2001年度日本オペレーションズ・リサーチ学会
秋季研究発表会

一般化2次割当問題に対する大規模近傍探索法の適用について

京都大学 *小島 健哉 KOJIMA Kenya
01704164 京都大学 柳浦 睦憲 YAGIURA Mutsunori
01001374 京都大学 茨木 俊秀 IBARAKI Toshihide

1 まえがき

本研究では, 新しく一般化2次割当問題を定式化し, それに対する局所探索法を提案する. 一般化2次割当問題とは, n 個の仕事と m 個のエージェントが与えられたとき, 各エージェントにおける複数の資源制約を満たした上で, 割当に伴うコストを最小化するよう, 全ての仕事をいずれかのエージェントに割当てる問題である. 割当コストとしては, 一次の割当コストに加え,2次の割当コストを扱うことが可能であり, その結果, 一般化割当問題,2次割当問題, グラフ彩色問題, 一部のチャネル割当問題などを含む非常に汎用性の高い問題となっている.

本研究の局所探索法では, 単純な近傍操作を連鎖的に複数回繰り返すことを許すという強力な近傍を用いることで, 性能の向上を図っている. ただし, この近傍内の解を全て列挙するのでは効率が悪いので, 改善解探索木を用いて, 良い解が含まれている領域のみを効果的に探索する工夫を加えている.

計算実験では, 一般化割当問題,2次割当問題, およびグラフ彩色問題のベンチマーク問題に提案手法を適用した. 各問題に対する専用解法による結果と比較して, やや時間は長くかかるものの, それほど見劣りしない解が得られている.

2 問題の定式化

一般化2次割当問題は, 仕事集合 $J=\{1,...,n\}$ と, エージェント集合 $I=\{1,...,m\}$ が与えられたとき, 資源集合 $K=\{1,...,r\}$ に関する制約条件を満たすよう各仕事をちょうど一つのエージェントに割当てる問題であり, 割当コストの総和 (一次の割当コストと2次の割当コストの和) を最小化することが目的である. この問題では, $j,j'\in J,\ i,i'\in I$, および $k\in K$ に対し,

$a_{jik}(\geq 0)$: 仕事 j をエージェント i に割当てる際に必要な資源 k の量

c_{ji}: 仕事 j をエージェント i に割当てる際にかかるコスト

$b_{ik}(\geq 0)$: エージェント i における資源 k の使用可能量

$u_{jj'}$: 仕事対 j,j' に関するコスト係数

$w_{ii'}$: エージェント対 i,i' に関するコスト係数

が与えられる. また, 割当を写像 $\pi: J \to I$ で表す. $\pi(j)=i$ は, 仕事 j をエージェント i に割当てることを意味し, また, エージェント i に割当てられる仕事の集合を

$$J_i(\pi)=\{j\in J\,|\,\pi(j)=i\}$$

と記す. これらを用い, 一般化2次割当問題は以下のように定式化される:

$$\begin{aligned}&\text{minimize} && cost(\pi)=\sum_{j\in J}c_{j\pi(j)}+\sum_{j,j'\in J}u_{jj'}w_{\pi(j)\pi(j')}\\ &\text{subject to} && \sum_{j\in J_i(\pi)}a_{jik}\leq b_{ik},\ i\in I,\ k\in K.\end{aligned}$$

3 局所探索法

局所探索法は, 適当な初期解から始め, 現在の解 π の近傍 $N(\pi)$ 内に π より良い解があればそれに置き換えるという操作を, 近傍内に改善解がなくなるまで反復する方法である. 我々のアルゴリズムでは, すべての割当 π を探索空間とする. 探索解の評価には, ペナルティ関数法を用い, 実行不可能解に対しては, 実行可能領域からの逸脱の度合いを示すペナルティを目的関数 $cost(\pi)$ に加える. すなわち, エージェント i に仕事集合 $S\ (\subseteq J)$ を割当てたときの資源 k に関するペナルティを

$$p_{ik}(S)=\max\left\{0,\sum_{j\in S}a_{jik}-b_{ik}\right\},\ i\in I,\ k\in K$$

とし, 解の評価関数を

$$pcost(\pi)=cost(\pi)+\sum_{i\in I}\sum_{k\in K}\alpha_{ik}p_{ik}(J_i(\pi))$$

と定義する. なお, α_{ik} は正の値をとるパラメータで, 探索状況に応じて適応的に制御する.

近傍 $N(\pi)$ には, シフト近傍, 交換近傍, および連鎖近傍の3つを用いる. 連鎖近傍については, 3.1節で説明する. シフト近傍 N_{shift} と交換近傍 N_{swap} は, それぞれ

$$\begin{aligned}N_{\text{shift}}(\pi)=&\ \{\pi'\,|\,\pi'\text{は}\pi\text{の一つの仕事の割当先を}\\&\quad\text{変更することで得られる割当}\}\\ N_{\text{swap}}(\pi)=&\ \{\pi'\,|\,\pi'\text{は}\pi\text{の二つの仕事の割当先を}\\&\quad\text{互いに交換することで得られる割当}\}\end{aligned}$$

と定義される.

3.1 連鎖近傍

連鎖近傍 $N_{\text{cyclic}}(\pi)$ は, $l(\geq 2)$ 個の相異なる仕事 $j_1,j_2,\ldots,j_l$ の割当先を同時に変更する解 π' の中で,

$$\begin{array}{lcll}\pi'(j_{s+1}) & = & \pi(j_s), & s=1,2,\ldots,l-1\\ \pi'(j_1) & = & \pi(j_l) & \end{array}\qquad(1)$$

を満たすものの集合である．ここで，連鎖近傍のサイズ $|N_{\mathrm{cyclic}}(\pi)|$ はエージェント数に関して指数的に増加するため，単純に全ての解 $\pi' \in N_{\mathrm{cyclic}}(\pi)$ を列挙して調べるのではなく，次に述べる改善解探索木を利用することにより，近傍サイズの縮小と探索の効率化を実現している．

3.2 改善解探索木

改善解探索木 $T(\pi)$ は解 π に応じて定義される根付き木であり，条件 (1) を満たすシフト操作の列を全て列挙するものである．深さ $d \geq 1$ の各節点 $v \in V$ には一つの仕事 $j \in J$ が割当てられる．また，節点 v の深さを d_v とし，v から根へ向かうパス上の節点に対応する仕事の列を $j^v_{d_v}, j^v_{d_v-1}, \ldots, j^v_1$ と記す．このパスに対し，シフト操作の列を

$$\begin{cases} \text{仕事 } j^v_d \text{ の割当先を } \pi(j^v_{d-1}) \text{ にシフト } (d > 1), \\ \text{仕事 } j^v_1 \text{ の割当先を } \pi(j^v_{d_v}) \text{ にシフト}, \end{cases}$$

と定義し，これらを同時に適用することによって得られる解を π_v とする．π と π_v の評価関数値 $pcost$ の差を

$$\Delta(v) := pcost(\pi_v) - pcost(\pi)$$

とし，$\Delta^-(v)$ を，π_v において仕事 j^v_1 をどのエージェントにも割当てないときの $pcost$ と $pcost(\pi)$ との差とする．$\Delta^-(v)$ は，$\Delta(v)$ を効率よく計算するため，および近傍内の探索解を選択する基準に利用する．

3.3 連鎖近傍の探索

改善解探索木 $T(\pi)$ において，$\Delta(v)<0$ なる節点 v を発見できれば，改善解が得られる．しかし，$T(\pi)$ 内の節点すべてを調べるのは効率的とはいえない．そこで，本研究では探索する節点を改善の見込みがあると思われるものに絞り込むためにいくつかの絞り込みルールを加えて効果的な探索を実現している．$T(\pi)$ の探索は，幅優先探索に基づいて行い，以下の絞り込みルールを用いて探索する節点数を制限する．

ルール 1. 最大 $d_{\max}=\min(m, \sqrt{n})$ の深さまでしか探索しない．

ルール 2. 深さ d (≥ 2) の節点の中から，$\Delta^-(v)$ の小さいものから順に $\lfloor n/(d_{\max} d) \rfloor$ 個を選択し，それらの子孫のみを探索する．

ルール 3. 節点 $v \in V$ から根へのパス上に，同じエージェントに割当てられた仕事が存在するとき，v およびその子孫の探索は行わない．

4 アルゴリズムの概要

提案手法では，シフト近傍，交換近傍，連鎖近傍の 3 つの近傍による局所探索を順次行う．いずれの近傍内にも改善解が見つからなくなった場合には，シフト近傍内の最良解を新たな初期解とし，再び局所探索を行う (反復局所探索法, ILS)．その際，タブーリストを用い，同じ初期解から探索を行わないようにしている．また，ペナルティ重み α_{ik} の適応的制御も行っている．これらの工夫は，(1 資源) 一般化割当問題において非常に有効であることが分かっており [3]，多資源の場合にも同様の効果が得られた．

5 実験結果

実験はワークステーション Sun Ultra 2 Model 2300 (300 MHz, 1 GB memory) 上で C 言語を用いて行った．実験の対象として，一般化割当問題と 2 次割当問題のベンチマーク問題を用い，提案手法 (ILS) をランダム多スタート法 (MLS)，一般化割当問題の専用アルゴリズムであるタブーサーチ (TS)[2]，および 2 次割当問題の専用アルゴリズムであるアニーリング法 (SA)[1] と比較した．ILS と MLS の制限時間は 1200 秒，TS の制限時間は 600 秒である．一般化割当問題，2 次割当問題に対して得られた最良値をそれぞれ表 1 と 2 に示す．ILS は MLS よりも良い結果を与え，それぞれの専用解法と比べてもそれほど見劣りしない結果を得ている．

表 1. 一般化割当問題に対する 3 つの
アルゴリズムによる最良値の比較

m	n	r	ILS	MLS	TS[2]
10	200	4	23425	24231	*23363
20	100	4	*6290	6483	6321
20	200	4	22590	25709	*22439

(*印は 3 つの内の最良であることを示す)

表 2. 2 次割当問題に対する 3 つの
アルゴリズムによる最良値の比較

問題例	ILS	MLS	SA[1]	既知の最良値
NUG30	*6124	6252	6128	6124
Sko42	15884	16126	*15830	15812
Sko72	66372	66872	*66372	66256
Sko90	*115624	116828	115796	115534

(*印は 3 つの内の最良であることを示す)

参考文献

[1] L.M. Gambardella , E.D. Taillard and M. Dorigo, "Ant Colonies for the Quadratic Assignment Problem," *Journal of the Operational Research Society,* 50 (1999) 167-176.

[2] S.Iwasaki , "多資源一般化割当問題に対する大規模近傍探索法の適用について," 特別研究報告書, 京都大学情報学研究科 (2000).

[3] M. Yagiura, T. Ibaraki and F. Glover, "An Ejection Chain Approach for the Generalized Assignment Problem," Technical Report #99013, Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, (1999) (available at http://www-or.amp.i.kyoto-u.ac.jp/~yagiura/).