

複雑な個数制限のついた一般化割当問題について

小木曾 由明 (指導教員: 柳浦 睦憲)

名古屋大学 工学部

1. 前書き

本研究では、割当個数に制約のある多資源一般化割当問題を考える。多資源一般化割当問題とは、いくつかの資源に関する制約条件の下で、複数の仕事を複数のエージェントに割り当て、割当に伴うコストを最小化する問題である⁽¹⁾。また割当個数の制約とは、例えば、「あるエージェント i に割り当てられる仕事の数は 2 個か 5 個」のように、割当個数が指定された候補のいずれかでなければならないというものである。

2. 問題の定義と定式化

割当個数に制約のある多資源一般化割当問題に対しては、エージェントの集合 $I = \{1, \dots, m\}$ とそれらに割り当てる仕事の集合 $J = \{1, \dots, n\}$ が与えられる。この時、資源集合 $K = \{1, \dots, l\}$ と各エージェントに割り当てられる個数集合に関するそれぞれの制約条件の下で、各仕事をちょうど一つのエージェントに割り当て、割当に伴うコストの合計を最小化する問題である。この問題では、 $i \in I, j \in J, k \in K$ に対して仕事 j をエージェント i に割り当てたときにかかるコスト c_{ij} 、仕事 j をエージェント i に割り当てたときに使用する資源 k の量 a_{ijk} 、エージェント i が持つ資源 k に対する資源容量 b_{ik} 、エージェント i に割り当てることが可能な仕事数の集合 S_i が与えられる。便宜上 S_i の要素を $\{e_{i1}, e_{i2}, \dots\}$ と表し、 $e_{i1} < e_{i2} < \dots$ を仮定する。例えば、あるエージェント i に割当可能な仕事数が「2 個、5 個、7 個」のいずれかである時、 $S_i = \{e_{i1}, e_{i2}, e_{i3}\}$ 、 $e_{i1} = 2, e_{i2} = 5, e_{i3} = 7$ となる。エージェント i に割り当てられている仕事数が S_i の要素 e_{is} と等しい時に $z_{is} = 1$ 、そうでない時に $z_{is} = 0$ の値をとる 0-1 変数 z_{is} を各 $i \in I$ と $S \in \{1, 2, \dots, |S|\}$ に対して用意する。また、仕事 j をエージェント i に割り当てた時に $x_{ij} = 1$ 、そうでない時に $x_{ij} = 0$ の値をとる 0-1 変数 x_{ij} を各 $i \in I$ と $j \in J$ の組に対して用意する。これらを用いると、問題は以下のように定式化できる:

$$\begin{aligned} \text{minimize } cost &= \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} & \forall i \in I, \forall j \in J \\ \text{subject to } \sum_{j=1}^n a_{ijk} x_{ij} &\leq b_{ik} & \forall i \in I, \forall k \in K \\ \sum_{j=1}^n x_{ij} &= \sum_{s=1}^{|S_i|} e_{is} z_{is} & \forall i \in I \\ \sum_{i=1}^m x_{ij} &= 1 & \forall j \in J \\ \sum_{s=1}^{|S_i|} z_{is} &= 1 & \forall i \in I. \end{aligned}$$

資源数 $k \geq 1$ の場合にはこの問題は NP 困難であることが知られている⁽¹⁾。一方、資源数 $k = 0$ の場合、割当可能な仕事数が偶数 (あるいは奇数) 全ての集合であるような特殊な個数制約を持つ場合にはこの問題が多項式時間で解けること、および割当可能な仕事数がどのエージェントにおいても 0 または 3 に限定される場合でもこの問題が NP 困難であることを証明した。

3. 提案アルゴリズム

3.1 局所探索と解の評価方法

局所探索法は、適当な初期解から始め、現在の解の近傍内に改善解が無いかを探索し、改善解が見つければ、解をそちらに置き換えるという操作を反復する手法である。本研究では、探索解の評価方法として、実行可能解からの逸脱の度合いをペナルティとしたペナルティ関数を用いる。このペナルティに重みをかけたものとコストの合計を評価関数とした。また、資源制約と個数制約それぞれに対するペナルティ重みを探索履歴に基

づいて適応的に更新する。

ひとつの仕事の割当先を変更することにより得られる解の集合をシフト近傍と呼び、異なるエージェントに割り当てられている二つの仕事の割当先を互いに交換することにより得られる解の集合を交換近傍と呼ぶ。提案手法ではこれら 2 つの近傍を用いる。

3.2. シフトキックと交換キック

局所探索法は高速であるが、得られる解の精度は十分良いとはかぎらない。そのためメタ戦略のひとつである反復局所探索法を用いることにより、精度の改善を図った。反復局所探索法は、過去の探索で得られた良い解にランダムな変形を加えたものを初期解として、局所探索法を反復する方法である。このランダムな変形に用いる操作を一般にキックと呼ぶ。この問題では、あるエージェント i に割り当てられている仕事数がある e_{is} から他の $e_{is'}$ に変化するような解への移動が起こりやすい。この性質を解決するため、あるエージェント i に現在割り当てられている仕事数を、その値に最も近い e_{is} より 1 段階少ない $e_{i(s-1)}$ になるまでエージェント i から仕事を他のエージェントへシフトし続ける操作をキックに利用した。これをシフトキックと呼ぶ。また二つの仕事の交換をランダムに複数回行う操作もキックに用いた。これを交換キックと呼ぶ。

3.3. アルゴリズムの流れ

step1 改善がなくなるまでシフト近傍による局所探索を行う。
step2 交換近傍の探索を行う。改善解が見つかった場合は現在の解をそちらに置き換えて step1 に戻る。さもなければ step3 へ進む。
step3 前回の重みの更新を行ってから一度も解が改善されていない場合は step4 へ進む。さもなければ、ペナルティ重みの更新を行ったのち step1 に戻る。
step4 終了条件が満たされた場合は step6 へ進む。さもなければ、シフトキックまたは交換キックのいずれか一方をランダムに選び、実行したのち step5 へ進む。
step5 step4 においてシフトキックを行った場合は交換近傍による探索を行う。改善解が見つかる、あるいは近傍内の全ての探索が終了した場合は step1 に戻る。交換キックを行った場合は即座に step1 に戻る。
step6 探索した中で最も良い実行可能解を出力して終了する。

4. 結果

上記のアルゴリズム (SFK-SWK) とこのアルゴリズムの交換キックを行わなかったもの (SFK)、シフトキックを行わなかったもの (SWK)、重みの更新毎にキックを行うもの (ESFK-SWK) と比較、検証を行い、資源数が少ない問題例では、SFK-SWK、ESFK-SWK が 4 つの中で良い結果を得ることができ、資源数の多い問題例では ESFK-SWK が 4 つの中で最も良い結果を得る事が出来た。また、ランダム多スタート局所探索法との比較を行い、提案アルゴリズムがランダム多スタート局所探索法よりも最も良い結果を得る事を確認出来た。

5. まとめ

割当個数に制約のある多資源一般化割当問題に対して、アルゴリズムを提案した。また、提案アルゴリズムで用いた種々のアイデアについて、それを用いないアルゴリズムと比較することによってその有効性を確認した。

参考文献

(1) M. Yagiura, S. Iwasaki, T. Ibaraki and F. Glover, "A Very Large-Scale Neighborhood Search Algorithm for the Multi-Resource Generalized Assignment Problem," Discrete Optimization, 1 (2004) pp.87-98.