Solving a "Hard" Problem to Approximate an "Easy" One: Heuristics for Maximum Matchings and Maximum Traveling Salesman Problems

Sándor P. Fekete[1], Henk Meijer[2], André Rohe[3], and Walter Tietze[1]

[1] Department of Mathematics, TU Berlin
10623 Berlin, Germany
{fekete,tietze}@math.tu-berlin.de
[2] Department of Computing and Information Science
Queen's University
Kingston, Ontario K7L 3N6, Canada
henk@cs.queensu.ca
[3] Forschungsinstitut für Diskrete Mathematik
Universität Bonn
53113 Bonn, Germany
rohe@or.uni-bonn.de

Abstract. We consider geometric instances of the Maximum Weighted Matching Problem (MWMP) and the Maximum Traveling Salesman Problem (MTSP) with up to 3,000,000 vertices. Making use of a geometric duality relationship between MWMP, MTSP, and the Fermat-Weber-Problem (FWP), we develop a heuristic approach that yields in near-linear time solutions as well as upper bounds. Using various computational tools, we get solutions within considerably less than 1% of the optimum.

An interesting feature of our approach is that, even though an FWP is hard to compute in theory and Edmonds' algorithm for maximum weighted matching yields a polynomial solution for the MWMP, the practical behavior is just the opposite, and we can solve the FWP with high accuracy in order to find a good heuristic solution for the MWMP.

1 Introduction

Complexity in Theory and Practice. In the field of discrete algorithms, the classical way to distinguish "easy" and "hard" problems is to study their worst-case behavior. Ever since Edmonds' seminal work on maximum matchings [7,8], the adjective "good" for an algorithm has become synonymous with a worst-case running time that is bounded by a polynomial in the input size. At the same time, Edmonds' method for finding a maximum weight perfect matching in a complete graph with edge weights serves as a prime example for a sophisticated combinatorial algorithm that solves a problem to optimality. Furthermore, finding an optimal matching in a graph is used as a stepping stone for many heuristics for hard problems.

A.L. Buchsbaum and J. Snoeyink (Eds.): ALENEX 2001, LNCS 2153, pp. 1–16, 2001.

The classical prototype of such a "hard" problem is the Traveling Salesman Problem (TSP) of computing a shortest roundtrip through a set P of n cities. Being NP-hard, it is generally assumed that there is no "good" algorithm in the above sense: Unless P=NP, there is no polynomial-time algorithm for the TSP. This motivates the performance analysis of polynomial-time heuristics for the TSP. Assuming triangle inequality, the best polynomial heuristic known to date uses the computation of an optimal weighted matching: Christofides' method combines a Minimum Weight Spanning Tree (MWST) with a Minimum Weight Perfect Matching of the odd degree vertices, yielding a worst-case performance of 50% above the optimum.

Geometric Instances. Virtually all very large instances of graph optimization problems are geometric. It is easy to see why this should be the case for practical instances. In addition, a geometric instance given by n vertices in $\mathbb{R}^d$ is described by only dn coordinates, while a distance matrix requires $\Omega(n^2)$ entries; even with today's computing power, it is hopeless to store and use the distance matrix for instances with, say, $n = 10^6$.

The study of geometric instances has resulted in a number of powerful theoretical results. Most notably, Arora [2] and Mitchell [16] have developed a general framework that results in polynomial time approximation schemes (PTASs) for many geometric versions of graph optimization problems: Given any constant ϵ, there is a polynomial algorithm that yields a solution within a factor of $(1+\epsilon)$ of the optimum. However, these breakthrough results are of purely theoretical interest, since the necessary computations and data storage requirements are beyond any practical orders of magnitude.

For a problem closely related to the TSP, there is a different way how geometry can be exploited. Trying to find a longest tour in a weighted graph is the so-called *Maximum Traveling Salesman Problem* (MTSP); it is easy to see that for graph instances, the MTSP is just as hard as the TSP. Making clever use of the special geometry of distances, Barvinok, Johnson, Woeginger, and Woodroofe [4] showed that for geometric instances in $\mathbb{R}^d$, it is possible to solve the MTSP in polynomial time, provided that distances are measured by a *polyhedral metric*, which is described by a unit ball with a fixed number $2f$ of facets. (For the case of Manhattan distances in the plane, we have $f = 2$, and the resulting complexity is $O(n^{2f-2}\log n) = O(n^2 \log n)$.) By using a large enough number of facets to approximate a unit sphere, this yields a PTAS for Euclidean distances.

Both of these approaches, however, do not provide practical methods for getting good solutions for very large geometric instances. And even though TSP and matching instances of considerable size have been solved to optimality (up to 13,000 cities with about 2 years of computing time [1]), it should be stressed that for large enough instances, it seems quite difficult to come up with small gaps within a very short (i.e., near-linear in n) time. Moreover, the methods involved only use triangle inequality, and disregard the special properties of geometric instances.

For the *Minimum* Weight Matching problem, Vaidya [19] showed that there is algorithm of complexity $O(n^{2.5}\log^4 n)$ for planar geometric instances, which was improved by Varadarajan [20] to $O(n^{1.5}\log^5 n)$. Cook and Rohe [6] also made heavy use of geometry to solve instances with up to 5,000,000 points in the plane within about 1.5 days of computing time. However, all these approaches use specific properties of planar nearest neighbors. Cook and Rohe reduce the number of edges that need to be considered to about 8,000,000, and solve the problem in this very sparse graph. These methods cannot be applied when trying to find a *Maximum* Weight Matching. (In particular, a divide-and-conquer strategy seems unsuited for this type of problem, since the structure of furthest neighbors is quite different from the well-behaved “clusters” formed by nearest neighbors.)

Heuristic Solutions. A standard approach when considering “hard” optimization problems is to solve a closely related problem that is “easier”, and use this solution to construct one that is feasible for the original problem. In combinatorial optimization, finding an optimal perfect matching in an edge-weighted graph is a common choice for the easy problem. However, for practical instances of matching problems, the number n of vertices may be too large to find an exact optimum in reasonable time, since the best complexity of an exact algorithm is $O(n(m+n\log n))$ [11] (where m is the number of edges)[1].

We have already introduced the Traveling Salesman Problem, which is known to be NP-hard, even for geometric instances. A problem that is hard in a different theoretical sense is the following: For a given set P of n points in $\mathbb{R}^2$, the Fermat-Weber Problem (FWP) is to minimize the size of a “Steiner star”, i.e., the total Euclidean distance $S(P)=\min_{c\in\mathbb{R}}\sum_{p\in P} d(c,p)$ of a point c to all points in P. It was shown in [3] that even for the case $n=5$, solving this problem requires finding zeroes of high-order polynomials, which cannot be achieved using only radicals.

Solving the FWP and solving the geometric maximum weight matching problem (MWMP) are closely related: It is an easy consequence of the triangle inequality that $\mathrm{MWMP}(P)\le\mathrm{FWP}(P)$. For a natural geometric case of Euclidean distances in the plane, it was shown in [10] that $\mathrm{FWP}(P)/\mathrm{MWMP}(P)\le 2/\sqrt{3}\approx 1.15$.

From a theoretical point of view, this may appear to assign the roles of “easy” and “hard” to MWMP and FWP. However, from a practical perspective, roles are reversed: While solving large maximum weight matching problems to optimality seems like a hopeless task, finding an optimal Steiner center c only requires minimizing a convex function. Thus, the latter can be solved very fast numerically (e.g., by Newton’s method) within any small ε. The twist of this paper is to use that solution to construct a fast heuristic for maximum weight matchings – thereby solving a “hard” problem to approximate an “easy” one. Similar ideas can be used for constructing a good heuristic for the MTSP.

[1] Quite recently, Mehlhorn and Schäfer [15] have presented an implementation of this algorithm; the largest dense graphs for which they report optimal results have 4,000 nodes and 1,200,000 edges.

Summary of Results. It is the main objective of this paper to demonstrate that the special properties of geometric instances make them much easier *in practice* than general instances on weighted graphs. Using these properties gives rise to heuristics that construct excellent solutions in near-linear time, with very small constants. Since the analytic worst-case ratio of FWP(P)/MWMP(P) is only $2/\sqrt{3} \approx 1.15$, it is certain that the difference to the optimum will never exceed 15%, but can be expected to be much less in practice.

1. This is validated by a *practical study on instances up to 3,000,000 points*, which can be dealt with in less than three minutes of computation time, resulting in error bounds of not more than about 3% for one type of instances, but only in the order of 0.1% for most others. The instances consist of the well-known TSPLIB, and random instances of two different random types, uniform random distribution and clustered random distribution.

To evaluate the quality of our results for both MWMP and MTSP, we employ a number of additional methods, including the following:

2. *An extensive local search by use of the chained Lin-Kernighan method* yields only small improvements of our heuristic solutions. This provides experimental evidence that a large amount of computation time will only lead to marginal improvements of our heuristic solutions.
3. *An improved upper bound* (that is more time-consuming to compute) indicates that the remaining gap between the fast feasible solutions and the fast upper bounds is too pessimistic on the quality of the heuristic, since the gap seems to be mostly due to the difference between the optimum and the upper bound.
4. *A polyhedral result on the structure of optimal solutions* to the MWMP allows the computation of the exact optimum by using a network simplex method, instead of employing Edmonds' blossom algorithm. This result (stating that there is always an integral optimum of the standard LP relaxation for planar geometric instances of the MWMP) is interesting in its own right and was observed previously by Tamir and Mitchell [18]. A comparison for instances with less than 10,000 nodes shows that the gap between the solution computed by our heuristic and the upper bound derived from the FWP(P) is much larger than the difference between our solution and the actual optimal value of the MWMP(P), which turns out to be at most 0.26%, even for clustered instances. Moreover, twice the optimum solution for the MWMP is also an upper bound for the MTSP. For both problems, this provides more evidence that additional computing time will almost entirely be used for lowering the fast upper bound on the maximization problem, while the feasible solution changes only little.

In addition, we provide a number of mathematical tools to make the results for the MWMP applicable to the MTSP. These results include:

5. The worst-case estimate for the ratio between MTSP(P) and FWP(P) is slightly worse than the one between MWMP(P) and FWP(P), since there are instances where we have FWP(P)/MTSP(P)$= 2/(2+\sqrt{2}) \approx 0.586 > 0.577 \approx 1/\sqrt{3} \geq$ FWP(P)/2MWMP(P). However, we show that for large n, the asymptotic worst-case performance for the MTSP is the same as for the MWMP. This means that the worst-case gap for our heuristic is also bounded by 15%, and not by 17%, as suggested by the above example.
6. For a planar set of points that are sorted in convex position (i.e., the vertices of a polyhedron in cyclic order), we can solve the MWMP and the MTSP in linear time.

The results for the MTSP are of similar quality as for the MWMP. Further evidence is provided by an additional computational study:

7. We compare the feasible solutions and bounds for our heuristic with an "exact" method that uses the existing TSP package CONCORDE for TSPLIB instances of moderate size (up to about 1000 points). It turns out that most of our results lie within the widely accepted margin of error caused by rounding Euclidean distances to the nearest integer. Furthermore, the (relatively time-consuming) standard Held-Karp bound is outperformed by our methods for most instances.

2 Minimum Stars and Maximum Matchings

2.1 Background and Algorithm

Consider a set P of points in $\mathbb{R}^2$ of even cardinality n. The Fermat-Weber Problem (FWP) is given by minimizing the total Euclidean distance of a "median" point c to all points in P, i.e., FWP(P) $= \min_{c \in R} \sum_{p \in P} d(c,p)$. This problem cannot be solved to optimality by methods using only radicals, since it requires to find zeroes of high-order polynomials, even for instances that are symmetric to the y-axis; see [3]. However, the objective function is strictly convex, so it is possible to solve the problem numerically with any required amount of accuracy. A simple binary search will do, but there are more specific approaches like the so-called Weiszfeld iteration [21,12]. We achieved the best results by using Newton's method.

The relationship between the FWP and the MWMP for a point set of even cardinality n has been studied in [10]: Any matching edge between two points p_i and p_j can be mapped to two "rays" (c, p_i) and (c, p_j) of the star, so it follows from triangle inequality that MWMP(P) $\leq$ FWP(P). Clearly, the ratio between the values MWMP(P) and FWP(P) depends on the amount of "shortcutting" that happens when replacing pairs of rays by matching edges; moreover, any lower bound for the angle ϕ_{ij} between the rays for a matching edge is mapped directly to a worst-case estimate for the ratio, since it follows from elementary trigonometry that $d(c,p_i) + d(c,p_j) \leq \sqrt{\frac{2}{1-\cos\phi_{ij}}} \cdot d(p_i,p_j)$. See Fig. 1. It was shown in [10] that there is always a matching with $\phi_{ij} \geq 2\pi/3$ for all angles

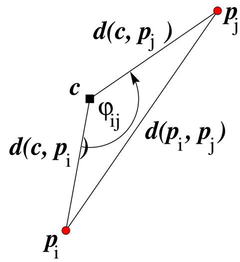

Fig. 1. Angles and rays for a matching edge (p_i, p_j).

ϕ_{ij} between rays. This bound can be used to prove that FWP(P)/MWMP(P) $\leq 2/\sqrt{3} \approx 1.15$.

Algorithm CROSS: Heuristic solution for MWMP

Input: A set of points $P \in \mathbb{R}^2$.

Output: A matching of P.

1. Using a numerical method, find a point c that approximately minimizes the convex function $\min_{c \in \mathbb{R}^2} \sum_{p_i \in P} d(c, p_i)$.

2. Sort the set P by angular order around c. Assume the resulting order is $p_1, \ldots, p_n$.

3. For $i = 1, \ldots, n/2$, match point p_i with point $p_{i+\frac{n}{2}}$.

Fig. 2. The heuristic CROSS.

If the above lower bound on the angle can be improved, we get a better estimate for the value of the matching. This motivates the heuristic CROSS for large-scale MWMP instances that is shown in Fig. 2. See Fig. 3 for a heuristic solution of the TSPLIB instance dsj1000.

Note that beyond a critical accuracy, the numerical method used in step **1** will not affect the value of the matching, since the latter only changes when the order type of the resulting center point c changes with respect to P. This means

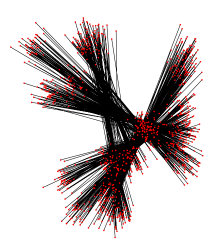

Fig. 3. A heuristic MWMP solution for the TSPLIB instance dsj1000 that is within 0.19% of the optimum.

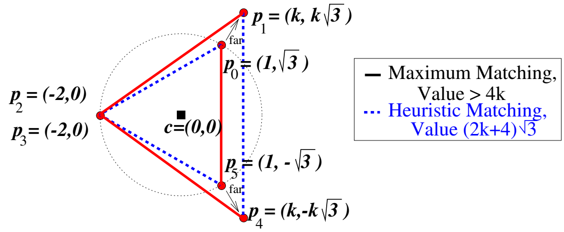

Fig. 4. A class of examples for which CROSS is 15% away from the optimum.

that spending more running time for this step will only lower the upper bound. We will encounter more examples of this phenomenon below.

The class of examples in Fig. 4 shows that the worst-case relative error estimate of about 15% is indeed best possible, since the ratio between optimal and heuristic matching may get arbitrarily close to $2/\sqrt{3}$. As we will see further down, this worst-case scenario is highly unlikely and the actual error is much smaller.

Furthermore, it is not hard to see that CROSS is optimal if the points are in convex position:

Theorem 1. *If the point set P is in strictly convex position, then algorithm CROSS determines the unique optimum.*

For a proof, observe that any pair of matching edges must be crossing, otherwise we could get an improvement by performing a 2-exchange.

2.2 Improving the Upper Bound

When using the value FWP(P) as an upper bound for MWMP(P), we compare the matching edges with pairs of rays, with equality being reached if the angle enclosed between rays is π, i.e., for points that are on opposite sides of the center point c. However, it may well be the case that there is no point opposite to a point p_i. In that case, we have an upper bound on $\max_j \phi_{ij}$, and we can lower the upper bound $FWP(P)$. See Fig. 5: the distance $d(c, p_i)$ is replaced by $d - \frac{\min_{j \neq i}(d(c,p_i)+d(c,p_j)-d(p_i,p_j))}{2}$.

Moreover, we can optimize over the possible location of point c. This lowers the value of the upper bound FWP(P), yielding the improved upper bound FWP'(P):

$$FWP'(P) = \min_{c \in I\!R^2} \sum_{p_i \in P} d(c, p_i) - \frac{\min_{j \neq i}(d(c, p_i) + d(c, p_j) - d(p_i, p_j))}{2}.$$

This results in a notable improvement, especially for clustered instances. However, the running time for computing this modified upper bound FWP'(P) is superquadratic. Therefore, this approach is only useful for mid-sized instances, and when there is sufficient time.

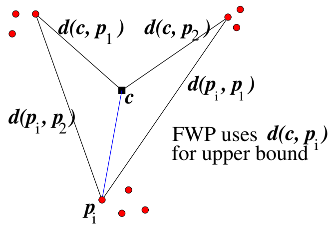

Fig. 5. Improving the upper bound.

2.3 An Integrality Result

A standard approach in combinatorial optimization is to model a problem as an integer program, then solve the linear programming relaxation. As it turns out, this works particularly well for the MWMP:

Theorem 2. *Let x be a set of nonnegative edge weights that is optimal for the standard linear programming relaxation of the MWMP, where all vertices are required to be incident to a total edge weight of 1. Then the weight of x is equal to an optimal integer solution of the MWMP.*

This theorem has been observed previously by Tamir and Mitchell [18]. The proof assumes the existence of two fractional odd cycles, then establishes the existence of an improving 2-exchange by a combination of parity arguments.

Theorem 2 allows it to compute the exact optimum by solving a linear program. For the MWMP, this amounts to solving a network flow problem, which can be done by using a network simplex method.

2.4 Computational Experiments

Table 1 summarizes some of our results for the MWMP for three classes of instances, described below. It shows a comparison of the FWP upper bound with different Matchings: In the first column the CROSS heuristic was used to compute the matching. In the second column we report the corresponding computing times on a Pentium II 500Mhz (using C code with compiler gcc -O3 under Linux 2.2). The third column gives the result of combining the CROSS matching with one hour of local search by chained Lin-Kernighan [17]. The last column compares the optimum computed by a network simplex using Theorem 2 with the upper bound (for $n < 10,000$). For the random instances, the average performance over ten different instances is shown.

The first type of instances are taken from the well-known TSPLIB benchmark library. (For odd cardinality TSPLIB instances, we follow the custom of dropping the last point from the list.) Clearly, the relative error decreases with increasing n.

The second type was constructed by choosing n points in a unit square uniformly at random. The reader may observe the near-linear running time. It

Table 1. Maximum matching results for TSPLIB (top), uniform random (center), and clustered random instances (bottom)

Instance	CROSS vs. FWP	time	CROSS + 1h Lin-Ker	CROSS vs. OPT
dsj1000	1.22%	0.05 s	1.07%	0.19%
nrw1378	0.05%	0.05 s	0.04%	0.01%
fnl4460	0.34%	0.13 s	0.29%	0.05%
usa13508	0.21%	0.64 s	0.19%	-
brd14050	0.67%	0.59 s	0.61%	-
d18512	0.14%	0.79 s	0.13%	-
pla85900	0.03%	3.87 s	0.03%	-
1000	0.03%	0.05 s	0.02%	0.02%
3000	0.01%	0.14 s	0.01%	0.00%
10000	0.00%	0.46 s	0.00%	-
30000	0.00%	1.45 s	0.00%	-
100000	0.00%	5.01 s	0.00%	-
300000	0.00%	15.60 s	0.00%	-
1000000	0.00%	53.90 s	0.00%	-
3000000	0.00%	159.00 s	0.00%	-
1000c	2.90%	0.05 s	2.82%	0.11 %
3000c	1.68%	0.15 s	1.59%	0.26 %
10000c	3.27%	0.49 s	3.24%	-
30000c	1.63%	1.69 s	1.61%	-
100000c	2.53%	5.51 s	2.52%	-
300000c	1.05%	17.51 s	1.05%	-

should also be noted that for this distribution, the relative error rapidly converges to zero. This is to be expected: for uniform distribution, the expected angle $\angle(p_i, c, p_{i+\frac{n}{2}})$ becomes arbitrarily close to π. In more explicit terms: Both the value FWP$/n$ and MWMP$/n$ for a set of n random points in a unit square tend to the limit $\int_{-1/2}^{1/2}\int_{-1/2}^{1/2}\sqrt{x^2+y^2}dxdy \approx 0.3826$.

The third type uses n points that are chosen by selecting random points from a relatively small expected number k of “cluster” areas. Within each cluster, points are located with uniform polar coordinates (with some adjustment for clusters near the boundary) with a circle of radius 0.05 around a central point, which is chosen uniformly at random from the unit square. This type of instances

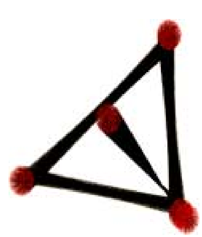

Fig. 6. A typical cluster example with its matching.

is designed to make our heuristic look bad; for this reason, we have shown the results for $k = 5$. See Figure 6 for a typical example with $n = 10,000$.

It is not hard to see that these cluster instances behave very similar to fractional instances with k points; moreover, for increasing k, we approach a uniform random distribution over the whole unit square, meaning that the performance is expected to get better. But even for small k, the reader may take note that for cluster instances, the remaining error estimate is almost entirely due to limited performance of the upper bound. The good quality of our fast heuristic for large problems is also illustrated by the fact that one hour of local search by Lin-Kernighan fails to provide any significant improvement.

3 The Maximum TSP

As we noted in the introduction, the geometric MTSP displays some peculiar properties when distances are measured according to some polyhedral norm. In fact, it was shown by Fekete [9] that for the case of Manhattan distances in the plane, the MTSP can be solved in linear time. (The algorithm is based in part on the observation that for planar Manhattan distances, FWP(P)=MWMP(P).) On the other hand, it was shown in the same paper that for Euclidean distances in $\mathbb{R}^3$ or on the surface of a sphere, the MTSP is NP-hard. The MTSP has also been conjectured to be NP-hard for the case of Euclidean distances in $\mathbb{R}^2$.

3.1 A Worst-Case Estimate

Clearly, there are some observations for the MWMP that can be applied to the MTSP. In particular, we note that MTSP(P)$\leq$ 2MWMP(P)$\leq$ 2FWP(P). On the other hand, the lower-bound estimate of $\sqrt{3}/2*$FWP(P) that holds for MWMP(P) does not imply a lower bound of $\sqrt{3}$FWP(P) for the MTSP(P), as can be seen from the example in Fig. 7, showing that a relative error of 17% is possible.

However, we can argue that asymptotically, the worst-case ratio FWP(P)/ MTSP(P) is analogous to the $\frac{2}{\sqrt{3}}$ for the MWMP, i.e., within 15% of 2:

Theorem 3. *For $n \to \infty$, the worst-case ratio of FWP(P)/MTSP(P) tends to $1/\sqrt{3}$.*

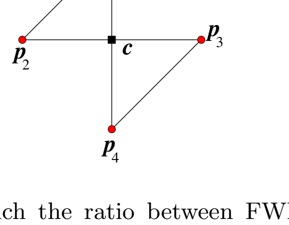

Fig. 7. An example for which the ratio between FWP and MTSP is greater than $1/\sqrt{3} \approx 0.577$.

Proof: The proof of the $\frac{2}{\sqrt{3}}$ bound for the MWMP in [10] establishes that any planar point set can be subdivided by six sectors of $\pi/3$ around one center point, such that opposite sectors have the same number of points. This allows a matching between opposite sectors, establishing a lower bound of $2\pi/3$ for the angle between the corresponding rays. This means that we can simply choose three subtours, one for each pair of opposite sectors, and achieve the same worst-case ratio as for a matching. In order to merge these subtours, we only need three edges between adjacent sectors. If there more than $n/2$ points "far" from the center, i.e., at least $\Omega(\mathrm{FWP}(P)/n)$ away from the center, then the resulting error tends to 0 as n grows, and we get the same worst-case estimate as for the MWMP.

This leaves the case that at least $n/2$ points are "close" to the center, i.e., only $o(\mathrm{FWP}(P)/n)$ from the center. Then we can can collect all points far from the center individually from the cluster close to the center. Now it is not hard to see that for this case, the length of the resulting tour converges to $2\mathrm{FWP}(P)$. □

3.2 A Modified Heuristic

For an even number of points in convex position, the choice of a maximum matching is rather straightforward. This leads to the CROSS heuristic described above. Similarly, it is easy to determine a maximum tour if we are dealing with an odd number of points in convex position: Each point p_i gets connected to its two "cyclic furthest neighbors" $p_{i+\lfloor \frac{n}{2} \rfloor}$ and $p_{i+\lceil \frac{n}{2} \rceil}$. However, the structure of an optimal tour is less clear for a point set of even cardinality, and therefore it is not obvious what permutations should be considered for an analogue to the matching heuristic CROSS. For this we consider the local modification called *2-exchanges*: One pair of (disjoint) tour edges (p_i, p_j) and (p_k, p_ℓ) gets replaced by the pair (p_i, p_k) and (p_j, p_ℓ), and the sequence $p_\ell, \ldots, p_i$ is reversed into $p_i, \ldots, p_\ell$.

Theorem 4. *If the point set P is in convex position, then there are at most $n/2$ tours that are locally optimal with respect to 2-exchanges, and we can determine the best in linear time.*

Proof: We claim that any tour that is locally optimal with respect to 2-exchanges must look like the one in Fig. 8: It consists of two *diagonals* $(p_i, p_{i+\frac{n}{2}})$ and $(p_{i+1}, p_{i+1+\frac{n}{2}})$ (in the example, these are the edges $(5, 11)$ and $(6, 0)$), while all other edges are *near-diagonals*, i.e., edges of the form $(p_j, p_{j+\frac{n}{2}-1})$.

First consider 2-exchanges that increase the tour length: It is an easy consequence of triangle inequality that a noncrossing disjoint "antiparallel" pair of edges as e_0 and e_1 in Fig. 9(a) allows a crossing 2-exchange that increases the overall tour length. In the following, we will will focus on identifying antiparallel noncrossing edge pairs.

Now we show that all edges in a locally optimal tour must be diagonals or near-diagonals: Consider an edge $e_0 = (p_i, p_j)$ with $0 < j - i \leq \frac{n}{2} - 2$. Then there are at most $\frac{n}{2} - 3$ points in the subset $P_1 = [p_{i+1}, \ldots, p_{j-1}]$, but at least

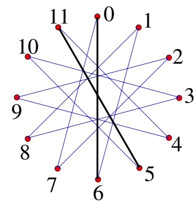

Fig. 8. A locally optimal MTSP tour.

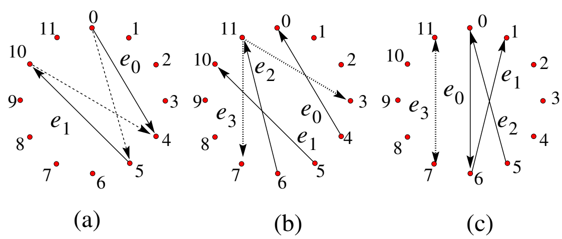

Fig. 9. Discussing locally optimal tours.

$\frac{n}{2}+1$ points in the subset $P_2 = [p_{j+1}, \ldots, p_{i-1}]$. This implies that there must be at least two edges (say, e_1 and e_2) within the subset P_2. If either of them is antiparallel to e_0, we are done, so assume that both of them are parallel. Without loss of generality assume that the head of e_2 lies "between" the head of e_1 and the head p_j of e_0, as shown in Fig. 9(b). Then the edge e_3 that is the successor of e_2 in the current tour is either antiparallel and noncrossing with e_1, or with e_0.

Next consider a tour only consisting of diagonals and near-diagonals. Since there is only one 2-factor consisting of nothing but near-diagonals, assume without loss of generality that there is at least one diagonal, say $(p_0, p_{\frac{n}{2}})$. Then the successor of $p_{\frac{n}{2}}$ and the predecessor of p_0 cannot lie on the same side of e_0, as shown in Fig. 9. Then there must be an edge e_3 within the set of points on the other side of e_0. this edge is noncrossing with both e_0 and e_1; either it is antiparallel to e_0 or to e_1, and we are done.

This implies that the existence of a diagonal in the tour and one of two possible choices of near-diagonals as the edge succeeding the diagonal in the tour determines the rest of the tour. Now it is straightforward to check that the resulting tour must look as in Fig. 8, concluding the proof. □

This motivates a heuristic analogous to the one for the MWMP. For simplicity, we call it CROSS'. See Fig. 10. From Theorem 4 it is easy to see that the following holds:

Corollary 1 *If the point set P is in convex position, then algorithm CROSS' determines the optimum.*

Algorithm CROSS': Heuristic solution for MTSP

Input: A set of points $P \in \mathbb{R}^2$.
Output: A tour of P.

1. Using a numerical method, find a point c that approximately minimizes the convex function $\min_{c \in \mathbb{R}^2} \sum_{p_i \in P} d(c, p_i)$.
2. Sort the set P by angular order around c. Assume the resulting order is $p_1, \ldots, p_n$.
3. For $i = 1, \ldots, n$, connect point p_i with point $p_{i+\frac{n}{2}-1}$.
Compute the resulting total length L.
4. Compute $D = \max_{i=1}^{n}[d(p_i, p_{i+\frac{n}{2}}) + d(p_{i+1}, p_{i+1+\frac{n}{2}})$ $-d(p_i, p_{i+\frac{n}{2}-1}) - d(p_{i+1}, p_{i+\frac{n}{2}})]$.
5. Choose the tour of length $L + D$ that arises by picking the two diagonals where the maximum in **4.** is attained.

Fig. 10. The heuristic CROSS'

3.3 No Integrality

As the example in Fig. 11 shows, there may be fractional optima for the subtour relaxation of the MTSP. The fractional solution consists of all diagonals (with weight 1) and all near-diagonals (with weight 1/2). It is easy to check that this solution is indeed a vertex of the subtour polytope, and that it beats any integral solution. (See [5] on this matter.) This implies that there is no simple analogue to Theorem 2 for the MWMP, and we do not have a polynomial method that can be used for checking the optimal solution for small instances.

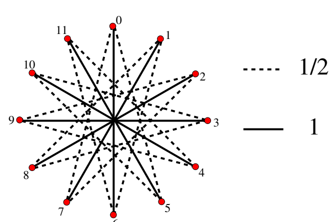

Fig. 11. A fractional optimum for the subtour relaxation of the MTSP.

3.4 Computational Experiments

The results are of similar quality as for the MWMP. See Table 2. Here we only give the results for the seven most interesting TSPLIB instances. Since we do not have an easy comparison with the optimum for instances of medium size, we give a comparison with the upper bound 2MAT, denoting twice the optimal solution for the MWMP. As before, this was computed by a network simplex

Table 2. Maximum TSP results for TSPLIB (top), uniform random (center), and clustered random instances (bottom)

Instance	CROSS' vs. FWP	time	CROSS' + 1h Lin-Ker	CROSS' vs. 2MAT
dsj1000	1.36%	0.05 s	1.10%	0.329%
nrw1379	0.23%	0.01 s	0.20%	0.194%
fnl4461	0.34%	0.12 s	0.31%	0.053%
usa13509	0.21%	0.63 s	0.19%	-
brd14051	0.67%	0.46 s	0.64%	-
d18512	0.15%	0.79 s	0.14%	-
pla85900	0.03%	3.87 s	0.03%	-
1000	0.04%	0.06 s	0.02%	0.02%
3000	0.02%	0.16 s	0.01%	0.00%
10000	0.01%	0.48 s	0.00%	-
30000	0.00%	1.47 s	0.00%	-
100000	0.00%	5.05 s	0.00%	-
300000	0.00%	15.60 s	0.00%	-
1000000	0.00%	54.00 s	0.00%	-
3000000	0.00%	160.00 s	0.00%	-
1000c	2.99%	0.05 s	2.87%	0.11 %
3000c	1.71%	0.15 s	1.61%	0.26 %
10000c	3.28%	0.49 s	3.25%	-
30000c	1.63%	1.69 s	1.61%	-
100000c	2.53%	5.51 s	2.52%	-
300000c	1.05%	17.80 s	1.05%	-

method, exploiting the integrality result for planar MWMP. The results show that here, too, most of the remaining gap lies on the side of the upper bound.

Table 3 shows an additional comparison for TSPLIB instances of moderate size. Shown are (1) the tour length found by our fastest heuristic; (2) the relative gap between this tour length and the fast upper bound; (3) the tour length found with additional Lin-Kernighan; (4) "optimal" values computed by using the Concorde code[2] for solving Minimum TSPs to optimality; (5) and (6) the two versions of our upper bound; (7) the maximum version of the well-known Held-Karp bound.

In order to apply Concorde, we have to transform the MTSP into a Minimum TSP instance with integer edge lengths. As the distances for geometric instances are not integers, it has become customary to transform distances into integers by rounding to the nearest integer. When dealing with truly geometric instances, this rounding introduces a certain amount of inaccuracy on the resulting optimal value. Therefore, Table 3 shows two results for the value OPT: The smaller one is the true value of the "optimal" tour that was computed by Concorde for the rounded distances, the second one is the value obtained by

[2] That code was developed by Applegate, Bixby, Chvátal, and Cook and is available at `http://www.caam.rice.edu/~keck/concorde.html`.

re-transforming the rounded objective value. As can be seen from the table, even the tours constructed by our near-linear heuristic can beat the “optimal” value, and the improved heuristic value almost always does. This shows that our heuristic approach yields results within a widely accepted margin of error; furthermore, it illustrates that thoughtless application of a time-consuming “exact” methods may yield a worse performance than using a good and fast heuristic. Of course it is possible to overcome this problem by using sufficiently increased accuracy; however, it is one of the long outstanding open problems on the Euclidean TSP whether computations with a polynomially bounded number of digits in terms of n suffices for this purposes. This amounts to deciding whether the Euclidean TSP is in NP. See [14].

The Held-Karp bound (which is usually quite good for Min TSP instances) can also be computed as part of the CONCORDE package. However, it is relatively time-consuming when used in its standard form: It took 20 minutes for instances with $n \approx 100$, and considerably more for larger instances. Clearly, this bound should not be the first choice for geometric MTSP instances.

Table 3. Maximum TSP results for small TSPLIB instances: Comparing CROSS’ and FWP with other bounds and solutions

Instance	CROSS’	CROSS’ vs. FWP	CROSS’ + Lin-Ker	OPT via CONCORDE	FWP’	FWP	Held-Karp bound
eil101	4966	0.15%	4966	[4958, 4980]	4971	4973	4998
bier127	840441	0.16%	840810	[840811, 840815]	841397	841768	846486
ch150	78545	0.12%	78552	[78542, 78571]	78614	78638	78610
gil262	39169	0.05%	39170	[39152, 39229]	39184	39188	39379
a280	50635	0.13%	50638	[50620, 50702]	50694	50699	51112
lin318	860248	0.09%	860464	[860452, 860512]	860935	861050	867060
rd400	311642	0.05%	311648	[311624, 311732]	311767	311767	314570
fl417	779194	0.18%	779236	[779210, 779331]	780230	780624	800402
rat783	264482	0.00%	264482	[264431, 264700]	264492	264495	274674
d1291	2498230	0.06%	2498464	[2498446, 2498881]	2499627	2499657	2615248

Acknowledgments

We thank Jens Vygen and Sylvia Boyd for helpful discussions, and Joe Mitchell for pointing out his paper [18].

References

1. D. Applegate, R. Bixby, V. Chvátal, and W. Cook. On the Solution of Traveling Salesman Problems. *Doc.Math.J.DMV*, Extra Volume ICM III (1998), 645–656.
2. S. Arora. Polynomial time approximation schemes for Euclidean TSP and other geometric problems. *Proc. 37th Annu. IEEE Sympos. Found. Comput. Sci.*, (1996), 2–11.
3. C. Bajaj. The algebraic degree of geometric optimization problems. *Discrete and Computational Geometry*, **3** (1988), 177–191.

4. A. I. Barvinok, D. S. Johnson, G. J. Woeginger, and R. Woodroofe. The Maximum TSP under polyhedral norms. *Proc. IPCO 1998*, Springer LNCS 1412, 195–201.
5. S. Boyd and W. R. Pulleyblank. Optimizing over the subtour polytope of the traveling salesman problem. *Mathematical Programming*, **49** (1991), 163–187.
6. W. Cook and A. Rohe. Computing minimum-weight perfect matchings. *INFORMS Journal on Computing* **11** (1999) 138–148. .
7. J. Edmonds. Paths, trees, and flowers. *Can. J. Mathematics*, **17** (1965), 449–467.
8. J. Edmonds. Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards (B)*, **69** (1965), 125–130.
9. S. P. Fekete. Simplicity and hardness of the Maximum TSP under geometric distances. *Proc. SODA 1999*, 337–345.
10. S. P. Fekete and H. Meijer. On Minimum Stars, Minimum Steiner Stars, and Maximum Matchings. *Discrete and Computational Geometry*, **23** (2000), 389–407.
11. H. N. Gabow. Data Structures for weighted matching and nearest common ancestors with linking. *Proc. SODA 1990*, 434–443.
12. H. W. Kuhn. A note on Fermat's problem. *Mathematical Programming*, **4** (1973), 98–107.
13. A. Löbel. Solving large-scale real-world minimum-cost flow problems by a network simplex method. *ZIB Preprint SC 96-7*, 1996.
14. D. S. Johnson and C. H. Papdimitriou. Computatonal complexity. Chapter 3 in: E. Lawler, J. Lenstra, A. Rinnooy Kan, D. Shmoys, eds., *The Traveling Salesman Problem*. Wiley, New York, 1985.
15. K. Mehlhorn and G. Schäfer. Implementation of $O(nm \log n)$ weighted matchings–The power of data structures. *Proc. WAE 2000*, Springer LNCS, 2000.
16. J. S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems. *SIAM J. Comput.*, **28** (1999), 1298–1309.
17. A. Rohe. Parallele Heuristiken für sehr große Traveling Salesman Probleme. *Diploma thesis*, Universität Bonn, 1997, 33–49.
18. A. Tamir and J. S. B. Mitchell. A maximum b-matching problem arising from median location models with applications to the roomates problem. *Mathematical Programming*, **80** (1998), 171–194.
19. P. M. Vaidya. Geometry helps in matching. *SIAM J. Comput.*, **18** (1989), 1201–1225.
20. K. R. Varadarajan. A divide-and-conquer-algorithm for min-cost perfect matching in the plane. *Proc. Found. Comp. Science*, **39** (1998), 320–329.
21. E. Weiszfeld. Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tôhoku Mathematics Journal*, **43** (1937), 355–386.