

# A new probabilistic analysis of Karger's randomized algorithm for minimum cut problems

Yang Dai<sup>a,1</sup>, Kazuo Iwano<sup>b,2</sup>, Naoki Katoh<sup>c,\*</sup>

<sup>a</sup> Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, 2-12-1 O-okayama, Meguro-ku, Tokyo 152, Japan

<sup>b</sup> Tokyo Research Laboratory, IBM Japan, 1623-14, Shimotsuruma, Yamato-shi, Kanagawa 242, Japan

<sup>c</sup> Department of Architecture and Architectural Systems, Kyoto University, Sakyo-ku, Kyoto 606-01, Japan

Received 8 July 1995; revised 3 June 1997

Communicated by K. Ikeda

---

## Abstract

Recently Karger proposed a new randomized algorithm for finding a minimum cut of an  $n$ -vertex graph (weighted or unweighted) with probability  $\Omega(n^{-2})$ . In this paper we present a new probabilistic analysis of Karger's randomized algorithm for a few classes of unweighted graphs. For random graphs whose edges are selected with a given probability  $p$ , ( $\log n/n \leq p \leq 1$ ), we show that the expectation of success probability of the algorithm is  $\Omega(p/n)$ . We also investigate a class of graphs with special structure that consists of two  $n$ -cliques and  $\gamma(n-1)$  edges between the two cliques. Here  $\gamma$  is a parameter satisfying  $0 < \gamma < 1$  that makes these  $\gamma(n-1)$  edges a unique minimum cut. We show that the algorithm finds the unique minimum cut with probability  $\Omega(\gamma/n^\gamma)$ . © 1997 Elsevier Science B.V.

*Keywords:* Randomized algorithms; Probabilistic analysis; Minimum cut; Minimum range cut

---

## 1. Introduction

Given a connected undirected simple graph  $G = (V, E)$  with  $n$  vertices and  $m$  unweighted edges, the minimum cut problem is to find a set of edges of minimum cardinality whose removal leaves two connected components. The cardinality of edges in the minimum cut is also called the edge connectivity of the graph. The minimum cut problem has numerous applications in many fields and is extensively studied. The current fastest deterministic algorithm for undirected, unweighted graphs requires  $O(m\lambda(G) \log(n^2/m))$  time [4,5], where  $\lambda(G)$  is the edge connectivity of  $G$ .

Recently, Karger [6] proposed a new randomized algorithm (called *contraction algorithm* below) for computing minimum cuts based on a simple operation of edge contraction. When the contraction algorithm terminates, a minimum cut is produced with probability  $\Omega(n^{-2})$ . So, if  $O(n^2 \log n)$  independent trials are performed, one

---

\* Corresponding author. Email: naoki@archi.kyoto-u.ac.jp.

<sup>1</sup> Email: dai@is.titech.ac.jp.

<sup>2</sup> Email: iwano@jp.ibm.com.

finds a minimum cut with high probability. It should be noted that the contraction algorithm also works for weighted graphs, where the total weights of edges in a cut is minimized. However, we restrict our analysis to the unweighted graphs throughout the paper. From our preliminary computational results the contraction algorithm produces minimum cuts with probability much higher than  $\Omega(n^{-2})$  for most of the randomly generated input instances. This implies a gap between theoretical and computational results for random graphs. We shall establish a lower bound stronger than  $\Omega(n^{-2})$  on the probability that the contraction algorithm succeeds in finding a minimum cut for random graphs.

The key point of our probabilistic analysis of the contraction algorithm is to newly give an upper bound of the number of edges contracted during the course of the algorithm, while the analysis given by Karger is based on the lower bound on the number of remaining edges. By introducing a scheme called contraction tree we obtain the lower bound on the success probability of the contraction algorithm for the following three classes of graphs.

**Complete graphs.** The contraction algorithm finds a specific minimum cut with probability  $\Omega(n^{-1})$  for complete graphs.

**Graphs with two cliques.** Each graph consists of two  $n$ -cliques and  $\gamma(n-1)$  edges between the two cliques, where  $\gamma$  is a parameter satisfying  $0 < \gamma < 1$  that makes the graph have a unique minimum cut separating the two cliques. When  $\gamma$  is close to 1 the value of the unique minimum cut becomes very close to the values of those  $2n$  cuts each of which separates a single vertex from the others. Although it is apparently difficult for the contraction algorithm to find the unique solution when  $\gamma$  is close to 1, we show that it finds the unique minimum cut with probability  $\Omega(\gamma/n^\gamma)$ .

**Random graphs.** The random graphs are generated by a model introduced in [9], i.e., given a probability  $0 < p < 1$ , each random graph is generated by including an edge between each pair of vertices with probability  $p$ . Therefore each random graph  $G$  is generated with certain probability  $q_G$ . For given  $p$  and  $n$ , we define  $G(n, p)$  as a set of all simple graphs with  $n$  vertices with each of which  $q_G$  is associated. We assume  $(\log n)/n \leq p \leq 1$  guaranteeing that almost all random graphs generated are connected [9]. When performing the contraction algorithm on a given instance  $G$  of random graph, let  $P_G(\text{surv})$  denote the probability that the contraction algorithm obtains a specific minimum cut for  $G$ . We shall show that the expectation of the probability  $P_G(\text{surv})$  over random graphs in  $G(n, p)$  is  $\Omega(p/n)$ .

## 2. A new probabilistic analysis for random graphs and graphs with two cliques

The contraction algorithm proposed by Karger [6] is very simple. It repeats one fundamental operation: contracting a randomly chosen edge in a graph. Contraction of an edge connecting two vertices  $v_1$  and  $v_2$  results in a new vertex  $v$  called *super vertex*, by removing all edges between  $v_1$  and  $v_2$  and letting the set of edges incident on  $v$  be the union of the sets of remaining edges incident on  $v_1$  and  $v_2$ . Karger gives an analysis of his algorithm based on the following observation: Fix some specific minimum cut of  $c$  edges. Suppose that no edges in the minimum cut are selected during the contractions. Then after each contraction the contracted graph must have a minimum degree of at least  $c$ , since the contracted graph has a minimum cut of  $c$  edges. Let  $m(j)$  denote the number of remaining edges at the beginning of the  $j$ th contraction. Since the number of vertices decreases by one for each contraction,

$$m(j) \geq (n - j + 1)c/2. \quad (1)$$

We introduce a new scheme to count the number of edges of contracted graphs in a different way. We represent a sequence of contractions by a binary tree called a *contraction tree*. Each leaf of the contraction tree corresponds

to a vertex in the original graph. Starting with  $n$  isolated leaves, each time an edge between two super vertices in the contracted graph is contracted, we create a new internal node as well as two arcs from the new node to its two children in the contraction tree corresponding to the two super vertices.

### 2.1. Complete graphs

In order to clarify our new technique, we start with analyzing the performance of the contraction algorithm when applied to complete graphs. It is clear that in a complete graph with  $n$  vertices a minimum cut separates one single vertex from the others, and there are  $n$  distinct minimum cuts. Although it looks useless to analyze the contraction algorithm for complete graphs, it does provide a starting point for the more nontrivial graphs dealt with in the subsequent sections.

Consider a particular minimum cut in a complete graph, and suppose that the contraction algorithm never contracts any edge in the minimum cut during the process of the algorithm. Let  $d(j)$  denote the sum of numbers of edges contracted during the first  $j$  contractions. We have

**Lemma 1.**  $d(j) \leq j(j+1)/2$ .

**Proof.** We prove it by induction on  $j$ . For  $j = 1$  the result holds since the number of contracted edges is one. Assume that the result holds for  $j - 1$ . Now we consider the case of  $j$ . Suppose that the  $j$ th contraction creates an internal node whose left subtree is  $T_\ell$  and right subtree is  $T_r$ . Suppose that  $T_\ell$  (respectively  $T_r$ ) has  $q_1$  (respectively  $q_2$ ) leaves. From the induction hypothesis, the contractions performed corresponding to  $T_\ell$  (respectively  $T_r$ ) contract at most  $q_1(q_1 - 1)/2$  (respectively  $q_2(q_2 - 1)/2$ ) edges. Notice that in order for  $T_\ell$  (respectively  $T_r$ ) to have  $q_1$  (respectively  $q_2$ ) leaves, we need  $q_1 - 1$  (respectively  $q_2 - 1$ ) contractions. Moreover, the number of edges contracted by the  $j$ th contraction is  $q_1q_2$ . In general, there may be other internal nodes which are not in the subtrees of  $T_\ell$  and  $T_r$ . For simplicity we assume that all these internal nodes form a single subtree containing  $q_3$  leaves. (The other cases can be treated similarly.) We have

$$d(j) \leq q_1(q_1 - 1)/2 + q_2(q_2 - 1)/2 + q_1q_2 + q_3(q_3 - 1)/2.$$

Since the current contraction is the  $j$ th contraction,

$$(q_1 - 1) + (q_2 - 1) + (q_3 - 1) + 1 = q_1 + q_2 + q_3 - 2 = j.$$

Thus we have

$$\begin{aligned} d(j) &\leq q_1(q_1 + q_2 - 1)/2 + q_2(q_1 + q_2 - 1)/2 + q_3(q_3 - 1)/2 \\ &\leq (q_1 + q_2)(q_1 + q_2 - 1)/2 + q_3(q_3 - 1)/2 + (q_1 + q_2 - 1)(2q_3 - 2)/2 \\ &= (q_1 + q_2 + q_3 - 2)(q_1 + q_2 + q_3 - 1)/2 = j(j+1)/2. \quad \square \end{aligned}$$

By Lemma 1 the maximum value of  $d(j)$  is attained when each contraction is carried out in such a way that an edge between a single vertex and a super vertex is contracted.

**Theorem 2.** *The contraction algorithm produces a specific minimum cut of an  $n$ -vertex complete graph with probability  $\Omega(n^{-1})$ .*

**Proof.** Let  $P_j(\text{dest})$  denote the conditional probability that some specific minimum cut is destroyed for the first time at the  $j$ th contraction under the condition that it has not been destroyed for the first  $j - 1$  contractions. Since a minimum cut has  $n - 1$  edges in an  $n$ -vertex complete graph,

$$P_j(\text{dest}) = \frac{n-1}{m(j)} \leq \frac{2(n-1)}{n(n-1) - j(j-1)}.$$

When the contraction algorithm terminates, each original vertex has been merged into one of the two remaining super vertices. This obviously defines a cut of the original graph. Let  $P(\text{surv})$  denote the survival probability of the specific minimum cut after  $(n-2)$  contractions. Thus,

$$\begin{aligned} P(\text{surv}) &= \prod_{j=1}^{n-2} (1 - P_j(\text{dest})) \geq \prod_{j=1}^{n-2} \left( 1 - \frac{2(n-1)}{n(n-1) - j(j-1)} \right) \\ &= \prod_{j=1}^{n-2} \frac{(n-j-1)(n+j-2)}{(n-j)(n+j-1)} = \frac{1}{2n-3}. \quad \square \end{aligned}$$

**Corollary 3.** *The contraction algorithm produces a minimum cut of an  $n$ -vertex complete graph with probability higher than  $1/2$ .*

### 2.2. Graphs with two cliques

The discussion in the previous subsection can be extended to a class of graphs with two cliques defined in Section 1. Let  $G$  be a graph of  $2n$  vertices that has a special structure of containing two  $n$ -cliques and a unique minimum cut separating them. Assume that the number of edges in the minimum cut is  $\gamma(n-1)$ , where  $0 < \gamma < 1$ . The graph has  $m = n(n-1) + \gamma(n-1)$  edges. Suppose that  $J$  contractions have been executed on the graph and no edges of the minimum cut are contracted. We observe that (i) when  $J < n$ , the most destructive way of contraction is to consecutively contract the edges all of which belong to the same clique until the clique becomes a single supervertex in such a way that every contraction contracts an edge connecting a single vertex and a super vertex, and that (ii) when  $n \leq J \leq 2n-2$ , the most destructive way of contraction is to contract edges in the other clique in the manner as described in (i). Thus the number of edges remaining at the beginning of the  $j$ th contraction satisfies

$$m(j) \geq \begin{cases} n(n-1) - \frac{1}{2}j(j-1) + \gamma(n-1), & j \leq n-1, \\ \frac{1}{2}n(n-1) - \frac{1}{2}(j-n)(j-n+1) + \gamma(n-1), & \text{otherwise.} \end{cases}$$

The following fact is required to obtain the desired results.

**Fact.** *Let  $x, y$  and  $z$  be given positive numbers with  $x \geq y$ . Then*

$$\frac{y+z}{x+z} \geq \frac{y}{x}. \tag{2}$$

The survival probability of the minimum cut is

$$\begin{aligned} P(\text{surv}) &\geq \prod_{j=1}^{n-1} \frac{n(n-1) - \frac{1}{2}j(j-1)}{n(n-1) - \frac{1}{2}j(j-1) + \gamma(n-1)} * \prod_{j=n}^{2n-2} \frac{\frac{1}{2}n(n-1) - \frac{1}{2}(j-n)(j-n+1)}{\frac{1}{2}n(n-1) - \frac{1}{2}(j-n)(j-n+1) + \gamma(n-1)} \\ &= \prod_{j=1}^{n-1} \frac{2n(n-1) - j(j-1)}{2n(n-1) - j(j-1) + 2\gamma(n-1)} * \prod_{i=1}^{n-1} \frac{n(n-1) - i(i-1)}{n(n-1) - i(i-1) + 2\gamma(n-1)}. \tag{3} \end{aligned}$$

We derive lower bounds of the two terms in (3) as follows. For the first one we obtain

$$\begin{aligned} \prod_{j=1}^{n-1} \frac{2n(n-1) - j(j-1)}{2n(n-1) - j(j-1) + 2\gamma(n-1)} &> \prod_{j=1}^{n-1} \frac{2n^2 - j^2 + j - 2n}{2n^2 - j^2 + j} \quad (\text{replace } \gamma \text{ by } 1) \\ &= \prod_{j=1}^{n-1} \frac{(\sqrt{2n} + j - 1)(\sqrt{2n} - j - 1) + (j + 2\sqrt{2n} - 2n - 1)}{(\sqrt{2n} + j)(\sqrt{2n} - j) + j} \\ &\quad (\text{by (2)}) \\ &> \frac{\sqrt{2n}}{(1 + \sqrt{2})^2(\sqrt{2n} - 1)}. \end{aligned}$$

Using the Gamma function [1] to evaluate the second term, we have

$$\begin{aligned} \prod_{i=1}^{n-1} \frac{n(n-1) - i(i-1)}{n(n-1) - i(i-1) + 2\gamma(n-1)} &= \prod_{i=1}^{n-1} \frac{(n-i)(n+i-1)}{(n-i+\gamma)(n+i-1+\gamma) - (\gamma + \gamma^2)} \\ &> \prod_{i=1}^{n-1} \frac{(n-i)(n+i-1)}{(n-i+\gamma)(n+i-1+\gamma)} = \Theta\left(\frac{\gamma}{n^\gamma}\right). \end{aligned}$$

**Theorem 4.** *The contraction algorithm finds the unique minimum cut of a graph defined above with probability  $\Omega(\gamma/n^\gamma)$ .*

### 3. Asymptotic analysis for random graphs

In this section we analyze the asymptotic performance of the algorithm for random graphs. The random graph model used here is the one described in [9], which was mentioned in Section 1. Each graph  $G$  in  $G(n, p)$ , generated with certain probability  $q_G$ , is expected to have  $pn(n-1)/2$  edges. It is obvious that with high probability the number of edges in a minimum cut of a graph  $G$  is at most  $p(n-1)$ . Let  $Ex(\cdot)$  denote the expectation of a random variable associated with graphs from  $G(n, p)$ . The following lemma can be proved in a manner similar to Lemma 1.

**Lemma 5.** *Let  $Ex(d(j))$  be the expectation of the number of edges contracted during the first  $j$  contractions. Then*

$$Ex(d(j)) \leq \frac{pj(j-1)}{2} + j. \tag{4}$$

**Proof.** First, consider a contraction process that at each contraction an edge is chosen between a super vertex and a single vertex. Since there must be at least one edge connecting the vertex with the super vertex, the expected number of edges contracted at the  $i$ th contractions is  $p(i-1) + 1$ . The expected number of edges contracted during the first  $j$  contractions is

$$\sum_{i=1}^j (p(i-1) + 1) = \frac{pj(j-1)}{2} + j. \tag{5}$$

As for the other contraction process, we prove (4) by induction on  $j$ . After the first  $j-1$  contractions, suppose we have three super vertices  $V_i, i = 1, 2, 3$ , in the current contracted graph, and each  $V_i$  contains  $q_i$  vertices of the original graph. (The other case can be treated in a similar manner.) Suppose that  $V_1$  and  $V_2$  are to be merged at the  $j$ th contraction. Then the expected number of edges to be contracted at the  $j$ th contraction is

$p(q_1q_2 - 1) + 1$ . We can show that the expected number of edges contracted during the first  $j$  contractions is at most  $pj(j - 1)/2 + j$  by an argument similar to that of Lemma 1.  $\square$

Note that when  $p = 1$  Lemma 5 coincides with Lemma 1.

**Theorem 6.** For random graphs in  $G(n, p)$ ,  $(\log n)/n \leq p \leq 1$ , the expectation of the probability that the contraction algorithm succeeds in finding a specific minimum cut is  $\Omega(p/n)$ .

**Proof.** By the same argument in the previous section the expected number of remaining edges before the  $j$ th contraction is

$$\text{Ex}(m(j)) = \text{Ex}(m - d(j - 1)) \geq \frac{pn(n - 1)}{2} - \frac{p(j - 2)(j - 1)}{2} - (j - 1). \tag{6}$$

When performing the contraction algorithm on a given instance  $G$  of the random graphs, we define the conditional probability  $P_{G_j}(\text{surv})$  (respectively  $P_{G_j}(\text{dest})$ ) that a specific minimum cut of  $G$  survives (respectively is destroyed) at the  $j$ th contraction, under the condition that no edges in the cut has been contracted by the first  $j - 1$  contractions. Denote by  $\text{Ex}(P_{G_j}(\text{surv}))$  (respectively  $\text{Ex}(P_{G_j}(\text{dest}))$ ) the expectation of  $P_{G_j}(\text{surv})$  (respectively  $P_{G_j}(\text{dest})$ ) taken over all graphs from  $G(n, p)$ . Similarly let  $\text{Ex}(P_G(\text{surv}))$  be the expectation of the probability that the algorithm succeeds in finding a specific minimum cut of  $G$ . Therefore the expectation  $\text{Ex}(P_G(\text{surv}))$  is given by  $\sum_{G \in G(n, p)} P_G(\text{surv})q_G$ .

$$\begin{aligned} \text{Ex}(P_{G_j}(\text{surv})) &= 1 - \text{Ex}(P_{G_j}(\text{dest})) \\ &\geq 1 - \frac{2p(n - 1)}{pn(n - 1) - p(j - 2)(j - 1) - 2(j - 1)} \\ &= \frac{(n - 2)(n - 1) - (j - 2)(j - 1) - 2(j - 1)/p}{n(n - 1) - (j - 2)(j - 1) - 2(j - 1)/p} \\ &= \frac{(n + j - 1 + 1/p)(n - j - 2 - 1/p) + (4j + 3/p + 1/p^2 - 2)}{(n + j + 1/p)(n - j - 1 - 1/p) + (4j + 3/p + 1/p^2 - 2)} \\ &\geq \frac{(n + j - 1 + 1/p)(n - j - 2 - 1/p)}{(n + j + 1/p)(n - j - 1 - 1/p)} \quad (\text{by (2)}). \end{aligned} \tag{7}$$

Recall if we use Karger’s lower bound (1) to give an expected number of remaining edges before the  $j$ th contraction we have

$$\text{Ex}(m(j)) \geq \frac{(n - j + 1)p(n - 1)}{2} \tag{8}$$

for random graphs. Comparing (6) with (8) we find that after a certain number of contractions, i.e.,  $j > n - 3 - \lceil 2/p \rceil$ , the right-hand side of (8) becomes greater than that of (6), which implies that Karger’s lower bound (8) becomes better than (6) for  $j > n - 3 - \lceil 2/p \rceil$ . Thus letting  $J = n - 3 - \lceil 2/p \rceil$ , and using Karger’s lower bound of (8) for  $j \geq J + 1$ , we have

$$\begin{aligned} \text{Ex}(P_G(\text{surv})) &\geq \prod_{j=1}^J \text{Ex}(P_{G_j}(\text{surv})) * \prod_{j=J+1}^{n-2} \left(1 - \frac{2}{n - j + 1}\right) \\ &\geq \prod_{j=1}^J \frac{(n + j - 1 + 1/p)(n - j - 2 - 1/p)}{(n + j + 1/p)(n - j - 1 - 1/p)} * \prod_{j=J+1}^{n-2} \left(1 - \frac{2}{n - j + 1}\right) \\ &\geq \frac{(n - J - 2 - 1/p)(n + 1/p)}{(n - 2 - 1/p)(n + J + 1/p)} * \frac{2}{(n - J)(n - J - 1)} = \Theta\left(\frac{p}{n}\right). \quad \square \end{aligned} \tag{9}$$

In Section 2.2 we considered a class of graphs with two cliques of the same size. We can apply the discussion to a more general case, which is described in [3]. Given a graph of  $2n$  vertices, the vertex set is randomly partitioned into two subsets with equal size. Edges joining vertices in the same subset are selected with probability  $p$  and edges joining vertices in different subsets are selected with probability  $p\gamma/(n-1)$ ,  $0 < \gamma < 1$ . The cut separating the two subsets is almost surely a unique minimum cut of the graph. Combining the discussion of this and the previous sections it is not difficult to obtain an expectation of probability  $\Omega(\gamma p/n^\gamma)$  that the contraction algorithm finds the desired minimum cut.

#### 4. Concluding remarks

In this paper we presented a new probabilistic analysis of Karger's randomized algorithm for a few classes of unweighted graphs. Our analysis was motivated by the observation of the gap between Karger's theoretical bound and our computational results of the contraction algorithm. The other motivation we like to point out came from the attempt of giving an interpretation of the practical behavior of our recently devised randomized algorithm [2] for solving minimum cut problems. Our algorithm approximately solves a minimum cut problem by repeatedly solving minimum range cut problems [8] with randomly given edge weights. Although no strictly better bound than Karger's has been obtained, the usefulness of our method is demonstrated by computational results in [2] for various types of randomly generated graphs, and it finds a minimum cut with a probability higher than or equal to the one that contraction algorithm does at a single trial [2]. Consequently the stronger lower bound of the contraction algorithm obtained in this paper will also give theoretical support on the effectiveness of our method.

#### References

- [1] J. Dieudonne, *Infinitesimal Calculus*, Hermann, Paris, 1971.
- [2] Y. Dai, H. Imai, K. Iwano, K. Katoh, K. Ohtsuka, N. Yoshimura, A new unified approximate approach to the minimum cut problem and its variants using minimum range cut algorithms, IBM Research Rept., RT0082, 1992; also: *Discrete Appl. Math.* 65 (1996) 167–190.
- [3] M.E. Dyer, A.M. Frieze, The solution of some random NP-hard problems in polynomial expected time, *J. Algorithms* 10 (1989) 452–489.
- [4] H.N. Gabow, A matroid approach to finding edge connectivity and packing arborescences, in: *Proc. ACM 23rd Symp. on Theoretical Computer Science*, 1991, pp. 112–122.
- [5] H.N. Gabow, Applications of a poset representation to edge connectivity and graph rigidity, in: *Proc. 32nd IEEE Symp. on Foundations of Computer Science*, 1991, pp. 812–821.
- [6] D.R., Karger, Global min-cuts in  $\mathcal{RNC}$  and other ramifications of a simple min-cut algorithm, in: *Proc. 3rd ACM/SIAM Symp. on Discrete Algorithms*, 1993, pp. 21–31.
- [7] D.R. Karger, Random sampling in matroids, with applications to graph connectivity and minimum spanning trees, in: *Proc. 34th IEEE Symp. on Foundations of Computer Science*, 1994, pp. 84–93.
- [8] N. Katoh, K. Iwano, Efficient algorithms for minimum range cut problems, IBM Research Report, in: *Proc. 2nd WADS, Lecture Notes in Computer Science*, vol. 519, Springer, Berlin, 1991, pp. 80–91; also in: *Networks* 24 (1994) 395–407.
- [9] E.M. Palmer, *Graphical Evolution*, John Wiley & Sons, New York, 1985.