最小カット問題の新しい確率的アルゴリズム：計算実験による性能評価

大塚 啓司* 戴 陽* 加藤 直樹* 岩野 和生†

* 神戸商科大学 管理科学科
† 日本ＩＢＭ 東京基礎研究所

$G=(V,E)$ を n 個の節点と m 本の枝からなる連結な無向グラフとするとき、G の初等的カットのなかで枝数最小のカット（最小カット）を求める確率的アルゴリズムを提案し、その性能を計算機実験で評価する。アルゴリズムは最大格差最小カットを求める高速解法にもとずくもので、その理論的評価は未解決であるが、計算機実験の結果、最小カットを求める厳密解法と比較すると少ない計算量で最小カットに近いカットが求められることが確かめられた。また同様の手法を G を等分割する最小カットを求める問題（グラフの２分割問題）に対しても開発し、その性能を従来の Kernighan and Lin による近似解法と比較実験した。

A New Randomized Algorithm for the Minimum Cut Problem and Its Performance Evaluation

Keisi Ohtsuka* Yang Dai* Naoki Katoh* Kazuo Iwano†

* Department of Management Science,
Kobe University of Commerce
† Tokyo Research Laboratory, IBM Japan

Let $G=(V,E)$ be a connected undirected graph with n vertices and m. We shall propose a new randomized algorithm for finding a minimum cut based on a fast algorithm for finding a minimum range cut. Though the theoretical evaluation has not been done yet, computer experiments show that our algorithm outputs a cut very close to the exact minimum cut with the running time shorter than the best known algorithm for the minimum cut problem.

We also propose a randomized algorithm based on the similar idea for finding a minimum cut of G into equal sizes. We also report the results of computer experiments that compare our algorithm with Kernighan and Lin's algorithm.

1　はじめに

$G=(V,E)$ を n 個の節点と m 本の枝からなる連結な無向グラフ（多重辺を持つことを許す）とするとき、G の初等的カットとは Vの分割 (V_1,V_2) によって定まる Eの部分集合 $E'=\{e=(u,v)|u\in V_1, v\in V_2\}$ のことである。G の初等的カットの中でその枝数最小のカットを**最小カット**と呼ぶ。最小カットの枝数は G の**枝連結度**と呼ばれる。小カットを求めるアルゴリズムは従来から良く研究されているが ([1], [9]) 、最近、永持、茨木の両氏によって $O(m+\lambda(G)n^2)$ のアルゴリズムが提案された [7] 。ここで$\lambda(G)$ は G の最小カットの枝数、p は枝を持つ点対の数を表す。本論文では最小カットを求める新しい確率的アルゴリズム (randomized algorithm) を提案し、計算機実験による性能評価について報告する。提案するアルゴリズムは筆者のうちの二人加藤、岩野が昨年発表した最大格差最小カットのアルゴリズム [4] にもとずいている。各枝 e に重み $w(e)$ が与えられている連結な無向グラフ（多重辺を持つことを許す）$G=(V,E)$ において、G のカットのなかでそのカットに属する枝の重みの中でその最大値と最小値の差（それを最大格差と呼ぶ）を最小にするカットが最大格差最小カットである。[4] は $O(m+n\log n)$ の算法を提案している。

本論文で提案するアルゴリズムのアイデアは次の通りである。まず与えられたグラフ G の各枝の重みを $[0,1]$ の一様乱数によって定める。そして [4] のアルゴリズムによって最大格差最小カットを求める。そのカットに属する枝の本数を数える。このプロセスをある回数繰り返してそのなかで位数最小のカットを最小カットとして出力する。このプロセス一回当りの計算量は $O(m+n\log n)$ であるので l回繰り返すと全体で $O(lm+ln\log n)$ の計算量を要する。与えられたグラフが多重辺を持たない場合（単純グラフの場合）、$l=o(\min\{n^2/m,n\})$ ならこの計算量は永持－茨木法より少ない。そこで計算実験では $l=\log n$, $l=\sqrt{n}$の 2 つの場合を考え、点数 50～1000 の単純グラフを対象とした。そして正確な最小カットと提案するアルゴリズムが与えるカットとの枝数の相対誤差によって近似度を評価した。その結果点数 50～150 の密なグラフにおいて $l=\log n$, $l=\sqrt{n}$に対しては最適解との相対誤差の平均は各々$7\sim 8\%$, $5\sim 6\%$ であった。また生成したグラフの 7 割以上に対して相対誤差は 10%以内に収まっていた。点数 150～1000 の疎なグラフにおいてはほとんど最小カットを得ることが出来た。この結果提案する近似アルゴリズムは厳密解を求めるアルゴリズムと比較してより少ない時間で最適解に近い解を求めることが確かめられた。

また同様の手法を G を等分割する最小カットを求める問題（グラフの等分割問題）に対しても開発し、その性能を従来の Kernighan and Lin [5] による近似解法（ KL 法）と比較実験した。この問題はＶＬＳＩ設計などに現れる問題であり、従来からよく研究されているが（ [10] ）、KL 法が一番優れているようである。提案する方法は最大格差最小等分割カットのアルゴリズムにもとずいている。等分割カットとは G のカットで Vを等しいサイズに分割するカットのことである。グラフの各枝 e に重み $w(e)$ が与えられている無向グラフに対してカットに属する枝の中で重みの最大値と最小値の差が最小の等分割カットを求める問題が最大格差最小等分割カット問題である。本論文ではこの問題に対する $O(m+n^3)$ 時間アルゴリズムを提案する。また$\epsilon-$近似等分割カットのなかで最大格差最小のカットを求める問題に対しても考察し、$O(m+n^2/\epsilon)$ 時間アルゴリズムを提案する。ここで$\epsilon-$近似等分割カットとはグラフを丁度等しいサイズに分割するのではなく $n(1-\epsilon)/2\le|V_i|\le n(1+\epsilon)/2$ $(i=1,2)$ を満たすような分割 (V_1,V_2) が与えるカットのことである。これらの多項式時間アルゴリズムを利用して、最小カット問題と同様のアイデアにもとずいてグラフの等分割問題と$\epsilon-$近似等分割問題に対する確率的アルゴリズムを提案する。グラフの等分割問題や$\epsilon-$近似等分割問題は NP-完全であることが知られているが [3] 、KL 法はかなり高速に最適解に近い解を与えることが知られている。そこでわれわれのアルゴリズムと KL 法との性能比較を行った。現在のところ KL 法を上回る性能は得られていないが、KL 法と同程度の時間で KL 法が与える解に近い解が得られることが確かめられた。また我々の方法は従来の逐次改善法とは異なるまったく新しい方法であるという点でさらに検討する必要性があろうと考えられる。

本論文の構成は以下の通りである。2 節では最小カットを求める確率的アルゴリズムを提案する。3 節ではその計算機実験の結果について述べる。4 節では最大格差最小等分割問題の多項式時間解法について述べる。5 節では最大格差最小ϵ- 近似等分割問題の多項式時間解法について述べる。6 節では 4 節で提案したアルゴリズムをグラフの等分割問題に対して適用した時の実験結果について報告する。

2 最小カット問題の確率的アルゴリズム

前節でも述べたように、われわれのアルゴリズムは最大格差最小カット問題のアルゴリズムにもとずいている。もともと与えられたグラフの枝には重みは付されていないが、われわれの方法では各枝 e に $[0,1]$ の一様乱数によって重み $w(e)$ を付加する。このように重みの与えられたグラフ G に対して G のカット Cの最大格差 $range(C)$ を $range(C) = \max_{e \in C} w(e) - \min_{e \in C} w(e)$ によって定義する。$w(e)$ が上のようにして与えられた一様乱数であるから Cを固定して考えると $range(C)$ は確率変数である。このとき、$range(C)$ の期待値は

$$E(range(C)) = \frac{|C|-1}{|C|+1}$$

となることが容易に示される（証明略）。したがって $|C|$ が小さいほど $range(C)$ の期待値も小さくなることは明らかである。このことから枝に一様乱数で重みを付加するという操作を繰り返す毎に、最大格差最小カットを求め、同時にその枝数も記憶しておき、今まで得られたカットの中で枝数最小のカットを記憶しておくと、繰り返しが進むにつれて最小カットにより近い解が得られるということは明らかであろう。アルゴリズムはあらかじめその繰り返し回数を定めておいて実行させる。アルゴリズムの詳細は以下の通りである。

Procedure *Randomized-Minimum-Cut*
begin
(1) l :=繰り返し回数、$\nu^* := +\infty$ とする。
(2) **do** $k = 1$ **to** l
(2.1) G の各枝 e に $[0,1]$ の一様乱数によって重み $w(e)$ を付加する。
(2.2) G の最大格差最小カット Cを求め $|C|$ をカウントする。
(2.3) $|C| < \nu^*$なら、$\hat{C} := C$, $\nu^* := |C|$ とおく。
end
(3) $\hat{C}$を出力する。
end

Figure 1. Algorithm *Randomized-Minimum-Cut*

[4] より最大格差最小カットが $O(m + n \log n)$ 時間で得られることから、次の定理が得られる。

定理 1 上の手続き *Randomized-Minimum-Cut* の計算量は $O(lm + ln \log n)$ である。

アルゴリズムが出力する近似解$\hat{C}$がどの程度最小カットに近いかを理論的に評価できることが望まれるが、この点はまだ未解決である。また容量付きのグラフの最小カットを求める問題への本解法の拡張などが今後の課題である。

3 Randomized-Minimum-Cut の実験結果

本節では前節で提案したアルゴリズム *Randomized-Minimum-Cut* の計算機実験の結果について述べる。実験で対象としたのは点数 50 ～ 1000 のグラフである。点数 50, 150 については疎密度 50%～90%の密なグラフを対象とし、$l = \log_2 n$, $l = \sqrt{n}$の２つの場合に対して実験を行った。その計算結果は表 1 に示されている。なお生成したグラフの数は各点数、各疎密度毎に２０である。点数 150～1000 については表 2 に示されているような疎なグラフを対象とし、$l = \log_2 n$ の場合に対して実験を行った。その結果は表 2 にまとめられている。なお疎なグラフの生成はランダムネットワーク生成のためのプログラム NETGEN（文献 [6] を参照）によった。

表 1　節点数５０と１５０の密なグラフに対する相対誤差（%）

	節点数　５０					
	疎密度					
繰り返し回数	90%	80%	70%	60%	50%	平均
$l = \log_2 n$	5.60	5.25	9.75	7.55	7.83	7.20
$l = \sqrt{n}$	5.07	4.48	7.65	7.55	7.83	6.52

	節点数　１５０					
	疎密度					
繰り返し回数	90%	80%	70%	60%	50%	平均
$l = \log_2 n$	4.28	5.22	9.20	14.20	10.75	8.73
$l = \sqrt{n}$	3.47	4.00	6.36	8.01	6.85	5.74

表２　節点数 150～1000 の疎なグラフに対する相対誤差（％）

	節点数　１５０							
枝数	975	980	979	976	983	1902	1926	1895
最小カットの値	3	5	2	2	2	2	2	2
相対誤差（％）	0	0	0	0	0	0	0	0

	節点数　３００							
枝数	3925	3920	3903	3928	3915	9474	9466	9492
最小カットの値	2	2	2	2	2	2	2	17
相対誤差（％）	0	0	0	0	0	0	0	11.6

	節点数　５００				節点数　１０００
枝数	4948	4947	19178	19223	9902
最小カットの値	2	2	3	5	1
相対誤差（％）	0	0	0	0	0

以上の計算結果から、密なグラフに対しては $l = \log n$, $l = \sqrt{n}$ に対しては最適解との相対誤差が各々7 ～ 8%, 5 ～ 6% 程度のカットが求められることが分かった。また疎なグラフに対しては十分実験を行なったとは言えないが、実験結果から判断すると、ほとんどの場合最小カットを与えている。

4　最大格差最小等分割カット問題

本節では、グラフの等分割問題に対しても、前節と同じような確率的アルゴリズムが、考えられることを示す。まず始めに、グラフの最大格差最小等分割カット問題を定義し、その効率の良いアルゴリズムをしめす。

等分割カットとは無向グラフ $G = (V, E)$ のカットで V を等しいサイズに分割するカットのことである。今、$n = |V|$ は、偶数であるとする。グラフの各枝 e に重み $w(e)$ が与えられている無向グラフに対してカットに属する枝の中で重みの最大値と最小値の差が最小の等分割カットを求める問題が最大格差最小等分割カット問題である。

本論文では、簡単のために、すべての辺の重みは、異なるものとする。今、$E[\alpha, \beta] = \{e \in E \mid \alpha \le w(e) \le \beta\}$ とする。また、$E[\alpha] = E[\alpha, \alpha]$ とする。区間 $[\alpha, \beta]$ が、*feasible* とは、$E[\alpha, \beta]$ の部分集合が、G の等分割カットを構成することをいう。区間 $[\alpha, \beta]$ が、*infeasible* とは、区間 $[\alpha, \beta]$ が feasible でないことを意味する。区間 $[\alpha, \beta]$ が *critical* とは、区間 $[\alpha, \beta]$ が feasible であり、区間 $[\alpha, \beta]$ のどの真の部分区間も feasible でないことを、意味する。

アルゴリズム Minimum-Range-Equi-Partition は、Martello ら [8] による最大格差最小問題の一般的な解法に基ずいており、*find-lower-value*(α, β), *find-upper-value*(α, β), *Feasible*(α, β) の３つのサブルーチンから成り立っている。*find-lower-value*(α, β) は、feasible な区間 $[\alpha, \beta]$ の中で critical な部分区間 $[a, \beta]$ を探す。ただし $\alpha \le a \le \beta$ である。*find-upper-value*(α, β) は、区間 $[\alpha, \beta]$ が与えられて、次に feasible な区間 $[a, b]$ を探す。次に feasible な区間とは、区間 $(\alpha, b]$ の部分区間が、critical な区間を含むような b のうち最小のものである。*Feasible*(α, β) は、区間 $[\alpha, \beta]$ が、feasible であるかどうかを調べる。

アルゴリズム Minimum-Range-Equi-Partition は、次の補題により、最初に調べる辺の数を $O(n)$ に減らしている。

補題 2 ([4]) T_{min} と T_{max} をそれぞれ、G の最小木と最大木としょう。このとき、任意のカット C について、C の重み最大の枝は T_{max} に属し、C の重み最小の枝は T_{min} に属する。 □

そして、本アルゴリズムは、critical な区間 $[\alpha, \beta]$ を α の小さい順に数えあげていく。その際、区間 $[\alpha, \beta]$ の feasibility を調べている時に、$E - E[\alpha, \beta]$ を、縮約してできるグラフ $G' = (V', E')$ を管理していく。そして、$v \in V'$ に対して

$f(v)$ を、v に縮約された元の点の数とする。

```
Procedure Minimum-Range-Equi-Partition
begin
(1)      最小木 T_min と最大木 T_max を計算し、E = T_min ∪ T_max とする。.
(2)      E の辺の重みの小さい順に w_1, w_2, ..., w_p とする。w_0 = 0 とする。
         G' = (V', E') を V' = {v}, E' = ∅ と定義する。
         β = w_1; z = w_p − w_1;
(3)      do k = 0 to p
(3.1)    α ← w_k;
(3.2)    (β, ok-bit) ← find-upper-value(α, β);
(3.3)    if ok-bit= false then exit;
(3.4)    (α, ok-bit) ← find-lower-value(α, β);
(3.5)    if ok-bit= false then exit;
(3.6)    if β − α < z then {z = β − α; α* = α; β* = β; }
         end
end


Procedure find-upper-value(α, β)
begin /* β = w_j とする。*/
(4)      do k = j to p
(4.1)    E[w_k] を uncontract して、G' と f(·) を更新する。
(4.2)    if Feasible(α, β) = true then return (w_k, true);
         end
         return (null, false);
end


Procedure find-lower-value(α, β)
begin /* α = w_i, β = w_j とする。*/
(5)      do k = i to j
(5.1)    E[w_k] を contract して、G' と f(·) を更新する。
(5.2)    if Feasible(α, β) = false then return (w_{k−1}, true);
         end
         return (null, false);
end


Procedure Feasible(α, β)
begin
(6)      V' の部分集合 V'' で Σ_{v∈V''} f(v) = n/2 となるものがあるかどうかを
         探す 0-1 Knapsack 問題を解く。
(7)      if 解が存在する then return true;
         else return false;
end
```

Figure 2. Algorithm *Minimum-Range-Equi-Partition*

Feasible ルーチンは、動的計画法により $O(n^2)$ で解くことができる [11] 。また、(1) は $O(m+n\log n)$ [2] で、(2) は $O(n\log n)$ で、(4.1) と (5.1) は $O(n)$ で解くことができるので、次の定理を得る。

定理 3 アルゴリズム *Minimum-Range-Equi-Partition* は、グラフの最大格差最小等分割問題を、$O(m+n^3)$ で正しく解く。 □

5 最大格差最小ϵ-近似等分割問題

次に、最大格差最小ϵ-近似等分割問題が、$O(m+n^2/\epsilon)$ で解けることを示そう。ここで、ϵ-近似等分割カットとは、点集合 Vを V_1と V_2 に分割し、その時、$(1-\epsilon)n/2 \leq |V_i| \leq (1+\epsilon)n/2$ $(i=1,2)$ を、満たすカットである。最大格差最小ϵ-近似等分割問題が、前節 Minimum-Range-Equi-Partition アルゴリズムの Feasible を、下記のϵ-Feasible に置き換えることによって得られる。

Procedure *ϵ-Feasible*
begin
(1) $Large \leftarrow \{v \in V' \mid f(v) \geq \epsilon n/2\}$; $Small \leftarrow V' - Large$;
$w(Small) = \sum_{v \in Small} f(v)$ とする。
全ての $v \in V'$を $Small$の元が $Large$ の元よりも、先になるように並べたものを $a_1, a_2, \cdots, a_l$とする。
(2.1) **if** $w(Small) \geq n/2$ **then return** true;
(2.2) **else**$(1-\epsilon)n/2 - w(Small) \leq j \leq n/2$ なる jに対して、$Large$ の部分集合 $Large_2$で$\sum_{v \in Large_2} f(v) = j$となるものがあるどうかを探す 0-1 Knapsack 問題を解く。
(2.2.1) **if** 解が存在する **then return** true;
(2.2.2) **else return** false;
end

Figure 3. Algorithm *ϵ-Feasible*

定理 4 *ϵ-Feasible* は、$O(n/\epsilon)$ 時間で、正しくϵ-近似等分割問題を解く。 □

Proof. (2.1) で、求める分割は、$(\{a_1, a_2, \ldots, a_{s^*}\}, V' - \{a_1, a_2, \ldots, a_{s^*}\})$ によって与えられる。ここで、s^*は、次の条件を満たすものである。$1 \leq s^* \leq |Small| - 1$, $\sum_{i=1}^{s^*} f(a_i) < n/2$, $\sum_{i=1}^{s^*+1} f(a_i) \geq n/2$. $f(s^*+1) < \epsilon n/2$ なので、$\sum_{i=1}^{s^*} f(a_i) > (1-\epsilon)n/2$ となる。

また (2.2.1) では、上記と同様に、$P = Large_2 \cup \{a_1, a_2, \ldots, a_{s^*}\}$ とすると、$(P, V'-P)$ が求める分割である。ここで、s^*は、次の条件を満たすものである。$1 \leq s^* \leq |Small| - 1$, $j + \sum_{i=1}^{s^*} f(a_i) < n/2$, $j + \sum_{i=1}^{s^*+1} f(a_i) \geq n/2$. (2.2.2) では、明らかに求める分割が存在しないことがわかる。

計算時間については、(2.2) が $O(|Large|n/2)$ で解け、$|Large| \leq 2/\epsilon$であるので、$O(n/\epsilon)$ となる。 □

系 5 最大格差最小ϵ-近似等分割問題が、$O(m+n^2/\epsilon)$ で解ける. □

6 確率的等分割カットアルゴリズムの実験結果

本節では前節で提案したアルゴリズム *Minimum-Range-Equi-Partition* の計算機実験の結果について述べる。実験で対象としたのは点数 100, 200 のグラフである。表 3 は節点数 100 の疎なグラフに対する KL 法との比較実験結果をまとめている。KL 法は最初任意に等分割を求めた後、節点を交換することによって解を改善できれば改善していくという逐次改善法である。表 3 の実験では KL 法の初期分割をを乱数で定めるという操作を 100 回行った。同様にわれわれの方法では枝に乱数で重みを与えるという操作を 100 回行った。われわれの方法では各試行で得られた格差最小の critical な区間に対応するカットの枝数のみをカウントする one-counting 法 と、各試行で得られたすべての critical な区間 に対応するカットを求めそのなかで枝数最小のカットを記憶するという all-counting 法の 2 通りの方法を実験した。

表にある最小値とは 100 回の試行のなかで得られた最小値であり、l_{first}はその最小値を初めて達成した回のことである。また# of min. は 100 回の試行のうち、最小値を得た回数を表している。この表からわかるようにわれわれの方法は KL 法を上回る性能は達成していないが、かなりいい方法と言える。

表 3　節点数１００の疎なグラフにおける KL 法との比較結果

		KL 法			新解法 (one-counting)			新解法 (all-counting)		
枝数		最小値	l_{first}	# of min.	最小値	l_{first}	# of min.	最小値	l_{first}	# of min.
111	Case 1	5	12	4	5	69	1	5	69	2
	Case 2	6	1	13	7	78	1	6	27	3
130	Case 1	11	64	1	16	39	1	15	2	3
	Case 2	9	5	9	10	20	1	10	20	1

また節点数200の場合において疎なグラフから密なグラフまで実験を行った。その結果は表4に示されている。この実験ではKL法の試行回数は１０回とした。またわれわれの方法ではone counting法で実験を行い、その試行回数は表に示されているように100～300回である。なお計算時間であるが、一回の試行当りの計算時間はＫＬ法とわれわれの方法は余り大差無いようである。

この表から分かるようにこの実験で用いたグラフにおいてはわれわれの方法はKL法より良くない解を出力しているようである。表3と比較するとグラフが密になるとわれわれの方法は余りいい結果を与えないようである。その理由の一つととして考えられるのはカットの枝の本数が増加するとその重みの格差の期待値が1に近づくので格差の大小から枝の本数が判別しにくくなることが挙げられる。

しかしながら我々の方法は計算の効率化の点や、critical な区間が求められた後、できるだけカットの枝数が少ないような等分割を求めるヒューリスティクの開発などの点においてまださまざまな改善の余地があり、今後の研究課題である。またϵ-近似等分割でよければ、われわれの方法はより高速になることからその点についても実験を試みる予定である。

また従来の近似解法はいずれも逐次改善法と呼ばれる種類の解法であるが、我々の方法はそれらとはまったく異なる解法であり、その点でも興味深い。

表4　節点数２００のグラフにおけるKL法との比較結果

枝数	230	396	788	1559	3087	5878
新解法の試行回数	200	200	200	300	100	200
				最小値		
ＫＬ法	10	61	200	512	1188	2510
新解法	13	96	299	664	1424	2801

謝辞

最大格差最小カットを最小カットを求める確率的アルゴリズムに適用できることを示唆して頂いた東京大学理学部今井浩氏に感謝します。またNETGENのプログラムを提供して頂いた日本ＩＢＭ東京基礎研究所中野淳氏、Kernighan and Linのプログラムを提供して頂いた大阪大学工学部中野秀男氏に感謝します。なおこの研究の一部は文部省科学研究費補助金総合研究（Ａ）02302047 (1990) によっている。

References

[1] Even, S. and Tarjan, R.E., Network flow and testing graph connectivity, *SIAM J. Computing*, Vol. 4 (1975), pp. 507-518.

[2] Fredman, M.L. and Tarjan, R.E., Fibonacci heaps and their uses in improved network optimization algorithms, *Journal of the ACM*, Vol. 34, No. 3, (1987) pp. 596-615.

[3] Garey, M.R., Johnson, D.S. and Stockmeyer, L., Some simplified NP-complete graph problems, *Theoretical Computer Science*, Vol.1 (1976), pp. 237-267.

[4] Katoh, N. and Iwano, K., Efficient algorithms for minimum range cut problems, IBM Research Report, RT0057 (1991) (also appeared in Proceedings of 2nd Workshop, WADS '91, Lecture Notes in Computer Science 519 Springer-Verlag, pp. 80-91 (1991)).

[5] Kernighan, B.W. and S. Lin, An effective heuristic procedure for partitioning graphs, *BSTJ*, Vol.49, No.2, (1970), pp. 291- 307.

[6] Klingman, D., Napier, A. and Stutz, J., NETGEN: A program for generating large scale capacitated assignment, transportation, and minimum cost flow network problems, *Management Science*, Vol.20, No.5 (1974), pp. 814-821.

[7] Nagamochi, H. and Ibaraki, T., Computing edge-connectivity in multiple and capacitated graphs, Proceedings of International Symposium SIGAL '90, Tokyo, Japan, August 1990, Lecture Notes in Computer Science 450 Springer-Verlag, pp. 12-20 (1990).

[8] Martello, S., Pulleyblank, W.R., Toth, P., and de Werra, D., Balanced Optimization Problems. *Operations Research Letters*, Vol. 3, No. 5, 275-278. 1984.

[9] Matula, D.W., Determining edge connectivity in $O(nm)$, *Proc. 28th IEEE Symp. Found. Compt. Sci.*, (1987), pp. 249-251.

[10] 中野秀男、中西義郎、回路の２分割に対する Simulated Annealing 法、電子通信学会研究資料、回路とシステム研究会、ＣＡＳ 84-124, 1984 年.

[11] Papadimitriou, C.H. and Steiglitz, K., *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.