アルゴリズム 71－7
（2000．1．17）

容量制限付き k-center 問題の近似解法の実験的評価

小谷 智昭† 山口 一章‡ 増田 澄男‡
† 神戸大学大学院自然科学研究科 ‡ 神戸大学工学部

あらまし k-center 問題は NP 最適化問題の中でも重要なものの一つである．容量制限付き k-center 問題は，k-center 問題のより現実的な設定である．容量制限付き k-center 問題に対する，これまでに知られる最も良い近似アルゴリズムの近似率は 6 であるが，これは実用的でないと思われる．本稿では容量制限付き k-center 問題に対する 3 つの発見的手法を示す．そのうちの 2 つを組み合わせることで，従来手法よりも高速でかつ良い解が得られることを実験により示す．

Experimental Performance of an Approximation Algorithm for the Capacitated k-Center Problem

Tomoaki Kotani†, Kazuaki Yamaguchi‡, and Sumio Masuda‡

† Graduate School of Science and Technology, Kobe University
‡ Faculty of Engineering, Kobe University

Abstract The k-center problem is one of the important NP optimization problems. The capacitated k-center problem is its more practical variation. Although a 6-approximation algorithm has been proposed, it is considered to be ineffective. In this paper, we present three heuristic algorithms for the capacitated k-center problem. Experimental results show that we can get better solutions more efficiently, by executing two of our methods sequentially.

1. まえがき

世の中には数多くの最適化問題が存在する．その中でも、NP 完全問題と同等の難しさを持つものは NP 最適化問題と呼ばれる．NP 最適化問題に対しては，最適解を求めるよりも近似解を求める方がより実用的である．

施設配置問題は，与えられた領域内において，利用者にとって都合の良い施設設置場所を求める問題であり，NP 最適化問題の中でも重要な問題の一つである．その中でも特に k-center 問題は最も基本的な問題で，多くの研究が行われている [1],[2],[3]．この問題は，次のように記述できる．

入力：完全グラフ $G=(V,E)$，2 頂点間の距離 $d(\cdot,\cdot)$，及び自然数 k $(k<|V|)$．ただし任意の $v,w,u\in V$ に対して

$$d(v,v)=0,$$
$$d(v,w)=d(w,v),$$
$$d(v,w)+d(w,u)\geq d(v,u)$$

が成立しているとする．

問題：評価値

$$\max_{v\in V}\ \min_{c\in C}\ d(v,c)$$

が最小となるような，要素数 k 以下の V の部分集合 C を求めよ．

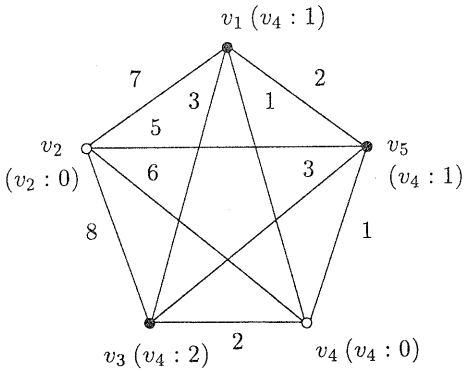

図1：k-center 問題の例（$k=2$）

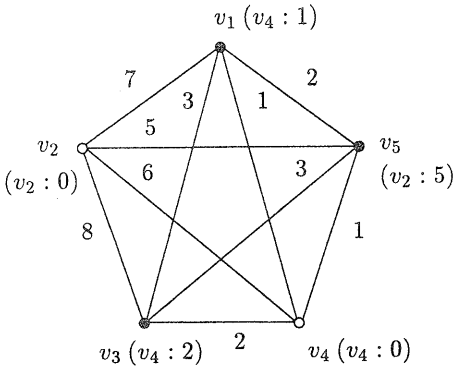

図2：容量制限付き k-center 問題の例
$(k=2,\ L=3)$

以降，この部分集合 Cに含まれる頂点のことをセンターと呼ぶ．この問題では，各頂点はそれぞれ最寄りのセンターに割り当てられる．上記の評価値は，各頂点から割り当てられたセンターへの距離の最大値である．

図1に k-center 問題と解の例を示す．図中，センターは○，その他の頂点は●で表している．また，各頂点について，割り当てられたセンター，及びそこへの距離を示している．この場合，評価値は2となる．

k-center 問題は，例えば県内に k 個の消防署を設置するとき，消防署から最も離れた地域から最寄りの消防署までの距離を評価基準として，出来るだけ良い配置場所を考えるという状況を数学的に定式化した問題であるといえる．またこの問題は，ネットワーク上の分散システムのサーバ配置の決定等にも利用できると思われる．

文献 [1] では，常に近似率2以下の近似解を出力するような多項式時間のアルゴリズムが存在すること，及び任意の $\epsilon>0$ に対し，近似率 $(2-\epsilon)$ 以下の近似解を求める問題はNP困難であることが示されている．

本稿では，k-center 問題の，より現実的な設定の一つである容量制限付き k-center 問題を扱う．この問題では，各センターに割り当てることが出来る頂点の数は L 個以下である．

容量制限付き k-center 問題の記述を以下に示す．

入力：完全グラフ $G=(V,E)$，2頂点間の距離 $d(\cdot,\cdot)$，自然数 k $(k<|V|)$ と L.

問題：Vの部分集合 Cと割当 $\phi: V \to C$ のうち，評価値

$$\max_{v\in V} d(v,\phi(v))$$

を最小にするようなものを求めよ．
ただし各 $c\in C$に対し，
$|\{v \mid \phi(v)=c\}| \le L$ である．

この評価値は，各頂点から割り当てられたセンターへの距離の最大値である．図2に例を示す．センター及びその他の頂点の表記は図1と同じである．

図2のグラフは図1と同じものであるが，$L=3$ という条件を与えている．そのため，頂点 v_5の最寄りのセンターは v_4であるが，もう一つのセンターである v_2に割り当てられることになる（v_4には頂点 v_1, v_3, v_4 が割り当てられているため）．

この問題を前述の消防署の配置で例えてみる．各地域が最寄りの消防署に頼るとすると，地域間の距離のばらつきにより特定の消防署への依存が強くなることが起こり得る．このとき，それら特定の消防署の近隣で同時多発的に火事が起こったとすると，全ての火事に対応しきれなくなる．このようにいくつかの消

防署の負担が極端に大きくなるのを防ぐために，一つの消防署が対応できる地域の数を考え，その上で消防署の配置を考えるという状況を表しているのがこの容量制限付き k-center 問題であるといえる．

容量制限付き k-center 問題に対し，まず近似率 10 のアルゴリズムが示された [4]．その後はあまり研究が行われておらず，現在，最も良い近似アルゴリズムは近似率 6 のアルゴリズムである [5]．

本稿では，容量制限付き k-center 問題に対し，いくつかの発見的手法を示す．そして，従来法との比較実験を行い，実用性について検討する．

2. 従来手法

容量制限付きの k-center 問題の近似解法として，文献 [5] では近似率 6 のアルゴリズムが示されている．その手順を以下に示す．

1. グラフ $G=(V,E)$ の辺を重さの昇順にソートしたものを $e_1, e_2, \cdots, e_m$ とし，各 i $(i=1,2,\cdots,m)$ について，重さ $w(e_i)$ 以下の辺のみからなる G の部分グラフを $G_i=(V,E_i)$ とする．ただし G_i は重みのないグラフとする．G_i において，2 頂点間の距離はその間の道上の辺の数の最小値で表される．これを $dist(\cdot,\cdot)$ とする．

2. 以下の処理 3〜7 を，$i=1,2,\cdots,m$ について，解が得られるまで実行する．

3. G_i を連結成分ごとに分け，各連結成分 $G_i^c=(V_c,E_c)$ に対し，以下の 4〜6 の操作を行う．

4. G_i^c において，次の条件を満たす極大な頂点集合 $M \subseteq V_c$ を求める．

   - 任意の 2 頂点間の距離が 3 以上．
   - 各頂点について，そこからの距離がちょうど 3 である頂点が 1 つ以上存在する．

   M の各頂点を *monarch* と呼び，ここにセンターを置く．

   M の求め方を以下に示す．まず V_c から任意の 1 点を選択し，これを *monarch* m_0 とする．このとき頂点 m_0 にマークをつけておく．次に m_0 からの距離が 1 または 2 である頂点のうち，マークされていないものの集合 $E_{12}(m_0)$ を求め，各頂点をマークする．さらに m_0 からの距離が 3 である頂点の中で，マークされていないものを 1 つ選び，m_1 とする．このとき m_0 を m_1 の親と呼ぶ．明らかに，親子である *monarch* 間の距離はちょうど 3 である．m_0 からの距離が 3 の頂点で，マークされていないものが残っていれば，新たに 1 つ選んで *monarch* とする．そのような頂点がなければ，m_1 との距離が 3 の頂点から *monarch* を選択する．このような操作を，全ての頂点がマークされるまで続ける．

5. G_i^c に関する 2 部グラフ

   $$G_i^{c\prime}=(M,V_c,E_c'),$$
   $$E_c'=\{(m,v)|\ m\in M, v\in V, dist(m,v)\le 2\}$$

   を構成する．さらに頂点 s,t，辺集合 $E_c''=\{(s,m),(v,t)|\ m\in M, v\in V\}$ を加え，各辺 e に正の実数の容量 $cap(e)$ 及び非負の実数の費用 $cost(e)$ を割り当てたネットワーク N_i^c を

   $$N_i^c=((M,V_c,E_c'\cup E_c''),cap,cost,s,t)$$

   $\forall m\in M$，$\forall v\in V_c$ について

   $$cap(s,m)=L,$$
   $$cap(m,v)=cap(v,t)=1,$$
   $$cost(m,v)=1\ \ (v\notin E_{12}(m)),$$
   $$cost(m,v)=0\ \ (v\in E_{12}(m)),$$
   $$cost(s,m)=cost(v,t)=0$$

   とし，N_i^c についての最小費用フローを求める．フローが流れる各辺 (m,v) について，頂点 v を m に割り当てる．

6. 集合 $\{m\} \cup E_{12}(m)$ の頂点のうち，5. の操作によりセンターに割り当てられなかったものの集合を $unassigned(m)$ とする．$unassigned(m)$ に対しては，

   - $unassigned(m)$ 中のいくつかの頂点にセンターを置き，それらに割り当てる．
   - m の親の $monarch$ に受け渡す．

   といったことを行う（詳細略）．

   なお $monarch$ m が子の $monarch$ から受け取った頂点集合については，さらに親の $monarch$ に渡すということは出来ず，必ず m 自身で割り当てを行わなければならない．このときの割り当て方は，既にセンターを置いてある m 自身に割り当てるか，$E_{12}(m)$ の頂点にセンターを置いて，そこに割り当てるかのいずれかとなる．

7. 以上 3〜6 の処理を終え，全ての頂点をセンターに割り当てたときに，設置したセンターの数が k 個以下であれば，解としてセンターの集合 C と割当 ϕ を出力して終了する．

この方法は，近似率の上限を理論的に保証してはいるものの，その値は実用的ではないと考えられる．

3. 提案手法

ここでは本稿で提案する手法についての説明を行う．従来手法では各センターが自分自身でなく他のセンターに割り当てられることが起こり得るが，これは不自然であると思われる．そのため全ての提案手法において，各センターは自分自身に割り当てられるものとする．

はじめに手法の説明の準備として，ボトルネック割当問題について述べる．

3.1. ボトルネック割当問題

2 部グラフ $G = (V_1, V_2, E)$ と各辺 e の重み $cost(e)$ が与えられたとき，G の最大マッチングのうち，マッチングに用いられている辺の $cost$ の最大値が最小のものを求める問題をボトルネックマッチング問題と呼ぶ．本稿では，V_1 と V_2 の頂点の対応が一対一でなく，V_2 の最大 γ 個の頂点が V_1 の一つの頂点に対応することを許すという場合を，ボトルネック割当問題と呼ぶことにする．

本稿で提案する手法においては，この問題を解く必要がある．ここで 2 部グラフ G を

$$G = (C, P, E),$$
$$C = \{\, c \mid c \in V,\ c \text{ はセンター} \,\},$$
$$P = \{\, p \mid p \in V - C \,\},$$
$$E = \{\, (c, p) \mid c \in C,\ p \in P \,\}$$

と定義する．このとき，入口 s，出口 t，辺集合 $E' = \{\, (s, c), (p, t) \mid c \in C, p \in P \,\}$ を加え，各辺 e に正の実数の容量 $cap(e)$ 及び非負の実数の費用 $cost(e)$ を割り当てたネットワーク N を

$$N = ((C, P, E \cup E'), cap, cost, s, t)$$
$$\forall c \in C,\ \forall p \in P \text{について}$$
$$cap(s, c) = L - 1,$$
$$cap(c, p) = cap(p, t) = 1,$$
$$cost(c, p) = d(c, p),$$
$$cost(s, c) = cost(p, t) = 0$$

とすると，プライマルデュアル法 [6] を用いて N についての最小費用フローを求めることにより，G のボトルネック割当を得ることが出来る．このとき得られた割当が，センターとそこに割り当てられる頂点との関係を表している．以後この手続きを，BA（Bottleneck Assignment）と表すことにする．

3.2. 提案手法 1

第一の手法は，容量制限の無い k-center 問題の近似アルゴリズム [1] を利用する方法である．まずこのアルゴリズムを以下に示す．

1. 乱数で V から頂点を一つ選択し，c_1 とする．

2. 各 $i=2,\cdots,k$ について，全ての頂点 $v \in V-\{c_1,c_2,\cdots,c_{i-1}\}$ に対し

$$\min_{c\in\{c_1,c_2,\cdots,c_{i-1}\}} d(v,c)$$

を求め，この値が最大となる v を c_i とおく．

この方法により近似率2倍以内の解が得られることが証明されている．

次に上記手法を用いた提案手法の手順を以下に示す．

1. 容量制限付き k-center 問題の入力を容量制限の無い k-center 問題の入力とみなし，k-center 問題の近似率2倍のアルゴリズムを用いて解（センターの集合 C）を求める．

2. その解に対して BA を用いて，容量制限付き k-center 問題の解を得る．

3.3. 提案手法 2

第二の手法は，$k=\alpha+1$ の時の容量制限付き k-center 問題の解から $k=\alpha$ の時の解を求めることを繰り返すというヒューリスティックである．具体的には次のようになる．

$k=\alpha+1$ の容量制限付き k-center 問題の近似解におけるセンター集合を $C=\{c_1,c_2,c_3,\cdots,c_{\alpha+1}\}$ とする．このとき

$C-\{c_1\}$,
$C-\{c_2\}$,
$C-\{c_3\}$,
$\vdots$
$C-\{c_{\alpha+1}\}$

をそれぞれセンター集合とした場合の BA の出力を比べ，最も良いものを $k=\alpha$ の近似解として出力する．

頂点集合 V を $k=|V|$ の解とし，本手法を連続して $|V|-\alpha$ 回用いることによって必要な k の値 $(=\alpha)$ の解を求める．しかしこの時 $|V|$ から $\alpha+1$ までの各 k の値に対して $k-1$ 回の BA を用いることになる．BA においてはプライマルデュアル法を使用しているため，計算時間が膨大になってしまう．そのため計算時間を少しでも短くするために，何らかの工夫をする必要がある．

$k=|V|$ から $k=\alpha+1$ の間において求める必要があるのはセンター集合と評価値だけである．すなわち各頂点がどのセンターに割り当てられるかという情報は必要ない．そこで実際には次のような手続きを用いる．

$k=\beta$ $(\alpha+1\le\beta\le|V|)$ の容量制限付き k-center 問題の近似解におけるセンター集合を $C=\{c_1,c_2,c_3,\cdots,c_\beta\}$ とする．このとき $k=\beta-1$ の時の近似解におけるセンター候補は $C-\{c_x\}$ $(1\le x\le\beta)$ となる．ここで2部グラフ G' を

$$G'=(C-\{c_x\},V-(C-\{c_x\}),E),$$
$$E=\{\,(u,v)\mid u\in C-\{c_x\},\ v\in V-(C-\{c_x\})\}$$

とし，E 中の全ての辺を長さ（d の値）の昇順にソートする．また，$m=|E|$ としたとき，$i=0,1,2,\cdots,m$ について，G' の部分グラフ $G'(i)$ を G' の全ての頂点と E の i 番目までの辺からなるものとする．すなわち $G'(0)$ は G' の全ての頂点のみからなるグラフとなる．このとき $i=1$ から m について，$G'(i)$ において $V-(C-\{c_x\})$ の全頂点を $C-\{c_x\}$ のいずれかの頂点に割り当てることが出来れば，i 番目の辺の長さが，$C-\{c_x\}$ をセンター集合とした場合の評価値となる．

3.4. 提案手法 3

第三の手法は BA を用いて解を改良するヒューリスティックである．解の改良は，解であるセンター集合においていくつかのセンターを変更することで行う．これにより，より良い評価値の解を得ることが期待できる．

容量制限付き k-center 問題の解を $C=\{c_1,c_2,c_3,\cdots,c_k\}$ とする．このときあるセンター c_i について，c_i に割り当てられた（c_i 以外の）頂点の集合を $P_i=\{p_{i1},p_{i2},p_{i3},\cdots,p_{iq}\}$ $(q\le L-1)$ とする．そして c_i から P_i 中

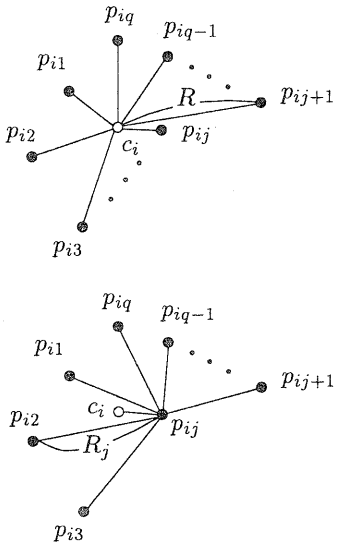

図3：RとR_jの関係

の頂点への距離の最大値

$$\max_{1\leq j\leq q} d(c_i, p_{ij})$$

をRとする．次に各p_{ij}を基準とし，p_{ij}からc_i及び$P_i - \{p_{ij}\}$中の頂点への距離の最大値をR_jとする（図3）．このとき$R_j < R$であれば，頂点集合$\{c_i\} \cup P_i$はc_iよりもp_{ij}の近くに分布しているということになるので，センターをc_iからp_{ij}に変更する．この処理を全てのセンターに対して行った後，再びBAを用いて割り当てを求める．

以上の手順を解が変化しなくなるまで繰り返すことにより，解の改良を行う．

4. 計算機実験

提案した手法について，従来手法との比較のための計算機実験を行った．提案手法の組合せにより，以下(A)～(H)の8通りの場合について考えた．

(A)：従来手法
(B)：乱数によりランダムにセンターを決定し，それに対してBAを適用．
(C)：提案手法1
(D)：提案手法2

(E)：(A)で得られた解に対し，提案手法3を適用．
(F)：(B)で得られた解に対し，提案手法3を適用．
(G)：(C)で得られた解に対し，提案手法3を適用．
(H)：(D)で得られた解に対し，提案手法3を適用．

実験における入力はグラフ$G(V,E)$の頂点数n，2頂点間の距離$d(\cdot,\cdot)$，自然数kとLである．使用したグラフは4通りであり，各グラフの各辺はそれぞれ1～30の長さを持っている．さらにkとLの組合せを2種類にし($k = L = \lceil\sqrt{n}\,\rceil$及び$k = L = \lceil\sqrt{2n}\,\rceil$)，計8パターンの入力について実験を行った．以下に入力(1)～(8)を示す．

[$k = L = \lceil\sqrt{n}\,\rceil$]

(1)：$n = 50$, $k = 8$, $L = 8$
(2)：$n = 100$, $k = 10$, $L = 10$
(3)：$n = 150$, $k = 13$, $L = 13$
(4)：$n = 200$, $k = 15$, $L = 15$

[$k = L = \lceil\sqrt{2n}\,\rceil$]

(5)：$n = 50$, $k = 10$, $L = 10$
(6)：$n = 100$, $k = 15$, $L = 15$
(7)：$n = 150$, $k = 18$, $L = 18$
(8)：$n = 200$, $k = 20$, $L = 20$

実験では，入力に対して各手法を用いたときの評価値及び実行時間を計測した．使用計算機はAT互換機（Pentium II 450MHz），使用プログラミング言語はC言語である．

5. 実験結果及び考察

実験結果を以下に示す．表1は$k = L = \lceil\sqrt{n}\,\rceil$とした入力(1)～(4)に対する結果であ

表1. $k = L = \lceil \sqrt{n} \rceil$

入力	(1)	(2)	(3)	(4)
(A)	10	20	27	28
従来	4	695	899	6903
(A)'	10	13	12	18
(A) + BA	4	702	909	6931
	(no)	(7)	(10)	(28)
(B)	14	13	17	17
ランダム	1	4	10	27
(C)	8	11	12	15
提1	1	5	11	25
(D)	7	9	9	14
提2	77	755	7107	38585
(E)	6	7	10	16
(A) +提3	5	715	928	7013
	(1)	(20)	(29)	(110)
(F)	9	9	10	17
(B) +提3	1	12	42	82
	(1)	(8)	(32)	(55)
(G)	8	7	9	15
(C) +提3	2	20	42	80
	(1)	(15)	(31)	(55)
(H)	6	8	9	14
(D) +提3	78	760	7117	38613
	(1)	(5)	(10)	(28)

表2. $k = L = \lceil \sqrt{2n} \rceil$

入力	(5)	(6)	(7)	(8)
(A)	10	8	15	28
従来	7	261	1185	5816
(A)'	10	8	15	18
(A) + BA	7	262	1185	5846
	(no)	(no)	(no)	(30)
(B)	8	10	16	18
ランダム	0	3	13	35
(C)	6	6	9	14
提1	0	3	23	34
(D)	6	5	8	13
提2	138	707	7094	46634
(E)	7	6	9	15
(A) +提3	8	266	1243	5901
	(1)	(5)	(58)	(85)
(F)	5	8	10	14
(B) +提3	1	14	74	89
	(1)	(11)	(61)	(54)
(G)	6	6	9	14
(C) +提3	1	9	48	136
	(1)	(6)	(25)	(102)
(H)	6	5	8	13
(D) +提3	138	715	7106	46668
	(0)	(8)	(12)	(34)

り，表2は $k = L = \lceil \sqrt{2n} \rceil$ とした入力 (5)～(8) に対する結果である．1段目の数値は評価値を表しており，2段目の数値は実行時間（単位：秒）を表している．(E)～(H) の3段目の () 内の数値は，提案手法3にかかった実行時間を表している．また，従来手法ではセンターが自分自身に割り当てられない場合が起こり得る．そこで参考のために従来手法で得られた解に BA を用い，全てのセンターを自分自身に割り当てるようにした場合の結果を (A)' として示す．3段目の () 内の数値は BA にかかった時間であり，(no) というのは元々全てのセンターが自分自身に割り当てられていた場合を表している．

まず (A)～(D) の結果について考える．表1，2より，全ての入力に対して，提案手法1, 2の方が従来手法よりも良い評価値を示していることが分かる．センターをランダムに選んだ場合でさえ，全般的に従来手法よりも評価値が良くなっている．ただ，この場合はセンターの初期配置に強く依存しているため，一概に従来手法より良いとはいえない．また従来手法の解とそれに BA を適用したものとの比較より，やはりセンターは自分自身に割り当てる方が自然であり，評価値も良くなると思われる．$k = L = \lceil \sqrt{n} \rceil$ の場合と $k = L = \lceil \sqrt{2n} \rceil$ の場合との比較では，やはり後者の方がセンターの選択，頂点の割り当て共に自由度が高いため，全般的に評価値が良くなっている．(A)～(D) の中で最も評価値の良いものは (D) の

提案手法2であるが，頂点数の増加にともない計算時間が膨大となるため，実用化には不向きである．しかし，この手法は近似アルゴリズムとしては何倍かの近似率を保証しているかもしれず，近似精度の理論的な解析が今後の課題となる．評価値，実行時間ともに良い値となっているのは，(C) の提案手法1である．この手法は実行時間がセンターをランダムに選んだ場合とあまり変わらず，評価値も提案手法2に近いものになっている．したがって，実用化という観点では，これらの手法の中で提案手法1が最も良い手法であると思われる．

次に (E)～(H) について考える．これらはそれぞれ (A)～(D) の解に提案手法3を適用したものであるが，全て元の解より評価値が良くなっている．評価値が変わらないものもあるが，これらについては各頂点と割り当てられたセンターとの距離の総和が小さくなっている．しかも提案手法3の実行時間はさほど大きくない．これより，センターを変更するという操作は非常に効果的であることが分かる．

今回実験した手法の中では，提案手法1と3を組み合わせたものが，評価値，実行時間ともに最も実用的であるといえる．今後，タブーサーチ等のメタヒューリスティックを用いてセンターを変更していくといった手法が，良い解を得るための手法の一つとして期待できる．

6. まとめと今後の課題

本稿では容量制限付き k-center 問題に対するいくつかの発見的手法を提案し，従来手法 [5] とともに計算機実験を行った．その結果，

- 6倍の近似率を保証している従来手法は，評価値，実行時間共に実用的ではない．
- k-center 問題の近似率2倍のアルゴリズムを利用した提案手法1と，センター変更を行う提案手法3を組み合わせたものが最も実用的である．
- センターを変更するという操作は非常に効果的である．

ということがわかった．

今後の課題としては，

- BA の高速化
- タブーサーチ，遺伝的アルゴリズム，simulated annealing 等のメタヒューリスティックの適用
- 提案手法2の近似精度の理論的解析

等が挙げられる．

参考文献

[1] M.E.Dyer and A.M.Frieze, "A simple heuristic for the p-center problem," *Operations Reserch Letters*, Vol.3, pp.285-288 (1985).

[2] D.Hochbaum and D.B.Shmoys, "A best possible heuristic for the k-center problem," *Mathematics of Operations Research*, Vol.10, pp.180-184 (1985).

[3] J.Plesnik, "A heuristic for the p-center problem in graphs," *Discrete Applied Mathematics*, Vol.17, pp.263-268 (1987).

[4] J.Bar-Ilan, G.Kortsarz and D.Peleg, "How to allocate network centers," *J. Algorithms*, Vol.15, pp.385-415 (1993).

[5] S.Khuller and Y.J.Sussmann, "The capacitated k-center problem,"*Proc. 4th Ann. European Symp. on Algorithms*, Lecture Notes in Comput. Sci. 1136, Springer-Verlag, 152-166 (1996).

[6] 浅野 孝夫，情報の構造 [下]，日本評論社 (1994).