

# 通信ネットワークのモデル化と最適化

小林 佑輔, 福永 拓郎

グラフ・ネットワークを扱う最適化問題は組合せ最適化の主要なトピックの一つである。本稿では、グラフ・ネットワーク最適化の通信ネットワークへの応用について述べる。通信技術は近年ますます複雑になっており、さまざまな新しいトピックが生まれている。本稿では、円板形領域損傷モデルにおける最大流最小カット問題と、仮想マシンの配置問題という二つの研究成果について解説する。

キーワード：通信ネットワーク, 円板形領域損傷モデル, 仮想マシン配置問題

## 1. はじめに

グラフ・ネットワークを扱う最適化問題は組合せ最適化の主要なトピックの一つである。ネットワークフロー, 最大流最小カット定理, 全域木問題などは大学の学部生向けの授業で取り上げられる機会も多く, ご存じの読者も多いだろう。現在でも活発に研究されており, 数理計画, 離散数学, 計算の複雑さなどの理論に支えられて非常に面白い進展が続いている。

応用面でも多様な広がりを見せている。ウェブグラフやソーシャルネットワーク上での最適化など, 従来の OR の研究では見られなかった応用例も生まれており話題は尽きないが, それらについては他の機会に譲り, ここでは通信ネットワークへの応用について触れた。通信ネットワークはおそらく最も基本的なグラフ・ネットワーク最適化の応用先ではないかと思う。実際, 上で触れたネットワークフローや最小カットなど, 古典的な問題の中には通信ネットワークへの応用から発想されたのではないかと思うものも多い。一方で, 現実の通信技術も近年ますます複雑になっており, さまざまな新しいトピックが生まれている。

著者たちはどちらも, グラフ・ネットワーク最適化の理論に取り組んでいる研究者である。それと同時に, 2012年にスタートした JST ERATO 河原林巨大グラフプロジェクトの一環として, 通信ネットワークへの応用に向けた研究にも取り組んでいる。これは, それ

まで理論研究を通して得られた知識を現実の応用に役立てようという試みである。本稿ではこのプロジェクトから生まれた成果を二つ紹介する。

一つは, 円板形領域損傷モデルにおける最大流最小カット問題についてである。通信ネットワークや交通ネットワークは現代社会において必要不可欠なインフラであり, 自然災害や攻撃, 自然発生的な故障などに対して頑健なネットワークの構築が求められている。ネットワークの連結度は「何個のノード(もしくはリンク)が故障したときに, ネットワークが分断されるか」を表す値であり, 頑健性・耐故障性の自然な尺度として認識されている。古典的な最大流最小カット問題はこのような尺度に基づいた最適化問題であるが, 現実には, 光ファイバーネットワークや道路網のようなネットワークシステムが, 地震や津波などによって広範囲に損傷を受け, 同時に複数のノードやリンクの損傷が起こる場合も考えられる。このような状況を扱うため, 円板形領域損傷モデルというモデルが通信ネットワークの分野で提案された。Kobayashi and Otsuki [1] は円板形領域損傷モデルにおける最大流問題に対して多項式時間アルゴリズムを与え, さらにこのモデルにおいて最大流と最小カットの間に成り立つ関係式を示した。本稿の前半では, この Kobayashi and Otsuki の成果を紹介する。

二つ目はデータセンターにおける仮想マシンの配置問題についてである。近年, クラウドコンピューティングと呼ばれる考え方が広く普及し, さまざまなサービスがインターネットを介して提供されているが, このために巨大なデータセンターが運営されている。たとえば, Google が運営しているデータセンターでは 2010年の時点で約 90万台のサーバが使用されていたという推定がある [2]。そのような巨大なデータセンターを効率よく運用するためには, 仮想化と呼ばれる

こばやし ゆうすけ

筑波大学

〒 305-8573 茨城県つくば市天王台 1-1-1

kobayashi@sk.tsukuba.ac.jp

ふくなが たくろう

国立情報学研究所, JST ERATO 河原林巨大グラフプロジェクト

〒 101-8430 東京都千代田区一ツ橋 2-1-2

takuro@nii.ac.jp

技術が不可欠である。仮想化とは1台のサーバがあたかも複数台のサーバであるかのように振る舞うことを可能とする技術である。物理的に存在するサーバと仮想的に存在するサーバを区別するために、前者を物理マシン、後者を仮想マシンと呼ぶ。1台の物理マシンは複数台の仮想マシンを収容することができるが、あまり多くの仮想マシンを収容してしまうとCPUやストレージなどの計算資源が不足し、パフォーマンスの低下を招いてしまう。このため、仮想マシンの効果的な割り当ての研究がこの数年、通信ネットワークの分野で盛んに研究されている。本稿の後半では、この仮想マシン割り当て問題に関して Fukunaga et al. [3] によって得られた結果について紹介する。

## 2. 円板形領域損傷モデルにおける最大流最小カットアルゴリズム

### 2.1 問題設定と最大最小定理

まず、本節で扱う円板形領域損傷モデルを導入しよう。ネットワークを表す無向グラフ  $G = (V, E)$  が平面上に描かれているとする。ただし、 $V$  を頂点集合（ノードの集合）、 $E$  を枝集合（リンクの集合）とし、各枝は線分で表されているものとする。 $s, t \in V$  を相異なる2頂点とし、 $r_B$  を損傷を表す円板の半径とする。この円板はグラフとは無関係に平面上に配置されるものであり、円板に含まれる頂点や、円板を通る枝が、すべて損傷して利用できなくなる設定を考える。ただし、 $s, t$  を中心とした半径  $r_P$  の円を“保護領域”とし、円板の中心が保護領域内に入る位置には円板を配置できないものとする。すなわち、中心が保護領域に入らないような半径  $r_B$  の円板全体の集合を  $\mathcal{H}(r_B, r_P)$  と表すと、これが起こりうる損傷の集合となっている。このとき、Neumayer et al. [4] によって導入された円板形領域損傷モデルにおける最小カット問題（最小円板カット問題と呼ぶ）は以下の問題である。

#### 最小円板カット問題

入力： 平面上に描かれた無向グラフ  $G = (V, E)$ 、2点  $s, t \in V$ 、円板半径  $r_B$ 、保護領域半径  $r_P$  ( $> r_B$ )。

出力：  $s$  と  $t$  を分離する最小個数の円板 ( $\mathcal{H}(r_B, r_P)$  の要素) の集合。

なお、保護領域は  $s$  や  $t$  の周辺で損傷が起こる自明な最適解を除く目的で導入されたものであり、応用上は、輸送や通信の拠点である  $s$  や  $t$  の周辺では十分に頑健なネットワークが構築されている状況をモデル化したものだと解釈できる。この問題の最適値を MIN-CUT

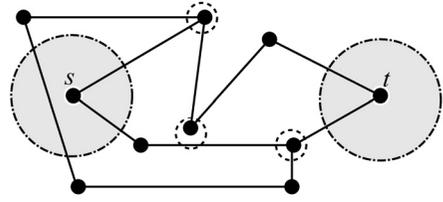


図1 最小円板カット問題と最大円板素パス集合問題の例

と表し、実行可能解を円板カットと呼ぶ。

最小円板カット問題の対にあたる問題として、円板形領域損傷モデルにおける最大流問題（最大円板素パス集合問題と呼ぶ）は以下のように定式化される [4]。

#### 最大円板素パス集合問題

入力： 平面上に描かれた無向グラフ  $G = (V, E)$ 、2点  $s, t \in V$ 、円板半径  $r_B$ 、保護領域半径  $r_P$  ( $> r_B$ )。

出力： どの2本も互いに円板素な最大本数の  $s$ - $t$  パスの集合。

ここで2本の  $s$ - $t$  パス  $P$  と  $P'$  が互いに円板素であるとは、 $P$  と  $P'$  の両方に損傷を引き起こす（すなわち両方のパスが通る）半径  $r_B$  の円板 ( $\mathcal{H}(r_B, r_P)$  の要素) が存在しないことを表す。最大円板素パス集合問題の最適値を MAX-FLOW と表し、実行可能解を円板素パス集合と呼ぶ。

例1. 図1のグラフについて考える。小さな破線で囲まれた円は半径  $r_B$  の円板を表しており、大きな一点鎖線で囲まれた円は半径  $r_P$  の保護領域を表している。この例において、MAX-FLOW = 1, MIN-CUT = 2 であることが容易に確認できる。

円板素パス集合  $P_1, P_2, \dots, P_k$  を任意に固定したときに、円板素の定義から、 $\mathcal{H}(r_B, r_P)$  の任意の円板は  $P_1, P_2, \dots, P_k$  のうち高々一つのパスとしか共有点をもたない。そのため、任意の円板カットのサイズはパスの数  $k$  以上となる。したがって、MAX-FLOW  $\leq$  MIN-CUT が成り立つことが示される。しかし、通常の最大流最小カット定理とは異なり、例1に見られるように、MAX-FLOW = MIN-CUT は必ずしも成立するとは限らない。Neumayer et al. [4] は、多くの場合において MAX-FLOW = MIN-CUT が成り立つことを実験的に確認し、任意の入力に対して MIN-CUT  $\leq$  MAX-FLOW + 1 が成り立つことを予想した。Kobayashi and Otsuki [1] はこの予想を肯定的に解決している。

定理1 ([1]). 最小円板カット問題の最適値 MIN-CUT

と最大円板素パス集合問題の最適値 MAX-FLOW に対して,

$$\text{MAX-FLOW} \leq \text{MIN-CUT} \leq \text{MAX-FLOW} + 1$$

が成り立つ.

なお, この関係式は Bienstock [5] によって与えられていた関係式  $\text{MIN-CUT} \leq 2 \cdot \text{MAX-FLOW} + 2$  を大幅に改善するものであり, 例 1 からわかるようにタイトな関係式となっている.

## 2.2 最大円板素パス集合問題の解法

本節では, Kobayashi and Otsuki [1] によって与えられた, 最大円板素パス集合問題に対する多項式時間アルゴリズムの概要について述べる.

提案アルゴリズムは, 1 本の  $s$ - $t$  パスから開始して, 互いに円板素な  $s$ - $t$  パスを 2 本, 3 本, ... と順に増やしていくものである. 以下では, すでに  $k-1$  本の互いに円板素な  $s$ - $t$  パス  $P_1, P_2, \dots, P_{k-1}$  が見つっていると仮定し, この仮定の下で, 互いに円板素な  $k$  本の  $s$ - $t$  パスが存在するならばそれを見つけ, 存在しないならば FALSE を返すアルゴリズムを与える.

まず,  $P_1, P_2, \dots, P_{k-1}$  はどの 2 本も交差することではなく, 各パスの  $s$  を始点としたときの 1 本目の枝は  $s$  の周りにこの順に時計回りに並んでいると仮定して一般性を失わない. このとき,  $i, j = 1, 2, \dots$  に対して,  $P_i$  を  $s$  から  $t$  の向きに辿り,  $P_j$  を  $t$  から  $s$  に辿ったときに時計回りに囲む領域を  $R(P_i, P_j)$  と表すものとする.  $P_k$  を領域  $R(P_{k-1}, P_1)$  に含まれる  $s$ - $t$  パスのうち,  $P_{k-1}$  と  $P_k$  が円板素<sup>1</sup>であるパスとする (たとえば  $P_k = P_1$  とすればこの条件を満たす). このとき,  $k$  本の  $s$ - $t$  パス  $P_1, P_2, \dots, P_k$  から開始し,  $k$  本のパスのうちの 1 本を新たなパスで置き換えていくことを繰り返すアルゴリズム (アルゴリズム 1) を考える. このアルゴリズムの動作例を表す図は [1] を参照されたい.

**定理 2** ([1]). アルゴリズム 1 を  $k = 2, 3, 4, \dots$  に順に適用することによって, 最大円板素パス集合問題を多項式時間で解くことができる.

この定理を示すために, [1] では MAX-FLOW の値を正確に特徴づける最大最小定理を与えており, 定理 1 もこの最大最小定理から導かれる.

アルゴリズム 1 の反復回数  $N$  は多くのケースで  $O(|V|)$  となり, その場合の最大円板素パス集合

問題に対するアルゴリズム全体の計算量は  $O(|V|^2 \cdot \text{MAX-FLOW})$  である<sup>2</sup>. [1] では, 最大円板素パス集合問題や最小円板カット問題に対する提案アルゴリズムを人工的に生成したグラフに対して適用し, それらの性能を評価している. その後, アルゴリズムの高速化が研究されており, 現在では頂点数が 10 万を超えるような現実的なネットワークへの適用が可能となっている [6].

---

### アルゴリズム 1 $k$ 円板素パスアルゴリズム

---

入力: 互いに円板素な  $k-1$  本の  $s$ - $t$  パス  $P_1, \dots, P_{k-1}$ , および,  $R(P_{k-1}, P_1)$  に含まれ,  $P_{k-1}$  と円板素になるような  $s$ - $t$  パス  $P_k$ .

出力: (もし存在すれば) 互いに円板素な  $k$  本の  $s$ - $t$  パス.

- 1: **for**  $l = k, k+1, \dots, k+N$  **do**
  - 2:   **if**  $P_l$  と  $P_{l-k+1}$  が円板素 **then**
  - 3:     **return**  $P_{l-k+1}, \dots, P_{l-1}, P_l$  (これらは互いに円板素)
  - 4:   **else**
  - 5:      $P_{l+1}$  を,  $R(P_{l-k+1}, P_{l-k+2})$  に含まれ,  $P_l$  と円板素になるような  $s$ - $t$  パスのうち,  $R(P_{l+1}, P_{l-k+2})$  の領域が最大となるパスとする.
  - 6:   **end if**
  - 7: **end for**
  - 8: **return** FALSE
- 

## 3. データセンターネットワークでの仮想マシン配置問題

### 3.1 モデル

1 節でも触れたとおり, 仮想マシンの効果的な割り当ての研究はこの数年, 通信ネットワーク分野で盛んに研究されている. Meng et al. [7] は実在のデータセンター (のような) システムのトラフィックを調べ, 仮想マシンの配置を工夫することでネットワークのスケラビリティが改善されることを実験により確かめた. 彼らの報告をきっかけに仮想マシン割り当てに関するさまざまなモデルが多くの研究者によって研究されている. しかしながら, それらのモデルのほとんどはデータセンターのネットワーク構造を考慮に入れて

<sup>1</sup> 厳密な議論のために, [1] では円板素の代わりに“時計回りに円板素”という概念を用いている.

<sup>2</sup>  $N$  は入力に応じて定まる値であり, その正確な定義は省略する.

いない。われわれの知る限り、唯一の例外は Cohen et al. [8] による研究である。彼らはネットワークの帯域に関する制約の下、指定されたノードに仮想マシンを配置する問題を提案し、いくつかの近似アルゴリズムを与えた。それに対して、Fukunaga et al. [3] は仮想マシン間の接続コストを考慮に入れた割り当て問題をモデル化し、近似アルゴリズムを与えた。まず、[3] で導入されたモデルを紹介する。

頂点集合  $V$ 、枝集合  $E$  からなる無向グラフを  $G = (V, E)$  で表し、各枝  $e$  には長さ  $l(e)$  が与えられているとする。物理マシンに対応する頂点のことを端点と呼び、端点の集合を  $T$  とする。各端点  $t \in T$  には容量  $c(t)$  が与えられており、端点  $t$  に対応する物理マシンは  $c(t)$  台の仮想マシンを収容できるとする。

いま、 $m$  人のクライアントがあり、クライアント  $i$  が  $r_i$  台の仮想マシンを要求する状況を考える。求めたいのは、これらの仮想マシンの物理マシンへの割り当てである。割り当てを関数  $\phi: T \times [m] \rightarrow \mathbb{Z}_+$  で表す。ただし、 $[m]$  は  $\{1, \dots, m\}$ 、 $\mathbb{Z}_+$  は非負の整数の集合である。この関数は、クライアント  $i$  の要求した仮想マシンを端点  $t$  に対応する物理マシンに  $\phi(t, i)$  台配置するというを意味しており、次の条件を満たすものとする。

- 各  $t \in T$  について、 $\sum_{i=1}^m \phi(t, i) \leq c(t)$ 、
- 各  $i \in [m]$  について、 $\sum_{t \in T} \phi(t, i) = r_i$ 。

同じクライアントから要求された仮想マシン間ではデータのやりとりのために頻繁に通信が発生するので、できれば同じ物理マシンに、もし無理でもなるべく近い物理マシンに配置したい。このような目的を反映させるため、クライアント  $i$  の仮想マシンを収容する物理マシンを張る  $G$  の連結部分グラフ  $S_i$  を同時に考え、枝長  $l(S_i) := \sum_{e \in S_i} l(e)$  の総和を最小化することを目的関数として定義する。

また、仮想マシンをコントロールするための管理マシンが頂点  $s$  に存在するとし、 $S_i$  は  $s$  も連結しなければならないモデルも考える。 $s$  が与えられる場合を集中管理モデルと呼び、与えられない場合を分散モデルと呼ぶことにする。集中管理モデルを簡潔にまとめると以下ようになる。

#### 仮想マシン配置問題 (集中管理モデル)

入力： 無向グラフ  $G = (V, E)$ 、枝長  $l: E \rightarrow \mathbb{R}_+$  (ただし、 $\mathbb{R}_+$  は非負実数の集合)、端点集合  $T \subseteq V$ 、容量関数  $c: T \rightarrow \mathbb{Z}_+$ 、頂点  $s \in V \setminus T$ 、要求  $r_1, \dots, r_m \in \mathbb{Z}_+$ 。

出力：  $G$  の連結部分グラフ  $S_1, \dots, S_m$  と割り当て

$\phi: T \times [m] \rightarrow \mathbb{Z}_+$  のうち、 $\sum_{i=1}^m l(S_i)$  を最小化するもの。ただし、各  $i \in [m]$  について、 $S_i$  は  $\phi(t, i) > 0$  であるようなすべての端点  $t$  と点  $s$  を張る。

図 2 は集中管理モデルの解の一例を表している。図中で黒い点で表されているのが頂点  $s$  である。クライアントの数は 3 であり、それぞれの要求は  $r_1 = 3$ 、 $r_2 = 6$ 、 $r_3 = 4$  の場合を考えている。

この問題はどちらのモデルであっても、 $m = 1$  で各端点の容量が 1 であるとき、シュタイナー木問題と呼ばれる問題と一致する。シュタイナー木問題は NP 困難であることが知られているので、仮想マシン配置問題も NP 困難である。このため、Fukunaga et al. [3] は、いくつかの近似アルゴリズムとヒューリスティックに基づくアルゴリズムを与えた。次の節では、そのうちの一つである、集中管理モデルで要求がすべて同じ値である場合に対する 12 近似アルゴリズムのアイデアを紹介する。12 近似アルゴリズムとは、多項式時間で実行可能解を計算し、その目的関数値は必ず最適値の 12 倍以下であることが保証できるようなアルゴリズムのことである。12 倍という保証が応用上はたして意味があるのか疑問にもたれる方もいるだろうが、これはあくまで最悪の場合の保証であり、多くの場合に保証よりもよい解をアルゴリズムは出力する。[3] では数値実験を通してアルゴリズムの実際の性能についても検証している。

### 3.2 アルゴリズム

この節では、モデルは集中管理モデルとし、要求がすべて同じ値  $r \in \mathbb{Z}_+$  を取る場合について考える。 $G$  の連結部分グラフ  $S$  に対し、 $S$  によって張られている端点の容量の和を  $c(S)$  で表す。 $k$  を正整数としたとき、 $k$ -シュタイナー木とは、 $s$  を張り、かつ  $c(S) \geq k$  であるような木  $S$  のことであるとする。長さ最小の  $k$ -シュタイナー木を求める問題には 2 近似アルゴリズムが知られている [9]。

まずはじめに、簡単なアルゴリズムを導入しよう。 $k = rm$  とし、枝長が最小となるような  $k$ -シュタイナー木  $S$  を計算する。次に、木の部分  $S_1, \dots, S_m$  を構築する。このとき、次のような条件を満たす部分木が必ず存在し、多項式時間で計算することができる。

- $\sum_{i=1}^m l(S_i) \leq l(S)$ 、
- 仮想マシン配置問題の制約を満たすような割り当て  $\phi$  が存在する。

得られた部分木のいくつかは  $s$  につながっていないので、そのような部分木に  $s$  までの最短路を加えること

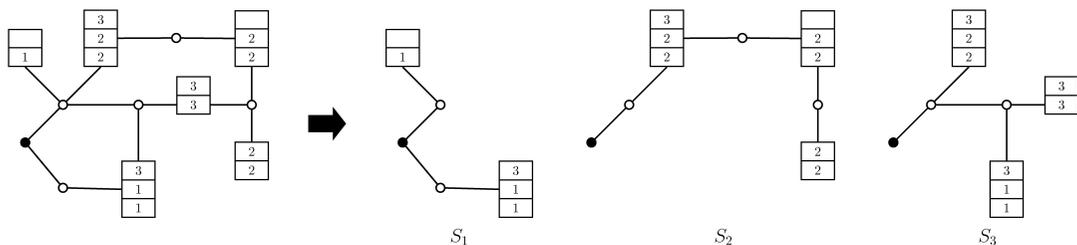


図2 集中管理モデルの問題と解の例

で、すべての木が  $s$  と連結しているように修正する。  
 このアルゴリズムにより、問題の実行可能解は構築することができる。ただし、得られた解の目的関数値が最適解と比べて非常に大きくなってしまふ場合がある。実際、このアルゴリズムの近似比が  $\Omega(|V|)$  になってしまうような例が存在する。このような現象は、最初のステップで計算された  $k$ -シュタイナー木が  $s$  から離れた端点を含んでしまい、最後のステップで付け加える最短路の長さが大きくなってしまふことによって起きる。

われわれのアイデアは上記のアルゴリズムの最初のステップを改良することである。各端点に、どれぐらい  $s$  から離れているかに応じてコストを定義する。そのうえで、アルゴリズムの最初のステップで計算する  $k$ -シュタイナー木として、枝長だけでなく端点のコストも最小化するものを選ぶ。ここで、そのような木を計算する問題を端点コスト  $k$ -シュタイナー木問題と呼ぶことにする。詳しい定義は以下のとおりである。

#### 端点コスト $k$ -シュタイナー木問題

入力： 無向グラフ  $G = (V, E)$ 、端点集合  $T \subseteq V$ 、頂点  $s \in V \setminus T$ 、容量関数  $c: T \rightarrow \mathbb{Z}_+$ 、枝長  $l: E \rightarrow \mathbb{R}_+$ 、端点コスト  $w: T \rightarrow \mathbb{R}_+$ 。

出力：  $k$ -シュタイナー木  $S$  と  $S$  によって張られる端点からなる集合  $U$  のうち、 $\sum_{t \in U} c(t) \geq k$  を満たし、 $l(S) + \sum_{t \in U} w(t)$  を最小化するもの。

この問題は長さ最小の  $k$ -シュタイナー木を求める問題（端点コストなし）に近似比を保存したまま帰着することができる。つまり、[9]の2近似アルゴリズムを用いれば、端点コスト  $k$ -シュタイナー木問題の2近似解も計算できる。

さて、残った課題は各端点の重みを具体的にはどのように定義したらよいかという点である。そのために、まずすべての端点  $t$  に対して  $c(t) \leq r$  が成り立つと仮定する。もしこの条件がある端点  $t$  で成り立たない場合は、新しい端点  $t_1, \dots, t_{\lceil c(t)/r \rceil}$  を追加し、各  $t_i$  を

長さ 0 の辺で  $t$  と結ぶ。  $t$  を端点集合から取り除き、 $t_1, \dots, t_{\lceil c(t)/r \rceil}$  の容量を  $\sum_{i=1}^{\lceil c(t)/r \rceil} c(t_i) = c(t)$  となるよう適切に定義すればよい。一般性を失うことなく  $c(t) \leq rm$  であるので、この操作の後でもグラフのサイズは高々  $m$  倍されるだけである。また、端点  $t$  と  $s$  の最短路の長さを  $l(s, t)$  と書く。そのうえで、端点  $t$  のコスト  $w(t)$  を  $l(s, t)c(t)/r$  と定義すればよいというのが、われわれの結論である。詳細は述べないが、このように設定した端点コストに対して端点コスト  $k$ -シュタイナー木問題を解き、得られた木を長さ最小  $k$ -シュタイナー木の代わりに用いれば、12 近似アルゴリズムが得られる。

#### 4. まとめ

本稿では、グラフ・ネットワーク最適化の理論の通信ネットワークへの応用例として、二つの研究成果を紹介した。これらはどちらも、通信ネットワークの分野ですで行われていた研究に着想を得て始めた研究である。これらが本当に応用上役に立つかはさまざまな意見があるだろう。この点も含めて、これからも理論と応用の双方向のフィードバックを大事にして研究を進めていきたい。

#### 参考文献

- [1] Y. Kobayashi and K. Otsuki, "Max-flow min-cut theorem and faster algorithms in a circular disk failure model," In *Proceedings of the IEEE INFOCOM 2014*, pp. 1635–1643, 2014.
- [2] J. Koomey, *Growth in Data Center Electricity Use 2005 to 2010*, Analytics Press, 2011.
- [3] T. Fukunaga, S. Hirahara and H. Yoshikawa, "Virtual machine placement for minimizing connection cost in data center networks," In *Proceedings of the International Workshop of Software-Defined Data Communications and Storage*, to appear.
- [4] S. Neumayer, A. Efrat and E. Modiano, "Geographic max-flow and min-cut under a circular disk failure model," In *Proceedings of the IEEE INFOCOM 2012*, pp. 2736–2740, 2012.
- [5] D. Bienstock, "Some generalized max-flow min-cut

problems in the plane,” *Mathematics of Operations Research*, **16**, pp. 310–333, 1991.

- [6] 大槻兼資, 小林佑輔, 室田一雄, “円板形領域損傷モデルにおける最大流最小カットアルゴリズムの実験的評価,” 日本オペレーションズ・リサーチ学会春季研究発表会アブストラクト集, pp. 126–127, 2014.
- [7] X. Meng, V. Pappas and L. Zhang, “Improving the scalability of data center networks with traffic-aware virtual machine placement,” In *Proceedings of*

*the IEEE INFOCOM 2010*, pp. 1–9, 2010.

- [8] R. Cohen, L. Lewin-Eytan, J. Naor and D. Raz, “Almost optimal virtual machine placement for traffic intense data centers,” In *Proceedings of the IEEE INFOCOM 2013*, pp. 355–359, 2013.
- [9] N. Garg, “Saving an epsilon: A 2-approximation for the  $k$ -MST problem in graphs,” In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pp. 396–402, 2005.