

複数論点交渉問題のための 効用空間の絞り込みに基づくマルチエージェント交渉手法

服部 宏充^{†,††} 伊藤 孝行^{†††} Mark Klein[†]

Using Iterative Narrowing to Enable Multi-Party Negotiations with Multiple Interdependent Issues

Hiromitsu HATTORI^{†,††}, Takayuki ITO^{†††}, and Mark KLEIN[†]

あらまし マルチエージェントシステムの研究分野において、複数の論点を含む交渉問題が注目を集めている。複数論点交渉問題に関する既存の研究は、個々の論点の独立性を仮定しており、問題設定が現実的ではなかった。筆者らは、複数の論点が依存関係を持つ、すなわちエージェントの効用が非線形の効用関数で表現される交渉問題に注目している。ここでは、複雑な効用空間に対して、エージェントに対して、過剰に詳細な選好情報の開示を要求することなく、効用の高い解を得ることが課題となる。本論文では、これらの課題を解決するための新たな交渉プロトコルを提案する。提案プロトコルでは、エージェントが合意に成功し、高い効用を得られる可能性を保ちながら、効用空間の絞り込みを反復実行する。本プロトコルの初期のラウンドでは、エージェントは、効用空間の中で有望な部分空間を粗く表明する入札を行い、全てのエージェントにとって有望な空間を把握し、次に、その空間のみを対象とする、より詳細な入札を行う。実験により、大規模な非線形効用空間に対して、提案プロトコルは最適解に近い解が得られることを示し、またエージェント数が大きくなった場合でも、現実的な計算量で解が得られる事を示す。

キーワード 複数論点交渉問題, 非線形効用, マルチエージェントシステム, 入札

1. まえがき

実世界の交渉問題では、通常、複数の論点が存在することから、マルチエージェント研究分野においても、複数論点交渉問題が重要な研究課題となってきた。複数の論点の存在を仮定した研究はこれまでも行われているが [1] ~ [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単一の最適点を持つ線形の効用関数として表現が可能であった。しかし、実世界の交渉問題では、複数の論点が全て独立しているケースは稀である。筆者等は、複数の論点が依存関係を持ち、エージェントの効用が多峰性を持つ

非線形効用関数として表現される複雑な交渉問題に注目しており [5] ~ [7], 詳細な選好情報の過剰な開示をすることなく、個々のエージェントが個人効用の最適化を試み、かつ社会的にも効用の高い合意を得る手法の開発を課題としている。

本論文では、入札と合意の決定を反復実行しながら、徐々に解空間を絞り込み、比較的大規模な問題に対しても、高い社会的効用と合意形成の成功率が得られる、非線形効用関数に適したマルチラウンドの交渉プロトコルを提案する。本提案プロトコルにおいて、初期のラウンドでは、エージェントが、個々の効用空間の中で有望な部分空間を粗く表明する入札を行う。その後、前のラウンドにおける入札から得られた、全てのエージェントにとって効用を得ることが可能な部分空間のみを対象にして、より詳細な入札を行い、最終的な解を得る。実験の評価により、非線形効用空間に対して、提案プロトコルが最適解に近似した解を得られる事を示す。非線形の効用関数に注目した、複数論点交渉問題に関する少数の既存研究において、例えば、

[†] マサチューセッツ工科大学, ケンブリッジ市
Sloan School of Management, Massachusetts Institute of Technology, NE20-336, Cambridge, MA 02142

^{††} 京都大学大学院情報学研究所, 京都市
Graduate School of Informatics, Kyoto University, Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501

^{†††} 名古屋工業大学大学院工学研究科, 名古屋市
Graduate School of Engineering, Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya, Aichi 466-8555

文献 [8] では、一人の売り手（不動産業者）と一人の買い手（借家人）による one-to-one の小規模な交渉問題が想定されており、文献 [9] では、可能な解の数が 27 個となる問題に基づいて議論が行われるなど、対象が小規模な問題であった。本論文では、より大規模な問題を想定しており、実験において、多数のエージェントが存在する場合でも提案プロトコルによる合意形成が可能であることを示している。

本論文の構成を以下に示す。2. では、本論文で注目している、エージェントの効用が非線形効用関数で表現される交渉問題のモデルについて述べる。3. では、効用空間の絞り込みを反復して行う、入札に基づいた交渉プロトコルについて述べる。次に、4. で、エージェント数、効用関数の起伏、およびエージェント間の効用関数に関する相関度を変化させた場合について、提案手法の性能を実験的に評価する。最後に、既存研究との比較を行い、今後の課題について議論する。

2. 非線形効用関数に基づくエージェント間交渉

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。各論点は、効用空間において直交する次元に一致する。論点 i_j は $[0, X]$ の範囲の整数を値として持つ（すなわち、 $i_j \in [0, X]$ ）。交渉の結果得られる合意案は、各論点の値のベクトル $\vec{s} = (i_1, \dots, i_M)$ 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。個々の制約は $c_k \in C$ と表し、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。制約 c_k は、合意 \vec{s} によって充足される場合にのみ、 $w_i(c_k, \vec{s})$ を効用値として持つことができる。図 1 に、論点 1、および論点 2 に関する、二項制約の例を示す。本例は、制約が充足された場合に得られる効用は 55 であり、論点 1 に関しては $[3, 7]$ 、論点 2 に関しては $[4, 6]$ の範囲内で合意が得られた場合にのみ、制約が充足可能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つ。

合意 \vec{s} に関するエージェント i の効用を $u_i(\vec{s}) = \sum_{c_k \in C} w_i(c_k, \vec{s})$ と定義する。この効用表現により、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では効用が低くなるため、空間内に効用値による高低が生じ、凹凸のある

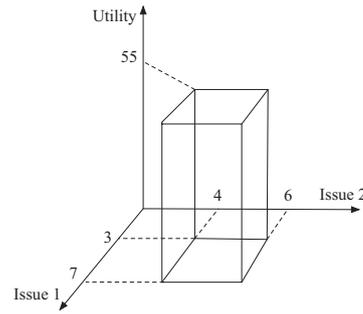


図 1 制約の例（二項制約）
Fig. 1 Example of a constraint

非線形の効用空間が形成される。ここで、本論文における効用空間とは、各論点が取り得る値のあらゆる組合せについて、効用関数によって得られる効用値を空間上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。線形の効用関数を前提とする既存の研究では、合意案の効用は個々の論点に関する効用の加重和であるため、効用空間が平坦な超平面となり、本論文で注目する非線形の効用空間の形状とは大きく異なったものとなっている。図 2 に、非線形の効用空間の例を示す。本例では、各々 $[0, 99]$ の値域を持つ 2 つの論点が存在するため、効用空間は 3 次元のグラフとして表されている。また本例では、単項制約（1 つの論点に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になる。本論文で用いる効用表現は、格子状の空間における各地点に効用値を割り当てるシンプルなものではあるが、ピクセル表現で様々な画像が近似的に描画できるのと同様、実用的な精度で様々な効用関数を近似的に表現できる。

現実を考え得るケースとして、エージェントが、合意案の効用を正確に評価するための完全な知識（効用関数）は持っているが、効用空間の規模が大きいため、最適な合意案を事前に把握することが困難である場合が考えられる。例えば、論点の数、および各論点を取り得る値が共に 10 個である場合、可能な合意案は $10^{10} = 100$ 億パターン存在することになり、網羅的な評価は計算量的に困難となる。非線形の効用関数を持つエージェント間で（準）最適な合意を発見するための探索手法として、非線形最適化手法であるシミュレーテッドアニーリングや、進化的アルゴリズムの利用が考えられるが、本論文では、これらの探索手法をエー

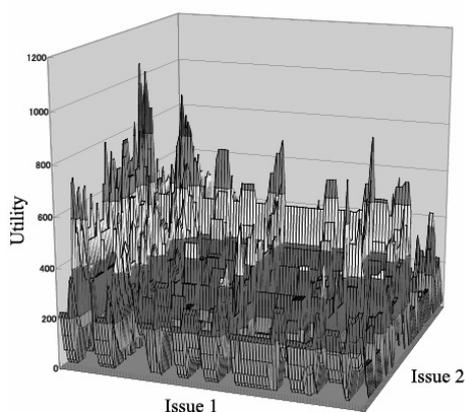


図 2 非線形効用空間の例

Fig. 2 Example of nonlinear utility space for a single agent

エージェント間交渉の解法として採用しない。本論文におけるエージェントは、利己的な効用最大化者であることを仮定している。これは、既存の研究においても頻繁に用いられる一般的な仮定である [10] ~ [12]。そのため、解の探索を行うために、エージェントが効用に関する完全な情報を公開する事は、必ずしも期待できない。なぜなら、全ての情報を公開することは、自分にとって不利となり得る情報も併せて公開することになるからである。従って、エージェントから効用に関する完全な情報を一カ所に集め、非線形最適化手法に基づいて求解を行う方法は非現実的である。また、以上の点から、本論文では、各エージェントは、交渉における有意性を確保するため、効用関数に関する詳細な情報の開示を出来る限り抑えることを好むとする。

本論文では、交渉問題における一般的な仮定として、交渉によって、社会的効用、すなわち全エージェントの効用の総和を最大化する合意の発見を試みるものとする。すなわち、交渉プロトコルの目的関数は以下となる。ここで、 A はエージェントの集合を表す。

$$\max_{\vec{s}} \sum_{i \in A} u_i(\vec{s}) \quad (1)$$

本論文で提案する交渉プロトコルが有効に機能する事例として、何らかの財に関する売買交渉が考えられる。売買における売り手、および買い手の選好は複雑である。例えば、車の購入を検討している買い手は、より安く購入することを好むが、内装の充実度が十分なものであれば、より高い価格で購入しても良い、といった選好や、さらに、塗装色が気に入らなければや

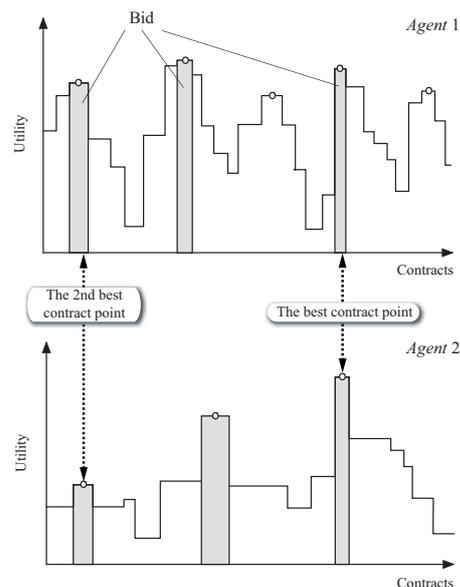


図 3 合意の決定

Fig. 3 Deal Identification

はり高い価格での購入は控える、という選好を持ち得る。売り手に関しても、多数の品数が一度の売買で売却可能であれば、価格のディスカウントを行う、といった選好を持つ事が考えられる。さらに、売買では、売り手と買い手の双方にとって、詳細な選好情報を公開する事は、自身の効用を低減させる可能性があるため、一般的に有効ではない。例えば、売り手が最低販売価格を進んで明かしてしまうと、買い手は当然その価格での購入を望み、逆に、買い手が最高購入価格を提示してしまうと、売り手はその価格での売却を希望する。このように、プレイヤー間で、得られる解に関する効用が相反する場合には、上述した通り、選好、すなわち効用に関する情報は、可能な限り明かさない事がプレイヤーにとって望ましい場合が考えられる。従って、依存関係を持つ複数の論点が存在し、かつ交渉に参加している他のプレイヤーに有用な情報を与えないために、効用に関する詳細な情報を明かさない事が求められるケースへの対応は、現実的に重要な問題である。本論文では、このような問題に対して有効に機能する交渉プロトコルを提案する。

3. 非線形効用関数を持つエージェントのための入札に基づく交渉プロトコル

エージェントが非線形効用関数を持つ場合の入札

に基づく交渉プロトコルは、主に2つのステップから成る [6] [7] .

入札の生成: エージェントは、自身の効用空間の中から、適切な部分空間を選び出す。次に、選び出された空間に解（合意）が含まれる事を求める入札を生成し、効用値を割り当てて、メディーエータ（オークション）に送信する。図3を例として具体的に述べる。本例では、横軸は可能な解を表し、縦軸は効用を表す。図中の白丸は、エージェントが入札を生成するために発見した、効用が高い解を表す。エージェントは、高い効用が得られる解を発見すると、その周辺に存在する、同一の効用が得られる解を調べる。それらの解が存在する部分空間の和集合を領域として、1つの入札を生成する。図3中の色付けされた矩形が各々1つの入札を表し、ここでは、2つのエージェントが、それぞれ3つの入札を生成している。

合意の決定: メディーエータは、全ての入札の組合せを考慮し、合意を決定する。ここで得られる合意は、各エージェントにつき、少なくとも1つの入札を含む。エージェントから複数の入札が生成されていれば、合意案は複数存在し得る。例えば、図3では、2つのエージェント間で、領域が重複する入札の組合せが2つ存在している。全てのエージェントが、個々の効用空間に存在する全ての合意案を提示し、かつメディーエータが網羅的に入札の組合せを探索して社会的効用を最大化する解を得るならば、シングルラウンドで最適解を得ることが保証される。しかし、2.で述べた通り、エージェントが効用関数に関する全ての情報を公開することは非現実的な設定である。さらに、メディーエータによる合意の決定を現実的な時間内で実行するためには、各エージェントの入札数に上限を設ける必要がある。本研究で現在用いている合意決定アルゴリズムでは、入札の組合せの数が6,400,000以内^(注1)であれば、現実的な時間内で解の計算が可能である。従って、エージェント数が N の場合、各エージェントに対する入札数の上限は、 $\sqrt[3]{6400000}$ となる。各エージェン

(注1): ここでの6,400,000という入札の組合せの上限は、現在想定している問題サイズとアルゴリズムにおいて、メモリ溢れを起こさず、求解を現実的な時間内で終わらせる事を考慮して、経験的に得られた値である。ハードウェアの性能（具体的にはメモリ量）が高くなればこの数値は変動するが、想定する問題の規模を大きくした場合、現実的な時間内で解を得るためには、やはり何らかの上限の設定が必要となる。従って、入札の組合せに上限を持つことで、個々のエージェントに対して入札数の上限が課される状況を考慮する事が必要である。

エージェント数	入札数の上限
2	2530
3	186
4	50
5	23
6	13
7	9
8	7
9	6
10	5

表1 入札数の上限

Table 1 Limit of the Number of Bids

トに課す上限は、エージェント数の増加に伴って表1のように低くなる。

エージェントは個人効用の最大化を求めるため、自分にとって最も効用の高い効用空間を用いて入札を生成しようとする。エージェント数が増加すると、たとえ効用空間中に高効用の部分空間が多数存在しても、その一部しか入札として利用できなくなる。その結果、高い社会的効用を得ることが困難になり、また、解を一つも発見できない、すなわち合意に失敗する場合も生じる。

合意形成に関する困難について、筆者らはすでに実験的に確認をしている [6]。各エージェントが、一様分布乱数に基づいて生成された効用を持ち、最も効用の高い $\sqrt[3]{6400000}$ 個の入札を提示する場合、シングルラウンドの交渉プロトコルでは、エージェント数が5に達すると、合意形成の失敗率が高くなり、例えば、エージェント数が5で、論点数が7の時点で合意形成の失敗率が40%近くまで上昇する。この結果から、本論文では、シングルラウンドの交渉プロトコルを複数ラウンドのプロトコルへと拡張し、入札と合意の決定を複数回実行することで、入札生成の際に考慮する効用空間のサイズを徐々に絞り込むアプローチを試みる。各ラウンドで、エージェントは、効用がより低い部分を除外し、効用がより高い部分が次のラウンドに残るような入札を生成する。それらの入札に基づいて合意を決定することで、全てのエージェントにとって効用が得ることが可能な部分空間が得られる。具体的には、合意を構成する各入札が示す部分空間が重複する空間が得られる。次のラウンドでは、エージェントが入札を生成する範囲は、この重複空間内に限られる。ラウンドが進むごとに問題の規模が小さくなるため、十分な選好表明のために必要な入札数も小さくなる。従って、本アプローチは、効用の高い合意形成を目的とする交渉において、エージェント数に関するスケラビ

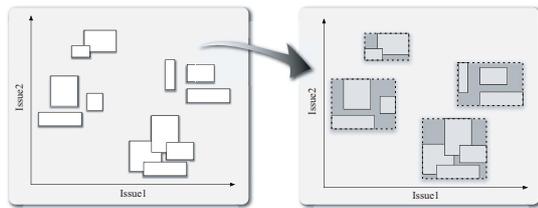


図 4 クラスター入札の例

Fig. 4 An example of a cluster-bid

リティを持つ．各ラウンドにおいて除外される有望な合意案の数が小さくなるため，合意形成の失敗率は大幅に減少し，また社会的効用の期待値も増加する．

本論文で提案する交渉プロトコルは，各々異なる交渉スキームを用いる 3 つのラウンドから成る．第一ラウンドでは，エージェントは，有望な合意案を含む領域を粗く特定するための "クラスター入札" を行う．次に，第二ラウンドでは，クラスター入札に基づいて特定された空間内で，さらに有望な部分空間を絞り込むための "最大制約入札" を行う．最後のラウンドでは，ここまで絞り込まれた空間の中から最終的な合意を決定するための "ピーク入札" を行う．各々の詳細を次節以降で述べる．

3.1 クラスター入札

クラスター入札の生成のため，まず，エージェントは制約のクラスタリングを行う．生成された各制約クラスターは，一つのクラスター入札となる．クラスター入札がカバーする領域は，クラスターに含まれる全ての制約の和集合である．図 4 に，二つの論点に関する二項制約を用いたクラスター入札の生成例を示す．実線の矩形は個々の制約がカバーする領域を表し，色付けされた破線の矩形はクラスター入札がカバーする領域を表す．図に示す通り，クラスター入札は，どの制約も充足しない空白の領域を含むが，少なくとも一つの制約を充足する空間を全てカバーしている．クラスター入札によって，効用が得られる可能性が高い空間を表明できると共に，合意には含まれることの無い領域を除外できる．クラスター入札の生成時において，どの程度の領域が除外されるかは，制約の効用空間における分布状態に依存する．制約が，効用空間中の比較的狭い（複数の）領域に固まって存在する場合，クラスター入札による効用空間の絞り込みは効果的に機能すると考えられる．

クラスター入札の生成アルゴリズムを以下に示す．

A: a set of agents

C: the constraints for an agent

```

1: function createClusterBids (C)
2:   CP := the center points for the constraints in C
3:   CL := clustering (CP, |A|-th root of 6400000)
4:   B := ∅
5:   foreach cluster cl ∈ CL do
6:     d := ∅;
7:     foreach constraint c included in cl do
8:       d := d ∪ the domain of c
9:     end foreach
10:    B := B ∪ {d}
11:   end foreach
12:   return B

```

エージェントは，効用空間中に $\sqrt[|A|]{6400000}$ を上限として制約のクラスターを生成し，生成されたクラスターから入札を生成する．ここで，エージェント数が多いために入札数の上限が低くなっている場合，一つのクラスターでより多くの制約をカバーしなければならないため，個々のクラスターサイズが大きくなってしまふ点に注意が必要である．また，アルゴリズムに示す通り，クラスター入札には，効用値が設定されない．これは，第一ラウンドでは，効用が得られる空間を特定することが目的で，このラウンドで得られる合意の効用には意味がなく，また，クラスター入札の数が入札数の上限を超えることは無いため，後のラウンドで行われるような，効用による入札の優先付けをする必要が無いためである．

クラスター入札によって有望であると表明される効用空間のサイズは，以後のラウンドで生成される入札に比べて大きい．しかし，各エージェントの効用空間に関して，詳細なトポロジーが公開されるわけではなく，また効用値も明かされない．従って，クラスター入札を行った段階では，入札を提示したエージェントの選好は明確にはならない．特に，クラスター入札がカバーする領域には，実際には効用が得られない部分も含まれているため，選好を推定するための情報としては不十分と言える．

3.2 最大制約入札

最大制約入札は，クラスター入札に基づいて得られた合意案がカバーする領域内で，有望な部分を特定するために生成される．図 5 に，最大制約入札の例を示す．エージェントは，第一ラウンドで得られた合意案がカバーする領域内の任意の地点を選択し，その地点を起点としたシミュレーテッドアニーリングを実行して，各々の領域内での局所最適解を発見する．次に，見つかった局所最適解によって充足可能な制約を特定し，カバーする領域のサイズが最も大きい制約のみを

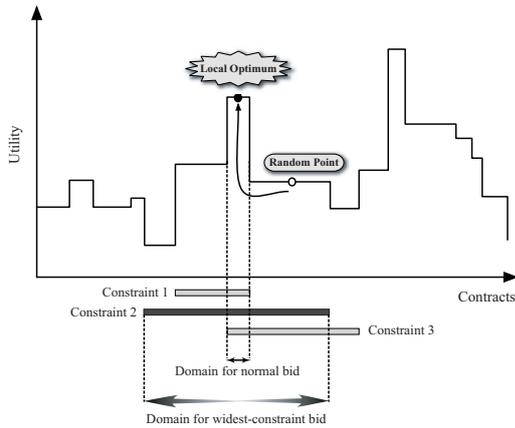


図 5 最大制約入札の例
Fig. 5 An Example of a Widest Bid

用いて、入札を生成する。従って、生成される最大制約入札は局所最適解を充足することが保証され、その周辺の効用が得られる空間もカバーすることになる。つまり、局所最適解を含むことから、高い効用が得られる可能性を維持し、また合意可能な領域を広く持つことで、合意形成に成功する可能性をも維持している。また、クラスター入札とは異なり、最大制約入札には、空白の領域、すなわち効用が 0 の領域が含まれないことも保証される。

ここで重要な点は、第二ラウンド（および第三ラウンド）では、第一ラウンドよりも入札数に関する制限が緩和される可能性がある事である。前のラウンドでの合意の決定によって複数の合意が得られている場合でも、各々の合意がカバーする領域は重複しないため、それぞれの合意に関して独立に処理が可能である。より具体的には、第二ラウンドにおいて合意を決定する際に組合せを計算する入札は、同一の合意がカバーする領域内で生成されたものだけに限定できる。従って、第一ラウンドでは効用空間全体を対象として入札を生成していたが、第二ラウンド以降では、前のラウンドで得られたそれぞれの合意がカバーする、狭い領域のみを対象とした入札の生成が可能となり、計算の複雑さを大幅に削減できる。具体的には以下の通りである。エージェント数が N であり、第一ラウンドで M 個の合意が得られたとする。合意の決定において、入札の組合せの数を 6,400,000 以下に抑えるためには、エージェントの入札数の上限を、各々の合意に

つき $\sqrt[N]{6400000/M}$ とすればよい^(注2)。例えば、エージェント数が 10 で、第一ラウンドで得られた合意の数が 5 の場合、エージェントの生成できる入札数の総和は最大で $5 \times \sqrt[10]{6400000/5} = 20$ となる。一方、第一ラウンドにおける入札数の上限は 5 である。この入札数の上限の緩和により、入札によって表明される効用の高い領域が大きくなるため、最終的に得られる効用、および合意形成の失敗率の改善に関して効果的である。

最大制約入札の生成アルゴリズムを以下に示す。

```

C: the agents' set of constraints
R: an overlap region (identified in round 1)
limsmp: the number of simulated annealing searches
performed per overlap
limo: the maximum number of bids allowed per overlap
1: function createWidestConstraintBids (C,R,limsmp,limo)
2:   foreach constraint c ∈ C do
3:     if c overlaps with R then
4:       O := O ∪ overlapping region between c and R
5:   end foreach
6:   foreach overlapping region o ∈ O do
7:     for i < (limsmp/|O|) do
8:       p := randomly picked point within o
9:       p' := simulatedAnnealing (p)
10:      d := 0
11:      foreach constraint c ∈ C (c is satisfied by p') do
12:        if c's domain is bigger than d then
13:          d := the domain of c
14:          v := the value of c
15:        end if
16:      end foreach
17:      B := B ∪ {(d,v)} (iff {(d,v)} is not redundant)
18:    end for
19:  end foreach
20:  n := min(|B|,limo)
21:  B' := the n bids in B with the highest volume
22:  return B'

```

本アルゴリズムは、第一ラウンドで得られた各々の解に対して 1 回ずつ実行される。また、また、入札数が上限を超える場合は、効用がより小さい入札を除外する。

3.3 ピーク入札

第二ラウンドで得られたそれぞれの解に関して、ピーク入札が生成される。エージェントは、各々の解がカバーする領域内の任意の地点を選択し、その地点を起点としたシミュレーテッドアニーリングを実行して、各々の領域内での局所最適解を発見する。次に、各々の局所最適解に関して入札を生成するが、入札がカバーする領域は、局所最適解と同一の効用が得られる周辺

(注2): M が大き過ぎると上限が低くなってしまったため、利用する合意の数を適宜設定する必要がある。

の全ての合意案をカバーするように決定される．このような入札の領域は，局所解によって充足される全ての制約の共通部分を取ることで，容易に決定できる．

C : the agents' set of constraints
 R : an overlap region (identified in round 2)
 lim_{smp} : the number of simulated annealing searches performed per overlap
 lim_o : the maximum number of bids allowed per overlap

```

1: function createPeakBids ( $C, R, lim_{smp}, lim_o$ )
2:   foreach constraint  $c \in C$  do
3:     if  $c$  overlaps with  $R$  then
4:        $O := O \cup$  overlapping region between  $c$  and  $R$ 
5:   end foreach
6:   foreach overlapping region  $o \in O$  do
7:     for  $i < (lim_{smp}/|O|)$  do
8:        $p :=$  randomly picked point within  $o$ 
9:        $p' :=$  simulatedAnnealing ( $p$ )
10:       $d :=$  intersection of every  $c \in C$  ( $c$  is satisfied by  $p'$ )
11:       $v :=$  the value of  $p'$ 
12:       $B := B \cup \{(d, v)\}$  (iff  $\{(d, v)\}$  is not redundant)
13:    end for
14:  end foreach
15:   $n := \min(|B|, lim_o)$ 
16:   $B' :=$  the  $n$  bids in  $B$  with the highest volume
17:  return  $B'$ 

```

本アルゴリズムは，第二ラウンドで得られた各々の解に対して 1 回ずつ実行される．また，入札数が上限を超える場合は，効用がより小さい入札を除外する．

3.4 入札に基づくマルチラウンド交渉プロトコル
異なる入札スキームに基づく複数のラウンドから成る交渉プロトコルのアルゴリズムを以下に示す．

A : a set of agents
 DL : the upper limit on how many overlaps can be returned

```

1: procedure iterativeNegotiation ( $A, DL$ )
2:    $CB := \bigcup_{i \in A} \{createClusterBids(C_i)\}$ 
   ( $C_i$ : a set of constraints of an agent  $i$ )
3:    $S_{cb} :=$  identification ( $CB$ )
4:    $S_{eb} := \emptyset$ 
5:   foreach  $s \in S_{cb}$  do
6:      $R :=$  a covering region by  $s$ 
7:      $EB := \bigcup_{i \in A} \{createWidestConstraintBids(C_i, R, lim)\}$ 
8:     if  $|EB| = |A|$  then
9:        $S_{eb} := S_{eb} \cup$  identification ( $EB$ )
10:      if the number of overlaps  $> DL$  then break
11:    end if
12:  end foreach
13:   $S_{fin} := \emptyset$ 
14:  foreach  $s \in S_{eb}$  do
15:     $R :=$  a covering region by  $s$ 
16:     $FB := \bigcup_{i \in A} \{createPeakBids(C_i, R, lim)\}$ 
17:    if  $|FB| = |A|$  then
18:       $S_{fin} := S_{fin} \cup$  identification ( $FB$ )
19:      if the number of overlaps  $> DL$  then break
20:    end if
21:  end foreach
22:  return the best solution in  $S_{fin}$ 

```

```

1: function identification ( $B$ )

```

```

2:    $S := \bigcup_{j \in B_0} \{b_{0,j}\}; i := 1$ 
3:   for  $i < |A|$  do
4:      $S' := \emptyset$ 
5:     foreach  $s \in S$  do
6:       foreach  $b_{i,j} \in B_i$  do
7:          $s' := s \cup b_{i,j}$ 
8:         if  $s'$  is consistent then
9:            $S' := S' \cup s'$ 
10:        end if
11:      end foreach
12:    end foreach
13:     $S := S'$ 
14:  end for
15:  return  $S$ 

```

メディエータは，まず，全てのエージェントに対して，クラスター入札を要求し，送信されてきた入札に基づいて合意を決定する．ここで得られた合意は，全てのエージェントにとって，効用が得られる可能性のある領域を示すことになる．それ以外の領域は，合意の可能性が無い領域として以後のラウンドでは除外される．ただし，3.1 で述べた通り，クラスター入札には空白の領域が含まれるため，実際には，いくつかのエージェントにとっては効用が得られない領域も含まれる可能性がある．次に，クラスター入札から得られた各々の合意案が示す領域だけに範囲を限定した最大制約入札を，全エージェントに要求し，合意を決定する．ここで得られた合意は，各エージェントにとって，より期待効用が高い領域を含む可能性がある．最後に，メディエータは，最大制約入札から得られた各々の合意案が示す領域だけに範囲を限定したピーク入札を要求し，ここまでのラウンドと同様に合意案を得る．最終的に複数の合意案が得られる可能性があるが，メディエータは最も効用の高い合意案を選択し，最終的な解とする（図 3 参照）．

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持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．各試行では，全エージェントの効用関数に対してシミュレーテッドアニーリング (SA) [13] を用いて，最大の社会的効用を求める．ここで得られる値は，提案手法の経済的効率性，すなわち最適値に対してどれだけ近い値が得られるかを評価するために用いられる．

本論文では，提案した効用空間の絞り込みに基づく交渉プロトコル (IN) と，その比較対象として，山

登り法 (HC) の二つの交渉プロトコルの評価を行う。HC プロトコルは、山登り法に基づき、メディーエータを介して行われる交渉プロトコルである [14]。探索は、ランダムに生成した初期解から開始される。メディーエータは、初期解を変形した新たな解候補を生成し、交渉に参加しているエージェントに送信する。送信された新たな解候補を、全てのエージェントが以前の解よりも好ましいと判断した場合のみ、解が更新される。探索は、全てのエージェントにとって受け入れ可能な解候補をメディーエータが生成できなくなるまで継続する。アルゴリズムは以下の通りである：

```

I: A set of issues,  $I = \{i_1, i_2, \dots, i_n\}$ 
V: A set of values for each issue,  $V_n$  is for an issue  $n$ 
1: function systematicLS (I, V)
2:    $S :=$  initial solution (set randomly)
3:   foreach  $i \in I$  do
4:     foreach  $j \in V_i$  do
5:        $S' := S$  with issue  $i$ 's value set to  $j$ 
6:       if all agents accept  $S'$  then  $S = S'$ 
7:     end foreach
8:   end foreach
9:   return  $S$ 

```

上記のアルゴリズムに基づく実装では、1つの論点に関して、全ての可能な変更が一回ずつ提案されるため、繰り返し処理のステップ数は、高々、論点の取りうる値と論点の数の積となる。例えば、値域が $[0, 9]$ で、整数を値として持つ論点が 10 個ある場合、ステップ数は 100 となる。HC プロトコルは、線形の効用空間を対象とした既存研究で線形の交渉問題に適用され、効用の高い解を得ることに成功したプロトコルの典型的なものであるため、本実験における比較対象とした [5]。

本実験における各パラメータは以下の通りである。

- エージェント数：2 ~ 10
- 効用空間：論点数は 10。各論点の値域は $[0, 9]$
- 効用関数：各エージェントの効用関数は、50 個の 10 次元制約から成る。制約において、各論点の値域のサイズは、最小で 3、かつ最大で 7 の範囲でランダムに決定される。従って、例えば、 $($ 論点 1, 論点 2 $) = ([2, 6], [2, 8])$ は妥当な制約として生成される。各制約の効用は $[1, 100]$ の範囲でランダムに決定される。

また、本実験における効用関数は、複数の制約をグループ化したクラスターによって構成される。クラスター数は 10 で、各クラスターに含まれる制約数は 5 である。クラスターサイズが平均して小さい場合、効用の高い領域が、効用が得られない広い領域に囲ま

れた効用空間が生成されることになる。逆に、クラスターサイズが大きい場合は、クラスター同士が重複する可能性も大きくなり、その場合は、山と谷が入り組んだ領域が多く存在する効用空間が生成される。本実験では、共有クラスターの数を変化させることにより、エージェントの効用関数の相関度を操作し、評価する。共有クラスターは、全てのエージェントによって共有されるクラスターであり、ゆえに全てのエージェントが、共有クラスターに含まれる制約を共通して持つことになる。ただし、共有クラスターに含まれる制約の効用は、各エージェントごとに異なる。共有クラスター以外のクラスターについては、各エージェントにつき、ランダムに設定される。本実験における効用関数の生成の手順を以下に示す。効用関数を生成するために、パラメータを設定する。ここで設定するパラメータは、論点数、値域サイズ、クラスターサイズ、制約の (各論点に関する) 値域サイズ、総クラスター数、共有クラスター数、エージェント数、および最大クラスター内制約数である。上述の通り、論点数、値域サイズ、制約の値域サイズ、総クラスター数、最大クラスター内制約数は、それぞれ、10, 10 ($[0, 9]$ であるため), 3~7, 10, および 5 である。パラメータの設定に続いてまず共有クラスターの生成を行う。ここでは、各共有クラスターのサイズと位置が決定される。共有クラスターのサイズは、パラメータ "クラスターサイズ" 以下でランダムに決定され、共有クラスターの位置は、クラスターの中心となる点が効用空間内でランダムに選択される。これにより、ランダムに決定された点を中心として、"クラスターサイズ" 以下のサイズを持つ超立方体として共有クラスターが定義される。次に、各エージェントに関して、プライベートクラスター、すなわち他のエージェントと共有しないクラスターを生成する。プライベートクラスターの生成方法は、共有クラスターと同様である。生成された全クラスターそれぞれに関して、パラメータ "最大クラスター内制約数" を上限として、各クラスター内に収まる制約を生成する。制約の生成では、各々の制約の中心が、クラスターの超立方体内にランダムにプロットされ、パラメータ "制約の値域サイズ" 以下で各論点に関する値域をランダムに決定する。ここでは、制約がカバーする領域が、クラスターの超立方体を超えないように生成される。

3.2 で述べたように、第二・第三ラウンドでは、前のラウンドで得られた複数の合意案に関して、入札の

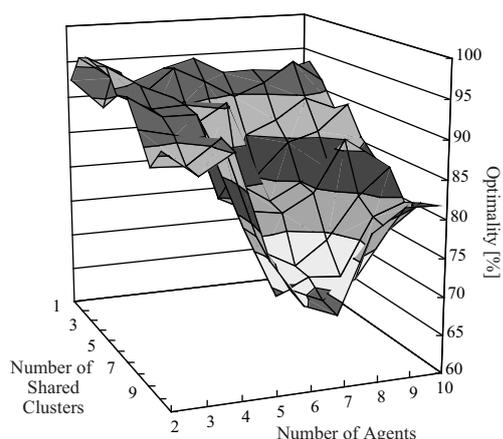


図6 提案プロトコルの最適率 (制約クラスターサイズ:小)
Fig. 6 Utility Values for the Small Cluster Case

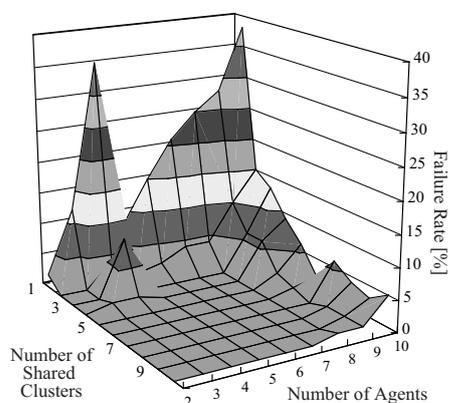


図7 合意形成の失敗率 (制約クラスターサイズ:小)
Fig. 7 Failure Rates for the Small Cluster Case

生成、および合意の決定を行うが、入札数の上限が低くなりすぎないように、本実験では、用いる合意の数を最大 50 とする。従って、前のラウンドで 50 以上の合意が得られた場合の入札数の上限は、 $\sqrt[3]{128000}$ となる。この上限を超える場合は、次のラウンドで用いる合意はランダムに選択する。

本実験では、クラスター入札の生成のために利用するクラスタリングアルゴリズムとして、CLARANS [15] を用いた。なお、本実験は Java2 (Ver. 1.4.2) で記述し、MacOSX 10.4.8 が動作している MacPro (CPU Dual 2GHz Xeon dual-core・メモリ 2GB) 上で行った。

4.2 実験結果

エージェントの効用関数を構成する制約クラスターのサイズが小さい場合、すなわちエージェントが共通して好む領域が狭い可能性が高い場合と、制約クラス

ターのサイズが大きい場合、すなわち制約クラスター (好ましい領域) が重複する可能性が高い場合について実験を行い、本論文における提案手法によって、最適解に近似した結果を得られる事を示す。図 6 に、効用関数における制約クラスターの各論点に関する値域のサイズを [3, 7] の範囲でランダムに決定した場合の最適率を示す。図に示す通り、共有クラスターの数 が 4 以下の場合、平均して 90% 程度の質の解が得られている。共有クラスター数がさらに増えた場合、すなわちエージェント間で効用関数の相関がより高くなった場合、最適率が低下している。この理由は次の通りである。効用関数の相関が高い場合、入札がカバーする領域が重複する頻度が高くなるため、多数の合意が得られる。計算量的困難により、次のラウンドでは、得られた合意の一部のみが利用され、残りは除外される。その結果、いくつかのエージェントは、より効用の高い入札を生成する機会を失うことになり、最適率の低下を招くことになる。また、エージェント数の増加に伴って最適率が低下し、その後上昇している。この理由は次のように考察できる。エージェント数が増加すると、入札に基づく合意の組合せは基本的に増加する。しかし、あるラウンドで多数の合意が得られたとしても、次のラウンドでは全ての合意が利用されるわけではない。4.1 で述べたように、本実験では、次のラウンドで利用される合意の数は最大でも 50 個である。そのため、エージェント数の増加につれて、除外される合意も増加することになる。エージェント数が 2 から 7 に増加する過程では、このような合意の除外が発生し、最適率が低下している。しかし一方で、エージェント数が増加すると、より多くのエージェントからの入札に関して、カバーする領域が重複しなければならぬため、合意は困難になっていく。さらに、エージェント数が 8 から 10 の場合、3. で示したように、入札数の上限はそれぞれ 7, 6, および 5 であり、大きな変化がない。従って、得られる合意の組合せの増加の度合いが小さくなり、かつ合意が一層困難になるため、合意の数は少なくなる。その結果、除外される入札の割合が小さくなり、最適率が逆に上昇している。第二および第三ラウンドでは、各々の合意がカバーする領域への入札数は、 $\sqrt[3]{6400000/\text{合意の数}}$ で決まるため、合意の数が少なくなるほど、狭い領域への入札数が増加する。その結果、最適率が徐々に上昇している。図 7 に、同様の設定における合意形成の失敗率を示す。エージェントの効用関数の相関が低い場合、

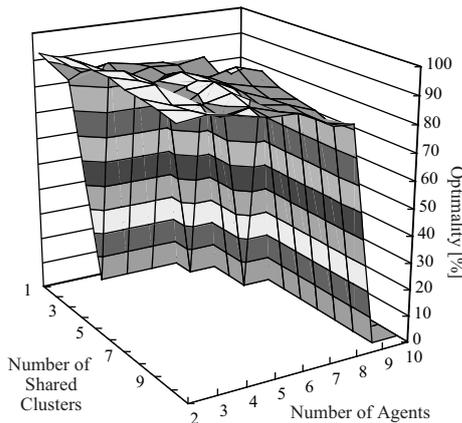


図8 提案プロトコルの最適率(制約クラスターサイズ:大)
Fig.8 Utility Values for the Medium Cluster Case

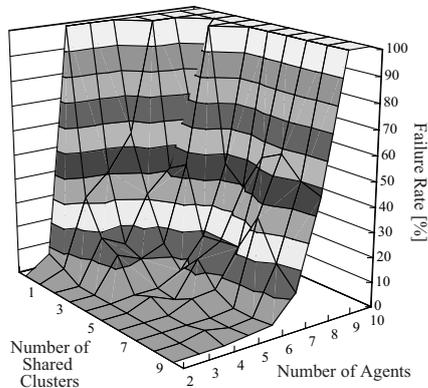


図9 合意形成の失敗率(制約クラスターサイズ:大)
Fig.9 Failure Rates for the Medium Cluster Case

すなわち共有クラスター数が1, 2個しか無い場合、全てのエージェントが満足する解の数が少なくなるため、合意形成の失敗率は高くなる。しかし、それ以外のケースでは、失敗率は一桁程度に抑えられている。

次に、効用関数における制約クラスターの各論点に関する値域のサイズを [3, 10] の範囲でランダムに決定する、クラスターのサイズがより大きい場合の結果を示す。図8に示す通り、提案プロトコルは依然として高い最適率を示している。しかし、合意形成の失敗率に関しては、より困難な問題、具体的には、エージェント数が多い、もしくは共有クラスター数が少ない場合、大幅に高くなっている(図9)。この失敗率の上昇の理由は次の通りである。クラスターサイズが大きい場合は、共有クラスター数が増えた場合のように、エージェントの入札から、多くの合意が得られる。

従って、クラスターサイズが小さい場合と同様、計算量的困難から、得られた合意の切り捨てが必要となり、その結果、失敗率が上昇することになる。

一方、HCプロトコルを用いた場合の結果は、全てのケースについて極端に悪く、平均的最適率は2%程度であった。これは、本実験で用いた効用関数は、エージェントが効用を得られない空間が相対的に多く存在する比較的求解が困難なケースではあるが、効用関数が非線形の形式で表現される、交渉問題の新しいパラダイムにおいては、既存の手法は全く機能しないことが明らかとなった。また、筆者らが文献[6]で行った実験よりも、本実験におけるHCプロトコルの解の質が悪化しているのに対して、本論文における提案手法では、高品質の解が得られている。この結果から、本論文で提案した、マルチラウンドのプロトコルへの拡張に基づく交渉プロトコルが、求解がより困難となった問題に対しても有効に機能することが確認できた。効用の高い結果を得る方法として、HCプロトコルによる求解を複数回実行し、得られた結果の中から最高の解を選ぶ事が考えられる。しかし、解の価値はエージェントごとに異なるため、“最高の解”とは何かを決定することが困難である。

さらに、交渉プロトコルのマルチラウンド化に関する評価を行った結果を表2、および図10に示す。ここでは、制約クラスターのサイズが小さい場合を用いた結果を示す。まず、表2に、クラスター入札の結果得られる解のサイズ、すなわち最大制約入札で入札対象となる空間サイズの平均値を示す。ここで、実際にクラスター入札が実行されるのは、エージェントの生成する入札数が上限 $\sqrt[3]{6400000}$ を超える場合である。本実験では、クラスター入札が実行された、エージェント数が5を超える場合について結果を示す。本実験における効用関数は、各論点の値域の平均サイズが5である、50個の10次元制約から成る。従って、効用値を与える効用空間のサイズは、 $5^{10} \times 50 (\approx 488000000)$ と概算できる。表2に示す通り、クラスター入札を用いた求解の結果、次の入札ステップで入札の対象とする空間のサイズが縮小される事が分かる。例えば、エージェント数が5で、共有クラスター数が10である場合、入札対象とする空間は、約0.005%にまで空間が縮小される。また、エージェント数が増えるほど生成可能なクラスター数が少なくなることから、各々のクラスターサイズが大きくなり、得られる解のサイズも大きくなるが、エージェント数と共有クラスター

共有クラ スター数	エージェント数					
	5	6	7	8	9	10
1	3820	62723	224371	201518	248887	586612
2	6526	57613	269513	286568	315636	834124
3	6111	55832	218649	234876	582417	1422199
4	7970	68901	354507	549852	1286446	2646808
5	7620	89056	324031	402732	1756843	3679303
6	10218	85340	396250	604874	1782727	5061514
7	10510	88722	320643	781750	5190786	7390000
8	14565	122421	387741	1049766	3904695	13200239
9	12516	119178	456756	1251088	5216579	17100000
10	22525	155718	610075	1882275	12087005	26316221

表 2 クラスタ入札に基づく解のサイズ
Table 2 Solution size based on cluster bid

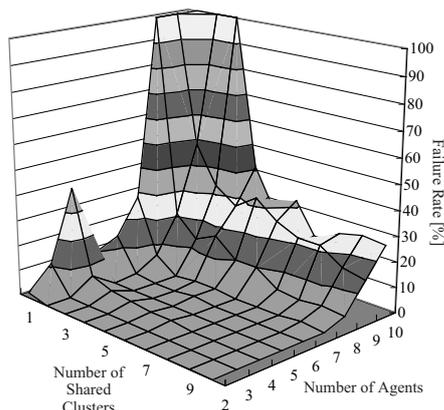


図 10 合意形成の失敗率 (制約クラスターサイズ: 小・最大制約入札無し)
Fig. 10 Failure Rates for the Medium Cluster Case without Extended Bid

数が共に 10 である場合でも、本来入札対象とする領域サイズの 5%まで、空間が縮小されている。

次に、最大制約入札に関する、合意形成の失敗率への影響を示す。図 10 は、図 7 と同様の設定に関して、最大制約入札を行わなかった場合の失敗率を示している。図に示す通り、失敗率は明らかに悪化しており、最大制約入札が、合意形成の失敗率を抑える点で効果的であることが確認できた。

本論文で示した実験では、個々のエージェントの効用関数を完全にランダムに生成するのではなく、4.1 で述べたように、共通して興味を持つ領域 (共有クラスター) を設定している。例えば、家族で購入する車を決定するような問題において、家族全員が乗れるだけの乗車定員が必要、といった共通認識が存在するならば、家族が共通して興味を持つ問題空間 (車種) を持ち得る。そのため、共有クラスターを用いた本実験の設定は現実的にも存在し得るケースを扱った妥当なものであると言える。一方、共有クラスターを用いる事なく、エージェント間の効用関数の相関を明示的に

導入せず、完全にランダムに効用関数を生成する場合は、合意形成自体が困難である事から、提案手法においても解の質の低下が予想される。このような合意形成困難な問題においても、より良い解を得るための手法の開発は今後の課題である。

5. おわりに

本論文では、相互に依存する複数の論点を扱うマルチエージェント交渉問題のために、交渉過程で注目する領域を反復的に絞り込むに手法を用いた、入札に基づく新規の交渉プロトコルを提案した。提案プロトコルでは複数のラウンドを経て解を得るが、交渉に参加するエージェントは、詳細な選好情報を過剰に公開することなく、自分にとって、より効用の高い合意の形成を試みる事が可能である。既存の研究では、エージェント数が 2 以上の規模の問題が扱われていない一方、本論文では、実験による評価から、エージェント数が最大 10 までの規模の問題に対して、提案プロトコルによって最適解に近似した解が得られることを示した。今後の課題としては、まず、入札生成 / 合意決定アルゴリズムの高速化による、スケーラビリティのさらなる改善が挙げられる。スケーラビリティの改善に関しては、例えば、10 個以上の論点を含む、より大規模な問題に対して、プロトコルをさらに多段階化した場合の有効性等について検証を行う必要がある。また、本論文では、エージェントが虚偽の入札を生成しない場合については考慮していない。そこで、入札生成と合意の決定を反復する過程において、虚偽の入札が生成される事を防ぐメカニズムについて考察する必要がある。

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] S. Fatima, M. Wooldridge and N. R. Jennings: "Optimal negotiation of multiple issues in incomplete information settings", *Proc. of AAMAS-2004*, pp. 1080-1087 (2004).
- [3] R. Y. K. Lau: "Towards genetically optimised multi-agent multi-issue negotiations", *Proc. of HICSS-2005* (2005).
- [4] L.-K. Soh and X. Li: "Adaptive, confidence-based multiagent negotiation strategy", *Proc. of AAMAS-2004*, pp. 1048-1055 (2004).
- [5] M. Klein, P. Faratin, H. Sayama and Y. Bar-Yam: "Negotiating complex contracts", *Group Decision*

- and Negotiation, **12**, 2, pp. 58–73 (2003).
- [6] 服部, 伊藤, M. Klein: “非線形効用関数を持つエージェントのためのオークションに基づく交渉プロトコル”, 電子情報通信学会論文誌, **J89-D**, 12, pp. 2648–2660 (2006).
- [7] T. Ito, M. Klein and H. Hattori: “Multi-issue negotiation protocol for agents: Exploring nonlinear utility spaces”, Proc. of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-2007) (2007).
- [8] X. Luo, N. R. Jennings, N. Shadbolt, H. fung Lung and J. H. man Lee: “A fuzzy constraint based model for bilateral, multi-issue negotiations in semi-competitive environments”, Artificial Intelligence, **148**, pp. 53–102 (2003).
- [9] M. Barbuceanu and W.-K. Lo: “Multi-attribute utility theoretic negotiation for electronic commerce”, Proc. of AMEC-2000, pp. 15–30 (2000).
- [10] A. Chavez and P. Maes: “Kasbah: An agent marketplace for buying and selling goods”, Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM-96), pp. 75–90 (1996).
- [11] T. Ito and T. Shintani: “Implementing an agent negotiation protocol based on persuasion: An approach to implement a group decision support system based on multi-agent negotiation”, Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97), pp. 592–597 (1997).
- [12] S. Saha and S. Sen: “Negotiating efficient outcomes over multiple issues”, Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS2006) (2006).
- [13] S. J. Russell and P. Norvig: “Artificial Intelligence: A Modern Approach”, Prentice Hall (2002).
- [14] H. Raiffa: “The Art and Science of Negotiation”, Belknap Press (1982).
- [15] R. T. Ng and J. Han: “Clarans: A method for clustering objects for spatial data mining”, IEEE Transactions on Knowledge and Data Engineering, **14**, 5, pp. 1003–1016 (2002).

(平成 xx 年 xx 月 xx 日受付)

服部 宏充 (正員)

2004 年名古屋工業大学大学院工学研究科電気情報工学専攻博士後期課程修了。博士(工学)。2004 年から日本学術振興会特別研究員(PD)。2004 年から 2005 年にかけて英国リバプール大学計算機科学科客員研究員。2006 年米国マサチューセッツ工科大学スローン経営大学院客員研究員。2007 年より京都大学大学院情報学研究科社会情報学専攻助教。現在に至る。2003

年度情報処理学会全国大会大会奨励賞受賞。マルチエージェントシステム, 電子商取引支援, 意思決定支援, 特にマルチエージェントの協調・交渉メカニズムに興味を持つ。AAAI, 電子情報通信学会, 人工知能学会, 情報処理学会, 日本ソフトウェア科学会各会員。

伊藤 孝行 (正員)

平成 12 年名古屋工業大学大学院工学研究科博士後期課程修了。博士(工学)。平成 11 年から平成 13 年にかけて日本学術振興会特別研究員(DC2 及び PD)。平成 12 年から平成 13 年にかけて南カリフォルニア大学 Information Sciences Institute (USC/ISI) 客員研究員。平成 13 年北陸先端科学技術大学院大学知識科学教育研究センター助教授。平成 15 年より名古屋工業大学大学院情報工学専攻助教授。平成 17 年から平成 18 年にかけて米国ハーバード大学 Division of Engineering and Applied Science 客員研究員及び, 米国マサチューセッツ工科大学 Sloan School of Management 客員研究員。平成 18 年より名古屋工業大学大学院産業戦略工学専攻助教授。現在に至る。2007 年長尾真記念特別賞および文部科学大臣表彰若手科学者賞受賞。2006 年 International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS2006) 最優秀論文賞受賞。2005 年日本ソフトウェア学会論文賞受賞。平成 16 年度 IPA 未踏ソフトウェア創造事業スーパークリエイター認定。マルチエージェントシステム, 計算論的メカニズムデザイン, オークション理論, 電子商取引支援に興味を持つ。AAAI, ACM, 電子情報通信学会, 情報処理学会, 日本ソフトウェア科学会, 人工知能学会, 計測制御自動学会各会員

Mark Klein

Dr. Mark Klein is a Principal Research Scientist at the MIT Center for Collective Intelligence, an Affiliate at the MIT Computer Science and AI Lab, affiliate faculty at the New England Complex Systems Institute, and co-director of the Robust Open Multi-Agent Systems (ROMA) research group. His research focuses on coordination science, a discipline that attempts to understand the cross-cutting fundamentals of coordination and apply it to help create better human organizations and software systems. He has made contributions in the areas of computer-supported conflict management for collaborative design, design rationale capture, business process re-design, exception handling in workflow and multi-agent software systems, service discovery, negotiation algorithms, understanding and resolving 'emergent' dysfunctions in distributed systems and, more recently, 'collective intelligence' systems to help people collaboratively solve complex problems like global warming.

Abstract Multi-issue negotiations are a central part of many coordination challenges, and thus represent an important research topic. Almost all previous work in this area has assumed that negotiation issues are independent, but this is rarely the case in real-world contexts. Our work focuses on negotiation with interdependent issues and, therefore, nonlinear (multi-optimum) agent utility functions. Since the utility functions are typically very complex, the challenge becomes finding high-quality negotiation outcomes without making unrealistic demands concerning how much agents reveal about their utilities. Since negotiations often involve more than two parties, the approach should also be scalable. In this paper, we propose a novel protocol for addressing these challenges, wherein agents approach agreements by iteratively narrowing the space of possible agreements. In the early stages, agents submit rough bids representing promising regions from their utility functions. In later stages, they submit increasingly narrow bids for the subset of those regions that the negotiating parties all liked. We show that our method outperforms existing methods in large nonlinear utility spaces, and is computationally feasible for negotiations with as many as ten agents.

Key words Multi-issue negotiation, Nonlinear utility, Multi-agent Systems, Bidding