I N F O R M A T I K

A Note on the Smoothed Complexity of the Single-Source Shortest Path Problem

Guido Schäfer

MPI–I–2003–1–018 October 2003

FORSCHUNGSBERICHT RESEARCH REPORT

M A X - P L A N C K - I N S T I T U T
F Ü R
I N F O R M A T I K

Stuhlsatzenhausweg 85 66123 Saarbrücken Germany

Author's Address

Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany
Email: ⟨*schaefer@mpi-sb.mpg.de*⟩

Acknowledgements

The author was supported by DFG graduated studies program "Quality Guarantees for Computer Systems" (GK 623), Saarbrücken.

Abstract

Banderier, Beier and Mehlhorn [BBM03] showed that the single-source shortest path problem has smoothed complexity $O(m + n(K - k))$ if the edge costs are K-bit integers and the last k least significant bits are perturbed randomly. Their analysis holds if each bit is set to 0 or 1 with probability $\frac{1}{2}$.

We extend their result and show that the same analysis goes through for a large class of probability distributions: We prove a smoothed complexity of $O(m + n(K - k))$ if the last k bits of each edge cost are replaced by some random number chosen from $[0, \ldots, 2^k - 1]$ according to some *arbitrary* probability distribution $\mathcal{P}$ whose expectation is not too close to zero. We do not require that the edge costs are perturbed independently. The same time bound holds even if the random perturbations are heterogeneous. If $k = K$ our analysis implies a linear average case running time for various probability distributions. We also show that the running time is $O(m + n(K - k))$ with high probability if the random replacements are chosen independently.

Keywords

single-source shortest path problem, average case, smoothed analysis

The results presented in this note are straight-forward extensions of the smoothed analysis for the single-source shortest path problem by Banderier, Beier and Mehlhorn [BBM03]. If the reader is familiar with the average case analysis for single-source shortest path algorithms, in particular with the results of Meyer [Mey02, Mey03] and Goldberg [Gol02], and also knows the work of Banderier et al. [BBM03] he/she might find this note little innovative.

The reasons why we decided to publish this note as a MPII research report are as follows. (i) This note was written down to a large extent when we found out that similar ideas were already used in [Mey02, Mey03] and [Gol02]. (ii) We believe that it makes sense to summarize the smoothed complexity result of Banderier et al. and moreover that it is important to clearly state possible extensions, thereby identifying essential properties of the underlying smoothing distribution for the analysis to go through.

1 Introduction

We consider the single-source shortest path (SSSP) problem with non-negative edge costs. Given a directed graph $G = (V, E)$, a non-negative cost function $c : E \to \mathbb{R}^+$, and a designated source node s, the task is to compute shortest path distances from s to all other nodes. We use n and m to denote the number of nodes and edges of G, respectively.

The SSSP problem is well studied both from a theoretical and from a practical point of view. Dijkstra's algorithm [Dij59] solves the SSSP problem in time $O(m + n \log n)$ if Fibonacci heaps [FT87] are used as the underlying priority queue data structure.

Average Case Complexity of SSSP. Meyer [Mey02, Mey03] was the first who presented an algorithm with linear average case running time $O(m + n)$ for *arbitrary* directed graphs and uniform (real) edge costs in $[0, 1]$. The algorithm itself can be implemented to achieve a worst case running time of $O(mn)$. (A parallel execution of any other SSSP algorithm having a worst case bound of $\mathcal{C}$ reduces this bound to $\mathcal{C}$ while the $O(n + m)$ average case complexity is retained.) Meyer also shows that his algorithm has expected linear running time if the edge costs are drawn from a probability distribution over $[0, 1]$, whose cumulative distribution function $F(x)$ satisfies (i) $F(0) = 0$, and (ii) $F'(0)$ is bounded from above by some constant. That is, $F(x)$ is slowly increasing from 0. Moreover, Meyer proves that his algorithm has linear running time with high probability if the edge costs are chosen uniformly and independently at random. The high probability argument can be extended to distributions that satisfy (i) and (ii) above. This is already implicit but not made explicit in [Mey02, Mey03].

Somewhat later, Goldberg [Gol02] presented a simple implementation of Dijkstra's algorithm that has a linear average case running time if the edge costs are integers chosen uniformly from $[1, \dots, 2^K]$. The worst case running time of the algorithm is $O(m + nK)$. Goldberg remarks that expected linear running time is achieved for any distribution satisfying that for each edge e: $\mathbf{E}[K - \log(c(e))] = O(1)$. He does not characterize the distributions satisfying this property. Goldberg proves that the algorithm has linear running time with high probability if the edge costs are chosen uniformly and independently.

Smoothed Analysis and its Application to SSSP. The concept of *smoothed analysis* was proposed by Spielman and Teng [ST01] as a hybrid between average case and worst case analysis to explain the success of algorithms that are known to work well in practice while presenting a poor worst case performance. The basic idea is to randomly perturb the input instances and to analyze the performance of the algorithm on the perturbed instances.

More formally, the smoothed complexity of an algorithm is defined as follows. Let $\bar{I}$ be an adversarial input instance. We perturb, or *smoothen*, $\bar{I}$ by some probability distribution $\mathcal{P}$. Let $N(\bar{I})$ denote the set of all smoothed instances that are obtainable from $\bar{I}$ by the perturbation. Then, the smoothed complexity is defined as the supremum over all $\bar{I}$ of the expected running time of the algorithm over $N(\bar{I})$:

$$\sup_{\bar{I}} \mathbf{E}_{I \stackrel{\mathcal{P}}{\leftarrow} N(\bar{I})} [T(I)] \tag{1}$$

Intuitively, the smoothed complexity of an algorithm is small if the worst case instances are isolated in the (instance $\times$ time) space. Spielman and Teng's striking result was to show that the smoothed complexity of the simplex method with a certain pivot rule and by perturbing the coefficients with a normal distribution is polynomial. In a series of later papers smoothed analysis was successfully applied to characterize the time complexity of other algorithms. (For a current list of papers related to smoothed analysis see ⟨*http://www-math.mit.edu/~spielman/SmoothedAnalysis*⟩.)

Let $c : E \to [0, \dots, 2^K - 1]$ be an arbitrary cost function on the edges of the graph G, i.e., we assume a K-bit representation of the edge costs. We assume that G and the cost function c are given by an adversary. We use a *partial bit randomization model*, where the adversarial edge costs are smoothed by setting the k least significant bits of each edge at random. For this model, Banderier, Beier and Mehlhorn [BBM03] showed that the algorithm of Goldberg [Gol02] has a smoothed complexity of $O(m + n(K - k))$ if each bit is set independently to 0 or 1 with probability $\frac{1}{2}$. Put differently, the last k bits are replaced by some number chosen uniformly at random from $[0, \dots, 2^k - 1]$. A consequence of the analysis is a linear average case running time of $O(m + n)$ if $k = K$.

Our Contribution. Here, we extend this setting. The last k bits of each edge cost $c(e)$ are replaced by some random number that is chosen according to an *arbitrary* probability distribution $\mathcal{P}$ over $[0, \dots, 2^k - 1]$. We show that the expected running time remains $O(m + n(K - k))$ for a large class of *permissible* probability distributions. Roughly speaking, a probability distribution $\mathcal{P}$ is permissible if its expectation is not too close to zero. Moreover, the same result holds even if the random replacements are heterogeneous, i.e., if each edge e has its own perturbation distribution $\mathcal{P}(e)$. We do not require independence of the random replacements. As a consequence, for $k = K$ we obtain a linear average case running time for various probability distributions. We also show that the same holds with high probability if the random replacements are chosen independently.

2 Goldberg's Adapted Multi-Level Bucket Algorithm

We introduce some terminology and review Dijkstra's algorithm. Then, we discuss a special priority queue, called the *multi-level bucket priority queue*, and outline the improvement of Goldberg.

We use $\delta(v)$ to denote the distance label of a node v, i.e., $\delta : V \to \mathbb{N}_0$. Initially, $\delta(s) := 0$ and $\delta(v) := \infty$ for all $v \in V$, $v \neq s$. A node is either *settled* or *unsettled*. A node is settled if its distance label is exact. Initially, all nodes are unsettled. All (finite) distance labels of unsettled nodes are kept in a priority queue, PQ. In each iteration, we remove an unsettled node with minimum distance label from PQ, relax all outgoing edges, and declare the node to be settled. The algorithm terminates if the priority queue is empty.

In a multi-level bucket priority queue distances of unsettled nodes are stored with respect to the minimum distance $\mathcal{M}$ that was removed last. The priority queue PQ is organized in

$K+1$ levels $L_0, \ldots, L_K$. We use $\mathcal{M}_{[0,K]}$ and $\delta(v)_{[0,K]}$ to denote the $K+1$ least significant bits of $\mathcal{M}$ and $\delta(v)$, respectively. The distance $\delta(v)$ of an unsettled node v is stored in level L_i if the most significant bit in which $\delta(v)_{[0,K]}$ differs from $\mathcal{M}_{[0,K]}$ is i. (Here, the least significant bit is bit 0, the second least significant bit is bit 1, and so on.) For example, $\delta(v)$ is stored in L_0 if $\delta(v)_{[0,K]}$ differs from $\mathcal{M}_{[0,K]}$ only in the least significant bit. We store $\delta(v)$ in L_0 if $\delta(v)$ and $\mathcal{M}$ are identical.

Operations `insert` and `decrease-key` take time $O(1)$. After a `delete-min` operation $\mathcal{M}$ might change and therefore some of the distances in PQ may have to be redistributed to the appropriate levels. If the (new) minimum $\mathcal{M}$ is in level L_i then only the distances in L_i need to be redistributed. Distances in higher levels are not affected. Moreover, distances always move to lower levels. The running time of Dijkstra's algorithm with the multi-level bucket priority queue is

$$O\left(m + n + \sum_{v \in V} M(v)\right), \tag{2}$$

where $M(v)$ denotes the number of times a node v is moved to a lower level. Clearly, in the worst case $M(v) = K+1$ for all $v \in V$ and we obtain a worst case running time of $O(m + nK)$. Dijkstra's algorithm together with the multi-level bucket priority queue is due to Denardo and Fox [DF79].

Goldberg [Gol02] combines the above algorithm with an idea which was before successfully used by Thorup [Tho99]. The idea is simple: Assume we have a lower bound $\mathcal{M}$ on the distances of all unsettled nodes. (Observe that the distance $\mathcal{M}$ removed last from PQ has this property.) Consider any unsettled node v and let $C(v)$ denote the minimum cost over all incoming edges $(u, v) \in E$ of v. If $\mathcal{M} + C(v) \geq \delta(v)$ we can conclude that the distance label $\delta(v)$ of v is exact. That is, we can alter the above algorithm as follows: Whenever we encounter a node v that satisfies the above condition, we relax all outgoing edges of v, and declare v as *settled*. As a consequence, a node v is never inserted or moved to a level L_i with $i \leq \lfloor \log(C(v) + 1) \rfloor - 1$. Thus, (2) reduces to

$$O\left(m + n + \sum_{v \in V} \big(K - \log(C(v) + 1)\big)\right). \tag{3}$$

In the worst case, the running time of the algorithm is still $O(m + nK)$. Goldberg proved that if the edge costs are chosen uniformly at random from $[1, \ldots, 2^K]$ then $\mathbf{E}[K - \log(C(v) + 1)] = O(1)$. That is, the above algorithm has a linear average case running time of $O(m + n)$ for uniform edge costs. Moreover, he proves that the same result holds with high probability. The high probability argument is involved.

3 Smoothed Analysis of SSSP

We first discuss the smoothing model we use. Then, we review the result obtained by Banderier et al. After that, we present our extension of the analysis and the high probability argument.

3.1 Partial Bit Randomization Model

We perturb the cost according to the partial bit randomization model. For each edge $e \in E$, the *smoothed cost* $\bar{c}(e)$ of e is obtained from the original cost $c(e)$ by replacing the k least significant bits of the binary representation of $c(e)$ by some random number $\varepsilon(e)$ chosen from $[0, \ldots, 2^k - 1]$. It is natural to assume that we flip a fair coin for each bit. That is, we choose $\varepsilon(e)$ uniformly at random from $[0, \ldots, 2^k - 1]$. This is the model considered by Banderier et al. [BBM03]. Here, however, we allow $\varepsilon(e)$ to be chosen according to some arbitrary probability distribution $\mathcal{P}$. At first sight, this may seem odd — however, the advantage is that for $k = K$ we obtain complete random costs chosen according to $\mathcal{P}$ from $[0, \ldots, 2^K - 1]$.

More precisely, we define the smoothed cost $\bar{c}(e)$ of an edge $e \in E$ as

$$\bar{c}(e) := 2^k \left\lfloor \frac{c(e)}{2^k} \right\rfloor + \varepsilon(e), \text{ where } \varepsilon(e) \stackrel{\mathcal{P}}{\leftarrow} [0, \ldots, 2^k - 1].$$

3.2 Smoothed Complexity of SSSP

For each edge $e \in E$, let $Z(e)$ be the number of leading zeros (if regarded from the most to the least significant bit) in the binary representation of $\bar{c}(e)$. Similarly, let $X(e)$ be the number of leading zeros in the random part $\varepsilon(e)$ of $\bar{c}(e)$. We have $Z(e) \leq K - k + X(e)$ for each edge e. For each $v \in V$, we define $Z(v) := \max\{Z(e) : e = (u, v) \in E\}$, and $Z(v) := 0$ if v has no incoming edge. Then, $Z(v) \leq K - k + \max\{X(e) : e = (u, v) \in E\}$. We can express the running time (3) of Goldberg's algorithm on the perturbed edge costs $\bar{c}(e)$ as

$$O\left(m + n + \sum_{v \in V} Z(v)\right) = O\left(m + n(K - k) + \sum_{e \in E} X(e)\right).$$

We concentrate on analyzing $\sum_{e \in E} X(e)$.

The next theorem is essentially the one proven by Banderier et al. [BBM03].

Theorem 1. *If $\varepsilon(e)$ is chosen uniformly at random from $[0, \ldots, 2^k - 1]$ we obtain a smoothed complexity of $O(m + n(K - k))$.*

Proof. We have $\mathbf{P}[X(e) \geq i] = \left(\frac{1}{2}\right)^i$ for each i. Thus, $\mathbf{E}[X(e)] = \sum_{i=1}^{k} \left(\frac{1}{2}\right)^i \leq 1$. Therefore, $\mathbf{E}\left[\sum_{e \in E} X(e)\right] = \sum_{e \in E} \mathbf{E}[X(e)] \leq m$. $\square$

3.3 Extensions

For each edge e, let $\varepsilon(e)$ be chosen according to $\mathcal{P}$ from $[0, \ldots, 2^k - 1]$. We call a probability distribution $\mathcal{P}$ *permissible* if it satisfies that $\mathbf{E}[X(e)] \leq \alpha$ for some constant α. The next theorem is an obvious extension of Theorem 1.

Theorem 2. *Let $\mathcal{P}$ be a permissible probability distribution. For each edge $e \in E$, let $\varepsilon(e)$ be chosen according to $\mathcal{P}$ from $[0, \ldots, 2^k - 1]$. Then, we obtain a smoothed complexity of $O(m + n(K - k))$.*

Proof. $\mathbf{E}\left[\sum_{e \in E} X(e)\right] = \sum_{e \in E} \mathbf{E}[X(e)] \leq \alpha m$. $\square$

Observe that we do not require that the $\varepsilon(e)$'s are independent. Moreover, each $\varepsilon(e)$ could even be chosen from a different permissible probability distribution $\mathcal{P}(e)$, satisfying $\mathbf{E}[X(e)] \leq \alpha(e)$. We do not even need that $\alpha(e)$ is a constant for every e; it suffices that $\sum_{e \in E} \alpha(e) \leq \alpha m$ for some constant α.

Theorem 3. *For each edge $e \in E$, let $\varepsilon(e)$ be chosen from a probability distribution $\mathcal{P}(e)$ over $[0, \ldots, 2^k - 1]$ with $\mathbf{E}[X(e)] = \mu(e)$. If $\sum_{e \in E} \mu(e) \leq \alpha m$ for some constant α, then we obtain a smoothed complexity of $O(m + n(K - k))$.*

Proof. $\mathbf{E}\left[\sum_{e\in E} X(e)\right] = \sum_{e\in E} \mu(e) \leq \alpha m$. □

If the $\varepsilon(e)$'s are chosen independently, we can obtain a high probability result.

Theorem 4. *For each edge $e \in E$, let $\varepsilon(e)$ be chosen independently from a probability distribution $\mathcal{P}(e)$ over $[0,\ldots,2^k-1]$ with $\mathbf{E}[X(e)] = \mu(e)$. If $\sum_{e\in E}\mu(e) \leq \alpha m$ for some constant α, then with high probability the running time is $O(m + n(K-k))$.*

Proof. It is sufficient to show that with high probability $\sum_{e\in E} X(e) \leq c_1 m$, for some constant c_1. Define $\mu := \mathbf{E}\left[\sum_{e\in E} X(e)\right] = \sum_{e\in E}\mu(e)$. We have $\mu \leq \alpha m$. We use the Hoeffding bound (see Appendix A). For a constant $c_1 > \mathrm{e}\alpha$, we obtain

$$\mathbf{P}\left[\sum_{e\in E} X(e) \geq c_1 m\right] \;\leq\; \left[\left(\frac{c_1 m}{\mathrm{e}\mu}\right)^{-c_1 m} \mathrm{e}^{-\mu}\right]^{1/k} \;\leq\; \mathrm{e}^{-\mu/k}.$$

The latter term is less than $1/m$ if $k < \mu/\log(m)$. □

Clearly, the conditions of Theorem 4 are satisfied if $\mathcal{P}(e)$ is permissible for each edge $e \in E$. In the next two lemmas, we characterize permissible probability distributions.

Lemma 1. *Let $\mathcal{P}(e)$ be such that $\mathbf{P}\left[\varepsilon(e) < 2^{k-i}\right] \leq p^i$, for some constant $p < 1$. Then, $\mathcal{P}(e)$ is permissible.*

Proof. Observe that for each $i \in [0,\ldots,k]$, $\mathbf{P}[X(e) \geq i] = \mathbf{P}\left[\varepsilon(e) < 2^{k-i}\right]$. Thus,

$$\mathbf{E}[X(e)] \;=\; \sum_{i=1}^{k} \mathbf{P}[X(e) \geq i] \;=\; \sum_{i=1}^{k} \mathbf{P}\left[\varepsilon(e) < 2^{k-i}\right] \;\leq\; \sum_{i\geq 1} p^i \;=\; \frac{p}{1-p} =: \alpha(e).$$

□

For example, if $p = 1/2$, then the above lemma captures (the discrete analogous of) the uniform, the exponential, and the normal distribution.

The following characterization is similar to the one given by Meyer [Mey02, Mey03].

Lemma 2. *Let $\mathcal{P}(e)$ be such that there exists an integer $\lambda := k - p$, for some constant p, such that for each $i \in [0,\ldots,\lambda]$, $\mathbf{P}\left[\varepsilon(e) < 2^i\right] \leq 2^{-\lambda}\cdot 2^i$. Then, $\mathcal{P}(e)$ is permissible.*

Proof. We have for each $i \in [0,\ldots,k]$, $\mathbf{P}[X(e) \geq i] = \mathbf{P}\left[\varepsilon(e) < 2^{k-i}\right]$. Thus,

$$\begin{aligned}
\mathbf{E}[X(e)] &= \sum_{i=1}^{k} \mathbf{P}[X(e) \geq i] \;=\; \sum_{i=1}^{k} \mathbf{P}\left[\varepsilon(e) < 2^{k-i}\right] \;=\; \sum_{i=0}^{k-1} \mathbf{P}\left[\varepsilon(e) < 2^{i}\right] \\
&= \sum_{i=0}^{\lambda} \mathbf{P}\left[\varepsilon(e) < 2^{i}\right] + \sum_{i=\lambda+1}^{k-1} \mathbf{P}\left[\varepsilon(e) < 2^{i}\right] \;\leq\; \sum_{i=0}^{\lambda} 2^{-\lambda}\cdot 2^i + \sum_{i=\lambda+1}^{k-1} 1 \\
&= \sum_{i=0}^{\lambda} 2^{-i} + (k-\lambda-1) \;\leq\; k-\lambda+1 \;=\; p+1 \;=:\; \alpha(e).
\end{aligned}$$

□

Consider Figure 1 for an example. Let $F(x)$ be the cumulative distribution function of $\mathcal{P}(e)$. (Recall that since $\mathcal{P}(e)$ is discrete, $F(x)$ has a piecewise constant and staircase-like form — the figure is slightly misleading.) The conditions in Lemma 2 are fulfilled if $F(x)$ is bounded from above by a function $g(x) := \min\{2^{-\lambda}\cdot(x+1), 1\}$, and the distance between $2^\lambda - 1$ and $2^k - 1$ is less than $(1-2^{-p})\cdot 2^k$, for some constant p.

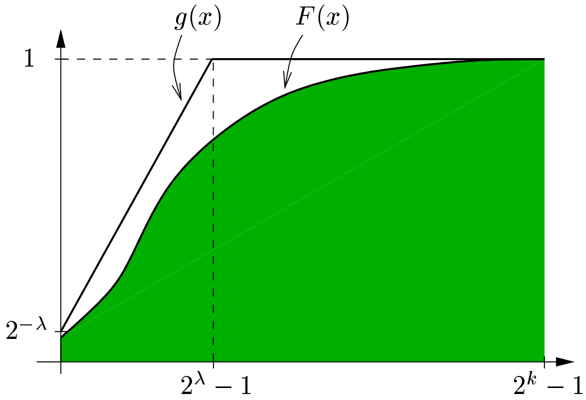

Figure 1: $F(x) := \mathbf{P}[\varepsilon(e) \leq x]$, $g(x) := \min\{2^{-\lambda} \cdot (x+1), 1\}$.

References

[BBM03] C. Banderier, R. Beier, and K. Mehlhorn, *Smoothed analysis of three combinatorial problems*, 28th International Symposium on Mathematical Foundations of Computer Science (MFCS), 2003.

[DF79] E. V. Denardo and B. L. Fox, *Shortest route methods: 1. reaching, pruning and buckets*, Operations Research **27** (1979), 161–186.

[Dij59] E. W. Dijkstra, *A note on two problems in connexion with graphs*, Numerische Mathematik **1** (1959), 260–271.

[FT87] M. L. Fredman and R. E. Tarjan, *Fibonacci heaps and their uses in improved network optimization algorithms*, Journal of the ACM **34** (1987), 596–615.

[Gol02] A. Goldberg, *A practical shortest path algorithm with linear expected time*, SIAM Journal on Computing (2002), submitted.

[Mey02] U. Meyer, *Design and analysis of sequential and parallel single-source shortest-paths algorithms*, Ph.D. thesis, Universität des Saarlandes, Germany, 2002.

[Mey03] U. Meyer, *Average-case complexity of single-source shortest-paths algorithms: lower and upper bounds*, Journal of Algorithms **48** (2003), 91–134.

[ST01] D. Spielman and S. H. Teng, *Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time*, Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, 2001, pp. 296–305.

[Tho99] M. Thorup, *Undirected single-source shortest paths with positive integer weights in linear time*, Journal of the ACM **46** (1999), 362–394.

A Tail Bounds

Theorem 5 (Hoeffding Bound). *Let $X_1, \ldots, X_m$ be independent random variables with $X_i \in [0, \ldots, k]$, for some $k > 0$. Define $X := \sum_{i=1}^{m} X_i$ and $\mu := \mathbf{E}[X]$. Then, for any t,*

$$\mathbf{P}[X \geq t] \leq \left[\left(\frac{t}{\mathrm{e}\mu} \right)^{-t} \mathrm{e}^{-\mu} \right]^{1/k}.$$

I N F O R M A T I K

Below you find a list of the most recent technical reports of the Max-Planck-Institut für Informatik. They are available by anonymous ftp from `ftp.mpi-sb.mpg.de` under the directory `pub/papers/reports`. Most of the reports are also accessible via WWW using the URL `http://www.mpi-sb.mpg.de`. If you have any questions concerning ftp or WWW access, please contact `reports@mpi-sb.mpg.de`. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
Library
attn. Anja Becker
Stuhlsatzenhausweg 85
66123 Saarbrücken
GERMANY
e-mail: `library@mpi-sb.mpg.de`

MPI-I-2003-NWG2-002	F. Eisenbrand	Fast integer programming in fixed dimension
MPI-I-2003-NWG2-001	L.S. Chandran, C.R. Subramanian	Girth and Treewidth
MPI-I-2003-4-009	N. Zakaria	FaceSketch: An Interface for Sketching and Coloring Cartoon Faces
MPI-I-2003-4-008	C. Roessl, I. Ivrissimtzis, H. Seidel	Tree-based triangle mesh connectivity encoding
MPI-I-2003-4-007	I. Ivrissimtzis, W. Jeong, H. Seidel	Neural Meshes: Statistical Learning Methods in Surface Reconstruction
MPI-I-2003-4-006	C. Roessl, F. Zeilfelder, G. Nürnberger, H. Seidel	Visualization of Volume Data with Quadratic Super Splines
MPI-I-2003-4-005	T. Hangelbroek, G. Nürnberger, C. Roessl, H.S. Seidel, F. Zeilfelder	The Dimension of C^1 Splines of Arbitrary Degree on a Tetrahedral Partition
MPI-I-2003-4-004	P. Bekaert, P. Slusallek, R. Cools, V. Havran, H. Seidel	A custom designed density estimation method for light transport
MPI-I-2003-4-003	R. Zayer, C. Roessl, H. Seidel	Convex Boundary Angle Based Flattening
MPI-I-2003-4-002	C. Theobalt, M. Li, M. Magnor, H. Seidel	A Flexible and Versatile Studio for Synchronized Multi-view Video Recording
MPI-I-2003-4-001	M. Tarini, H.P.A. Lensch, M. Goesele, H. Seidel	3D Acquisition of Mirroring Objects
MPI-I-2003-2-004		?
MPI-I-2003-2-003	Y. Kazakov, H. Nivelle	Subsumption of concepts in $DL\ \mathcal{FL}_0$ for (cyclic) terminologies with respect to descriptive semantics is PSPACE-complete
MPI-I-2003-2-002	M. Jaeger	A Representation Theorem and Applications to Measure Selection and Noninformative Priors
MPI-I-2003-2-001	P. Maier	Compositional Circular Assume-Guarantee Rules Cannot Be Sound And Complete
MPI-I-2003-1-018		A Note on the Smoothed Complexity of the Single-Source Shortest Path Problem
MPI-I-2003-1-017	G. Schäfer, S. Leonardi	Cross-Monotonic Cost Sharing Methods for Connected Facility Location Games
MPI-I-2003-1-016	G. Schäfer, N. Sivadasan	Topology Matters: Smoothed Competitive Analysis of Metrical Task Systems"
MPI-I-2003-1-015	A. Kovács	Sum-Multicoloring on Paths
MPI-I-2003-1-014	G. Schäfer, L. Becchetti, S. Leonardi, A. Marchetti-Spaccamela, T. Vredeveld	Average Case and Smoothed Competitive Analysis of the Multi-Level Feedback Algorithm
MPI-I-2003-1-013	I. Katriel, S. Thiel	Fast Bound Consistency for the Global Cardinality Constraint

MPI-I-2003-1-012	D. Fotakis, R. Pagh, P. Sanders, P. Spirakis	Space Efficient Hash Tables with Worst Case Constant Access Time
MPI-I-2003-1-011	P. Krysta, A. Czumaj, B. Voecking	Selfish Traffic Allocation for Server Farms
MPI-I-2003-1-010	H. Tamaki	A linear time heuristic for the branch-decomposition of planar graphs
MPI-I-2003-1-009	B. Csaba	On the Bollobás – Eldridge conjecture for bipartite graphs
MPI-I-2003-1-008	P. Sanders	Soon to be published
MPI-I-2003-1-007	H. Tamaki	Alternating cycles contribution: a strategy of tour-merging for the traveling salesman problem
MPI-I-2003-1-006	M. Dietzfelbinger, H. Tamaki	On the probability of rendezvous in graphs
MPI-I-2003-1-005	M. Dietzfelbinger, P. Woelfel	Almost Random Graphs with Simple Hash Functions
MPI-I-2003-1-004	E. Althaus, T. Polzin, S.V. Daneshmand	Improving Linear Programming Approaches for the Steiner Tree Problem
MPI-I-2003-1-003	R. Beier, B. Vöcking	Random Knapsack in Expected Polynomial Time
MPI-I-2003-1-002	P. Krysta, P. Sanders, B. Vöcking	Scheduling and Traffic Allocation for Tasks with Bounded Splittability
MPI-I-2003-1-001	P. Sanders, R. Dementiev	Asynchronous Parallel Disk Sorting
MPI-I-2002-4-002	F. Drago, W. Martens, K. Myszkowski, H. Seidel	Perceptual Evaluation of Tone Mapping Operators with Regard to Similarity and Preference
MPI-I-2002-4-001	M. Goesele, J. Kautz, J. Lang, H.P.A. Lensch, H. Seidel	Tutorial Notes ACM SM 02 A Framework for the Acquisition, Processing and Interactive Display of High Quality 3D Models
MPI-I-2002-2-008	W. Charatonik, J. Talbot	Atomic Set Constraints with Projection
MPI-I-2002-2-007	W. Charatonik, H. Ganzinger	Symposium on the Effectiveness of Logic in Computer Science in Honour of Moshe Vardi
MPI-I-2002-1-008	P. Sanders, J.L. Träff	The Factor Algorithm for All-to-all Communication on Clusters of SMP Nodes
MPI-I-2002-1-005	M. Hoefer	Performance of heuristic and approximation algorithms for the uncapacitated facility location problem
MPI-I-2002-1-004	S. Hert, T. Polzin, L. Kettner, G. Schäfer	Exp Lab A Tool Set for Computational Experiments
MPI-I-2002-1-003	I. Katriel, P. Sanders, J.L. Träff	A Practical Minimum Scanning Tree Algorithm Using the Cycle Property
MPI-I-2002-1-002	F. Grandoni	Incrementally maintaining the number of l-cliques
MPI-I-2002-1-001	T. Polzin, S. Vahdati	Using (sub)graphs of small width for solving the Steiner problem
MPI-I-2001-4-005	H.P.A. Lensch, M. Goesele, H. Seidel	A Framework for the Acquisition, Processing and Interactive Display of High Quality 3D Models
MPI-I-2001-4-004	S.W. Choi, H. Seidel	Linear One-sided Stability of MAT for Weakly Injective Domain
MPI-I-2001-4-003	K. Daubert, W. Heidrich, J. Kautz, J. Dischler, H. Seidel	Efficient Light Transport Using Precomputed Visibility
MPI-I-2001-4-002	H.P.A. Lensch, J. Kautz, M. Goesele, H. Seidel	A Framework for the Acquisition, Processing, Transmission, and Interactive Display of High Quality 3D Models on the Web
MPI-I-2001-4-001	H.P.A. Lensch, J. Kautz, M. Goesele, W. Heidrich, H. Seidel	Image-Based Reconstruction of Spatially Varying Materials
MPI-I-2001-2-006	H. Nivelle, S. Schulz	Proceeding of the Second International Workshop of the Implementation of Logics
MPI-I-2001-2-005	V. Sofronie-Stokkermans	Resolution-based decision procedures for the universal theory of some classes of distributive lattices with operators
MPI-I-2001-2-004	H. de Nivelle	Translation of Resolution Proofs into Higher Order Natural Deduction using Type Theory
MPI-I-2001-2-003	S. Vorobyov	Experiments with Iterative Improvement Algorithms on Completely Unimodel Hypercubes
MPI-I-2001-2-002	P. Maier	A Set-Theoretic Framework for Assume-Guarantee Reasoning
MPI-I-2001-2-001	U. Waldmann	Superposition and Chaining for Totally Ordered Divisible Abelian Groups
MPI-I-2001-1-007	T. Polzin, S. Vahdati	Extending Reduction Techniques for the Steiner Tree Problem: A Combination of Alternative-and Bound-Based Approaches