

複数論点交渉問題におけるエージェントの効用空間の 公開範囲に基づく交渉手段の実現

藤田 桂英 伊藤 孝行 服部 宏充

マルチエージェントシステムの研究分野において、複数論点交渉問題 (Multi-issue negotiation) が注目を集めている。特に、筆者らは一般性が高く実世界に近い問題が扱える複数の論点が相互依存関係にある交渉問題に注目している。本論文ではエージェントがメデイエータに公開している範囲に基づいて閾値を調整するメカニズムを提案する。また、エージェントがどのくらい自分の選好情報を公開しているかに注目し、本手法を用いることで過剰にメデイエータに自分の効用情報を公開することなく最適率の高い解を得られることを示す。さらに、この手法が個々のエージェントの入札数が減らすことに成功しており、計算時間を減らしていることをシミュレーション実験により示す。

Multi-issue negotiation protocols represent a promising field since most negotiation problems in the real world involve interdependent multiple issues. Our work focuses on negotiation with interdependent issue and, therefore, nonlinear (multi-optimum) agent utility functions. In this paper, we propose a new threshold adjustment mechanism in which agent who open their local information more than the others. In addition preliminary experimental results demonstrate that the threshold adjusting mechanism can reduce the computational cost, and the amount of private information that is required for an agreement among agent keeping enough optimality.

1 まえがき

マルチエージェント研究分野において複数論点交渉問題が重要な研究課題となってきた。複数の論点を対象とした研究はこれまでに行われているが [1] [2] [3] [4] 既存の研究では論点の独立性が仮定されており、エージェントの効用は線形の効用関数として表現可能であった。しかし実世界の問題では複数の論点が全て独立していることは稀であり、複数の論点が相互依存関係にある場合が多い。そこで非線形関数を

もつエージェントのためのオークションをもととした交渉プロトコルが提案されている [5]。まず、エージェントは自分の効用空間においてサンプリングを行う。次に、サンプリングポイントを調整して局所解を発見する。そして、それをもとにエージェントの効用の大きい領域に対して入札の生成を行い、メデイエータが全ての入札の組み合わせを考慮して最終的な合意案を発見する手法である。

しかし、[5]の手法に対してエージェントの個人効用の公開については議論されていなかった。本来、エージェントの個人効用はできる限りメデイエータに対して公開されるべきではない。本論文では入札の際の公開範囲を調整し、交渉する手法を提案する。まず、エージェントは共通の初期閾値を用いて既存のプロトコル [5] で交渉を行う。もし解が発見出来なかった場合、メデイエータは各エージェントにどれだけ自分の効用の情報を公開しているかに基づいて閾値の下げ幅を指定する。その後、各エージェントはメデイエータの指定した下げ幅に基づいて閾値を下げ、再び交渉

Using Autonomous Threshold Adjustment to Enable Multi-Party Negotiations with Multiple Interdependent Issues

Katsuhide Fujita, 名古屋工業大学情報工学科, Dept. of Computer Science, Nagoya Institute of Technology
Takayuki Ito, 名古屋工業大学大学院産業戦略専攻, Dept. of Master of Techno-Business Administration, Nagoya Institute of Technology
Hiromitsu Hattori, 京都大学大学院情報学研究科社会情報学専攻, Dept. of Social Informatics, Kyoto University

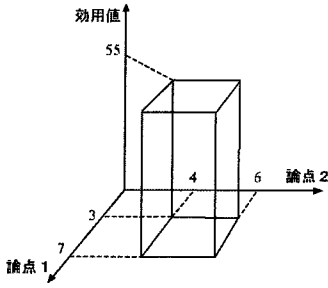


図1 二項制約の例

行う。そして、一連の操作を解が発見される、もしくは解がないものと判断されるまで繰り返す。

本論文の構成を以下に示す。2. ではそれぞれのエージェントがもつ非線形で複数論点を扱う効用空間と既存の交渉手法について述べる。3. では閾値調整を用いる新たな手法を提案し、計算量が既存の手法より少ないことを示す。4. で実験的に提案する手法の評価を行い、最後に全体のまとめについて述べる。

2 非線形効用関数に基づく交渉

2.1 複雑な効用空間

本論文では、 N 個のエージェントが合意形成を試みる交渉の状況を考える。論点が M 個存在し、個々の論点を $i_j \in I$ と表す。論点は $[0, X]$ の範囲の整数を値として持つ (すなわち、 $s_j \in [0, X]$)。交渉の結果得られ

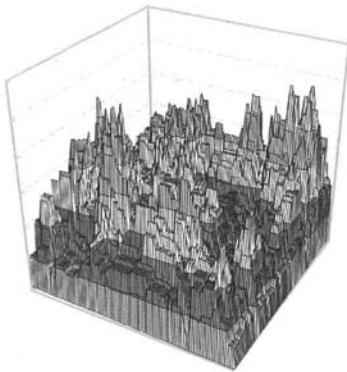


図2 非線形効用空間の例

る合意案は、各論点の値のベクトル $\vec{s} = (s_1, \dots, s_M)$ として表現される。

エージェントの効用関数は制約を用いて表現する。 l 個の制約が存在するとし、個々の制約は $c_k \in C$ と表す。制約は、単一、もしくは複数の次元 (論点) に関して、制約充足条件となる値の範囲、および効用値を持つ。図1は、論点1、および論点2に関連する二項制約の例を示したものである。本例では論点1に関しては $[3, 7]$ 、論点2に関しては $[4, 6]$ で範囲で合意が得られた場合充足可能であり、この場合得られる効用が55であることを示している。交渉に参加する全てのエージェントは、個々にユニークな制約集合を持っている。

合意 \vec{s} に関するエージェント i の効用を $u_i(\vec{s}) = \sum_{c_k \in C, \vec{s} \in x(c_k)} w_i(c_k, \vec{s})$ と定義する。ここで、 $x(c_k)$ は制約 c_k を充足可能な合意案の集合である。この効用表現により、多くの制約を充足可能な地点は効用が高くなり、逆に充足する制約数が少ない地点は効用が低くなる。このため、空間内に効用値による高低が生じ、凹凸のある非線形の効用空間が形成される。本論文での効用空間とは、各論点を取りうる値のあらゆる組み合わせについて効用関数によって得られる効用値を空間状にプロットして得られるグラフを意味し、空間の次元数は論点数+1となる。図2に非線形の効用関数の例を示す。この効用空間では2つの論点を対象としており三次元グラフとなる。また各論点の値域は $[0, 99]$ で、単項制約 (1つの論点しか関係しない制約)、二項制約 (2つの論点に関係する制約) がそれぞれ50個と100個あるとする。図2に示す通り、非線形の効用空間は山と谷が入り組んだ複雑なものとなる。

2.2 オークションに基づく交渉プロトコル

文献[5]で示されるオークションに基づく交渉プロトコルは以下の4つのステップからなる。

[Step1: サンプリング] 効用の高い合意案が存在する範囲を特定するために、エージェントがそれぞれの効用空間でサンプリングを行う。エージェントは一様分布に従う乱数に基づいてあらかじめ決まった数の合

意案候補を取得する。ここで取得するサンプル数が少なすぎる場合、エージェントが高効用のサンプルを見逃してしまい不十分な効用しか得られない可能性があることに注意が必要である。

[Step2: サンプリング調整] Step1 で得られたサンプルが局所最適解を的確に取られていることが保証されない。そこで、各エージェントはシミュレーションアニーリング [6] に基づく非線形の最適化を行い、サンプリングポイント周辺の局所最適解の発見を試みる。

[Step3: 入札の生成] エージェントは、サンプルを調整して得られた各合意案に関して効用を計算する。ここでの効用とは合意案を充足する制約の効用の総和である。効用が閾値を越えている場合に限りエージェントは入札を生成する。生成する入札は合意案と同一の効用が得られる周辺全てをカバーするように生成される。このような入札は、合意案 s に充足される全ての制約の共通範囲を取ることで容易に生成できる。Step1, Step2, および Step3 は以下のようにまとめられる。ここでは、記述の簡単のため、制約 c の充足によって得られる効用を v_c と表す。

SN : The number of samples

T : Temperature for Simulated Annealing

V : A set of values for each issue, V_m is for an issue m

B : A set of bid

```

1: procedure bid-generation_with_SA( $Th, V, SN, T, B$ )
2:    $P_{smp1} := \emptyset$ 
3:   while  $|P_{smp1}| < SN$ 
4:      $P_{smp1} := P_{smp1} \cup \{p_i\}$  (randomly selected from  $P$ )
5:    $P := \prod_{m=0}^{|I|} V_m, P_{sa} := \emptyset$ 
6:   for each  $p \in P_{smp1}$  do
7:      $p' := \text{simulatedAnnealing}(p, T)$ ,
8:      $P_{sa} := P_{sa} \cup \{p'\}$ 
9:   for each  $p \in P_{sa}$  do
10:     $u := 0, B := \emptyset, BC := \emptyset$ 
11:    for each  $c \in C$  do
12:      if  $c$  contains  $p$  as a contract and  $p$  satisfies  $c$  then
13:         $BC := BC \cup c$ ,

```

```

14:     $u := u + v_c$ 
15:    if  $u > Th$  then
16:       $B := B \cup (u, BC)$ 

```

[Step4: Winner Determination] メディエータは全ての入札の組み合わせを考慮し、最終的な合意を決定する。具体的には、まず各入札が持つ論点に関する値の範囲の共通部分を求める。ここでは各エージェントにつき1つの入札を採用し、互いに無矛盾であるものとする。2つ以上の合意案が見つかった場合、メディエータはそれらの合意案の中で最も効用が高いものを最終的な合意として選択する。メディエータは社会効用を最大化する入札の組み合わせを見つける際に枝刈り付きの幅優先探索を行う。

Step4 は以下のように記述される。

B : A set of bid-set of each agent ($B = \{B_0, B_1, \dots, B_n\}$, a set of bids from agent i is $B_i = \{b_{i,0}, b_{i,1}, \dots, b_{i,m_e}\}$)

```

1: procedure search_solution( $B$ )
2:    $SC := B_0, i := 1$ 
3:   while  $i < |Ag|$  do
4:      $SC' := \emptyset$ 
5:     for each  $s \in SC$  do
6:       for each  $b_{i,j} \in B_i$  do
7:          $s' := s \cup b_{i,j}$ 
8:         if  $s'$  is consistent then
9:            $SC' := SC' \cup s'$ 
10:     $SC := SC', i := i + 1$ 
11:    $maxSolution := getMaxSolution(SC)$ 
12:   return  $maxSolution$ 

```

上記のアプローチを用いることで、最適解の発見を保証できることは容易に説明できる。全てのエージェントが網羅的にサンプリングを行い、かつ閾値を0に設定しているのならばエージェントは自身の全ての効用空間を入札によって表現することができる。そして全てのエージェントはメディエータに入札を渡し、それを用いて網羅的な探索を行うので、最適解の発見は保証されるのである。しかし、入札生成と最適解の発見の際の計算コストは効用空間のサイズに依存する。現実的な適用を考えた場合、すべての空間を対象としてしまうと計算機上で実行できる計算コストを超えて

しまう。そこでサンプル数を抑え、エージェントが生成する入札数に制限を設けることで現実的な範囲で停止させる必要がある。このアプローチが現実的に適用可能なのは、問題空間が非常に小さい時のみである。

既存の手法 [5] ではすべてのエージェントの閾値はメディエータの指定された同一の数値であった。しかし閾値が同一の場合、自分の効用空間をあまり公開しないエージェントと過剰に公開するエージェントが存在し、各エージェントが平等に自分の効用空間を公開していないという問題がある。そこで、本論文で提案する閾値調整プロトコルを用いて、他のエージェントを考慮することで各エージェントごとに異なる数値の閾値を設定し、各エージェントが平等に自分の効用の情報を公開するようにする。

3 閾値調整メカニズム

3.1 閾値調整の概要

本メカニズムの前提としてメディエータに公開している範囲が多いエージェントは、公開している範囲が少ないエージェントに対してより多くの範囲を公開するように説得を行えるとする。つまり、もし合意に達しなかった場合は、公開範囲の少ないエージェントは合意が形成できるようにより多くの範囲を公開するようにすべきといえる。ここでいう公開範囲とは図3が示すように、ある閾値においてどれほどエージェント自身が持つ効用空間をメディエータに対して公開したかである。

まず、エージェントで統一の閾値(初期値)を設定する。初期の閾値は後々下げていくため、高めの値に設定しておく。その後、その初期値で既存の手法で最適解を見つける。もし最適解が見つかった場合はそこで終了する。もし見つからない場合は現在の閾値におい

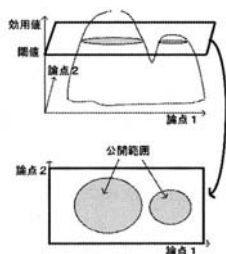


図3 公開範囲

て各エージェントが自分の効用空間をどれほど公開しているかを計算しメディエータに報告する。そしてメディエータは、全エージェントのうち個々のエージェントが、どれくらいの割合を公開しているかをもとにして次のラウンドの下げ幅を決定し、各エージェントに下げ幅分閾値を下げるように指令をだす。下げ幅を決定する際、メディエータは公開範囲が少ないエージェントには下げ幅を大きく、公開範囲が大きいエージェントには下げ幅を小さくするなるに下げ幅を決める。つまり、閾値を下げることは公開範囲が大きくなるので、次のラウンドでの各エージェント公開範囲を調整することになるのである。そして、この操作をエージェント間で合意が形成されるかもしくは合意が発見できないと判断されるまでこの操作が繰り返される。

図4に3エージェントにおける閾値調整のプロセスを示す。矢印の上の部分か閾値調整をする前の閾値と公開範囲、下の部分が閾値調整した後の閾値と公開範囲を示している。閾値調整をする前の公開範囲が狭い Agent 3 に注目してほしい。閾値調整をした後、Agent 3 の下げ幅が最も大きく、公開範囲が3エージェントの中で最大となっている。このように公開範囲が少ないエージェントは閾値の下げ幅を増やすことで公開範囲が大きくなる。

以下に閾値調整メカニズムの詳細を示す。

Ar: Area Range of each agent

$$(Ar = \{Ar_0, Ar_1, \dots, Ar_n\})$$

- 1: procedure threshold_adjustment()
- 2: loop:
- 3: $i := 1, B := \emptyset$
- 4: while $i < |Ag|$ do
- 5: bid_generation_with_SA(Th_i, V, SN, T, B_i)
- 6: $maxSolution := search_solution(B)$
- 7: if find $maxSolution$
- 8: break loop
- 9: elseif all agent can lower the threshold
- 10: $i := 1$
- 11: $SumAr := \sum_{i \in |Ag|} Ar_i$
- 12: while $i < |Ag|$ do
- 13: $Th_i := Th_i - C * (SumAr - Ar_i) / SumAr$
- 14: $i := i + 1$
- 15: end while
- 16: else
- 17: break loop

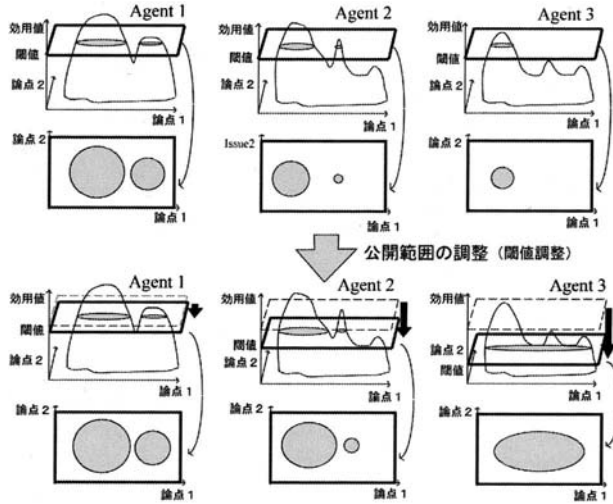


図4 閾値調整プロセス

18: `return maxSolution`

`bid_generation_with_SA()` は前章の Step1~3 を行う関数, `search_solution()` は前章の Step4 操作を行う関数である。

上のアルゴリズムは前章での Step1, Step2, Step3, Step4 を利用している。

既存の手法 [5] ではこれらのステップを何度も繰り返すことはしなかったが, 新たに提案した閾値調整を用いた手法では何度もこれらのステップを繰り返すことで過剰にエージェントの効用情報を公開することを防いでいる。

3.2 計算量の削減

前章で述べた既存の交渉プロトコルにおいて, 組み合わせ最適化問題を解くステップ, Step4 を行う際にもっとも計算量を要する。一方, 新たに提案する手法ではエージェントは少しずつ閾値を下げており, 無駄な範囲を入札として生成しない。よって, 入札の組み合わせ最適化問題を解く際にかかる計算量を削減できる。

図5で計算量を減らしていることを説明する。図5では二人のエージェントの交渉の様子を示している。

既存の閾値調整を行わない手法 [5] では, 低い閾値 (点線) を入札を生成する際に用いるため, 図5で, 灰色の部分も入札を行う。このため, メディエータは灰色部分の入札数に対して組み合わせ最適化問題を解かなければならない。つまり, 図5の灰色部分を総当たりで探索する既存の手法 [5] は, 指数関数的な時間が

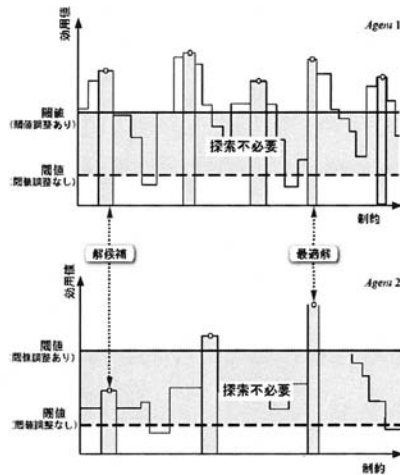


図5 計算量削減の概念図

一方、閾値調整を用いた場合、高い値から少しずつ下げていくため図5の灰色部分を探索することなく高い閾値で終了できる。これにより Step4 での計算量を削減でき、全体としての計算量を減らしている。

4 評価実験

4.1 実験の設定

本実験では、ランダムに生成された効用関数をもつエージェント間の交渉を 100 回試行した結果の平均値を取る。

最適率を求める実験で、網羅的な探索を行った場合、問題がスケールアップするにつれて計算量的困難が生じる。そこで、最適値に近似した値を求めるために、シミュレーテッドアニーリング (SA) [6] を用いることとし、比較の為に基準として用いる。SA は初期の温度を 50 度として、2500 回の繰り返し処理を経て温度を 0 まで下げることで、初期解はランダムに選択する。また、公開範囲の実験では、効用空間における格子状の 1 点の効用値を公開した時、1 だけ公開したと指標を定義して評価を行った。もし 100 点公開したら 100 だけ効用を公開したということとなる。本実験におけるパラメータは以下の通りである。

- エージェント数：3
- 論点の値域：[0,9]
- 制約数：10(単項制約), 5(二項制約), 5(三項制約), すなわち、単項制約を除く各次元の制約数は 5。
- 制約の最大効用：100 *(論点数)。従って多くの論点に関して条件を満たす制約は、平均して、効用がより高くなる。本設定は、多くの問題領域において妥当だと考えられている。例えばミーティングスケジューリングにおいて、各論点を個々の参加者のスケジューリングの可否として定義した場合、より多くの参加者に関する制約が、より少数の参加者に関する制約より重要とするのは、自然な設定である。
- 制約の最大範囲：7。この設定の下では、例えば、以下の制約が妥当なものとして生成される。(論点 1, 論点 2, 論点 3) = ([2,6],[2,9],[1,3])。
- サンプル数：(論点数) *200

- 個々のエージェントがサンプリングの際に行うシミュレーテッドアニーリングでは初期の温度を 30 とし、処理を 30 回繰り返す。サンプリング時のシミュレーテッドアニーリングの目的はサンプリングポイント周辺の(局所的な)最適解を得ることである。そのため、探索範囲が拡大しすぎないように初期の温度設定を低めにし、処理回数も少なく抑えてある。
- 閾値調整を使用した方法では初期の閾値を 900 として下げていき、閾値が 200 となったなら、それ以降は下げない。
- 閾値調整を行わない方法では入札の際の閾値を 200 としている。
- 閾値調整を行う方法では下げ幅決める式として $50 * (\text{SumAr} - \text{Ar}_i / \text{sumAr})$ を使用した。(SumAr : 全エージェントの公開範囲の和, Ar_i : エージェント i の公開範囲)
- 最適率を求める実験における閾値調整を行わない手法と、公開範囲に関する実験の閾値調整を行わない手法の数の制限をかけている場合は 1 エージェント辺りの上限を $\sqrt[3]{6400000}$ (N: エージェント数) とする。これは全てのエージェントの入札数の積が計算機上で行える限界 6400000 を越えないようにするためである。また、公開範囲を求める実験では論点数が 6 以上になると計算機上で行える限界を超えてしまうため行うことはできない。そのため制限がなしの数値は理想的な場合として実験している。
- 本実験のためのプログラムは JAVA2(1.5) で記述し、Mac OS 10.4 が動作している iMac (Core2Duo 2.33GHz メモリ 1.5GB) 上で実験を行った。

4.2 実験結果

図 6 に各手法の効用空間の比較を示す。“(A) 閾値調整あり”の折れ線は、本論文で提案する閾値調整を用いた手法を示す。“(B) 閾値調整なし(入札数制限あり)”の折れ線は、既存の手法 [5] で使われている入札数を $\sqrt[3]{6400000}$ (N: エージェント数) で制限し、閾値調整を行わない手法を示す。“(C) 閾値調整なし(入札

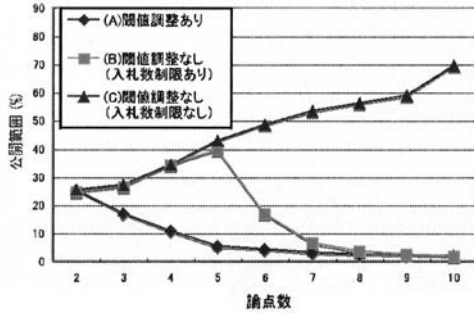


図 6 公開範囲の比較

数制限なし)”で示す折れ線は、閾値調整と入札数制限を行わない場合を示す。また比較の指標として(公開率) = (公開範囲)/(全体の効用空間の範囲)を用いた。

“(C) 閾値調整なし(入札数制限なし)”においては論点数の増加に従って公開率が増加している。これから分かるように、もし入札数の制限や閾値調整を行わないと他の方法に比べて過剰に情報を公開していることが分かる。

次に“(B) 閾値調整なし(入札数制限あり)”においては論点数5より大きくなると公開率が下がっている。これは論点数5以上になると入札数が制限されるからである。しかし、制限範囲は経験的に定められた値であるため、適切な制限範囲を定めるのは困難であり“(A) 閾値調整あり”の場合に比べて論点数が3~7の範囲では多くの自分の効用の情報を公開している。

一方、“(A) 閾値調整あり”においては公開率は他の閾値調整を実施しない方法に比べて押さえられている。このように閾値調整を行うことで無駄に自分の効用情報を公開することを防ぎながら交渉を行えることが分かる。

図7では“(A) 閾値調整あり”と“(B) 閾値調整なし”の最適率の比較を行っている。図7の最適率のグラフから閾値調整を行った場合と行わない場合では差はほとんど無いといえる。しかし論点数が3~7周辺では閾値調整を行った場合の最適率が少し下がって

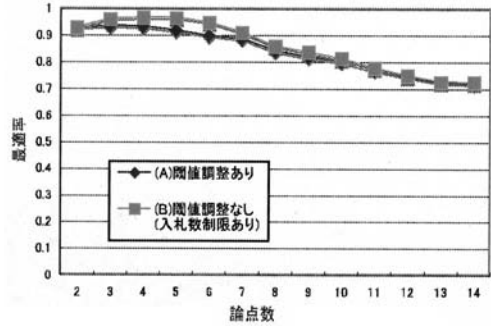


図 7 最適率の比較

いる。最適率が下がる理由は任意のエージェントが高い効用を持ち、そのエージェントが高い効用を持つ部分において他のエージェントが低い効用を持つ場合、閾値調整をしなければ発見できるが、閾値調整をすることでより高い最適解となる組み合わせを見逃してしまうからである。しかし、上記のようになる場合は少なく、最適率に大きな差がない。

以上の実験から閾値調整を行う手法を用いることで最適率の大きな変化なしにエージェントの公開範囲を最低限にすることが可能であることが分かる。

図8(1)では各エージェントがもつ入札数の平均を示している。入札数が大きいほど組み合わせ最適化のために多くの計算量が必要である。“(C) 閾値調整なし(入札数制限なし)”では入札数が指数関数的に増加している。また、図8(1)では、“(A) 閾値調整あり”、“(B) 閾値調整なし(入札数制限あり)”の差が分かりにくいので、図8(2)に拡大して示す。提案した“(A) 閾値調整あり”のグラフは全ての論点数において効果的に入札数を減らしている。一方“(B) 閾値調整なし(入札制限あり)”では論点数6以降一定値を取り続けている。これは論点数6以降で入札数の制限が行われていると示している。入札数制限が行われる前の論点数では入札数は閾値調整を行う場合よりかなり大きく、計算時間がよりかかっている。これより閾値調整が計算量の観点から見ても効果的であることが分かる。

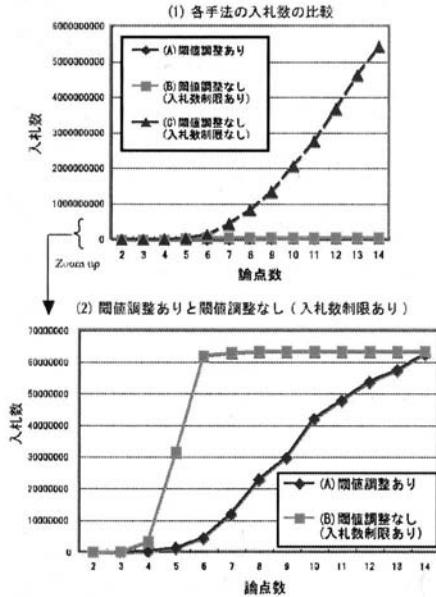


図8 入札数の比較

5 おわりに

本論文ではマルチエージェントの複数論点交渉問題における公開範囲調整に基づく交渉手法を提案した。また、高い最適率を保ったままメタデータに公開する範囲を最低限にすることをシミュレーション実験により示した。人の代わり交渉を行うエージェントを考えたとき個人情報である自分の効用空間をできる限り公開しないで交渉を行うことはより現実的な設定である。さらに提案した手法は、最も計算量を要する組み合わせ最適解を求めるステップの計算量を減らし、全体として計算量が減らしていることを示した。

参考文献

- [1] P. Faratin, C. Sierra and N.R. Jennings: "Using similarity criteria to make issue trade-offs in automated negotiations", *Artificial Intelligence*, pp. 142:205-237 (2002).
- [2] L.K. Soh and X.Li: "Adaptive, confidence-based multiagent negotiation strategy", *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems(AAMAS2004)* (2004).
- [3] S.Fatima, M.Wooldridge and N.R. Jennings: "Optimal negotiation of multiple issues in incomplete information settings", *Proc. of Autonomous Agents and Multi-Agent Systems (AAMAS2004)* (2004).
- [4] R.Y.K.Lau: "Towards genetically optimised multi-agent multi-issue negotiations", *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS05)*(2005).
- [5] T.Ito, H.Hattori, and M.Klein. Multi-issue negotiation protocol for agents: Exploring nonlinear utility spaces In *JCAI-2007*, pp. 1347-1352, (2007).
- [6] S.J.Russell and P.Norving: "Artificial Intelligence: A Modern Approach", Prentice Hall(2002).
- [7] M.Klein and P.Faratin and H.Sayama and Y. Bar-Yam: "Negotiating Complex Contracts", *Group Decision and Negotiation*, 12 (2): 58-53 (2003).
- [8] T.Bosse and C. M. Jonker: "Human vs. Computer Behaviour in Multi-Issue Negotiation", *Proc. of First International Workshop on Rational, Robust, and Secure Negotiations in Multi-Agent Systems* pp. 11- 24 (RRS-2005)(2005).
- [9] R. J. Lin and Sengcho T Chou: "Bilateral Multi-Issue Negotiations in a Dynamic Environment", *Proc. of AMEC-2003* (2003)
- [10] M. Barbuceanu and W. Lo: "Multi-attribute Utility Theoretic Negotiation for Electronic Commerce", *Proc. of AMEC-2000*, pp 15-308(2000)
- [11] K.Fujita and T.Ito: "A Preliminary Analysis of Computational Complexity of the Threshold Adjusting Mechanism in Multi-Issue Negotiations", *RRS2007*(2007)
- [12] K.Fujita and T.Ito, "An Approach to Implementing A Threshold Adjusting Mechanism in Very Complex Negotiations: A Preliminary Result", *KICSS2007*(2007)
- [13] 藤田桂英, 伊藤孝行, "多論点交渉問題におけるエージェントの公開情報の調整に基づく交渉手段", *FIT2007*(2007)