ELSEVIER

European Journal of Operational Research 102 (1997) 157–175

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

Theory and Methodolgy

# Estimating the Held-Karp lower bound for the geometric TSP

Christine L. Valenzuela [a,*], Antonia J. Jones [b]

[a] *School of Computing and Mathematics, University of Teesside, Borough Road, Middlesbrough, Cleveland, TS1 3BA, United Kingdom*
[b] *Department of Computer Science, University of Wales, Cardiff, PO BOX 916, Cardiff CF2 4YN, United Kingdom*

Received 23 January 1995; accepted 13 June 1996

**Abstract**

The Held-Karp lower bound (HK) provides a very good problem-specific estimate of optimal tour length for the travelling salesman problem (TSP). This measure, which is relatively quick and easy to compute, has enormous practical value when evaluating the quality of near optimal solutions for large problems where the true optima are not known. Although HK can be evaluated exactly by Linear Programming techniques, code for doing this efficiently for problems larger than a few hundred cities is not readily available or easy to produce. In addition Linear Programming implementations (even efficient ones) do not scale well and rapidly become impractical for problems with many thousands of cities. In this study we concentrate on the iterative estimation approach proposed by Held and Karp in their original papers. Our paper provides implementation details for two iterative schemes which both use the subgraph speed-up technique.

We begin by surveying the important theoretical issues which underpin the original iterative approach of Held and Karp (now known as subgradient optimisation). We then present some detailed practical guidelines for the evaluation of HK for large geometric TSP problems, and also provide some empirical evidence demonstrating the robustness of the iterative schemes we have used. Finally our estimation of the Goemans and Bertsimas constant provides an independent confirmation of the value published recently by Johnson, McGeoch and Rothberg and simultaneously supports the claim that our approach does indeed produce reliable results. © 1997 Elsevier Science B.V.

*Keywords:* Travelling salesman; Held-Karp lower bound; Minimum 1-tree; Lagrangian relaxation; Subgradient optimization

## 1. Introduction

The Held-Karp lower bound (HK) is the solution to the linear programming (LP) relaxation of the standard integer programming formulation of the travelling salesman problem (TSP) (Reinelt, 1994; Johnson, 1996). Unfortunately LP code for evaluating HK for problems larger than a few hundred cities is not readily available or easy to produce. In addition Linear Programming implementations (even efficient ones) do not scale well and rapidly become impractical for problems with many thousands of cities. For the reason outlined above we concentrate here on the iterative estimation approach proposed by Held and Karp in their original papers.

The iterative approach pioneered by Held and Karp (1970, 1971) uses a technique called Lagrangian relaxation to produce a sequence of connected graphs which increasingly resemble tours. This technique, based on the notion of a 1-tree has been used extensively as a basis for many successful branch and bound procedures to solve the travelling salesman problem (TSP) Balas and Toth (1985). The

* E-mail: christine@tees.ac.uk.


*PII* S0377-2217(96)00214-7

approach, now generally known in the literature as subgradient optimization (Held, 1974), bounds the TSP from below by solving its Lagrangian (pseudo) dual. Branch and bound methods based on subgradient optimisation proceed in two stages. The first stage, called the initial ascent, computes a lower bound to the TSP. The second stage incorporates a branch and bound procedure into the relaxation process to produce feasible solutions to the TSP via a general ascent. The product of the initial ascent has subsequently become known in the literature as the Held-Karp lower bound (or strictly speaking an estimate of it).

Gerhard Reinelt (Reinelt, 1994) pays attention to the computation of various lower bounds for the TSP. His results clearly establish that Lagrangian Relaxation based on 1-trees produces the best results of all the iterative schemes he tried. Reinelt's approach to estimating HK combines different features obtained from several key studies with some ideas of his own. Our study differs from (and complements) Reinelt's in several ways:

- We look at two approaches to evaluating the bound, both of which are different to Reinelt's.
- We do not deviate significantly from the techniques as they appear in the original papers.
- We consider random Euclidean instances as well as problems from TSPLIB (a library of TSP problems which includes many with pathological distributions of cities (Reinelt, 1995)).

Lagrangian relaxation methods based on 1-trees evaluate HK by iteratively perturbing a set of weights allocated to the vertices of a complete graph. These weights augment the edge costs and the process attempts to force all the vertex degrees of a minimum 1-tree to a value of 2, where a minimum 1-tree is a minimum spanning tree (MST) on the vertices 2, $3, \ldots, n$ together with the two lowest cost edges incident with city 1. In other words the algorithms produce weighted minimum 1-trees which increasingly resemble tours. Although practical implementations of this scheme usually fall short of evaluating the *true* value of HK, suitable applications of the iterative process usually produce very good estimates as can be seen in the results section of the present paper.

The Held-Karp lower bound, which is relatively quick and easy to compute using the above techniques, has enormous practical value when evaluating the quality of near optimal solutions for large problems where the true optima are not known. Many real applications of the TSP exist with as many as 10,000 or even 100,000 cities (example problems from x-ray crystallography and printed circuit board drilling problems are available in TSPLIB (Reinelt, 1995)). Unfortunately the largest non-trivial TSP instance to be solved to optimality involves only 7,397 cities, and the computation took many CPU-years (Applegate, 1995).

The HK is never less than 2/3 of the cost of the optimum tour when the triangle inequality is satisfied (Wolsey, 1980; Shmoys and Williamson, 1990). This worse-case analysis, however, fails to capture the efficiency of the bound in practice. Various authors quote the results of computational experiments which demonstrate the gap to be considerably less than 1% for problems up to several hundred cities (Christofides, 1979; Volgenant and Jonker, 1982). The Held-Karp lower bound is on average within 0.8% of the optimum for random Euclidean instances with many thousands of cities (Johnson, 1990, 1996).

Goemans and Bertsimas (1991) give a probabilistic analysis of the behaviour of the Held-Karp lower bound. In particular they prove that for uniformly identically independently distributed points in a unit square there exists a constant $c_2 > 0$ such that

$$\lim_{n \to \infty} \frac{\mathrm{HK}(I_n)}{\sqrt{n}} = c_2, \tag{1}$$

almost surely, where $\mathrm{HK}(I_n)$ is the Held-Karp lower bound for problem instance $I_n$.

In a recent paper Johnson, McGeogh and Rothberg have estimated $c_2$ to be $0.70805 \pm 0.00007$. Our estimate of $c_2$ is extremely close to this figure, even though it is based on significantly less data. The close agreement of these independent estimates simultaneously supports the figure obtained by Johnson and his team and lends credibility to the bounds computed by our own computer programs.

Having established a requirement for a reliable lower bound, a number of questions arises as a consequence:

- How can it be implemented?
- Which iteration and step length formulae are most appropriate?

- How can the iteration sequence length be established?
- What are the execution time issues?
- How reliable an estimate is it in practice, especially as $n$ gets large?

The last of these questions is significantly more difficult than the other four, since the answer depends on information concerning the expected value of the minimal tour length (in particular the value of the constant $c_1$ described in the next section).

Questions like these arose as a consequence of our work on Evolutionary Divide And Conquer (EDAC), in which we use genetic algorithms to produce approximate solutions to TSP problems of several thousand cities (Valenzuela, 1994, 1995a,b). The TSP problems themselves were generated as uniformly distributed random points on a square in the Euclidean plane. The areas of these squares were adjusted in relation to $n$, the problem size, so as to produce expected minimum tour lengths of 100 units, based on Stein's constant (see below). As we tuned the EDAC algorithms to produce increasingly better solutions, however, it soon became apparent that Stein's estimate of the constant $c_1$ was much too large, and the need for reliable estimates of the optima became paramount. The shortage of implementation details for HK in the literature at the time (early 1993) together with the lack of published experimental work on the calculation of the Held-Karp lower bound for problems of more than a few hundred cities motivated the work of this paper.

## 2. Geometric TSP problems in the unit square region

For a uniformly random distribution of cities the classic work by Beardwood et al. (1959) (BHH) obtains an asymptotic best possible upper bound for the minimum tour length for large $n$. Let $\{X_i\}$, $1 \le i < \infty$, be independent random variables uniformly distributed over the unit square, and let $L_n$ denote the shortest closed path which connects all the elements of $\{X_1, \ldots, X_n\}$. In the case of the unit square they proved, for example, that there is a constant $c_1 > 0$ such that, with probability 1,

$$\lim_{n\to\infty} L_n n^{-1/2} = c_1. \tag{2}$$

In general $c_1$ depends on the geometry of the region considered.

One can use the estimate provided by the BHH theorem in the following form: the expected length $L_n^*$ of a minimal tour for an $n$-city problem, in which the cities are uniformly distributed in a square region of the Euclidean plane, is given by

$$L_n^* \approx c_1\sqrt{n}\,. \tag{3}$$

The constant $c_1$ was originally estimated by Beardwood, Halton and Hammersley to be about 0.75. Later Stein (1977) produced a figure of 0.765 for $c_1$. It seems clear from our experiments (and those carried out by others) that both of these estimates, which were derived empirically from relatively small problems, are in fact rather too large (this fact has been noted by Johnson (1996) who estimates $c_1$ to be 0.7124).

The Goemans and Bertsimas (1991) result is closely related to the BHH theorem, and they show that $c_2 \le c_1$. Our own experiments suggest a value $c_2 \approx 0.708$ (see Table 9). Estimating $c_1$ empirically as $n$ becomes large is much harder and is not attempted here.

## 3. The Held-Karp lower bound

The Held-Karp lower bound is evaluated as a 1-tree relaxation, where a 1-tree on an $n$ city problem is a connected graph with vertices 1, 2,..., $n$ consisting of a tree on the vertices 2, 3,..., $n$ together with two edges incident with city 1.

Evaluation of a Held-Karp lower bound requires the computation of a sequence of "minimum 1-trees", where a minimum 1-tree is a Minimum Spanning Tree (MST) on the vertices 2, 3,..., $n$ together with the two lowest cost edges incident with city 1.

A tour is simply a 1-tree in which each vertex has degree 2. If a minimum 1-tree is a tour, then it is a tour of minimum cost. A minimum 1-tree is illustrated in Fig. 1.

We attach to the $i$th city an arbitrary weight $\pi_i$. Let $e_{ij}$ be the length of the edge joining the $i$th and $j$th city of the geometric TSP, i.e. the edge costs of

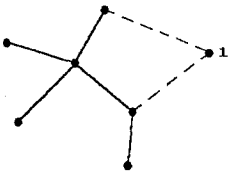


Fig. 1. A minimum 1-tree.

the original problem. Suppose we now perturb the edge costs according to the transformation

$$c_{ij} = e_{ij} + \pi_i + \pi_j, \tag{4}$$

and compute a second minimum 1-tree with respect to the new edge costs $c_{ij}$. The new minimum 1-tree will, in general, differ from the first minimum 1-tree. However, the ordering of tour lengths with respect to this transformation is not affected, since the transformation of edge lengths adds $2\Sigma\pi_i$ to every tour length. We write $L(c_{ij}, T)$ for the cost of a 1-tree or tour $T$ with respect to the edge costs $c_{ij}$.

Let $U$ be the set of 1-trees with the given vertices and $V$ be the set of tours. Then $V \subseteq U$, because a tour is a 1-tree in which every vertex has degree 2. Hence

$$\min_{T\in U} L(c_{ij}, T) \le \min_{T\in V} L(c_{ij}, T). \tag{5}$$

Since $c_{ij} = e_{ij} + \pi_i + \pi_j$ we have

$$L(c_{ij}, T) = L(e_{ij}, T) + \sum_{i=1}^{n} \pi_i d_i^T, \tag{6}$$

where $d_i^T$ is the degree of vertex $i$ in the 1-tree or tour $T$. If $T$ is a tour then $d_i^T = 2$ for $1 \le i \le n$ and then

$$L(c_{ij}, T) = L(e_{ij}, T) + \sum_{i=1}^{n} \pi_i 2 \tag{7}$$

If the RHS of Eq. (5) is attained for a minimal tour $T^*$ we have for any $\pi = (\pi_1, ..., \pi_n)$

$$\min_{T\in U} L(c_{ij}, T) \le L(e_{ij}, T^*) + \sum_{i=1}^{n} \pi_i 2. \tag{8}$$

Now write $c^* = L(e_{ij}, T^*)$ for the optimal tour length of the original problem, and $c_T = L(e_{ij}, T)$ for the cost of a 1-tree with respect to the $e_{ij}$. Then together Eq. (7) and Eq. (8) yield, for any $\pi$

$$\min_{T\in U} \left\{ c_T + \sum_{i=1}^{n} \pi_i d_i^T \right\} \le c^* + \sum_{i=1}^{n} \pi_i 2. \tag{9}$$

Hence for any $\pi$

$$\min_{T\in U} \left\{ c_T + \sum_{i=1}^{n} \pi_i (d_i^T - 2) \right\} \le c^*. \tag{10}$$

Write

$$w(\pi) = \min_{T\in U} \left\{ c_T + \sum_{i=1}^{n} \pi_i (d_i^T - 2) \right\}. \tag{11}$$

Then each $w(\pi)$ is a lower bound for $c^*$ and we can define the Held-Karp lower bound as

$$\mathrm{HK} = \max_{\pi} w(\pi). \tag{12}$$

In general HK is not equal to the minimum tour length but it is usually very close.

Now given $\pi$ we can easily find the corresponding minimum 1-tree using Kruskal's algorithm (Horowitz and Sahni, 1978). Suppose the minimum in Eq. (11) is attained at a 1-tree $T = T(\pi)$. Let $v_T = (d_1^T - 2, ..., d_n^T - 2)$ then

$$w(\pi) = c_{T(\pi)} + \pi \cdot v_{T(\pi)}. \tag{13}$$

Held and Karp provided a theoretical framework which supports an iterative improvement scheme for $w(\pi)$. If $\pi$ becomes $\overline{\pi}$ then

$$c_{T(\pi)} + \overline{\pi}\, v_{T(\pi)} \ge \min_{T\in U} \{ c_T + \overline{\pi} \cdot v_T \} = w(\overline{\pi}), \tag{14}$$

and if $w(\overline{\pi}) \ge w(\pi)$ then subtracting Eq. (13) from Eq. (14) we obtain

$$\overline{\pi} \cdot v_{T(\pi)} - \pi \cdot v_{T(\pi)} \ge w(\overline{\pi}) - w(\pi) \ge 0. \tag{15}$$

This establishes

**Lemma 1.** *Let $\overline{\pi}$ and $\pi$ be such that $w(\overline{\pi}) \ge w(\pi)$. Then*

$$(\overline{\pi} - \pi) \cdot v_{T(\pi)} \ge w(\overline{\pi}) - w(\pi) \ge 0. \tag{16}$$

Hence the hyperplane through $\pi$ having $v_{T(\pi)}$ as normal determines a closed halfplane containing all points $\overline{\pi}$ having $w(\overline{\pi}) \ge w(\pi)$. As Held and Karp observed, this halfspace includes any point where $w(.)$ assumes its maximum value, and a sufficiently small step produces a point closer than $\pi$ to any such maximum point. These observations form the basis of subgradient optimization; although we do not know the exact direction for gradient ascent we do know a direction which moves into the correct halfspace, viz. $v_{T(\pi)}$. It remains to determine suitable

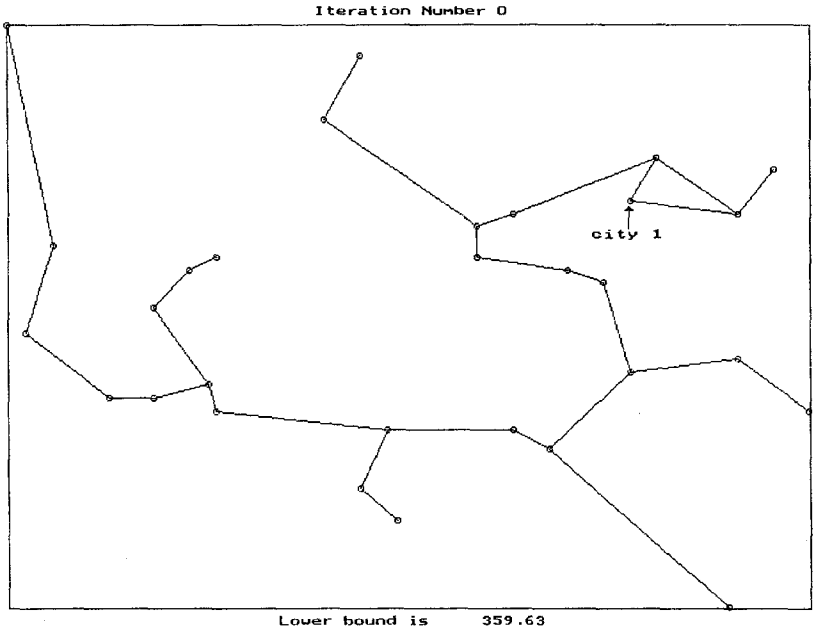


Fig. 2. Initial weighted 1-tree for a 30 city problem.

step sizes. However, it is important to note that the objective function values $w(\pi)$ are not monotonic as $\pi$ approaches a value $\pi^*$ in the optimal set.

Lagrangian relaxation on the minimum 1-tree involves the iterative improvement of the $\pi$ vector, producing weighted minimum 1-trees which increasingly resemble tours, having more vertices of degree 2 as the process progresses. A successful scheme is

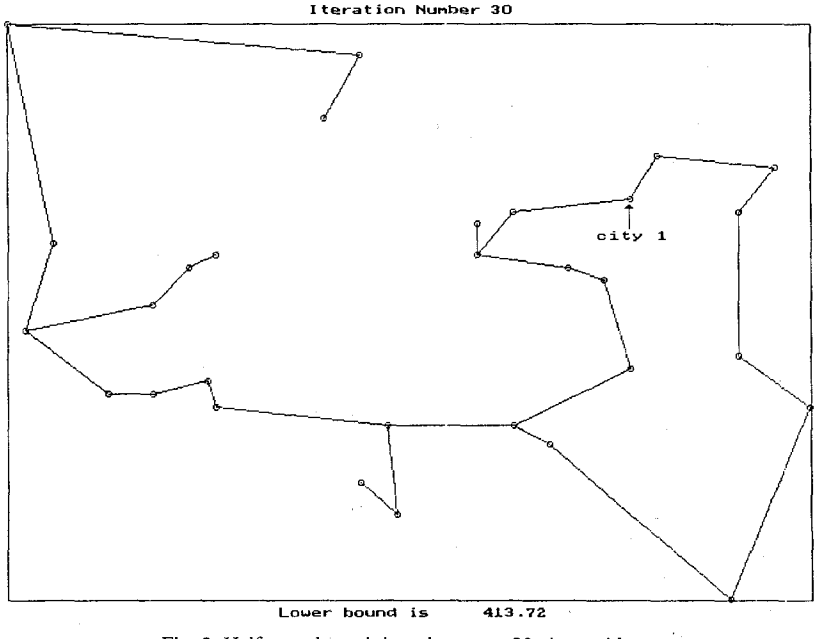


Fig. 3. Half way through iterations on a 30 city problem.

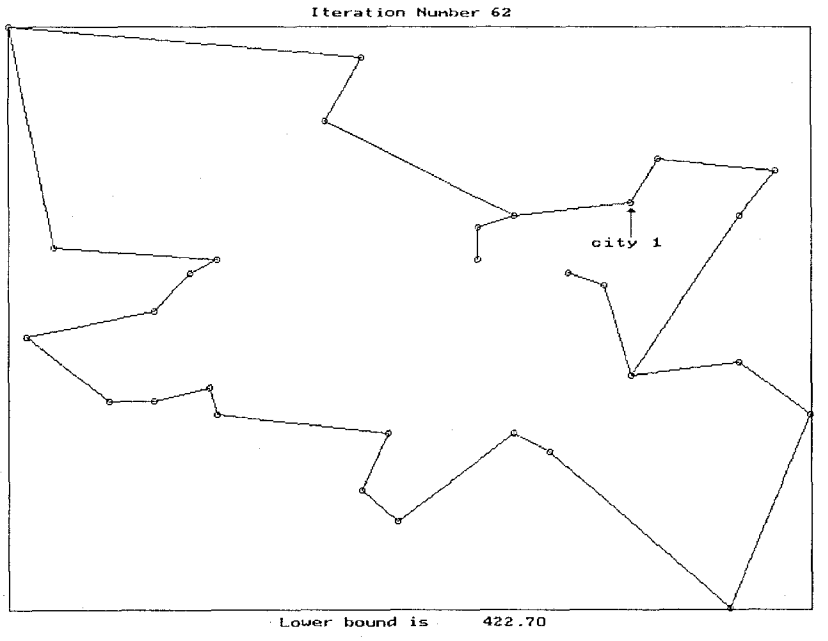


Fig. 4. Final iteration on a 30 city problem.

illustrated for a 30 city problem in Fig. 2, Fig. 3 and Fig. 4.

In practice a tour of minimum cost is rarely attained by Lagrangian relaxation alone, and there is usually a gap between the Held-Karp lower bound and the length of the optimum tour. Attempts to close this duality gap (and thus solve the TSP problems) can be made by using one of the branch and bound algorithms which employ Lagrangian relaxation. Like all exact methods applied to NP-hard/complete problems, however, the required execution time for branch and bound algorithms on travelling salesman problems grows in an exponential fashion with increasing $n$, making branch and bound generally unsuitable for problems larger than a few hundred cities. Fortunately the evaluation of the Held-Karp lower bound via subgradient optimization does not have such a large computational overhead. A single MST evaluation has complexity $O(n^2)$ but, of course, the overall complexity depends on the iterative sequence length.

## 4. Iteration formulae and step length

Held and Karp's ascent method is based on updating formula for the multipliers, $\pi^{(m)} \to \pi^{(m+1)}$, where $m$ is a sequence number,

$$\pi_i^{(m+1)} = \pi_i^{(m)} + t^{(m)}\left(d_i^{(m)} - 2\right), \quad \text{for } 1 \le i \le n. \tag{17}$$

where $d_i^{(m)} = d_i^T$, with $T = T(\pi^{(m)})$, at the $m$th iteration and $t^{(m)}$ is the "step length". For a particular vertex, $i$, the associated cost, $\pi_i$, is increased if $d_i > 2$, and decreased if $d_i < 2$. Such adjustments to the cost vector, $\pi$, result in changes in the transformed edge costs, $c_{ij} + \pi_i + \pi_j$, decreasing or increasing their chances of being included in the next minimum 1-tree to be evaluated. Thus the iteration attempts to force all the vertices to $d_i = 2$.

Some theoretical results are now outlined. These establish convergence of the ascent scheme and suggest suitable step sizes for the algorithm.

The most general result (Polyak, 1967) guarantees that $w(\pi)$ will converge to max $w(\pi)$ under the sufficient conditions

$$t^{(m)} \to 0, \quad \sum_{m=1}^{\infty} t^{(m)} = \infty. \tag{18}$$

Held and Karp (1971) state and prove three lemmas which provide the theoretical foundations upon which specific calculation formulae can be devised for $t^{(m)}$. The first lemma, which we have already outlined, established the rationale for the iteration. The second lemma indicates the limits on appropriate step sizes:

**Lemma 2.** *If* $w(\overline{\pi}) > w(\pi)$ *and*

$$0 < t < \frac{2(w(\overline{\pi}) - w(\pi))}{\|v_{T(\pi)}\|^2}, \tag{19}$$

*where* $\| \|$ *denotes the Euclidean norm, then*

$$\|\overline{\pi} - (\pi + tv_{T(\pi)})\| < \|\overline{\pi} - \pi\|. \tag{20}$$

Thus if $t$ is in the indicated range, then the point $\pi + tv_{T(\pi)}$ is closer to $\overline{\pi}$ than $\pi$.

The third lemma proved by Held and Karp, further guides the choice of $t$ values. Let $P(\overline{w})$ denote the polyhedron of feasible solutions to $\overline{w} \le c_T + \pi \cdot v_{T(\pi)}$ then

**Lemma 3.** *We assume that* $\overline{w} < \max\ w(\pi)$. *Let* $\{\pi^{(m)}\}$ *be a sequence of points obtained by the process of* Eq. (17), *where*

$$t^{(m)} = \frac{\lambda_m(\overline{w} - w(\pi^{(m)}))}{\sum_{i=1}^{n} (d_i^{(m)} - 2)^2}, \tag{21}$$

*then*

1. *If, for some* $\epsilon$, $0 < \epsilon < \lambda_m \le 2$ *for all* $m$, *then* $\{\pi^{(m)}\}$ *either includes a point* $\pi^{(l)} \in P(\overline{w})$, *or converges to a point on the boundary of* $P(\overline{w})$.
2. *If* $\lambda_m = 2$ *for all* $m$, *then the sequence,* $\{\pi^{(m)}\}$ *always includes a point* $\pi^{(l)} \in P(\overline{w})$.

Note that if a value of $w(\pi^{(m)}) > \overline{w}$ is obtained then $t$ becomes negative. As a result of this Eq. (17) will temporarily adjust the weights, $\pi$, in the wrong direction. However, Lemma 3 guarantees convergence even for cases such as these as long as the $\lambda_m$ are in the prescribed range.

Obviously the best choice for $\overline{w}$ in Eq. (21) would be max $w(\pi)$, ensuring convergence either to max $w(\pi)$ or to a point on the boundary. Of course the exact value of max $w(\pi)$ is unknown, since it is precisely this value which we wish to determine. Therefore we must use either an underestimate or an overestimate in an iterative approximation scheme based on this equation.

Although many variations on the above iterative schemes have been suggested in the literature (see Balas and Toth, 1985 for a survey) we concentrate here on just two of these, one is very firmly based on Eq. (21) and the other is not. We believe that the similarity in the results obtained using these two different schemes (see below) illustrates the robustness of the general approach.

*Method 1.* Held (1974) suggest using an overestimate in place of the underestimate $\overline{w}$, because a good value for the latter is not easy to find. If we denote the overestimate by $U$ then Eq. (21) becomes

$$t^{(m)} = \frac{\lambda_m(U - w(\pi^{(m)}))}{\sum_{i=1}^{n} (d_i^{(m)} - 2)^2}, \tag{22}$$

with $0 < \lambda_m \le 2$. They use a scheme which sets $\lambda_m = 2$ for $2n$ iterations (where $n$ is the problem size), and then successively halves both the value of $\lambda_m$ and the number of iterations until the number of iterations reaches some small threshold value $z$. The value $\lambda_m$ is then halved every $z$ iterations until $t^{(m)}$ is sufficiently small.

If an overestimate of max $w(\pi)$ is used, then Polyak's condition $t^{(m)} \to 0$ in Eq. (18) cannot be guaranteed unless a sequence of $\lambda_m$ which tends towards zero is used. Thus their scheme ensures that $t^{(m)} \to 0$, but the procedure violates the other condition $\Sigma t^{(m)} = \infty$ of Eq. (18), hence it is possible to converge to a point not in the optimal set (although they report that this almost never happened).

Held, Wolfe and Crowder use Eq. (17) for $\pi^{(m)}$ with $\pi^{(0)} = (0, 0, ..., 0)$.

*Method 2.* Volgenant and Jonker do not base their evaluation scheme for $t^{(m)}$ on Eq. (21). Instead they

define $t^{(m)}$ as a positive decreasing scalar with the following properties:

1. $t^{(m)} - 2t^{(m-1)} + t^{(m-2)} =$ constant (second order differences constant),
2. $t^{(M)} = 0$, where $M$ is the length of the sequence,
3. $t^{(1)} - t^{(2)} = 3(t^{(M-1)} - t^{(M)})$, (where 3 is quoted by Volgenant and Jonker as a constant obtained empirically).

Although Volgenant and Jonker do not describe the details, using this information the linear recurrence relation can be solved to obtain $t^{(m)}$ in terms of $t^{(1)}$, $m$ and $M$ only. The explicit solution is actually required for implementation and we therefore give a derivation in Appendix 1. The required formula that finally emerges is

$$t^{(m)} = (m-1)\left(\frac{2M-5}{2(M-1)}\right)t^{(1)} - (m-2)t^{(1)} + \frac{1}{2}\frac{(m-1)(m-2)}{(M-1)(M-2)}t^{(1)}. \tag{23}$$

Volgenant and Jonker suggest a starting value of $t^{(1)} = 0.01L(e_{ij}, T)$, with the subsequent updating of $t^{(1)}$ as better values become available. In our experiments we used values of $t^{(1)}$ proportional to $L(e_{ij}, T)/n$, where $n$ is the number of cities in the TSP. Thus we related our multipliers to average edge lengths. After some experimentation we settled on

$$t^{(1)} = \frac{1}{2n}L(e_{ij}, T). \tag{24}$$

We updated $t^{(1)}$ each time a better value for $w(\pi)$ and an associated $T = T(\pi)$, was obtained.

Volgenant and Jonker use a modified updating formula for $\pi^{(m)}$

$$\pi_i^{(m+1)} = \begin{cases} \pi_i^{(m)}, & \text{if } d_i^{(m)} = 2, \\ \pi_i^{(m)} + 0.6t^{(m)}(d_i^{(m)} - 2) \\ \quad + 0.4t^{(m)}(d_i^{(m-1)} - 2), & \text{otherwise.} \end{cases} \tag{25}$$

They comment that the addition of a term in $(d_i^{(m-1)} - 2)$ has a damping effect on the observed degree oscillations during the progress of the iteration scheme. In Eq. (25) $\pi^{(0)} = (0, 0, ..., 0)$.

In their account Volgenant and Jonker set the maximum number of steps, $M$, for the Held-Karp iteration to a quadratic polynomial in $n$. The experimental results reported herein suggest that acceptable estimates can be obtained using a number of steps sublinearly related to $n$. Although it is difficult to relate the techniques of Volgenant and Jonker directly to the convergence conditions of Polyak in Eq. (18), our experiments demonstrate their utility in practice.

## 5. 1-Tree evaluations

The calculation of the Held-Karp lower bound is dominated by $M$ minimum spanning tree (MST) evaluations, where $M$ is the sequence length. For a complete graph each MST calculation has time complexity $O(n^2)$. According to Johnson (1988), it is possible to get a good result for random points in the Euclidean plane by considering a subgraph containing edges to the 20 nearest neighbours instead of the complete graph. Using Kruskal's algorithm (Horowitz and Sahni, 1978) it is then possible to considerably reduce the time spent evaluating MSTs. A good implementation of Kruskal's algorithm has time complexity $O(e \log e)$, where $e$ denotes the number of edges in the subgraph. As $e = 20n$ for a single MST with 20 nearest neighbours, the time complexity for MST evaluation is reduced from $O(n^2)$ to $O(n \log n)$. The final value for the Held-Karp lower bound is obtained from a single $O(n^2)$ 1-tree evaluation on the complete graph using the $\pi$ vector for the best $w(\pi)$ found in the iteration scheme. This final step is necessary because all the theory is based on the notion of a complete graph.

Our experiments also indicate that 20 near neighbours are enough for such TSP problems. When applying subgraph techniques to problems obtained from TSPLIB (a public domain source of many TSP problems), however, 20 near neighbours for each city frequently proved insufficient to give a good estimate and sometimes this approach failed even to produce a subgraph that was connected (Reinelt, 1994; Johnson, 1995a). The difficulties with TSPLIB problems proved to be at their most acute when cities tended to be concentrated in "clusters".

### 5.1. The procedure Held-Karp(Coords, Nbhd)

The generic algorithm used for the experiments described in this paper is given in Algorithm 1.

Procedure Held-Karp (Coords, Nbhd)

{Coords holds the city coordinates. Nbhd[$i$][] records the $x$ nearest neighbours to city $i$. $e_{ij}$ are the edge lengths of the original problem.}

begin

initialize city weights, $\pi^{(1)} = (0, 0, ..., 0)$,
initialize subgraph Held-Karp lower bound (based on 20 nearest neighbours), HK(subgraph) = 0,

for $m = 1$ to $M$ {$M$ is the sequence length},
evaluate subgraph edge-weights $c_{ij}^{(m)} = e_{ij} + \pi_i^{(m)} + \pi_j^{(m)}$, where $2 \le i \le n$, $j \in$ Nbhd[$i$],
use Kruskal's algorithm find MST (say $S$) on above subgraph if it exists. Quit procedure if no MST.
compute $c_{1j}^{(m)} = e_{1j} + \pi_1^{(m)} + \pi_j^{(m)}$ (for all $j \in$ Nbhd[1]) {N.B. $\pi_1^{(m)} = 0$ always},
find the two least $c_{1j}^{(m)}$ over all $j \in$ Nbhd[1] (say $c_{1a}$, $c_{1b}$).
add the edges {1, $a$}, {1, $b$} to the MST $S = S^{(m)}$ to obtain a graph ($T = T^{(m)}$ say),
add the costs of these edges to obtain $L(c_{ij}^{(m)}, T)$ {the cost of the minimum 1-tree $T$ wrt the augmented edge costs $c_{ij}^{(m)}$},
record vertex degrees $d^{(m)}$ of 1-tree $T$,
compute $w(\pi) = L(c_{ij}^{(m)}, T) - 2\Sigma\pi_i^{(m)}$,
if $w(\pi) >$ HK(subgraph) then HK(subgraph) = $w(\pi)$,
compute $t^{(m)}$ {according to Eq. (22) (Method 1) or Eqs. (23) and (24) (Method 2), respectively},
update $\pi_i^{(m+1)}$ {according to Eq. (17) or Eq. (25), respectively},
endfor.

{Final step}
evaluate complete graph edge-weights $c_{ij} = e_{ij} + \pi_i^{(M+1)} + \pi_j^{(M+1)}$, where $1 \le i < j \le n$,
use Kruskal's algorithm find MST (say $S$) on complete graph,
compute $c_{1j} = e_{1j} + \pi_1^{(M+1)} + \pi_j^{(M+1)}$, for all $2 \le j \le n$ {N.B. $\pi_1 = 0$ always},
find the two least $c_{1j}$ over all $2 \le j \le n$ (say $c_{1a}$, $c_{1b}$).
add the edges {1, $a$}, {1, $b$} to the MST $S$ to obtain a graph ($T$ say),
add the costs of these edges to obtain $L(c_{ij}, T)$ {the cost of the minimum 1-tree T wrt the augmented edge costs $c_{ij}$},
$w(\pi) = L(c_{ij}, T) - 2\Sigma\pi_i^{(M+1)} =$ HK(completegraph),
return[HK(subgraph), HK(completegraph)]
end.

Algorithm 1. Pseudo-code description of the genetic Held-Karp procedure.

## 6. Experimental results for HK

In our experiments we use the triple pseudo-random number generator of Wichmann and Hill (1982) (period about $3 \times 10^{13}$) to produce all our instances of random points in the unit square. For the random points problems 64 bit reals are used to store coordinate pairs and the distance, $d$, between a pair of cities with coordinates $(x_1, y_1)$ and $(x_2, y_2)$ is computed as:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}, \tag{26}$$

i.e. normal Euclidean distance. For the instances from TSPLIB, the distance formulae specified in the library itself have been used.

### 6.1. How many near neighbours should be stored?

We used the Volgenant-Jonker formula for experiments to determine how many near neighbours of each city should be stored for random points in the unit square. The idea is to minimize the number of edges required in the subgraph for the main iteration scheme, whilst still getting a good estimate of the Held-Karp lower bound. Ideally the best value obtained from the iteration sequence on the subgraph should equal the final value from the $O(n^2)$ 1-tree calculation.

We experimented with storing 8, 10, 15, 20 and 25 near neighbours for each city in various problems, using short iteration sequences of $M = 100$. The results (for single examples of each problem size) are given in Table 1.

The results presented in Table 1 indicate that 20 near neighbours are probably sufficient for random points in the unit square, certainly for problem sizes up to $n = 10{,}000$.

For problems from TSPLIB experiments using subgraphs constructed from edges to the 20 nearest neighbours out of each city frequently produced poor results, and with several problems (for example dsj1000, fl3795 and p654) the resulting subgraph was not even connected. Closer examination reveals that many of the problems causing difficulties have pathological distributions of cities, typically in clusters.

Better results for TSPLIB instances were eventually obtained, firstly by extending the number of edges out of each city in the subgraph to more near neighbours, and secondly by ensuring that the corresponding lists of near neighbours for each city included representatives from each of the four surrounding quadrants (upper left, lower left, upper right and lower right), as long as cities were present in each quadrant. The idea for including cities from surrounding quadrants was first suggested by Johnson and McGeoch Johnson (1995a). In our implementation surrounding cities with vertical or horizontal coordinates identical to the reference city were not included as quadrant representatives, and to en-

Table 1
Comparison with the final MST when the number of nearest neighbours is varied

| Problem size | | Values obtained for Held-Karp lower bound, using $x$ neighbours | | | | |
|---|---|---|---|---|---|---|
| | | $x = 8$ | $x = 10$ | $x = 15$ | $x = 20$ | $x = 25$ |
| 100 | HK(subgraph) | 0.7371 | 0.7371 | 0.7371 | 0.7371 | 0.7371 |
| | HK(completegraph) | 0.7371 | 0.7371 | 0.7371 | 0.7371 | 0.7371 |
| 200 | HK(subgraph) | 0.7569 | 0.7570 | 0.7570 | 0.7570 | 0.7570 |
| | HK(completegraph) | 0.7569 | 0.7570 | 0.7570 | 0.7570 | 0.7570 |
| 500 | HK(subgraph) | 0.7346 | 0.7339 | 0.7339 | 0.7338 | 0.7337 |
| | HK(completegraph) | 0.7313 | 0.7333 | 0.7333 | 0.7337 | 0.7337 |
| 1000 | HK(subgraph) | 0.7281 | 0.7281 | 0.7279 | 0.7279 | 0.7279 |
| | HK(completegraph) | 0.7271 | 0.7274 | 0.7279 | 0.7279 | 0.7279 |
| 2000 | HK(subgraph) | 0.7211 | 0.7211 | 0.7211 | 0.7211 | 0.7211 |
| | HK(completegraph) | 0.7206 | 0.7209 | 0.7211 | 0.7211 | 0.7211 |
| 5000 | HK(subgraph) | 0.7157 | 0.7157 | 0.7157 | 0.7157 | 0.7157 |
| | HK(completegraph) | 0.7153 | 0.7156 | 0.7157 | 0.7157 | 0.7157 |
| 10000 | HK(subgraph) | 0.7147 | 0.7146 | 0.7146 | 0.7146 | 0.7146 |
| | HK(completegraph) | 0.7142 | 0.7145 | 0.7146 | 0.7146 | 0.7146 |

sure that such cities were not excluded from near neighbours lists, we only reserved part of each list for quadrant representatives.

We ran some experiments with neighbours lists 30 cities long and 5 places reserved on each list for cities in each of the four surrounding quadrants, when such cities were present ($5 \times 4 = 20$ cities (maximum), leaving 10 places for closest cities wherever they might be). Then we ran some more experiments with 40 neighbours and 7 places reserved for each quadrant, and finally we tried 50 near neighbours and 10 places for each quadrant. Trials using 40 near neighbours gave better results for some problems than those with 30 near neighbours, but extending to 50 near neighbours did not seem to improve the situation any further. We therefore settled on 40 near neighbours with 7 places reserved for each quadrant for subsequent experiments on TSPLIB problems.

### 6.2. *Which iteration formula?*

We implemented two iterative schemes for evaluating the Held-Karp lower bound, and most of the experiments described in this section were designed to determine which of these two approaches are the best. In addition, however, we also include some results which indicate how our implementations compare to Reinelt's Lagrangian relaxation for 1-trees (Reinelt, 1994).

In our first set of experiments we compare Held, Wolfe and Crowder's formula (HWC) with that of Volgenant and Jonker's (VJ), for various sized uniform Euclidean TSPs in the unit square.

For the HWC method $U$, the upper bound, was obtained from the best value produced by repeated runs of a 3-Opt algorithm. In order to obtain good upper bounds, the 3-Opt algorithm was executed on populations of nearest neighbour tours (Johnson, 1990).

The number of iterations used in each experiment was based on the geometric formula used by Held, Wolfe and Crowder. Polyak (19)69 has shown that if Eq. (21) is used with the underestimate $\overline{w}$ replaced by max $w(\pi)$, the unknown maximum, then geometric convergence is obtained.

We experimented with HWC approach (Method 1), using the HWC evaluation formula setting $z = 10$, and halting the iteration scheme when $\lambda = 0.000002$. We determined the threshold using $\lambda$ rather than $t$ because it was more convenient. It simplified the task of maintaining comparability with the work of HWC, i.e. setting up similar sequence lengths for similar sized problems. The HWC experiments were run first and the sequence lengths were noted and the same values used for the VJ experiments. A summary of the results for 8 different instances in the range 100 to 20000 cities appears in Table 2.

The results of Table 2 indicate that the two iterative schemes, due to Held, Wolfe, Crowder (Method

Table 2
Summary results for random problems $HK/\sqrt{n}$ for $100 \leq n \leq 20{,}000$

| Problem size | Total number of iterations | $U$ = Upper bound used in HWC formula [1] (Method 1) | HK lower bound from HWC formula (Method 1) | HK lower bound from VJ formula (Method 2) |
|---|---|---|---|---|
| 100 | 493 | 0.7398 | 0.7372 | 0.7372 |
| 200 | 886 | 0.7694 | 0.7577 | 0.7577 |
| 500 | 2090 | 0.7492 | 0.7345 | 0.7345 |
| 1000 | 4082 | 0.7548 | 0.7284 | 0.7284 |
| 2000 | 8074 | 0.7469 | 0.7214 | 0.7214 |
| 5000 | 20084 | 0.7495 | 0.7160 | 0.7160 |
| 10000 | 40074 | 0.7547 | 0.7148 | 0.7148 |
| 20000 [2] | 80064 | 0.7527 | 0.7111 | 0.7111 |

[1] The upper bound $U$ is obtained from 3-Opt experiments, as explained in the text.
[2] The final MST was not done on the 20000 city problem.

Table 3
Comparing the HWC and VJ formulae on selected problems from TSPLIB. (Italic indicates better performance)

| Problem | Size | Total iterations | HK for HCW | % below true HK for HWc | HK for VJ | % below true HK for VJ |
|---|---|---|---|---|---|---|
| lin318 | 318 | 1393 | 41889 | 0 | 41889 | 0 |
| pcb442 | 442 | 1847 | 50484 | 0.032 | *50495* | *0.0099* |
| d2103 | 2103 | 8497 | *79269* | *0.048* | 79266 | 0.052 |
| dsj1000 | 1000 | 4082 | *18414750* | *0.71* | 18373294 | 0.94 |
| fl3795 | 3795 | 15249 | 27912 | 1.99 | *27961* | *1.82* |
| fnl4461 | 4461 | 17920 | 181568 | 0.0011 | *181569* | *0.00055* |
| pcb3038 | 3038 | 12202 | 136582 | 0.0044 | *136587* | *0.00073* |
| pr2392 | 2392 | 9661 | *373490* | *0* | 373489 | 0.00027 |
| rl5934 | 5934 | 23782 | 548461 | 0.0018 | *548466* | *0.00091* |

1), and Volgenant, Jonker (Method 2), respectively, yield almost identical results for the sequence lengths used.

Table 3 extends the above comparisons to examples selected from TSPLIB. The upper bounds used for the HWC formula for the TSPLIB instances was obtained by adding 6% to the best upper bound known for each particular problem. (Best upper bounds recorded for TSPLIB instances are especially good, often optimal. In general it would be unwise to assume the availability of upper bounds of this quality). Table 3 does not produce conclusive evidence favouring one formula or the other for TSPLIB instances. However, our chosen formula for future experiments is the Volgenant-Jonker formula because the need to estimate an upper bound is eliminated.

The percentage figures in columns 5 and 7 of Table 3 indicate the gaps between our estimated values for HK and the exact values computed by Johnson (1995b) (see also Reinelt, 1994 for exact values for selected TSPLIB instances). Clearly these gaps are very small indeed for most of the TSPLIB instances listed in the table. Our evaluations for dsj1000 and fl3795 are, relative to the other instances in the table, exceptionally wide of the mark, (by almost 1% and 2% respectively). Instances such as this are also discussed in (Johnson, 1996).

Having failed to establish any significant differences in performance between the two techniques we implemented, we set up some experiments to compare our results with those obtained by Reinelt. We ran our VJ code on the 24 instances from TSPLIB selected by Reinelt for similar experiments. He used these instances to validate his sparse subgraph speed-up technique for Lagrangian relaxation technique based on 1-trees (Table 10.9, page 178). Like Reinelt we used an (arbitrary) sequence length of 300 iterations on our subgraphs followed by a final iteration on the complete graph (Table 4). Our results were better than Reinelt's for 14 instances and worse for

Table 4
Deviation of HK estimate from best upper bound with sequence lengths of 300

| Problem | Reinhelt (% below best upper bound) | Our VJ results (% below best upper bound) |
|---|---|---|
| d198 | 5.65 | 7.87 |
| lin318 | 0.61 | 0.44 |
| fl417 | 3.50 | 5.77 |
| pcb442 | 0.63 | 0.61 |
| u574 | 0.57 | 0.53 |
| p654 | 4.23 | 5.62 |
| rat783 | 0.41 | 0.39 |
| pr1002 | 0.99 | 0.92 |
| u1060 | 0.87 | 0.88 |
| pcb1173 | 0.99 | 0.96 |
| d1291 | 1.70 | 1.74 |
| rl1323 | 1.72 | 1.66 |
| fl1400 | 6.38 | 4.23 |
| u1432 | 0.47 | 0.41 |
| fl1577 | 5.40 | 6.93 |
| d1655 | 1.24 | 1.38 |
| vm1748 | 1.37 | 1.36 |
| rl1889 | 1.74 | 1.72 |
| u2152 | 0.60 | 0.67 |
| pr2392 | 1.23 | 1.21 |
| pcb3038 | 0.84 | 0.83 |
| fl3795 | 4.61 | 6.12 |
| fnl14461 | 0.58 | 0.56 |
| rl5934 | 1.23 | 1.79 |
| average | 1.98 | 2.28 |

Table 5
Sequence length versus problem size

| Problem size | Mean number of iterations required | 95% confidence limit |
|---|---|---|
| 100 | 417 | 542 |
| 200 | 734 | 1028 |
| 500 | 1400 | 1880 |
| 1000 | 2848 | 4199 |
| 2000 | 3330 | 4708 |
| 5000 | 4290 | 6392 |

the other 10. However Reinelt's results were better on average than ours. He obtained a percentage deviation from the best known upper bound that averaged 1.98 over the 24 problems, whilst our experiments produced an average figure of 2.28%. Table 4 summarises the results of these experiments.

### 6.3. How many iterations?

The experiments documented in the previous section demonstrate a remarkable robustness in the 1-tree relaxation scheme and the question naturally arises as to whether the long sequence lengths suggested in the original papers are really necessary. The excellent results obtained by us and by Reinelt for many of the 24 TSPLIB instances under test in the previous section suggest that sequences as short as 300 can be sufficient for many problems up to and including those with about 5000 cities.

The next set of experiments strongly suggests that relatively short iteration sequences are sufficient for random points in the unit square.

The results presented in Table 5 indicate, for a range of problem sizes, approximately how many iterations are required in order to equal the "goal" Held-Karp lower bound value obtained from the earlier geometric convergence scheme. Once this "goal" has been reached, correct to 5 significant figures, for a particular problem, no larger sequence lengths are tested. For each problem size 10 problem instances are generated by producing the required number of random points in the unit square. For a particular problem instance, iteration sequences start at 100 and are incremented by 100 to generate

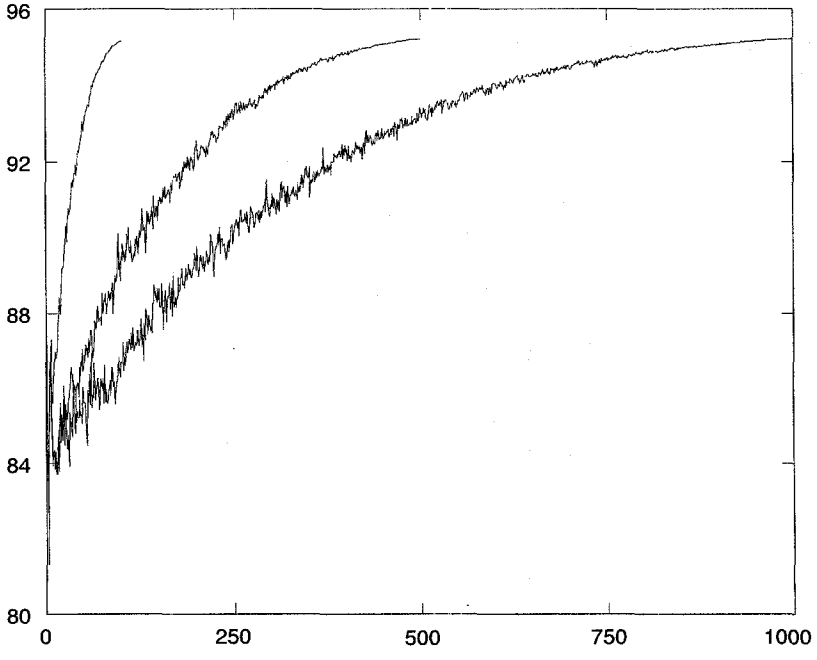


Fig. 5. Held-Karp iteration sequences, $w(\pi)$ ($y$ axis) vs iteration number ($x$ axis) for $M = 100$ (left), $M = 500$ (centre) and $M = 1000$ (right).

Table 6
Percentages off best HK bound obtained using short sequences

| Problem size | Sequence lengths | | |
|---|---|---|---|
| | 100 | 200 | 300 |
| 100 | 0.02% | 0% | 0.01% |
| 200 | 0.09% | 0.07% | 0.03% |
| 500 | 0.10% | 0.04% | 0.02% |
| 1000 | 0.07% | 0.03% | 0.02% |
| 2000 | 0.04% | 0.02% | 0.02% |
| 5000 | 0.04% | 0.02% | 0.01% |
| 10000 | 0.04% | 0.02% | 0.02% |

iteration sequences of lengths 100, 200, 300, 400 etc. in turn. These iteration sequences of increasing length are tried in turn until the "goal" has been reached for that instance. Table 5 records averages and 95% confidence limits for the sets of 10 instances. (The 95% confidence limits assume a Gaussian distribution and represent iteration sequence lengths expected to succeed in 95% of cases).

It is clear from Table 5 that the long sequences used on the larger problems in the previous experiments are not necessary.

A least-squares line on the logarithms of problem size, $n$, versus mean sequence length from Table 5 yields the following rough guide for the choice of $M$, the sequence length,

$$M = 28n^{0.62}. \tag{27}$$

Extrapolating from this formula would suggest that about 17,000 iterations would be required for a 30,000 city problem and about 147,000 for a million cities. However such estimates are highly speculative, and the rough guide given for $M$ above is only valid within the range 100–5000 cities.

From other experiments we deduced that very good approximations can be obtained by using very short sequences indeed (of about 100 iterations). This point is illustrated in Fig. 5 which shows the application of the Volgenant and Jonker scheme for three different sequence lengths, $M$. The results produced from short sequences on seven problem instances in the range 100–10,000 cities are presented in Table 6. These instances are all uniform random points. Clearly excellent results can be obtained using very short sequence lengths.

Experiments with instances from TSPLIB indicate a less straight-forward relationship between problem size and sequence length, instances with pathologically distributed cities generally requiring longer iteration sequences than typical instances of similar size. In such cases a self-adjusting sequence length may be more appropriate. Such a scheme has been described by Helbig-Hansen and Krarup (1974).

### 6.4. What are the run-time issues?

As previously noted a good implementation of Kruskal's algorithm has a time complexity of $O(e\log e)$, where $e$ is the number of edges in the minimum spanning tree calculation. As the evaluation of a single 1-tree for a Held-Karp iteration scheme is dominated by Kruskal's algorithm, then it would also be expected to have a time complexity of $O(e\log e)$. In our implementations $e = 20n$ for the main iteration loop, making $O(n\log n)$ the time complexity for a single minimum 1-tree evaluation.

Execution times were measured for Held-Karp subgraph evaluations using problem sizes varying between 100 and 20000 cities. 100 iterations were timed in each case using the Volgenant and Jonker formula. Analysis of these results provided empirical evidence to support a time complexity $O(n\log n)$ for the Held-Karp 1-tree evaluations. Obviously, the time complexity for the entire calculation must depend on the number of iterations required for a particular problem size, as well as the cost of one iteration. In addition the final $n^2$ complete graph calculation must not be forgotten, and neither should the evaluation of the nearest neighbours lists required at the start of the program (the nearest neighbours lists are required in order to determine which edges should be

Table 7
Timing Held-Karp runs on a SPARC 10, including nearest neighbours and final MST calculations

| Problem size | Number of iterations | Time (sec) |
|---|---|---|
| 100 | 487 | 4.2 |
| 200 | 748 | 12.6 |
| 500 | 1320 | 73.3 |
| 1000 | 2028 | 393.1 |
| 2000 | 3117 | 919.6 |
| 5000 | 5502 | 5449.3 |

Table 8
HK/$\sqrt{n}$ and confidence intervals

| Problem size | Number of samples | Mean | Standard deviation | 95% confidence intervals | |
|---|---|---|---|---|---|
| | | | | Upper | Lower |
| 100 | 5000 | 0.77015 | 0.02245 | 0.77077 | 0.76953 |
| 200 | 2000 | 0.75032 | 0.01506 | 0.75098 | 0.74966 |
| 500 | 1000 | 0.73389 | 0.00909 | 0.73445 | 0.73333 |
| 1000 | 500 | 0.72588 | 0.00667 | 0.72646 | 0.72530 |
| 2000 | 200 | 0.72065 | 0.00464 | 0.72129 | 0.72001 |
| 5000 | 50 | 0.71578 | 0.00249 | 0.71647 | 0.71509 |
| 10000 | 20 | 0.71340 | 0.00177 | 0.71418 | 0.71262 |
| 20000 | 10 | 0.71184 | 0.00116 | 0.71256 | 0.71112 |

included in the subgraph). Although shortcuts for evaluating nearest neighbours lists in time $O(n\log n)$ are known (for examples see Bently (1975) for $k-d$ trees and Shamos and Hoey (1975) for Voronoi diagrams) we implemented this routine as a simple $O(n^2)$ calculation. Our code for the final MST on the complete graph also ran in $O(n^2)$ time.

Table 7 gives a summary of run-times for our code obtained on a SPARC 10 workstation for problems in the range 100–5000 cities. Iteration sequence lengths were obtained from Eq. (27). Estimating HK on a 100 city problem takes less than five seconds whilst running the same code on 5000 cities requires about one and a half hours. Applying our code to problems with 10,000 cities or more proved problematic due to the memory requirements of the final MST calculation. However, the fact that our final value for HK(subgraph) always equals HK(completegraph) in all our experiments with random points in the unit square (with 20 nearest neigh-

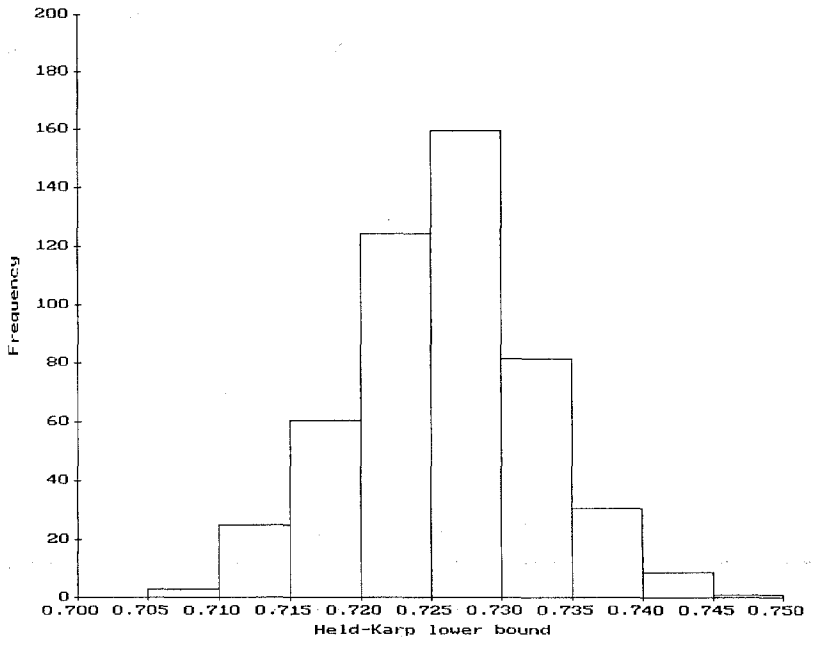


Fig. 6. Values for HK/$\sqrt{n}$ for 500 problems of 1000 cities.

Table 9
Results of fitting to the three equations

| Equation | $a$ | $b$ | $c$ | $d$ | Total error squared |
|---|---|---|---|---|---|
| 28, part I | 0.70864 | 0.7794 | −0.5518 | – | $2.37 \times 10^{-7}$ |
| 28, part II | 0.70675 | 0.6248 | – | – | $2.91 \times 10^{-6}$ |
| 28, part III | 0.70787 | 0.5513 | 0.6345 | 0.7963 | $2.72 \times 10^{-8}$ |

bours and sequence length guided by Eq. (27)), suggests that it may be possible to leave out the final MST for practical purposes.

### 6.5. Estimating the Goemans and Bertsimas constant

Table 8 gives some measurements of $HK/\sqrt{n}$ for uniform random points, with distances evaluated as 64 bit floating point reals. The stated confidence intervals assume that samples are taken from Gaussian distributions, and the histogram in Fig. 6 indicates that this would seem to be a reasonable assumption.

The results of Table 8 for 100–20,000 cities were fitted to the three equations

$$y = a + bn^c,$$

$$y = a + \frac{b}{n^{0.5}},$$

$$y = a + \frac{b}{n^{0.5}} + \frac{c}{n} + \frac{d}{n^{1.5}}, \tag{28}$$

where $y = HK/\sqrt{n}$. This was done using Mathematica™ 'FindMinimum' for the squared error of Eq. (28) summed over the eight data points.

According to Goemans and Bertsimas (1991) the variance of $HK/\sqrt{n}$ decreases rapidly with increasing $n$. This reduction can be observed in column 4 of Table 8, and justifies the use of smaller numbers of data points as the problem size, $n$, increases. The best fit is illconditioned, the values of the later coefficients being sensitively dependent upon the value of $a$ (the asymptotic value we are trying to estimate). Table 9 shows the fitted coefficients and the total sum squared error between the resulting curve and the experimental data. It would appear that the best fit is obtained using the third equation of Eq. (28). Fig. 7 shows the best-fit curve against the

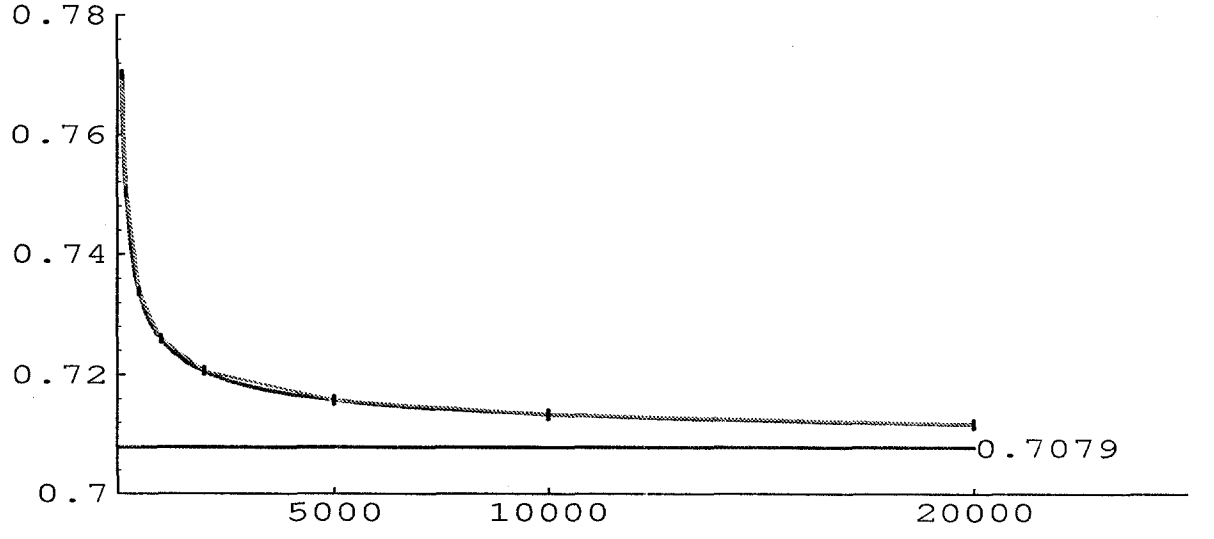


Fig. 7. Best fit curve (solid trace) against results (light grey). The error bars indicated are at the 95% confidence level.

$0.70787 \pm 0.0008$ resulted from applying the 95% confidence interval to Eq. (28). This value is very close to the estimate of $0.70805 \pm 0.00007$ obtained by Johnson (1996), fitting the same equation. Although it is interesting to note that the values we obtained for the coefficients labelled $c$ and $d$ in our third equation differed significantly from those obtained by Johnson et al., it is perhaps to be expected given the wide confidence intervals of our results and the illconditioned nature of the fit.

## 7. Conclusion

The main aim of this paper was to provide some practical guidelines regarding the iterative estimation of the Held-Karp lower bound for large TSP problems.

The extreme robustness of the iteration schemes based on subgradient optimization are now self-evident. We have demonstrated that in most instances our implementation produces estimates for HK within a very small fraction of a percent of the true (LP) value on a range of problems from TSPLIB (the largest deviations were about 2% on problems with extreme pathological distributions of cities).

We compared the performance of implementations using two different iteration formulae and failed to produce significant evidence strongly endorsing either one over the other. In addition we ran some experiments to compare the performance of our iterative schemes with those of Reinelt (1994) on a selection of TSPLIB instances using short iterative sequences. Once again the results were very similar, thus supplying further evidence of the extreme robustness of these schemes. However our favoured formula is that of Volgenant and Jonker, because an estimate for an upper bound is not needed with this method.

It seems that it is not necessary to use a full $n^2$ matrix of intercity distances for evaluating the minimum 1-trees. Strictly speaking a final $O(n^2)$ evaluation on the complete graph should be carried out, but in all our experiments on random points in the unit square where at least 20 edges to each city are used in our subgraph computations, HK(subgraph) = HK(completegraph). Thus it would appear that this final step can be omitted for practical purposes.

If the edges between each city and its 20 nearest neighbours are measured and recorded, Kruskal's algorithm can be used for evaluating the minimum 1-trees and a time complexity of $O(n \log n)$ for each evaluation achieved. The number of iterations required, $M$, appears to be sublinearly related to $n$, the problem size, and it would appear that in most instances sequence lengths of about 100–300 are sufficient for a good estimate of HK for problems in the range 100–10,000 cities. For instances from TSPLIB 40 near neighbours were required in order to get good results and we had to take steps to ensure that problems with pathological distributions of cities produced reasonably balanced subgraphs for the purposes of computing the MST values.

Run-times for our iterative schemes currently scale approximately $O(n^2)$, but this could be improved by using more efficient algorithms for producing the nearest neighbour lists and by eliminating the final MST. Actual run-times varied between about 5 seconds for a 100 city problem to about 1.5 hours for a 5000 city problem on a SPARC 10 workstation using the generous sequence lengths suggested by Eq. (27). Slightly inferior estimates can be obtained more quickly using iteration sequences 100–300 long.

Our estimate of $c_2 \approx 0.70787 \pm 0.0008$ is very close to that obtained by Johnson (1996).

Estimating $c_1$ empirically poses severe computational problems, but knowing this value more accurately is important in algorithm evaluation. It may be possible to refine the BHH approach to produce a more precise theoretical estimate for $c_1$ and this would certainly be desirable, particularly because such an analysis may give more information regarding the distribution of minimal tour lengths for random problems on the unit square, which itself remains an intractable problem.

Finally there remains the question (which we have hitherto avoided) of whether $c_1 = c_2$. Given our own and other experimental estimates this seems increasingly unlikely.

## Acknowledgements

We would like to thank David S. Johnson of AT&T for his many helpful comments on our work. We are also indebted to the anonymous referees for

their constructive criticism of our original submission.

## Appendix A

We sketch here the derivation details for the formula given in Eq. (23).

**Theorem.** *Suppose the finite sequence* $t^{(m)} \geq 0$ $(1 \leq m \leq M)$ *is monotonic decreasing with* $t^{(M)} = 0$, *and satisfies*

$$t^{(m)} - 2t^{(m-1)} + t^{(m-2)} = c, \tag{29}$$

*where* $c$ *is constant, and*

$$t^{(1)} - t^{(2)} = 3(t^{(M-1)} - t^{(M)}). \tag{30}$$

*Then*

$$t^{(m)} = (m-1)\left(\frac{2M-5}{2(M-1)}\right)t^{(1)} - (m-2)t^{(1)} + \frac{1}{2}\frac{(m-1)(m-2)}{(M-1)(M-2)}t^{(1)}. \tag{31}$$

*Remark.* Eq. (29) asserts that second order differences are constant. The constant in Eq. (30) is somewhat arbitrary. Volgenant and Jonker suggest that a ratio of 3 between the length of the initial and final interval empirically appears to yield marginally better results than other values.

**Proof.** The proof consists of first determining the relationship between the initial point of the sequence and the constant $c$ and then solving the difference Eq. (29). For $1 \leq m \leq M-1$ write

$$l(I_m) = t^{(m)} - t^{(m+1)}, \tag{32}$$

for the length of the $m$th interval. Then Eq. (29) asserts

$$l(I_{m-1}) - l(I_m) = c > 0, \quad 2 \leq m \leq M-1, \tag{33}$$

and we also have

$$\sum_{m=1}^{M-1} l(I_m) = t^{(1)} - t^{(M)} = t^{(1)}, \tag{34}$$

since $t^{(M)} = 0$. From Eq. (33) we have

$$\sum_{j=1}^{m-1} l(I_j) - l(I_{j+1}) = l(I_1) - l(I_m) = (m-1)c, \quad 2 \leq m \leq M-1, \tag{35}$$

Putting $m = M-1$ yields

$$l(I_1) - l(I_{M-1}) = (M-2)c. \tag{36}$$

Using Eq. (30) we obtain

$$l(I_1) = \frac{3}{2}(M-2)c. \tag{37}$$

From the righthand equation in Eq. (35) we obtain

$$\sum_{m=2}^{M-1}(l(I_1) - l(I_m)) = \left(\sum_{m=2}^{M-1}(m-1)\right)c, \tag{38}$$

which can be rewritten as

$$(M-2)l(I_1) - \sum_{m=2}^{M-1} l(I_m) = \frac{1}{2}(M-1)(M-2)c, \tag{39}$$

which in turn using Eq. (34) gives

$$(M-1)l(I_1) - t^{(1)} = \frac{1}{2}(M-1)(M-2)c. \tag{40}$$

Substituting this into Eq. (37) then gives

$$t^{(1)} = (M-1)(M-2)c, \tag{41}$$

which relates the initial point of the sequence to the constant $c > 0$. It now remains only to solve the difference Eq. (29). The auxiliary equation has a repeated root $\lambda = 1$ so the general solution to the homogeneous equation

$$t^{(m)} - 2t^{(m-1)} + t^{(m-2)} = 0, \tag{42}$$

is

$$t^{(m)} = A + Bm, \tag{43}$$

where $A$ and $B$ are arbitrary constants determined by the initial conditions. We next find a particular solution to Eq. (29). Substituting $t^{(m)} = \alpha m^2$ and solving for $\alpha$ we obtain $\alpha = c/2$. Thus the general solution of Eq. (29) is

$$t^{(m)} = A + Bm + \frac{c}{2}m^2. \tag{44}$$

Solving the equations derived by substituting $m = 1$ and $m = 2$ we obtain

$$A = 2t^{(1)} - t^{(2)} + c,$$

$$B = -t^{(1)} + t^{(2)} - \frac{3}{2}c. \tag{45}$$

Substituting these values into Eq. (44) yields

$$t^{(m)} = (m-1)t^{(2)} - (m-2)t^{(1)} + \tfrac{1}{2}c(M-1)(M-2). \quad (46)$$

Expressing $t^{(2)}$ in terms of $t^{(1)}$ using Eq. (37), and $c$ in terms of $t^{(1)}$ using Eq. (41) we finally obtain the solution given in Eq. (31).

## References


Applegate, D., Bixby, R., Chvátal, V. and Cook, W., 1995. *Finding cuts in the TSP (A preliminary report).* Report No. 95–05, Center for Discrete Mathematics and Theoretical Computer Science (DIMACS), Rutgers University, Piscataway, N.J.

Balas, E., and Toth, P., 1985. Branch and Bound Methods. In: E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys (eds), *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization.* John Wiley and Sons, New York.

Beardwood, J., Halton J.H. and Hammersley, J.M., 1959. The shortest path through many points. *Proceedings of the Cambridge Philosophical Society*, 55, 299–327.

Bently, J.L., 1975. Multidimensional binary search trees used for associative search. *Communications of the ACM*, 18, 309–517.

Christophides, N., 1979. The Traveling Salesman Problem. In: N. Christophides, A. Mingozzi, P. Toth and C. Sandi (eds), *Combinatorial Optimization.* John Wiley and Sons, New York.

Goemans, M.X. and Bertsimas, D.J., 1991. Probabilistic Analysis of the Held and Karp Lower Bound for the Euclidean Traveling Salesman Problem. *Mathematics of Operations Research*, 16(1), 72–89.

Helbig-Hansen, K.H. and Krarup, J., 1974. Improvements of the Held-Karp algorithm for the symmetric traveling salesman problem. *Mathematical Programming*, 7, 87–96.

Held, M. and Karp, R.M., 1970. The Traveling Salesman Problem and Minimum Spanning Trees. *Operations Research*, 18, 1138–1162.

Held, M. and Karp, R.M., 1971. The Traveling Salesman Problem and Minimum Spanning Trees, part II. *Mathematical Programming*, 1, 6–25.

Held, M., Wolfe, P. and Crowder, H.P., 1974. Validation of subgradient optimization. *Mathematical Programming*, 6, 62–88.

Horowitz, E. and Sahni, S., 1978. *Fundamentals of Computer Algorithms*, Pitman.

Johnson, D.S., 1988. Talk presented at the Mathematical Programming Symposium, Tokyo.

Johnson, D.S., 1990. Local Optimization and the Traveling Salesman Problem. *Automata Languages and Programming*, 17th International Colloquium Proceedings.

Johnson, D.S. and Mcgeoch, L.A., 1995a. The Traveling Salesman Problem: A Case Study in Local Optimization. To appear in: E.H.L. Aarts and J.K. Lenstra (eds), *Local Search in Combinatoric Optimization*, John Wiley and Sons, New York.

Johnson, D.S., 1995b. Personal communication.

Johnson, D.S., McGeoch, L.A. and Rothberg, E.E., 1996. Asymptotic experimental analysis for the Held-Karp traveling salesman bound. *Proceedings of the 1996 ACM-SIAM symposium on Discrete Algorithms.*

Polyak, B.T., 1967. A general method of solving extremal problems. *Soviet Mathematics Doklady*, 8, 593–597. (Translation of Doklady Akademii Nauk SSSR 174, 1967.)

Polyak, B.T., 1969. Minimization of unsmooth functionals. *Computational Mathematics and Mathematical Physics* 14–29. (Translation of Žurnal Vyčislitel'noĭ Matematiki i Matematičesko Fiziki, 9, 509–521.)

Reinelt, G., 1994. *The Traveling Salesman: Computational Solutions for TSP Applications.* Lecture Notes in Computer Science, 840, Springer-Verlag, Berlin.

Reinelt, G., 1995. A library of TSP problems from: E-mail: elib@ZIB-Berlin.de. Telnet: elib.zib-berlin.de. Just send the message 'help' via email to find out how to use elib.

Shamos, M.I. and Hoey, D., 1975. Closest Point Problems. *Proceedings of the 16th IEEE Annual Symposium on Foundations of Computer Science*, 151–162.

Shmoys, D.B. and Williamson, D.P., 1990. Analyzing the Held-Karp TSP Bound: A Monotonicity Property with Application. *Information Processing Letters*, 35, 281–285.

Stein, D., 1977. *Scheduling Dial-a-Ride transportation systems: an asymptotic approach.* Ph.D thesis, Harvard University, Cambridge, MA.

Valenzuela, C.L. and Jones, A.J., 1994. Evolutionary Divide and Conquer(I): a novel genetic approach to the TSP. *Evolutionary Computation*, 1(4), 313–333.

Valenzuela, C.L. and Jones, A.J., 1995a. A Parallel Implementation of Evolutionary Divide and Conquer for the TSP. *Proceedings of the First IEE / IEEE conference on Genetic ALgorithms in Engineering Systems: Innovations and Applications (GALESIA) Sheffield U.K. 12 – 14 September 1995*, 499–504.

Valenzuela, C.L., 1995b. *Evolutionary Divide and Conquer: a novel genetic approach to the TSP.* Unpublished Ph.D. thesis, Imperial College, University of London.

Volgenant, T. and Jonker, R., 1982. A branch and bound algorithm for the symmetric traveling salesman problem based on the 1-tree relaxation. *European Journal of Operational Research*, 9, 83–89.

Wichmann, B.A. and Hill, D., 1982. *A Pseudo-random Number Generator.* National Physical Laboratory U.K. Report DITC 6–82.

Wolsey, L., 1980. Heuristic analysis, linear programming and branch and bound. *Mathematical Programming Study*, 13, 121–134.