

G-067

マルチエージェント間の交渉による修正の連鎖を用いたスケジュール作成 Scheduling by Multiagent Negotiation Using a Chain of Changes

梶村 友哉[†]

Tomoya MASUMURA

高橋 和子[‡]

Kazuko TAKAHASHI

1. はじめに

複数のエージェントの予定を考慮にいたスケジュール作成問題は、通常制約充足問題の1つとしてとらえられ、全解探索で解決しようとする一般にNP困難になる。しかし、この種の問題では、必ずしも最適解を求める必要はなく、全員がある程度満足度のいく解(妥協解)を求めれば十分なことも多い。さらに、原案が与えられた時に状況の変更によってそれを修正する場合に再び全解探索をするのは無駄である。

本稿では、原案が与えられたスケジュールに対し、エージェント同士の交渉をベースとした修正案を生成する方法について述べる。

本稿の構成は以下の通りである。まず、第2.で本研究の対象とする問題について述べ、第3.章でアルゴリズムを示す。第4.章でこの方法の特徴を述べ、第5.章でまとめと今後の課題を述べる。

2. 対象とする問題

ここでは複数の枠に複数のエージェントを配置する問題を対象として考える。エージェント同士の交渉をベースとしてこの問題を解く代表的なものとしては、1対1の交渉で最も良い解を求めそれをトーナメント式に繰り返していき、1対多の交渉で最も良い解を返してくれるものを採用するという契約ネットプロトコル型の方法がある。いずれも交渉が成立した場合は2者の間でリソース(配置された枠)の交換が実行される。しかし、ここでいう最も良い解とは、2者の間のみでリソース交換を実行した時の評価値を比較したものであり、3者以上のエージェントがそれぞれのリソースを交換した場合の評価値は考慮に入られていない。そのため、最適解が存在するのにもかかわらずそれを発見できない場合がある。

例として図1(a)のような原案が与えられたとする。

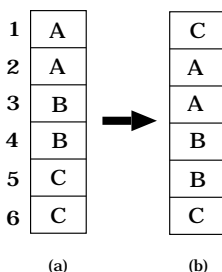


図1: 3者以上がリソースの交換を行って問題解決する例

エージェントA, B, Cが1から6までの時間枠に2回ずつ割り当てられる。A, Bは連続する時間枠の組を希

[†]関西学院大学 大学院 理学研究科

[‡]関西学院大学 理工学部 情報科学科

望し、Cは3つ以上離れた時間枠の組を希望するものとする。また、Aは枠5, 6では都合が悪いものとする。この原案では、Cの希望がかなえられていないため、Cは自分の都合のよい枠に配置変更してもらうよう他のエージェントと交渉する必要がある。Cは自分の割り当てられた枠の1つとAの割り当てられた枠の1つを交換するようAに要求する。しかし、この要求を受け入れてしまうと、原案でかなっていたAの希望が消失することになるので、Aはこの要求を拒否する。また、CはBに対しては要求を持たないため、交渉は行われず、この問題には解が存在しないという結果になってしまう。

しかし、この例では(b)のような解が存在する。つまり(a)から(b)にいたるためには、A, B, Cの3者が交渉に参加しなければならない。

本稿では、この問題を解決し、解がある場合には必ず解を見つけ、ない場合には妥協解を見つける効率のよいアルゴリズムを提案する。また、解がある場合でも計算をある閾値で打ちきることによって妥協解にいたることができるとを示す。

3. アルゴリズム

3.1 概要

m 個の枠に n 個のエージェントを、与えられた条件を満たすように割り当てるスケジュール作成問題を解くアルゴリズムを与える。ここで、エージェント間の通信は行うが、全体を監視したり調整したりするマネージャの存在は仮定していない。以下では、説明の簡単化のため、 n 人のエージェントを2個ずつ $2n$ 個の枠に割り当てるものとするが、一般の場合にも容易に拡張できる。

初期案が与えられた時、2段階で修正案を作成する。エージェント同士が自分の割り当てられた枠を交換することで修正を行っていく。各エージェントには各案での自分の割り当てに対する都合のよさを表すutilityが与えられている。utilityが、ある閾値を満たさないとき、そのエージェントを不満エージェントとよぶ。

STEP1では、最適案の生成をめざす。この段階においては、不満エージェントは自分の不満が解消されることのみを目的としていわば強引に案の変更を求める。案が変更されることによって生じた(あるいは最初からの)不満エージェントは再び、自分の不満を解消するように行動する。これを繰り返す、全員が不満を解消した場合、その結果を最終案とする。ここで行われるのは1対1の通信である。

STEP2では、STEP1で解が求められなかった場合、その時の案から開始して妥協案の生成をめざす。この段階においては、各エージェントが閾値を下げることによって譲歩し、全員が一定以上の満足を得るような解を得ることを目的として交渉を行う。交渉相手は自分以外の全員を対象とし、全員のutilityの和をもっとも高くする

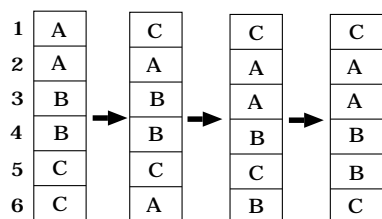


図 2: 本アルゴリズムにしたがった解決過程

案を採用する．ここで行われるのは1対多の通信である．

エージェント A_i が枠 N, M に割り当てられたときの A_i の utility を $U_i(N, M)$ と表し，その閾値を t_i と表す．また $U_i(N, M)$ と t_i は，ともに0以上1以下の値をとる．

3.2 最適案の生成

1. 初期案を現案とし， $t_i = 1$ とする．
2. 現案に対する不満エージェントがない場合は，その案を最終案として終了する．そうでなければ不満エージェントの1つを取り出しそれを A_i として以下を行う．
3. 現案で自分の割り当てられた枠を N, M とする． $U_i(N, R) \geq t_i$ または $U_i(M, R) \geq t_i$ である $R (R \neq N, M)$ が存在しなければ解なしとして終了する．存在すれば，その中の1つをとりあげ，現案で R に割り当てられているエージェントを A_j とする．また， R と交換する予定の枠を C とおく．
4. A_i から A_j に枠 C と R の交換要求を送り，枠の交換を実行して新しい案の候補 P を生成する．
5. P が既に出現したものならば，3に戻る．新たに出現したものならば，それを現案として2に戻る．

図1にあげた例をこのアルゴリズムにしたがって解いた過程を図2に示す．

3.3 妥協案の生成

譲歩の度合いを表す関数を f とする．

1. 各エージェント $A_s (s = 1, \dots, n)$ に対して $t_s = t_s - f(A_s)$ とする．
2. 現案に対する不満エージェントがない場合は，その案を妥協案として終了する．不満エージェントが存在する場合，不満エージェントの1つを取り出しそれを A_i とする． $t_i \leq 0$ ならば，交渉は決裂したとみなされ案は生成されない．それ以外の場合，以下の動作に進む．
3. 現案で自分の割り当てられた枠を N, M とする． $U_i(N, R) \geq t_i$ または $U_i(M, R) \geq t_i$ である R が存在しなければ1に戻る．存在すれば，それらをすべて候補としてとりあげ， R_1, \dots, R_k とする．現案で各 $R_j (j = 1, \dots, k)$ に割り当てられているエージェントをそれぞれ A_j とする．また， R_j と交換する

予定の枠を C_j ，交換せず残しておく枠を D_j とそれぞれおく．

4. A_i から各 A_j に枠 C_j と R_j の交換要求を送る．
5. 各 A_j は， $U_j(L_j, C_j) \geq t_j$ ならば $(U_j(L_j, C_j), R_j)$ を， $U_j(L_j, C_j) < t_j$ ならば *reject* を， A_i にそれぞれ返す．ただし L_j は現案で自分に割り当てられている R_j と異なる枠である．
6. A_i は，返された各 $(U_j(L_j, C_j), R_j)$ に対して， $U_j(L_j, C_j) + U_i(D_j, R_j)$ を計算し，最大になる j を選び A_j に枠 C_j と R_j の交換要求を送り，枠の交換を実行してそれを現案とし，2に戻る．*reject* しか返ってこなかった場合は，1に戻る．

4. 議論

最適案の決定においては不満エージェントや枠の交換相手を非決定的に選んでいるが，どのような選び方をしても，最適解が存在する場合は必ずそれが得られる．しかし，解が存在しない場合や存在しても膨大な計算時間が必要な場合，一定回の修正を繰り返した後に停止させて，STEP2 にすすむことも可能である．

また，現実の応用問題を考えると，エージェントの中に重要度の高い(意見の強い)ものとそうでないものがあったり，エージェントによって特定の枠に割り当てられるよりも連続する枠を割り当てられる方を優先したいなど，様々な場合がある．本手法においては，譲歩の度合いを表す関数 f や utility の決め方にそのような様々な制約条件を反映することが可能である．

5. まとめ

本稿では，原案が与えられたスケジュールに対し，エージェント同士の交渉をベースとした修正案を生成する方法について述べた．

このアルゴリズムでは，最適解がある場合は必ずそれを見つけ，解がない場合は全員が一定以上の満足をする妥協解を得ることができる．

この方法は，交渉の回数や総じての計算量は少なく，3者以上のエージェントがそれぞれのリソースを交換する解しかないような問題にも対応可能である．

今後はこのアルゴリズムを実装して実験評価を行い，utility や閾値の決め方に関しても検討していく予定である．

参考文献

- [1] 伊藤孝行, 新谷虎松. モバイルエージェント間の多重交渉に基づくグループ代替案選択支援システムについて, 情報処理学会論文誌, Vol. 39, No.12, pp. 3165-3176, 1998.
- [2] Wooldridge, M. *An Introduction to MultiAgent Systems*, 2002.
- [3] Zlotkin, G. and J.S. Rosenschein. *Rules of Encounter*, 1994.