

サービスを考慮した施設配置問題に対する 近似アルゴリズムの実験評価

中央大学大学院 理工学研究科 情報工学専攻

中塩 英良 浅野 孝夫

概要

施設配置問題の1つとして、2004年に Shmoys, Swamy and Levi で提唱されたサービスを考慮した施設配置問題がある。本稿ではこの問題に対して、容量制約なし施設配置問題に帰着させることによって得られる手法、容量制約なし施設配置問題において現在もっとも近似率の良い1.52近似に基づく手法と、1.61近似に基づく手法を提案する。そして、Shmoys et al. によって提案された6近似アルゴリズムの実験的性能評価を行い、提案手法が高い性能を持つことを示す。また、この問題をより現実に近い“容量制約つき”の場合に発展させ、容量制約つきサービスを考慮した施設配置問題(ソフト容量版)に対する解法も提案する。

Experimental Evaluation of Approximation Algorithms for the Facility Location Problem with Service Installation Costs

Information and System Engineering Course,
Graduate School of Science and Engineering, Chuo University

Hidekazu Nakashio Takao Asano

Abstract

We consider a generalization of the uncapacitated facility location problem, the facility location problem with service installation costs which Shmoys, Swamy and Levi proposed in 2004. For this problem, we propose a heuristic for reducing it to the uncapacitated facility location problem and heuristics based on 1.61-, 1.52-approximation algorithms for the uncapacitated facility location problem. By evaluating the experimental performance of 6-approximation algorithm proposed by Shmoys et al. (2004) and ours, we show the availability of our heuristics. We also apply the soft-capacitated facility location problem with service installation costs, apply our heuristics for the soft-capacitated facility location problem with service installation costs.

1 序論

施設配置問題は1960年代前半からオペレーションズ・リサーチ(OR)の分野で中心的な問題として幅広く研究されている問題であり、工場、倉庫、学校、病院などの設立地計画をモデル化した問題といえる。応用問題としては、データやキャッシュなどのコンテンツ配置問題、ネットワークの安全性(耐故障性)を考慮した様々なネットワーク設計問題が挙

げられる。そのため、近年のインターネット普及に伴い、ウェブのプロキシサーバーの設置計画などにも応用されており、より現実的な環境に対応させた様々な関連問題も広く研究されている。

本稿では、施設配置問題の1つである**サービスを考慮した施設配置問題**(Facility Location Problem with Service Installation Costs)についての研究を述べる[4]。この問題の応用として、倉庫と小売店間

の在庫管理やインターネット問題が挙げられる。目的は、この問題に対して Shmoys et al. によって発表された 6 近似アルゴリズム [4] よりさらに性能の良いアルゴリズムを開発する事である。具体的には、**容量制約なし施設配置問題** (Uncapacitated Facility Location Problem) に帰着させることによって得られる手法と、容量制約なし施設配置問題において現在もっとも近似比率の良い 1.52 近似 [2], 1.61 近似アルゴリズム [1] の考え方に基づく手法を提案する。

2 問題定義

最も有名な施設配置問題は容量制約なし施設配置問題と呼ばれるものである。これは、入力として n_f 個の開設候補施設の集合 F と n_c 人の利用者集合 C が与えられる。施設 i には施設を開設するための非負の開設コスト f_i が与えられ、施設 i と利用者 j の間には非負の接続コスト c_{ij} が与えられる。なお、接続コストは三角不等式を満たすものとする。この入力において、いくつかの施設を開設し、すべての利用者を適切な施設に接続することを考える。

これに対して、サービスを考慮した施設配置問題では、入力として n_s 個のサービス集合 S が加えられ、各利用者にはそれぞれが必要とするサービス $g(j)$ が決められている。また、各施設 i にサービス l をインストールするためのサービスインストールコスト f_i^l が与えられる。このとき本問題の目的は、総コスト (開設コストとサービスインストールコストと接続コストの合計) を最小とすることである。すなわち、いくつかの施設を開設し、開設した施設に対してインストールするサービスを決定し、すべての利用者を必要としているサービスがインストールされている施設に接続させることである。

この問題は、容量制約なし施設配置問題と同様、以下のように整数計画問題として定式化することができる。

$$\begin{aligned} \min \quad & \sum_{i \in F} f_i y_i + \sum_{i \in F} \sum_{l \in S} f_i^l y_i^l + \sum_{i \in F} \sum_{j \in C} c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{i \in F} x_{ij} = 1 \quad (j \in C), \\ & x_{ij} \leq y_i^{g(j)} \quad (i \in F, j \in C), \\ & x_{ij} \leq y_i \quad (i \in F, j \in C), \\ & x_{ij}, y_i, y_i^l \in \{0, 1\} \quad (i \in F, j \in C, l \in S). \end{aligned}$$

3 Shmoys et al. の手法

ここでは、Shmoys, Swamy and Levi が提案した、6 近似アルゴリズム [4] について述べる。このアルゴリズムは**施設の優先順位** O と呼ばれる仮定の下で、6 近似を達成するアルゴリズムである。まず、施設の優先順位について述べることにする。

施設の優先順位 O

施設 i と i' を比べたとき、すべてのサービス $l \in S$ に対して、

$$f_i^l \leq f_{i'}^l$$

が成り立つならば、施設の優先順位は $i \leq i'$ である。

次にアルゴリズムの概観を説明する。6 近似アルゴリズムはフェイズ 1 とフェイズ 2 から構成される。フェイズ 1 では実行可能な双対問題の解と一時的に開設する (仮開設) 施設集合、一時的にインストールする (仮インストール) 集合を求める。フェイズ 2 ではフェイズ 1 の仮開設施設、仮インストール集合から正式に開設、インストールするものを決定し、すべての利用者を必要とするサービスがインストールされている施設に接続する。

アルゴリズム

フェイズ 1

各利用者 j に施設 i への費用 α_j を設ける。費用 α_j は、施設 i に接続するための費用、 $g(j)$ をインストールするための費用 θ_{ij} 、施設 i を開設させるための費用 β_{ij} からなるものとする。

すべての利用者を未凍結の状態にする。3つの条件 (1. 利用者を接続, 2. $g(j)$ をインストール, 3. 施設を開設) が成立したら凍結する。未凍結の利用者の費用 α_j をすべての割合で増加させながら以下の (a), (b), (c) を繰り返す。すべての利用者が凍結したら終了とする。なお、(a), (b), (c) の条件が同時に成立する場合は、任意の順番で行なう。

- (a) **if** (i, j に対して, $\alpha_j \geq c_{ij}$) **then**
 利用者 j を施設 i へ仮接続する。
- **if** ($y_i^{g(j)} = 0$) **then**
 θ_{ij} を増やし始める。
 - **if** ($y_i^{g(j)} = 1$ かつ $y_i = 0$) **then**
 β_{ij} を増やし始める。

- **if** ($y_i^{g(j)} = 1$ かつ $y_i = 1$) **then**
利用者 j を凍結する.
- (b) **if** (i, l に対して, $\sum_{j \in G_l} \theta_{ij} = f_i^l$) **then**
施設 i にサービス l を仮インストールする.
- **if** ($y_i = 0$) **then**
 β_{ij} を増やし始める.
 - **if** ($y_i = 1$) **then**
利用者 j を凍結する.
- (c) **if** (i に対して, $\sum_j \beta_{ij} = f_i$) **then**
施設 i を仮開設し, 3 つ条件が揃っている利用者 j を凍結する.
- $\alpha_j, \theta_{ij}, \beta_{ij}$ は未凍結な利用者にものみ発生し, 凍結した利用者は各操作に参加しない.

フェイズ 2

フェイズ 2 では正式に開設する施設, インストールするサービス, 利用者の接続を決定する. ここで, F は仮開設した施設の集合を表し, $F_l \subseteq F$ はサービス l が仮インストールされた仮開設施設集合を表す. 施設 $i \in F$ に対して, 仮開設した時刻を t_i とする. また施設 $i \in F_l$ に対して, 施設 i にサービス l がインストールされた時刻を t_{il} とする.

開設施設の決定

仮開設施設集合 F から正式に開設施設集合を決定する. もし利用者 j において, β_{ij} と $\beta_{i'j}$ の両方が活性 ($\beta_{ij}, \beta_{i'j} > 0$) であるとき, 施設 $i, i' \in F$ は**従属施設**と呼ばれる. このとき, 施設の優先順位 O をもとに, 仮開設した施設を考え, 極大独立集合 $F' \subset F$ を選ぶ. 集合 F' を開設施設集合とする.

サービスインストールの決定

サービス l と集合 F_l を考える. もし利用者 $j \in G_l$ において, θ_{ij} と $\theta_{i'j}$ の両方が活性 ($\theta_{ij}, \theta_{i'j} > 0$) であるとき, 施設 $i, i' \in F_l$ は**サービス- l -従属**と呼ばれる. このとき, 特定の順序によって集合 F_l から極大独立集合 F_l' を選ぶ. ここでいう特定の順序とは,

- (1) 開設施設集合 $F_l \cap F'$ から t_{il} の早い順に考える.
 - (2) 開設されていない施設 $F_l - F'$ から t_i の早い順に考える.
- である. 具体的には, まず特定の順序 (1) に該当

される施設を調べていき, すべての施設にサービス l をインストールする (極大独立集合 F_l' に加える). その後, 特定の順序 (2) に該当する施設を調べていき, 施設の優先順位 O をもとに, 施設 i と施設 i' が従属施設であるような施設 i' を見つけ, この施設 i' にサービス l をインストールする. この施設 i' は施設 i の隣接施設であり, $\mathbf{nbr}(i)$ と表す. ここで注意すべきことは, $\mathbf{nbr}(i)$ はサービス l ではなく施設 i でのみ決定されるという点である.

接続

すべての利用者 j を, 必要としているサービス $g(j)$ がインストールされているもっとも近い開設施設 i に接続する.

4 提案手法

この手法は容量制約なし施設配置問題に帰着させることによって解を得るものである. ポイントは与えられた入力に対して 1 つのサービスに絞って考えると, 容量制約なし施設配置問題と等価となる点である (図 1 参照). 本稿では現在もっとも近似率の良い 1.52 近似アルゴリズムを用いている. 具体的な手順を以下に示す.

1. 与えられた入力に対して, サービス l を必要としている利用者にも注目し, 新たに開設コストを $f_i + f_i^l$ とする. これにより, サービスの数 n_s だけ容量制約なし施設配置問題の入力を構築する.
2. 上記の入力に対して, 容量制約なし施設配置問題に対するアルゴリズムを適用する.
3. 得られた n_s 個の解を足し合わせ, 本問題の解とする.

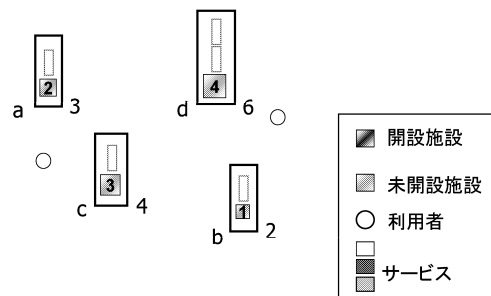


図 1 帰着の様子

5 1.61 近似に基づく提案手法

1.61 近似に基づく提案手法は Jain, Mahdian and Saberi の手法 [1] を本問題に拡張したものである。

アルゴリズムの流れとしては, Shmoys et al. の手法のフェイズ 1 と類似しているが, 利用者が凍結したら, 凍結する要因となった施設以外の施設への貢献を無効にし, 未接続の状態に戻す点が異なる. この操作によってアルゴリズムの終了時には, 利用者は凍結する要因となった施設にのみ接続している. さらに, 1.61 近似に基づく提案手法の特徴として重要な点は, 凍結した利用者に対しても総コストが減少するならば接続を変更することである.

アルゴリズムの詳細を述べる前に, 1.61 近似に基づく提案手法で用いられる, 利用者 j による施設 i への貢献 β_{ij} を定義する.

貢献 β_{ij} の定義

- **if** (利用者 j が未凍結) **then**
 $\beta_{ij} = \max\{\alpha_j - c_{ij} - \theta_{ij}, 0\}$ とする.
- **if** (利用者 j が凍結) **then**
 $\beta_{ij} = \max\{c'_{ij} - c_{ij}, 0\}$ とする.

なお, 接続を変更することで, 施設 i' にインストールされているサービスが必要なくなったらアンインストールする. また, 接続している利用者がいなくなったら, 施設 i' は未開設に戻すことにする.

アルゴリズム

各利用者 j に施設 i への費用 α_j を設ける. 費用 α_j は, 施設 i に接続するための費用, $g(j)$ をインストールするための費用 θ_{ij} , 施設 i を開設させるための費用 β_{ij} からなるものとする. 以下詳しい手順を述べる.

すべての利用者を未凍結の状態とする. 3つの条件 (1. 利用者を接続, 2. $g(j)$ をインストール, 3. 施設を開設) が成立したら凍結する. 未凍結の利用者の費用 α_j をすべての割合で増加させながら以下の (a), (b), (c) を繰り返す. すべての利用者が凍結したら終了とする.

- (a) **if** (i, j に対して, $\alpha_j \geq c_{ij}$) **then**
 利用者 j を施設 i に接続する.

- **if** ($y_i^{g(j)} = 0$) **then**
 θ_{ij} を増やし始める.
 - **if** ($y_i^{g(j)} = 1$ かつ $y_i = 0$) **then**
 β_{ij} を増やし始める.
 - **if** ($y_i^{g(j)} = 1$ かつ $y_i = 1$) **then**
 利用者 j を凍結する.
- (b) **if** (i, l に対して, $\sum_j \theta_{ij} = f_i^l$) **then**
 施設 i にサービス l をインストールする.
- **if** ($y_i = 0$) **then**
 β_{ij} を増やし始める.
 - **if** ($y_i = 1$) **then**
 利用者 j を凍結する.
- (c) **if** (i に対して, $\sum_j \beta_{ij} = f_i$) **then**
 施設 i を開設し, 3つ条件が揃っている利用者 j を凍結する.
- **if** (j が既に i' に接続 かつ $c_{ij} < c'_{ij}$) **then**
 利用者 j の接続を i' から i に変更し, 必要なければ施設 i' にインストールされているサービス $g(j)$ をアンインストールする.

6 1.52 近似に基づく提案手法

1.52 近似に基づく提案手法は Mahdian, Ye and Zhang の手法 [2] を本問題に拡張したものである. このアルゴリズムにおいて, 総コストが減少する値として $gain$ を用いる.

$gain(i, l)$ の定義

未開設施設 i を開設し, さらにその施設にサービス l をインストールして, 利用者を必要とするサービスがインストールされている最も近い開設施設に接続する. このとき, 接続コストが c から c' になるとすると,

$$gain(i, l) = c - c' - f_i - f_i^l$$

と定義する.

(初期解) スケーリングパラメータを $\delta = 1.504$ とする. このとき, 開設コストを一律に δ 倍した問題に対して, 1.61 近似に基づく提案手法を適用する.

(貪欲改善) 開設コストをもとに戻した問題に対して, 総コストが減少する ($gain(i, l) > 0$ である) ぎり以下を繰り返す.

- **if** (ある未開設施設 i , ある未インストールサービス l の $\frac{gain(i,l)}{f_i}$ が正かつ最大) **then**
 施設 i を開設し, サービス l をインストールして, 利用者は必要とするサービスがインストールされている最も近い開設施設に接続を変更する.

間のランダムな正整数
 (6) 施設と利用者の接続コストは, 施設と利用者のユークリッド距離

7 計算機実験 1

7.1 実験の方法

今回の実験では,

- A. Shmoys et al. が提案する 6 近似アルゴリズム
- B. 容量制約なし施設配置問題に帰着させる提案手法 (以下, 提案手法と呼ぶ)
- C. 1.61 近似に基づく提案手法 (以下, 1.61 近似ベースとも呼ぶ)
- D. 1.52 近似に基づく提案手法 (以下, 1.52 近似ベースとも呼ぶ)

の 4 手法を Visual C++ を用いて実装し, 実際的な性能評価を行なう. 最適解としては, IP を線形計画ソフト NUOPT で解くことで得られる整数解を用いることにする. 実験で使用した計算機を表 1 に示す. 入力には次の入力データを用いこの入力データの一部を変化することで様々な実験を行なう.

表 1 実験環境

CPU	Intel Pentium4 2.8 GHz
メモリ	512 MB RDRAM
OS	Microsoft windows XP
コンパイラ	Visual C++ 6.0

標準入力データ 1

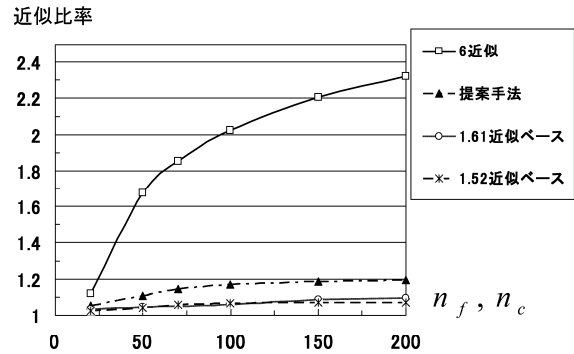
- (1) 施設は, 整数格子点の座標 (0, 0) から (500, 500) の間でランダムに 100 個ずつ発生
- (2) 利用者は, 整数格子点の座標 (0, 0) から (500, 500) の間でランダムに 100 個ずつ発生
- (3) サービスは 50 種類ずつ発生させ, 利用者が必要とするサービスはランダムに決定
- (4) 開設コストは, 1 から 1000 の間のランダムな正整数
- (5) サービスインストールコストは, 1 から 300 の

図と表の説明

図は, 実験を 20 回行ったときの平均の近似比率, 以下では特に明記しないかぎり簡略化して近似比率と呼ぶ, を表すもので, 縦軸にはこの近似比率を用いている.

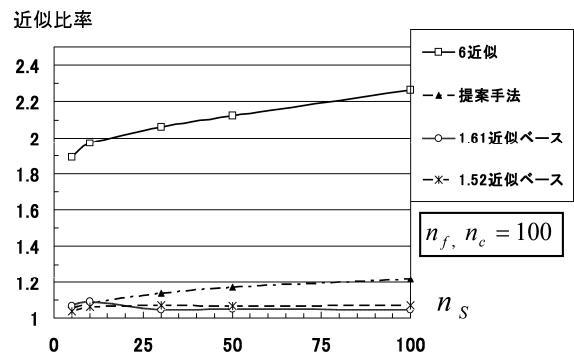
実験 1 施設数 n_f , 利用者数 n_c を変化した場合

標準入力データの施設数 n_f , 利用者数 n_c をそれぞれ同じ割合で増加した場合の計算機実験を行う. ここでは, 線形計画ソフト NUOPT で IP を解くことのできる $n_f = n_c = 200$ までの計算機実験を行う.



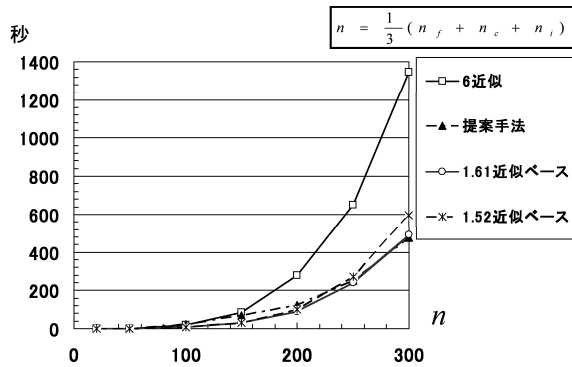
実験 2 サービス数 n_s を変化した場合

標準入力データのサービス数 n_s を増加した場合の計算機実験を行う.



実験 3 計算時間 ($n = \frac{1}{3}(n_f + n_c + n_s)$)

入力のサイズを n とし, 表 1 に示した計算機を使用したときの計算時間を算出する.



7.2 計算機実験1のまとめ

実験1より、提案する3つの手法は、Shmoysらによって提案された6近似アルゴリズムより、かなり良い値を出力することがわかる。特に1.61近似、1.52近似に基づく提案手法はもっとも良い値を出力している。これは6近似アルゴリズムは開設する施設数が極端に少なく、逆に提案手法は開設する施設数が多いので、余分なコストがかかってしまっているためと考えられる。

実験2より、サービス数 n_s が少ない場合、提案手法、1.61近似、1.52近似に基づく手法の間に差がないことがわかる。これは、提案手法はサービスの数だけ1.52近似アルゴリズムを再帰的に実行するので、サービスの数が少なければ、それらを足し合わせた時に生じる誤差が少なくなるためと考えられる。

実験3より、提案する3つの手法は計算時間においても短縮できていることがわかる。したがって、提案する3つの手法は近似比率と計算時間の両面で有効な手法であるといえる。

8 容量制約つきサービスを考慮した施設配置問題 (ソフト容量版)

サービスを考慮した施設配置問題では、施設にインストールできるサービスの容量は設けていない。そこで、“容量制約つき”の場合に発展させた、容量制約つきサービスを考慮した施設配置問題 (ソフト容量版) を提唱し、この問題に対する解法を提案する。

8.1 問題定義

この問題は、サービスを考慮した施設配置問題の入力において、各施設 i に整数 u_i が付随している、

基本的に施設 i は u_i 個のサービスしかインストールできないというものである。ソフト容量版では、各施設が (コピーして) 何回でも開設可能である。

問題の目的は、容量制約を守りつつ、総コスト (開設コストとサービスインストールコストと接続コストの合計) を最小とすることである。すなわち、いくつかの施設を複数回開設し、開設した施設に対して容量制約を守りつつインストールするサービスを決定し、すべての利用者を必要としているサービスがインストールされている施設に接続させることである。この問題も、以下のように整数計画問題 (IP) として定式化することができる。

$$\begin{aligned}
 \min \quad & \sum_{i \in F} f_i y_i + \sum_{i \in F} \sum_{l \in S} f_i^l y_i^l + \sum_{i \in F} \sum_{j \in C} c_{ij} x_{ij} \\
 \text{s.t.} \quad & \sum_{i \in F} x_{ij} = 1 \quad (j \in C), \\
 & x_{ij} \leq y_i^{g(j)} \quad (i \in F, j \in C), \\
 & x_{ij} \leq y_i \quad (i \in F, j \in C), \\
 & \sum_{l \in S} y_i^l \leq u_i y_i \quad (i \in F), \\
 & x_{ij}, y_i^l \in \{0, 1\} \quad (i \in F, j \in C, l \in S), \\
 & y_i \text{ は非負の整数} \quad (i \in F).
 \end{aligned} \tag{1}$$

サービスを考慮した施設配置問題との違いは、制約式 (1) が追加されたことである。この制約式は、施設 i が k 個のサービスをインストールするとき、施設を $\lceil k/u_i \rceil$ 回開設しなければならないことを表している。

8.2 提案手法1

ここでは、容量制約つきサービスを考慮した施設配置問題に対する解法を提案する。この提案手法1はMahdian, Ye and Zhang [2] によって提案された2.89近似アルゴリズムの考え方も基にしている。具体的には、サービスを考慮した施設配置問題に帰着させ、サービスを考慮した施設配置問題に対するアルゴリズムを適用するというものである。

1. 施設 i を開設するための開設コストを $(1-\lambda)f_i$ 、施設 i にサービス l をインストールするためのサービスインストールコストを $f_i^l + \lambda_i \cdot \frac{f_i}{u_i}$ とするサービスを考慮した施設配置問題の入力を構築する。
2. 1で作られた入力において、スケーリングパラ

メータを γ_c/γ_f とする. このとき, 開設コストを一様に γ_c/γ_f 倍する.

3. 上記の入力に対して, サービスを考慮した施設配置問題に対するアルゴリズムを適用する.
4. 結果を, 各施設 $i \in F$ に対して $\lceil \sum_{l \in S} y_l^i / u_i \rceil$ 回施設を開設した容量制約つきサービスを考慮した施設配置問題の解として出力する.

8.3 提案手法 2

この節では, Mahdian, Ye and Zhang [3] によって提案された 2 近似アルゴリズムの考え方に基づく提案手法 2 を述べる.

1. 施設 i を開設するための開設コストを

$$f_i(k) = \begin{cases} 0 & (k = 0) \\ (1 + \frac{k-1}{u_i})f_i & (k \geq 1) \end{cases}$$

とし, サービスを考慮した施設配置問題の入力を構築する. ここでいう k とは施設にインストールするサービス数を指す.

2. 上記の入力に対して, サービスを考慮した施設配置問題に対するアルゴリズムを適用する.
3. 結果を, 各施設 $i \in F$ に対して $\lceil \sum_{l \in S} y_l^i / u_i \rceil$ 回施設を開設した容量制約つきサービスを考慮した施設配置問題の解として出力する.

9 計算機実験 2

9.1 実験の方法

今回の実験では,

- A. 1.61 近似に基づく手法を適用した提案手法 1
- B. 1.61 近似に基づく手法を適用した提案手法 2

の 2 手法を Visual C++ を用いて実装し, 実際の性能評価を行なう. 最適解としては, IP を線形計画ソフト NUOPT で解くことで得られる整数解を用いることにする. 入力には次の入力データを用い, この入力データの一部を変化することで様々な実験を行なう.

標準入力データ 2

- (1) 施設は, 整数格子点の座標 $(0, 0)$ から $(500, 500)$

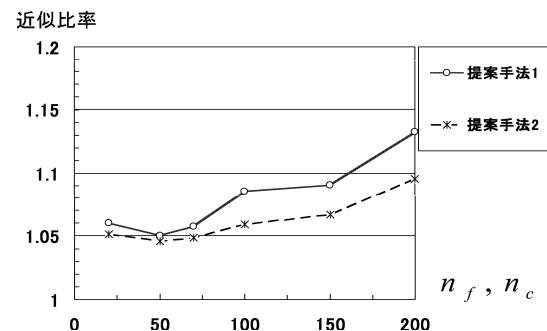
の間でランダムに 100 個ずつ発生

- (2) 利用者は, 整数格子点の座標 $(0, 0)$ から $(500, 500)$ の間でランダムに 100 個ずつ発生
- (3) サービスは 50 種類ずつ発生させ, 利用者が必要とするサービスはランダムに決定
- (4) 開設コストは, 1 から 1000 の間のランダムな正整数
- (5) 施設の容量は, 1 から 10 の間のランダムな正整数
- (6) サービスインストールコストは, 1 から 300 の間のランダムな正整数
- (7) 施設と利用者の接続コストは, 施設と利用者のユークリッド距離

図と表の説明

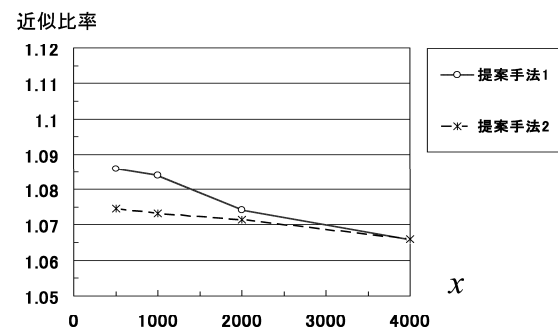
図は, 実験を 20 回行ったときの平均の近似比率, 以下では特に明記しないかぎり簡略化して近似比率と呼ぶ, を表すもので, 縦軸にはこの近似比率を用いている.

実験 1 施設数 n_f , 利用者数 n_c を変化させた場合



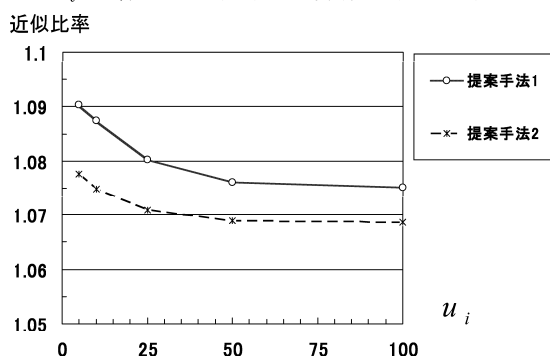
実験 2 接続コストを変化させた場合

座標範囲 (x, x) を広くすることで接続コストが増加した場合を考える.



実験 3 施設の容量 u_i を変化させた場合

施設の容量 u_i を一定とし、標準入力データの施設の容量 u_i を増加した場合の計算機実験を行う。



9.2 計算機実験 2 のまとめ

実験 1 より、提案手法 1 と提案手法 2 が共に良い近似比率を得ているといえる。これより、容量制約つきサービスを考慮した施設配置問題に対する解法として、サービスを考慮した施設配置問題に帰着させる手法は有効であるといえる。また提案手法 2 の方が良い近似比率を得ている要因としては、容量制約つきサービスを考慮した施設配置問題の場合においても、2 近似アルゴリズムの考え方が優れているためと考えられる。

実験 2 より、接続コストを増加させていくと、提案手法 1 と提案手法 2 が共に近似比率が良くなっていることがわかる。これは提案手法 1 と提案手法 2 が共に、1.61 近似に基づく提案手法に帰着させる際に、開設コストの影響を受けやすいためと考えられる。つまり、接続コストを増加することで開設コストの影響が少なくなり、1.61 近似に基づく提案手法の本来の性能が発揮されるため、近似比率がよくなっているといえる。

実験 3 より、容量の制約を厳しくした場合、提案手法 1 と提案手法 2 が共に近似比率が悪くなっていることがわかる。そして容量の制約を緩めていくにつれて、1.61 近似アルゴリズムの本来の性能が発揮され、近似比率が良くなっている。

10 結論

本論文では、Shmoys, Swamy and Levi が提唱したサービスを考慮した施設配置問題を扱った。この問題に対して、3 つの手法を提案し、Shmoys et al. によって提案された 6 近似アルゴリズムと実験的性

能を比較、検討することによって、提案手法が近似比率と計算時間の両面で高い性能を持つことを示すことができた。さらに、この問題をより現実に近い“容量制約つき”の場合に発展させ、容量制約つきサービスを考慮した施設配置問題に対する手法も提案した。計算機実験により、サービスを考慮した施設配置問題に帰着させるこの手法は有効であることを示すことができた。

今後の課題としては、これらの提案手法に対して理論的な近似保証を与えることが挙げられる。また、計算機実験において結果が悪くなるような入力データを発見し、実験を行うことも挙げられる。

謝辞

本研究は、一部、21 世紀 COE プログラムおよび文部科学省科学研究費補助金からの援助のもとで行われたものである。

参考文献

- [1] K. Jain, M. Mahdian and A. Saberi: A new greedy approach for facility location problems. *ACM Symposium on Theory of Computing*, 2002, pp. 731–740.
- [2] M. Mahdian, Y. Ye and J. Zhang: Improved approximation algorithms for metric facility location problems. *APPROX*, <http://www.mit.edu/~mahdian/pub.html>, 2002.
- [3] M. Mahdian, Y. Ye and J. Zhang: A 2-Approximation Algorithm for the Soft-Capacitated Facility Location Problem. <http://www.mit.edu/~mahdian/pub.html>, *APPROX*, 2003.
- [4] D. B. Shmoys, C. Swamy and R. Levi: Facility Location with Service Installation Costs. *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2004, pp. 1081–1090.