アルゴリズム 88－2
（2003. 1. 20）

代表的施設配置近似アルゴリズムの実験的性能評価

九里 史朗　　浅野孝夫

中央大学大学院 理工学研究科 情報工学専攻

概要

施設配置問題とは, 人々が効率良く施設を利用するために開設する施設を決める問題である. この問題は現実的な環境において頻繁に発生する問題であり, k メディアン問題, k センター問題など様々な関連問題が考えられている. 本稿では, 容量制限なしメトリック施設配置問題に対して, 双対フィット法に基づく代表的な近似アルゴリズムの実験的性能評価を行う. そして, 現在最も良い 1.52 近似アルゴリズムに対しても, 局所改善を施して実験的性能評価を行い有効性を調べる.

Experimental Performance of Representative Approximation Algorithms for the Metric Facility Location Problem

Shiro Kunori　　Takao Asano

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

Abstract

The facility location problem is to decide which facilities are open to use them effectively. This problem appears frequently in real environments, and various related problems have been proposed such as k-median and k-center problems. In this paper, we evaluate experimental performance of representative algorithms for the metric uncapacitated facility location problem including algorithms that are analyzed by dual fitting method. We also propose some heuristics for the best known 1.52-approximation algorithm.

1 はじめに

容量制限なしメトリック施設配置問題 (metric uncapacitated facility location problem) は, 以下, 簡略化して施設配置問題と呼ぶが, 設立地計画をモデル化した問題であり, 1960 年代前半から OR の分野で中心的な問題として幅広く研究されている. この問題は, 入力として n_f 個の開設候補の施設集合 F と n_c 人の利用者集合 C からなる完全 2 部グラフが与えられる ($n = n_f + n_c,\ m = n_f \times n_c$ とする). そして, 施設 i には正整数の開設コスト f_i が与えられ, 施設 i と利用者 j との間には接続コスト c_{ij} (非負, 対称で三角不等式を満たす) が与えられる. この入力において, 適切に施設を開設し, すべての利用者を開設した施設に接続したい (割り当てたい) . そして, 総コスト (開設コストと接続コストの合計) が最小になるような施設の開設と接続を求めることがこの問題の目的である.

この問題の具体例な入力例 (すべての開設コスト f_i は一定, 施設数 n_f は 100 個, 利用者数 n_c は 100 人, 接続コストは 2 点間のユークリッド距離の値とした例) を図 1 に, この入力例に対する出力 (最適解) を図 2 に示す. なお以下の図において, □は開設候補の施設 (図 2 では未開設施設), ■は開設された施設, ◯は利用者, そして実線分は接続を表している.

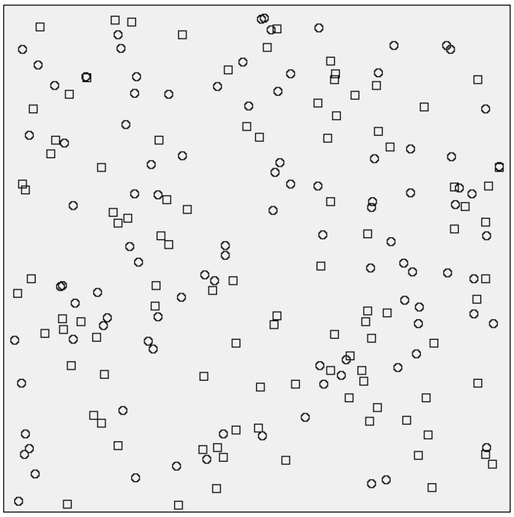

図 1. 入力例

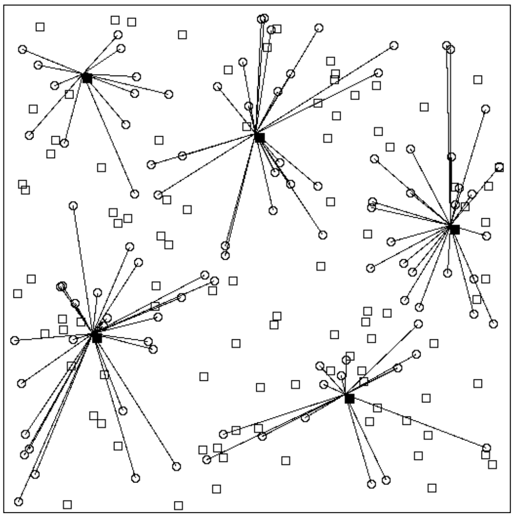

図 2. 出力例

この問題は以下のように整数計画問題 (IP) として定式化できる. ここで, 変数 x_{ij} は施設 i に利用者 j が接続しているかどうかを表し, 変数 y_i は施設 i が開設しているかどうかを表している.

$$\min \sum_{i\in F} f_i y_i + \sum_{i\in F}\sum_{j\in C} c_{ij} x_{ij} \tag{1}$$

$$\text{s.t.} \sum_{i\in F} x_{ij} = 1 \quad (\forall j \in C), \tag{2}$$

$$x_{ij} \le y_i \quad (\forall i \in F, j \in C), \tag{3}$$

$$x_{ij} \in \{0,1\} \quad (\forall i \in F, j \in C), \tag{4}$$

$$y_i \in \{0,1\} \quad (\forall i \in F). \tag{5}$$

この整数計画問題を多項式時間で解くのは困難であるため, 線形計画問題 (LP) に緩和する. したがって, 制約式 (4) は以下の (6) に, (5) は以下の (7) になる.

$$x_{ij} \ge 0 \quad (\forall i \in F, j \in C), \tag{6}$$

$$y_i \ge 0 \quad (\forall i \in F). \tag{7}$$

2 これまでの主な研究の流れ

この問題は NP 困難な問題であり, 1997 年に初めての定数近似アルゴリズム [9] が提案されて以来, 様々な手法に基づく近似アルゴリズムが提案されている. これを以下の表 1 にまとめる (ここで ϵ は小さい正数とする).

表 1. これまでの主な研究

年	近似比率	主な手法	文献
1997	3.16	LP 丸め	[9]
1998	2.408	LP 丸め, 貪欲改善法	[3]
1998	1.736	LP 丸め	[2]
1998	$5+\epsilon$	局所探索法	[6]
1999	3	主双対法	[5]
1999	1.853	主双対法, 貪欲改善法	[1]
2001	1.861	双対フィット法	[7]
2001	1.61	双対フィット法	[4]
2002	1.582	LP 丸め	[10]
2002	1.52	貪欲改善法	[8]

本稿では, 上記の **1.861 近似**, 1.861 近似の改善である **1.61 近似**, 1.61 近似を用いて初期解を求める **1.52 近似**, そして 1.52 近似を局所改善したもの (以下, **改善案**) の 4 つのアルゴリズムを C++言語で実装し, 実験的性能評価を行う.

3　代表的な近似アルゴリズム

4 つの近似アルゴリズムを概説する前に, 双対問題の定義と双対フィット法を概説する.

3.1　双対問題の定義

(LP) の双対問題 (D) は以下のとおりである.

$$\max \sum_{j \in C} \alpha_j \tag{8}$$

$$\text{s.t.} \sum_{j \in C} \beta_{ij} \ \leq \ f_i \quad (\forall i \in F), \tag{9}$$

$$\alpha_j - \beta_{ij} \leq \ c_{ij} \quad (\forall i \in F, j \in C), \tag{10}$$

$$\alpha_j, \ \ \beta_{ij} \ \geq \ 0 \quad (\forall i \in F, j \in C). \tag{11}$$

この双対変数に対しては, α_j は利用者 j の費用, β_{ij} は施設 i への j による貢献, すなわち $\beta_{ij} = \max(\alpha_j - c_{ij}, 0)$, と考えることができる.

3.2　双対フィット法

この手法は解析のみに用いられる手法である. まず, 緩和した主相補条件を満たす双対問題の解 (実行不可能解) をアルゴリズムにより求める. そして解析で, この実行不可能解を縮小して実行可能解にするが, その縮小する値が近似比率になることを用い, この値を求める.

3.3　1.861 近似アルゴリズム [7]

アルゴリズム (1.861 近似)

未接続の利用者の費用 α_j をすべて同じ割合で増加しながら以下の (a), (b) を繰り返す. すべての利用者が接続されたら終了とする.

(a) **if** (ある未開設施設 i とある未接続の利用者 j に対して, $\sum_j \beta_{ij} = f_i$) **then**
　施設 i を開設し, $\alpha_j \geq c_{ij}$ であるすべての未接続な利用者 j を施設 i に接続する.

(b) **if** (ある開設施設 i とある未接続の利用者 j に対して, $\alpha_j = c_{ij}$) **then**
　利用者 j を施設 i に接続する.

このアルゴリズムの (a) は, 幾人かの未接続な利用者による未開設施設 i への貢献の合計が i の開設コストと等しくなったら, i を開設として, i に貢献している利用者は i に接続することを意味する. (b) は, 利用者 j の費用が j とある開設施設との接続コストに等しくなったら, j をこの施設に接続することを意味する. また, (a), (b) から一度接続された利用者の接続は不変であることがわかる.

ある入力例に対する (a) の実行を図 3 に, (b) の実行を図 4 に示す. なお, 以下のアルゴリズムの実行例を示す図において, 左が実行前, 右が実行後を表していて, 明記されていない接続コストは, ∞ に近い大きな値かつ距離の性質を満たす値としている.

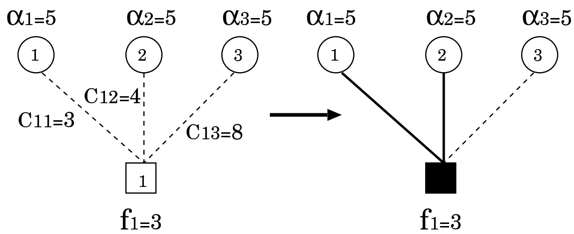

図 3. アルゴリズムの (a) の実行

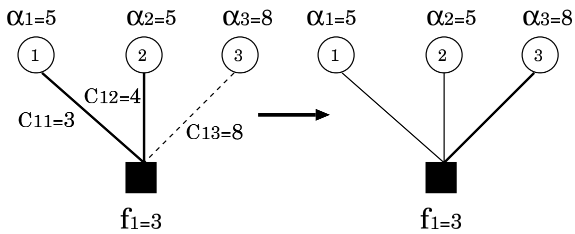

図 4. アルゴリズムの (b) の実行

解析

アルゴリズムでは, ある利用者 j はある施設に一度接続すると他の施設への影響は考えないものとしている. 具体的には, (a) の条件において $\sum_j \beta_{ij} = f_i$ となる j を未接続の利用者のみから考えている. そのため, アルゴリズムの最後において施設 i に接続していない j の β_{ij} を含む $\sum_{j \in C} \beta_{ij}$ を考えると, $\sum_{j \in C} \beta_{ij} = \sum_{j \in C} \max(\alpha_j - c_{ij}, 0) > f_i$ となるため (D) の制約式 (9) は成立しない. よって, 求めた解を α とすると α は実行不可能解である. ここで, 緩和条件 $\sum_{j \in C} \max\{(\frac{\alpha_j}{R} - c_{ij}), 0\} \leq f_i$ が成立

する最小の $R \geq 1$ を求めることができれば, $\frac{\alpha}{R}$ は実行可能解であり, $\frac{\sum_{j \in C} \alpha_j}{R}$ は最適解のコストの少なくとも $\frac{1}{R}$ 倍である. そのため近似比率は R である. しかし, この R は近似比率が最悪となるインスタンスがわからないかぎり求められない. そのため, 次の factor-revealing LP という定式化の解 z_k を用いて近似比率を求める.

この定式化の前に, 制約について述べる. $\max(\alpha_j - c_{ij}, 0)$ より, $\alpha_j \geq c_{ij}$ となる利用者 j のみを考えるが, このような利用者は初めの k までとする (以下では, 利用者が k 人までの問題を考える). そして α_i を $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_k$ と昇順に並びかえる. このとき, 以下の 2 つが成り立つ.

- 利用者 j, j' と施設 i に対して, $\alpha_j \leq \alpha_{j'} + c_{ij'} + c_{ij}$ である (距離の性質).
- 施設 i への総貢献は, i の開設コストを越えない. つまり, ある利用者 j と施設 i に対して, $\sum_{l=j}^{k} \max(\alpha_l - c_{il}, 0) \leq f_i$ である.

これらを制約としたとき, $\sum_{j=1}^{k} \max\{(\frac{\alpha_j}{R} - c_{ij}), 0\} \leq f_i$ となる最小の R を求めたい. ここで, 開設コストを f, 距離コストを d_j, 貢献を α_j として変形すると, $\frac{\sum_{j=1}^{k} \alpha_j}{f + \sum_{j=1}^{k} d_j} \leq R$ となるため, $\frac{\sum_{j=1}^{k} \alpha_j}{f + \sum_{j=1}^{k} d_j}$ の最大値を求めたい.

この定式化 (factor-revealing LP) は以下のとおりである.

$$
\begin{aligned}
&\max\ z_k = \frac{\sum_{j=1}^{k} \alpha_j}{f + \sum_{j=1}^{k} d_j} \\
&\text{s.t. } \alpha_j \leq \alpha_{j+1} \qquad (\forall j \in \{1, \ldots, k-1\}), \\
&\quad \alpha_j \leq \alpha_l + d_j + d_l \qquad (\forall j, l \in \{1, \ldots, k\}), \\
&\quad \sum_{l=j}^{k} \max(\alpha_l - d_l, 0) \leq f \quad (\forall j \in \{1, \ldots, k\}, \\
&\quad \alpha_j,\ d_j,\ f \geq 0 \qquad (\forall j \in \{1, \ldots, k\}).
\end{aligned}
$$

定理 3.1 [7]

$k \geq 1$ に対して z_k の上界はアルゴリズムの近似比率と等しく, $z_k < 1.861$ である.

3.4 1.61 近似アルゴリズム [4]

1.861 近似では一度決めた接続は不変であったが, 総コストが減少するなら**接続を変更する**としたのが 1.61 近似である.

アルゴリズムの流れは 1.861 近似と同じであるが, 利用者 j から施設 i への貢献 β_{ij} の定義を以下のように変更する.

- **if** (利用者 j は未接続) **then**
  $\beta_{ij} = \max(\alpha_j - c_{ij}, 0)$ とする.
- **if** (利用者 j は既に施設 i' と接続) **then**
  $\beta_{ij} = \max(c_{i'j} - c_{ij}, 0)$ とする.

この定義から, 既に施設に接続している利用者においても貢献を考えることがわかる. これに伴い, 1.861 近似のアルゴリズムの (a) の実行において, 既に接続している利用者 j も考える. すなわち, j が既に別の施設 i' に接続しているなら, j の接続を i' から i に変更する. この操作では $\beta_{i'j}$ は不変であり, β_{ij} のみが変わる. なお, 接続を変更することで i' へ接続している利用者がいなくなったら, i' は未開設とする (以下のアルゴリズムも同様とする).

ある入力例に対して接続を変更する実行を図 5 に示す.

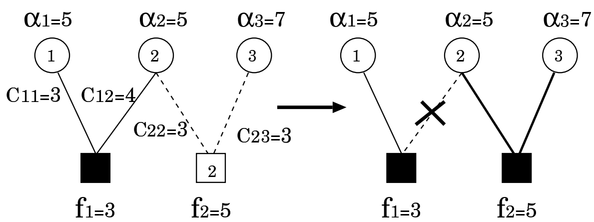

図 5. 接続を変更する実行

解析

このアルゴリズムによる解も, 1.861 近似と同じく実行不可能解である. そのため双対フィット法を用いる.

3.5 1.52 近似アルゴリズム [8]

このアルゴリズムは, コストスケーリング後に 1.61 近似を用いることで, 貢献を多く集めることのできる経済的な施設のみを開設してお

き, その後の貪欲改善によりできるだけ多く総コストを減少できる未開設施設を開設するアルゴリズムである.

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. もし未開設施設 i を開設することで接続コストが c から c' になるなら, $gain(i) = c - c' - f_i$ と定義する.

アルゴリズム (1.52 近似)

(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, 開設コストを一様に δ 倍した問題に対して, 3.4 節の 1.61 近似を適用する.

(貪欲改善) 開設コストを一様に $\frac{1}{\delta}$ 倍した問題 (つまりもとの問題) に対して, 総コストが減少する ($gain(i) > 0$ である) かぎり以下を繰り返す.

- **if** (ある未開設施設 i の $\frac{gain(i)}{f_i}$ が最大) **then**
  i を開設して, 利用者は最も近い開設施設に接続を変更する.

貪欲改善の条件において, $gain(i)$ ではなく $\frac{gain(i)}{f_i}$ を用いるのは, $gain(i)$ が大きいだけでなく f_i も小さい施設を開設したいためである.

ある入力例に対する貪欲改善の実行を図 6 に示す (左は初期解を求めた直後を表し, 便宜上 $\delta = 1.5$ としている). ここで, 図 6 の $gain(2)$ は $(4-3)+(8-3)-5$, つまり 1 である.

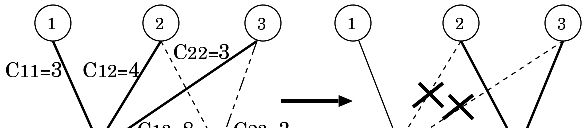

図 6. 貪欲改善の実行

3.5.1 改善案 (1.52 近似の局所改善)

このアルゴリズムは, 1.52 近似と同じ方法で初期解を求め, 1.52 近似の貪欲改善において施設を開設することだけではなく**開設施設を削除する (未開設にする)** ことも考慮するアルゴリズムである (この考え方は [1] で提案されている). すなわち, もし i が未開設なら開設し, そうではなく i が開設なら削除する. そしてその後に, i 以外の開設施設を削除すると総コストが減少するなら, 繰り返し削除することを考える. なお, このとき最も総コストが減少する値を, $gain(i)$ と定義する.

アルゴリズムの貪欲改善では, このような $gain$ をすべての施設で考え, $gain(i) > 0$ であるかぎり, $\frac{gain(i)}{f_i}$ が最大の i に対して開設か削除, もしくはその両方を実行する.

3.5.2 計算機実験

1.52 近似と改善案のスケーリングパラメータ δ の値は, 解析において近似比率が最も良くなる $\delta = 1.504$ を用いている. ここでは, δ を変化した場合の性能評価を行う.

入力データとして, 施設と利用者を座標 $(0, 0)$ から $(500, 500)$ の間にそれぞれランダムに 100 個ずつ ($n_f = n_c = 100$) 発生し, 開設コストはすべての施設で一定 ($f_i = 1000$), 接続コストは施設と利用者のユークリッド距離としたものを用いる. 近似比率は, それぞれの δ において実験を 20 回行ったときの平均値を用いる. この結果を図 7 に示し (横軸は δ の値を表し, 縦軸は $\delta = 1.504$ の近似比率に対する比率 (倍率) の平均を表している), 表 2 に δ が 1.0 から 1.4 の間の図 7 の詳細を表す.

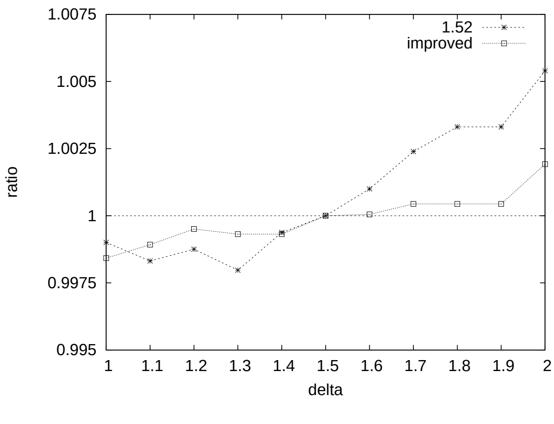

図 7. パラメータ δ による性能評価

表 2. 図 7 の詳細の一部

	1.52 近似		改善案	
δ	平均	最悪	平均	最悪
1.0	0.99901	1.02409	0.99842	1.00000
1.1	0.99831	1.01530	0.99892	1.00532
1.2	0.99875	1.01530	0.99951	1.00000
1.3	0.99797	1.01530	0.99932	1.00000
1.4	0.99938	1.01530	0.99932	1.00000

図 7 より, 解析では $\delta = 1.504$ とした場合に近似比率が最も良くなるが, この入力データでは δ が 1.504 より小さい $\delta = 1.3$ の場合の方が近似比率が良いことがわかる. 改善案については, $\delta = 1.0$ とした場合, つまりコストスケーリングを行わずに初期解を求めた場合の方が近似比率は良くなることがわかる.

4 実験的性能評価

上記の 4 つの近似アルゴリズムに対して様々な計算機実験を行い, それぞれの近似アルゴリズムの性能評価を行う. 最適解の値は, (LP) を線形計画ソフト XPRESS-MP で解くことで得られた目的関数値を用いる. 入力データは,

- 施設と利用者は, 座標 $(0, 0)$ から $(500, 500)$ の間でランダムに 100 個ずつ発生,
- 開設コストは, 1 から 1000 の間のランダムな正整数,
- 施設と利用者の接続コストは, 2 点間 (施設と利用者の間) のユークリッド距離

としたものを用い, この入力データ (以下, **標準入力データ**) の一部を変更することで様々な実験を行う. 以下の図の縦軸は, それぞれの横軸の値で実験を 20 回行ったときの近似比率の平均値を表している.

4.1 計算機実験

実験 1: 施設数と利用者数による影響

- 施設数 n_f, 利用者数 n_c を増加する場合

  標準入力データの n_f, n_c をそれぞれ同じ割合で増加した場合の性能評価を行う. この結果を図 8 に示す (横軸は施設数, 利用者数を表している).

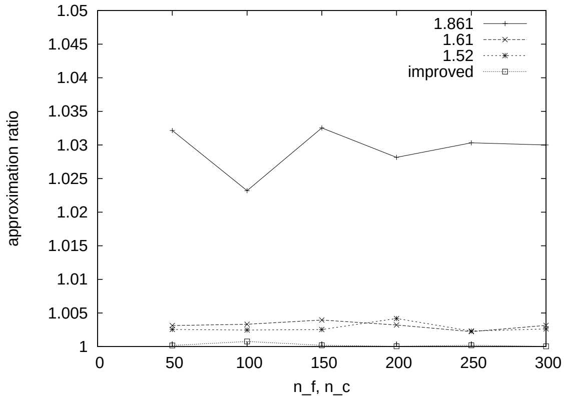

図 8. n_f, n_c を増加した結果

- 利用者数 n_c のみを増加する場合

  現実的な環境においては, 利用者数が施設数より多くなる場合が一般的である. そのため, 標準入力データの n_f を $n_f = 100$ と固定して, n_c のみを増加した場合の性能評価を行う. この結果を図 9 に示す (横軸は利用者数を表している).

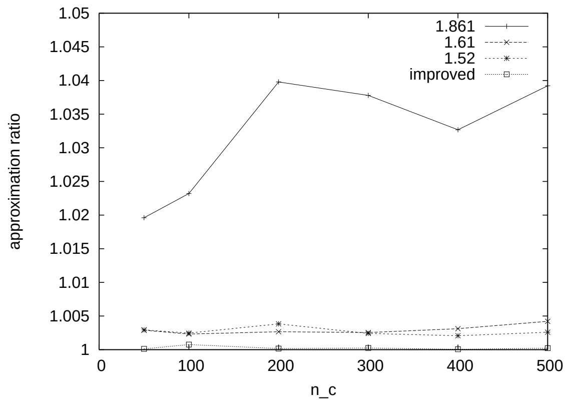

図 9. n_c のみを増加した結果

図 8, 9 より, 1.861 近似は利用者数 n_c が多い場合, 性能がわずかに悪くなるが, その他のアルゴリズムは施設数と利用者数による影響を受けないことがわかる.

実験 2: 開設コストと接続コストによる影響

現実的な環境においては, 開設コストまたは接続コストのどちらかが極端に大きい場合もある. そのため, どちらか一方のみを増加して性能評価を行う.

- <u>開設コスト f_i を増加する場合</u>

  標準入力データの開設コスト f_i を, ランダムではなくすべての施設で一定とし, この値を増加した場合の性能評価を行う. この結果を図 10 に示す (横軸は開設コストを表している).

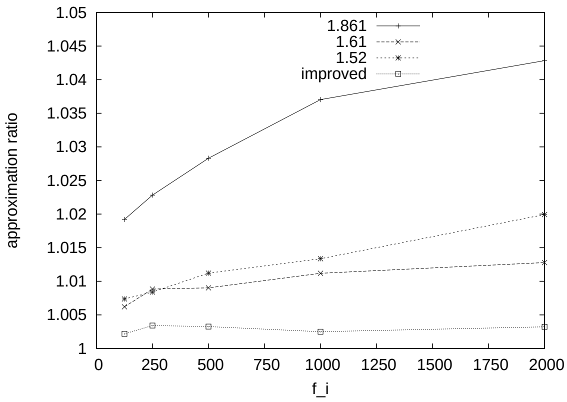

図 10. 開設コスト f_i を増加した結果

- <u>接続コストを増加する場合</u>

  座標範囲を広くすることで接続コストが大きい場合を考える. 座標 $(0, 0)$ から (x, x) までのことを便宜上, 最大座標 x と呼ぶ. このとき, 標準入力データの開設コストを一定 $(f_i = 1000)$ として, 最大座標を 500 ではなくさらに増加した場合の性能評価を行う. この結果を図 11 に示す (横軸は最大座標を表している).

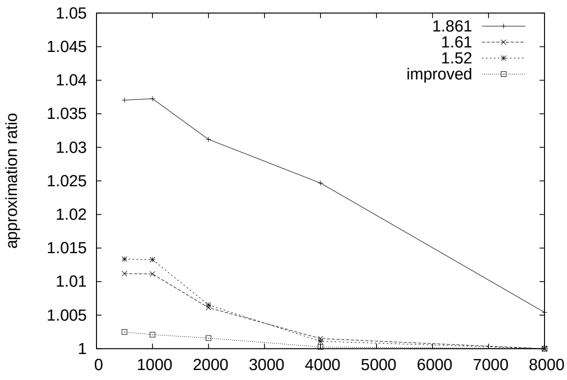

図 11. 接続コストを増加した結果

図 10, 11 より, 開設コストによる影響は接続コストの影響に比べて大きいことがわかる. また, 開設コストを一定にすると改善案も性能が悪くなることがわかる. これは, 初期解で開設される施設の多くが最適な開設施設とは異なるため, その後の貪欲改善により最適な開設施設に改善されにくいことが原因であると思われる.

実験 3: 接続コストを最短パスの長さとする場合の影響

接続コストの取り方を, ユークリッド距離ではないもので, 距離の性質を満たすものとした場合を考える. すなわち, 施設と利用者の 2 部グラフのそれぞれの辺に 0 から 707($500 \times \sqrt{2} \approx 707$) の間のランダムな正整数を与え, このときの 2 部グラフの最短パス長を接続コストとする. このように標準入力データの接続コストを変更し, n_f, n_c を同じ割合で増加した場合の性能評価を行う. この結果を図 12 に示す (横軸は n_f, n_c を表している).

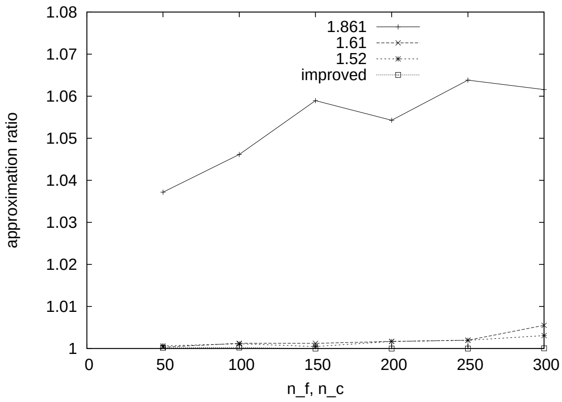

図 12. 接続コストを最短パス長とした結果

図 12 より, 1.861 近似はかなり性能が悪くなることがわかる. これは, 1.861 近似では接続を変更できないことが大きく影響したためであり, 他のアルゴリズムでは接続を変更することで近似比率の良い解が求まると思われる.

4.2 計算機実験のまとめ

本稿の計算機実験では, 理論的な近似比率よりもかなり良い近似比率を得ることができた. また, 削除を加えた改善案は, 他のアルゴリズムよりもかなり性能が良く, 最適解もしくはそれにかなり近い解を求めることができた.

1.52 近似は, 解析では 1.61 近似より良い近似比率となるが, 本稿の実験では 1.61 近似よりも悪い近似比率となる場合があった. これは, 1.52 近似の初期解で開設される施設が, 最適解の開設施設とは大きく異なることが原因であると思われる.

本稿の計算機実験から, 施設数と利用者数を変えても性能にあまり変化はみられないが, 開設コストをすべての施設で一定かつ大きな値とした場合には性能が悪くなることがわかった. また, 接続コストを 2 部グラフの最短パスの長さとした場合には, 1.861 近似の性能はかなり悪くなることがわかった.

5 おわりに

様々な入力データにおいて, 上記の 4 つの近似アルゴリズムの実験的性能評価を行った.

本稿の入力データに対する近似比率は, 解析による値よりもかなり良くなってしまったため, 最悪の近似比率となるインスタンスを考え, このインスタンスに対しても性能評価を行うことが今後の課題である. また, 1.463 近似より良いアルゴリズムは存在しないと考えられている ([3]) ため, 新たな手法のアルゴリズムの提案により近似比率を 1.463 に近づけることも今後の重要な課題である.

謝辞

本研究は, 一部, 中央大学理工学研究所, 21 世紀 COE プログラム, 文部省科学研究費補助金および電気通信普及財団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 facility location and k-median problems, *FOCS*, 1999, pp. 378–388.

[2] F.A. Chudak: Improved approximation algorithms for uncapacitated facility location, *IPCO*, 1998, pp. 180–194.

[3] S. Guha, S. Khuller, Greedy strikes back: Improved facility location algorithms, *SODA*, 1998, pp. 649–657.

[4] K. Jain, M. Mahdian and A. Saberi: A new greedy approach for facility location problems, *STOC*, 2002, pp. 731–740.

[5] K. Jain and V.V. Vazirani: Primal-dual approximation algorithms for metric facility location and k-median problems, *FOCS*, 1999, pp. 2–13.

[6] M. Korupolu, G. Plaxton, and R. Rajaraman: Analysis of a local search heuristic for facility location problems, *SODA*, 1998, pp.1–10.

[7] M. Mahdian, E. Markakis, A. Saberi and V.V. Vazirani: A greedy facility location algorithm analyzed using dual fitting, *5th International Workshop on Randomization and Approximation Techniques in Computer Science, Vol. 2129*, 2001, pp. 127–137.

[8] M. Mahdian, Y. Ye and J. Zhang: Improved approximation algorithms for metric facility location problems, http://www.mit.edu/˜mahdian/pub.html, 2002.

[9] D.B. Shmoys, E. Tardos and K.I. Aardal: Approximation algorithms for facility location problems, *STOC*, 1997, pp. 265–274.

[10] M. Sviridenko: An Improved Approximation algorithm for the metric uncapacitated facility location problems, *IPCO*, 2002, pp. 230–239.