割当制約つき複数ナップサック問題に対する近似アルゴリズムの実験的評価

Experimental Evaluations of Approximation Algorithms for the Multiple Knapsack Problem with Assignment Restrictions

藤井海斗 * 森本尚之 † 宮崎修一 ‡ 岡部寿男 ‡

Kaito Fujii Naoyuki Morimoto Shuichi Miyazaki Yasuo Okabe

概要

複数の電力源がある場合に，電力を効率的に使うためには，電力を家電にうまく割り当てる必要がある．この問題は割当制約つき複数ナップサック問題として定式化できる．本研究では，既存の近似アルゴリズムを実装し，実験的な評価をおこなった．また，それらのアルゴリズムを改良することを目指して，いくつかの変更を提案し，それらについても実験をおこなった．最後に，実用的な規模を想定して例題を生成し，各アルゴリズムの性能を比較した．

1 はじめに

地球温暖化問題が世界全体における課題となっている現在，省エネルギーな電力システムを構築することはますます重要になっている．そのためには，太陽光発電や風力発電などのさまざまなエネルギーを効果的に取り入れることが必要不可欠である．しかし，これらの電源は不安定な場合もあるので，安定な電力と組み合わせなければならない．そこで，個別の電力消費機器それぞれに対して，性質を考慮して適した電源を割り当てることで消費電力を抑える方法がいくつか提案されている [7, 10]．そのためには，機器と電源との組ごとに独立した配電が必要となってしまうことが技術的な課題であったが，近年，回路交換などの技術を用いることで，異なる電源の電力をひとつの送電線で配送する技術が提案・実装されている [6, 11]．よって，機器の電源への割当をうまく決めることの重要性がますます高まっているといえる．

本研究では，複数の電源と家電があるとき，電源の容量の範囲内で利用者の利得を最大化するような割当を探す問題を考える．この問題は，割当制約つき複数ナップサック問題として定式化できることが Miyazaki ら [8] によって示唆されている (詳細は 2 節で説明する)．本研究では，実際の電源割当問題を効率的かつ高精度に解くことを目標として，さまざまな既存のアルゴリズムを実装し，その近似の精度を測定した．

貪欲法は，さまざまな組合せ最適化問題に対して実用的に高い性能を発揮することが広く知られている．一方で，多くの精度保証も知られており，例えばナップサック問題に対しては，貪欲法をうまく使うことで多項式時間近似スキーム (PTAS) が得られる [5]．Dawande ら [2] は，すべてのアイテムについて利得とサイズが等しい場合，割当制約つき複数ナップサック問題に対して貪欲法が 3-近似になることを証明した．一般の割当制約つき複数ナップサック問題に対する理論的な精度保証は知られていないが，実用的には高い性能を見せることが期待できる．

Fleischer ら [4] は一般化割当問題に対して $e/(e-1)$-近似が得られるアルゴリズムを発表した．Feige と Vondrák [3] は，それを改善してある小さな定数 ρ に

* 京都大学 情報学研究科 知能情報学専攻
Department of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University.
fujii@ml.ist.i.kyoto-u.ac.jp

† 京都大学 物質—細胞統合システム拠点
Institute for Integrated Cell-Material Sciences, Kyoto University.
nmorimoto@icems.kyoto-u.ac.jp

‡ 京都大学 学術情報メディアセンター
Acedemic Center for Computing and Media Studies, Kyoto University.
{shuichi,okabe}@media.kyoto-u.ac.jp

ついて $(e/(e-1)-\rho)$-近似を達成した．割当制約つき複数ナップサック問題は一般化割当問題の特別な場合であるため，同様の $(e/(e-1)-\rho)$-近似が達成できる．割当制約つき複数ナップサック問題に対するこれよりよい近似比のアルゴリズムは知られていない．

割当制約のない複数ナップサック問題に対しては，Chekuri と Khanna [1] が PTAS の存在とそれが最良であることを証明している．そのため，割当制約つき複数ナップサック問題についても PTAS よりよい近似は不可能である．これは，多項式時間で実現可能な最良の近似精度に対する，現在知られているなかで最も大きい下限である．

Match-and-FPTAS は，Miyazaki ら [8] によって 2015 年に提案された．これは，Nutov ら [9] が提案した 2-近似アルゴリズムを利用したものである．任意の割当制約つき複数ナップサック問題について，割当可能なすべてのアイテム a とナップサック b の組について容量とサイズの比が k 以上であるとき，Match-and-FPTAS は $(1+\frac{2}{k+1}+\epsilon)$-近似を達成する．この近似比は，$k$ が大きくなると $1+\epsilon$ に漸近する．実際の問題例では，電源の容量は家電一つが消費する電力よりはるかに大きいことが期待できるため，k の値が大きく，Match-and-FPTAS がよい性能を発揮すると考えられる．

本研究では，貪欲法と Match-and-FPTAS を実装し，ランダムに生成したデータセットに対して，その性能を調べた．ソルバによって得た最適解と比較したところ，どちらのアルゴリズムもほとんど最適な解を返していることがわかった．計算時間は，貪欲法が Match-and-FPTAS に比べて約 100 倍速かった．

また，それらのアルゴリズムを改良することを目指して，いくつかの変更を提案し，それらについても実験をおこなった．その結果，小さな変更を加えることによって，どちらのアルゴリズムも実用的な性能が向上することがわかった．小さいサイズの例題に対しては Match-and-FPTAS が，大きいサイズの例題に対しては貪欲法がよりよい結果を出した．

最後に，実用的な規模を想定して例題を生成し，各アルゴリズムの性能を比較したところ，Match-and-FPTAS に比べて貪欲法がよい結果だった．

2 問題の定式化

2.1 割当制約つき複数ナップサック問題

割当制約つき複数ナップサック問題を次のように定義する．まずアイテム全体の集合を $I=\{a_1,\cdots,a_n\}$，ナップサック全体の集合を $J=\{b_1,\cdots,b_m\}$ とする．このとき，それぞれのアイテム $a\in I$ には利得 $p(a)$ とサイズ $l(a)$ が，それぞれのナップサック $b\in J$ には容量 $c(b)$ が決まっているものとする．また，アイテム $a\in I$ とナップサック $b\in J$ に対して，a が b に割当可能かどうかがあらかじめ決まっており，割当可能ならば，またその場合のみ $(a,b)\in E$ となっているものとする．ナップサックの容量を超えないようにアイテムをナップサックに割り当て，割り当てられたアイテムの利得の合計を最大化したい．この問題は，次のような整数計画問題として定式化できる．

$$
\begin{aligned}
\text{Maximize} \quad & \sum_{(a,b)\in E} p(a)x_{(a,b)} \\
\text{subject to} \quad & \sum_{a\in\Gamma(b)} l(a)x_{(a,b)} \le c(b),\ \forall b\in J \\
& \sum_{b\in\Gamma(a)} x_{(a,b)} \le 1,\ \forall a\in I \\
& x_{(a,b)} \in \{0,1\},\ \forall (a,b)\in E
\end{aligned}
$$

ただし，$\Gamma(a)=\{b\mid (a,b)\in E\}$，$\Gamma(b)=\{a\mid (a,b)\in E\}$ とする．$x_{(a,b)}$ はアイテム a をナップサック b に入れるかどうかを表す変数であり，$x_{(a,b)}=1$ のとき入れる，$x_{(a,b)}=0$ のとき入れないことを意味する．割当制約は，I と J を頂点集合とする二部グラフで表すことができ，完全二部グラフのとき割当制約のない複数ナップサック問題になる．

2.2 電力の割当問題への応用

例えば水力発電，風力発電，太陽光発電など複数の電源があり，それらは電力供給の安定性や価格が異なっているものとする．一方，各家電は安定性や価格により，どの電源から電力を得ることができるか (もしくは得たいか) が決まっているものとする．また，各家電に消費電力とその家電を使うことによる満足度が，

各電源にその電源から使える電力の容量が定まっているとする．どの家電をどの電源に割り当てるべきかという問題は，各電源に割り当てられた家電の消費電力の合計が容量以下であるという条件のもとで，割り当てられた家電の利得の合計を最大化する問題として定式化できる．

家電ひとつひとつをアイテムに，電源をナップサックに対応させ，家電の消費電力をサイズ，家電を使えることによる満足度を利得，電源の容量をナップサックの容量とみなすと，この問題は割当制約つき複数ナップサック問題と等価になる．

3 既存のアルゴリズム

3.1 貪欲法

貪欲法は，精度保証は知られていないが，一般的なヒューリスティックである．まず，すべてのアイテムを利得とサイズの比 $p(a)/l(a)$ が大きい順にソートしておく．そして，この順番にアイテムを取り出し，それが割当可能でかつアイテムのサイズの分だけ容量が空いているナップサックがないか探していく．このとき，ナップサックはある順序で並んでおり，割当可能かどうかその順で調べていくとする (ナップサックの順序については，4 節の実験のところで述べる)．もし見つかれば，そのアイテムはそのナップサックに割り当てる．もし見つからなければ，そのアイテムはどこにも割り当てないとする．これをすべてのアイテムについておこない，その結果を解として出力する．

このアルゴリズムの計算時間は，最初のソートに必要な $\mathrm{O}(n \log n)$ と割当可能かどうか調べていく過程で必要な $\mathrm{O}(mn)$ の合計 $\mathrm{O}(n \log n + mn)$ である．

Algorithm 1 貪欲法

```
for each a ∈ I do
    π(a) := p(a)/l(a)
for each b ∈ J do
    v(b) := c(b)
for π(a) が大きい順にすべての a ∈ I について do
    for each b ∈ J do
        if (a, b) ∈ E かつ v(b) ≥ l(a) then
            a を b に割り当てて，v(b) := v(b) − l(a).
            break
```

3.2 Match-and-FPTAS

Match-and-FPTAS は Miyazaki ら [8] によって 2015 年に提案されたアルゴリズムである．このアルゴリズムは，多項式時間で $(1 + \frac{2}{k+1} + \epsilon)$-近似を達成できる．ただし，$k$ とは割当制約つき複数ナップサック問題におけるサイズと容量の比の最小値であり，以下で定義される．

$$k = \min \left\{ \frac{c(b)}{l(a)} \;\middle|\; (a, b) \in E \right\}.$$

このアルゴリズムの概要は以下のとおりである．まず，LP 緩和問題を最大フローに帰着して解き，目的関数値を保ったまま，解のなかで分数になっている要素の数をできる限り減らす．次に，残った分数の要素から，対応する二部グラフ上における最大サイズのマッチングを求める．最後に，LP 緩和解とマッチングの結果を用いて各ナップサックについての単一ナップサック問題に分割し，それぞれを $(1+\epsilon)$-近似の FPTAS を用いて解く．

このアルゴリズムの計算時間は，$\mathrm{O}(T(m,n) + m^2n^2 + mn^3\epsilon^{-1})$ である．ただし，$T(m,n)$ はそれぞれの集合の頂点の数が m と n であるような二部グラフ上で，LP 緩和問題を最大フローに帰着して解くのに必要な時間である．

4 実験

4.1 例題の生成方法

特にことわらない限り，データの生成は次のようにおこなった．まず，アイテム数 n をナップサック数 m の 10 倍とした．各アイテムの利得とサイズはともに $[0, 1]$ 上の一様分布に従って決めた．ナップサックの容量は，$[k_{\min}, 10]$ 上の一様分布に従って生成した．ただし，$k_{\min}$ は容量の下限であり，これを調整することで容量とサイズの比が一定以上になるようにした．割当制約は，すべてのアイテムとナップサックの組についてそれぞれ独立に，確率 $p = 0.5$ で割当可能となるように決めた．また，Match-and-FPTAS のパラメータ ϵ は 0.1 とした．ひとつの m について 100 個の例題を

生成し，そのすべての例題を解いた結果の平均値をプロットした．

4.2　実験環境

次のような実験環境において実験した．

- CPU: 1.7GHz Intel Core i7.
- メモリ: 8GB.
- OS: Mac OS X 10.10.
- 言語: C++.

4.3　最適解との比較

整数計画ソルバ SCIP[*1]を用いて最適解を求め，貪欲法，Match-and-FPTAS が算出する解との比較をおこなった．$k_{\min} = 5$ とした．例題のサイズを大きくすると SCIP の計算時間が莫大になってしまうため，$m = 5$ 以下についてのみ実験をおこなった．

その結果は図 1, 図 2 のようになった．貪欲法，Match-and-FPTAS ともにほとんど最適な解を返していることがわかった．また，計算時間は貪欲法のほうが Match-and-FPTAS に比べて約 100 倍高速だった．

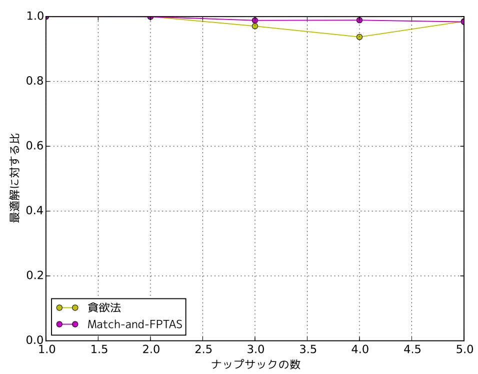

図 1　各アルゴリズムの解と最適解の比較

4.4　既存アルゴリズムの改良

既存アルゴリズムの実用的な性能を上げることを目指して，貪欲法と Match-and-FPTAS に変更を加えて実験をおこなった．$k_{\min} = 1, 5, 9$, $n = 10m, 30m$ の 6 通りに対して実験した．ただし，例題のサイズが大きいとソルバを用いて最適解を求めることができないため，元の貪欲法の解の値を 1 としたときの，その他のアルゴリズムの解の値をプロットした．

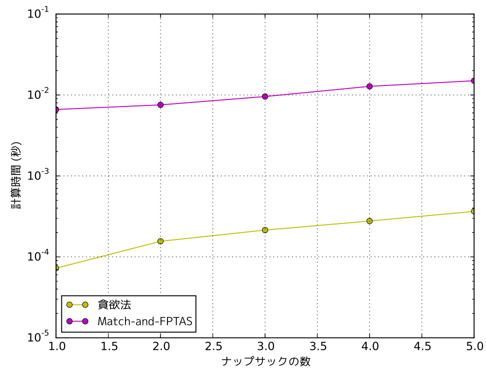

図 2　それぞれのアルゴリズムの計算時間

4.4.1　重みつき Match-and-FPTAS と利得重みつき Match-and-FPTAS

Match-and-FPTAS では，LP の解において分数となっている枝から，最大サイズのマッチングを求める．ここでは，解の重みを考慮してマッチングを求めることを考える．枝 $(a, b) \in E$ の重みを $x_{(a,b)}$ とみなして最大重みマッチングを求める方法を重みつき Match-and-FPTAS，枝の重みを $x_{(a,b)}p(a)/l(a)$ とみなす方法を利得重みつき Match-and-FPTAS と呼ぶことにする．ただし，これらの方法においては，マッチされないアイテムがある可能性があるため，元の Match-and-FPTAS で成り立っていた精度保証は失われている．

4.4.2　昇順貪欲法と降順貪欲法

貪欲法に次のような変更を加えたアルゴリズムを考える．元の貪欲法では，アイテムを割り当てるナップサックの優先順位を適当に決めていた．ここでは，優先順序をナップサックの容量の昇順または降順で並べた昇順貪欲法と降順貪欲法を考える．

4.4.3　実験結果と考察

実験結果は図のようになった (図 3, 図 4, 図 5, 図 6, 図 7, 図 8)．

重みつき Match-and-FPTAS と利得重みつき Match-and-FPTAS は，元の Match-and-FPTAS と比べてよい解を返した．しかし，これらすべてが，m

[*1] http://scip.zib.de/

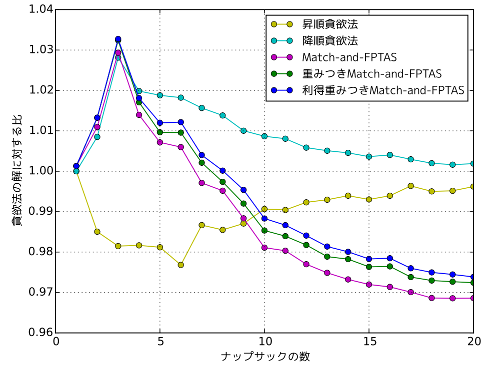

図 3　それぞれのアルゴリズムの解の，元の貪欲法の解に対する比 ($k_{\min} = 1$，$n = 10m$ の場合)

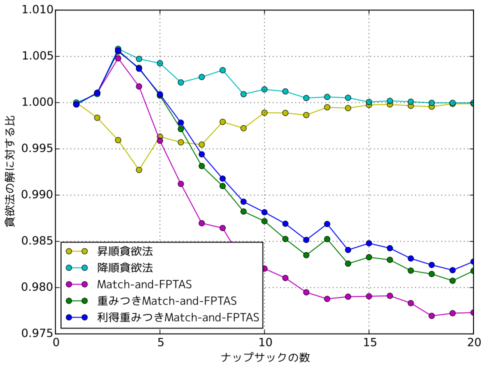

図 4　それぞれのアルゴリズムの解の，元の貪欲法の解に対する比 ($k_{\min} = 5$，$n = 10m$ の場合)

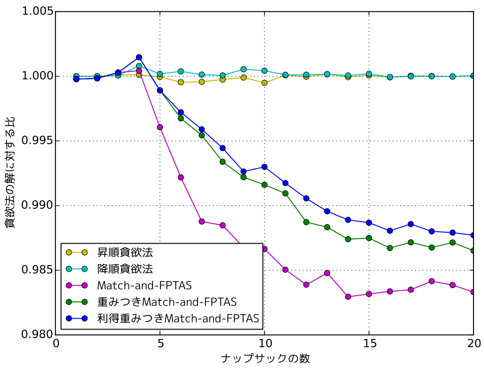

図 5　それぞれのアルゴリズムの解の，元の貪欲法の解に対する比 ($k_{\min} = 9$，$n = 10m$ の場合)

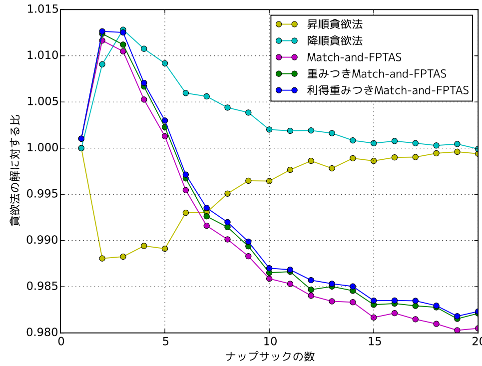

図 6　それぞれのアルゴリズムの解の，元の貪欲法の解に対する比 ($k_{\min} = 1$，$n = 30m$ の場合)

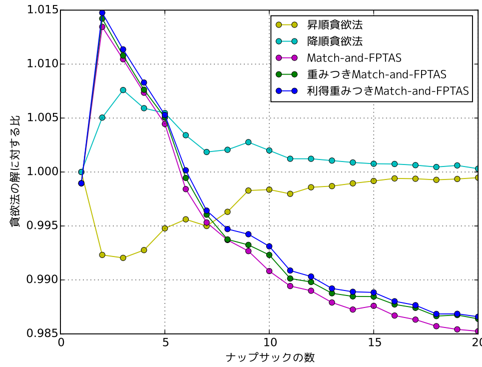

図 7　それぞれのアルゴリズムの解の，元の貪欲法の解に対する比 ($k_{\min} = 5$，$n = 30m$ の場合)

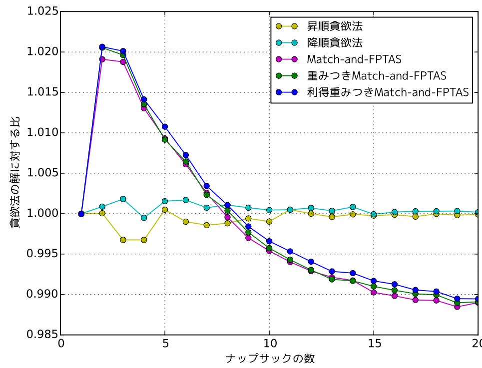

図 8　それぞれのアルゴリズムの解の，元の貪欲法の解に対する比 ($k_{\min} = 9$，$n = 30m$ の場合)

が大きくなると貪欲法よりも悪い結果になった．降順貪欲法と比べると，$k_{\min} = 1$，$n = 10m$ のときは降順貪欲法と同程度もしくはより悪いが，$k_{\min} = 9$，$n = 30m$ のときは小さい m について降順貪欲法よりよい結果を出している．

3 種類の貪欲法を比較すると，降順貪欲法，元の貪欲法，昇順貪欲法の順によい結果を出した．この差は $k_{\min}$ が小さいときのほうが明確に現れた．m が大きくなると昇順貪欲法，降順貪欲法ともに元の貪欲法の結果に近づいた．降順貪欲法と昇順貪欲法が，元の貪欲法に対して，ほぼ上下対称な結果になっていることが見てとれた．

4.5 パラメータ ϵ の影響

Match-and-FPTAS のパラメータ ϵ とは，各ナップサックについて FPTAS を実行するときに，その近似精度を指定するパラメータである．ϵ の値を小さくすると近似の精度がよくなることが期待できるが，一方で計算時間が大きくなってしまう．この ϵ の値が実験結果にどのように影響しているかを調べた．ただし，計算時間が大きくなってしまうため，ひとつのパラメータに対して例題を生成する個数は 10 個とし，その平均をとった．その結果は図 9, 図 10 のようになった．

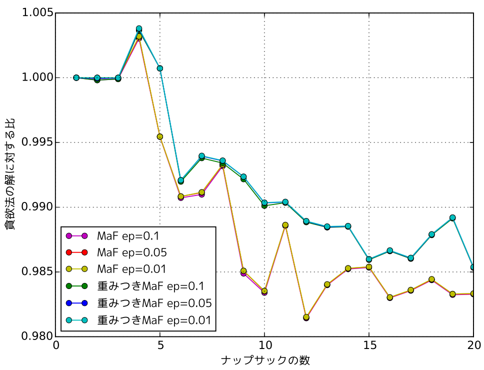

図 9 さまざまな ϵ に対する Match-and-FPTAS の貪欲法との比 ($k_{\min} = 1$ の場合)

この結果から明らかなように，ϵ の値を変えても Match-and-FPTAS，重みつき Match-and-FPTAS ともに解の値がほとんど変化しない．これは，この程度の規模の例題では $\epsilon = 0.1$ でもほとんどの場合単一

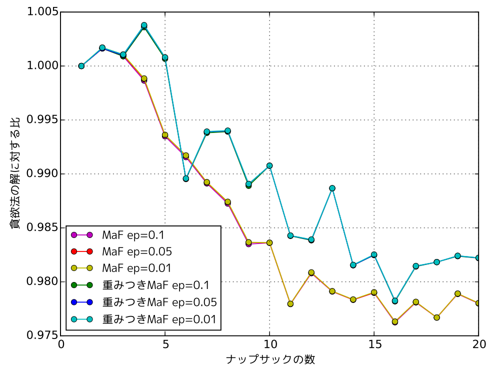

図 10 さまざまな ϵ に対する Match-and-FPTAS の貪欲法との比 ($k_{\min} = 5$ の場合)

ナップサック問題の最適解が求まっているからではないかと考えられる．よって，実用的には $\epsilon = 0.1$ で十分であるといえる．

4.6 実際の規模を想定した実験

次に，実際の集合住宅をモデルとして例題を生成し，実験をおこなった．世帯数 50 戸, 各世帯の家電数 30 個の集合住宅を想定し，各家電の消費電力は $[10, 2000]$W 上の一様分布に従って決めた．利得はモデル化が難しいが，ここでは $[0, 1]$ 上の一様分布に従うと仮定した．電源は全部で 5 個あるものとする．そのうち 3 つを電力会社からの電源とし，その容量は $[50, 200]$kW 上の一様分布に従って生成した．残り 2 つを太陽光発電による電源とし，$[10, 50]$kW 上の一様分布に従って容量を決めた．電力会社からの電源を安定，太陽光発電による電源を不安定とし，各家電には安定か不安定かという特性をそれぞれ独立に確率 0.5 で割り当てた．そのうえで，安定同士または不安定同士なら確率 0.9 で割当可能であり，安定なものと不安定なものは確率 0.1 で割当可能であると決めた．

100 個の例題を生成し，それらを各アルゴリズムを使って解き，得られた解の元の貪欲法の解に対する比の平均をとったところ，表 1 のようになった．上述の実験とは異なり, 昇順貪欲法がもっともよい性能を見せた．$n = 300m$ なので, 先ほどまでの実験から, Match-and-FPTAS がよい性能を見せることが期待されたが, 変更を加えたものも含めて Match-and-FPTAS は貪

欲法に比べて悪い結果だった．

表 1　実際の規模を想定した例題に対する実験結果 (元の貪欲法の解に対する比)

昇順貪欲法	1.007
降順貪欲法	0.9978
Match-and-FPTAS	0.9214
重みつき Match-and-FPTAS	0.9294

5　おわりに

本研究では，電力の割当問題に応用することを目指して，割当制約つきナップサック問題に対するアルゴリズムの実験的な評価をおこなった．貪欲法と Match-and-FPTAS はともに最適解の 9 割以上のよい解を返していることがわかった．また，重みつき Match-and-FPTAS，利得重みつき Match-and-FPTAS によって元の Match-and-FPTAS よりよい解が求まることがわかった．人工的な例題に対しては，m が小さい場合に，Match-and-FPTAS は貪欲法に比べてよい性能を示した．しかし，現実の規模を想定した例題では貪欲法のほうがよい結果であり，その理由はわかっていない．さらなる実験によってその原因を探ることは今後の課題である．

謝辞

本研究は，JST 京都地域スーパークラスタープログラム，JSPS 科研費 24500013 および 15K15979 の助成を受けたものである．

参考文献

[1] C. Chekuri and S. Khanna, A polynomial time approximation scheme for the multiple knapsack problem, *SIAM Journal on Computing*, 35(3), pp.713–728, 2006.

[2] M. Dawande, J. Kalagnanam, P. Keskinocak, R. Ravi and F.S.Salman, Approximation algorithms for the multiple knapsack problem with assignment restrictions, *Journal of Combinatorial Optimization*, 4(2), pp.171–186, 2000.

[3] U. Feige and J. Vondrák, Approximation algorithms for allocation problems: Improving the factor of $(1-1/e)$, *Proceedings of the 47th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 667–676, 2006.

[4] L. Fleischer, M. Goemans, V. Mirrokni and M. Sviridenko, Tight approximation algorithms for maximum general assignment problems, *Proceedings of the seventeenth annual ACM-SIAM Symposium on Discrete Algorithm (SODA)*, pp.611–620, 2006.

[5] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, A Series of Books in the Mathematical Sciences, W.H.Freeman, 1979.

[6] E. Gelenbe, Energy packet networks: smart electricity storage to meet surges in demand, *Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques*, pp.1–7, 2012.

[7] T. Kato, K. Tamura and T. Matsuyama, Adaptive storage battery management based on the energy on demand protocol, *Proceedings of the 3rd International Conference on Smart Grid Communications*, pp.43–48, 2012.

[8] S. Miyazaki, N. Morimoto and Y. Okabe, Approximability of two variants of multiple knapsack problems, *Proceedings of the 9th International Conference on Algorithms and Complexity (CIAC)*, (Lecture Notes in Computer Science 9079), pp.365–376, 2015.

[9] Z. Nutov, I. Beniaminy and R. Yuster, A $(1-1/e)$-approximation algorithm for the generalized assignment problem, *Operations Research Letters*, 34(3), pp.283–288, 2006.

[10] K. Sakai and Y. Okabe, Quality-aware energy routing toward on-demand home energy networking:(Position paper), *Proceedings of the 8th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, pp.1041–1044, 2011.

[11] T. Takuno, Y. Kitamori, R. Takahashi and T. Hikihara, AC power routing system in home based on demand and supply utilizing distributed power sources, *Energies*, 4(5), pp.717–726, 2011.