アルゴリズム 72−6
(2000. 3. 21)

施設配置問題に対する近似アルゴリズムの実際的性能評価

高田 英幸　　浅野 孝夫
中央大学大学院 理工学研究科 情報工学専攻

容量無し施設配置問題は1960年代前半からORの分野で研究されている問題で, ある地域内において, あるサービスを行なう施設をどのように配置したら全利用者にとって最も効率よく利用できるかという問題である. 目的は, 利用費用と配置費用の合計を最小にするような施設の配置場所と全利用者の施設割り当ての決定である. 本論文では理論的な観点から提案された容量無し施設配置問題に対する代表的な近似アルゴリズムの実際的性能評価を行なう.

Practical Performances of Approximation Algorithms for the Metric Uncapacitated Facility Location Problem

TAKADA Hideyuki　　ASANO Takao
Department of Information System and Engineering, Chuo University

The metric uncapacitated facility location problem has been studied in operations research since the early 60's and defind as follows. There is a set of locations at which we may build facilities, and a set of client locations to be serviced by facilities. The objective is to determine a set of locations which to open facilities so as to minimize the total facility and assignment costs. In this paper, we survey several approximation algorithms for this problem and estimate their practical performances.

1 序論

容量無し施設配置問題 (the metric uncapacitated facility location problem) とは, 利用者に好都合になるように施設を配置する場合で, 施設には供給できるサービスに制限がない. この問題は, 多項式時間では解くことができないと信じられている **NP 困難**な問題の 1 つである. そのため, 最適性を譲る代わりに入力サイズの多項式時間で良い解を生成するようなアルゴリズムを開発することが望まれる. このようなアルゴリズムを近似アルゴリズムという.

1997 年に Shmoys-Tardos-Aardal が初めての定数近似率 3.16 のアルゴリズム [4] を提案した. これは, 問題が整数計画法に定式化できることを利用する方法である. アルゴリズムは整数計画問題を緩和して線形計画問題を解き, その基底解を整数解へ丸める. さらに, 線形計画問題の解からランダム配置を行うことで, 近似率 1+2/e のアルゴリズムを 1998 年に Chudak が達成している [2]. 1999 年, Charikar-Guha はスケーリングと局所改善による近似率 1.728 のアルゴリズムを達成している [1].

本論文では, 上記のような理論的観点から提案された代表的なアルゴリズムを概観し, 計算機実験を通してそれらの実際的性能を評価する.

2 容量無し施設配置問題

容量無し施設配置問題の入力は施設を配置できる候補地の集合 F, 各候補地の配置費用 f_i $(i \in F)$, 利用者の集合 D と, 候補地と利用者を結ぶ無向辺の距離関数 c_{ij} $(i \in F, j \in D)$ である. 候補地の数 $|F|$ を n, 利用者の数 $|D|$ を m, 集合 N を F と D の和集合 $(N = F \cup D)$ とする. また, 配置費用 f_i は非負で, 距離関数 c は非負, 対称 $(c_{ij} = c_{ji})$ で, 三角不等式 $(c_{ij} + c_{jk} \geq c_{ik}$ $(i, j, k \in N))$ を満たすものとする. なお, 解析では $n = m$ と扱っている.

目的は, 候補地からいくつかの施設を選択して配置し, 全利用者を配置された施設に割り当てることである. 利用者と割り当てられた施設との距離を利用費用としている. 利用者 j が施設 i を利用する場合の利用費用は c_{ij} である. 最終的に, 利用費用と配置費用の合計を最小にするような配置場所と全利用者の利用施設割り当てを決定することである.

図 1 に入力例とその出力を示す. 上図の 50 点からなる○は配置できる候補施設, 50 点からなる●は利用者である. 下図はその入力のある解で, 利用者と直線でつながっている中心点が配置された施設, 利用者はその施設を利用することを示す. 距離関数 c_{ij} を施設 i と利用者 j の直線距離としているため, すべての利用者は配置された最近の施設に割り当てられている.

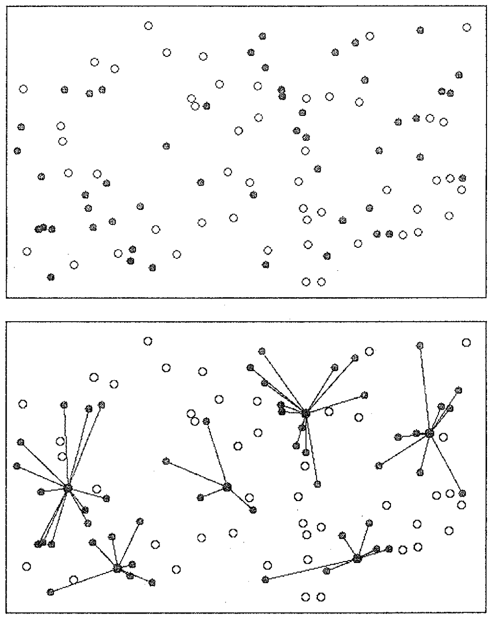

図 1: 入力と出力

3 貪欲法

最も単純な方法として以下の貪欲法が挙げられる. 初期状態は施設が 1 つも配置されていない状態で解の値を無限大とする. 各ステップで現在の解の値を最も減少させる候補施設を 1 つ配置する. すなわち, 利用者 j の施設を $\sigma(j)$,

配置されていない施設の集合を $\mathcal{F}$ としたとき

$$\max_{i\in\mathcal{F}}\{-f_i+\sum_{j\in D}\max\{0,c_{\sigma(j)j}-c_{ij}\}\}$$

が正ならば, その施設 i を配置し, 利用者を最近の施設に割り当てる. これを解の値が減少しなくなるまで繰り返す.

4 3.16 近似アルゴリズム

Shmoys-Tardos-Aardal による近似率 3.16 のアルゴリズム [4] の特徴は, 問題を整数計画問題に定式化して利用している点にある. 定式化に使われる決定変数は x_{ij} $(i\in F, j\in D)$ と y_i $(i\in F)$ である. 候補施設 i が配置されるならば y_i は 1, 配置されないならば 0 にする. 一方の x_{ij} は割り当てを示し, 利用者 j が施設 i を利用するならば 1, 利用しないならば 0 にする. こうして以下のような整数計画問題 (IP) に定式化できる.

$$\min \ \sum_{i\in F}f_iy_i+\sum_{i\in F}\sum_{j\in D}c_{ij}x_{ij}$$

subject to

$$\begin{array}{rcllr}
\sum_{i\in F}x_{ij} & = & 1 & (j\in D), & (1)\\
x_{ij} & \le & y_i & (i\in F, j\in D), & (2)\\
x_{ij} & \in & \{0,1\} & (i\in F, j\in D), & (3)\\
y_i & \in & \{0,1\} & (i\in F). & (4)
\end{array}$$

目的関数は配置された施設の配置費用と, 利用者とその利用施設との距離の合計を最小化することを示す. 制約式 (1) は, 各利用者が必ずある施設に割り当てられなければならないことを示す. (2) は配置されていない施設に利用者を割り当てることがないようにする制約式である. (3), (4) は整数制約である. 目的関数と制約式 (1)-(4) より得られた解 $(\boldsymbol{x}_{IP},\boldsymbol{y}_{IP})$ は最適解である. しかし, 実行可能領域は非凸であるため, この定式化は実際的な計算において有用でない. 多項式時間で解けるように, 問題を線形計画問題 (LP) に緩和し, LP の解を実行可能な整数解に丸めるという手法がしばしば用いられる. それは, 0-1 整数制約 (3)-(4) を以下のようにすることである.

$$\begin{array}{rcllr}
x_{ij} & \ge & 0 & (i\in F, j\in D), & (5)\\
y_i & \ge & 0 & (i\in F). & (6)
\end{array}$$

4.1 Step 1 : LP の計算

目的関数と制約式 (1)-(2), (5)-(6) より計算された実行可能解を $(\boldsymbol{x}^*,\boldsymbol{y}^*)=(x^*_{ij},y^*_i)$, 解の値を LP* とする. 最小化問題なので線形計画緩和問題の解 LP* は整数計画問題の解 IP* よりも常に小さくなる. すなわち, 以下の関係が成立する.

$$\mathrm{LP}^*\le\mathrm{IP}^*.$$

4.2 Step 2 : フィルターアルゴリズム

$(\boldsymbol{x}^*,\boldsymbol{y}^*)$ から各 $j\in D$ に対して $c_j(\alpha)$ を計算する. α は区間 $(0,1)$ のパラメータ値である. まず, ある利用者 j と全施設を近い順に並べかえる. 便宜上, $c_{\pi(1)j}\le c_{\pi(2)j}\le\cdots\le c_{\pi(n)j}$ とおく. $i^*=\min\{i' : \sum_{i=1}^{i'}x^*_{\pi(i)j}\ge\alpha\}$ を計算し $c_j(\alpha)=c_{\pi(i^*)j}$, $\alpha_j=\sum_{i\in F:c_{ij}\le c_j(\alpha)}x^*_{ij}$ とする. また, $\bar{x}_{ij}$ を以下のように定める.

$$\bar{x}_{ij}=\begin{cases}x^*_{ij}/\alpha_j & (c_{ij}\le c_j(\alpha))\\ 0 & (\text{それ以外}).\end{cases}$$

さらに, 各 $i\in F$ に対して $\bar{y}_i=\min\{1,y^*_i/\alpha\}$ とおく. これ以降, 各利用者 j の利用費用が高々 g_j であるとき g-close という.

補題 4.1 $(\boldsymbol{x}^*,\boldsymbol{y}^*)$ より, 以下の 1, 2 を満たす g-close な実行可能解 $(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}})$ が計算できる.

1. $\sum_{i\in F}f_i\bar{y}_i\le\frac{1}{\alpha}\sum_{i\in F}f_iy^*_i$.
2. $g_j\le c_j(\alpha)\le\frac{1}{1-\alpha}\sum_{i\in F}c_{ij}x^*_{ij}$ $(j\in D)$. □

4.3 Step 3 : ラウンドアルゴリズム

$(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}})=(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}})$ とコピーする. $\hat{F}=\{i\in F : 0<\hat{y}_i<1\}$ を不完全な施設の集合, $\hat{D}=\{j\in D : \hat{x}_{ij}>0, i\in\hat{F}\}$ を $\hat{F}$ の施設に割り当てら

れている利用者の集合とする．アルゴリズムは，$\hat{D}$ から一番小さい g_j を選び $j' = j$ とする．j' が割り当てられている施設の集合 $S = \{i \in \hat{F} : \hat{x}_{ij'} > 0\}$ と，その施設の集合に割り当てられている利用者の集合 $T = \{j : \exists i \in S, \hat{x}_{ij} > 0\}$ を計算する．集合 S から f_i が最小である施設

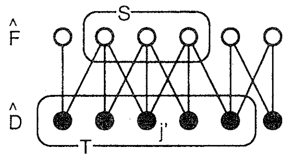

図 2: 利用者 j' の集合 S と集合 T

i を配置し，j' と集合 T の利用者を割り当てる．それは，配置する施設を $i' = i$ としたとき，$\hat{y}_{i'} = 1$, $i \in S - \{i'\}$ に対して $\hat{y}_i = 0$ とする．$j \in T$ にも施設 i' を割り当てるので $\hat{x}_{i'j} = 1$, $i \in F - \{i'\}$ には $\hat{x}_{ij} = 0$ とする．$\hat{F}$, $\hat{D}$ を更新して上記のことを $\hat{D}$ が空になるまで繰り返す．

最後に，$\hat{x}_{i'j} > 0$ かつ $\hat{y}_{i'} = 1$ となるような i' が存在するときは代表を 1 つ選んで i' と改めておき，$\hat{x}_{i'j} = 1$, $i \in F - \{i'\}$ には $\hat{x}_{ij} = 0$ とする．$\hat{y}_i > 0$ で，すべての j に対して $\hat{x}_{ij} = 0$ ならば，$\hat{y}_i = 0$ にする．こうして $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ は 0-1 整数になる．

補題 4.2 $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ より，以下の 1, 2 を満たす $3g$-close な実行可能整数解 $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ が計算できる．

1. $\displaystyle\sum_{i\in F} f_i \hat{y}_i \le \sum_{i\in F} f_i \bar{y}_i \le \frac{1}{\alpha}\sum_{i\in F} f_i y_i^*$.
2. $\displaystyle 3g_j \le 3c_j(\alpha) \le \frac{3}{1-\alpha}\sum_{i\in F} c_{ij}x_{ij}^*$ $(j \in D)$.

証明： 施設 $i' \in S$ を配置したとき，$\sum_{i\in S}\hat{x}_{ij} = 1$, $\hat{x}_{ij} \le \hat{y}_i$, 最小の配置費用の施設を配置することより，

$$f_{i'} = \min_{i\in S} f_i \le \sum_{i\in S} f_i \hat{x}_{ij} \le \sum_{i\in S} f_i \hat{y}_i$$

である．そのときの利用者 $j \in T$ の割り当て費用 $c_{i'j}$ は，選ばれた利用者 $j' = \min_{j\in\hat{D}} g_j$ と j の共通の施設を $i \in S$ としたとき，$c_{i'j'} + c_{ij'} + c_{ij} \le g_{j'} + g_{j'} + g_j \le 3g_j$ である． □

補題 4.1, 補題 4.2 より $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ は，

$$\begin{aligned} & \sum_{i\in F} f_i\hat{y}_i + \sum_{i\in F}\sum_{j\in D} c_{ij}\hat{x}_{ij} \\ \le\ & \frac{1}{\alpha}\sum_{i\in F} f_i y_i^* + 3\sum_{j\in D} c_j(\alpha) \\ \le\ & \frac{1}{\alpha}\sum_{i\in F} f_i y_i^* + \frac{3}{1-\alpha}\sum_{i\in F}\sum_{j\in D} c_{ij}x_{ij}^*. \end{aligned}$$

である．

α を区間 $(\beta, 1)$ で一様に選び，それを決定性アルゴリズムに適応させる．

補題 4.3 $\int_0^1 c_j(\alpha)d\alpha = \sum_{i=1}^n c_{ij}x_{ij}^*$. □

ある α を区間 $(\beta, 1)$ でランダムに選んだときの期待値は，補題 4.3 より，

$$\begin{aligned} & E[\frac{1}{\alpha}\sum_{i\in F} f_i y_i^* + 3\sum_{j\in D} c_j(\alpha)] \\ \le\ & \frac{\ln(1/\beta)}{1-\beta}\sum_{i\in F} f_i y_i^* + \frac{3}{1-\beta}\sum_{j\in D}\sum_{i\in F} c_{ij}x_{ij}^*. \end{aligned}$$

$\max\{\frac{\ln(1/\beta)}{1-\beta}, \frac{3}{1-\beta}\}$ を最小にするのは，$\beta = 1/\mathrm{e}^3 < 0.051$ である $(3/(1-\mathrm{e}^3) < 3.16)$.

定理 4.4 α を区間 $(0.051, 1)$ でランダムに発生させたとき,3.16 近似アルゴリズムである.□

上記の議論から，$c_j(\alpha)$ はステップ関数であるため，$\frac{1}{\alpha}\sum_{i\in F} f_i y_i^* + 3\sum_{j\in D} c_j(\alpha)$ を最小にするような $(0.051, 1)$ 区間の α を簡単に選べて，近似率 3.16 の決定性アルゴリズムが得られる．

5　$(1 + 2/\mathrm{e})$ 近似アルゴリズム

Chudak の $1+2/\mathrm{e}$ 近似アルゴリズム [2] は，線形計画問題とその双対問題の解を用いて近似率 $1+2/\mathrm{e}$ を達成している．以下は線形計画問題に対する双対問題 (D) の定式化である．

$$\max \sum_{j\in D} v_j$$

subject to

$$\begin{aligned} \sum_{j\in D} w_{ij} &\le f_i \quad (i \in F), \\ v_j - w_{ij} &\le c_{ij} \quad (i \in F, j \in D), \\ w_{ij} &\ge 0 \quad (i \in F, j \in D). \end{aligned}$$

線形計画問題の実行可能解 $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ の目的関数値を LP* としたとき, この双対問題の実行可能解 $(\boldsymbol{v}^*, \boldsymbol{w}^*)$ の目的関数値 D* は LP* と一致する. また相補条件より, $x^*_{ij} > 0$ ならば $v^*_j - w^*_{ij} = c_{ij}$ であり, したがって $w^*_{ij} \geq 0$ より $c_{ij} \leq v^*_j$ である. これは, ある利用者 j を $x^*_{ij} > 0$ となるどの施設 i に割り当てても, 利用費用が高々 v^*_j であることを示している. $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ は v^*-close であると言える.

5.1 Step 1 : LP の計算

目的関数と制約式 (1)-(2), (5)-(6) より計算された実行可能解を (x^*_{ij}, y^*_j) とする. また, 双対問題との関係より v^*_j をある利用者 j の $x_{ij} > 0$ となる最も遠い施設までの距離 c_{ij} とする.

5.2 Step 2 : クラスターセンターの計算

1. $S \leftarrow D,\ Cc \leftarrow \emptyset$
2. $S \neq \emptyset$ である限り 3 から 6 を繰り返す
3. 　最小の $v^*_j + C^*_j$ $(j \in S)$ を選ぶ
4. 　$j_0 = j,\ \ Cc \leftarrow Cc \cup \{j_0\}$
5. 　$Q \leftarrow \{k \in S : Ne(k) \cap Ne(j_0) \neq \emptyset\}$
6. 　$S \leftarrow S - Q$

なお, 関数 $Ne(j)$ は $j \in D$ の $\{i \in F : x^*_{ij} > 0\}$ の集合, C^*_j は $\sum_{i \in F} c_{ij} x^*_{ij}$ とする. このとき, 集合 Cc がクラスターセンターになる.

5.3 Step 3 : 配置と割り当て

$L = \bigcup_{j \in Cc} Ne(j)$ をセンター候補施設集合, $R = F - L$ は残りの候補施設で非センター施設集合とする. 各 $j \in Cc$ に対し $Ne(j)$ から独立に確率 y^*_i で配置される施設を必ず 1 つ選ぶ. ここではそのため 0-1 乱数を使い, その値が y^*_i 以下ならば, 施設 $i \in L$ の配置を決定する. さらに, 非センター施設 $i \in R$ を独立に確率 y^*_i で配置を決定する. 最後に, 全利用者を配置された最近の施設に割り当てて終了する.

定理 5.1 この乱数使用確率アルゴリズムで LP 解を整数解に丸めたときの期待値は

$$\begin{aligned} &\sum_{i \in F} f_i y^*_i + \sum_{j \in D} C^*_j + \frac{2}{\mathrm{e}} \sum_{j \in D} v^*_j \\ =\ & (1 + \frac{1}{\mathrm{e}})\mathrm{LP}^* < 1.736\mathrm{IP}^*. \end{aligned}$$

である. □

6 1.8526 近似アルゴリズム

Charikar-Guha の 1.726 近似は条件付きなので, 1.8526 近似アルゴリズム [1] を示す.

6.1 3 近似アルゴリズム

最初に 3 近似アルゴリズムを示す. これはある初期解を局所探索で改善する. 解が改善されなくなるまで局所探索を繰り返す.

まず, 配置費用の昇順に施設を並べかえる. 1 番目の施設が最も配置費用の安い施設になる. S_i と C_i を $i = 1, \ldots, n$ 番目までの施設を配置したときの配置費用と利用費用とする. $\min_{i \in F}\{S_i + C_i\}$ を実現する i 番目までの施設を配置した解を初期解とする.

局所探索は集合 F から施設 i をランダムに決定し, 以下のように定義される $gain(i)$ を計算する. なお, $\sigma(j)$ は利用者 j の現在の利用施設を示す.

ランダムに選ばれた施設 i が現在配置されていないとき, $c_{ij} < c_{\sigma(j)j}$ となる利用者 j に i のマークを付ける. マークされた利用者の集合を $\hat{D}(i)$ とする. さらに, 現在配置されているすべて施設 $\mathcal{F}$ について以下を行なう. $\hat{D}(i')$ を施設 $i' \in \mathcal{F}$ に割り当てられている利用者 j の集合とする. $\hat{D}(i') \setminus \hat{D}(i)$ の利用者すべてを施設 i に割り当て, 施設 i' を取り除いたときの解の減少値を $gain'(i')$ とする. すなわち,

$$gain'(i') = f_{i'} + \sum_{j \in \hat{D}(i') \setminus \hat{D}(i)} (c_{i'j} - c_{ij})$$

である. $gain'(i') > 0$ ならば, 施設 i' に i のマークを付け, $\hat{D}(i') \setminus \hat{D}(i)$ の利用者に i のマークを付ける. $gain'(i') \leq 0$ ならば $gain'(i') = 0$

とする．よって，現在の解の減少値 $gain(i)$ は，

$$gain(i) = -f_i + \sum_{i' \in \mathcal{F}} gain'(i') + \sum_{j \in \hat{D}(i) \setminus \hat{D}(i')} (c_{\sigma(j)j} - c_{ij})$$

である．$gain(i) > 0$ ならば施設 i を配置し，マーク i の施設を取り除き，さらに，マーク i の利用者を施設 i に割り当てる．

ランダムに選ばれた施設 i が現在にすでに配置されているときは，全利用者を最適な割り当てに変更する．

定理 6.1 [1] 上記のアルゴリズムは $S \leq (1+\epsilon)(S_{LP}+2C_{LP})$ かつ $C \leq (1+\epsilon)(S_{LP}+C_{LP})$ の解を生成する．ただし，S を配置費用の合計値，C を利用費用の合計値としたとき，S_{LP} と C_{LP} は LP* の配置費用の合計値と利用費用の合計値である． □

6.2 $(2.414+\epsilon)$ 近似アルゴリズム

最初に配置費用をスケーリングし，3 近似アルゴリズムを使う．スケーリングとは配置費用を一様に δ 倍し，最後に元の配置費用に戻すことである．したがって，定理 6.1 より，ある小さな値 $\epsilon' > 0$ に対して

$$(1+\epsilon')\left[(1+\delta)S_{LP} + (1+\frac{2}{\delta})C_{LP}\right]$$

を満たす解が得られる．

定理 6.2 [1] $\delta = \sqrt{2}$, $\epsilon = (1+\sqrt{2}\epsilon')$ としたとき，スケーリングを用いた上記のアルゴリズムの近似率は $1+\sqrt{2}+\epsilon$ である． □

6.3 1.8526 近似アルゴリズム

1.8526 近似アルゴリズムは配置費用を一様に $\ln(3\delta) = 2/(3\delta)$ を満たす $\delta = 0.7192$ でスケーリングし，初期解を Jain-Vazirani [3] による近似率 3 の双対アルゴリズムの解とする．その後，配置費用をもとに戻し**貪欲改善**を行なう．貪欲改善とは $gain(i)/f_i$ が最大な施設 $i \in F$ を配置することである．そのときの解は高々，

$$(1+\ln(3\delta))S_{LP} + (1+\frac{2}{3\delta})C_{LP}$$

となり以下の定理が得られる．

定理 6.3 [1] 配置費用を一様に $\delta = 0.7192$ でスケーリングし貪欲改善をすることで，近似率 1.8526 が達成できる． □

次に，近似率 3 の双対アルゴリズムを示す．これは双対問題 (D) の解の性質を用いる．

6.3.1 Step 1: v_j^*, タイトな枝, 配置候補施設の計算

まず，双対問題の変数を $v_j^* = 0$, $w_{ij}^* = 0$ とし，利用者 $j \in D$ に**非連結**のラベルを付ける．非連結の v_j^* を一様に上げていく．例えば $v_j^* = v_j^* + 1$. $v_j^* = c_{ij}$ になったとき，枝 (i,j) を**タイト**とする．双対問題の制約式を満たすために w_{ij}^* も一様に上げる．もし，$\sum_j w_{ij}^* = f_i$ に達したら，利用者 j のラベルを**連結**とし，施設 i を**配置候補施設**とする．そのとき，施設 i と非連結な利用者 j' とのタイトな枝 (i,j') の存在を調べる．存在するなら，利用者 j' も連結とする．また，非連結の利用者 j の v_j^* を一様に上げていき，配置候補施設とタイトな枝をもつようになったら，j を連結とする．すべての j が連結になったら終了する．

6.3.2 Step 2：配置する施設の決定

Step 1 の配置候補施設と全利用者をノードとし，枝をタイトな枝としたグラフを T とする．T のあるノード a と b に高々 2 のパスが存在するなら，枝 (a,b) を加える．枝を加えられたグラフを T^2, 配置候補施設ノードのみからなる T^2 の部分グラフを H とする．グラフ H の最大独立集合 I は Step 1 で配置候補施設になった順に施設ノードを選ぶことで得られる．それは，ノード i が選ばれたらその近傍ノード i' は選ばれない．集合 I の施設を配置する．

6.3.3 Step 3：利用者の割り当て

配置された施設 $i \in I$ と利用者 j にタイトな枝があるなら i に割り当てる．タイトな枝がな

いときは, 利用者 j のタイトな枝 $(i',j), i' \notin I$ の近傍ノード $i \in I$ に割り当てる.

定理 6.4 [1] 双対アルゴリズムの配置費用を S, 利用費用を C とすると, $3S + C \leq 3\mathrm{D}^*$ である. □

7 実際的性能評価

容量無し施設問題に対し4つの近似アルゴリズム (貪欲法, 近似率 3.16, 近似率 1.736, 近似率 1.8526) を C 言語でプログラム化し, 計算機実験を行った. 実験数は 12 個で, そのインスタンスはすべて $|F| = |D| = 100$, 施設の配置費用は 500 から 800 までの乱数値, 集合の各要素の x 座標は 0 から 400, y 座標は 0 から 300 の乱数, 距離関数 c_{ij} は各 2 点間 i, j の直線距離とした. アルゴリズムがもつ最悪な近似率に到達するようなインスタンスをつくれなかったため, インスタンスを乱数にしてしまった. この方法で発生させたインスタンスに対し線形計画緩和問題を解き, 整数解でないものだけを使用した.

図 3 は4つのアルゴリズムの目的関数値を LP 解の目的関数値で割ったものである. x 座標は問題番号, y 座標が近似率である. 近似率は常に満たされていた.

次に, 貪欲法, 近似率 3.16, 近似率 1.736 の解を初期解とし貪欲改善を行なった. 貪欲改善は 近似率 1.852 のアルゴリズムで用いた方法である. その結果を図 4 に示す. 近似率の悪い解がかなり改善された.

貪欲改善では改善のときには必ず 1 つの施設が配置され, いくつかの施設が取り除かれる. よって, 施設を取り除くだけで解を改善するステップがない. このステップを貪欲改善された後に組み込んで実験を行なった. 初期解を図 3 とし, 改善後の結果を図 5 に示す. 図 4 と比べると結果が良くなるものと, 悪くなるものがある. ただし, 結果が悪くなるものは少なかった.

8 おわりに

容量無し施設配置問題に対する 4 つの近似アルゴリズムについて示し, そのアルゴリズムの実際的性能を比較した. アルゴリズムの近似率は最悪を保証しているので, 近似率が高いから結果が悪くなるというわけでない. ただし, 近似率 1.852 のアルゴリズムでは貪欲改善を行なうため, 4 つのアルゴリズムのなかでは良い結果を示した. そのことから, 3 つのアルゴリズムの解を初期解とし貪欲改善により解を改善した. 実験により, 貪欲改善の有効性が示された. また, 貪欲改善に施設を取り除くステップを加えた実験も行なった. 加えたものはさらなる解の改善ができたものが多かった. 貪欲改善を加えた解析ができなかったが, 近似率のさらなる改善が見込まれる.

本研究は, 一部, 中央大学理工学研究所, 私立大学ハイテク・リサーチ・センターおよび文部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] M. Charikar and S. Guha: Improved combinatorial algorithms for the facility location and k-median problems, *Proc. of the 40th Annual IEEE Symposium on Foundations of Computer Science*, 1999, pp. 378-388.

[2] F. A. Chudak: Improved approximation algorithms for uncapacitated facility location, in R. E. Bixby, E. A. Boyd, and R. Z. Ríos-Mercado, editors: *Integer Programming and Combinatorial Optimization, Lecture Notes in Computer Science* 1412, Springer, 1998, pp. 180-194.

[3] K. Jain and V. V. Vazirani: Primal-dual approximation algorithms for metric facility location and k-median problems, *Proc. of the 40th Annual IEEE Symposium on Foundations of Computer Science*, 1999, pp. 2-13.

[4] D. B. Shmoys, É. Tardos, and K. I. Aardal: Approximation algorithms for facility location problems, *Proc. of the 29th Annual ACM Symposium on Theory of Computing*, 1997, pp. 265-274.

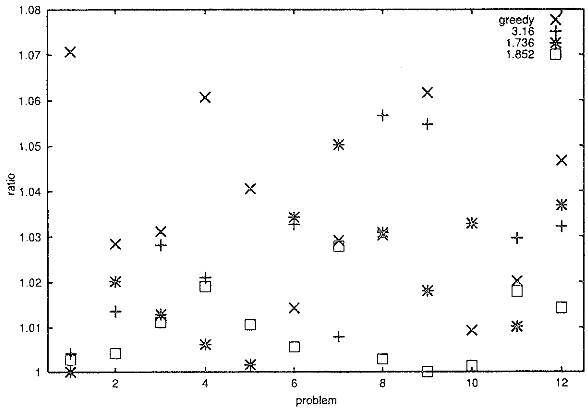

図 3: 4 つのアルゴリズムの近似率

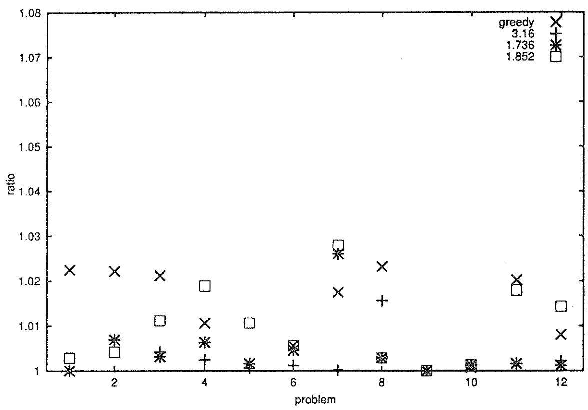

図 4: 貪欲改善を用いた近似率

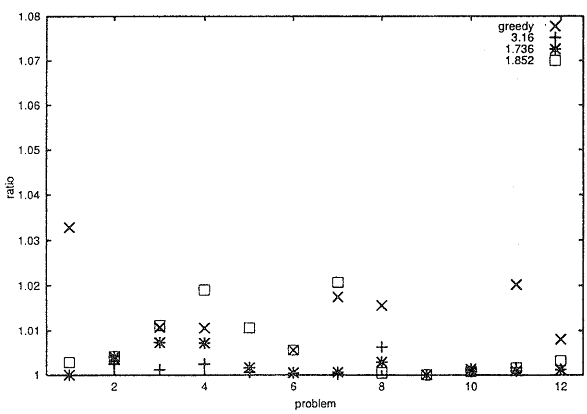

図 5: 貪欲改善と施設を取り除くステップを組み込んだ結果