Experiments on Curve Reconstruction

Ernst Althaus
and
Kurt Mehlhorn
and
Stefan Näher
and
Stefan Schirra

An instance of the curve reconstruction problem is a finite sample V of an unknown curve γ. The task is to connect the points in V in the order in which they lie on γ. Several new algorithms for the curve reconstruction problem have been proposed in recent years. All algorithms compute a subgraph of the Delaunay triangulation of V. Most algorithms make the decisions of which edges to put into the reconstructions locally and run in time $O(n \log n)$ worst case, one algorithm uses a global criterion and computes the traveling salesman tour of V. Our implementation of this algorithm has exponential worst case running time. Our experiments show that the TSP-algorithm is far superior with respect to reconstruction quality. In our experiments its running time was never more than 13 times the running time of the fastest of the other algorithms.

We also describe an experimental platform for curve reconstruction algorithms.

Categories and Subject Descriptors: F.2.2 [**Theory of Computation**]: Analysis of algorithms and problem complexity

General Terms: Algorithms, Design, Experimentation, Performance

Additional Key Words and Phrases: Curve Reconstruction

1. INTRODUCTION

An instance of the curve reconstruction problem is a finite sample V of an unknown curve γ. The task is to construct a graph $G(V, E)$ such that two points in V are connected by an edge of G iff the points are adjacent on γ. The graph G is called a polygonal reconstruction of γ. The curve reconstruction problem and the related surface reconstruction problem have received a lot of attention in the graphics and the computational geometry community. Figure 1 illustrates the curve reconstruction problem.

Address: Max-Planck-Insititut für Informatik, Im Stadtwald, 66123 Saarbrücken, Germany

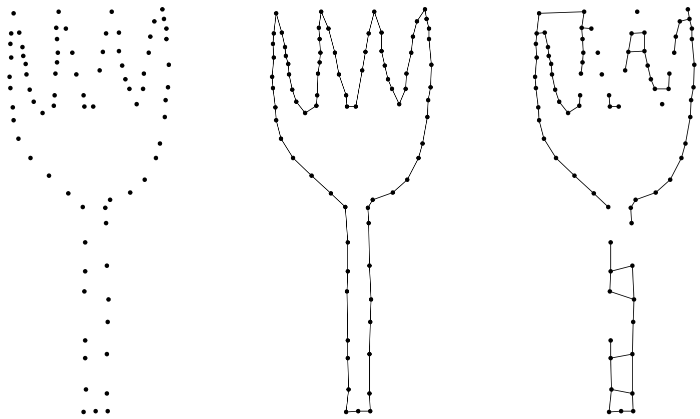

Fig. 1. Part (a) shows a finite set V of points, part (b) shows the Traveling Salesman tour of the points, and part (c) shows and the output of the reconstruction algorithm of Amenta, Bern, and Eppstein.

Many algorithms have been proposed for the curve reconstruction problem. We are interested in *reconstruction algorithms with guaranteed performance*, i.e., algorithms which provably solve the reconstruction problem under certain assumptions on γ and V. If the curve is closed, smooth, and uniformly sampled, several methods are known to work ranging over minimum spanning tree [FG94], α-shapes [BB97; EKS83], β-skeleton [KR85], and r-regular shapes [Att97]. A survey on these techniques appears in [Ede98]. The case of non-uniformly sampled closed curves was first treated successfully by Amenta, Bern and Eppstein [ABE98] and subsequently improved algorithms such as [DK99; Gol99] appeared. Open non-uniformly sampled curves were treated in [DMR99]. Giesen [Gie99a] recently obtained the first result for uniformly sampled non-smooth curves. He considered the class of *benign semi-regular curves.* An (open or closed) curve is semi-regular if a left and a right tangent exists in every point of the curve; the two tangents may however be different. A semi-regular curve is benign if the turning angle at every point of the curve is less than π, see Figure 2. Giesen showed that the Traveling Salesman path, tour respectively, of the sample set V solves the curve reconstruction problem for uniformly sampled benign open, closed respectively, semi-regular curves. Althaus and Mehlhorn [AM99] strengthened his result in two directions: They showed that the TSP solves the reconstruction problem even for non-uniformly sampled benign semi-regular curves and that, in the context of curve reconstruction, the TSP can be constructed in polynomial time.

In this paper we describe a testbed for curve reconstruction algorithms and report about an experimental evaluation of the curve reconstruction algorithms of Amenta, Bern, and Eppstein (ABE), Dey and Kumar (DK), Gold (Gold), Dey, Mehlhorn, and Ramos (DMR), and the TSP-algorithm. We also report about experimental results with some TSP-heuristics.

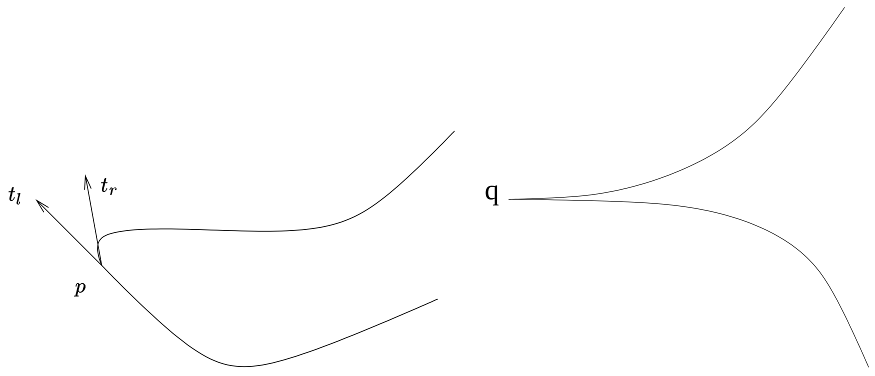

Fig. 2. Two semi-regular curves, one benign and one not. In the left curve, the two tangents t_l and t_r at a point p of the curve are shown. The turning angle at p is $\alpha < \pi$. In the curve on the right the turning angle at q is π.

2. ALGORITHMS

We included the following algorithms in our experimental study.

—Amenta-Bern-Eppstein [ABE98]
—Dey-Kumar [DK99]
—Gold [Gol99]
—Dey-Mehlhorn-Ramos [DMR99]
—TSP-reconstruction [Gie99a; AM99]

We refer to the first four algorithms as *local filtering algorithms*. We briefly describe the algorithms and their implementations.

The ABE algorithms first computes the Voronoi diagram of V. It then construct the Delaunay triangulation D of $V \cup V'$, where V' is the set of vertices of the Voronoi diagram of V. It puts an edge of D into the reconstruction if both of its endpoints belong to V. It discards all other edges of D.

The DK algorithm operates in two phases. In the first phase it connects each point in V to its nearest neighbor in the Delaunay diagram of V. In the second phase it iterates over the points in V in arbitrary order. If a point p has degree one and pq is the unique edge incident to p, the algorithm considers the open halfspace H orthogonal to pq, not containing q, and having p in its boundary. If H contains a point in V, the edge pr where r is a nearest neighbor of p in H is added to the reconstruction.

Gold's algorithm computes the Delaunay diagram and the Voronoi diagram of V. Let e be an edge of the Delaunay triangulation and let f be the dual edge of the Voronoi diagram. The algorithm puts either e into the polygonal reconstruction or f into the medial axis. It puts e into the polygonal reconstruction, if the endpoint of f are on different sides of e and if the endpoints of f are outside the center ball of e.

All algorithms above have a worst-case running time of $O(n \log n)$.

The DMR algorithm is fairly complex and we not discuss it here. The algorithm has a worst-case running time of $O(n \log n)$, but our implementation has a worst-case running time of $O(n^2)$.

The TSP-algorithm computes the Traveling Salesman Tour of the sample points. Giesen [Gie99b] has shown that the tour is a subgraph of the Delaunay triangulation if the curve is uniformly sampled. Althaus and Mehlhorn [AM99] have shown that the tour can be found by solving the so-called subtour linear programon the edges of the Delaunay triangulation of V. The LP uses a variable x_{uv} for every edge uv of the Delaunay triangulation and minimizes $\sum c_{uv}x_{uv}$, where c_{uv} is the Euclidean distance of u and v subject to the constraints:

$$\begin{aligned}\sum_{v\in V\setminus u} x_{uv} &= 2 \text{ for all } u \in V\\ \sum_{u\in V', v\notin V'} x_{uv} &\geq 2 \text{ for } V' \subset V, \emptyset \neq V' \neq V\\ x_{uv} &\geq 0\\ x_{uv} &\leq 1\end{aligned}$$

Althaus and Mehlhorn show that the optimal solution of the Subtour LP is guaranteed to be integral (and hence is equal to the TSP) if V is a dense sample of a benign semi-regular curve. The Subtour-LP can be solved in polynomial time by use of the Ellipsoid method, see [Sch86]. We use the simplex method and the cutting plane framework. One starts with the LP consisting only of the degree constraints and then solves a sequence of LPs. In each iteration one checks whether the solution to the current LP satisfies all subtour elimination constraints and, if not, one adds a violated cut constraint to the LP. We use two strategies for finding violated cut-constraints. The first strategy is fast but approximative and the second strategy is slow but exact. For the first strategy we compute the connected components formed by the edges uv with $x_{uv} = 1$ and check the cut constraints for the connected components. We add all violated constraints. For the second strategy we compute a minimum cut in the graph with edge capacities x_{uv}.

If we refer to the TSP-algorithm, we think of the algorithm, which computes the optimal solution of the Subtour LP, and returns the tour, if the solution is unique and integral, and `FAILURE` otherwise. This algorithm could be implemented in polynomial time.

All our implementations are based on LEDA [LED; MN99]. All algorithms use the graph data type, the geometry kernels, and the Delaunay diagram algorithm. ABE and Gold's algorithms also use the Voronoi diagram algorithm and DMR uses a data structure for nearest neighbor queries. The TSP algorithm uses, in addition, the min-cut and the connected components algorithm, it uses SOPLEX [SoP] or CPLEX [CPL] to solve linear programs, and it uses ABACUS [ABA] for the administration of the linear program. All geometric programs can be run with either the rational or the floating point kernel of LEDA.

3. SELF-CONSISTENCY

All algorithms studied in this paper come with a guarantee of the form: If the curve γ is from a certain class of curves and the sample set V is sufficiently dense, the algorithm will reconstruct γ. It is not specified what the algorithm does, if the hypothesis of the theorem is not satisfied. The algorithm may either fail, i.e., indicate that it could not find a curve, or "invent" a curve. From a practical point of view this situation is unsatisfactory as a user has in general no way of distinguishing reconstruction from invention. The situation is aggravated by the fact that the sampling densities required by the theorems are quite high and that the algorithms tend to work for smaller densities and hence are likely to be used in situations not covered by the theorems. *It would be nice to have algorithms that never invent curves.*

A reconstruction algorithm is called *self-consistent* if is has the following property. On an input V it either outputs `FAILURE` or `SUCCESS`. In the latter case it also outputs a curve Γ passing through V such that for any sample V' from Γ with $V \subseteq V'$, it will also output Γ.

A reconstruction algorithm that is not self-consistent can change its mind if given additional sample points that seem to confirm the output of the algorithm.

THEOREM 1. *DMR and TSP are self-consistent, DK, ABE, and Gold are not self-consistent.*

PROOF. DMR is constructed to be self-consistent. For the DK, ABE, and Gold it is easy to come up with examples that show non-self-consistency. It remains to show self-consistency for TSP. We show that if the solution for the Subtour LP for a set V is unique and integral then the same is true for the Subtour LP for $V \cup \{z\}$ for any z on an edge of the integral solution of the Subtour LP of V (and different from all points in V).

The claim is a simple consequence of the splitting-off Lemma of Lovász: Consider an optimal solution of the Subtour LP for $V \cup \{z\}$. The value C of this solution is at most the length C_0 of the otimal tour of V (since the tour for V is also a tour for $V \cup \{z\}$). The splitting-off Lemma allows to construct a solution for the subtour LP for V from the solution for $V \cup \{z\}$. It implies the existance of a set of triples (e, f, r), where e and f are edges incident to z and r is a non-negative real, such that

—for each edge e incident to z the sum of the third components of all triples containing e is equal to x_e (the value of the edge e in the Subtour LP for $V \cup \{z\}$) and

—a solution for the Subtour LP for V can be obtained by modifying the solution for the Subtour LP for $V \cup \{z\}$ as follows: for each edge uv with $u \neq z$ and $v \neq z$ increase x_{uv} by the sum of the third components of all triples (uz, zv, r). Delete all edges incident to z.

For any edge $e = uv$ let y_e be the increase of x_e in this construction. The cost of the obtained solution is $C + \sum_{uv} y_{uv}(c_{uv} - c_{uz} - c_{zv})$. This cost is at most C (and hence at most C_0) since $c_{uv} \leq c_{uz} + c_{zv}$ by the triangle inequality. Since the solution for the Subtour LP for V is unique and is equal to the tour for V, the cost of the solution cannot be smaller than C_0 and hence for any edge $e = uv$ with $y_{uv} > 0$ we must have $c_{uv} = c_{uz} + c_{zv}$, i.e., z lies on the line segment $\overline{uv}$. Moreover, $y_e + x_e$ must be integral for every edge $e = uv$.

In the tour for V there is only one edge passing through z (since optimal tours are non-self-intersecting) and hence there can be only one edge uv with $y_{uv} > 0$. Thus our set of triples consists of a single triple (uz, zv, r) and since the degree

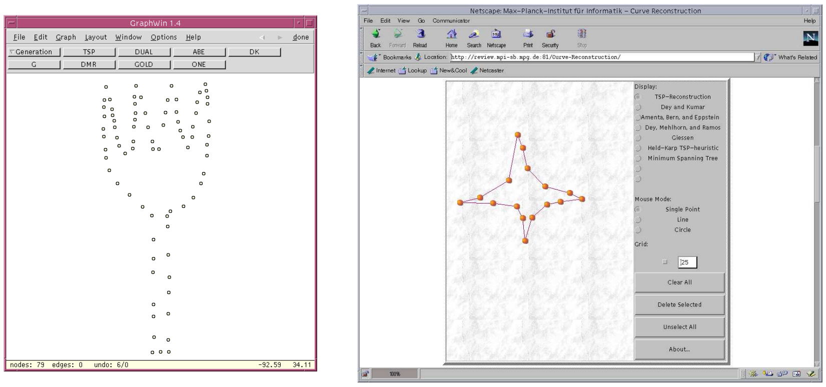

Fig. 3. A screen-shot of the testbed and the Internet demo.

constraint at z must be satisfied for the optimal solution of the Subtour LP for $V \cup \{z\}$ we conclude that $r = 1$. We conclude that the optimal solution of the subtour LP for $V \cup \{z\}$ is unique and integral. $\square$

4. THE TESTBED

Reproducibility of experiments is a major concern for experimental algorithms. We therefore provide our reconstruction algorithms and problem generators as a LEDA extension package. As part of the package we provide a literate programming document describing all the implementations. The package also contains a testbed that allows to experiment with reconstruction algorithms (not just ours) and visualizes the outcome of the reconstructions. Figure 3 shows a screen-shot of the testbed. The package will be available at `http://www.mpi-sb.mpg.de/LEDA/friends/leps.html`.

We also offer an Internet interface to our implementations. It is available at `http://review.mpi-sb.mpg.de:81/Curve-Reconstruction/`. It consists of a JAVA-applet that allows the user to construct problem instances and to select an algorithm. The applet contacts a server at the MPI to run the algorithm and displays the result of the reconstruction.

5. THE EXPERIMENTS

We performed experiments that compare the reconstruction quality and the running time of our implementations. All experiments were carried out with the floating point geometry kernel of LEDA. We used several curves for our experiments. Since the results are fairly consistent over the curves, we report only about the experiments for the curve shown in Figure 4. The curve is essentially a sinusoidal curve. We use p to denote the number of periods, n to denote the number of sample points and $\rho = n/p$ to denote the number of points per period.

5.1 Reconstruction Quality

In order to test the reconstruction quality of the different algorithms we used a fixed curve and varied the sampling density. More precisely, we used a fixed number n

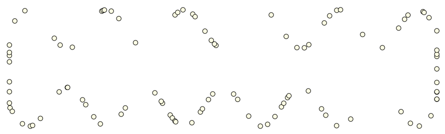

Fig. 4. Our sample curve (5 periods and 300 points).

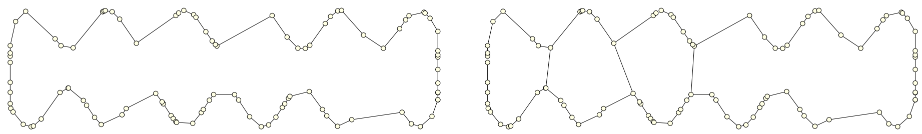

Fig. 5. A reconstruction with the TSP-algorithm and the algorithm of Dey and Kumar.

of points and varied the number p of periods in our curve. For each value of p we chose n random points on the curve. We repeated each experiment ten times. We measured the running time and in what percentage of the cases the algorithms found the reconstruction. The results are shown in Figure 6. We observe that the TSP-algorithm can work with much sparser samples than the other algorithms. Figures 1 and 5 show two reconstruction examples. Among the other algorithms, DMR does worst (this is to expected, since it is designed to reconstruct open and closed curves and hence omits more edges), DK is best, followed by Gold and ABE.

5.2 Running Time

In order to compare efficiency we performed two experiments. For varying sampling density and a fixed number of points, and for fixed sampling density and a varying number of points. The first experiment was described in the previous section. For the second experiment we used a curve with $\rho = 100$ points per period, varied the number n of points, and kept the number of periods at $p = n/100$. We used sampling densities for which all algorithms were able to solve the reconstruction problem. The results of the second experiment are shown in Figure 7.

All filtering algorithms start by computing the Delaunay triangulation of the sample set V. DK and Gold do linear additional work, DK selects the nearest of each points and then the nearest neighbor in the other halfspace for each point of degree one. Gold constructs the Voronoi diagram (= a linear time process, when the Delaunay diagram is available) and then performs a local test for the each pair of primal and dual edge. The running time of our Delaunay program [Dwy87] is $O(n)$ in the best case and $O(n \log n)$ in the worst case. For the instances in Figure 6 the running time improves as a function of the number of periods. This is due to the fact that the algorithm uses divide-and-conquer on the x-coordinates of the points and that the running time of the conquer step decreases as the number of periods increases (observe that the number of points contained in a vertical stripe decreases as the number of periods increases). In Figure 6 the running times of DK and Gold are shifts of the cost of the Delaunay computation.

The algorithms ABE and DMR use $O(n \log n)$ (DMR may use $O(n^2)$ additional

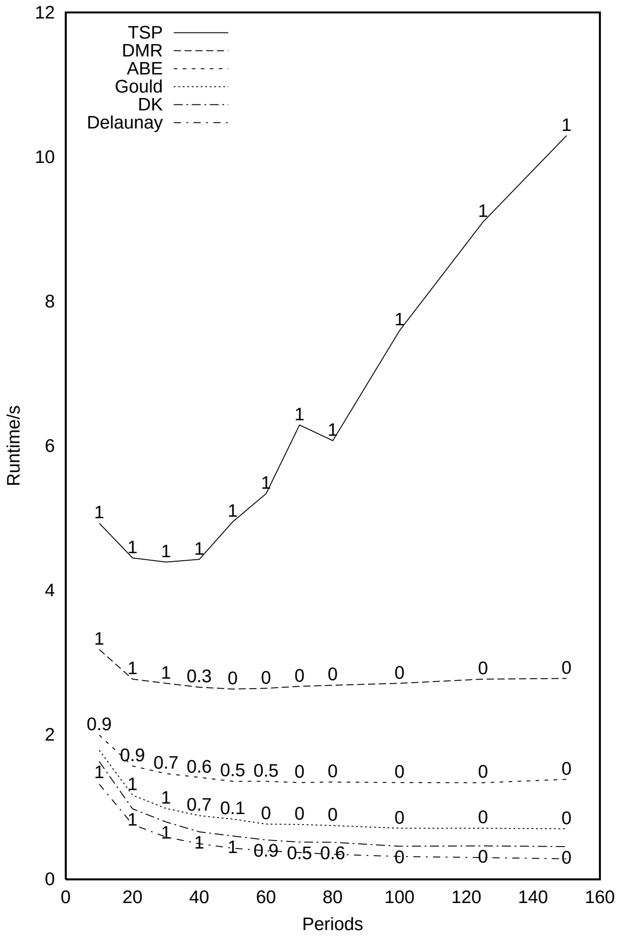

Fig. 6. The abscissa shows the number of periods and the ordinate shows the running time (average over ten runs). The numbers on the curves indicate the fraction of problems solved, e.g., 0.4 indicates that 40% of the problems were solved. For all experiments we used $n = 3000$ points. The curve Delaunay indicates the time of the Delaunay diagram computation.

time in the worst case, but we do not know of a worst-case example) additional time. ABE constructs the Delaunay triangulation of $V \cup V'$, where V' is the set of Voronoi vertices of V. The running time of the second Delaunay computation seems to be independent of the number of periods. DMR uses a data structure for proximity queries which has a running time of $\Theta(n \log n)$.

The running time of the TSP-algorithm also conforms to intuition. For fixed problem size the running time in addition to the time for computing the Delaunay triangulation decreases as a function of sampling density (= increases as a function of the number of periods), since the reconstruction problem becomes simpler as the sampling density increases. For sparse samples the number of phases of the cutting plane algorithm goes up. Moreover, the connected components heuristic is no longer able to find the required cuts. For fixed sampling density and varying n

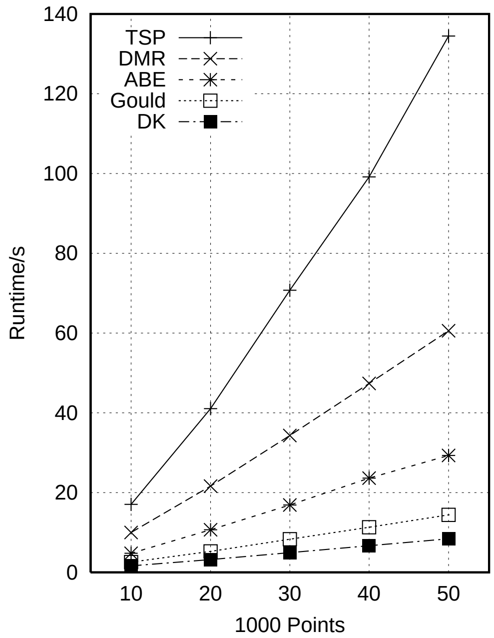

Fig. 7. The running times for a curve with $\rho = 100$ points per period and varying number of periods. We used $n = i \cdot 10^4$ sampling points for $i = 1, 2, \ldots, 5$. The running times of the algorithms follow approximately the relation DK : Gold : ABE : DMR : TSP = 1 : 1.3 : 3 : 6 : 13.

the running time grows linearly in n. The explanation is that a small (typically 5 or 6) iterations of the cutting plane algorithm suffice for sampling densities where the other algorithms are able to find solutions, that the linear time heuristic for finding cuts is sufficient, and that the expected running time of the simplex algorithm is linear in the number of constraints and variables.

5.3 Robustness

The DMR and ABE implementations suffer from robustness problems when run with the floating point kernel of LEDA. We discuss the issue for the ABE algorithm. The ABE first constructs the Voronoi diagram of V and then the Delaunay triangulation of $V \cup V'$, where V' is the set of Voronoi vertices of V. If V contains nearly co-circular points (which might be a frequent situation in the curve reconstruction scenario), V' contains clusters of points and this makes the construction of the Delaunay diagram of $V \cup V'$ a difficult task.

The robustness problems can be removed by switching to the rational kernel of LEDA which guarantees that all geometric predicates are evaluated correctly (and uses for a floating point filter for speed). The switch to the rational kernel can cause a significant increase in running time. It is the greatest increase that we have ever experienced. The reason is that the construction of the Delaunay diagram of $V \cup V'$ requires many exact tests, if V contains many co-circular points.

We propose to overcome the robustness problem in the ABE algorithm by rounding the Voronoi vertices to a fine grid. This will identify Voronoi vertices that lie

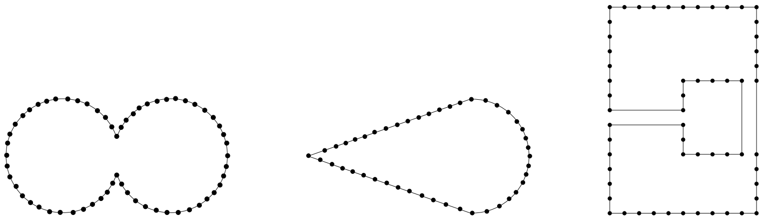

Fig. 8. Part (a) shows an example where the heuristics NM, NA, and FA fail, part (b) an example where CH, NN, and MF fail. Part (c) shows a three optimal tour different from the traveling salesman tour.

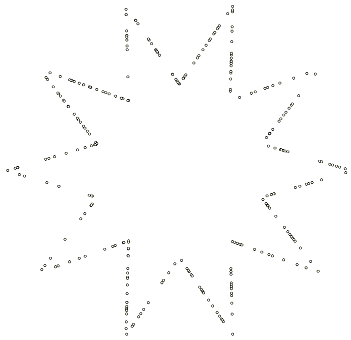

Fig. 9. A star with 10 spikes and 300 sample points.

close together.

6. TSP HEURISTICS

We have implemented some TSP- and tour improvement heuristics. The TSP heuristics are an iterative version to compute the Held-Karp Bound, Christofides Algorithm (CH), the nearest merger algorithm (NM), the nearest neighbor heuristic (NN), the multiple fragment greedy heuristic (MF), the nearest addition algorithm (NA) and the farthest addition algorithm (FA). In addition we have implemented the *2-opt* and *3-opt* improvement heuristics. For details of these algorithms we refer to [Be92]

No heuristic reconstructs the tulip of Figure 1 correctly. Furthermore no heuristic solves the reconstruction problem, i.e., for all TSP-heuristics mentioned above there is a closed curve and an arbitrary dense sample such that the algorithm does not find the optimal tour and hence fails to reconstruct. For the improvement heuristics an initial (bad) tour is also chosen. Figure 8 shows examples on which the heuristics fail.

The heuristics NM, NA and FA fail even on smooth curves. They do not reconstruct any of the curves considered in Section 5. The remaining heuristics (CH, NN, and MF) reconstruct uniformly sampled smooth curves (even the minimum spanning tree does) but fail by construction in sharp corners.

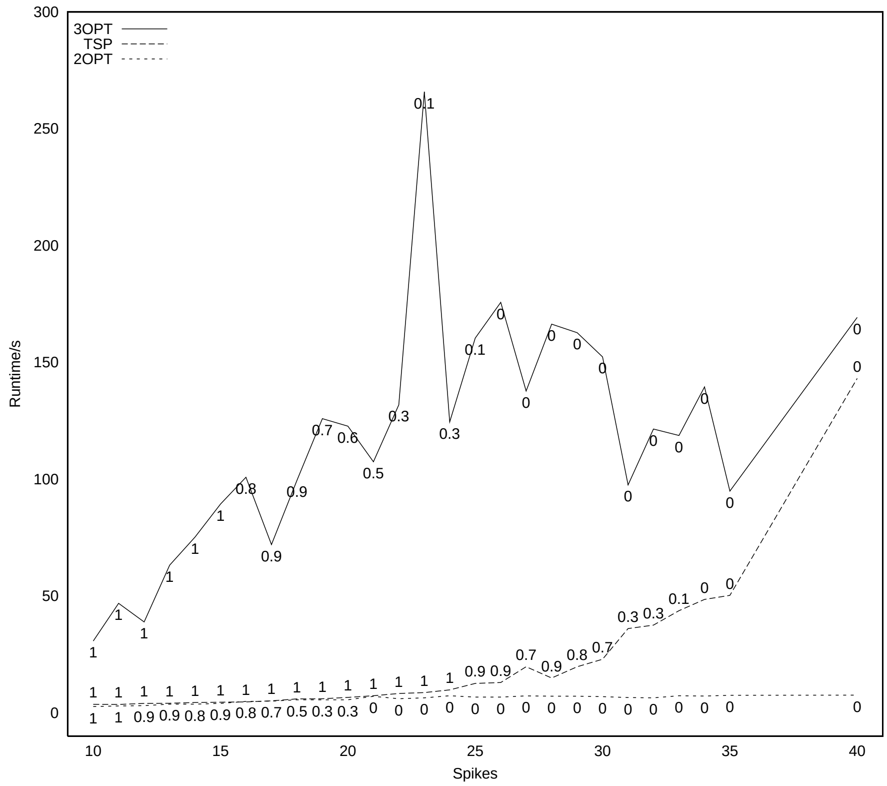

Fig. 10. The abscissa shows the number of spikes of the start and the ordinate shows the running time (average over ten runs). The numbers on the curves indicate the fraction of problems solved. For all experiments we used $n = 3000$ points.

The improvement heuristics do quite well. We give experimental results for the curve shown in Figure 9, a simple star with a varying number of spikes. The results are shown in Figure 10. For dense samples both heuristics succeed, 3-opt can cope with smaller sampling density than 2-opt and the TSP-algorithm can cope with even smaller sampling density. This was to be expected. For sampling densities where the heuristics worked, the TSP-algorithms is almost as fast as 2-opt and considerably faster than 3-opt. We find this surprising. We have to admit however, that our implementations of the improvement heuristics are fairly crude.

REFERENCES

[ABA] ABACUS–A Branch And CUt System. `www.informatik.uni-koeln.de/ls_juenger/projects/abacus.html`.

[ABE98] N. Amenta, M. Bern, and D. Eppstein. The crust and the β-skeleton: Combinatorial curve reconstruction. *Graphical Models and Image Processing*, pages 125–135, 1998.

[AM99] E. Althaus and K. Mehlhorn. Polynomial time TSP-based curve reconstruction. accepted for SODA 2000, 1999.

[Att97] D. Attali. r-regular shape reconstruction from unorganized points. In *Proceedings of the 13th Annual ACM Symposium on Computational Geometry (SCG'97)*, pages 248–253, 1997.

[Be92] J. Bentley Fast Algorithms for Geometric Traveling Salesman Problems In *ORSA Jornal on Computing*, 4, 4: 387–411, 1992.

[BB97] F. Bernardini and C.L. Bajaj. Sampling and reconstructing manifolds using α-shapes. In *Proceedings of the 9th Canadian Conference on Computational Geometry (CCCG'9 7)*, pages 193–198, 1997.

[CCPS98] W.J. Cook, W.H. Cunningham, W.R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. John Wiley & Sons, Inc, 1998.

[CPL] CPLEX. `www.cplex.com`.

[DK99] T.K. Dey and P. Kumar. A simple provable algorithm for curve reconstruction. In *Proceedings of the 10th ACM-SIAM Symposium on Discrete Algorithms (SODA'99)*, pages 893–894, 1999.

[DMR99] T.K. Dey, K. Mehlhorn, and E.A. Ramos. Curve reconstruction: Connecting dots with good reason. In *Proceedings of the 15th Annual ACM Symposium on Computational Geometry (SCG'99)*, pages 197–206, 1999. `www.mpi-sb.mpg.de/~mehlhorn/ftp/cure.ps` .

[Dwy87] R.A. Dwyer. A faster divide-and-conquer algorithm for constructing Delaunay triangulations. *Algorithmica*, 2:137–151, 1987.

[Ede98] H. Edelsbrunner. Shape reconstruction with the Delaunay complex. In *LATIN'98: Theoretical Informatics*, volume 1380 of *Lecture Notes in Computer Science*, pages 119–132, 1998.

[EKS83] H. Edelsbrunner, D.G. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):71–78, 1983.

[FG94] L.H. Figueiredo and J.M. Gomes. Computational morphology of curves. *Visual Computer*, 11:105–112, 1994.

[Gie99a] J. Giesen. Curve reconstruction, the TSP, and Menger's theorem on length. In *Proceedings of the 15th Annual ACM Symposium on Computational Geometry (SCG'99)*, pages 207–216, 1999.

[Gie99b] J. Giesen. Regularity Considerations in Curve Reconstruction with Delaunay Complex. unpuplished, 1999.

[Gol99] C. Gold. Crust and anti-crust: A one-step boundary and skeleton extraction algorithm. In *Proceedings of the 15th Annual ACM Symposium on Computational Geometry (SCG'99)*, pages 189–196, 1999.

[KR85] D.G. Kirkpatrick and J.D. Radke. A framework for computational morphology. *Computational Geometry*, pages 217–248, 1985.

[LED] LEDA (Library of Efficient Data Types and Algorithms). `www.mpi-sb.mpg.de/LEDA/leda.html`.

[MN99] K. Mehlhorn and S. Näher. *The LEDA Platform for Combinatorial and Geometric Computing*. Cambridge University Press, 1999.

[Sch86] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, 1986.

[SoP] SoPlex. `www.zib.de/Optimization/Software/Soplex`.