On Very Large Scale Assignment Problems

May 1993 WP# 3572-93

Yusin Lee*
James B. Orlin**

* GTE Laboratories
Waltham, MA 02154

** Sloan School of Management
Massachusetts Institute Technology
Cambridge, MA 02139

On Very Large Scale Assignment Problems

Yusin Lee
Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139 USA

James B. Orlin
Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA 02139 USA

Abstract

In this paper we present computational testing results on very large scale random assignment problems. We consider a fully dense assignment problem with $2n$ nodes. Some conjectured or derived properties regarding fully dense assignment problems including the convergence of the optimal objective function value and the portion of nodes assigned with their kth best arc have been verified for networks up to $n = 100,000$ in size. Also we demonstrate the power of our approach in solving very large scale assignment problems by solving a one million node, one trillion arc random assignment problem.

1 Introduction

In this paper we present computational testing results on very large scale random assignment problems. We consider a fully dense assignment problem with $2n$ nodes whose arc costs are identically independently distributed with density $f(c) = (r+1)c^r$. Avram and Bertsimas [3] showed that for these problems $\lim_{n\to\infty} \frac{E(P_n^*)}{n^{1-\frac{1}{r+1}}}$ converges to some value, where P_n^* is the objective function value for a $2n$-node random assignment problem. For the case $r = 0$, $f(c)$ is uniform (0,1). In this case $\lim_{n\to\infty} E(P_n^*)$ is shown to be bounded between 1.51 and 2, where the lower bound is due to B. Olin [17] and the upper bound is due to Karp [10].

Another interesting property of the random assignment problem is the fraction of nodes matched by the kth best arc in an optimum assignment. Dimitris Bertsimas [6]

conjectured that at optimum, one half of the nodes are matched by its best adjacent arc, half of the rest are matched by the second best adjacent arc, and so on. A similar observation is also made by B. Olin [17].

Only very limited computational testings have been done in the past to verify these properties. B. Olin [17] carried out some computational studies for networks of several hundred nodes. Pardalos and Ramakrishnan [18] solved for networks up to $n = 10,000$. Networks solved in both research are fully dense. In this research we used our algorithm to solve a random network of $n = 1,000,000$ in size. We then solved an extensive number of instances for networks up to $n = 100,000$ to verify the properties described above. We also observed the difference between the cost distribution of the arcs in the network and the arcs that are in an optimum solution.

In section 2 of the paper we introduce briefly the *QuickMatch* algorithm [13] that is used as the assignment problem solver in this research. Computational results are presented in section 3.

2 The Algorithm

2.1 The Assignment Problem

Consider a problem of matching n persons to n tasks where for each person i and for each task j there is an associated cost c_{ij} of assigning person i to task j. The assignment problem is the problem of matching n persons to n tasks so as to minimize the total cost. Although the assignment problem is traditionally phrased in terms of assigning persons to tasks, it also models applications in a wide range of different settings. For example, the assignment problem also has applications in vehicle routing and signal processing, and it is an important relaxation of the traveling salesman problem. For a survey of the applications, see Ahuja, Magnanti, and Orlin [1].

The standard integer programming version of the assignment problem is defined as follows:

$$\text{let } x_{ij} = \begin{cases} 1 & \text{if person } i \text{ is assigned to task } j \\ 0 & \text{otherwise.} \end{cases}$$

$$\textit{Minimize} \sum_{(i,j)\in A} c_{ij}x_{ij} \tag{1a}$$

$$\textit{Subject to} \sum_{i\in N} x_{ij} = 1 \qquad \forall j = 1,\ldots,n, \tag{1b}$$

$$\sum_{j\in N} x_{ij} = 1 \qquad \forall i = 1,\ldots,n, \tag{1c}$$

$$x_{ij} \geq 0, \ \textit{integral} \qquad \forall\, i,j. \tag{1d}$$

The assignment problem is a special case of the transportation problem, which in turn is a special case of the minimum cost flow problem. As is well known, all corner

task node k with $\mathcal{L}(k) < \mathcal{L}(w)$. For other nodes, $\pi(k)$ or $\gamma(k)$ is kept unchanged. A heuristic used in the QuickMatch algorithm is to run Dijkstra's algorithm and the Reverse Dijkstra's algorithm alternatively, i.e., initiate the Dijkstra's algorithm from a person node and a task node alternatively. A second heuristic is to set an upper bound T for the size of the shortest path trees. If a shortest path tree has already labeled T permanent nodes and still has not reached a target node, the tree is abandoned and a new tree is restarted from another node. As shown in the pseudo code, T is initially set to 2 and doubled at each outer iteration. More details of the QuickMatch algorithm can be found in Lee and Orlin [13].

A critical part of the algorithm for solving dense assignment problems above is to verify efficiently whether an optimum solution with respect to the first k arcs is also an optimum solution to the fully dense network. Let π, γ be an optimum set of dual prices for the assignment problem restricted to $G(k)$ and x^* be the corresponding optimum flow. It follows from the complementary slackness property of linear programming that if the reduced cost of an arc (i,j), defined as $\bar{c}_{ij} = c_{ij} - \pi_i + \gamma_j$, is greater than or equal to 0 for all $(i,j) \in G$, then x^* is also optimum for G. Let $A(k)$ be the set of the k lowest cost arcs, and c^* be the maximum arc cost in $A(k)$. If $c^* \geq \max(\pi_i - \gamma_j) \forall (i,j) \in A$, all arcs in $A \setminus A(k)$ will have positive reduced cost since these arcs all have cost greater or equal to c^*. Thus an optimum solution with respect to A_L is also an optimum solution with respect to A. Moreover, $\max(\pi_i - \gamma_j)$ can be calculated in $O(n)$ time. Therefore optimality can be checked efficiently. A pseudo code of the optimality checking process is shown below.

Subroutine Optimality Check
begin
  Let c^* be the maximum arc cost in $A(k)$;
  Let (π, γ) denote the optimum dual price determined by the algorithm for $G(k)$;
  $\pi_i^* = \max\{\pi_i | i \textit{ a person node}\}$;
  $\gamma_j^* = \min\{\gamma_j | j \textit{ a task node}\}$;
  if $(c^* > \pi_i^* - \gamma_j^*)$ **return** (π, γ is optimum)
  else return (π, γ is possibly not optimum)
end

3 Computational Analysis of the Assignment Problem

3.1 A Trillion Arc Assignment Problem Instance

As discussed in section 2.2, if one has the freedom to generate arcs in a suitable order, then one can often solve very large scale fully dense networks without generating most of the arcs. We demonstrate the power of this approach by solving a fully dense

network with one million nodes on each side, which has a trillion arcs in total. Using the algorithm described in section 2.2, we chose $2n \log n$ as the initial number of arcs, which is approximately 40,000,000. The algorithm terminated in one iteration. The 40×10^6 arc problem was solved in 28 minutes on a CRAY Y-MP M9 2/21000, using 1 CPU and 370×10^6 words memory. Notice that the number of arcs actually generated and solved is only a very small fraction of the entire set of one trillion arcs. If one wants to solve the fully dense network in its original form, it will take at least several thousand times more CPU time and memory space. In the remaining parts of this paper we will present computational results of up to $n = 100,000$ in size. Most of the instances are solved on a VAX-9000 machine. Networks of that size can be solved in a few minutes with the QuickMatch algorithm [13].

3.2 The Objective Function Value

Mezard and Parisi [16] predicted that the objective function value of a fully dense random assignment problem whose arc costs are uniformly distributed in $(0,1)$ converges to $\frac{\pi^2}{6} = 1.64493$ as the number of nodes n approaches infinity. B. Olin [17] solved networks of up to $n = 250$ in size and found P_n^*, the objective function value for instances of size n converging to 1.64. Pardalos and Ramakrishnan [18] studied networks of up to $n = 10,000$ in size and was unable to reject the null hypothesis that P_n^* convergs to $\frac{\pi^2}{6}$ as $n \to \infty$. In this section we check this conjecture by tracing the objective function value of fully dense assignment instances up to a size of 100,000 nodes on each side. A total of 26200 networks covering 8 network sizes are studied, with detailed testing for the cases $n = 2000, 4000$, 8000, 10,000 and 16,000. The data is shown in figure 2 and related statistics are shown in table 1. The columns $\overline{P_n^*}$, $\sigma_{P_n^*}$ and $\sigma_{\overline{P_n^*}}$ are average P_n^*, the estimated variance for P_n^*, and the variance for the estimated mean of P_n^*, respectively. The last column shows how far the observed $\overline{P_n^*}$ deviates from $\frac{\pi^2}{6}$. For networks sizes that we solved 5000 instances, the observed average value for P_n^* is about 1 to 3 standard deviations away from $\frac{\pi^2}{6}$. Assuming normal distribution for the observed P_n^*, 3 standard deviations would correspond to a confidence level of 99.7%, 2 standard deviations corresponds to 95.5%, and 1 standard deviation corresponds to 68.3% (two-tail). We also observe the trend that the difference decreases as n grows large. The trend that the objective function value approaches $\frac{\pi^2}{6}$ from below agrees with the observation by B. Olin [17] until n=10,000. Then it seems to bounce around for larger sizes. However, we point out that while the deviation of $\overline{P_n^*}$ from $\frac{\pi^2}{6}$ (in standard deviations) seems to remain close to ± 1 for $n \geq 16000$, one should notice that the data size is very different for $n \geq 20000$ networks.

B. Olin [17] found that the objective function value is normally distributed for instances of $n = 250$. We verified the distribution for up to $n = 16,000$ instances. The quantile-quantile plot for the observed P_n^* of n=2000, 4000, 8000, and 16000 networks are plotted in figure 3. In all cases all data points lie reasonably close to

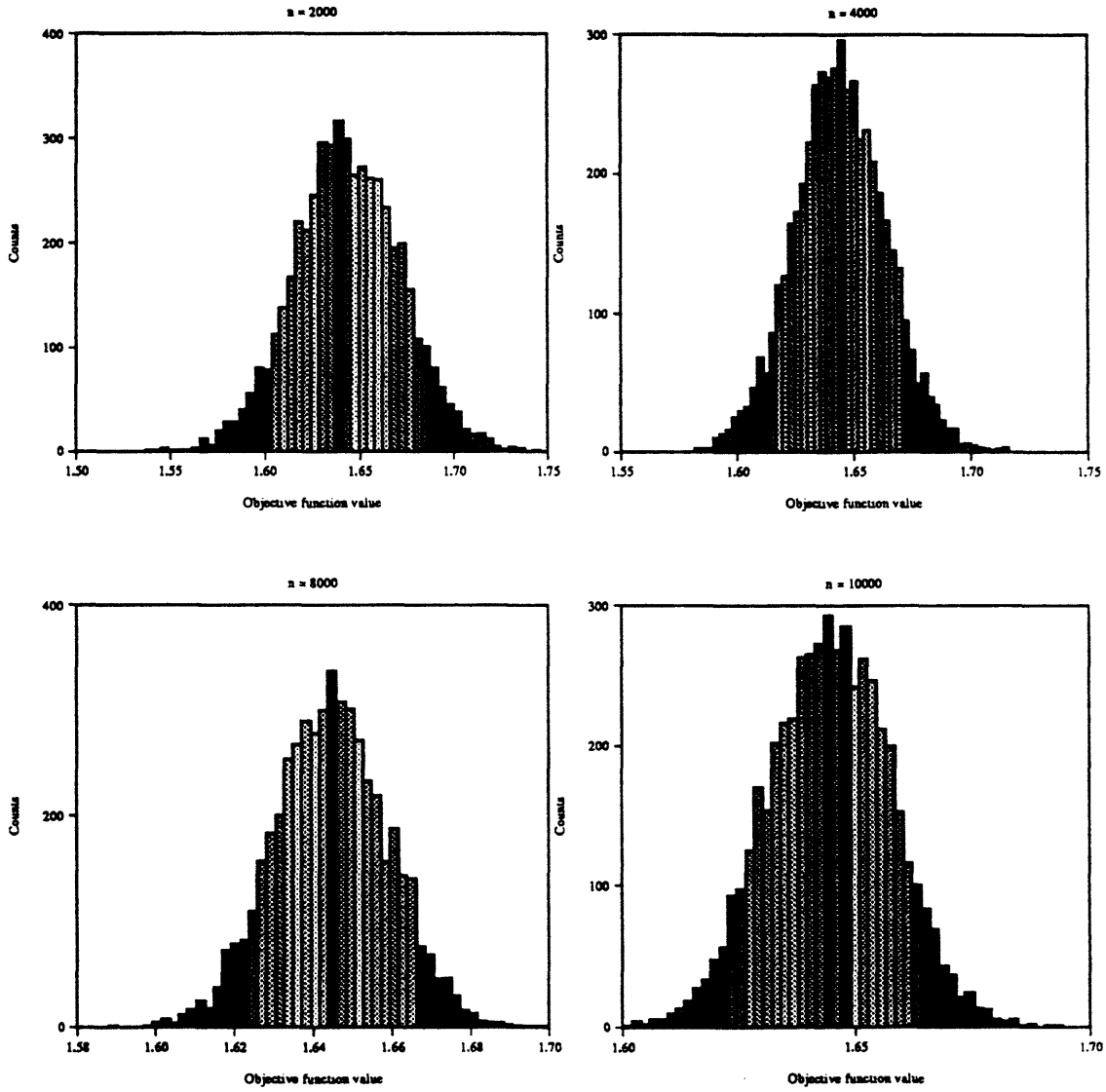


Figure 2: Objective function values.

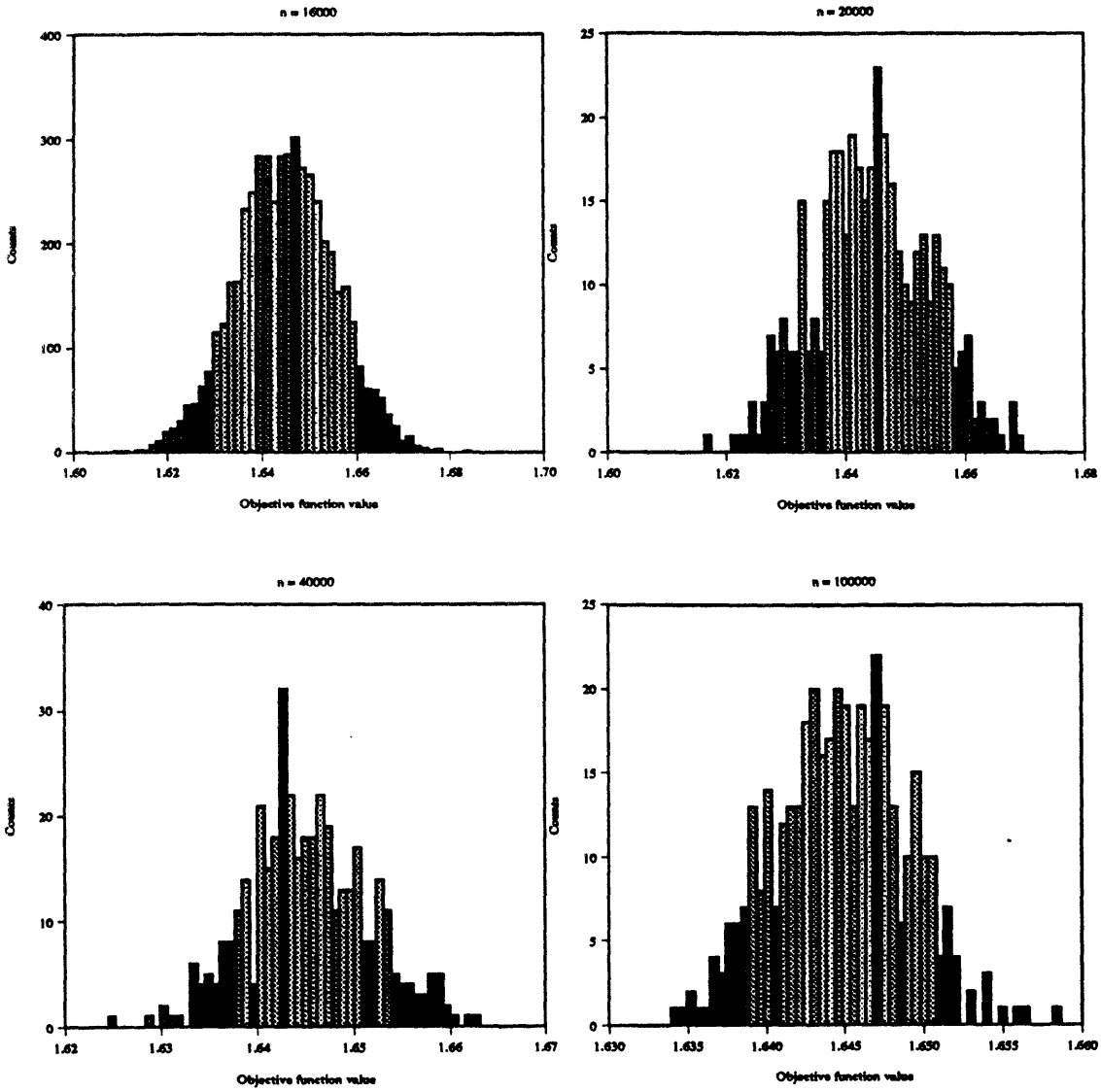


Figure 2: (continued)Objective function values.

observations	n	$\overline{P^*_n}$	$\sigma_{P^*_n}$	$\sigma_{\overline{P}^*_n}$	$(\overline{P^*_n} - \frac{\pi^2}{6})/\sigma_{\overline{P}^*_n}$
5000	2,000	1.64371	0.02782	0.0003935	-3.11
5000	4,000	1.64413	0.01949	0.0002756	-2.91
5000	8,000	1.64451	0.01429	0.0002021	-2.10
5000	10,000	1.64456	0.01298	0.0001836	-1.99
5000	16,000	1.64508	0.01020	0.0001442	1.04
400	20,000	1.64440	0.00965	0.000482	-1.12
400	40,000	1.64530	0.00629	0.000314	1.16
400	100,000	1.64474	0.00410	0.000205	-0.97

Table 1: Statistics of objective function values. All networks are fully dense.

the 45 degree line, indicating that the data is distributed very close to normal.

3.3 Arc Preference of Optimum Solutions

Consider a fully dense assignment problem and a corresponding optimum solution. For each arc $(i,j) \in A$, we say that $(i,j) \in A^k$ if (i,j) is the k-th least cost arc incident to i or if (i,j) is the k-th least cost arc incident to j or both. Let A_{opt} be the set of arcs used in an optimum solution. Also let A^k_{opt} be the intersection of A^k and A_{opt}. In other words, A^k_{opt} is the set of arcs that are the kth best adjacent arc of one of its end nodes and also in the optimum solution. In this section we study the number of arcs in A^k_{opt} for each k. The following is a conjecture by Dimitris Bertsimas [6] and B. Olin [17].

Conjecture 1 $\lim_{n\to\infty} \frac{|A^k_{opt}|}{n} = \frac{1}{2^k}$.

Let i be a node and let $mate(i)$ be the node assigned to i in A^*. Also let κ_i be the number of arcs adjacent to node i that has cost less than $c_{i,mate(i)}$. In other words, $A^k_{opt} = \{(i,j) \in A_{opt}, \kappa_i = k-1 \text{ or } \kappa_j = k-1\}$. We verify conjecture 1 by observing κ_i for the nodes in the network. Figure 4 shows the plot of observed κ_i, where the horizontal axis are κ_i and the vertical axis are observation counts, plotted in log scale. The straight line in each of the plots corresponds to the κ_i predicted by conjecture 1. One can see visually that the data supports the conjecture for the networks studied. We divide the variance of the observed data by the variance of its predicted value by conjecture 1. This is an analog of the standard way of calculating R^2 as a measure of the goodness of fit for regression models. However, the ratio in this case will not be restricted to be less than or equal to 1. The calculated ratio for n = 2,000, 4,000, 8,000, 10,000, 16,000, 20,000, 40,000, and 100,000 are 1.022, 0.996, 0.984, 1.004, 0.996, 0.999, 1.000, 0.993, respectively. All these ratios are very close to 1.0, which indicates that the observed data is very close to what conjecture 1

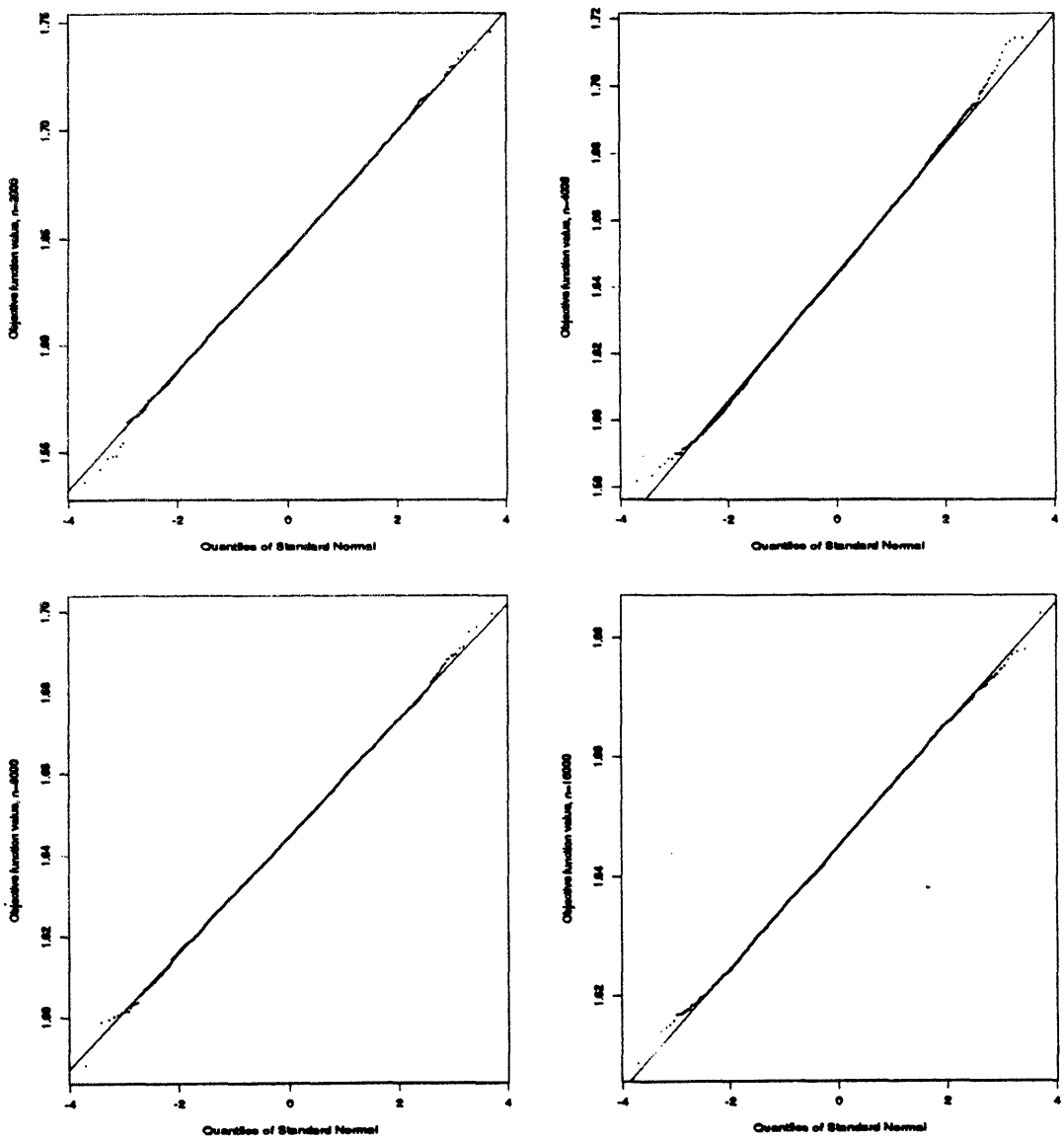


Figure 3: Quantile-quantile plot for observed objective function values.

predicts. We also calculate the correlation between observed and predicted data. In all cases the coefficient of correlation is 1.000.

Next we study $\min(\kappa_i,\ \kappa_j)$ for each arc $(i,j) \in A_{opt}$. In almost all cases the maximum of $\min(\kappa_i,\ \kappa_j)$ is less than 10. This suggests that for random networks up to $n = 100,000$ in size, in most cases the union of the 10 least cost arcs adjacent to each node is sufficient to yield an optimum solution to the original fully dense network. The size of this union will be no more than $20n$. For $n = 100,000$, $20n < 2n \log n \approx 33n$. This seems reasonable since one will need more than $20n$ arcs to cover the 10 least cost arcs for each node. The observed data is plotted in figure 5. We observe that the graphs look similar to their counterparts in figure 4 except for a different slope. Let $\tilde{A}^k_{opt} = \{(i,j) \in A : \min(\kappa_i,\ \kappa_j) = k\}$. Table 2 shows the regression results on $\min(\kappa_i,\ \kappa_j)$ and $\frac{|\tilde{A}^k_{opt}|}{n}$. The straight line in each of the plots in figure 5 is the corresponding fitted line. Based on our observed data and the regression result, we have the following conjecture.

Conjecture 2 $\lim_{n\to\infty} \frac{|\tilde{A}^k_{opt}|}{n} = \beta^k$, *where* $0.26 < \beta < 0.30$.

3.4 Cost Distribution of the Arcs Used in an Optimum Solution

The cost distribution of the arcs in A^k_{opt} is different from that of the arcs in A^k. If conjecture 1 is true and if the two distributions were the same, then the objective function value would converge to 2 as n becomes large (which is not the case), since the expected cost of arcs in A^k is $\frac{k}{n+1}$. We verify this by plotting the arc-cost percentile graph as shown in figure 6. In the graphs the horizontal axis is the arc cost, the vertical axis is the percentage of arcs in A^k (solid line) or A^k_{opt} (dotted line) whose cost is less than some cost. Figures 6a, 6b, 6c, and 6d show the data for $k = 1$, 2, 3, and 4, respectively. The curves in figures 6b, 6c, and 6d clearly shows that cost of arcs in A^k and A^k_{opt} are of different distributions for $k = 2, 3$, and 4, while A^k_{opt} includes more low cost arcs in A^k. However, the two curves in figure 6a almost overlap each other. This leads to the following conjecture.

Conjecture 3 *In an assignment problem, the distribution of the cost of arcs in* A^1 *is the same as the arcs in* A^1_{opt}.

We provide a heuristic (non-rigorous) argument as to why the conjecture might be valid. We conjecture that this heuristic argument can be made approximately rigorous, but we are unaware of such a proof. Let $\hat{c}_{ij} = c_{ij} - \min(c_{ij} : j = 1, \ldots, n)$. Then $\hat{c}_{ij} = 0$ for $(i,j) \in A^1$, but $\hat{c}_{ij}$ is asymptotically uniformly distributed (0,1) for $(i,j) \notin A^1$. Also, an optimum solution with respect to $\hat{c}$ is optimum with respect to c. In optimizing with respect to $\hat{c}$, there will be no particular pattern in the arc costs

n	regression	R^2
2000	$y = 0.267^x$	0.991
	(-48)	
4000	$y = 0.252^x$	0.996
	(-74)	
8000	$y = 0.272^x$	0.990
	(-46)	
10000	$y = 0.265^x$	0.994
	(-61)	
16000	$y = 0.259^x$	0.996
	(-74)	
20000	$y = 0.278^x$	0.993
	(-58)	
40000	$y = 0.265^x$	0.988
	(-49)	
100000	$y = 0.264^x$	0.996
	(-86)	

Table 2: Regression results for various network sizes, where x is $\min(\kappa_i, \kappa_j)$ and y is $\frac{|A^k_{opt}|}{n}$. Numbers in parenthesis are the t values corresponding to the logarithm (base 2) of the corresponding data. Data for iterations 7 and 8 for $n = 4000$ network, and also iteration 7 for $n = 2000$ network are dropped from the regression.

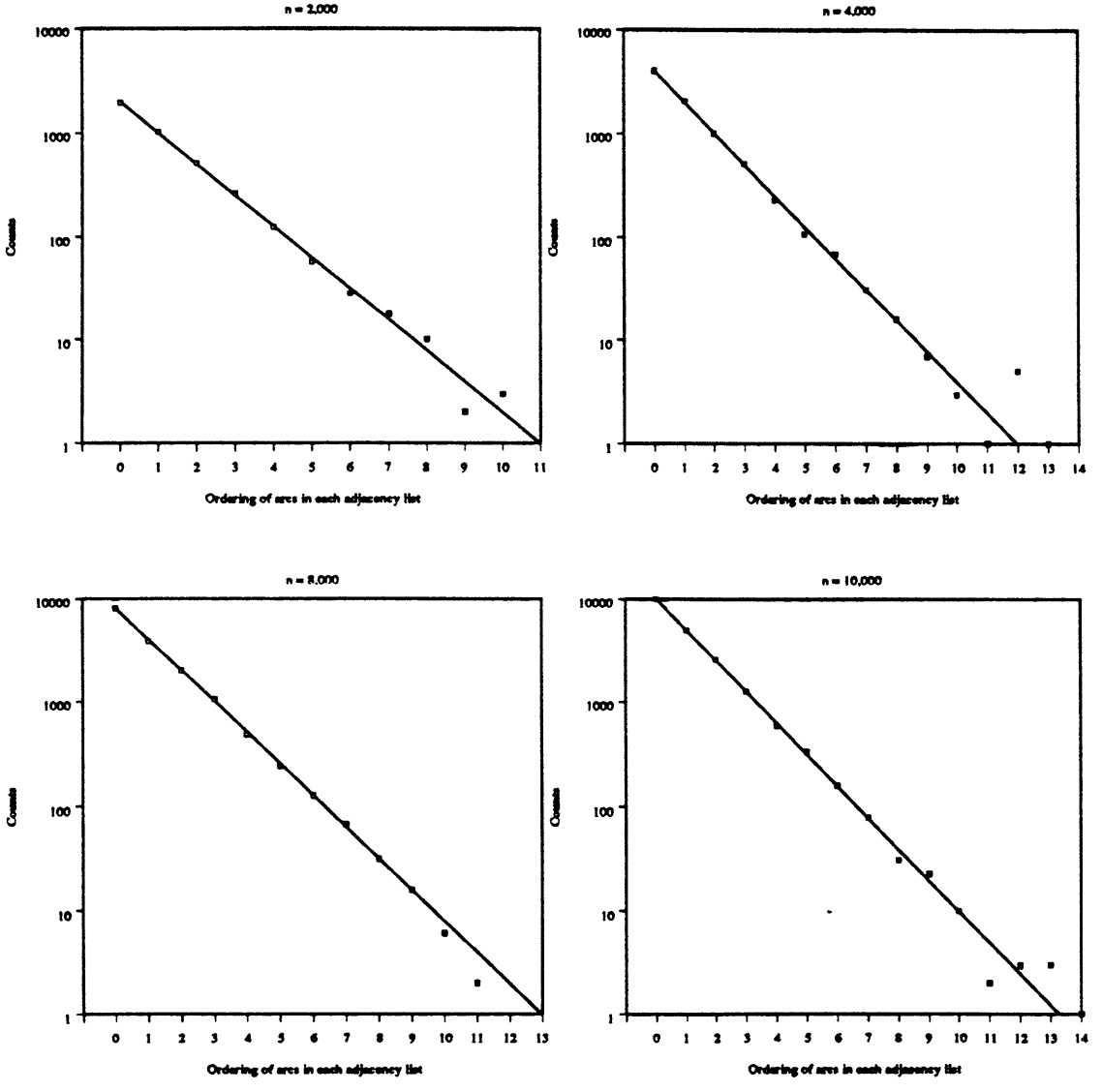


Figure 4: Observed κ.

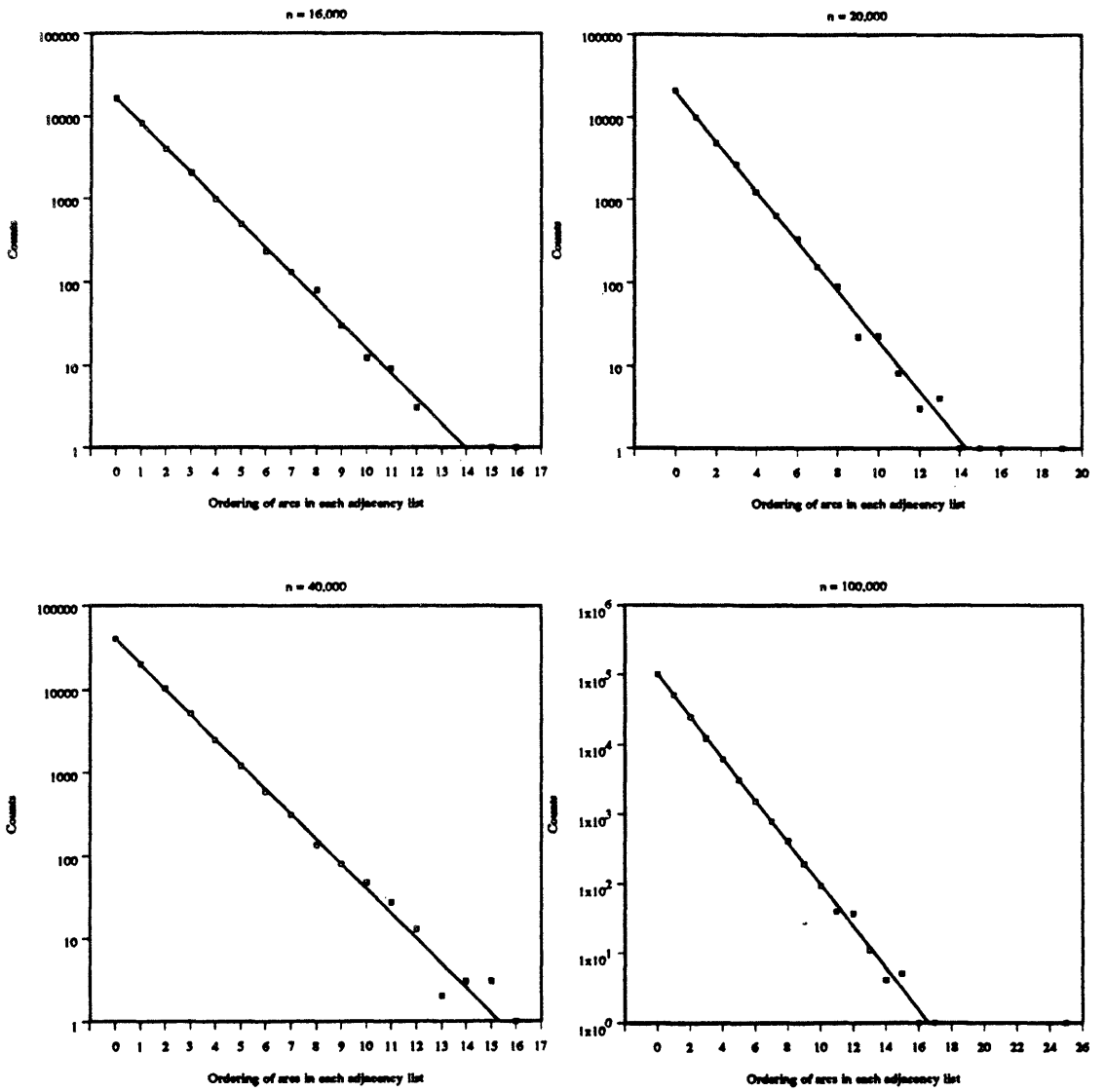


Figure 4: (continued)Observed κ.

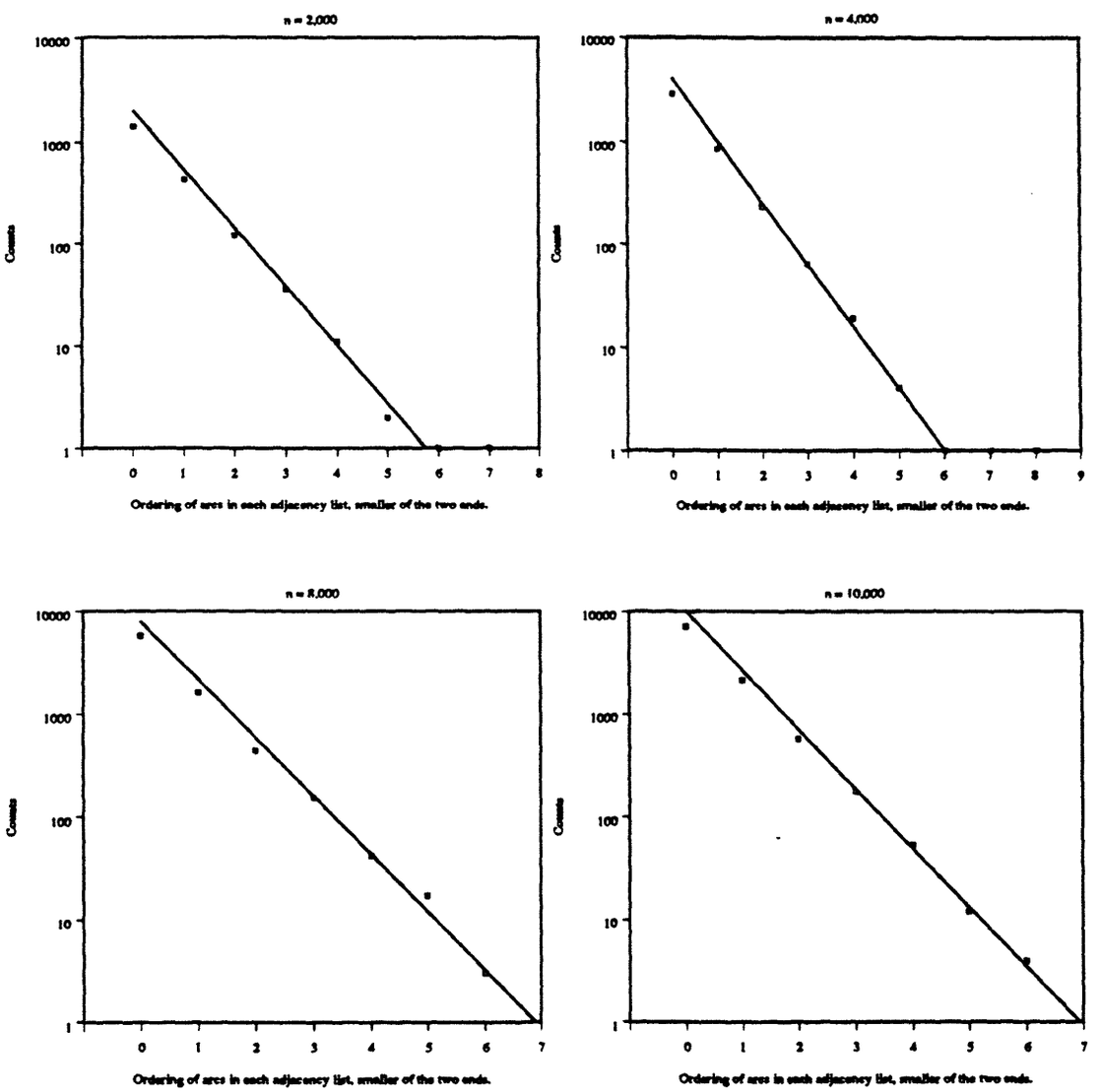


Figure 5: $\min(\kappa_i,\ \kappa_j)$ for arcs (i, j) used in an optimum solution.

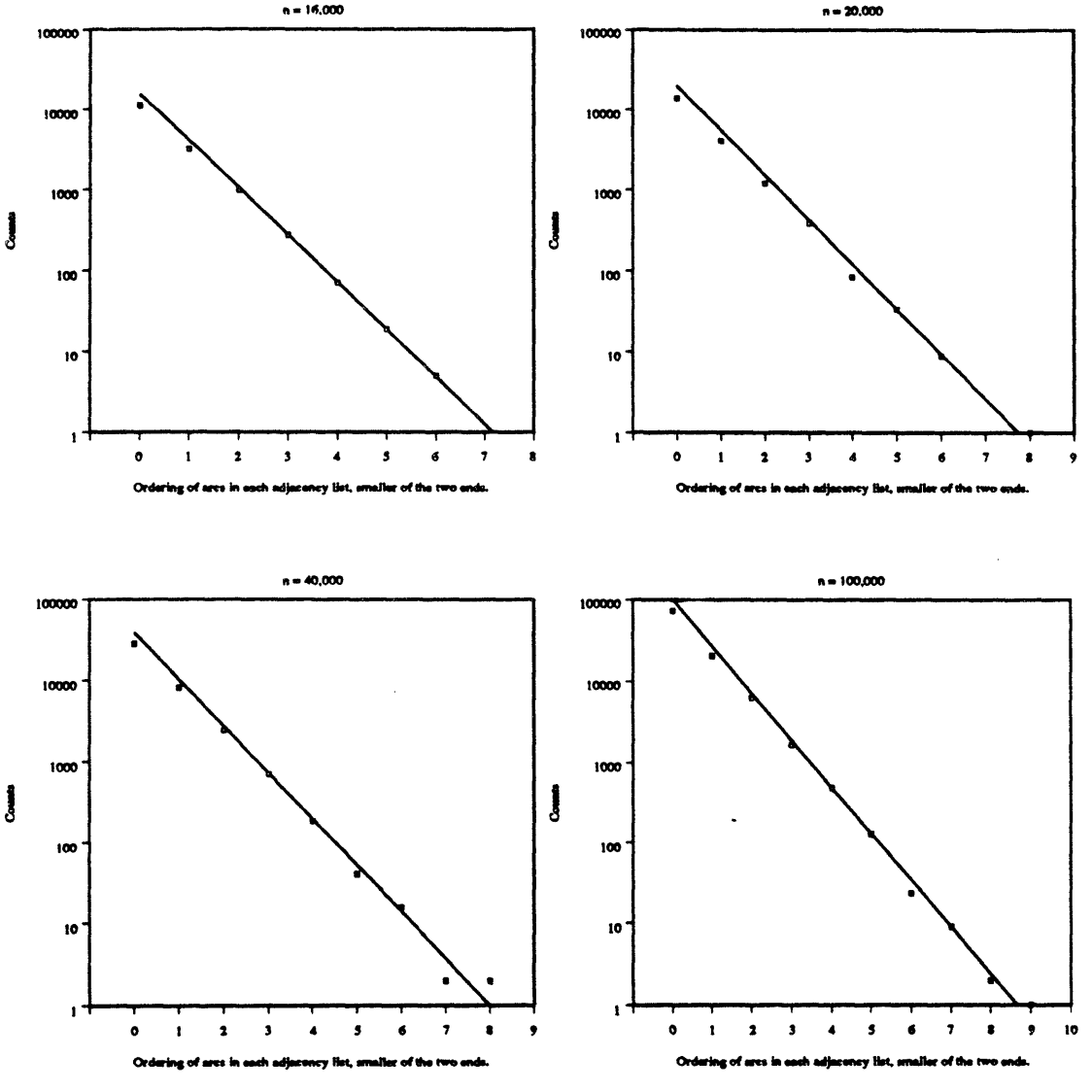


Figure 5: (continued)$\min(\kappa_i,\ \kappa_j)$ for arcs (i, j) used in an optimum solution.

of A^1 and so the costs for arcs in A^1_{opt} should have the same distribution as the costs for arcs in A^1. However, one would expect that the costs of arcs in A^2_{opt} would be less than the arc costs in A^2.

Figure 7 shows the cost distribution of arcs used in an optimum solution for an $n = 2000$ network. Figure 7a, 7b, 7c, and 7d corresponds to the arcs in A^1_{opt}, A^2_{opt}, A^3_{opt}, and A^4_{opt}, respectively. Figure 8 shows the distribution of the arc costs in A^1, A^2, A^3, and A^4 of the same network. Figure 9 shows the cost distribution of all arcs in A^* of various networks.

3.5 Generalization of the Arc Cost Distribution

In previous sections we focus on random networks whose arc costs are distributed with density $f(c) = (r+1)c^r$ for the case $r = 0$. In this section we focus on the case $r \neq 0$. Avram and Bertsimas [3] showed that for these problems $\lim_{n\to\infty} \frac{E(P_n^*)}{n^{1-\frac{1}{r+1}}}$ converges to some value. Unfortunately they are unable to solve for the value it converges to. In this section we try to verify the convergence prediction and achieve a first estimate on what the limit might be. Table 3 shows the statistics for fully dense networks whose sizes range from $n = 2000$ to $n = 8000$ and $r = 2, 3, 5$, and 10. The data does agree with the prediction. Also we observe that the objective function values are very different from the case $r = 0$.

We also observed that for the $r > 0$ case, the number of arcs k needed such that an optimum solution to $G(k)$ is also an optimum solution to G is much higher than that of the $r = 0$ case. We show this by observing $\min(\kappa_i, \kappa_j)$. Figure 10 shows $\min(\kappa_i, \kappa_j)$ for $n = 20,000$ networks, where $r = 2$, 3, 5, and 10 in figures 10a, 10b, 10c, and 10d, respectively. One can see the difference by comparing figure 10 to figure 5.

Acknowledgments

The authors would like to thank Cray Research, Inc. and to John Gregory in particular for providing us access to the Cray Y-MP, on which we solved the trillion arc problem. We would also like to thank Dimitris Bertsimas and Rakesh Vohra for their suggestions concerning a draft of the manuscript, for their conjectures on large scale assignment problems, and for their pointers to the literature. This research is supported in part by grant AFOSR-88-0088 from the Air Force Office of Scientific Research, and by a grant from the United Parcel Service.

References

[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network flows: Theory, Algorithms and Applications.* Prentice Hall, 1993.

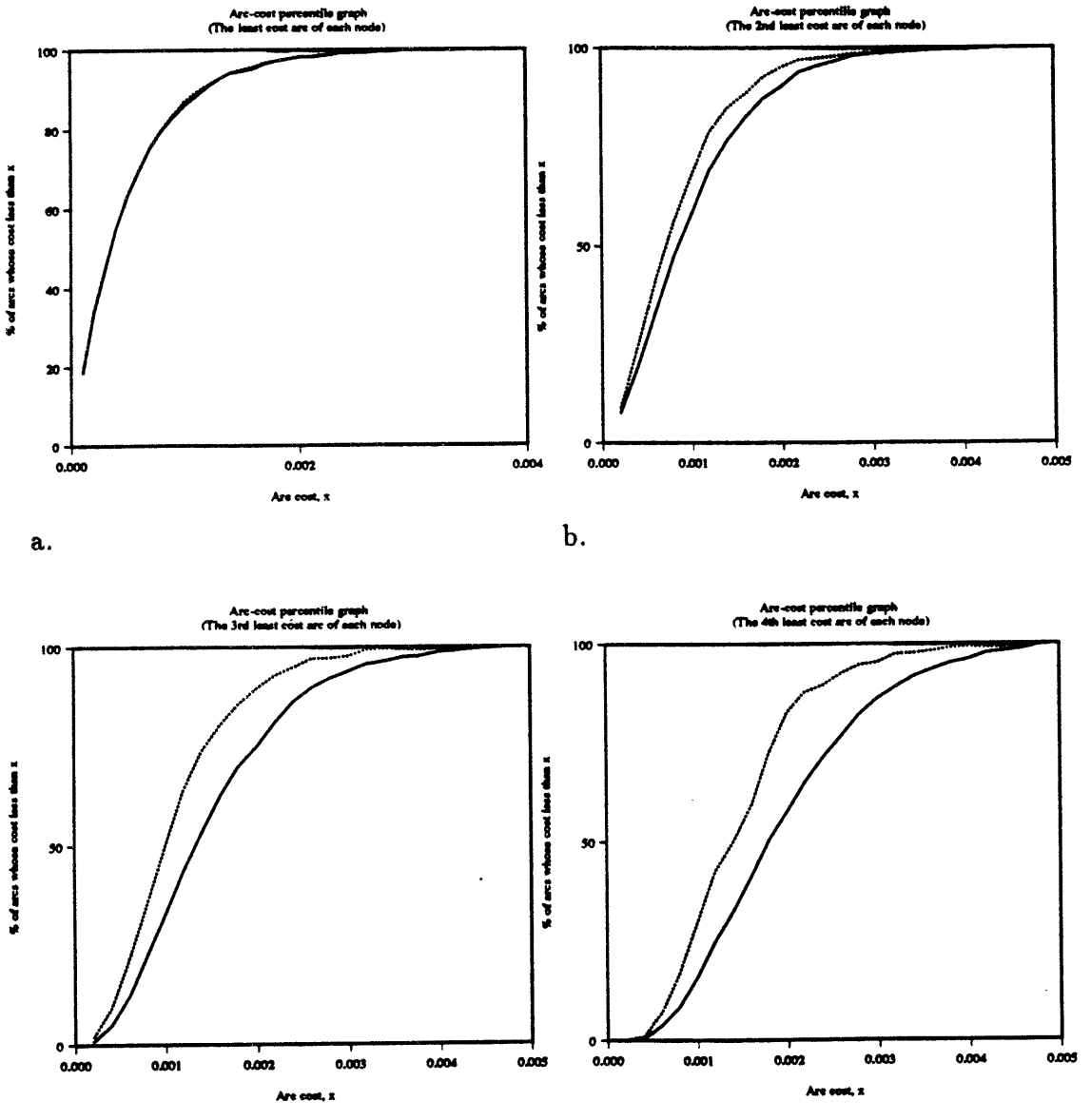
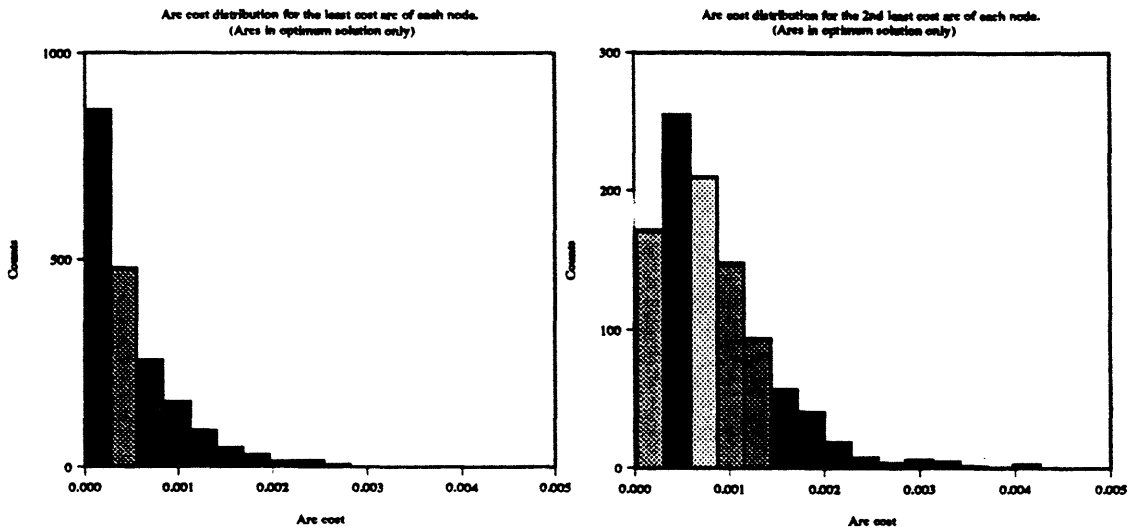
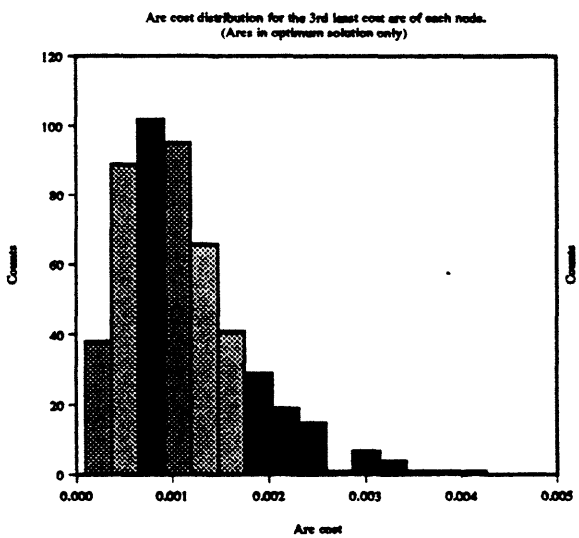


Figure 6: The arc-cost percentile graphs. The horizontal axis is the arc cost, the vertical axis is the percentage of arcs in A^k (solid line) or A^k_{opt} (dotted line) whose cost is less than some cost.

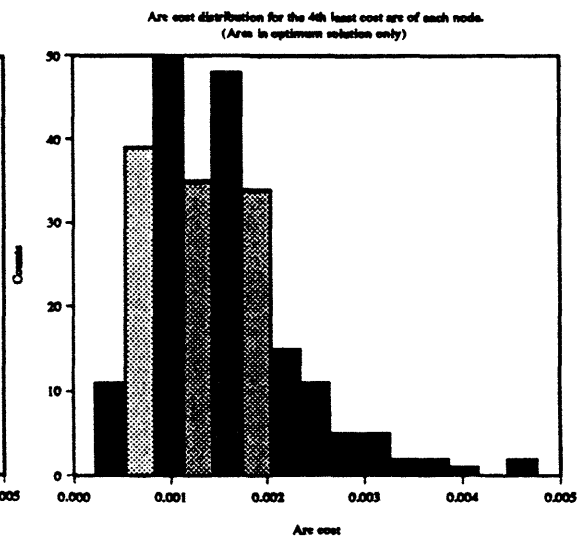


a.

b.

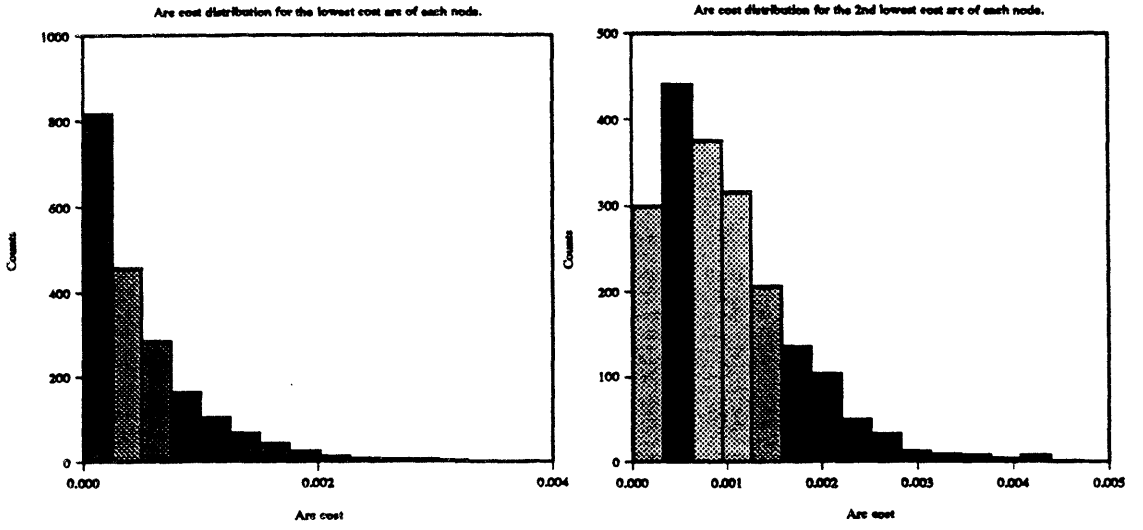


c.

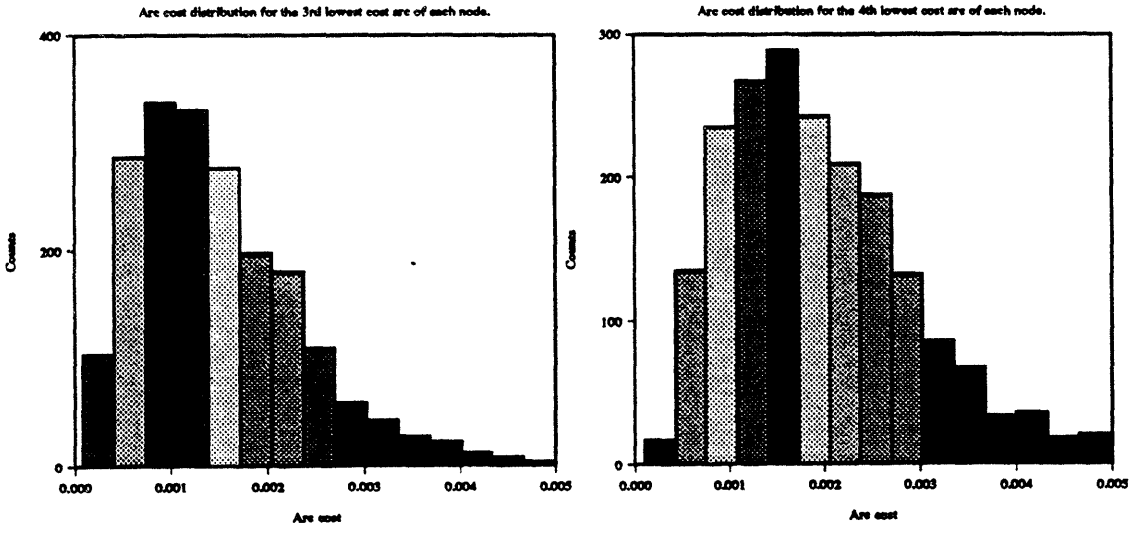


d.

Figure 7: Cost of arcs in A^1_{opt}, A^2_{opt}, A^3_{opt}, and A^4_{opt} of an $n = 2000$ network.



a. b.



c. d.

Figure 8: Cost of arcs used in A^1, A^2, A^3, and A^4 of an $n = 2000$ network.

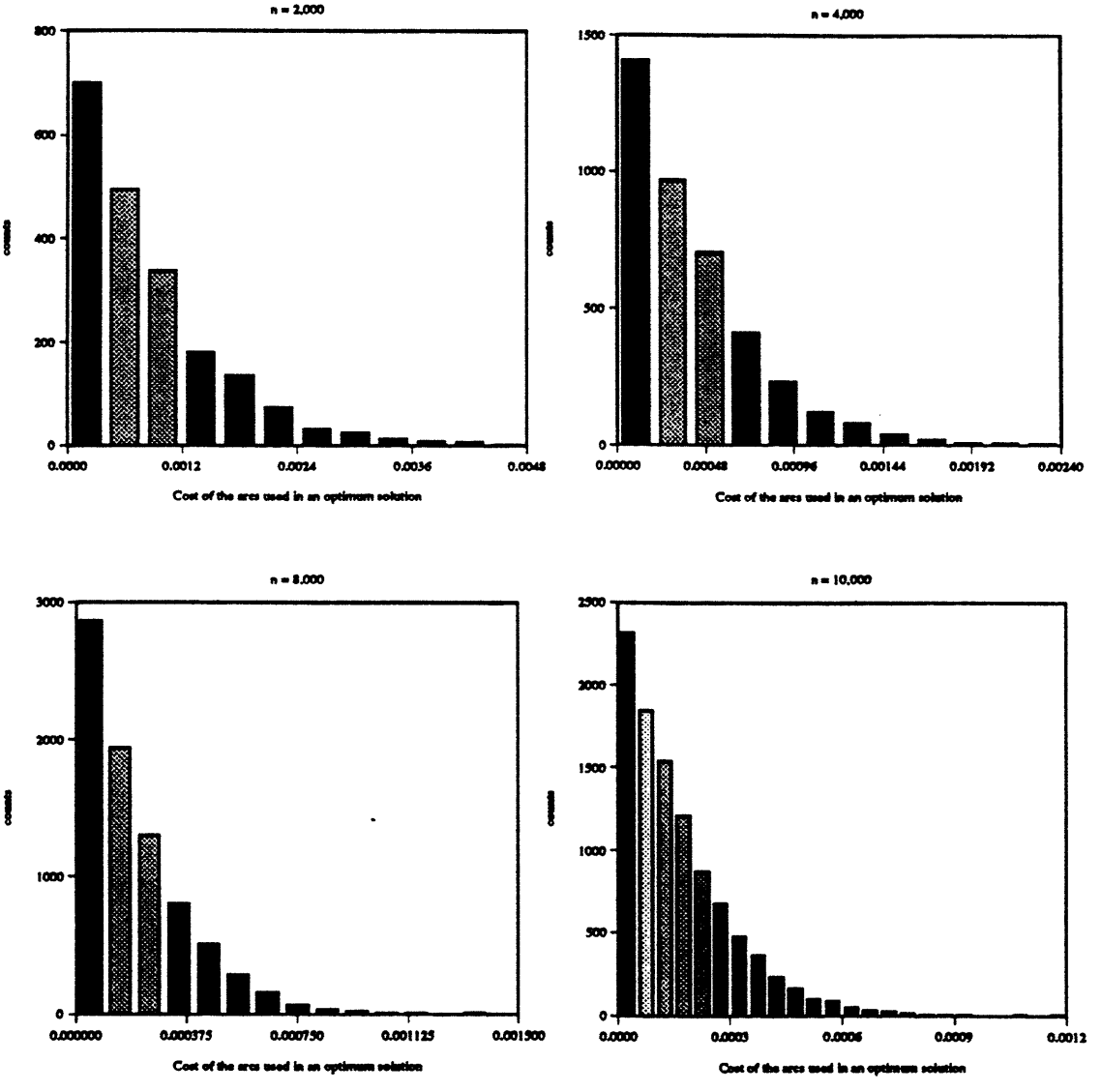


Figure 9: Cost of arcs used in optimum solutions.

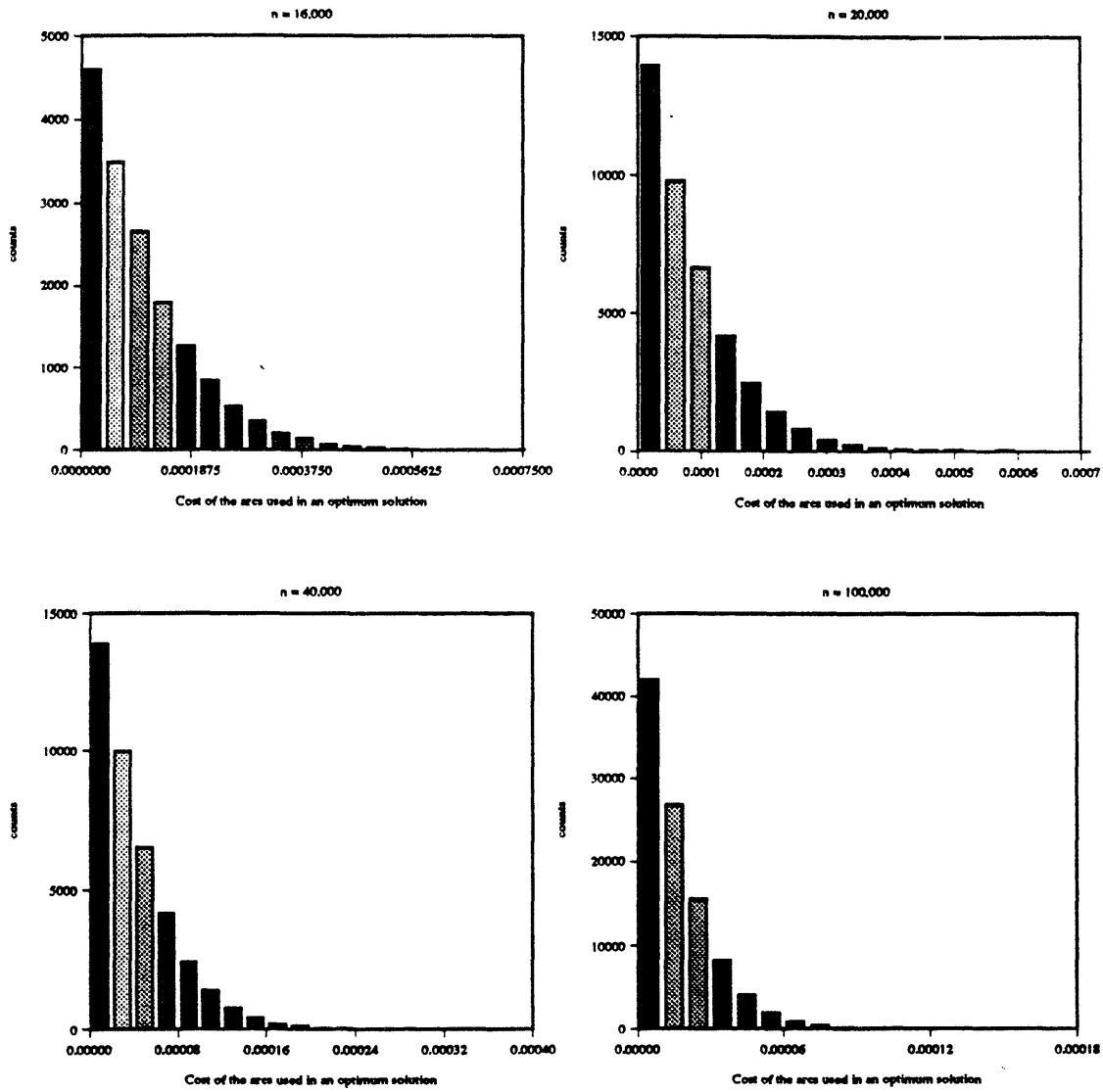


Figure 9: (continued)Cost of arcs used in an optimum solution.

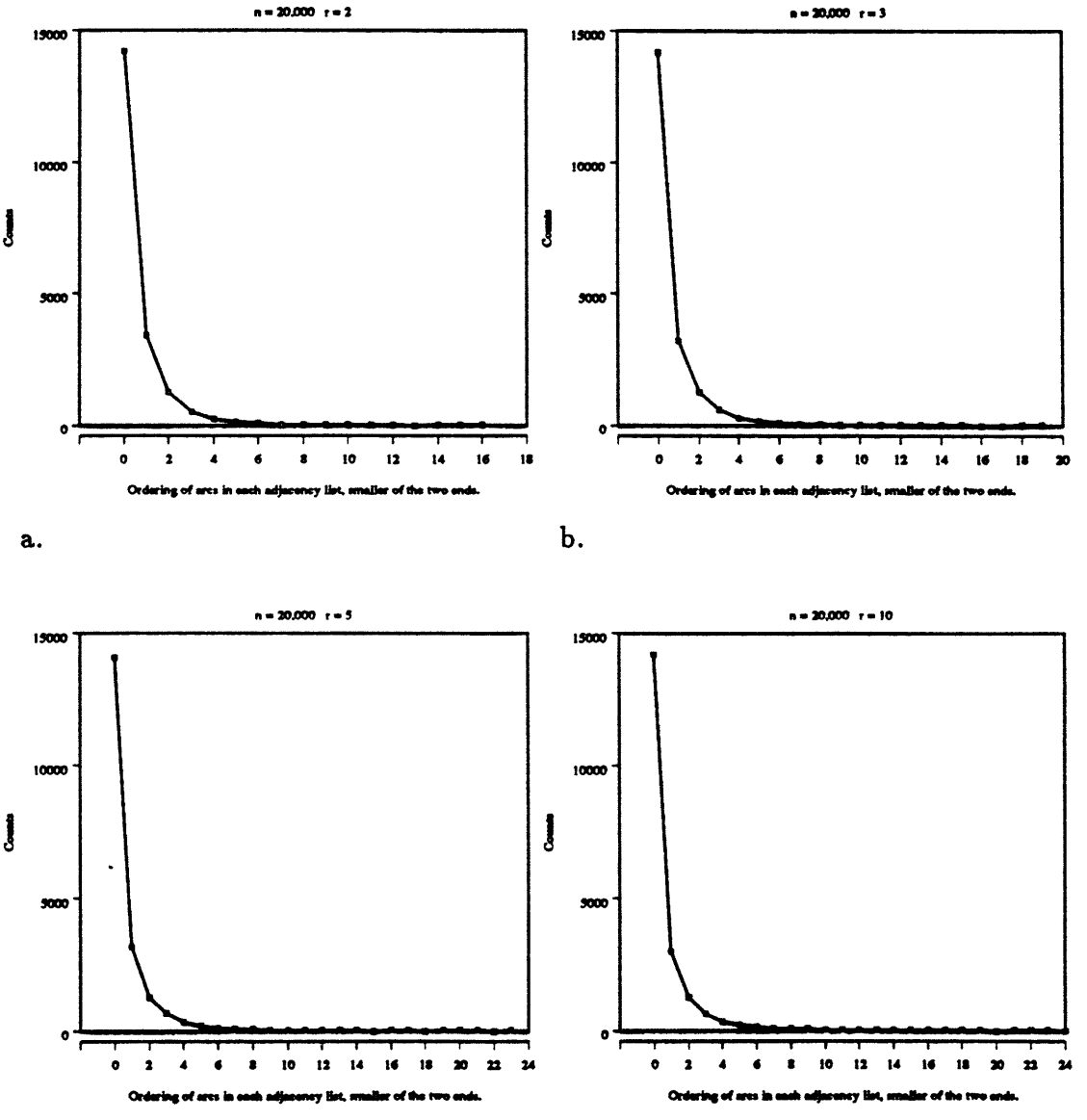


Figure 10: $\min(\kappa_i,\ \kappa_j)$ for arcs (i,j) used in an optimum solution. Network size is $n = 20,000$. Arc cost distribution is $r = 2, 3, 5$, and 10.

r	n	$\overline{P}_n^*$	$\sigma_{P_n^*}$	$\sigma_{\overline{P}_n^*}$	$\overline{P}_n^*/n^{1-\frac{1}{r+1}}$
2	2000	166.596	0.893	0.200	1.0495
	8000	421.770	1.361	0.304	1.0544
	20000	776.065	1.835	0.410	1.0533
3	2000	306.165	1.197	0.268	1.0237
	8000	865.579	2.616	0.585	1.0233
	20000	1721.570	2.577	0.576	1.0237
5	2000	565.898	2.410	0.539	1.0043
	8000	1792.618	2.433	0.544	1.0021
	20000	3850.566	4.571	1.022	1.0031
10	2000	997.464	2.237	0.500	0.9953
	8000	3516.254	3.160	0.707	0.9950
	20000	8088.215	4.551	1.018	0.9950

Table 3: Statistics of objective function values for $r \neq 0$. All networks are fully dense. Number of observations are 20 for all cases.

[2] Mustafa Akgul. A forest primal-dual algorithm for the assignment problem. *Bilkent University, Ankara, Turkey, Research Report: IEOR-9014*, 0(0):1–2, Oct 1990.

[3] Florin Avram and Dimitris Bertsimas. On a characterization of the minimum assignment and matching in the independent random model. In *The third symposium in integer programming and combinatorial optimization, Enrice, Italy*, April 1993.

[4] M. L. Balinski. Signature methods for the assignment problem. *Operations Research*, 33(3):527–536, May–Jun 1985.

[5] Dimitri P. Bertsekas. The auction algorithm for assignment and other network flow problems: A tutorial. *Interfaces*, 20(4):133–149, Jul–Aug 1990.

[6] Dimitris J Bertsimas, 1993. Personal communication.

[7] Harold N. Gabow and Robert E. Tarjan. Faster scaling algorithms for network problems. *SIAM Journal on Computing*, 18(5):1013–1036, Oct 1989.

[8] Ming S. Hung. A polynomial simplex method for the assignment problem. *Operations Research*, 31(3):595–600, May-Jun 1983.

[9] Ming S. Hung and Walter O. Rom. Solving the assignment problem by relaxation. *Operations Research*, 28(4):969–982, Jul–Aug 1980.

[10] Richard M. Karp. An upper bound on the expected cost of an optimal assignment. Technical report, Computer Science Division, University of California, Berkeley.

[11] Richard M. Karp. An algorithm to solve the mxn assignment problem in expected time O(mn log n). *Networks*, 10:143–152, 1980.

[12] J. Kennington and Z. Wang. A shortest augmenting path algorithm for the semi-assignment problem. *Operations Research*, 40(1):178–187, Jan–Feb 1992.

[13] Yusin Lee and James B. Orlin. QuickMatch: A very fast algorithm for the assignment problem. Submitted to Mathematical Programming., 1993.

[14] Vahid Lotfi. A labeling algorithm to solve the assignment problem. *Computers and Operations Research*, 16(5):397–408, 1989.

[15] G. M. Megson and D. J. Evans. A systolic array solution for the assignment problem. *The computer journal*, 33(6):562–569, 1990.

[16] M. Mezard and G. Parisi. Replicas and optimization. *Journal de physique Lettres.*

[17] Birgitta Olin. *Asymptotic Properties of Random Assignment Problems.* PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 1992.

[18] P. M. Pardalos and K. G. Ramakrishnan. On the expected optimal value of random assignment problems: Experimental results and open questions.