both chain nodes having degree one. If the chain is interpreted as a *unordered set* of customers in a route without specification of a sequence, the root

node is added to the set for further processing in a sequencing algorithm. In the following we will assume the latter interpretation.

The algorithm can be summarized as follows:

Given a graph G=(V,E), a 'demand' function $\mathbf{D} : V \rightarrow \Re$ and a 'distance' function $d : E \rightarrow \Re$:

1. Construct a spanning tree
2. Relabel the nodes with a postorder notation
3. Execute the OCP-algorithm
4. Complete the solution to obtain a feasible set of routes
5. Optimise the sequences within the routes.

A short explanation of the steps in the procedure follows.

The *first step* aims to obtain a rooted tree on which the OCP-algorithm can work. Any rooted tree is a feasible starting point. In terms of the objective function a minimum spanning tree could be a reasonable starting point. In terms of an estimate of number of routes (or some lower bound on it) a spanning tree with a fixed degree constraint on the root node can be an alternative. Our implementation uses a minimum spanning tree.

The *second step* is required by the OCP-algorithm in order to have a numbering of vertices such that each vertex has a smaller number than its father. The pseudocode written in the MISRA and TARJAN [1975] article is the basis for a recursive procedure.

The *third step* is the heart of the procedure. The procedure outlined in fig. 1 of the MISRA and TARJAN [1975] article is followed. The procedure is adapted to cope with the type of objective function, i.e. minimal instead of maximal total edge cost.

In the *fourth step* the root node is added to the sets of vertices in the chain, in which it is not yet included. A distance matrix is prepared to obtain an optimal sequence within the set of customers of each route.

The *fifth step* uses an exact Travelling Salesman Problem-algorithm to obtain a minimum distance objective.

2. Embedding the optimal chain partition into a descent method

A descent method is an iterative technique exploring a set of solutions denoted by *X*, by repeatedly making moves from one solution *s* to another *s'* in the neighbourhood *N(s)* of *s*.

Let a descent method be generally formulated as follows (GLOVER et al. [1993]):

begin

Choose an initial solution *s* in *X*

stop := false

repeat

Generate a sample V^* of solutions in *N(s)*

Find a best *s'* in V^*

if $f(s') \geq f(s)$ then stop := true

else $s := s'$

until stop

end.

The initial solution *s* is found using the five step procedure described in the previous section starting from a symmetric distance matrix, e.g. generated from a coordinates vector of *n* customers.

The neighbourhood of a solution is defined in terms of the tree on which the OCP-algorithm works. In the tree, generating the current solution, alternately each of the *n-1* edges is forbidden. A neighbourhood tree is generated by putting the distance equal to infinity an by reproducing a minimum spanning tree. All trees obtained in this way are investigated, i.e. $V^* = N(s)$. If an improvement is found further edges are forbidden without restoring the original distances for the set of earlier forbidden edges. The procedure stops if no improvement is found or if no spanning tree can be generated. The latter happens when the graph of non-forbidden edges, from which the spanning tree has to be generated, is not connected.

3. An example problem

The chain partitioning procedure and its embedding into the descent heuristic will be explained using a playtoy example. In the book by EILON et al. [1971, chapter 9] on distribution management ten vehicle routing problems are investigated. For the case of illustration their problem no. 3 is used. This problem is a 21-customer problem with one depot taken from an article by GASKELL [1967]. The original problem includes a drop time at each customer's site and a limit on the maximal route distance. These additional constraints are neglected in this example problem. The objective function is to minimise the total distance travelled, using straight line distances and taking into consideration a finite vehicle capacity.

In a first step a minimum spanning tree is set up using the pairs of arcs (in increasing order of straight line distance; each pair has smallest customer index first) : (3,4), (5,7), (14,16), (7,9), (0,14), (6,8), (1,2), (8,10), (0,12), (17,20), (12,15), (18,20), (19,21), (17,21), (16,17), (13,16), (2,5), (9,10), (8,11), (10,12) and (3,6). The resulting spanning tree is shown in figure 2.

The post order numbering of the set of nodes is (in each pair mentioned the first element indicates the original number, the second the post order number): (0,21), (1,14), (2,15), (3,10), (4,9), (5,16), (6,11), (7,17), (8,13), (9,18), (10,19), (11,12), (12,20), (13,5), (14,7), (15,8), (16,6), (17,4), (18,0), (19,2), (20,1) and (21,3). The tree with post order numbering is shown in figure 3.

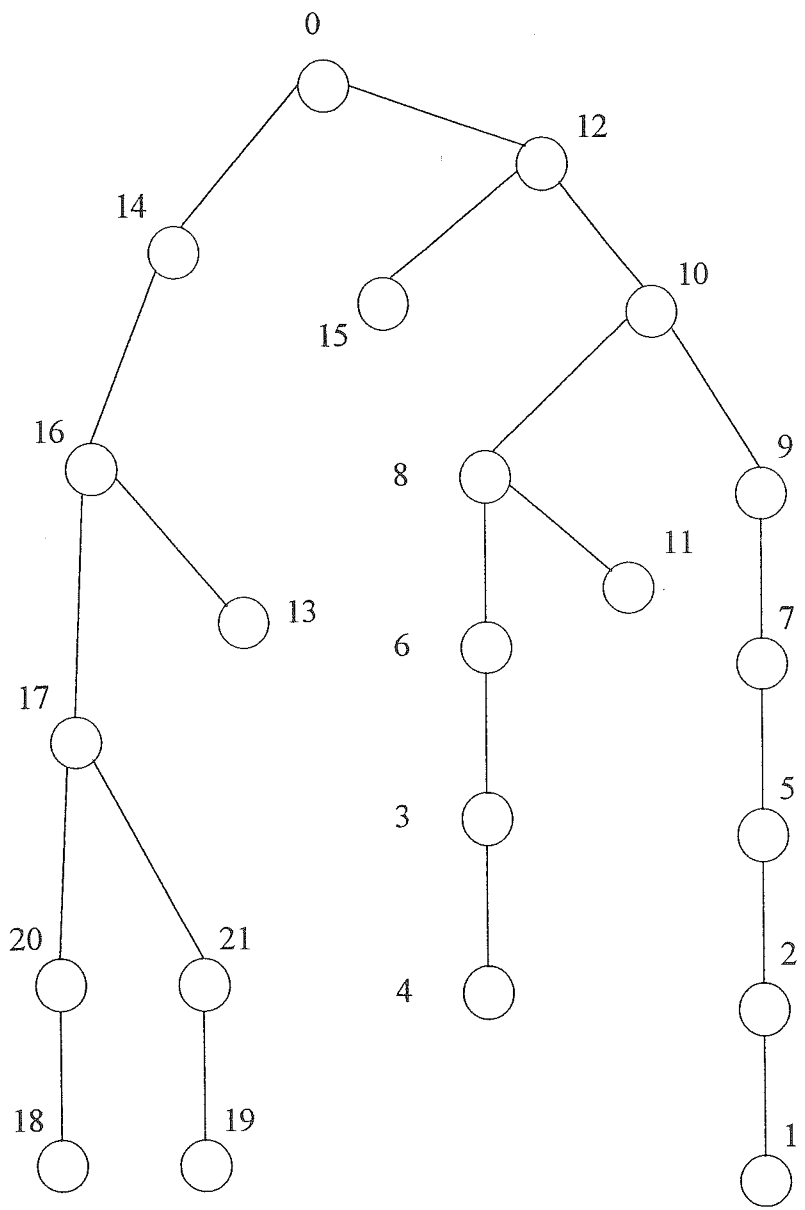

Figure 2

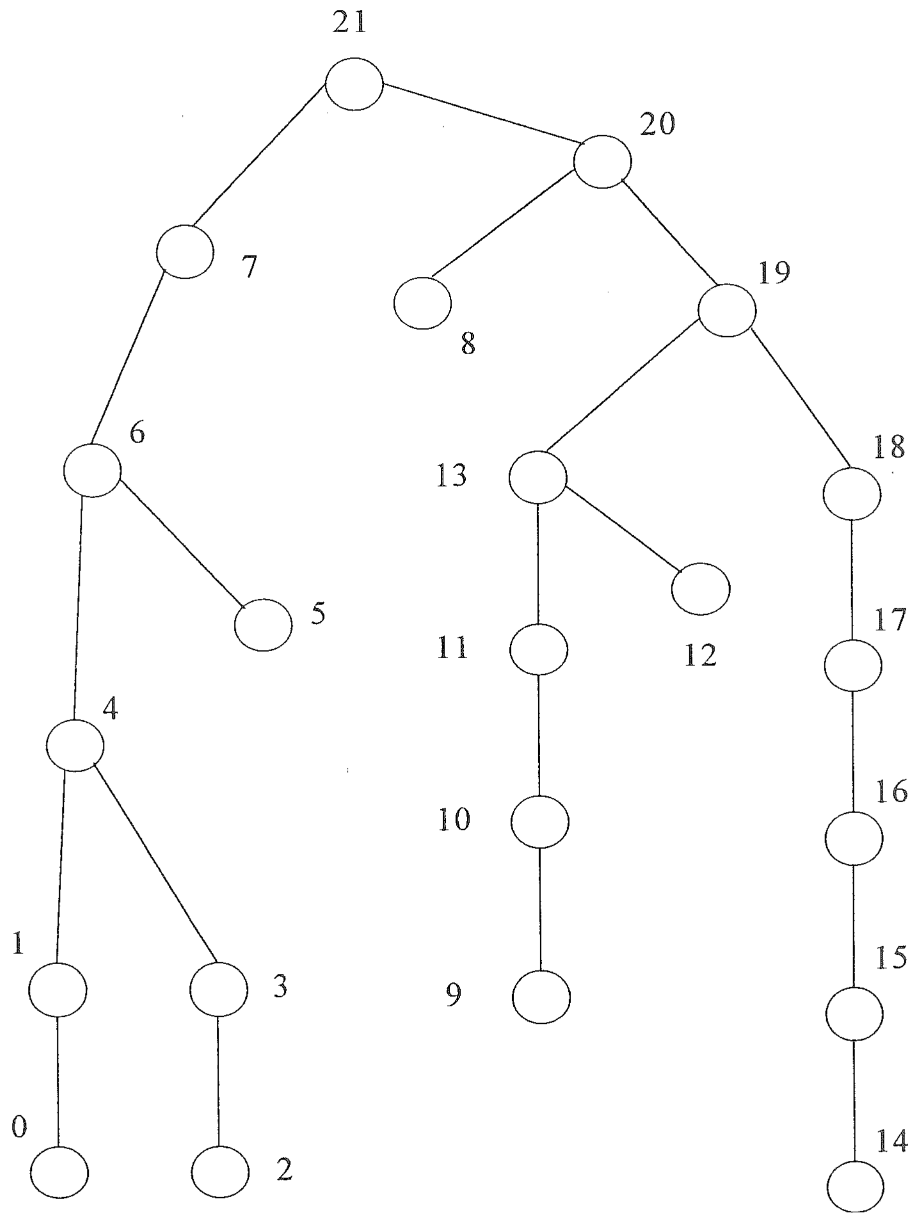

Figure 3

The chains generated by the OCP-algorithm (ordered in increasing post order number and connected to the root node) together with their TSP solution are:

(0,1,4,21)	74
(2,3,21)	77
(9,10,21)	86
(12,21)	46
(14,15,16,17,18,21)	112
(11,13,19,21)	63
(8,20,21)	43
(5,6,7,21)	42
Total	543

The chains generated are shown in figure 4. The edges not in the chains are crossed.
The initial number of tours is equal to eight.

4. Conclusion

A correspondence between the vehicle routing problem and the optimal chain partitioning of a tree is formulated. Its main advantage is that neighbourhood search always generates a feasible solution for the vehicle routing problem. Neighbourhood search methods (as descent methods, tabu search and simulated annealing) for problems with constraints have low efficiency because they generate infeasible solutions. The method is illustrated on a small example. Further experimentation is required to evaluate the method's capability of generating high quality solutions.

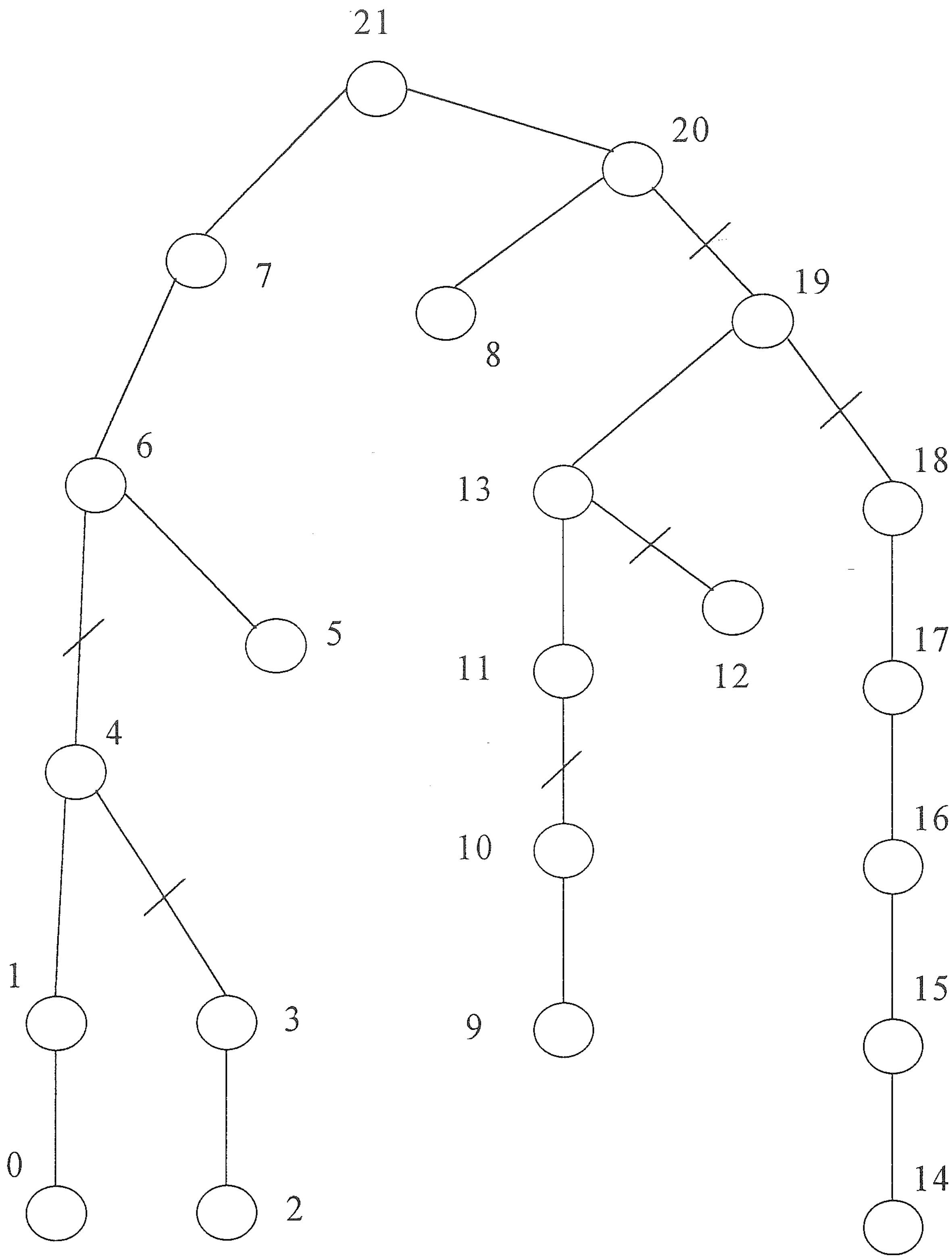

Figure 4

References

ALFA, A.S., S.S. HERAGU and M. CHEN, 1991, A 3-opt based simulated annealing algorithm for vehicle routing problems, *Computers and Industrial Engineering*, vol. 21, pp. 635-639

CASOTTO, A., F. ROMEO and A. SANGIOVANNI-VINCENTELLI, 1987, A parallel simulated annealing algorithm for the placement of macro-cells, *IEEE Transactions on Computer-Aided Design*, vol. CAD-6, no. 5, pp. 838-847

CHRISTOFIDES, N., A. MINGOZZI and P. TOTH, 1981, Exact algorithms for the vehicle routing problem based on spanning tree and shortest path relaxations, *Mathematical Programming*, vol. 20, pp. 255-282

EILON, S., C.D.T. WATSON-GANDY and N. CHRISTOFIDES, 1971, *Distribution management: mathematical modelling and practical analysis*, Griffin, London, 240 pp.

GASKELL, T.J., 1967, Bases for vehicle fleet scheduling, *Operational Research Quarterly*, vol. 18, pp. 281-295

GLOVER, F., E. TAILLARD and D. DE WERRA, 1993, A user's guide to tabu search, *Annals of Operations Research*, vol. 41, pp. 3-28

GOLDEN, B.L. and W.R. STEWART, 1985, Empirical analysis of heuristics, in LAWLER, E.L., J.K. LENSTRA, A.H.G. RINNOOY KAN and D.B. SCHMOYS (eds.), *The Travelling Salesman Problem*, J. Wiley, pp. 207-249

JANSSENS, G.K. and A. VAN BREEDAM, 1995, A Simulated Annealing Postprocessor for the Vehicle Routing Problem, in V.J. RAYWARD-SMITH (ed.), *Applications of Modern Heuristic Methods*, A. Waller Ltd., Henley-on-Thames (U.K.), pp. 175-192

LAPORTE, G. and Y. NOBERT, 1987, Exact algorithms for the vehicle routing problem, *Annals of Discrete Mathematics*, vol. 31, pp. 147-184

LIN, S. and B.W. KERNIGHAN, 1973, An effective heuristic algorithm for the vehicle routing problem, *Operations Research*, vol. 21, pp. 498-516

MAGNANTI, T.L., 1981, Combinatorial optimization and vehicle fleet planning: perspectives and prospects, *Networks*, vol. 11, pp. 197-213

MISRA, J. and R.E. TARJAN, 1975, Optimal chain partitions of trees, *Information Processing Letters*, vol. 4, no. 1, pp. 24-26

OSMAN, I., 1993, Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem, *Annals of Operations Research*, vol. 41, pp. 421-451