Case Studies:
Bin Packing &
The Traveling Salesman Problem

TSP: Part II

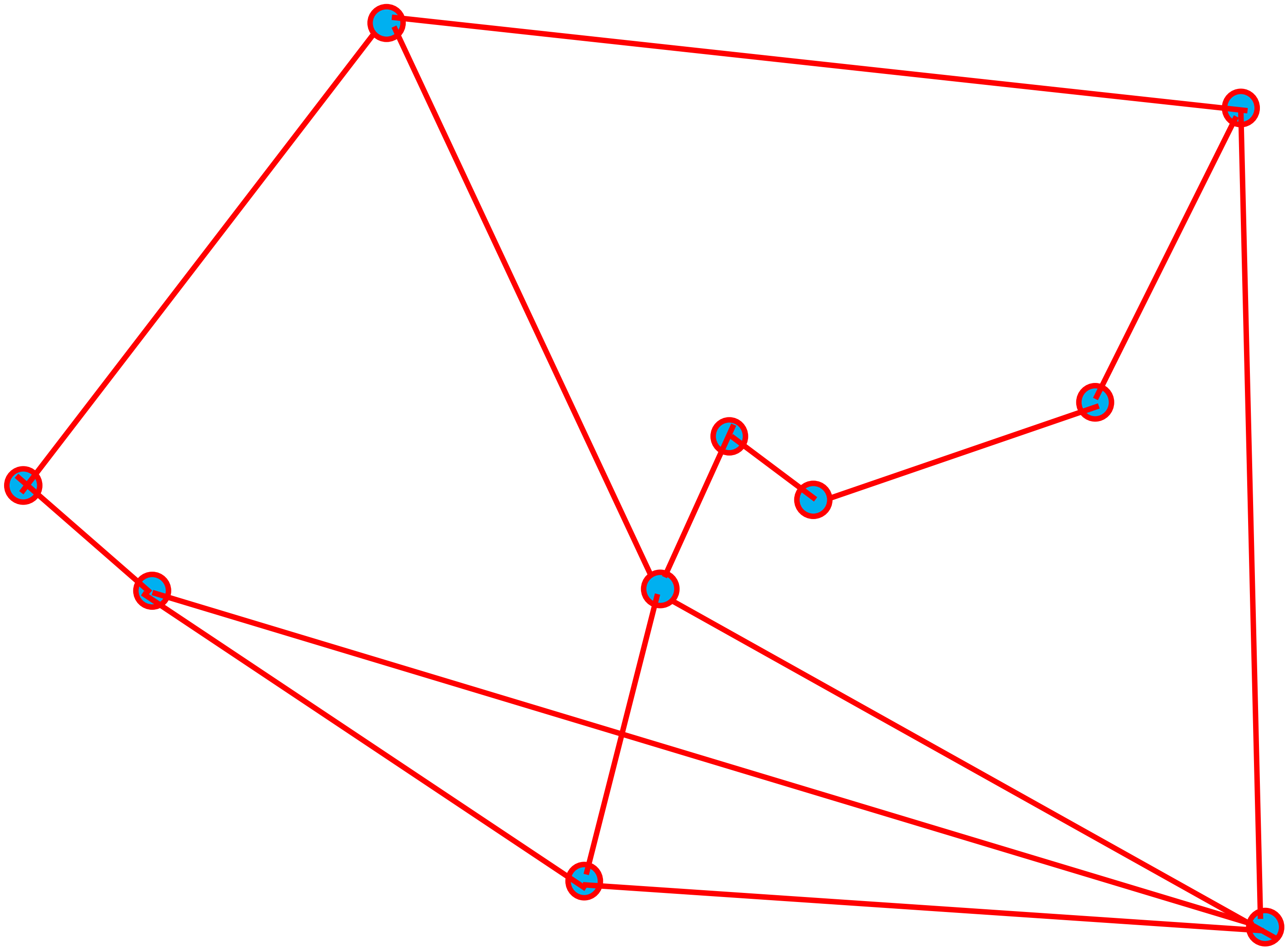
David S. Johnson
AT&T Labs – Research

To the Students of the 2010 Microsoft School on Data Structures and Algorithms

- Thanks for all your "Get Well" wishes. I am back in the USA now and almost fully recovered. I am truly sorry I was unable to present my Friday lectures and my Q&A session. I had been looking forward to both.
- Given that I missed the Q&A session, feel free to send me email if you have any questions I might help you with (technical or otherwise). My email address is dsj@research.att.com.
- I hope these slides (and the Bin Packing slides I am also uploading) are still of some value, even without the vocal commentary I would have provided had I been able to give the talks. I still owe you a bibliography, but you can find many of my own TSP and bin packing papers at http://www.research.att.com/~dsj/, along with NP-completeness columns and other goodies.
- Best wishes to you all -- David Johnson, 18 August, 2010.

Special Request

2-Opt Animation: Nearest Neighbor Starting Tour



Special Bonus: Picture from Shaggier Times (~1976)

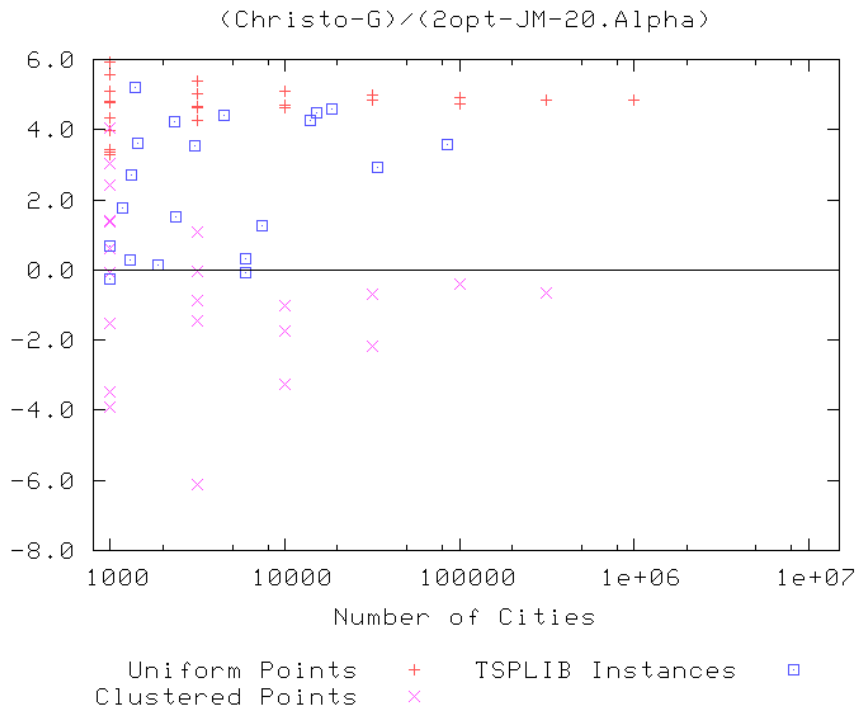
And Now,

Back to the show.

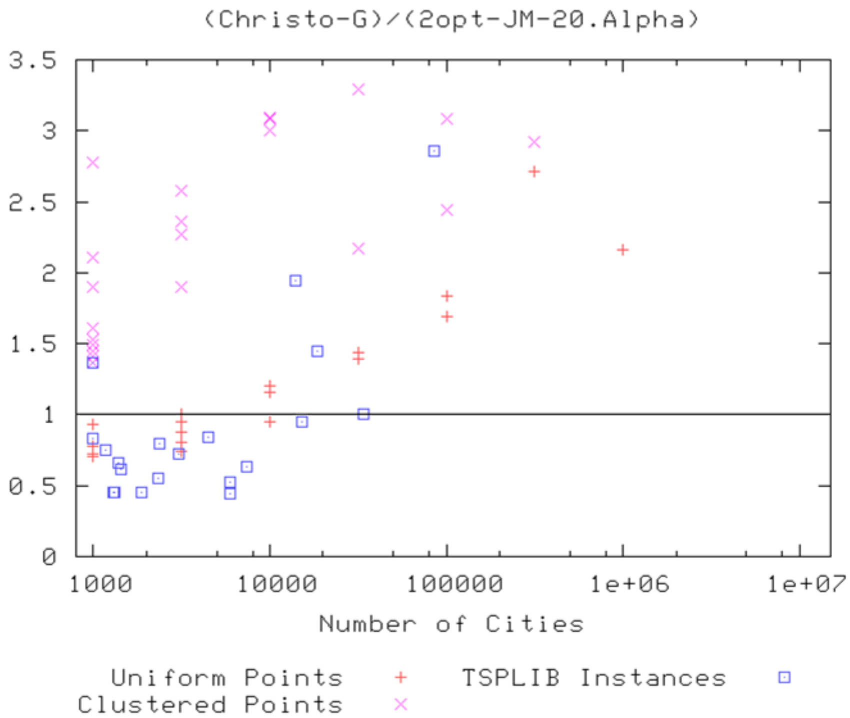
For more on the TSP algorithm performance, see the website for the DIMACS TSP Challenge:

http://www2.research.att.com/~dsj/chtsp/index.html/

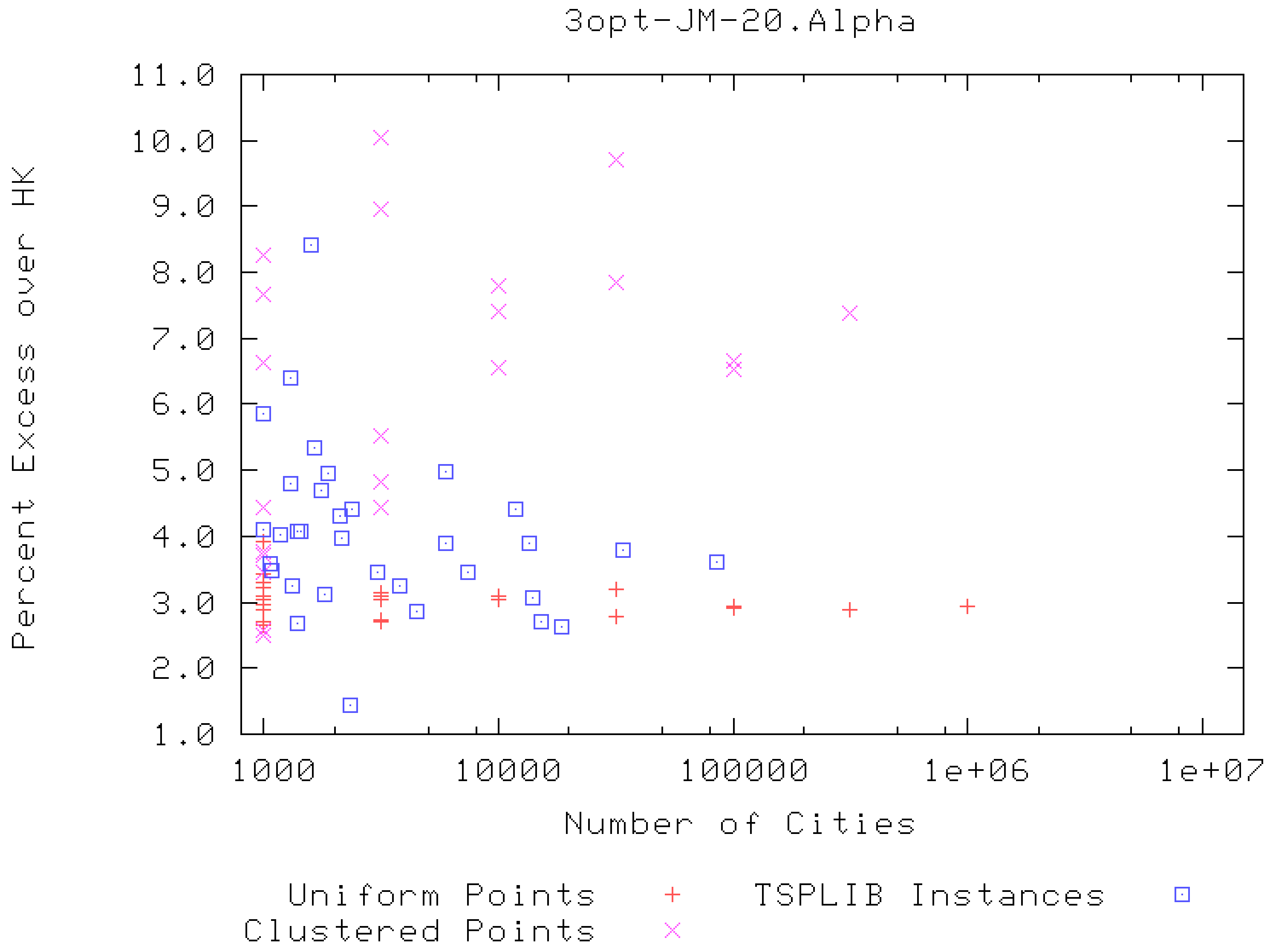
Comparison: Smart-Shortcut Christofides versus 2-Opt



Tour Length



Normalized Running Time


3opt-JM-20.Alpha
Percent Excess over HK
11.0
10.0
9.0
8.0
7.0
6.0
5.0
4.0
3.0
2.0
1.0
1000
10000
100000
1e+06
1e+07
Number of Cities
Uniform Points
TSPLIB Instances
Clustered Points

pla7397

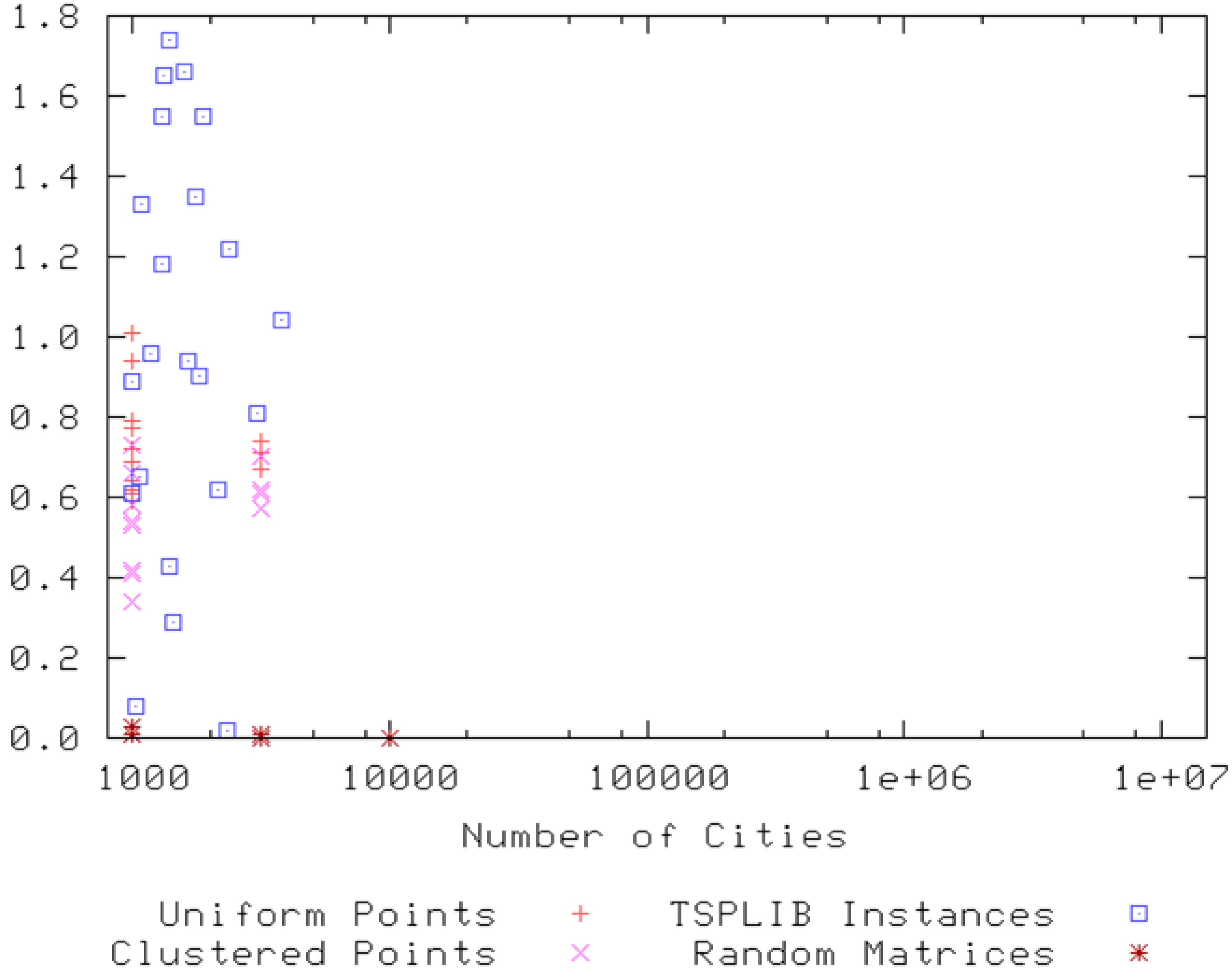
Percent over HK	Normalized Seconds	Implementation
-0.5406	6.500	AppHK-R-10F
-0.5170	12.180	AppHK-R-20F
-0.3037	55.150	AppHK-R-20S
0.0000		HK-bounds
0.0000	55.420	HK-ABCC
0.5806		Optval
0.5806	9272.040	Helsgaun-N
0.5807	17197.710	MLLKH-N
0.5861	1897.390	ILK-NYYY-10N
0.5861	9163.720	MLLKH-.5N
0.5999	1567.990	Helsgaun-.1N
0.6016	1887.170	ILK-NYYY-N-b10
0.6077	193.480	ILK-NYYY-N
0.6078	1796.190	MLLKH-.05N
0.6116	8064.000	ILK-JM-10N
0.6305	3515.080	ILK-JM-N-b10
0.7565	303.830	ILK-JM-N
0.7606	129.100	ILK-JM-.3N
0.7647	382.240	CLK-ABCC-N-b10
0.8204	60.230	ILK-NYYY-Ng
0.8257	24.460	CLK-ACR-N
0.8343	33.610	BSDP-10
0.8374	24.890	BSDP-8
0.8422	23.420	BSDP-6
0.8478	331.340	CLK-ABCC-10N
0.8482	23.060	CLK-ABCC-N.Sparc

Held-Karp (or "Subtour") Bound

- Linear programming relaxation of the following formulation of the TSP as an integer program:
- Minimize $\sum_{\text{city pairs } \{c,c'\}} (x_{\{c,c'\}} d(c,c'))$
- Subject to
  - $\sum_{c' \in C} x_{\{c,c'\}} = 2$, for all $c \in C$.
  - $\sum_{c \in S, c' \in C-S} x_{\{c,c'\}} \geq 2$, for all $S \subset C$ (subtour constraints)
  - $0 \leq x_{\{c,c'\}} \leq 1$, for all pairs $\{c,c'\} \subset C$.

Linear programming relaxation

Percent by which Optimal Tour exceeds Held-Karp Bound



Computing the Held-Karp Bound

- Difficulty: Too many "subtour" constraints:

$$\Sigma_{c \in S, c' \in C-S} x_{\{c,c'\}} \geq 2, \text{ for all } S \subset C$$

(There are 2^N-2 such S)

- Fortunately, if any such constraint is violated by our current solution, we can find such a violated constraint in polynomial time:
- Suppose the constraint for S is violated by solution $\mathbf{x}$. Consider the graph G, where edge $\{c,c'\}$ has capacity $x_{\{c,c'\}}$. For any pair of vertices (u,v), $u \in S$ and $v \in C-S$, the maximum flow from u to v is less than 2 (and conversely).
- Consequently, an S yielding a violated inequality can be found using $O(N)$ network flow computations, assuming such an inequality exists.

Computing the Held-Karp Bound

- Pick a city c. If the desired cut exists, there must be some other city c′ such that the max flow from c to c′ is less than 2 (a "small flow").
- Test all candidates for c′ (N-1 flow computations)
- If no small flows found, no subtour constraint is violated.
- Otherwise, let c* be a c′ with a small flow.
- Initialize S to {c}.
- For each other city c′ in turn, merge c′ with all the cities in S and test whether the flow from the merged vertex to c* remains small.
  - If yes, add c′ to S.
  - Otherwise, add c′ to C-S.
- Once all N-2 candidates for c′ have been tested, output S.

(Total time can be reduced to that for a constant number of flow computations using more algorithmic ideas.)

pla7397

Percent over HK	Normalized Seconds	Implementation
-0.5406	6.500	AppHK-R-10F
-0.5170	12.180	AppHK-R-20F
-0.3037	55.150	AppHK-R-20S
0.0000		HK-bounds
0.0000	55.420	HK-ABCC
0.5806		Optval
0.5806	9272.040	Helsgaun-N
0.5807	17197.710	MLLKH-N
0.5861	1897.390	ILK-NYYY-10N
0.5861	9163.720	MLLKH-.5N
0.5999	1567.990	Helsgaun-.1N
0.6016	1887.170	ILK-NYYY-N-b10
0.6077	193.480	ILK-NYYY-N
0.6078	1796.190	MLLKH-.05N
0.6116	8064.000	ILK-JM-10N
0.6305	3515.080	ILK-JM-N-b10
0.7565	303.830	ILK-JM-N
0.7606	129.100	ILK-JM-.3N
0.7647	382.240	CLK-ABCC-N-b10
0.8204	60.230	ILK-NYYY-Ng
0.8257	24.460	CLK-ACR-N
0.8343	33.610	BSDP-10
0.8374	24.890	BSDP-8
0.8422	23.420	BSDP-6
0.8478	331.340	CLK-ABCC-10N
0.8482	23.060	CLK-ABCC-N.Sparc

Optimization: State of the Art

Lin-Kernighan [Johnson-McGeoch Implementation]
1.4% off optimal
10,000,000 cities in 46 minutes at 2.6 Ghz

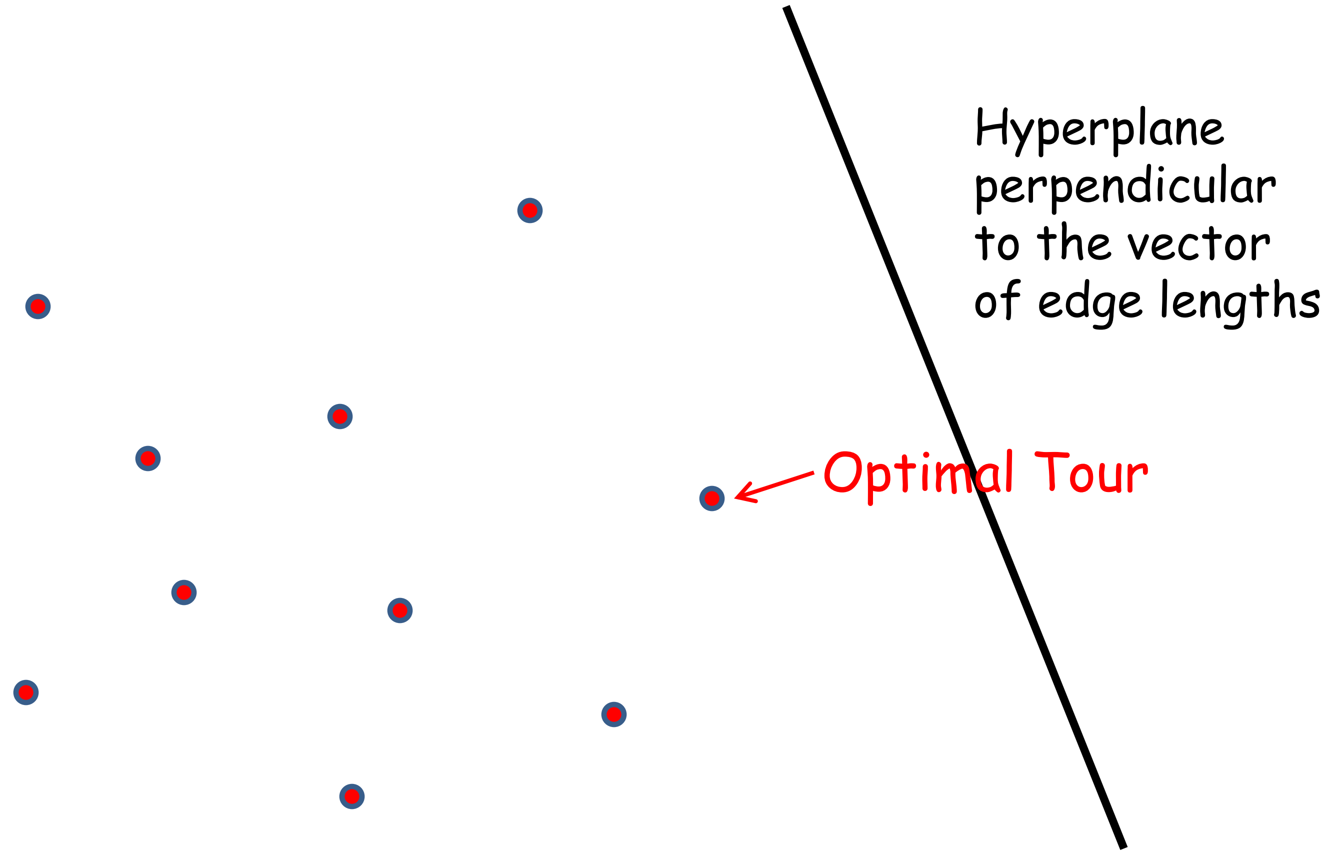
Iterated Lin-Kernighan [J-M Implementation]
0.4% off optimal
100,000 cities in 35 minutes at 2.6 Ghz

Concorde Branch-and-Cut Optimization
[Applegate-Bixby-Chvatal-Cook]
Optimum
1,000 cities in median time 5 minutes at 2.66 Ghz

Concorde

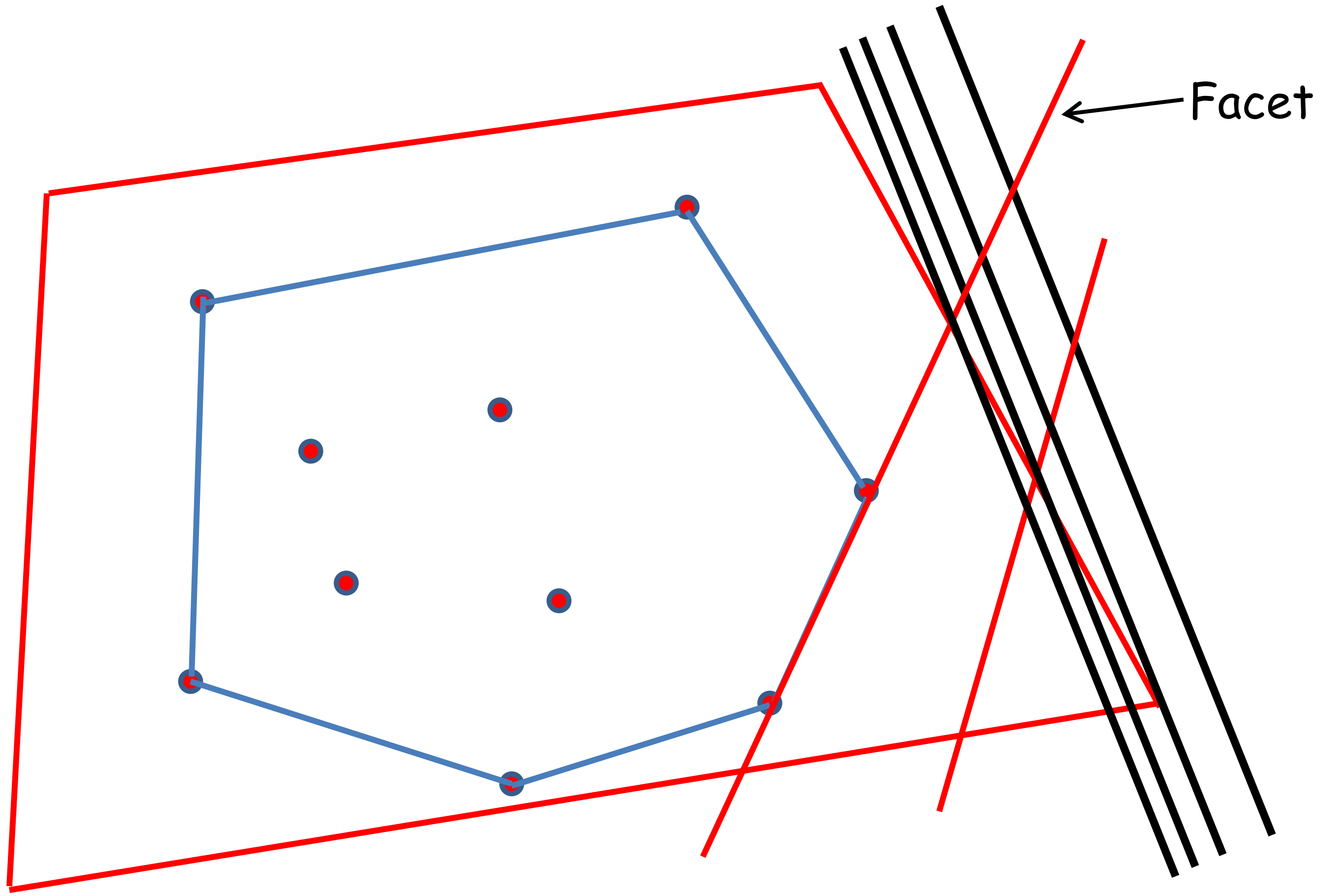
- "Branch-and-Cut" approach exploiting linear programming to determine lower bounds on optimal tour length.
- Based on 30+ years of theoretical developments in the "Mathematical Programming" community.
- Exploits "chained" (iterated) Lin-Kernighan for its initial upperbounds.
- Eventually finds an optimal tour and proves its optimality (unless it runs out of time/space).
- Also can compute the Held-Karp lower bound for very large instances.
- Executables and source code can be downloaded from **`http://www.tsp.gatech.edu/`**

Geometric Interpretation



• -- Points in $R^{N(N-1)/2}$ corresponding to a tour.

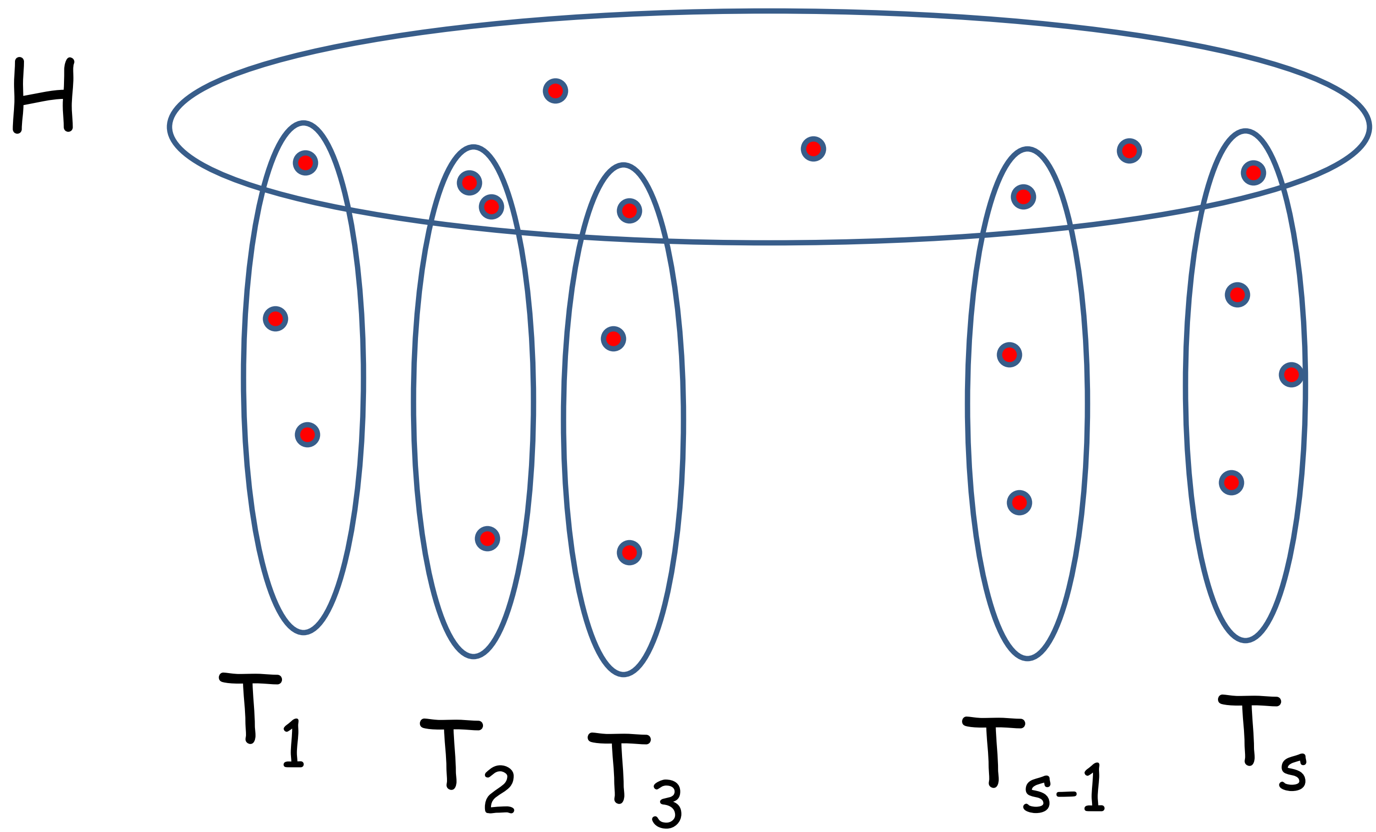
Optimal Tour is a point on the convex hull of all tours.



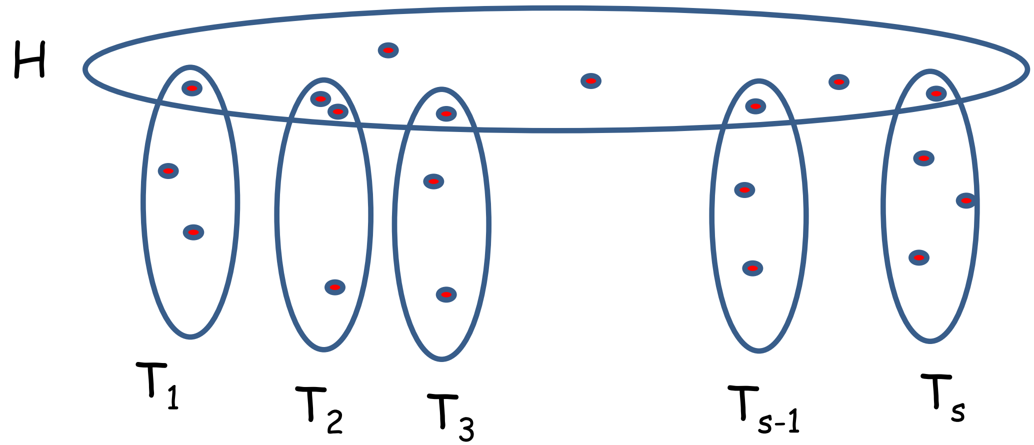
Unfortunately, the LP relaxation of the TSP can be a very poor approximation to the convex hull of tours.

To improve it add more constraints ("cuts").

One Facet Class: Comb Inequalities



Teeth T_i are disjoint, s is odd,
all regions contain at least one city.



- For Y the handle or a tooth, let x(Y) be the total value of the edge variables for edges with one endpoint in Y and one outside, when the function x corresponds to a tour
- By subtour inequalities, we must have $x(Y) \geq 2$ for each such Y. It also must be even, which is exploited to prove the *comb inequality:*

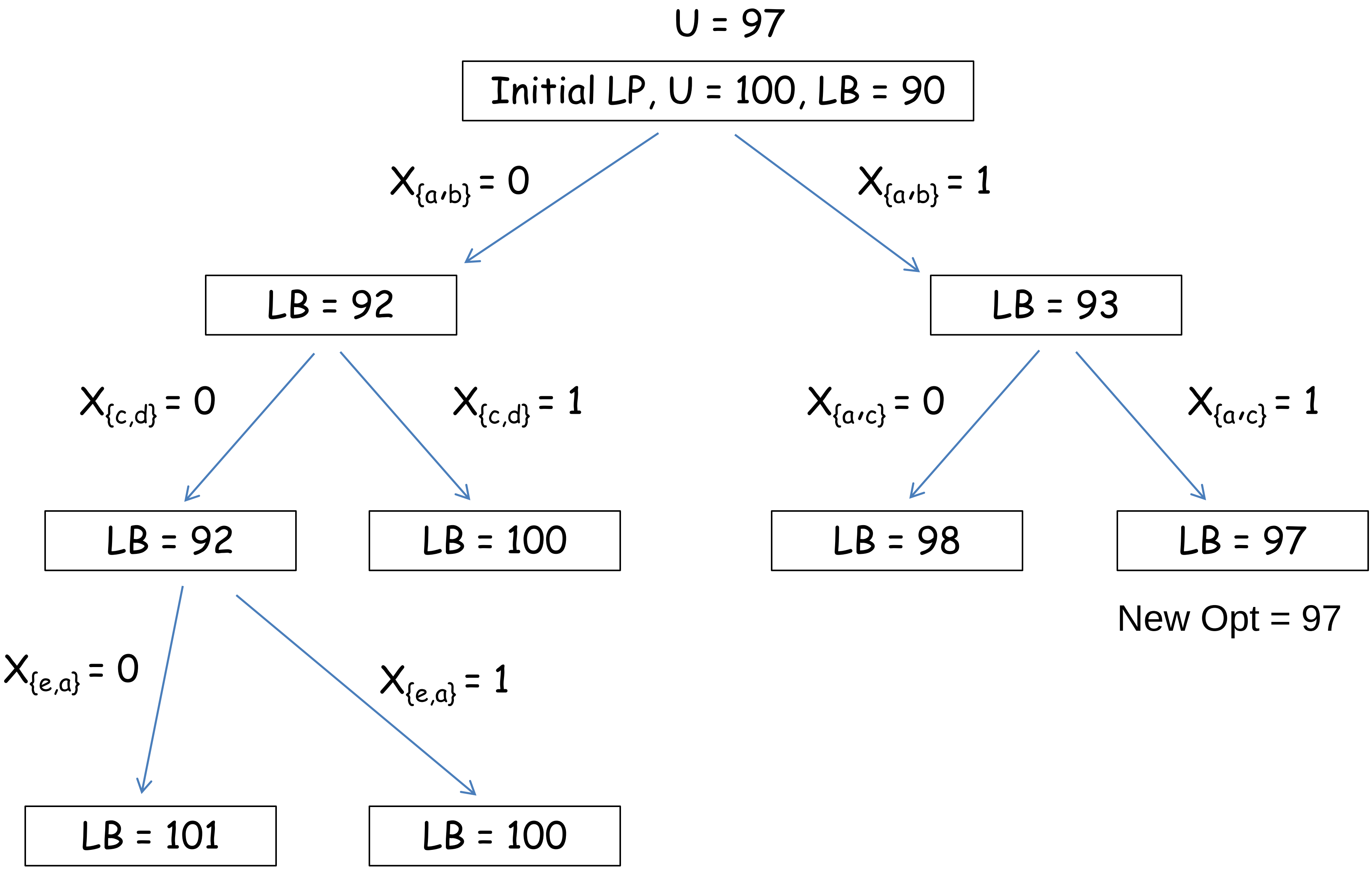
$$x(H) + \sum_{i=1}^{s} x(T_i) \geq 3s + 1$$

Branch & Cut

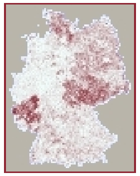
- Use a heuristic to generate a initial "champion" tour and provide provide an upper bound $U \geq OPT$.
- Let our initial "subproblem" consist of an LP with just the inequalities of the LP formulation (or some subset of them).
- Handle subproblems as follows:

Branch & Cut

- Keep adding violated inequalities (of various sorts) that you can find, until
  - (a) LP Solution value $\geq U$. In this case we prune this case and if no other cases are left, our current tour is optimal.
  - (b) Little progress is made in the objective function. In this case, for some edge $\{c,c'\}$ with a fractional value, split into two subproblems, one with $x_{\{c,c'\}}$ fixed at 1 (must be in the tour, and one with it fixed at 0 (must not be in the tour).
- If we ever encounter an LP solution that is a tour and has length $L' < L$, set $L = L'$ and let this new tour be the champion. Prune any subproblems whose LP solution exceeds or equals L. If at any point all your children are pruned, prune yourself.



U = 97
Initial LP, U = 100, LB = 90
X{a,b} = 0
X{a,b} = 1
LB = 92
LB = 93
X{c,d} = 0
X{c,d} = 1
X{a,c} = 0
X{a,c} = 1
LB = 92
LB = 100
LB = 98
LB = 97
New Opt = 97
X{e,a} = 0
X{e,a} = 1
LB = 101
LB = 100



Milestones in the Solution of TSP Instances

Computer codes for the TSP have become increasingly more sophisticated over the years. A conspicuous sign of these improvements is the increasing size of nontrivial instances that have been solved, moving from Dantzig, Fulkerson, and Johnson's solution of a 49-city problem in 1954 up through the solution of a 24,978-city problem 50 years later.

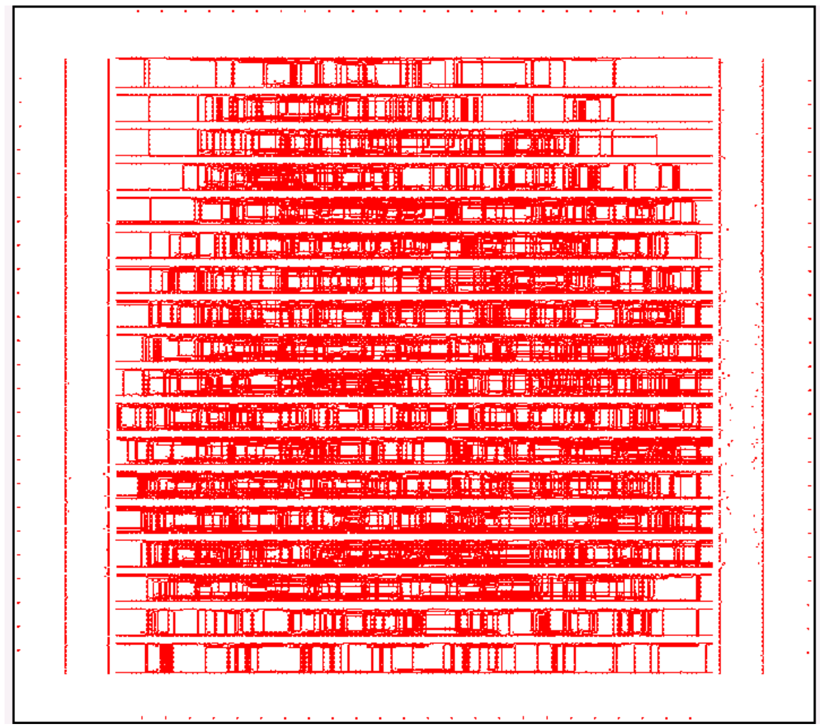
Year	Research Team	Size of Instance	Name
1954	G. Dantzig, R. Fulkerson, and S. Johnson	49 cities	dantzig42
1971	M. Held and R.M. Karp	64 cities	64 random points
1975	P.M. Camerini, L. Fratta, and F. Maffioli	67 cities	67 random points
1977	M. Grötschel	120 cities	gr120
1980	H. Crowder and M.W. Padberg	318 cities	lin318
1987	M. Padberg and G. Rinaldi	532 cities	att532
1987	M. Grötschel and O. Holland	666 cities	gr666
1987	M. Padberg and G. Rinaldi	2,392 cities	pr2392
1994	D. Applegate, R. Bixby, V. Chvátal, and W. Cook	7,397 cities	pla7397
1998	D. Applegate, R. Bixby, V. Chvátal, and W. Cook	13,509 cities	usa13509
2001	D. Applegate, R. Bixby, V. Chvátal, and W. Cook	15,112 cities	d15112
2004	D. Applegate, R. Bixby, V. Chvátal, W. Cook, and K. Helsgaun	24,978 cities	sw24798

Current World Record (2006)

Research Team

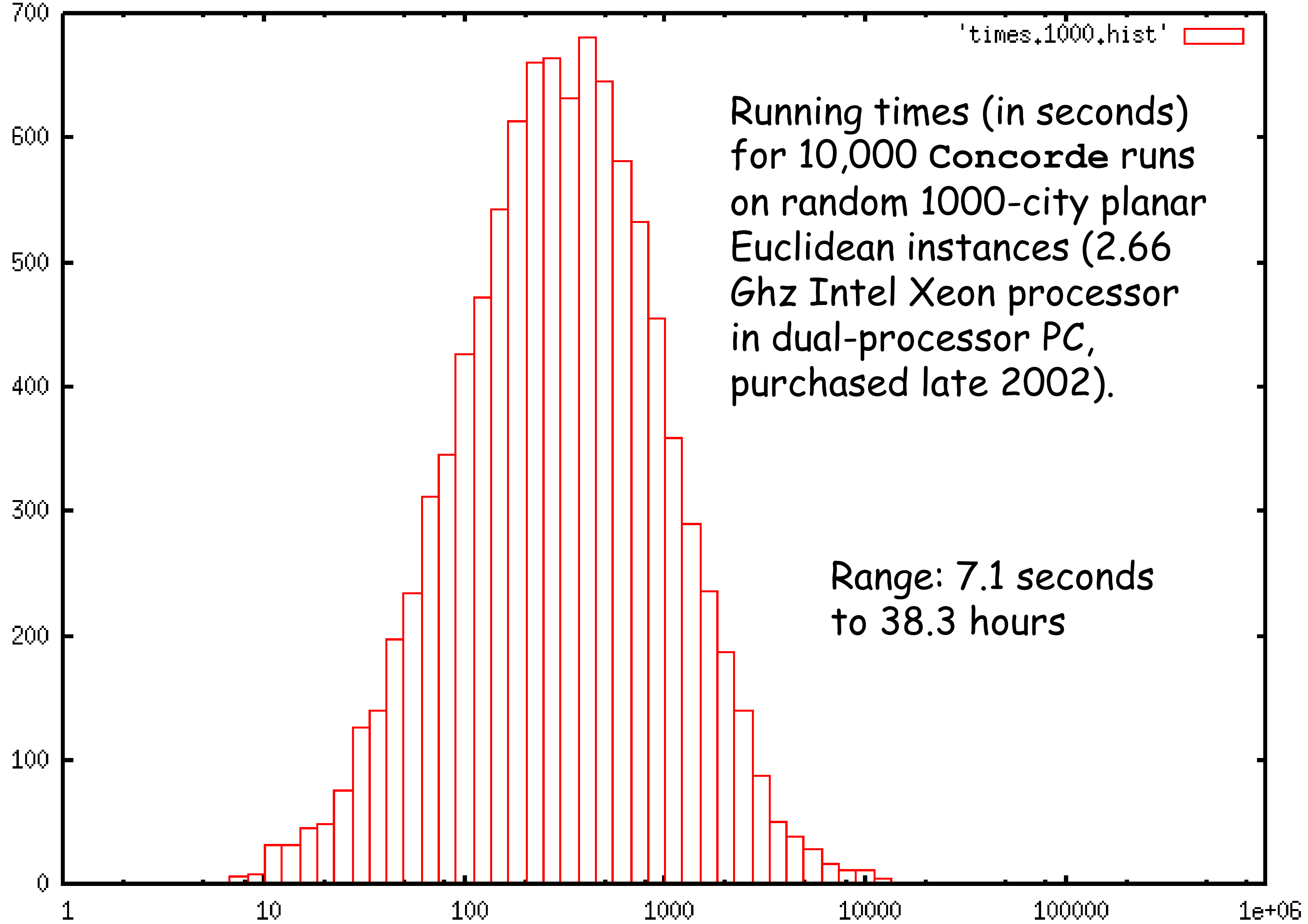
- David Applegate, AT&T Labs - Research
- Robert Bixby, ILOG and Rice University
- Vašek Chvátal, Concordia University
- William Cook, Georgia Tech
- Daniel Espinoza, University of Chile
- Marcos Goycoolea, Universidad Adolfo Ibanez
- Keld Helsgaun, Roskilde University

Using a parallelized version of the Concorde code, Helsgaun's sophisticated variant on Iterated Lin-Kernighan, and 2719.5 cpu-days



N = 85,900

The optimal tour is 0.09% shorter than the tour DSJ constructed using Iterated Lin-Kernighan in 1991. In 1986, when computers were much slower, we could only give the Laser Logic people a Nearest-Neighbor tour, which was 23% worse, but they were quite happy with it…



'times.1000.hist'
Running times (in seconds) for 10,000 Concorde runs on random 1000-city planar Euclidean instances (2.66 Ghz Intel Xeon processor in dual-processor PC, purchased late 2002).
Range: 7.1 seconds to 38.3 hours
700
600
500
400
300
200
100
0
1
10
100
1000
10000
100000
1e+06

Concorde Asymptotics
[Hoos and Stützle, 2009 draft]

- Estimated median running time for planar Euclidean instances.
- Based on
  - 1000 samples each for N = 500,600,…,2000
  - 100 samples each for N = 2500, 3000,3500,4000,4500
  - 2.4 Ghz AMD Opteron 2216 processors with 1MB L2 cache and 4 GB main memory, running Cluster Rocks Linux v4.2.1.

$$0.21 \cdot 1.24194^{\sqrt{N}}$$

Actual median for N = 2000: ~57 minutes, for N = 4,500: ~96 hours

Theoretical Properties of Random Euclidean Instances

Expected optimal tour length for an N-city instance approaches $C\sqrt{N}$ for some constant C as $N \to \infty$. [Beardwood, Halton, and Hammersley, 1959]

Key Open Question:

What is the Value of C?

The Early History

- 1959: BHH estimated $C \approx .75$, based on hand solutions for a 202-city and a 400-city instance.
- 1977: Stein estimates $C \approx .765$, based on extensive simulations on 100-city instances.
- Methodological Problems:
  - Not enough data
  - Probably not true optima for the data there is
  - Misjudges asymptopia

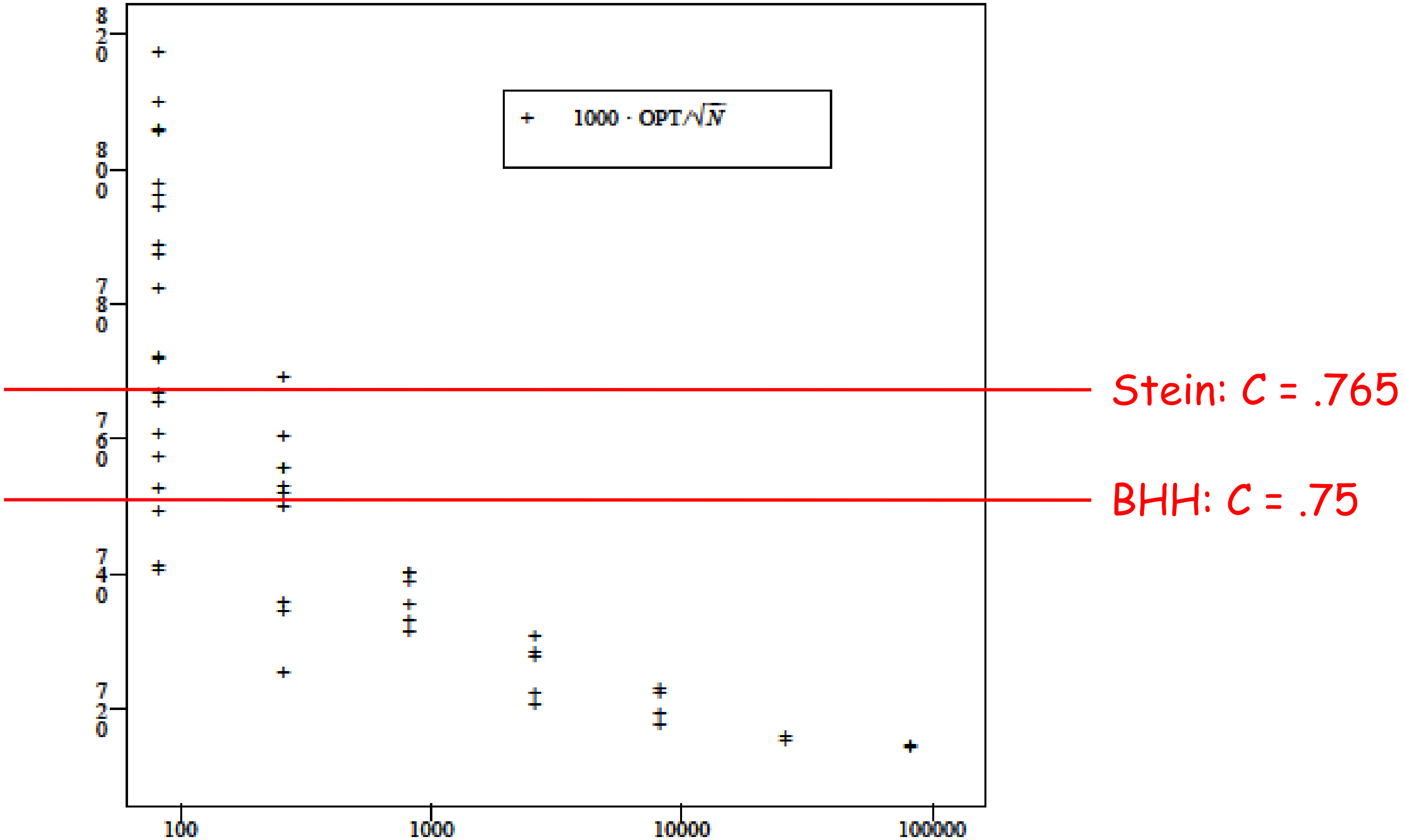


Figure from [Johnson, McGeoch, Rothberg, 1996]

What is the dependence on N ?

- Expected distance to nearest neighbor proportional to $1/\sqrt{N}$, times n cities yields $\Theta(\sqrt{N})$
- $O(\sqrt{N})$ cities close to the boundary are missing some neighbors, for an added contribution proportional to $(\sqrt{N})(1/\sqrt{N})$, or $\Theta(1)$
- A constant number of cities are close to two boundaries (at the corners of the square), which may add an additional $\Theta(1/\sqrt{N})$
- This yields target function

$$OPT/\sqrt{N} = C + \beta/\sqrt{N} + \gamma/N$$

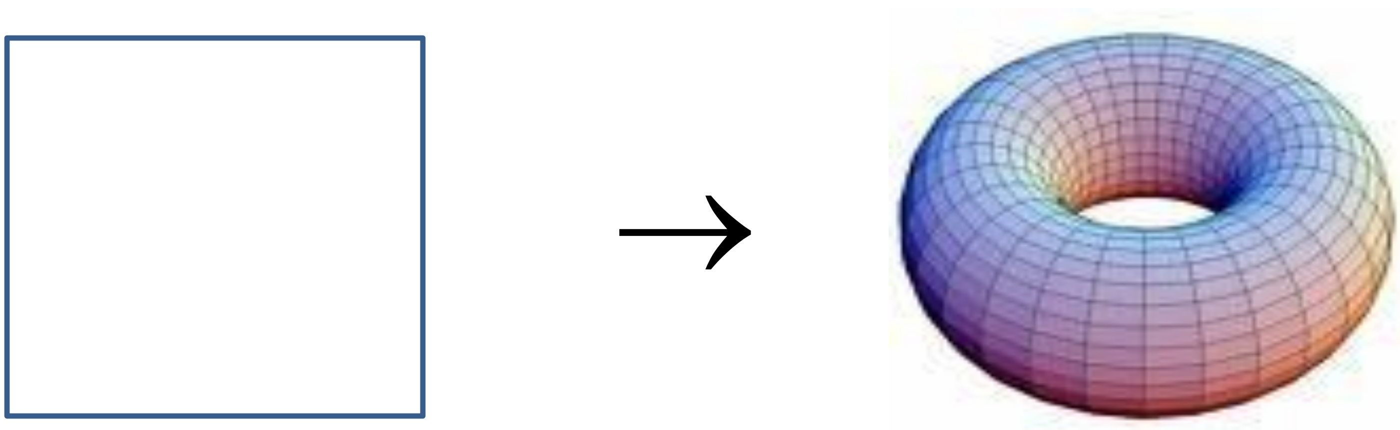
Asymptotic Upper Bound Estimates (Heuristic-Based Results Fitted to $OPT/\sqrt{N} = C + \beta/\sqrt{N}$)

- 1989: Ong & Huang estimate $C \leq .74$, based on runs of 3-Opt.
- 1994: Fiechter estimates $C \leq .73$, based on runs of "parallel tabu search"
- 1994: Lee & Choi estimate $C \leq .721$, based on runs of "multicanonical annealing"
- Still inaccurate, but converging?
- Needed: A new idea.

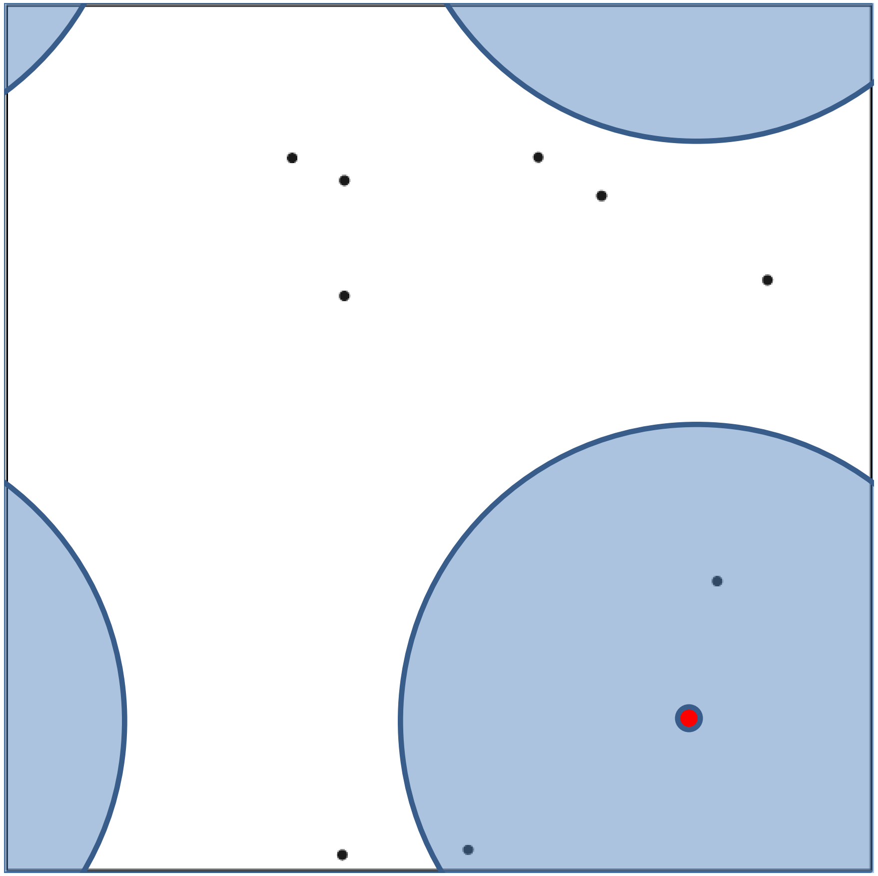
New Idea (1995): Suppress the variance added by the "Boundary Effect" by using

Toroidal Instances

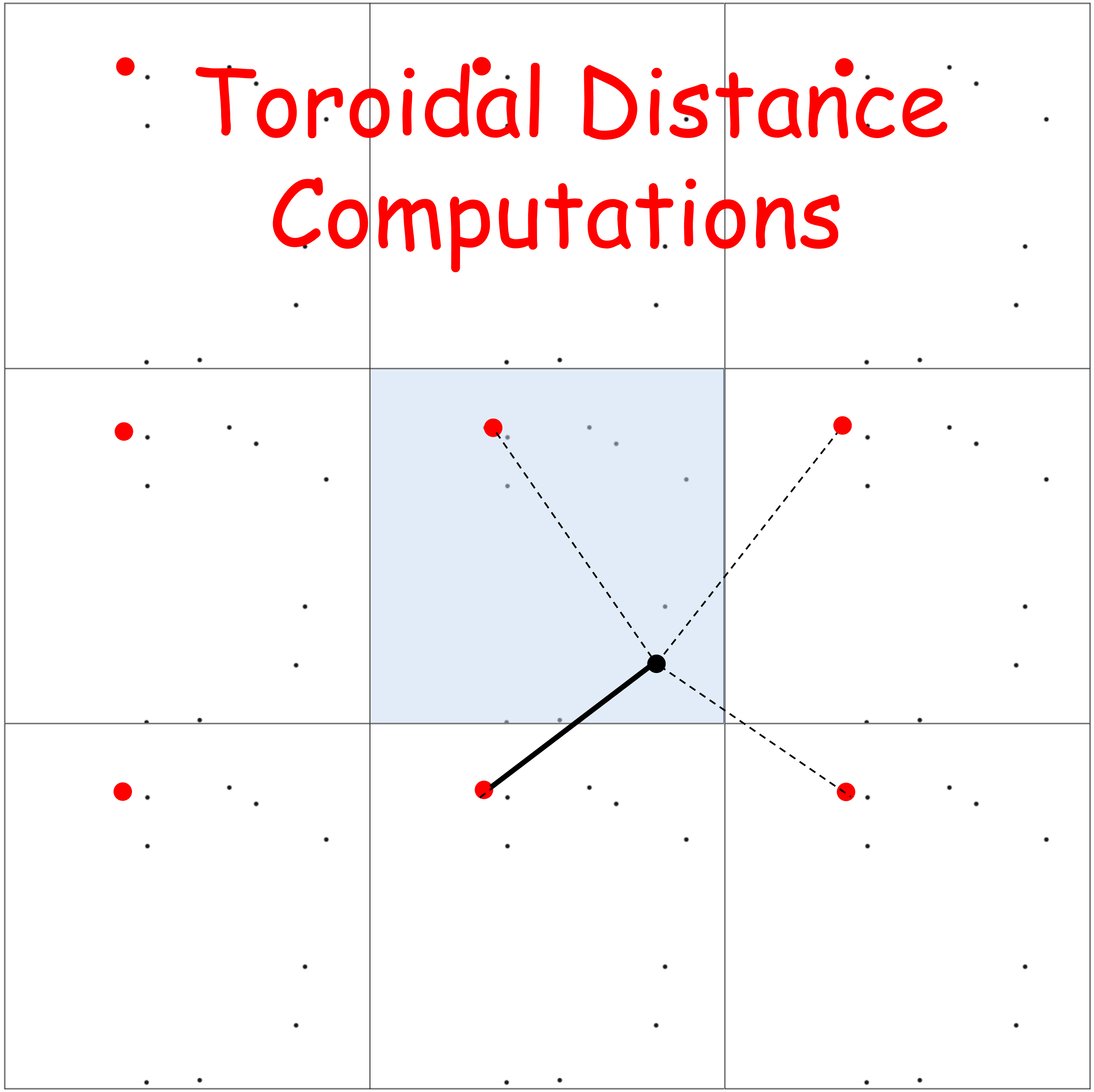
- Join left boundary of the unit square to the right boundary, top to the bottom.



Toroidal Unit Ball



Toroidal Distance Computations



Toroidal Instance Advantages

- No boundary effects.
- Same asymptotic constant for $E[OPT/\sqrt{N}]$ as for planar instances [Jaillet, 1992] (although it is still only asymptotic).
- Lower empirical variance for fixed N.

Toroidal Approaches

1996: Percus & Martin estimate

$C \approx .7120 \pm .0002$.

1996: Johnson, McGeoch, and Rothberg estimate

$C \approx .7124 \pm .0002$.

2004: Jacobsen, Read, and Saleur estimate

$C \approx .7119$.

Each coped with the difficulty of computing optima in a different way.

Percus-Martin

(Go Small)

- Toroidal Instances with $N \leq 100$:
  - 250,000 samples, $N = 12,13,14,15,16,17$
    ("Optimal" = best of 10 Lin-Kernighan runs)
  - 10,000 samples with $N = 30$
    ("Optimal" = best of 5 runs of 10-step-Chained-LK)
  - 6,000 samples with $N = 100$
    ("Optimal" = best of 20 runs of 10-step-Chained-LK)
- Fit to $OPT/\sqrt{N} = (C + a/N + b/N^2)/(1+1/(8N))$

(Normalization by the expected distance to the kth nearest neighbor)

Jacobsen-Read-Saleur

(Go Narrow)

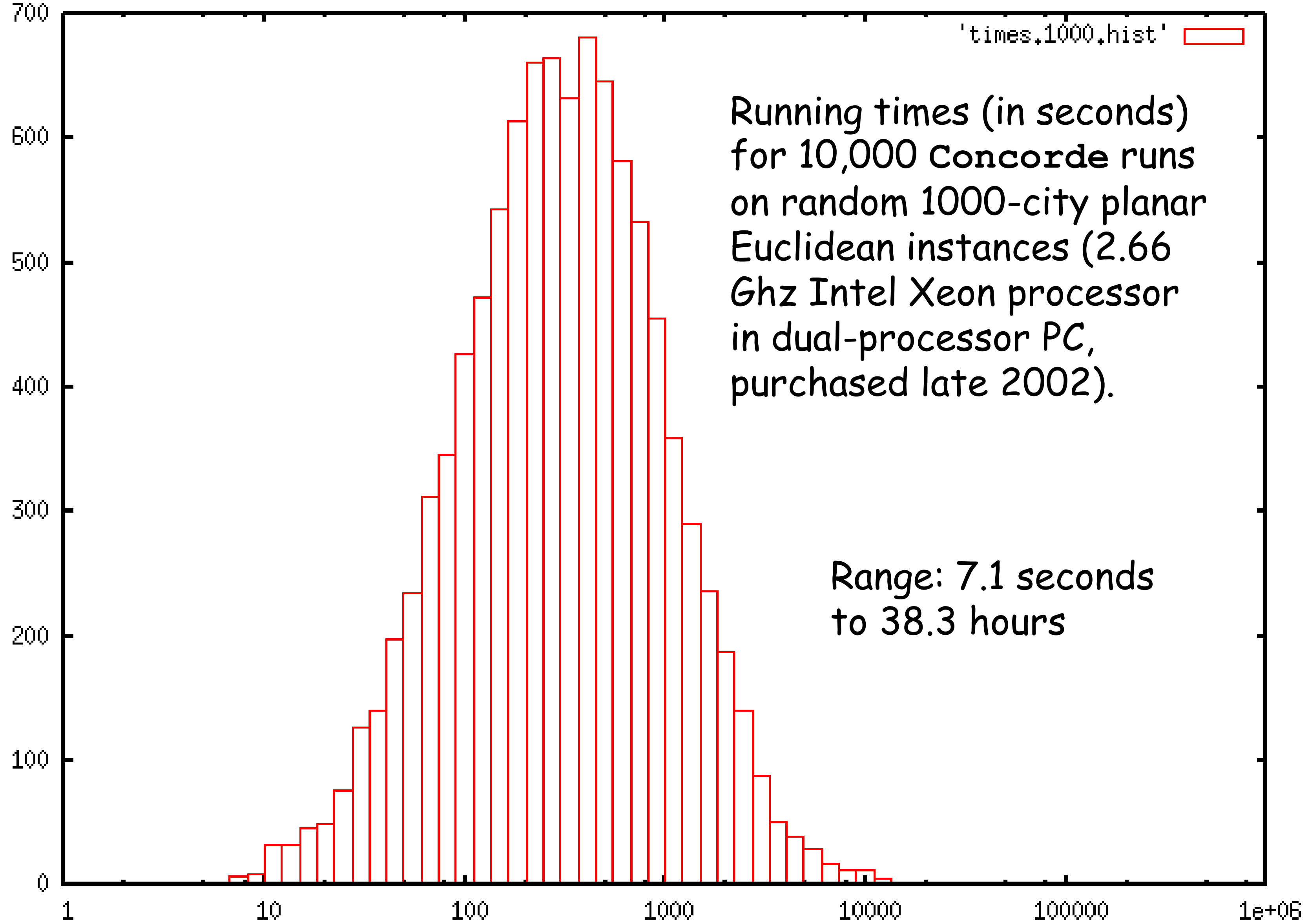
- Cities go uniformly on a $1 \times 100{,}000$ cylinder – that is, only join the top and bottom of the unit square and stretch the width by a factor of 100,000.
- For $W = 1,2,3,4,5,6$, set $N = 100{,}000W$ and generate 10 sample instances.
- Optimize by using dynamic programming, where only those cities within distance k of the frontier (~kw cities) can have degree 0 or 1, $k = 4,5,6,7,8$.
- Estimate true optimal for fixed W as $k \to \infty$.
- Estimate unit square constant as $W \to \infty$.
- With $N \geq 100{,}000$, assume no need for asymptotics in N

Johnson-McGeoch-Rothberg
(Go Held-Karp)

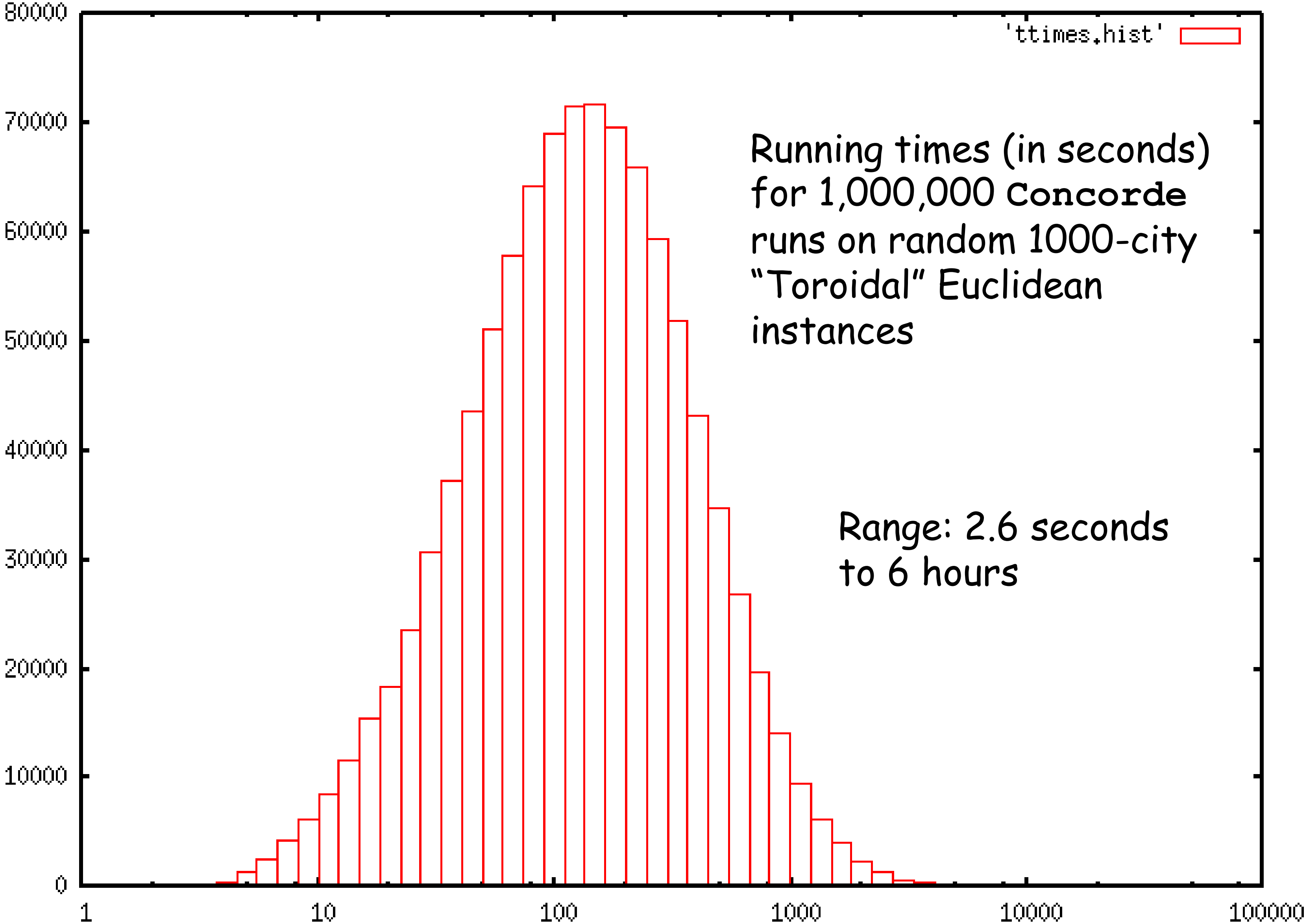
- Observe that
  - the Held-Karp (subtour) bound also has an asymptotic constant, i.e., $HK/\sqrt{n} \rightarrow C_{HK}$ [Goemans, 1995] , and is easier to compute than the optimal.
  - $(OPT-HK)/\sqrt{N}$ has a substantially lower variance than either OPT or HK.
- Estimate
  - C_{HK} based on instances from N=100 to 316,228, using heuristics and **Concorde**-based error estimates
  - $(C- C_{HK})$ based on instances with N = 100, 316, 1000, using **Concorde** for $N \leq 316$ and Iterated Lin-Kernighan plus **Concorde**-based error estimates for N = 1000.

Modern Approach: Use Concorde

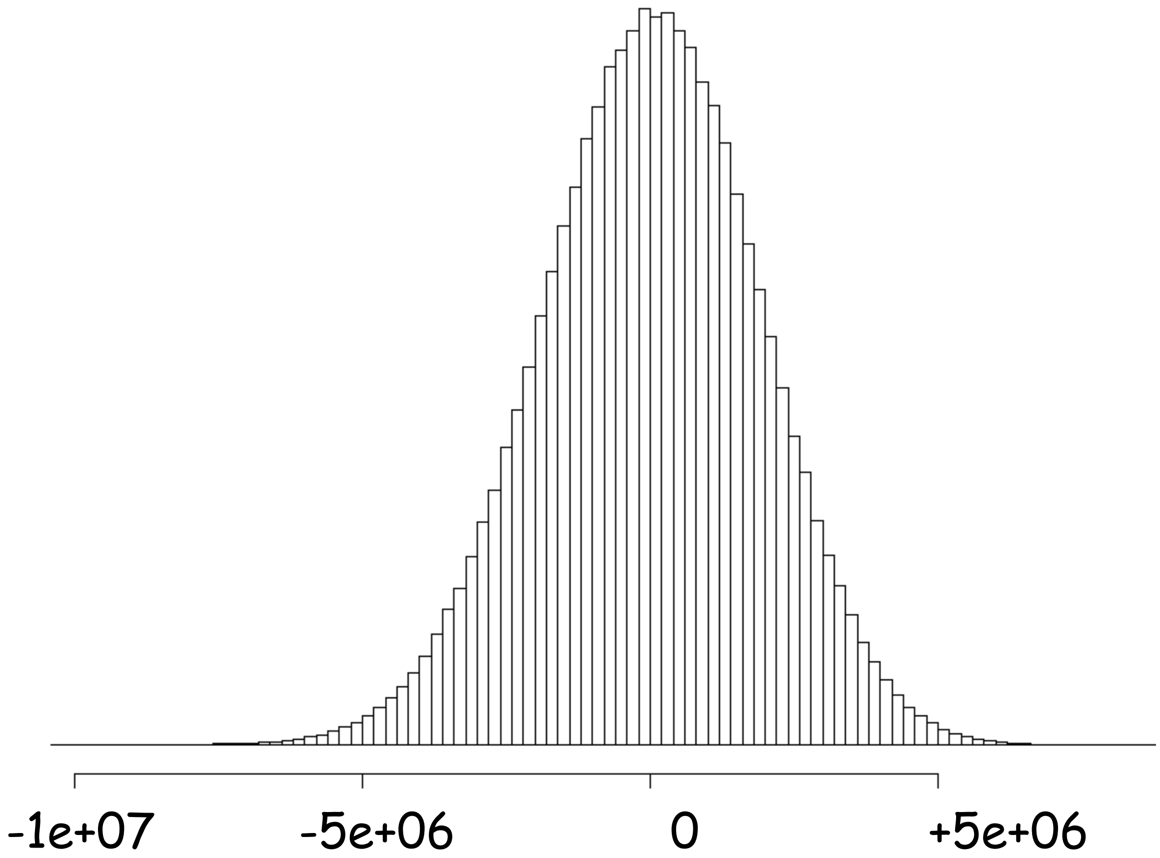
- Can compute true optima and Held-Karp for Toroidal as well as Euclidean.
- Faster for Toroidal than for Euclidean.



'times.1000.hist'
Running times (in seconds) for 10,000 Concorde runs on random 1000-city planar Euclidean instances (2.66 Ghz Intel Xeon processor in dual-processor PC, purchased late 2002).
Range: 7.1 seconds to 38.3 hours
700
600
500
400
300
200
100
0
1
10
100
1000
10000
100000
1e+06

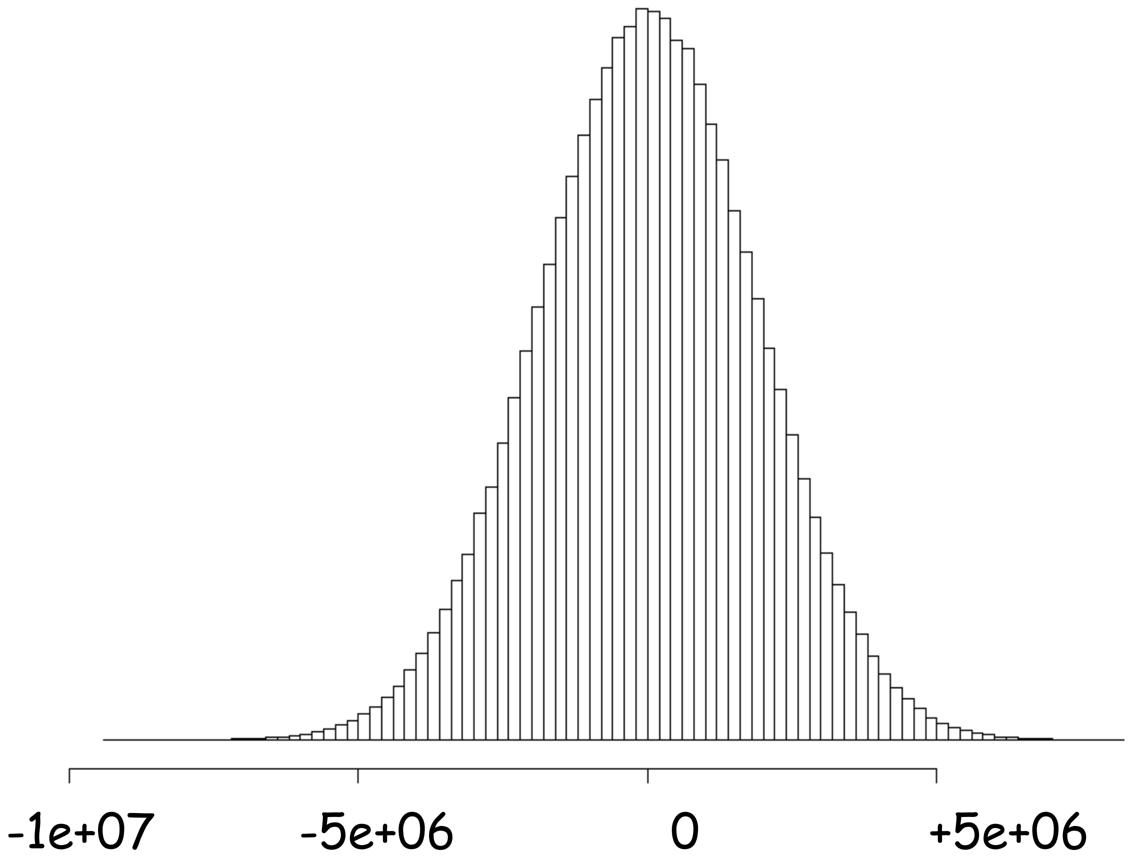

80000
70000
60000
50000
40000
30000
20000
10000
0
1
10
100
1000
10000
100000
'ttimes.hist'
Running times (in seconds) for 1,000,000 Concorde runs on random 1000-city "Toroidal" Euclidean instances
Range: 2.6 seconds to 6 hours

Optimal Tour Lengths: One Million 100-City Instances



Optimal Tour Lengths Appear to Be Normally Distributed

Optimal Tour Lengths:
One Million 1000-City Instances



With a standard deviation that appears to be independent of N

The New Data

- Solver:
  - Latest (2003) version of **Concorde** with a few bug fixes and adaptations for new metrics
- Primary Random Number Generator:
  - **RngStream** package of Pierre L'Ecuyer, described in
    - "AN OBJECT-ORIENTED RANDOM-NUMBER PACKAGE WITH MANY LONG STREAMS AND SUBSTREAMS," Pierre L'ecuyer, Richard Simard, E. Jack Chen, W. David Kelton, Operations Research 50:6 (2002), 1073-1075

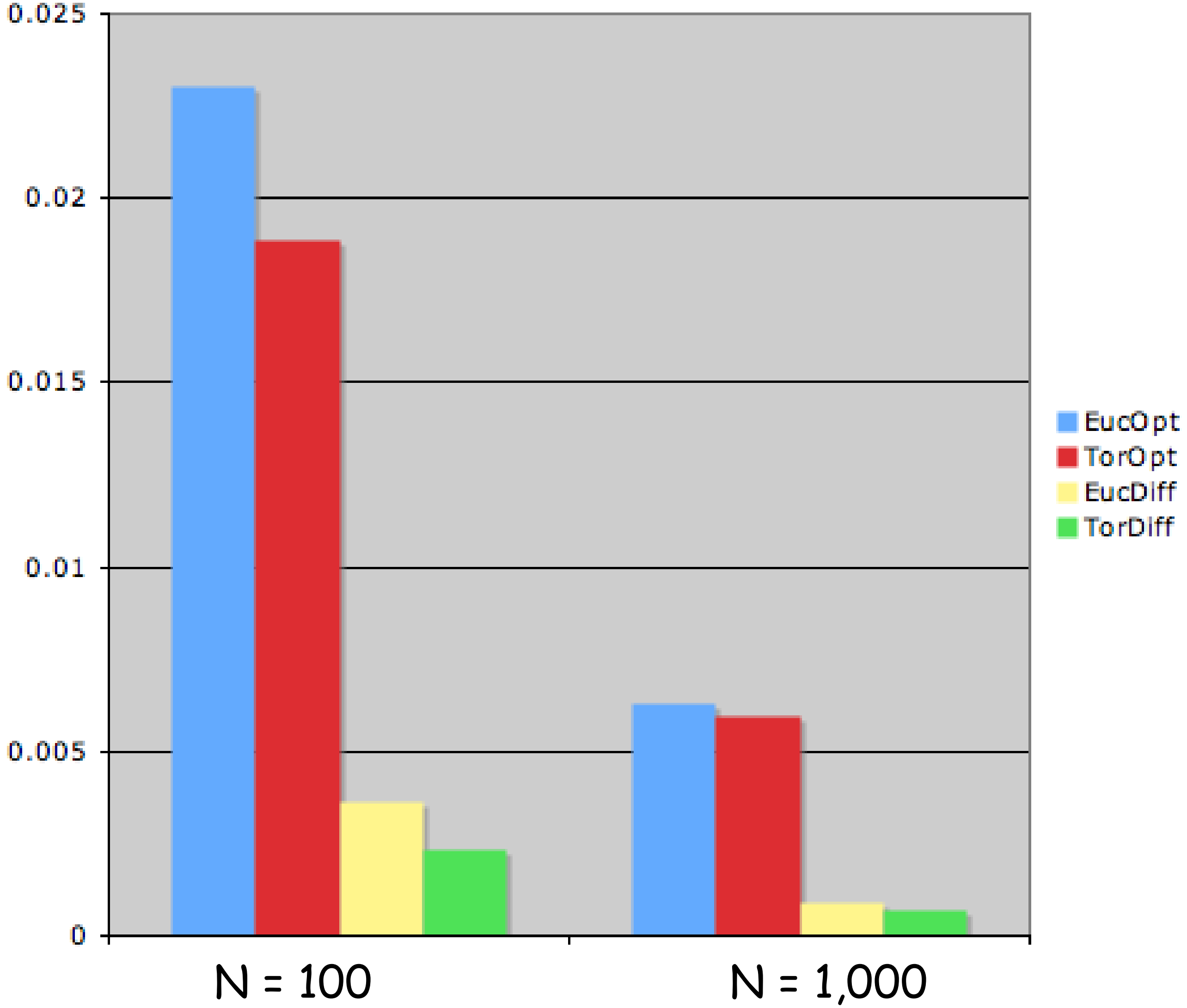
Toroidal Instances

Number of Cities	Number of Instances	OPT	HK
N = 3, 4, …, 49, 50	1,000,000	X	X
N = 60, 70, 80, 90, 100	1,000,000	X	X
N = 200, 300, …, 1,000	1,000,000	X	X
N = 110, 120, …, 1,900	10,000	X	X
N = 2,000	100,000	X	X
N = 2,000, 3,000, …, 10,000	1,000,000		X
N = 100,000	1,000		X
N = 1,000,000	100		X

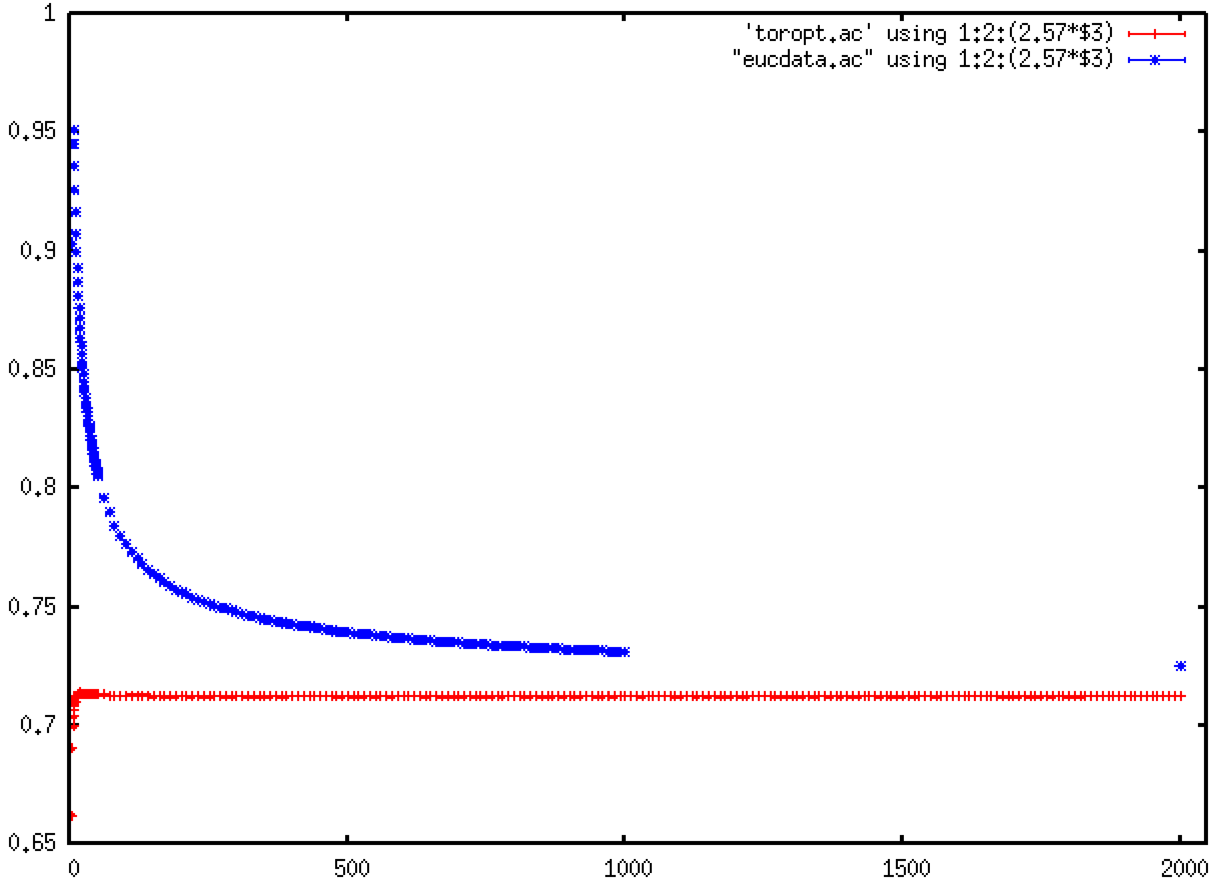
Euclidean Instances

Number of Cities	Number of Instances	OPT	HK
N = 3, 4, …, 49, 50	1,000,000	X	X
N = 60, 70, 80, 90, 100	1,000,000	X	X
N = 110, 120, …, 1,000, 2,000	10,000	X	X
N = 1,100, 1,200 …, 10,000	10,000		X
N = 20,000, 30,000, …, 100,000	10,000		X
N = 1,000,000	1,000		X

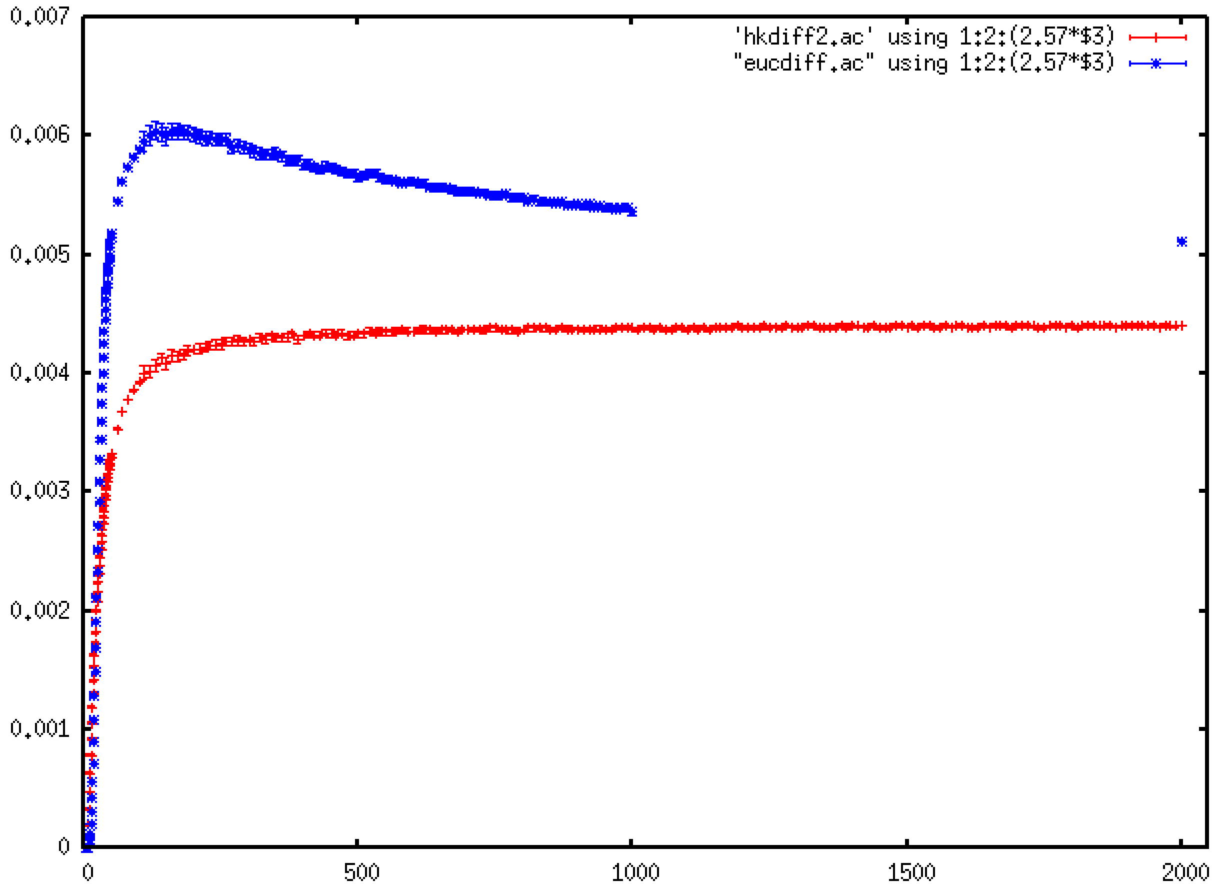
Standard Deviations



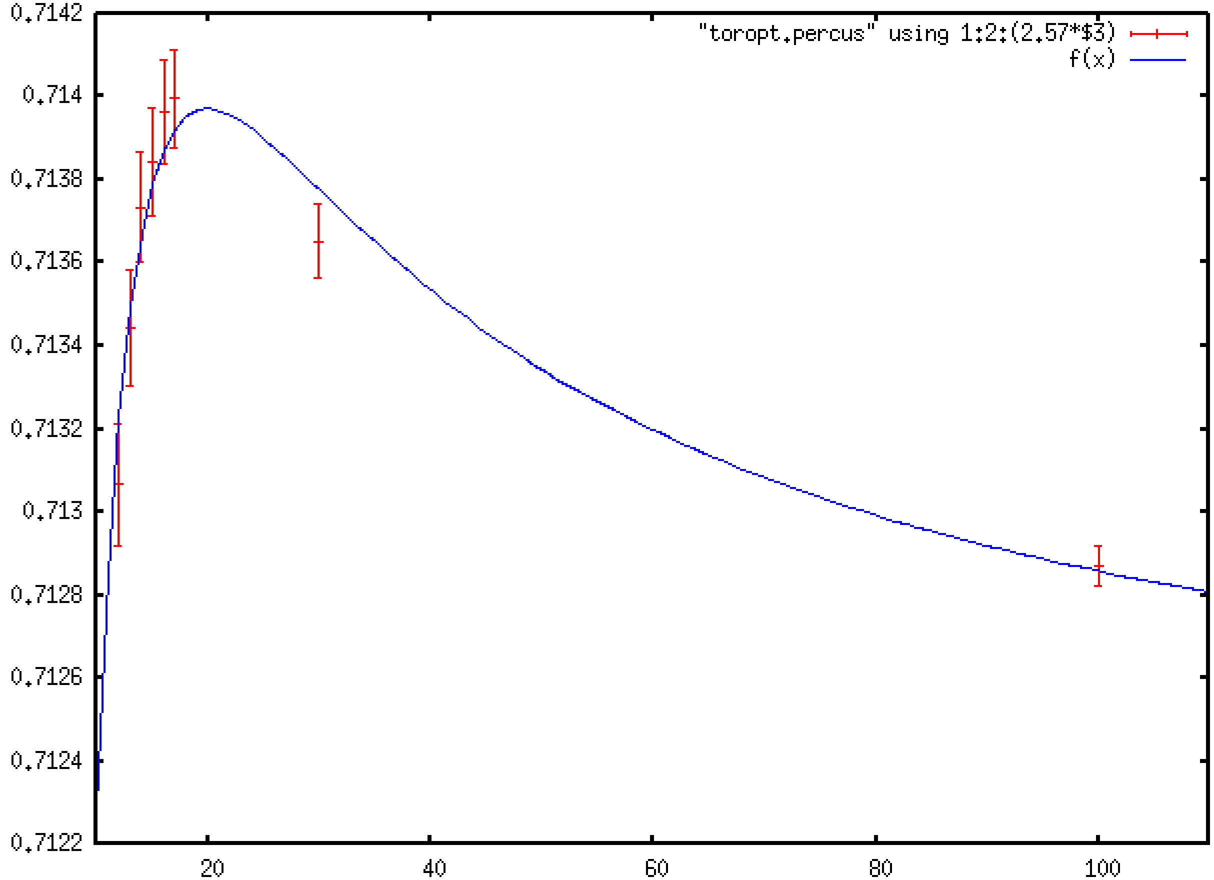
99% Confidence Intervals for OPT/√N for Euclidean and Toroidal Instances



99% Confidence Intervals for (OPT-HK)/√N for Euclidean and Toroidal Instances

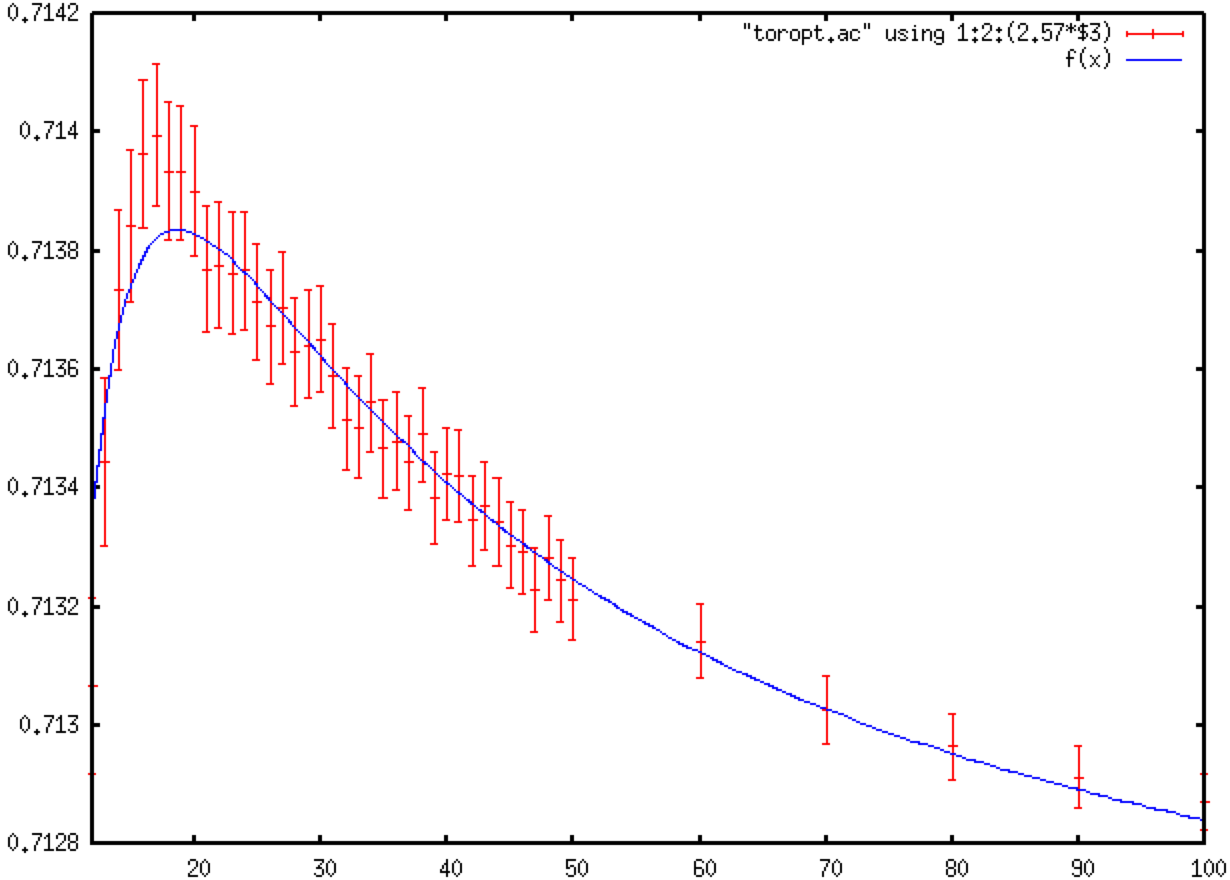


Gnuplot Least Squares fit for the Percus-Martin values of N -- $OPT/\sqrt{N} = (C + a/N + b/N^2)/(1+1/(8N))$



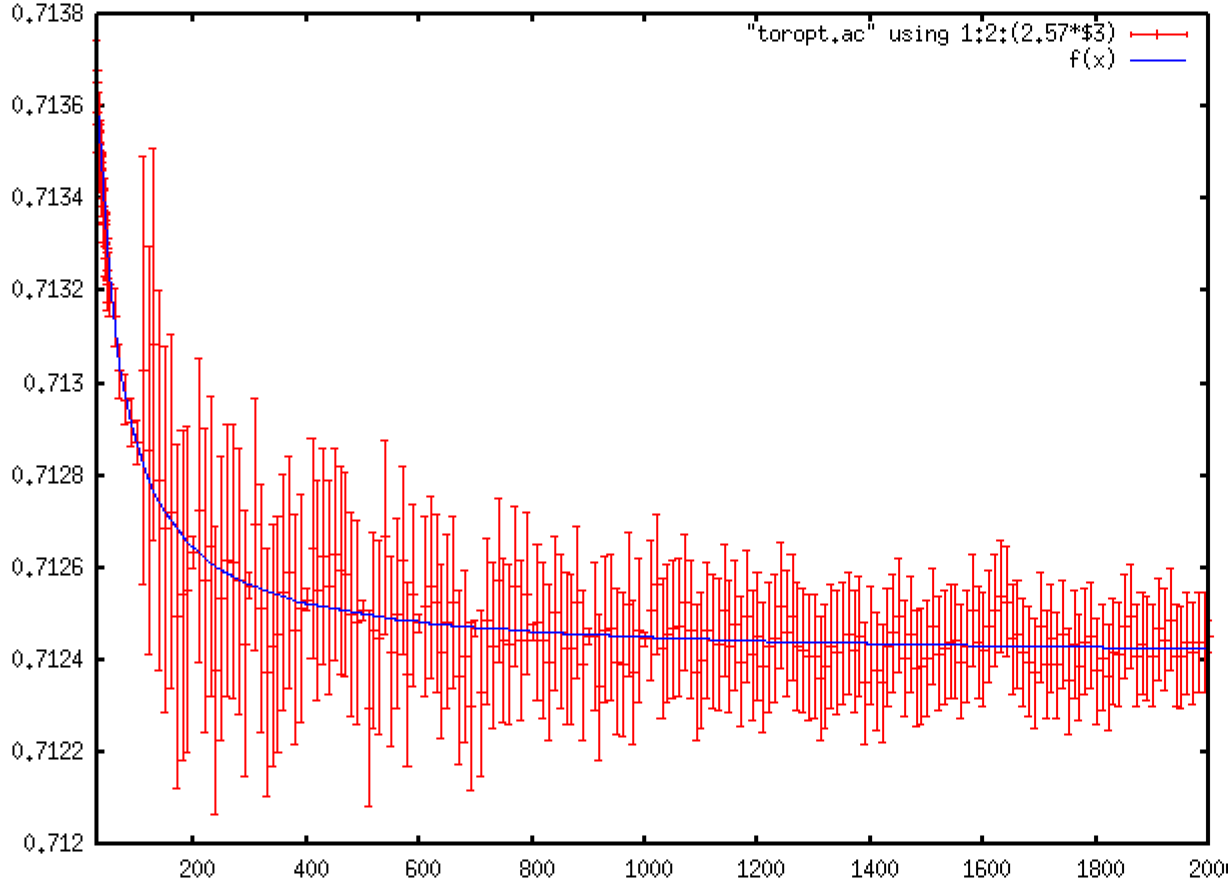
C = .712234 ± .00017 versus originally claimed C = .7120 ± .0002

Least Squares fit for all data from [12,100] -- OPT/√N = (C + a/N + b/N²)



C = .712333 ± .00006 versus *C* = .712234 ± .00017

Least Squares fit for all data from [30,2000] -- $OPT/\sqrt{N} = (C + a/N + b/N^2)$



$C = .712401 \pm .000005$ versus $C = .712333 \pm .00006$

What is the right function?

Power Series in 1/N – the Percus-Martin Choice

Range of N	Function	C	Confidence
[30,2000]	$C + a/N + b/N^2$	.712401	± .000005
[100,2000]	$C + a/N + b/N^2$	.712403	± .000010
[100,2000]	$C + a/N$	.712404	± .000006

Justification: Expected distance to the k^{th} nearest neighbor is provably such a power series.

What is the right function?

OPT/sqrt(N) = Power Series in 1/sqrt(N))

Range of N	Function	C	Confidence
[100,2000]	$C + a/N^{0.5}$	.712296	± .000015
[100,2000]	$C + a/N^{0.5} + b/N$	.712403	± .000030
[100,2000]	$C + a/N^{0.5} + b/N + c/N^{1.5}$	.712424	± .000080

Justification: This is what we saw in the planar Euclidean case (although it was caused by boundaries).

What is the right function?

OPT = (1/sqrt(N) · (Power Series in 1/N)

Range of N	Function	C	Confidence
[100,2000]	$C + a/N^{0.5}$	.712296	± .000015
[100,2000]	$C + a/N^{0.5} + b/N^{1.5}$	.712366	± .000022
[100,2000]	$C + a/N^{0.5} + b/N^{1.5} + c/N^{2.5}$	.712385	± .000040

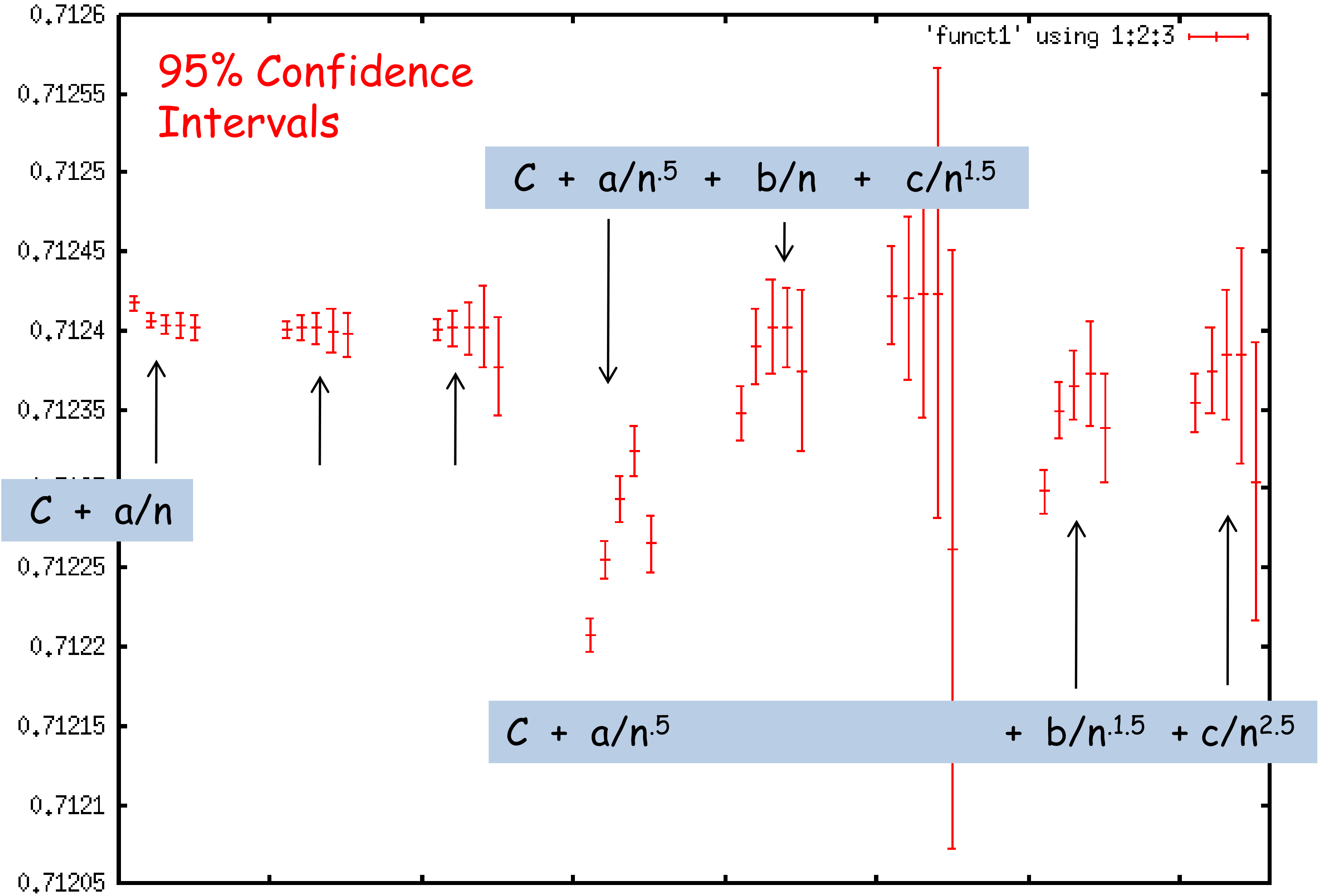
Justification: Why not?

What is the right function?

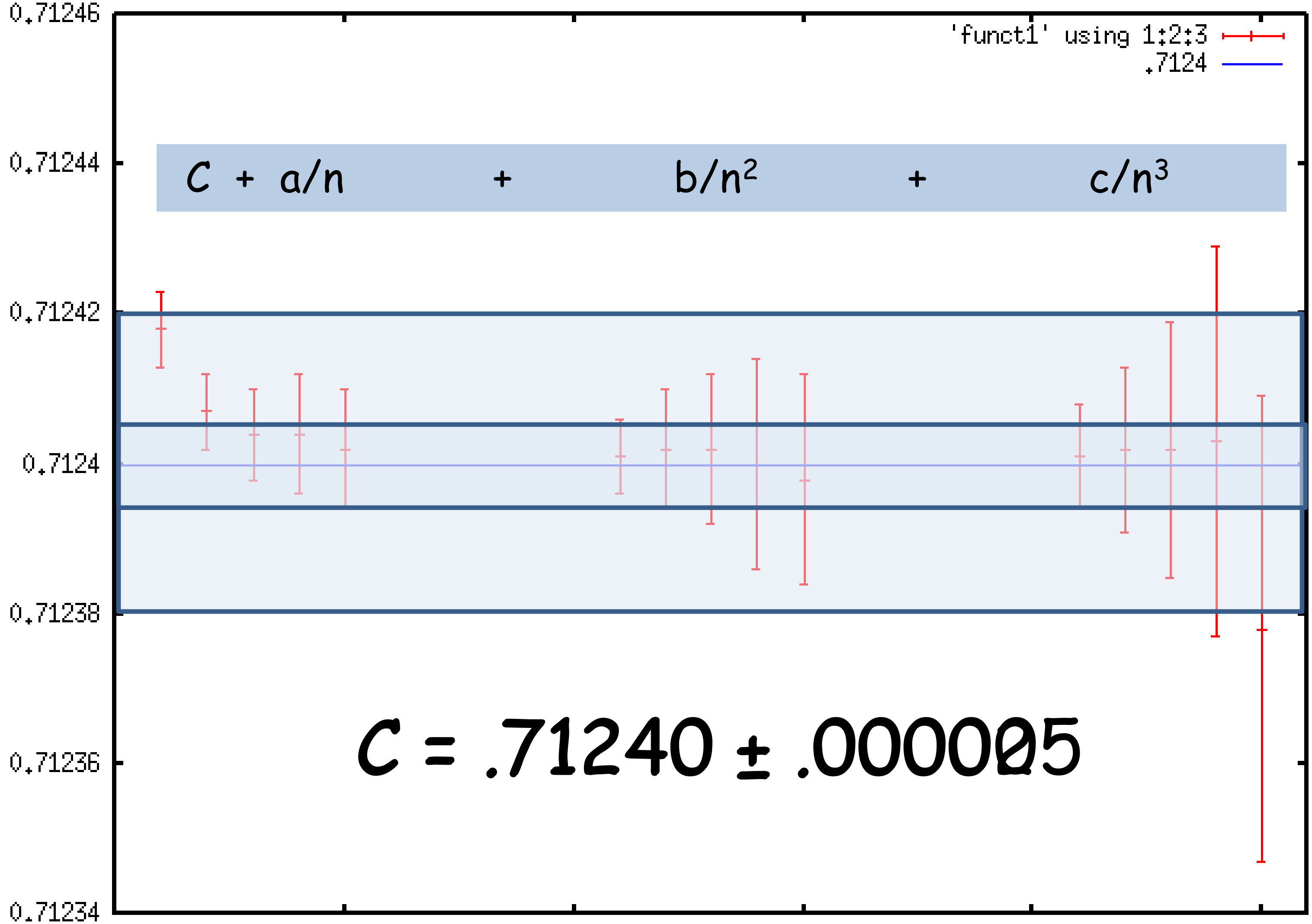
Range of N	Function	C	Confidence
[30,2000]	$C + a/N + b/N^2$	.712401	± .000005
[100,2000]	$C + a/N + b/N^2$	.712403	± .000010
[100,2000]	$C + a/N$	.712404	± .000006
[100,2000]	$C + a/N^{0.5}$	.712296	± .000015
[100,2000]	$C + a/N^{0.5} + b/N$	.712403	± .000030
[100,2000]	$C + a/N^{0.5} + b/N + c/N^{1.5}$	.712424	± .000080
[100,2000]	$C + a/N^{0.5} + b/N^{1.5}$	.712366	± .000022
[100,2000]	$C + a/N^{0.5} + b/N^{1.5} + c/N^{2.5}$	.712385	± .000040

Effect of Data Range on Estimate

[30,2000], [60,2000], [100,2000], [200,2000], [100,1000]



The Winners?

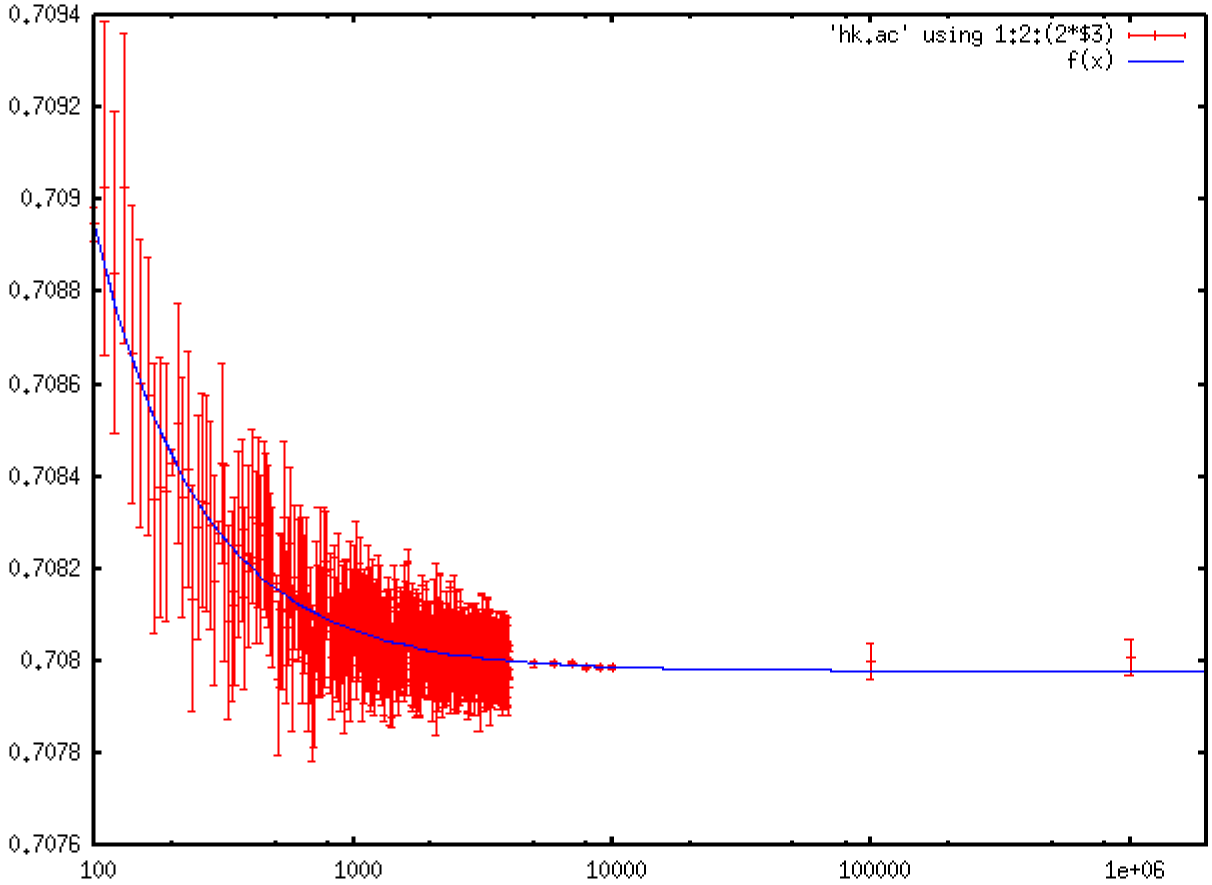


0.71246
0.71244
0.71242
0.7124
0.71238
0.71236
0.71234
'funct1' using 1:2:3
.7124
C + a/n + b/n2 + c/n3
C = .71240 ± .000005

Question

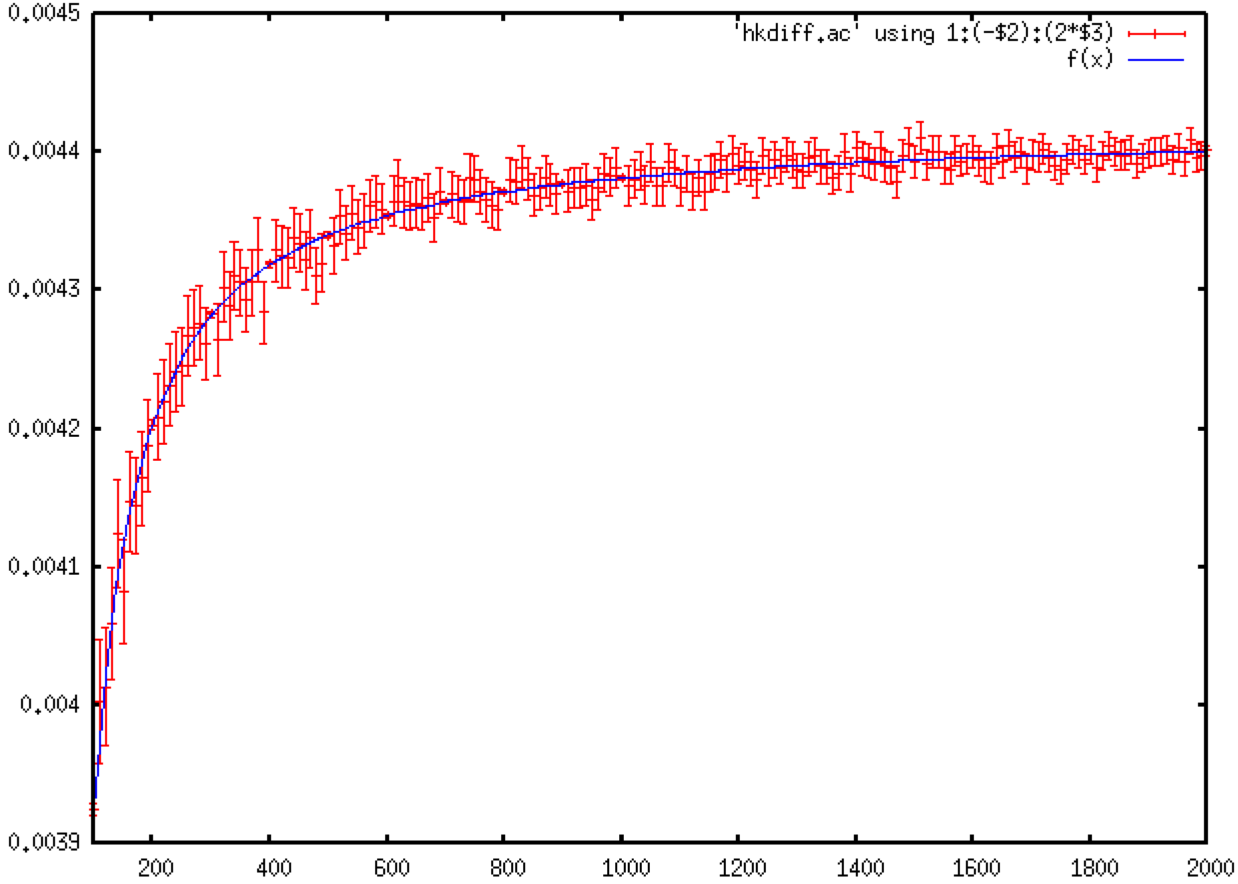
Does the HK-based approach agree?

$C_{HK} = .707980 \pm .000003$



95% confidence interval derived using $C + a/N + b/N^2$ functional form

$C - C_{HK} = .004419 \pm .000002$



95% confidence interval derived using $C + a/N + b/N^2$ functional form

HK-Based Estimate

$C - C_{HK} = .004419 \pm .000002$

$+ \quad C_{HK} = .707980 \pm .000003$

$C = .712399 \pm .000005$

Versus (Conservative) Opt-Based Estimate

$C = .712400 \pm .000020$

Combined Estimate?

$C = .71240 \pm .00001$

OPEN PROBLEM:

What function truly describes the data?

Our data suggests $OPT/sqrt(N) \approx$

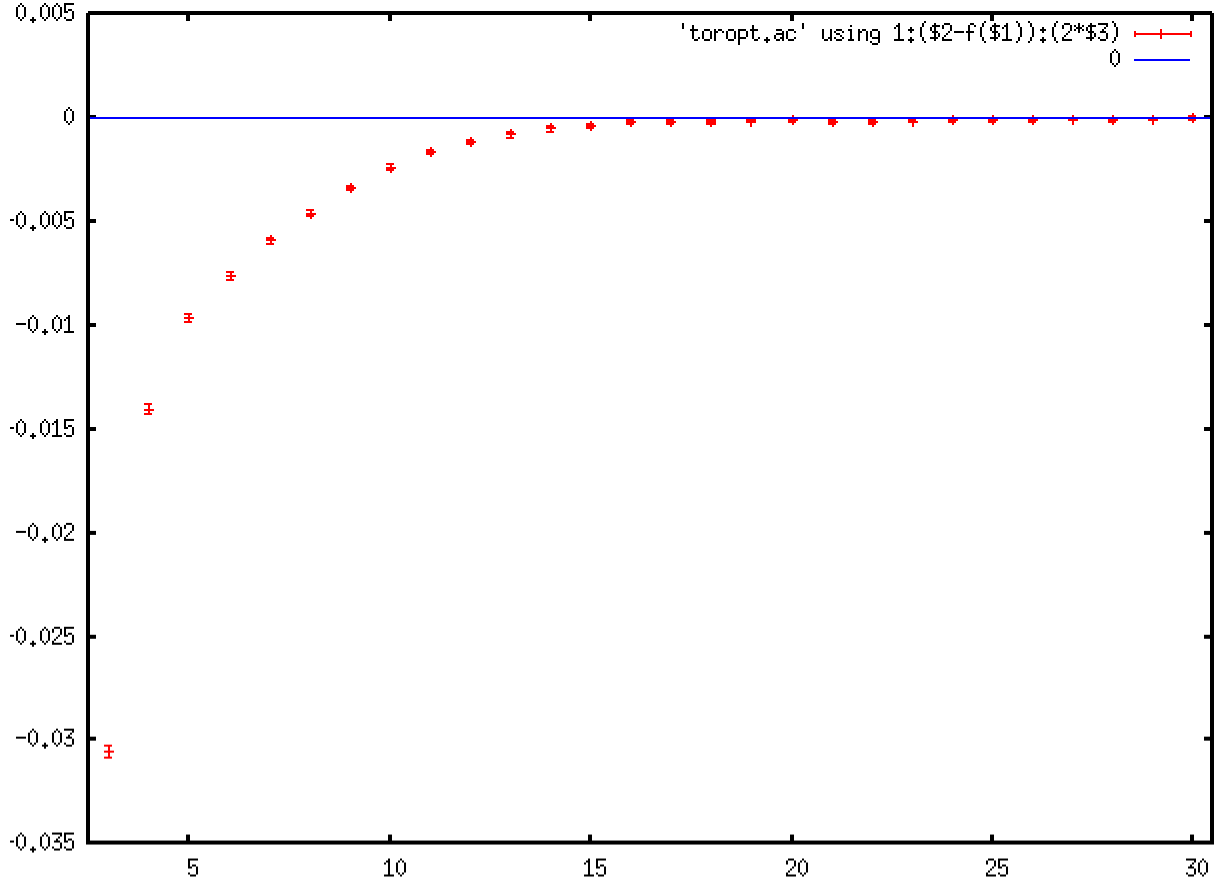
$$.71240 + a/N - b/N^2 + O(1/N^3),$$

$$a = .049 \pm .004, \qquad b = .3 \pm .2$$

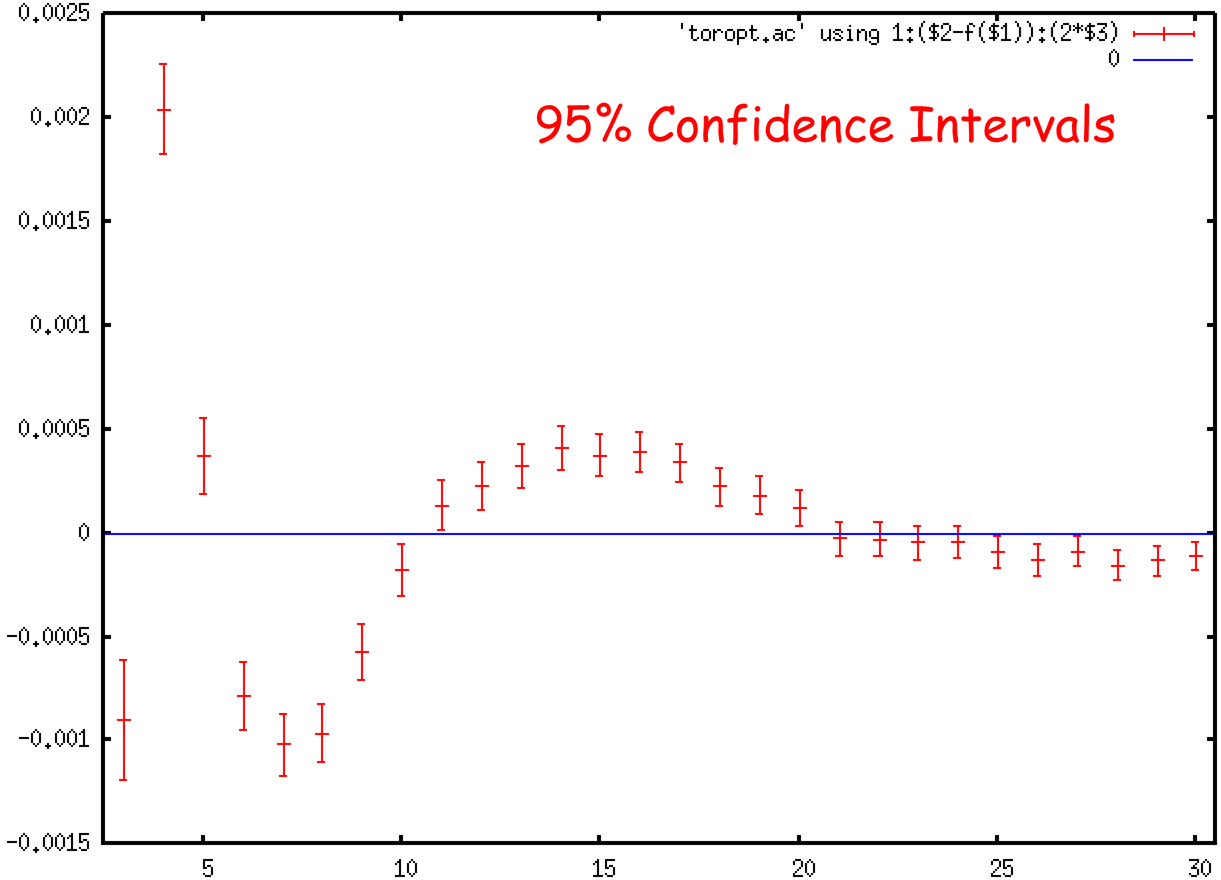
(from fits for ranges [60,2000] and [100,2000])

But what about the range [3,30]?

(95% confidence intervals on data) – f(N), $3 \le N \le 30$



Fit of $a + b/N + c/N^2 + d/N^3 + e/N^4$ for [3,30]

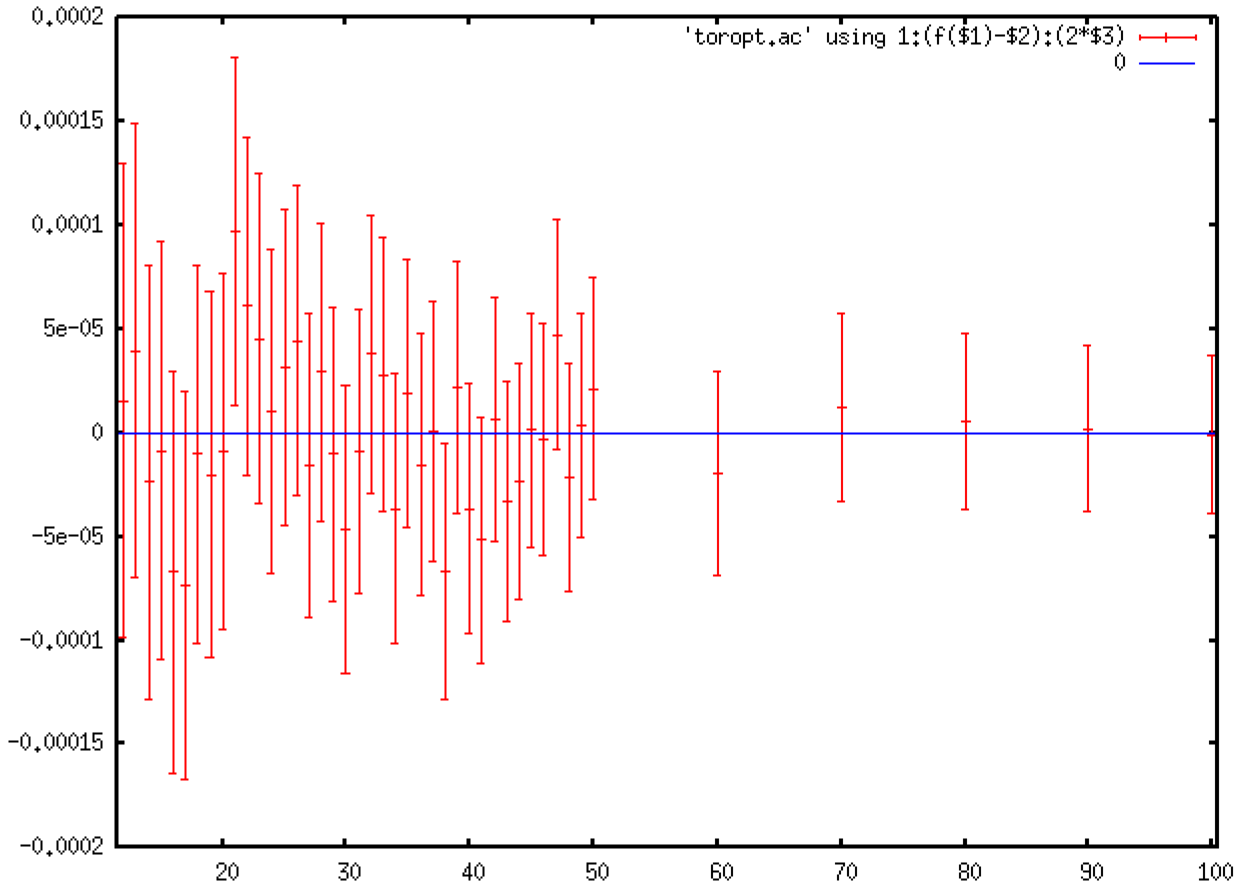


To date, no good fit of any sort has been found.

Problem

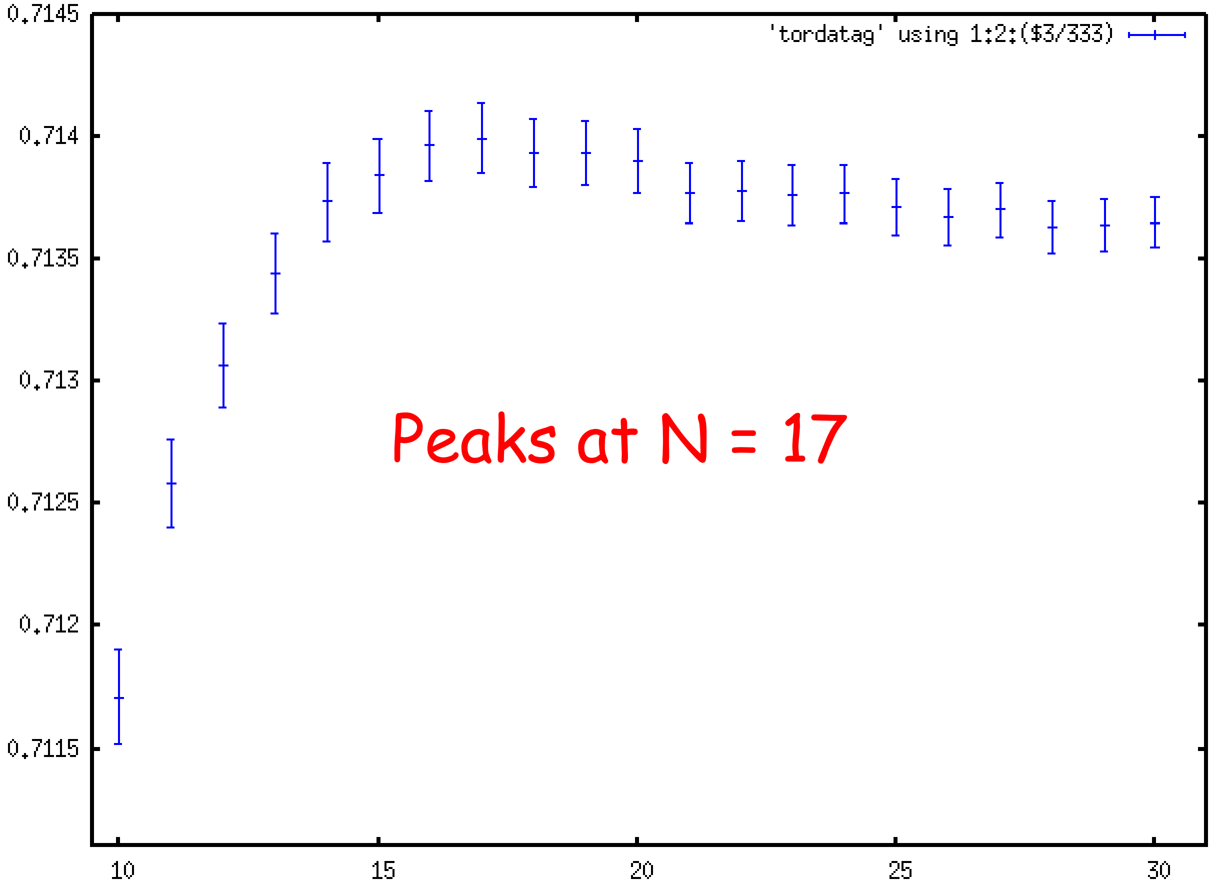
- Combinatorial factors for small N may make them unfittable:
  - Only one possible tour for N = 3 (expected length of optimal tour can be *given* in closed form)
  - Only 3, 12, 60, 420, … possible tours for N = 4, 5, 6, 7, …, so statistical mechanics phenomena may not yet have taken hold.
- So let's throw out data for N < 12

Fit of $a + b/N + c/N^2 + d/N^3 + e/N^4$ for [12,2000]

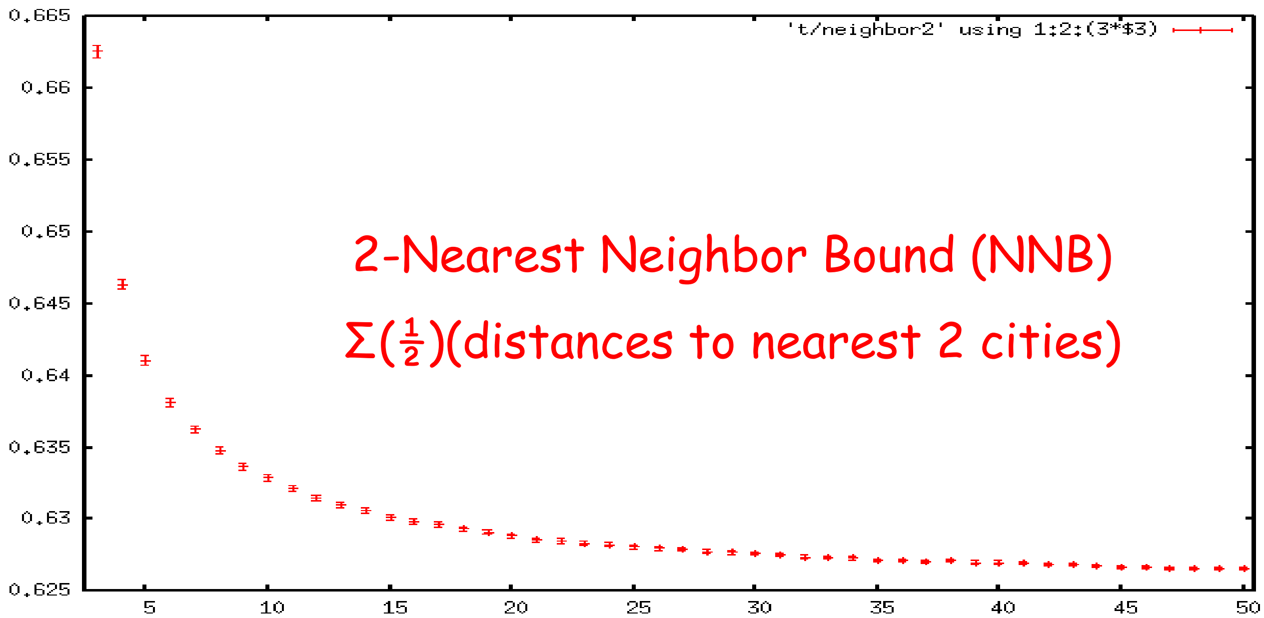


Still Questionable...

Unexplained Phenomenon: Rise and then Fall

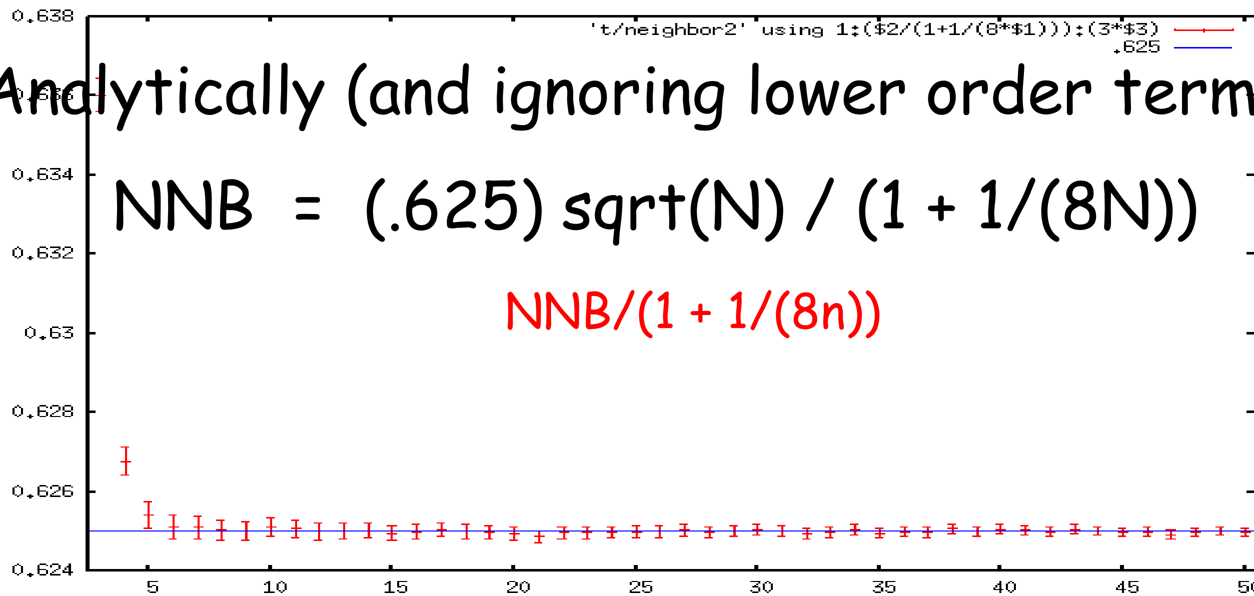


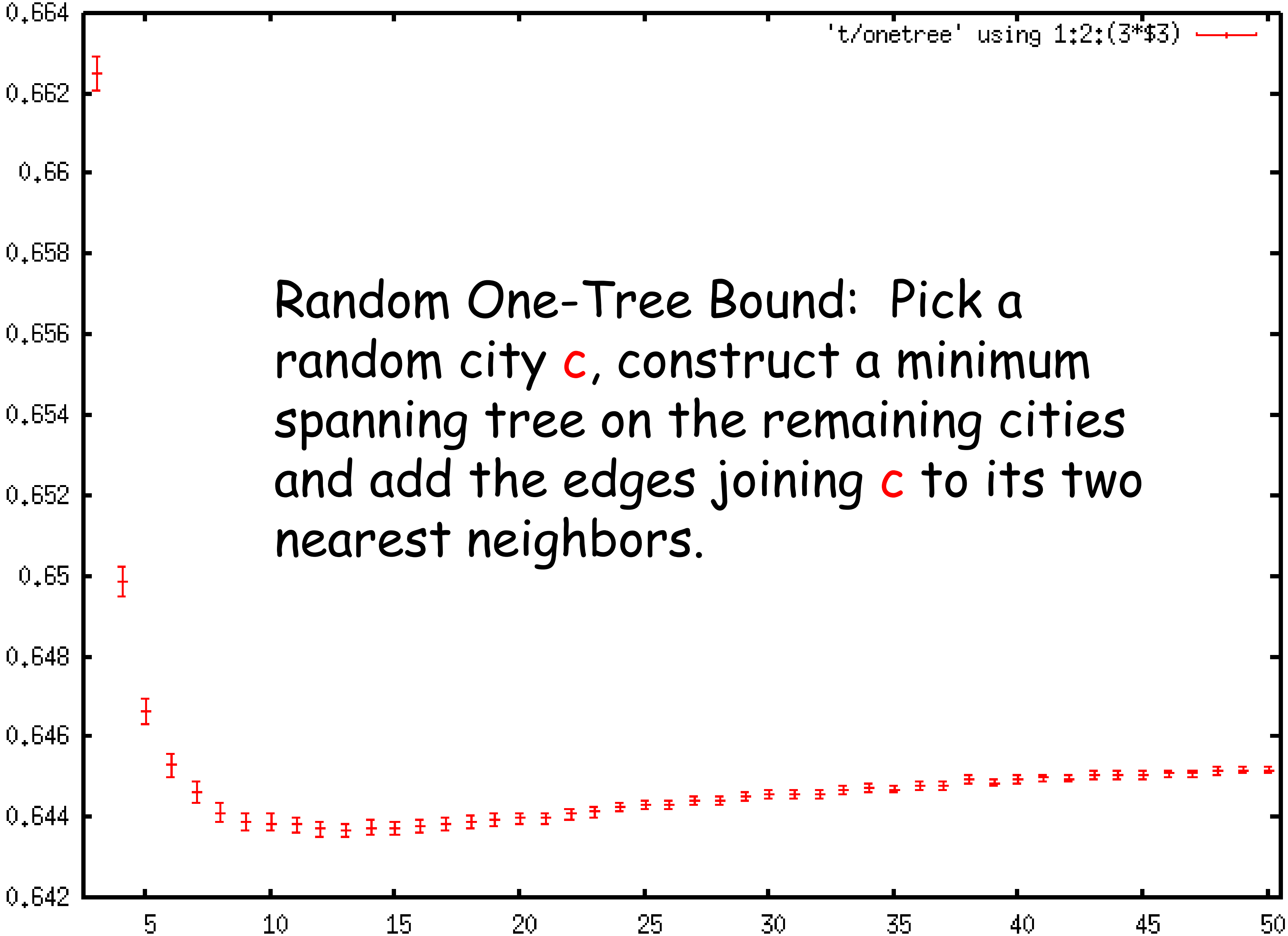
99.7% confidence intervals on OPT/$\sqrt{n}$,
$10 \leq n \leq 30$.



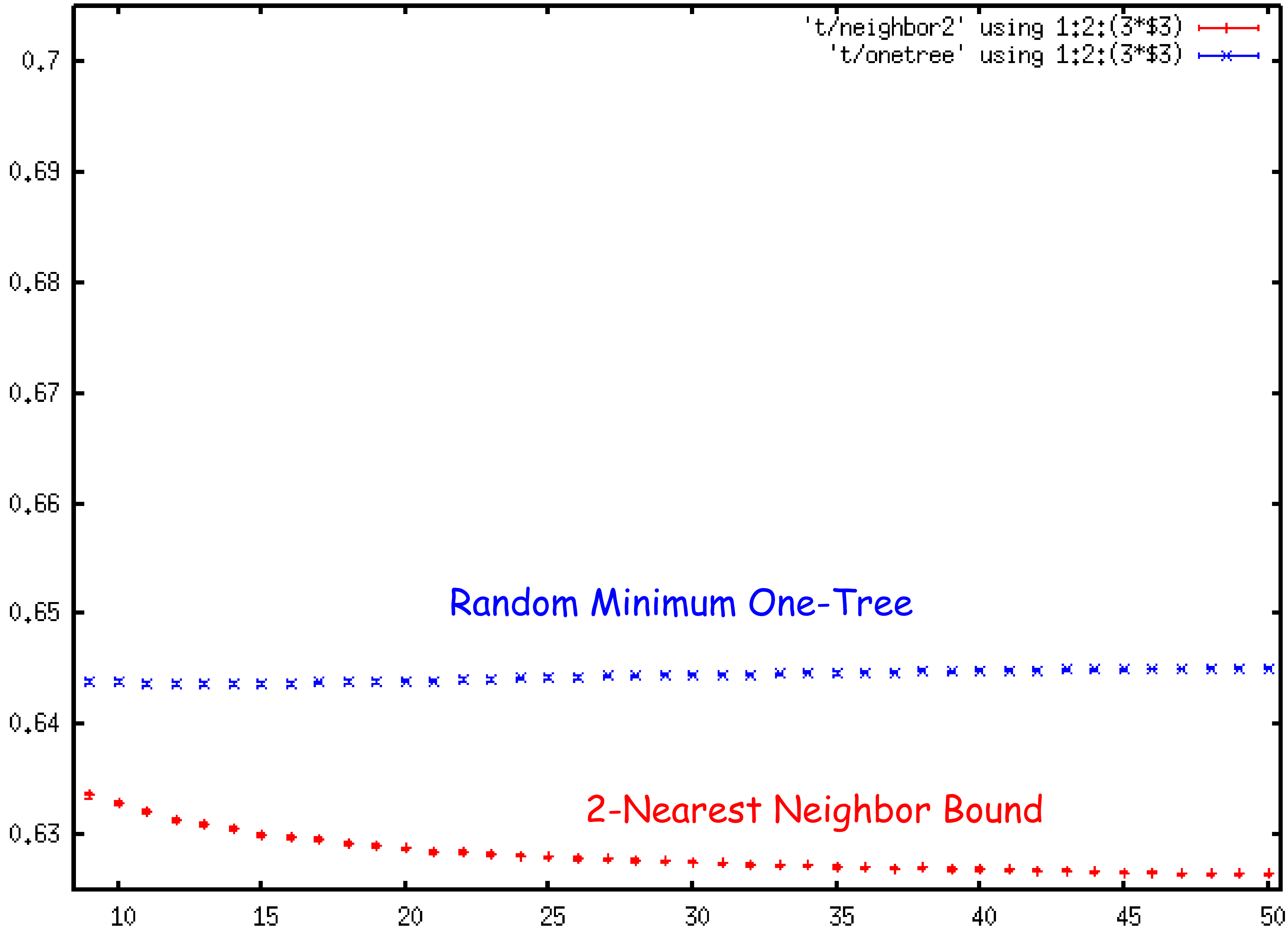
Analytically (and ignoring lower order terms),

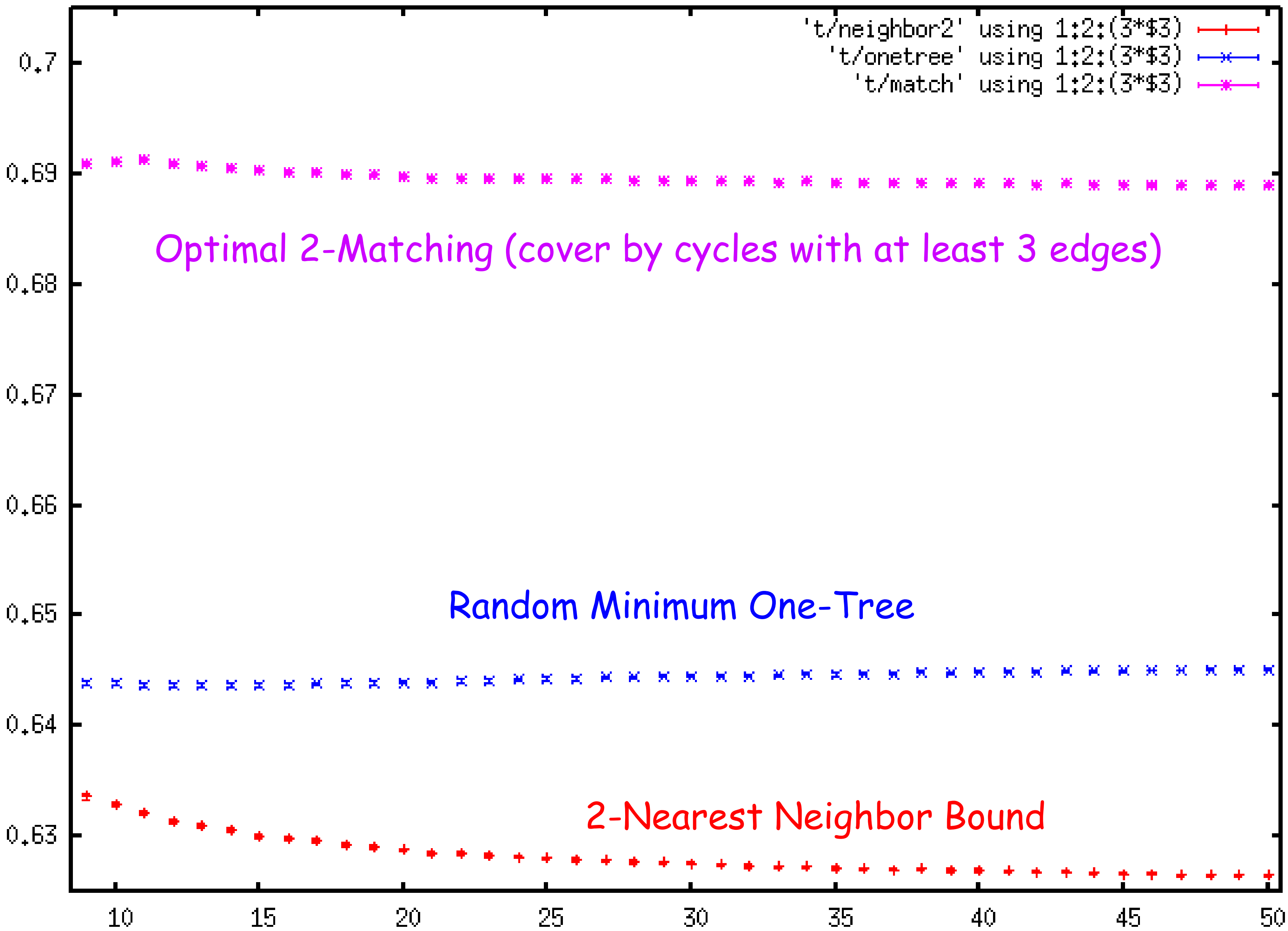
$$NNB = (.625)\, sqrt(N) / (1 + 1/(8N))$$



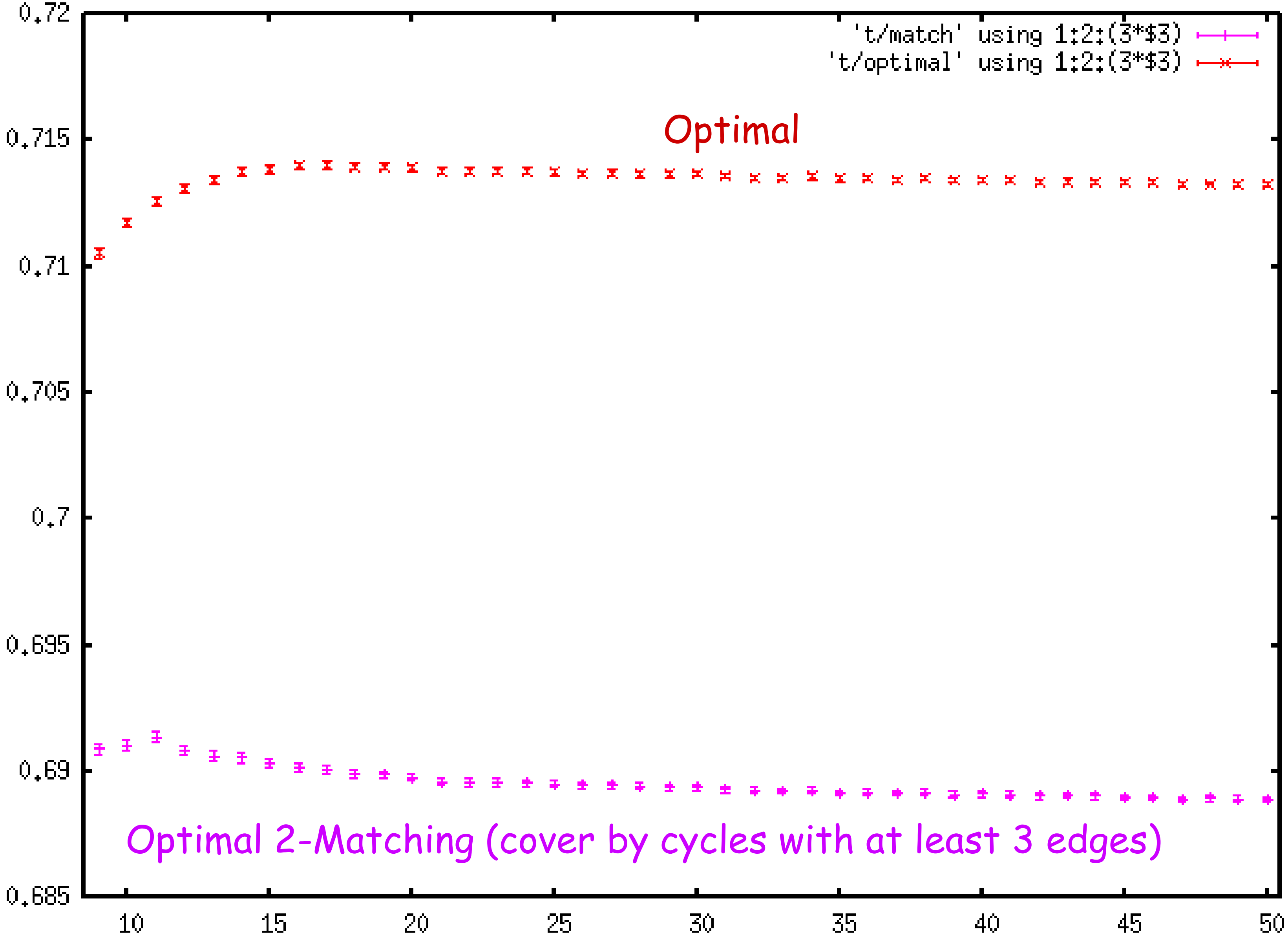


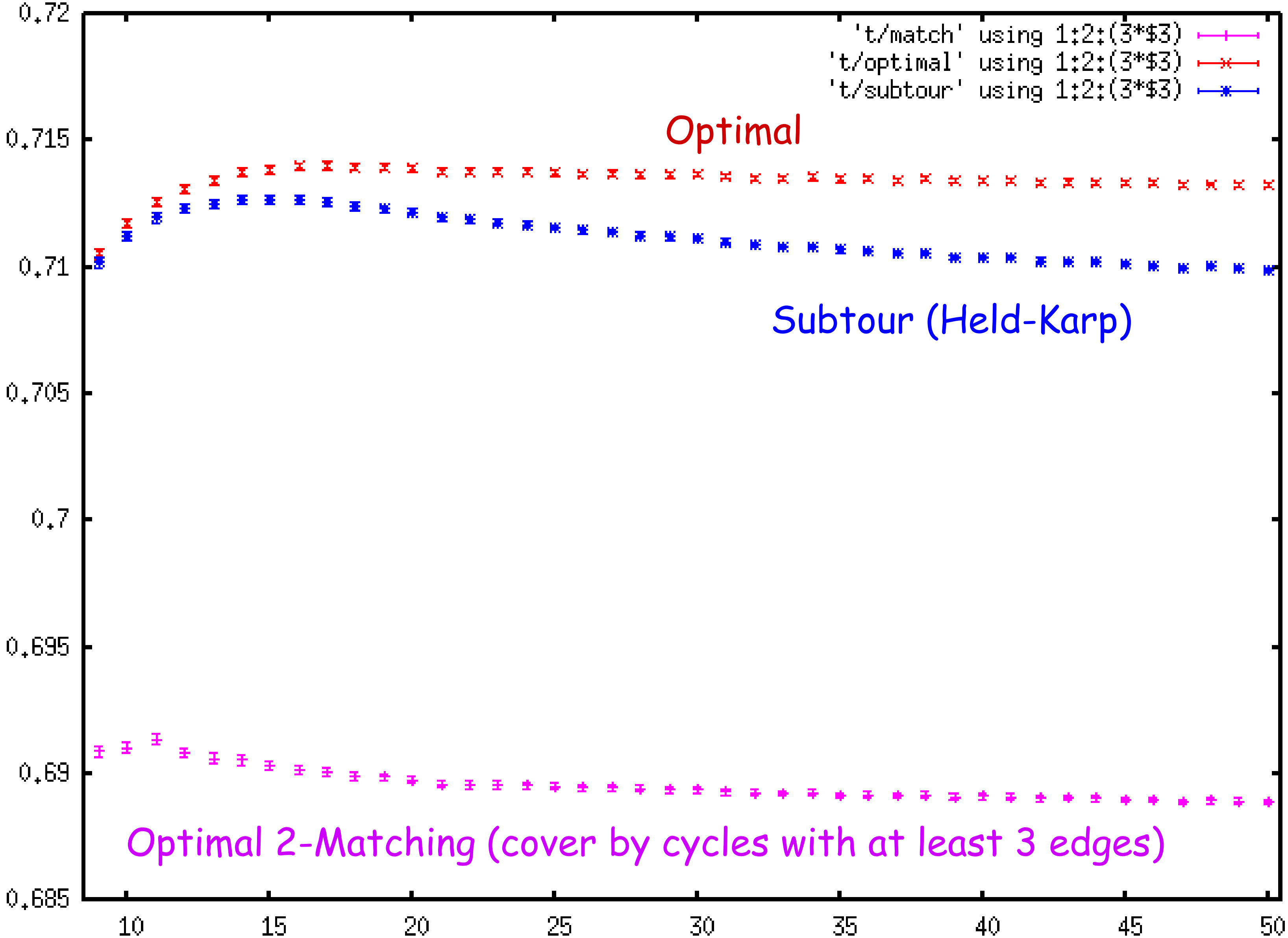
Random One-Tree Bound: Pick a random city c, construct a minimum spanning tree on the remaining cities and add the edges joining c to its two nearest neighbors.


't/neighbor2' using 1:2:(3*$3)
't/onetree' using 1:2:(3*$3)
Random Minimum One-Tree
2-Nearest Neighbor Bound
0.7
0.69
0.68
0.67
0.66
0.65
0.64
0.63
10
15
20
25
30
35
40
45
50



't/neighbor2' using 1:2:(3*$3)
't/onetree' using 1:2:(3*$3)
't/match' using 1:2:(3*$3)
0.7
0.69
0.68
0.67
0.66
0.65
0.64
0.63
10
15
20
25
30
35
40
45
50
Optimal 2-Matching (cover by cycles with at least 3 edges)
Random Minimum One-Tree
2-Nearest Neighbor Bound

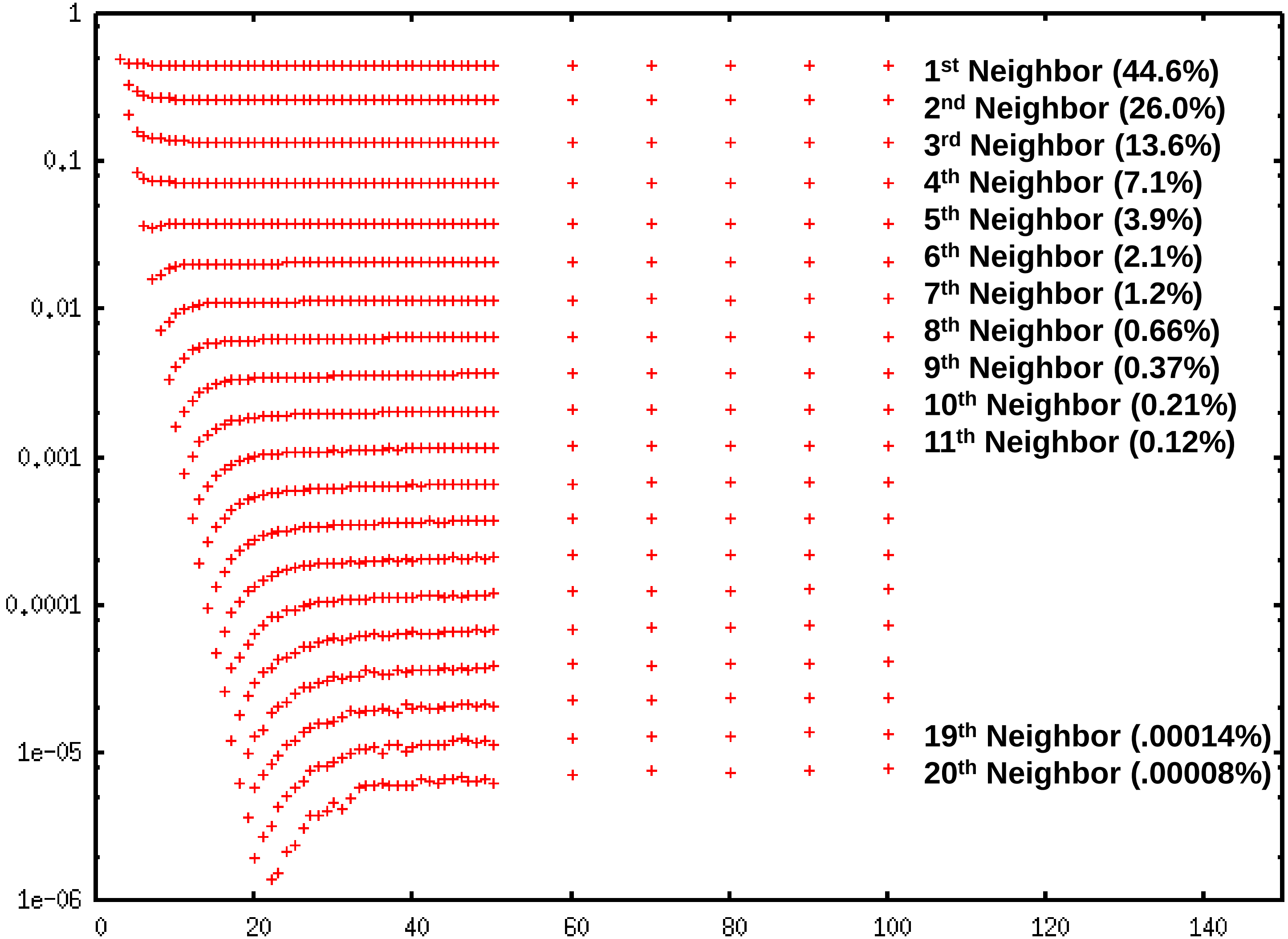

't/match' using 1:2:(3*$3)
't/optimal' using 1:2:(3*$3)
Optimal
Optimal 2-Matching (cover by cycles with at least 3 edges)
0.72
0.715
0.71
0.705
0.7
0.695
0.69
0.685
10
15
20
25
30
35
40
45
50


't/match' using 1:2:(3*$3)
't/optimal' using 1:2:(3*$3)
't/subtour' using 1:2:(3*$3)
Optimal
Subtour (Held-Karp)
Optimal 2-Matching (cover by cycles with at least 3 edges)
0.72
0.715
0.71
0.705
0.7
0.695
0.69
0.685
10
15
20
25
30
35
40
45
50

“Explaining“ The Expected Optimal Tour Length

- The fraction of optimal tour edges that go to k^{th} nearest neighbor seems to be going to a constant a_k for each k.

Fraction of Optimal Tour Edges



"Explaining" The Expected Optimal Tour Length

- The fraction of optimal tour edges that go to k^{th} nearest neighbor seems to be going to a constant a_k for each k.
- If d_k is the expected distance to your k^{th} nearest neighbor, we then get (asymptotically)

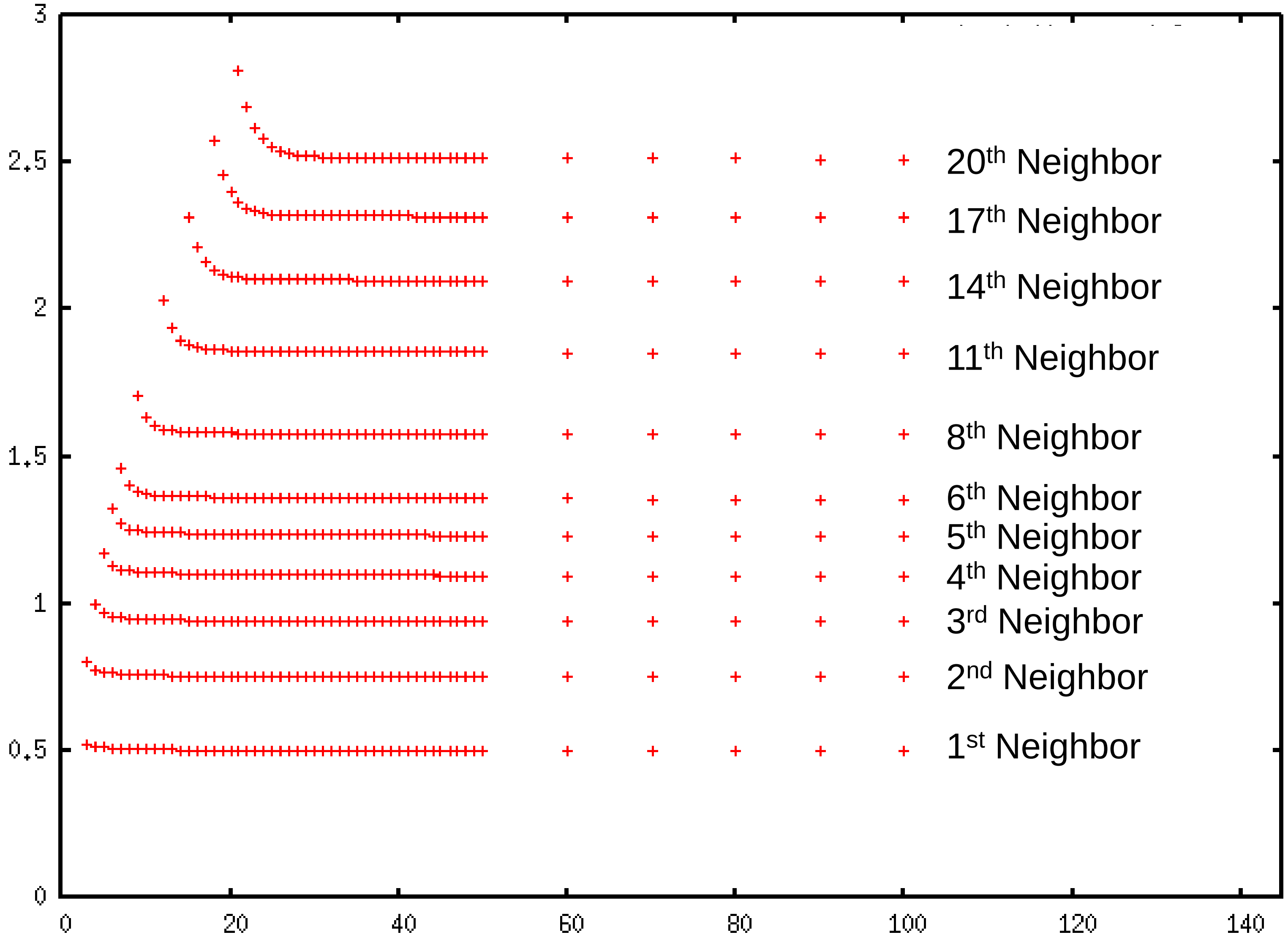
$$OPT_N \approx \sum_k (Na_k) d_k$$

- Or

$$OPT_N/sqrt(N) \approx \sum_k a_k (d_k sqrt(N))$$

- $d_k sqrt(N)$ also appears to go to a constant for each k

(√N)·(Average distance to k^{th} Nearest Neighbor)

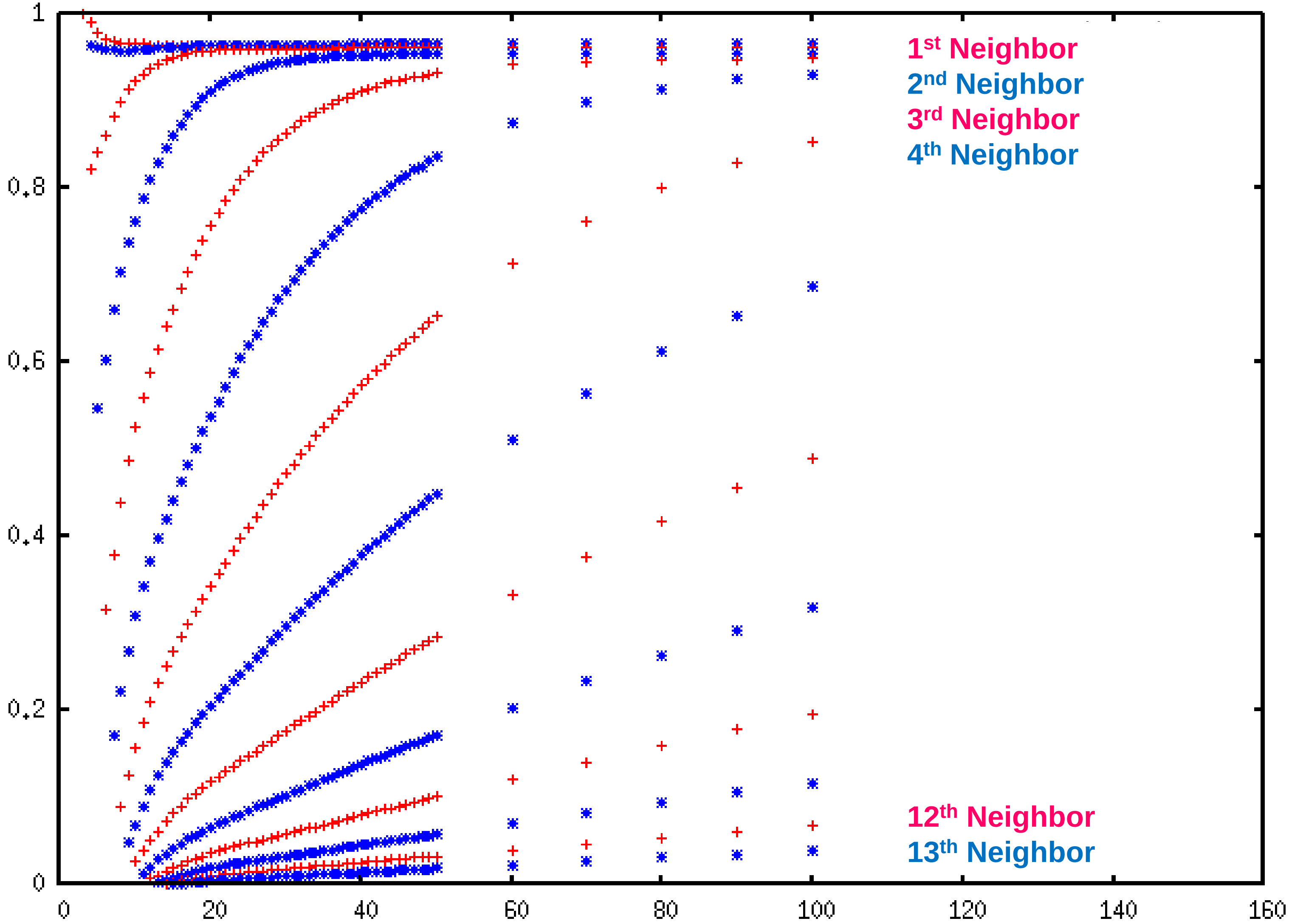


Hole in the Reasoning

Tour edges to k^{th} nearest neighbors are likely to be shorter than the *average* distance to a k^{th} nearest neighbor

K^{th} Nearest Neighbors

(Average length in optimal tour)/(Average length overall)



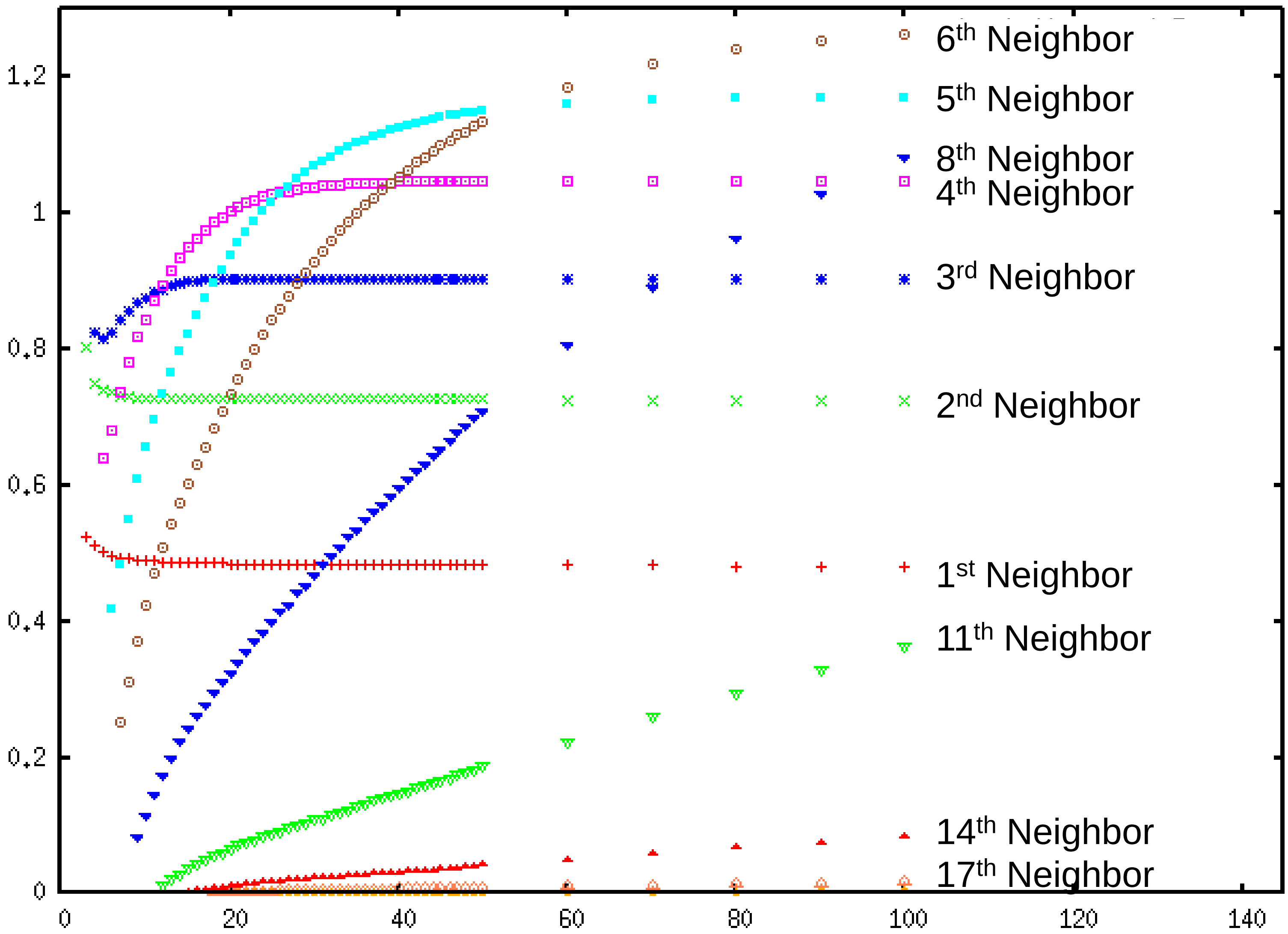
Suggests Balancing Phenomena

- Decrease in overall average distance to k^{th} nearest neighbor, approaching d_k from above
- Increase for each k in

$$\frac{\text{(average length of tour edges to } k^{th} \text{ nearest neighbors)}}{\text{(average distance to } k^{th} \text{ nearest neighbors overall)}}$$

- So how do these balance out?...

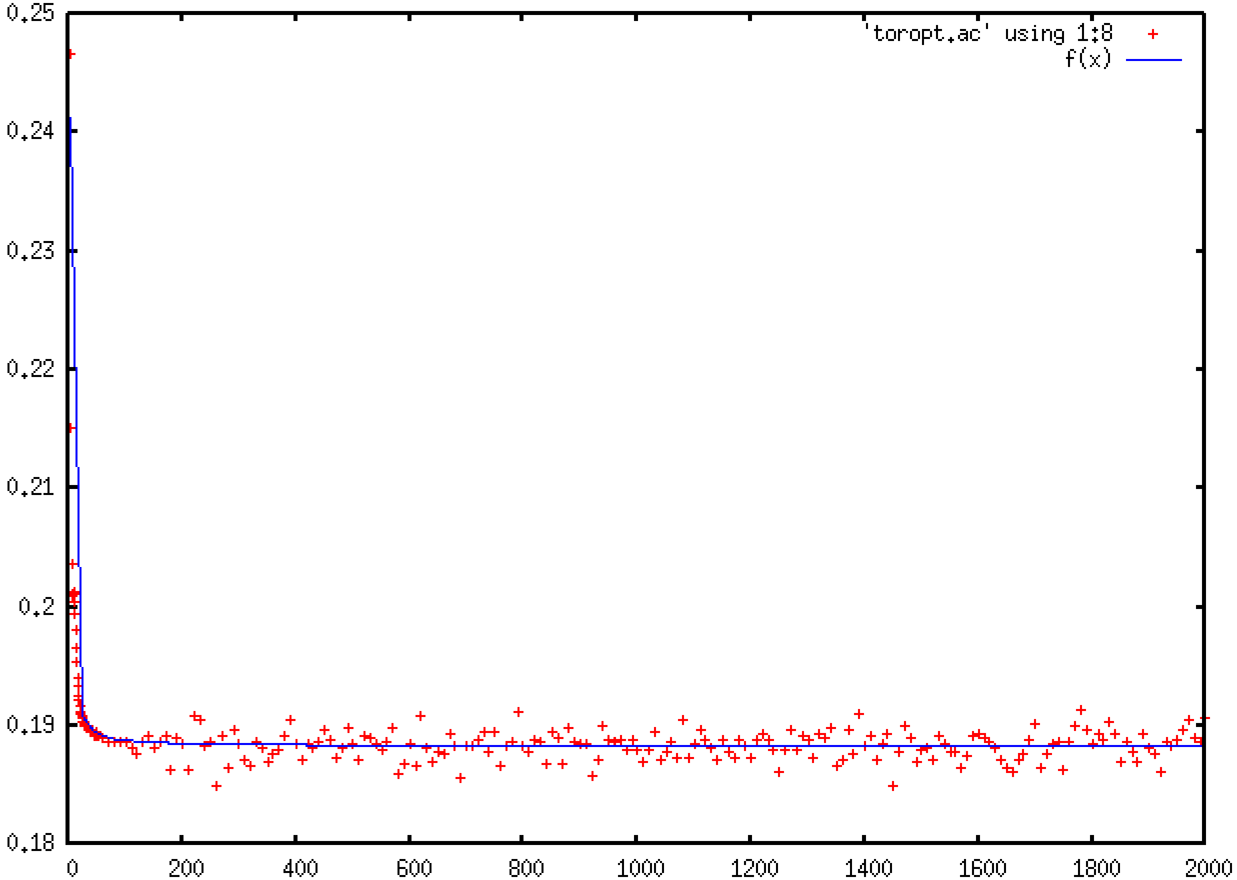
(√N)·(Average Length of k^th Nearest Neighbor Edges in Optimal Tour)



More Anomalies: Standard Deviations

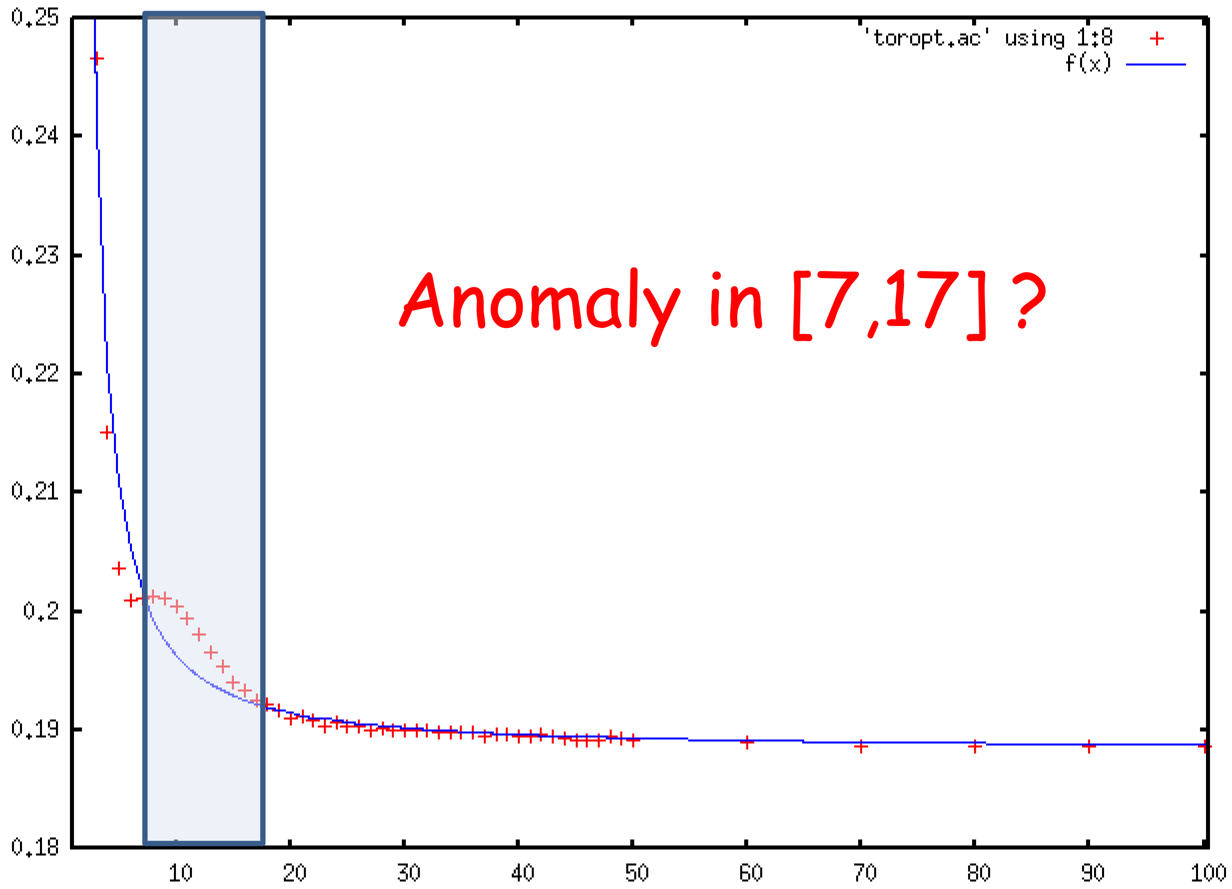
- [Cerf et al., 1997] conjectured that the standard deviation of OPT is asymptotic to a constant.
- Our data appears to confirm this.
- But what about the WAY it converges?

Standard Deviation for OPT (Fit to a + b/N)

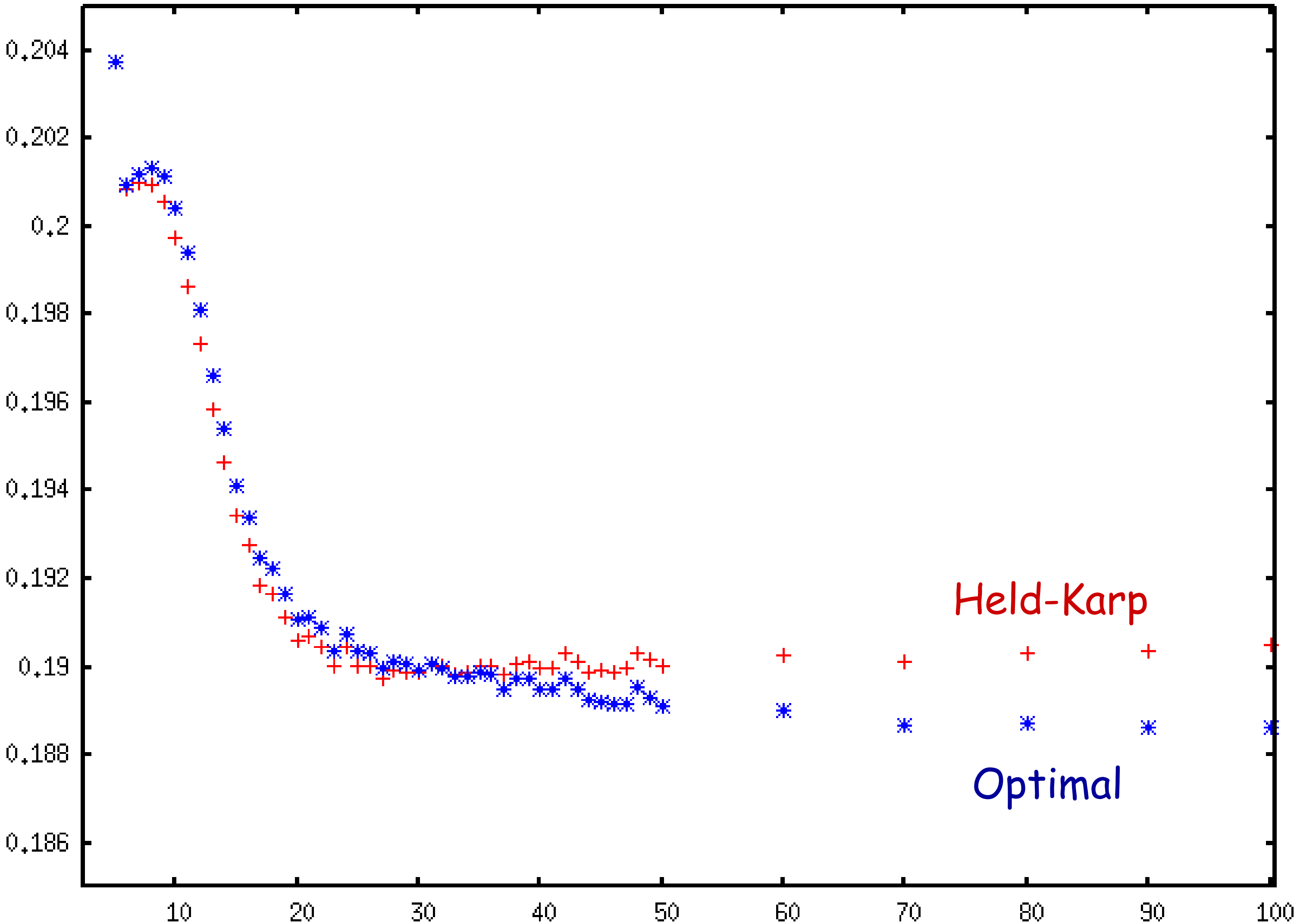


Asymptotic Std Dev = .1883 ± .0004

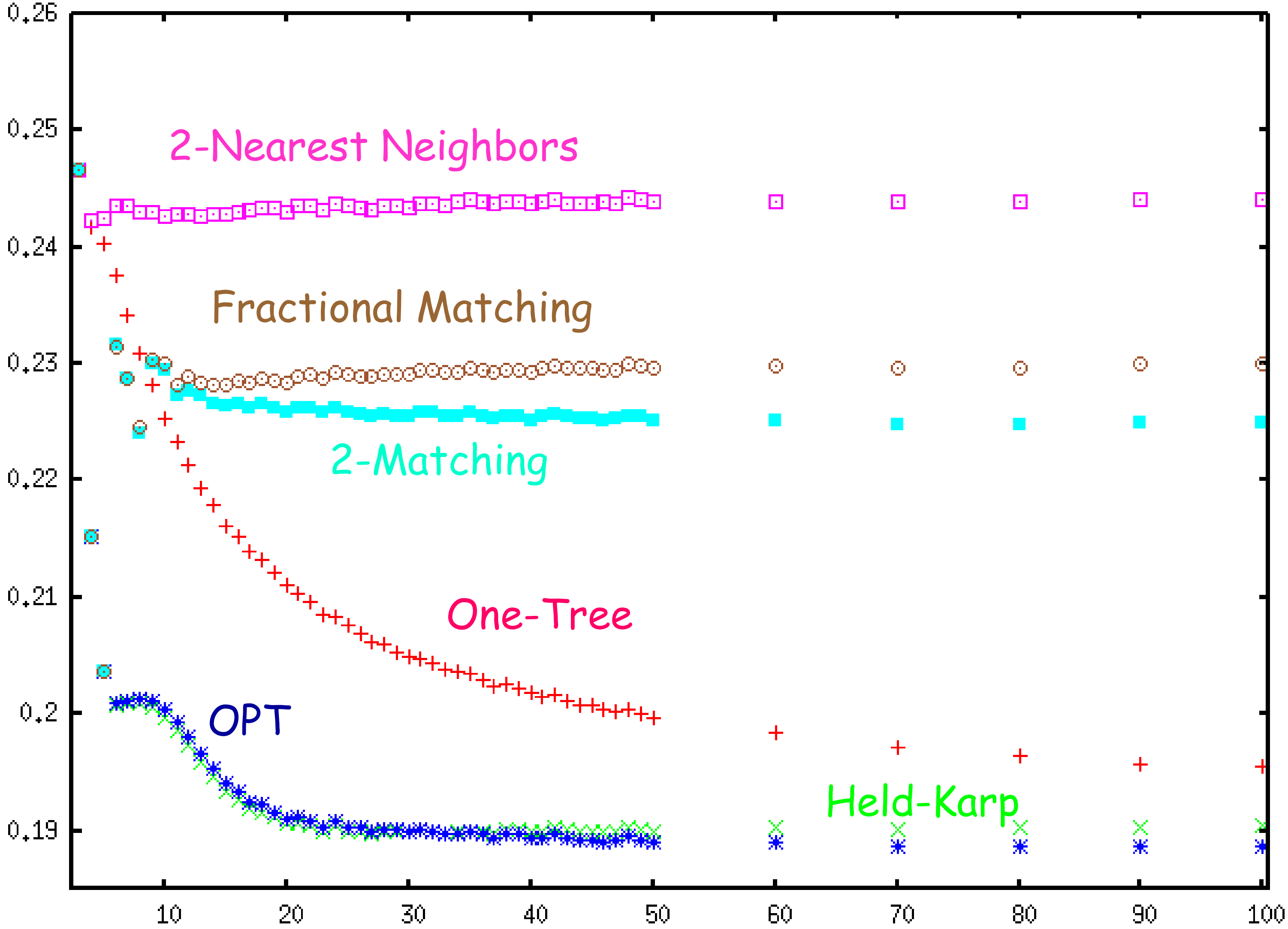
Standard Deviations for OPT, 3 ≤ N ≤ 100



Optimal versus Held-Karp



Standard Deviation Comparisons



Stop!