

ID	論点 1	論点 2	論点 3	論点 4	効用
1	[2, 4]	∅	[4, 6]	∅	20
2	∅	5	[3, 7]	[1, 6]	40
3	[3, 8]	∅	∅	∅	25
4	4	[2, 7]	9	[4, 5]	50

表1 エージェントがもつ制約情報

のように表現できる。 Ag はエージェントの集合を表し、 $|Ag| = N$ である。

$$\operatorname{argmax}_{\bar{s}} \sum_{n \in Ag} u_n(\bar{s})$$

言い換えると、提案プロトコルは社会的効用、すなわち全ての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。

本論文では全エージェントがすべての効用情報を公開して、非線形最適化手法を利用する手法を採用しない。なぜならば、交渉の場での効用に関する情報の公開は、実世界のユーザにとって好ましくないからである。具体的な例として、車など商品の売買に対する交渉が挙げられる。売買交渉のようにプレイヤー間で、得られる合意案に関する効用が相反する場合、売り手の最低販売価格が明かされると、買い手は当然その価格での購入を望む。逆に、買い手の最高購入価格が明かされると、売り手はその価格での売却を希望する。以上のように交渉において他のエージェントに自身の効用情報が漏洩することは、他のエージェントの効用を減少させてでも自分の効用を増加させるような利己的な戦略がとれるようになり、結果としてプライバシー情報が公開されたエージェントの得られる効用が減少する可能性がある。

3. 交渉における論点の相互依存関係

3.1 論点の相互依存関係

本論文では、もし、論点 X と論点 Y に関する制約が存在するならば、論点 X と論点 Y は相互依存関係であると考えられる。例えば、あるエージェントが表1のような制約情報を保持する場合を考える。表1の制約1では、論点1が2~4、論点3が4~6の範囲内に存在する場合に効用値20を得る。エージェントは制約1を持っているため、論点1と論点3に関する二項制約を持っている。以上の場合、エージェントにとって論点1と論点3は相互依存関係となる。

次に、エージェントがもつ論点の相互依存関係の強さを示す指標として相互依存度 (interdependency rate) を定義する。本論文では論点 X と論点 Y に関する相互依存度をエージェントがもつ論点 X と論点 Y に関係する制約数と考える。以下に、エージェント n の論点 j と論点 jj の相互依存度 ($D_n(i_j, i_{jj})$) を示す。

$$D_n(i_j, i_{jj}) = \#\{c_k | \delta(c_k, i_j) \neq \emptyset \wedge \delta(c_k, i_{jj}) \neq \emptyset\}$$

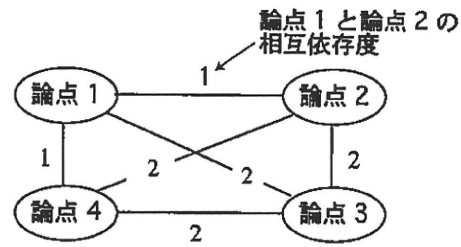


図2 相互依存関係グラフ

エージェントがある論点同士に対して、多数の相互依存関係にある制約つまり効用関数をもつとき相互依存度が強くなり、逆にある論点間に対して少数の相互依存関係にある制約をもつとき相互依存度が弱くなる。例えば、表1において制約2と制約4が論点3と論点4に対して相互依存関係なので論点3と論点4の相互依存度は2となる。

また、エージェントは論点間の相互依存関係を相互依存関係グラフ (interdependency graph) により保持している。以下に相互依存関係グラフの定義を示す。

$$G(P, E, w) : P = \{1, 2, \dots, |I|\} (\text{有限集合}),$$

$$E \subset \{\{x, y\} | x, y \in P\}, w : E \rightarrow R$$

相互依存関係グラフはノードを各論点、エッジを相互依存関係の有無、重みを相互依存関係度とした、重み付き無向グラフである。図2は相互依存関係グラフの例を示している。図2が示すように、頂点が各論点、エッジが相互依存関係の有無、各エッジの重みは相互依存度を示している。また、エッジの有無で依存関係が存在するかを判断する。相互依存関係グラフを用いることで、エージェントは各論点間の相互依存関係を容易に管理することができる。

3.2 エージェントによる相互依存関係の抽出

エージェントが相互依存関係グラフを作成するために、各論点の相互依存関係抽出機構が必要である。本論文では、エージェントがもつすべての制約を調査し、各論点の相互依存関係を調査する手法を採用する。本手法を用いた場合、すべての依存関係が調査できるため誤りが少ない相互依存関係グラフを作成できる。また、計算時間は制約数に対して線形時間となる。Algorithm1にアルゴリズムの詳細を示す。

4. 論点間の依存関係に基づく合意形成モデル

本章では、相互依存関係グラフに基づいて論点グループごと交渉論点を分割し、入札に基づく交渉手法 [Ito 07] を基に合意形成を行なう手法を提案する。本手法は分割しない場合と比較して、高いスケーラビリティが期待できる。以下に論点グループに基づく交渉手法の詳細を示す。

Algorithm 1 get_Dependency(C) C : a set of constraint

```

1: for  $c \in C$  do
2:   for  $i := 0$  to Number of issues do
3:     for  $j := i + 1$  to Number of issues do
4:       if Issue  $i$  and Issue  $j$  are interdependent in  $c$ 
         then
5:         interdependencyGraph[ $i$ ][ $j$ ] $++$ 
6:       end if
7:     end for
8:   end for
9: end for

```

[Step1: 相互依存関係の抽出]

エージェントは自身の効用情報を探索し、相互依存関係グラフを作成する。各論点間の相互依存関係の有無や相互依存関係度の抽出は 3.2 節に示したように、エージェントがもつすべての制約を調査し、正確な相互依存関係グラフを作成する。その後、エージェントはメディアータに相互依存関係グラフを公開する。

[Step2: 論点グループの決定]

本ステップではメディアータがエージェントからの相互依存関係グラフから、論点間における相互依存関係の損失が少ない最適な論点グループを決定する。本手法では、論点グループ数はあらかじめ決められているものとする。まず、メディアータは社会的相互依存関係グラフを作成する。社会的相互依存関係グラフとはノードを各論点、エッジを相互依存関係の有無とした無向重み付きグラフである。また、社会的相互依存関係グラフにおける i_j と i_{jj} 間のエッジの重みは以下の式で定義する: $\sum_{n \in N} D_n(i_j, i_{jj})$ ($D_n(i_j, i_{jj})$ エージェント n における論点 i_j と論点 i_{jj} 間の相互依存度)。

論点グループ数は少ないほど、最終的に見つかる合意案の最適率が高いが 1 グループが考慮する論点数が増えるため、交渉問題に対するスケーラビリティが増加する。一方、論点グループ数が多いほど、最終的に見つかる最適率が下がるが、1 グループが考慮する論点数が減るため、交渉問題に対するスケーラビリティが減少する。

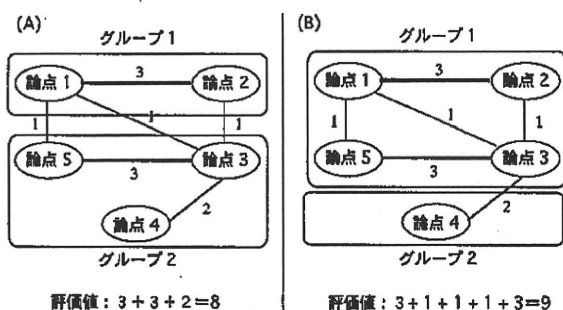


図3 グループ分割における評価値

次に、社会的相互依存関係グラフを用いて、交渉問題を論点グループを決定する。本論文では、グループ数をあらかじめ決めておき、シミュレーテッドアニーリング (SA)[Russell 02] で評価値を最大化する手法を採用する。SA で採用する評価値はグループに分割した場合に存在するエッジの重みの和である。図3は二つのグループに分割する際に用いる評価値の具体例を示している。(A)のようにグループ分けした場合の評価値は、グループ1には論点1と論点2間におけるエッジの評価値3、グループ2には論点3、論点4と論点5間におけるエッジの評価値3および2の和の8となる。また、(B)のようにグループ分けした場合、評価値は論点1、論点2、論点3および論点5間のエッジの重みの和の9となる。

エージェントにとって自身の相互依存度が高い論点同士を分割することは、最適な合意案ではない低い効用値の合意案が生成されるリスクが存在する。つまり、全ての論点を考慮すれば効用が高く最適な合意案だとしても、グループに分割して相互依存関係を無視すると、効用が低い合意案が生成される可能性がある。したがって、論点グループを決定する際には各論点の相互依存関係を反映したグループを決定する必要がある。

[Step3: 入札の生成]

まず、エージェントは全ての論点を対象にして入札を生成する。具体的には、まず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サンプリングポイントを取得する。次に、効用空間のサンプリングの調整を行う。各エージェントは SA に基づく最適化を行い、サンプリングポイント周辺の合意案候補の発見を試みる。最後に、入札の生成を行う。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 z に充足される全ての制約の共通範囲を取ること容易に生成できる。

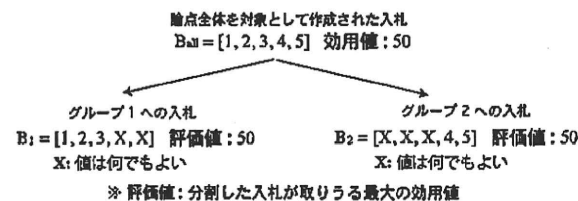


図4 エージェントによる入札の分割

次に、全ての論点を対象にして生成した入札をグループ別に分割し、分割した入札に対して評価値を決定する。評価値は、分割した入札が取りうる最大の効用値とする。

例えば図4が示すようにすべての論点数が5であり、全論点対象に生成した入札 ($B_{all} = [1, 2, 3, 4, 5]$ でそのときの効用値 50) を論点 1-3 が対象のグループ 1 の入札 (B_1) と論点 4,5 が対象のグループ 2 の入札 (B_2) の 2 つに分割する場合を考える。以上の設定のとき、 $B_1 = [1, 2, 3, X, X]$ 、 $B_2 = [X, X, X, 4, 5]$ (X : 値は何でもよいことを示す) と分割される。また、 B_1, B_2 とも評価値として 50 が設定される。全範囲の入札を行なう際に SA によるサンプリングポイントの最適化を行なうため、 B_1, B_2 とも分割した入札の取りうる最大値は $B_{all} = [1, 2, 3, 4, 5]$ となることが保証される。

本論文では、分割された入札における評価値として、分割された入札が取りうる最大値としている。本論文の評価値の設定により存在する合意案候補のうち最適な合意案を発見することが可能となる。

[Step4: 合意案の発見]

各エージェントの全ての入札の組み合わせを考慮し、グループ内で関係する論点間の合意案を決定する。ここでは、各エージェントにつき1つの入札を採用し、グループに属する論点間で互いに無矛盾で、最も評価値の高い入札の組み合わせを決定する。具体的には、まず、入札が持つグループ内の論点に関する値の範囲の共通部分が存在するかどうか判定する。その後、共通部分の中で最も評価値が高いものを、合意案として選択する。各入札がもつ社会的効用を最大化する入札の組み合わせを見つけるために、枝刈り付きの幅優先探索を行う。

そして、各グループから生成された合意案を組み合わせ、最終的な合意案を作成する。もし、グループ外の論点に関して相互依存関係が全くなければ、最終的に組み合わせた最終的な合意案は、各グループで作成された合意案の評価値と一致するため、最適な合意案を発見できる。

5. 評価実験

5.1 実験設定

本実験では、エージェント間の交渉を 100 回試行し平均値を取る。効用空間作成のパラメータは以下の通りである。

- エージェント数：6
- 論点数：3-15
- 論点の値域：[0, 9]
- 制約数（非線形効用空間）：10（単項制約）、5（二項制約）、5（三項制約）。
- 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であると考えられる。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])。

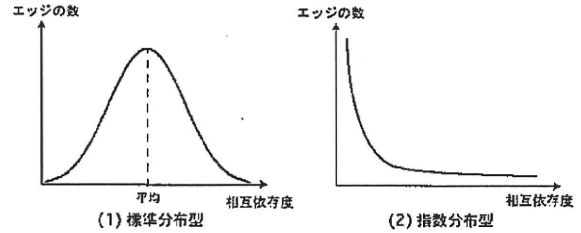


図5 各論点間の相互依存度の分布

以上のパラメータ条件において、各論点間の相互依存関係以外はランダムに決定される。各論点間の相互依存関係に関しては、(1)ランダムに決定（標準分布型）(2)指数乱数に基づいて決定（指数分布型）の2パターンに関して実験する。(1)標準分布型は、効用関数がどの論点に対して相互依存関係になるかは一様分布に基づく乱数によって決定される。一方、(2)指数分布型は、効用関数がどの論点に対して相互依存関係になるかは指数分布に基づく乱数によって決定される。図5は本実験で使用する、標準分布型および指数分布型の相互依存度とエッジ数の分布を示している。(1)標準分布型においては平均となる相互依存度のエッジ数が最も多く、平均から離れるにつれてエッジ数が少なくなっていく。一方、指数分布型では相互依存度が低いエッジ数が多く、相互依存度が多いエッジ数が少なくなっている。(1)は[Ito07]をはじめとした、交渉プロトコルの一般的な評価の際に用いられており、小規模の交渉問題を想定している。一方、(2)は論点数が膨大である大規模な交渉問題を想定している。

本実験では主に3つの手法の比較を行う。“(A) 論点グループ”は本論文で提案した論点グループに基づく交渉手法である。“(B) Basic Bidding”は[Ito07]で提案されている入札に基づく交渉手法であり、本論文で提案した論点グループを生成しない手法である。“(C) Q-Factor”は[Marsa-Maestre 09a, Marsa-Maestre 09b]において提案されている Maximum Weight Interdependent Set(MWIS)に基づく手法である。MWISは(B)と比較して、高いスケーラビリティを示すことが実験的に分かっている。

以下は、(A)-(C)で共通して用いる、入札の生成に関するパラメータ [Ito 07] である。

- 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 SA では初期の温度を 30 とし、処理を 30 回繰り返す。
- 入札生成時の閾値：100
- 1 エージェント当りの入札数の上限： $\sqrt[3]{6400000}$ (N:

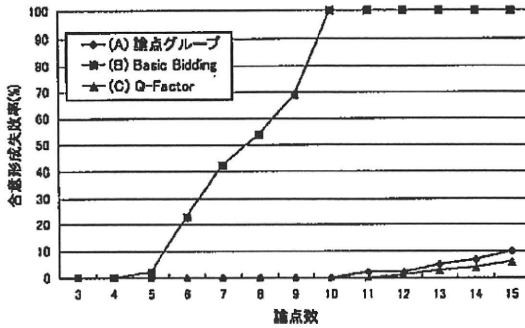


図6 合意形成失敗率の比較 (標準分布型)

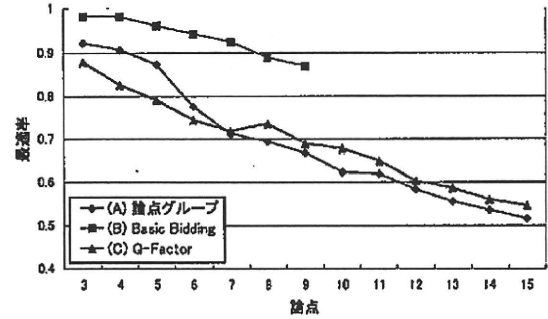


図7 最適率の比較 (標準分布型)

エージェント数). 本設定により, 勝者決定において, 6,400,000 以上の入札の組合せは計算されないため, 実用的な計算時間内で合意案が得られる.

(A) で用いる論点グループに基づく交渉プロトコルにおいて, グループ分割の際に用いる SA では初期の温度を 30 とし, 処理を 10 回繰り返す. 本設定は, グループ分割における計算コストは高々論点数 C グループ数である. 以上から, 繰り返し回数が少なくなるように抑えている. また, 論点グループ数は 3 に設定した. (C) で用いる Q-Factor の式は $Q = u^\alpha * v^\beta$ (u : 効用値, v : 入札がもつ範囲) を採用し, $\alpha = 0.5, \beta = 0.5$ とする.

最適率を求める実験において, 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, 全エージェントの効用関数をすべてメディアータに公開し, SA を用いた探索により, 発見した解を近似最適解とする. 近似最適解を求めるための SA は初期の温度を 50 度として, 500 回の繰り返し処理を経て温度を 0 まで下げることとし初期解はランダムに選択する. SA に基づく手法は, エージェントが効用に関する完全な情報をメディアータに公開し, 集中処理による求解を行うことになる. 以上から, プライバシー情報の公開について全く考慮しておらず, 実際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非現実的な設定である. 本実験では, 提案プロトコルの性能評価の際に, 厳密解に近似した解が必要であるために, 比較対象としてのみ本手法を採用する. 本実験では (最適率) = (各手法で求めた解の効用値) / (近似最適解の効用値) を使用する. また, 最適率を求める際は合意形成失敗の交渉は結果に考慮せず, すべて合意形成に成功した場合のみ考えることとする.

実験は JAVA2(1.5) で記述し, Mac OS 10.5 が動作している iMac (Core2Duo 2.33GHz メモリ 1.5GB) で行った.

5.2 実験結果

図 6 は標準分布型の合意形成失敗率の比較を示している. (A) と (B) を比較すると, (A) の方が低い合意形成失敗率になっており, 合意形成の失敗を回避出来ているこ

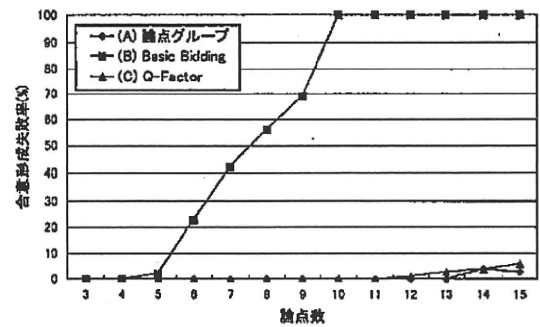


図8 合意形成失敗率の比較 (指数分布型)

とがわかる. (A) が合意形成失敗の回避に有効な理由は, 入札の組み合わせ最適解を求める際の計算コストが削減されているためである. また, (A) と (C) を比較した場合, 合意形成失敗率に大きな差は見られなかった. (A), (C) とともに, グループ数や Q-Factor などの値の調整により合意形成失敗率を大きく改善することが可能である.

図 7 は標準分布型の最適率の比較を示している. (A) と (B) を比較した場合, (A) が (B) より少し低い値になっている. (A) の最適率が (B) より少し低い理由は, グループ分割により存在する論点間の相互依存関係を無視するからである. しかし, 無視する論点間の依存関係が少なくなるように最適化を行っているため, 大きく最適率が減少することを防ぐことができる. (A) と (C) を比較した場合, 同様の最適率になっている. 同様の最適率になる理由は, (A), (C) とともに合意案の最適率を減少をできる限り防ぎながら, スケーラビリティを向上させる手法だからである.

図 8 と図 9 は指数分布型における合意形成失敗率を示している. 標準分布型と比較して, (A)-(C) の合意形成失敗率に大きな変化はなかった. 一方, (A) の最適率が標準分布型と比較して改善している. (A) の最適率が改善している理由として, 論点グループに基づく手法は, 論点の相互依存関係を無視する場合が少ないほど, 最適率が高くなり, 指数分布型では標準分布型より相互依存関

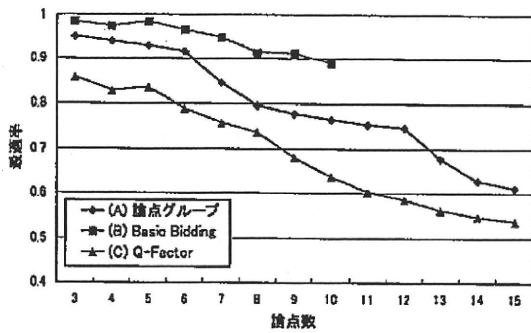


図9 最適率の比較 (指数分布型)

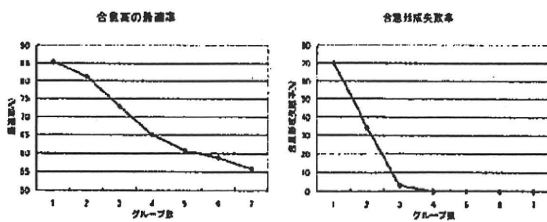


図10 グループ数の影響

係を無視する可能性が低くなるからである。一方、(B)や(C)は標準分布型と比較して大きな差はない。以上から、(B)や(C)は論点間の相互依存関係数の変化が、結果に大きく影響を与えないことが分かる。

図10は論点グループに基づく交渉手法においてグループ数が最適率、合意形成失敗率に与える影響を示している。本実験では、論点数：7、エージェント数：4として、効用空間の作成方法を標準分布型とした。論点グループに基づく交渉手法において、グループ数が増加するごとに最適率が減少している。最適率減少の理由は、グループ数が増えると論点間の相互依存関係を無視する確率が増加するため、最適解を保証できる場合が減少するからである。一方、グループ数が増加するごとに合意形成失敗率が減少している。合意形成失敗率減少の理由はグループ数が増加すると、1グループが扱う論点数が減少するため、計算コストを大幅に削減できるからである。以上のように本論文で提案している論点グループに基づく交渉手法は、グループ数を適切に設定して、最適率、合意形成失敗率間のトレードオフを調整する必要がある。

6. 関連研究

本研究の先行研究として効用空間の絞り込みに基づく交渉手法 [服部 07] と閾値調整交渉メカニズム [藤田 08] が提案されている。効用空間の絞り込みに基づく交渉手法 [服部 07] において、メディエータは、クラスター入札、最大制約入札そしてピーク入札を行なうことで、ラ

ウンドごとに合意形成を探索する範囲を絞りこみながら交渉を行っていく。本文で提案されているメカニズムが論点数10のような大規模な交渉問題においても実行可能であり高い最適率と低い合意形成失敗率であることが示されている。しかし、評価実験において共有クラスタと呼ばれるすべてのエージェントに共有されている制約を定義した環境下で実験を行っているため本論文の同様の実験環境ではない。[服部 07] の実験環境と比較して、本論文の実験設定の方がランダム性が低く、合意案を発見しにくい環境である。さらに、論点数10の場合の実験しか行われていない。以上から論点数15の段階で合意形成失敗率がほぼ0である提案手法の方が、スケーラビリティが高いと考えられる。また、閾値調整メカニズム [藤田 08] は入札に関する閾値を調整しながら交渉を行なうことで、メディエータに公開する範囲を最低限にすることに成功している。しかし、論文において合意形成失敗率つまりスケーラビリティは [服部 06, Ito 07] と同様であることが示されている。また、手法の目的がスケーラビリティの向上ではなく、公開するプライバシー情報を最低限にすることであり、本論文の目的と異なる。

多くの既存研究は線形の効用関数のみに対応したものの [Bosse 05, Faratin 02, Fatima 04] だが、複数論点交渉問題における非線形効用関数に注目した文献も増え始めている。文献 [Lin 03] は、遺伝的アルゴリズムに基づく双方向型の交渉プロトコルを提案している。ここでは、エージェントが、他のエージェントの提案を取り入れながら自身の選好を更新し、高効用の解の決定を試みる。しかし、本文では、売り手と買い手のような、立場の異なる2種類のエージェント間の交渉問題が想定されているが、エージェントの選好に、より多くのバリエーションが存在するケースに関する議論も行われておらず、大規模な問題に関する性能が不明である。文献 [Klein 03] では、二項制約を含む中規模の双方向型の交渉問題を対象とし、SAに基づくメディエータが準最適な解を得る手法が提案されている。一方、本論文では、さらに高次の依存関係と多数のエージェントが存在するより複雑な交渉問題に関して、高い合意案の最適性を議論している。文献 [Lai 06] は二者間の複数論点の交渉問題を取り扱っており、提案した手法がパレート最適性を満たす解を探ることが可能であることが示されている。また、文献 [Robu 05] では電子商取引における二者間の複数の商品取引や複数論点の交渉問題に対して Utility Graph を用いて取り組んでいる。さらに、文献 [Fatima 07] は時間制約が存在する二者間交渉を提案している。これらの研究は興味深い観点から様々な手法が提案されているが、どれも二者間に限った設定である。文献 [Shew 08] はメディエータを導入し交渉を行うフレームワークを提案している。さらに、agenda(議題)別に交渉を行っていく手法も導入されている。

また、分散制約最適化問題 (DCOP) において、セキュリ

ティを考慮した数々の論文が存在する [Maheswaran 08]. しかし、多くの分散制約充足問題における設定はエージェントが必ず協力することを前提としており、エージェントの個人合理性が十分に仮定されていない。組み合わせオークションにおいても、セキュリティを考慮した手法 [P. Cramton 06] が存在する。しかし、交渉において“全て”のエージェントが勝者になるが、オークションでは少数のエージェントが勝者になる前提がある。以上から本論文の問題に対して単純に、組み合わせオークションや DCOP の手法を導入するのは困難である。

7. おわりに

本論文では、論点グループに基づく合意形成手法を提案した。論点グループに基づく交渉手法では、まず、エージェントが作成した相互依存関係グラフに基づいて、存在する相互依存度が最大になるように論点グループ構成を決定する。そして、論点グループごとにエージェントは入札を生成し、組み合わせ最適解を求める。最終的にグループごとに生成された合意案を組み合わせ最終合意案を発見する。シミュレーション実験を用いて、既存の手法と比較して低い合意形成失敗率であることを示した。また、論点グループ数や相互依存関係グラフの性質が合意案の最適率や合意形成失敗率に与える影響を解析した。

今後の課題として、分割した入札に対する評価値の決定方法に関して、エージェントの戦略を考慮することが考えられる。本論文では、分割した入札がとりうる最大の効用値を評価値と設定したが、戦略によって評価値を設定する方が現実的である。また、本論文では基数的効用に基づいているが序数的効用の方が現実的な場合も存在する。今後、本論文で扱っている効用関数や交渉手法が序数的効用でも適用できるか解析する必要がある。

謝辞

本研究は JST 戦略的創造研究推進事業（さきがけ）の「知の創成と情報社会」研究領域により支援を受けている。

◇ 参考文献 ◇

- [Bosse 05] Bosse, T. and Jonker, C. M.: Human vs. Computer Behaviour in Multi-Issue Negotiation, in *Proc. of 1st International Workshop on Rational, Robust, and Secure Negotiations in Multi-Agent Systems (RRS-2005)*, pp. 11–24 (2005)
- [Faratin 02] Faratin, P., Sierra, C., and Jennings, N. R.: Using Similarity Criteria to Make Issue Trade-offs in Automated Negotiations, in *Artificial Intelligence*, pp. 142:205–237 (2002)
- [Fatima 04] Fatima, S. S., Wooldridge, M., and Jennings, N. R.: Optimal Negotiation of Multiple Issues in Incomplete Information Settings, in *Proc. of Third International Joint Conference on Autonomous Agent and Multi-Agent Systems (AAMAS-2004)*, pp. 1080–1087 (2004)
- [Fatima 07] Fatima, S. S., Wooldridge, M., and Jennings, N. R.: Approximate and online multi-issue negotiation, in *Proc. of 6th International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2007)*, pp. 947–954 (2007)
- [Fujita 08] Fujita, K., Ito, T., and Klein, M.: A representative-based multi-round protocol for multi-issue negotiations, in *Proc. of 7th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-2008)* (2008)
- [Ito 07] Ito, T., Hattori, H., and Klein, M.: Multi-issue Negotiation Protocol for Agents: Exploring Nonlinear Utility Spaces, in *Proc. of 20th International Joint Conference on Artificial Intelligence (IJCAI-2007)*, pp. 1347–1352 (2007)
- [Klein 03] Klein, M., Faratin, P., Sayama, H., and Bar-Yam, Y.: Negotiating Complex Contracts, *Group Decision and Negotiation*, Vol. 12, No. 2, pp. 58–73 (2003)
- [Lai 06] Lai, G., Li, C., and Sycara, K.: A general model for pareto optimal multi-attribute negotiations, in *Proc. of The 2nd International Workshop on Rational, Robust, and Secure Negotiations in Multi-Agent Systems (RRS-2006)* (2006)
- [Lin 03] Lin, R. J. and Chou, S. T.: Bilateral Multi-Issue Negotiations in a Dynamic Environment, in *Proc. of the AAMAS Workshop on Agent Mediated Electronic Commerce (AMEC-2003)* (2003)
- [Maheswaran 08] Maheswaran, R. T., Pearce, J. P., Varakantham, P., Bowring, E., and Tambe, M.: Privacy Loss in Distributed Constraint Reasoning: A Quantitative Framework for Analysis and its Applications, *Autonomous Agents and Multi-Agent Systems*, Vol. 13, pp. 27–60 (2008)
- [Marsa-Maestre 09a] Marsa-Maestre, I., Lopez-Carmona, M. A., Velasco, J. R., Ito, T., Fujita, K., and Klein, M.: Balancing Utility and Deal Probability for Negotiations in Highly Nonlinear Utility Spaces, in *Proc. of Twenty-first International Joint Conference on Artificial Intelligence (IJCAI-09)* (2009)
- [Marsa-Maestre 09b] Marsa-Maestre, I., Juan R. Velasco, and M. A. L.-C., and Hoz, de la E.: Effective bidding and deal identification for negotiations in highly nonlinear scenarios, in *Proc. of The 8th International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2009)*, pp. 1057–1064 (2009)
- [P. Cramton 06] P. Cramton, Y. S. and Steinberg, R.: *Combinational Auctions*, MIT Press (2006)
- [Robu 05] Robu, V., Somefun, D. J. A., and Poutre, J. L.: Modeling complex multi-issue negotiations using utility graphs, in *Proc. of the 4th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2005)* (2005)
- [Russell 02] Russell, S. J. and Norvig, P.: *Artificial Intelligence: A Modern Approach*, Prentice Hall (2002)
- [Shew 08] Shew, J. and Larson, K.: The Blind Leading the Blind: A Third-Party Model for Bilateral Multi-issue Negotiations under Incomplete Information, in *Proc. of The 1st International Workshop on Agent-based Complex Automated Negotiations (ACAN-2008)* (2008)
- [藤田 08] 藤田 桂英, 伊藤 孝行, 服部 宏充: 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, コンピュータソフトウェア (日本ソフトウェア科学会論文誌)「ソフトウェアエージェントとその応用特集号」, Vol. 25, No. 4, pp. 167–180 (2008)
- [服部 06] 服部 宏充, 伊藤 孝行, Klein, M.: 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 D-I, Vol. J89-D, No. 12, pp. 2648–2660 (2006)
- [服部 07] 服部 宏充, 伊藤 孝行, Klein, M.: 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, 電子情報通信学会論文誌 D-I,「ソフトウェアエージェントとその応用特集号」, Vol. J90-D, No. 9, pp. 2336–2348 (2007)

〔担当委員：大須賀 昭彦〕

2009年12月18日 受理

著者紹介



藤田 桂英(学生会員)

2008 年名古屋工業大学情報工学科卒業。2010 年同大学院産業戦略工学専攻博士前期課程修了。同年同大学院情報工学専攻博士後期課程在学中。2010 年 6 月より米国マサチューセッツ工科大学 Sloan School of Management, Visiting Student。現在に至る。マルチエージェントシステムに興味を持つ。2009 年第 24 回電気通信普及財団賞(テレコムシステム技術学生賞)受賞他。IEEE, 情報処理学会, 電子情報通信学会各学生会員。



伊藤 孝行(正会員)

2000 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)。1999 年から 2001 年にかけて日本学術振興会特別研究員(DC2 及び PD)。2000 年から 2001 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。2001 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。2003 年より名古屋工業大学大学院情報工学専攻助教授。2005 年から 2006 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。2006 年より名古屋工業大学大学院産業戦略工学専攻准教授。2008 年より米国マサチューセッツ工科大学 Sloan School of Management 客員研究員。2009 年より科学技術振興機構(JST) さきがけ大規模型研究員。2010 年より東京大学政策ビジョン研究センター客員研究員。現在に至る。2007 年文部科学大臣表彰若手科学者賞受賞。情報処理学会長尾真記念特別賞受賞。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科学会論文賞受賞。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。第 66 回情報処理学会全国大会優秀賞及び奨励賞受賞。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, 大学発ソフトウェアベンチャーに興味を持つ。ACM 上級会員, AAAI, 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, 計測制御自動学会, 日本経済学会各会員。



Mark Klein

Dr. Mark Klein is a Principal Research Scientist at the MIT Center for Collective Intelligence, as well as an Affiliate at the MIT Computer Science and AI Lab (CSAIL) and the New England Complex Systems Institute (NECSI). His research focuses on understanding how computer technology can help groups, especially large ones, make better decisions about complex problems. 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 systems, service discovery, negotiation algorithms, 'emergent' dysfunctions in distributed systems and, more recently, 'collective intelligence' systems to help people collaboratively solve complex problems like global warming.

Representative based Multi-Round Protocol based on Revealed Private Information for Multi-issue Negotiations

Katsuhide Fujita* Takayuki Ito[†] Mark Klein[‡]

Abstract

Multi-issue negotiation protocols represent a promising field since most negotiation problems in the real world involve multiple issues. Our work focuses on negotiation with interdependent issues in which agent utility functions are nonlinear. Existing works have not yet focused on the private information of agents. In addition, they were not scalable in the sense that they have shown a high failure rate for making agreements among five or more agents. In this paper, we focus on a novel multi-round representative based protocol that utilizes the amount of revealed agents' private information. Experimental results demonstrate that our mechanism reduces the failure rate in making agreements and is scalable for the number of agents compared with existing approaches.

Keywords: Multi-issue Negotiation, Nonlinear Function

1 Introduction

We envision a future in which large numbers of participants collaborate, negotiate, and reach consensus through computer-supported negotiation support systems for global problems. Collaboratorium [1, 2], one such pioneering work that enables many people to participate in an argument on such worldwide problems as global warming, provides a platform for coordinating large-scale arguments through web based collaboration tools. In our research, we consider a tool that supports large-scale negotiations among the world's people. In such

*School of Techno-Business Administration Nagoya Institute of Technology/ Visiting Student, Center for Collective Intelligence, Sloan School of Management, Massachusetts Institute of Technology, Gokiso, Showa-ku, Nagoya, Aichi, 466-8555 JAPAN, Tel:052-735-7968, Fax:052-735-7404, E-mail:fujita@itolab.mta.nitech.ac.jp

[†]Visiting Scholar, Center for Collective Intelligence, Sloan School of Management, Massachusetts Institute of Technology/ School of Techno-Business Administration, Department of Computer Science and Engineering, Nagoya Institute of Technology/ Researcher, PREST, Japan Science and Technology Agency (JST).

[‡]Center for Collective Intelligence MIT Sloan School of Management

a situation, eliciting people's utility spaces and automatically finding and suggesting possible agreements would be valuable. People could reach agreements based on such system suggestions.

Multi-issue negotiation protocols represent an important field of study. Even though much previous work exists in this area [3, 4, 5], most of it deals exclusively with simple negotiations involving independent multiple issues. These studies of negotiations mainly assume that agents have an incentive to cooperate to achieve win-win agreements because the situation is not a zero-sum game. Many real-world negotiation problems, however, are complex and involve interdependent multiple issues. Thus, we focus on complex negotiation with interdependent multiple issues.

The bidding based negotiation protocol offers high performance on multi interdependent issues negotiation. However, it has two main issues. 1) **Privacy:** Existing works have not yet addressed agents' private information, which should not be revealed excessively because agents who reveal too much utility information suffer a disadvantage. For example, suppose that several companies are collaboratively designing and developing a new car model. If one company reveals more utility information than the other companies, those other companies can learn more of that company's utility information. As a result, the company will face a disadvantage in subsequent negotiations. Furthermore, explicitly revealing utility information is dangerous from a security standpoint. 2) **Scalability for the number of agents:** The bidding based negotiation protocol does not have high scalability for the number of agents. In the bidding based negotiation protocol, the mediator needs to find the optimum combination of submitted bids from the agents. However, the computational complexity for finding solutions is too large. The number of agent bids was limited in existing work [6]. Limiting bids causes low optimality and high failure rate for agreements.

To resolve privacy issues, we define an agent's **revealed area** that represents the amount of his/her revealed utility space. This revealed area numerically defines which agents are cooperative and which are not. Additionally, the mediator can understand how much of the agent's private information has been revealed in the negotiation.

Moreover, we propose a **representative based protocol** that has high scalability for the number of agents and considers the agent's private information. In our protocol, we first select representatives who revealed more of their utility space than the others. These representatives reached an agreement on alternatives and proposed them to the other agents. Finally, the other agents can express their own intentions concerning agreement or disagreement. In this protocol, agents who revealed more private utility information can have a greater chance to be representatives who will attend to reach an agreement on behalf of the other agents. Although agents tend to avoid revealing their own private information, they have an incentive to reveal it to be representatives.

The representative based protocol has been inspired by the parliamentary systems in England, Canada, Australia, Japan, etc. in which representatives are making an agreement on behalf of other people. In a situation in which many

people have to reach an agreement, directly reflecting all members' opinions is quite difficult. Doing so requires much time and energy and is not scalable. Although voting is one option, voting might have paradoxical results [7].

We expand our mechanism to be multi-round by using the Threshold Adjustment Protocol [8]. The multi-round mechanism improves the failure rates and achieves fairness in terms of the revealed area. This means that the amounts of the revealed areas are almost the same among agents. Further, a representative mechanism can prevent the unfair solutions that can exist in the original Threshold Adjustment Protocol.

The representative based protocol drastically reduces computational complexity because only representative agents try to reach a consensus. The experimental results demonstrate that our protocol reduces the failure rate in making agreements and that it is scalable on the number of agents compared with existing approaches. We also demonstrate that our protocol reduces the revealed area compared with existing works. Furthermore, we investigate the detailed effect of the representative selection method in our protocol and call the selection method RAS in which agents who reveal a larger utility area are selected as representatives. In the experiments, we compare RAS with a selection method in which representative agents are randomly selected (RANDOM).

The remainder of this paper is organized as follows. First, we describe a model of non-linear multi-issue negotiation and an existing work's [6] problems. Second, we define the revealed area and proposed our new negotiation mechanism. Third, we describe the multi-round negotiation protocol. Fourth, we present an experimental assessment of this protocol. Finally, we describe related work and draw conclusions.

2 Negotiation Using Complex Utility Space

2.1 Complex Utility Model

We consider the situation where n agents want to reach an agreement. m issues, $s_j \in S$ must be negotiated. The number of issues represents the number of dimensions of the utility space. For example, if there are three issues, the utility space has three dimensions. The issues are not "distributed" over agents who are all negotiating a contract that has N (e.g., 10) issues. All agents are potentially interested in the values for all N issues. Issue s_j has a value drawn from the domain of integers $[0, X]$, *i.e.*, $s_j \in [0, X]$. A discrete domain can come arbitrarily close to a real domain by increasing the domain size. As a practical matter, many real-world issues that are theoretically real (delivery date, cost) are discretized during negotiations. Our approach, furthermore, is not theoretically limited to discrete domains. The deal determination part is unaffected, although the bid generation step must be modified to use a nonlinear optimization algorithm suited to real domains. A contract is represented by a vector of issue values $\vec{s} = (s_1, \dots, s_m)$.

An agent's utility function is described in terms of constraints. There are

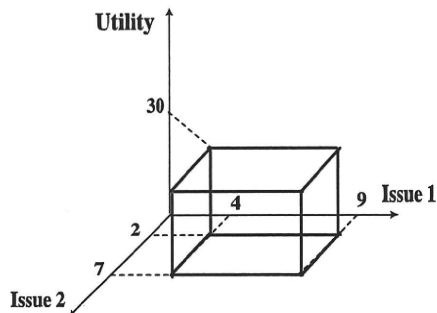


Figure 1: Example of a constraint

l constraints, $c_k \in C$. Each constraint represents a region with one or more dimensions and has an associated utility value. Constraint c_k has value $w_i(c_k, \vec{s})$ if and only if it is satisfied by contract \vec{s} . Figure 1 shows an example of a binary constraint between issues 1 and 2. This constraint has a value of 30 and holds if the value for issue 1 is in the range [4, 9] and the value for issue 2 is in the range [2, 7]. Every agent has its own, typically unique, set of constraints.

An agent's utility for contract \vec{s} is defined as $u_i(\vec{s}) = \sum_{c_k \in C, \vec{s} \in x(c_k)} w_i(c_k, \vec{s})$, where $x(c_k)$ is a set of possible contracts (solutions) of c_k . This expression produces a "bumpy" nonlinear utility space with high points where many constraints are satisfied and lower regions where few or no constraints are satisfied. This represents a crucial departure from previous efforts on multi-issue negotiation, where contract utility is calculated as the weighted sum of the utilities for individual issues, producing utility functions shaped like flat hyperplanes with a single optimum. Figure 2 shows an example of a nonlinear utility space. There are two issues, *i.e.*, two dimensions, with domains [0, 99]. There are 50 unary constraints (*i.e.*, that relate to one issue) as well as 100 binary constraints (*i.e.*, that interrelate to two issues). The utility space is highly nonlinear with many hills and valleys.

In our utility function, we assume interdependency between the issues. For example, since an agent has a binary constraint between issues 1 and 2, as Figure 1 shows, they are interdependent for the agent. Therefore, our utility space is highly interdependent.

As is common in negotiation contexts, we assume that agents do not share their utility functions with each other to preserve a competitive edge. In fact, generally agents do not completely know their desired contracts in advance because their own utility functions are simply too large. If we have 10 issues with 10 possible values per issue, for example, this produces a space of 10^{10} (10 billion) possible contracts, which is too many to evaluate exhaustively. Agents must thus operate in highly uncertain environments.

Finding an optimal contract for individual agents with such utility spaces can be handled using such well-known nonlinear optimization techniques as sim-

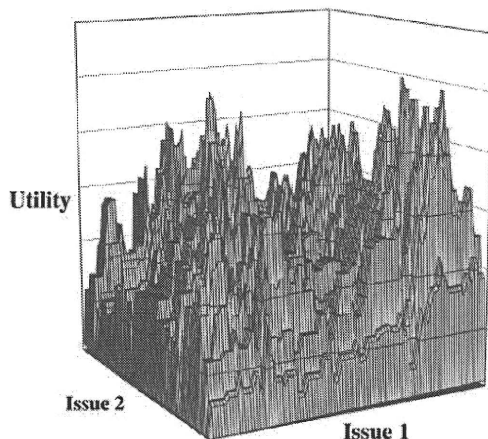


Figure 2: Complex utility space for single agent

ulated annealing or evolutionary algorithms. We cannot employ such methods for negotiation purposes, however, because they require that agents fully reveal their utility functions to a third party, which is generally unrealistic in negotiation contexts.

The objective function for our protocol can be described as follows:

$$\arg \max_{\vec{s}} \sum_{i \in N} u_i(\vec{s}). \quad (1)$$

In other words, our protocol tries to find contracts that maximize social welfare, *i.e.*, the total utilities for all agents. Such contracts, by definition, will also be Pareto optimal.

2.2 Existing Bidding based Protocol

In a previous work [6], agents reach an agreement based on the following steps. This is called a **basic bidding based mechanism**.

[Generate bids] Each agent samples its utility space to find high-utility contract regions. A fixed number of samples are taken from a range of random points, drawn from a uniform distribution. Note that if the number of samples is too low, the agent may miss some high utility regions in its contract space and thereby potentially end up with a sub-optimal contract.

There is no guarantee, of course, that a given sample will lie on a locally optimal contract. Each agent, therefore, uses a nonlinear optimizer based on simulated annealing [9] to find the local optimum in its neighborhood. Figure 3 exemplifies this concept. Black dots are sampling points and white dots are locally optimal contract points.

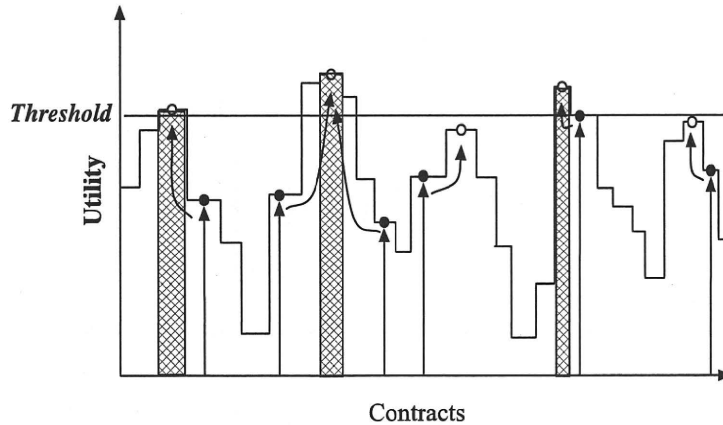


Figure 3: Generating bids

For each contract \vec{s} found by adjusted sampling, an agent evaluates its utility by a summation of the values of satisfied constraints. If that utility is larger than reservation value δ (**threshold**), then the agent defines a bid that covers all the contracts in the region with that utility value. This is easy: the agent merely finds the intersection of all the constraints satisfied by that \vec{s} .

[Find the Solutions] In negotiation, the mediator takes the middle position and identifies the final contract by finding all the combinations of bids, one from each agent, that are mutually consistent, *i.e.*, that specify overlapping contract regions (Figure 4)¹. If there is more than one such overlap, the mediator selects the one with the highest summed bid value (and assuming truthful bidding, the highest social welfare).

2.3 Scalability and Privacy Problems

Since it is a combinatorial optimization calculation, computational complexity for finding solutions exponentially increases based on the number of bids. For example, if there are 10 agents and each agent has 20 bids, the number of bids is 20^{10} . To make our negotiation mechanism scalable, the computational complexity must be reduced to find solutions.

We limited the number of bids for each agent to handle the computational complexity in the basic bidding based protocol [6]. The concrete number of bids in this limitation was $\sqrt[3]{6,400,000}$, a number that reflects our experimental calibration in 2005. But even though CPUs are faster now, the limitation

¹A bid has an acceptable region. For example, if a bid has regions $[0,2]$ for issue 1 and $[3,5]$ for issue 2, the bid is accepted by a contract point $(1,4)$, which means that issue 1 takes 1 and issue 2 takes 4. If a combination of bids, *i.e.*, a solution, is consistent, definitely overlapping regions exist. For instance, a bid with regions (issue 1, issue 2) = $([0,2],[3,5])$, and another bid with $([0,1],[2,4])$ is consistent.

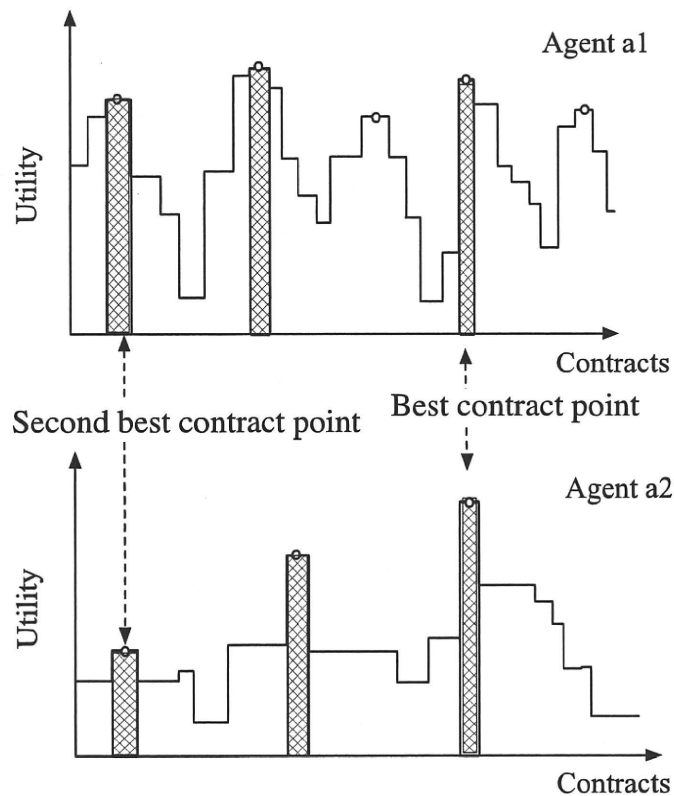


Figure 4: Find solutions

number does not differ so much because this is an exponential problem. Table 1 shows the limitation numbers of bids in one agent. This number quickly drops by increasing the total number of agents. Because of the limitation of bids, the failure rate for finding agreements quickly increases along with increasing the number of agents. When the number of agents is five and the number of issues is seven, we experimentally observed that the failure rate is around 40%. In fact, a strong trade-off exists between increasing the number of total bids and finding good quality solutions. Increasing the number of total bids is not an effective approach for finding good quality agreements.

Thus, it is necessary to build another mechanism that will find higher quality solutions without limiting the bids. Our mechanism proposed in this paper is highly scalable. The other issue with existing protocols is that they are not concerned with privacy in utility spaces. Even in a collaborative situation among people, it is normal to keep one's own utility space closed as long as one is not asked to do otherwise. Our new mechanism achieves such a situation by defining

Num. of agents	Limit of bids	Num. of agents	Limit of bids
2	2530	7	9
3	186	8	7
4	50	9	6
5	23	10	5
6	13		

Table 1: Limitation of the bids

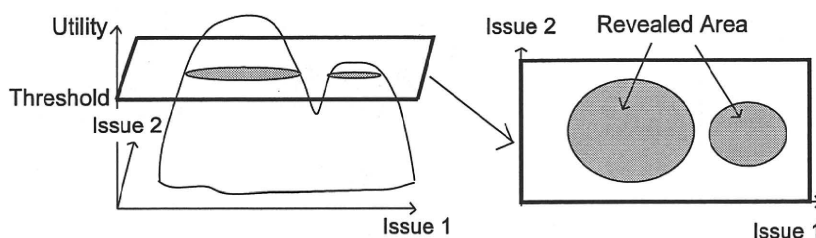


Figure 5: Revealed area

the revealed area in utility spaces.

3 Multi-Round Representative based Protocol based on Revealed Private Information

3.1 Revealed Area for Agent

We focus on the amount of private information agents revealed in the negotiation. We employ **revealed area** as a measure of the amount of revealed utility space. Figure 5 shows an intuitive example of a revealed area, defined as an agent's possible contract points that are revealed in his utility space on his threshold.

For an agent, it is important for him/her to know how much his/her private information is revealed compared with the other agents. The mediator can judge whether an agent is cooperative based on the amount of revealed private information.

We use a **threshold** that is employed in generating bids as a measure of adjusting agents' revealed areas. Since directly adjusting the revealed area is difficult because agents have complex utility spaces, we consider adjusting their threshold to adjust their revealed areas. The threshold is employed for an agent to generate his/her bids based on utility values above the threshold. The threshold was originally adopted to adjust the number of bids. However, in this paper, we also utilize it for determining an agent's revealed area while handling complex utility space.

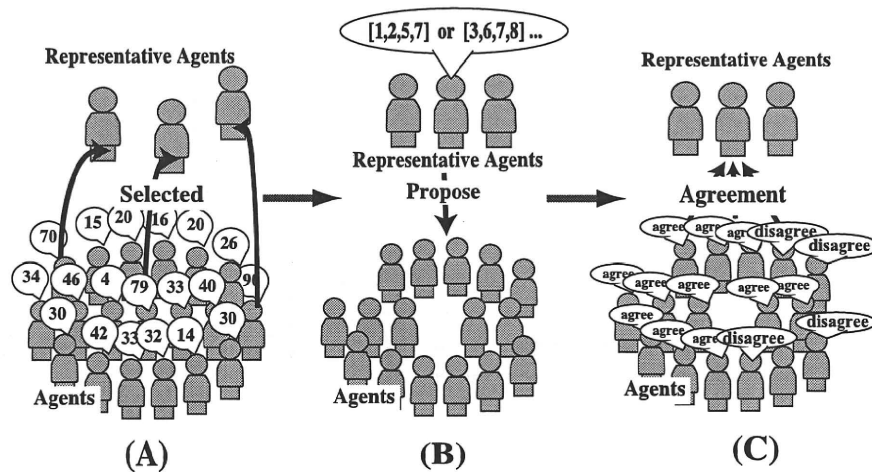


Figure 6: Representative based protocol

3.2 Representative based Protocol

Representative-based protocol consists of three steps. The first step is to select the representative agents (Step1). The second step is to find solutions, and propose them to the other agents (Step2). The third step is to respond to the agreement by the other agents (Step3).

We assume each agent uses a reservation value to determine whether to “agree” or “disagree” with the representative agents. Actually, for practical applications, the reservation value can be determined by a human user. In addition, we assume that the number of representatives is static in representative based protocol. This protocol consists of the following steps.

[Step 1: Selection of Representative Agents] Representative agents are selected based on the amount of their revealed areas, as shown in Figure 6 (A). First, each agent submits how much he can reveal his utility space to the mediator. Namely, each agent submits the numeric value of the amount of his possible revealed area. The mediator selects the representative agents who revealed a large area. We call this selection method RAS. This step is main additional coordination processes by the use of representatives. By employing RAS, all agents are satisfied with the mediator’s decision because it is the best method for all agents to find optimal solutions.

[Step 2: Proposing by Representatives] Representative agents find solutions and propose them to other agents, as shown in Figure 6 (B). First, representative agents find solutions by employing a breadth-first search with branch cutting to find solutions (from lines 3 to 14 in `representative_protocol()`).

Next, the representative agents ask the other agents whether they “agree” or “disagree.” Step 2 is repeated until all the other agents agree or the solutions

found by the representatives are rejected by the other agents.

[Step 3: Respond to Agreement by other Agents] First, the other agents receive the solutions from the representative agents. Then they judge whether they “agree” or “disagree” by determining whether the solution’s utility is higher than their own reservation value (Figure 6 (C)).

Steps 1, 2, and 3 are captured as Algorithms 1 and 2:

Algorithm 1 representative_protocol(B)

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 $B_i = \{b_{i,0}, b_{i,1}, \dots, b_{i,m_i}\}$)

RB : A set of bid-set of each representative agent

($RB = \{RB_0, RB_1, \dots, RB_m\}$, a set of bids from representative agent i is $RB_i = \{rb_{i,0}, rb_{i,1}, \dots, rb_{i,l_i}\}$)

SC : A set of solution-set of each representative agent

($SC = \{SC_0, SC_1, \dots, SC_n\}$, a set of bids from agent i is $SC_i = \{sc_{i,0}, sc_{i,1}, \dots, sc_{i,m_i}\}$)

```

1:  $RB := select\_representative(B)$ 
2:  $SC := RB_0, i := 1$ 
3: while  $i < the\ number\ of\ representative\ agents$  do
4:    $SC' := \emptyset$ 
5:   for  $s \in SC$  do
6:     for  $rb_{i,j} \in RB_i$  do
7:        $s' := s \cup rb_{i,j}$ 
8:     end for
9:   end for
10:  if  $s'$  is consistent then
11:     $SC' := SC' \cup s'$ 
12:  end if
13:   $SC := SC', i := i + 1$ 
14: end while
15: while  $i < |SC|$  do
16:  if  $ask\_agent(SC_i)$  is true &  $SC_i$  Utility is maximum then
17:    return  $SC_i$ 
18:  else
19:    return No Solution
20:  end if
21: end while

```

This protocol is scalable for the number of agents. In a representative protocol, combinatorial optimization only occurs in negotiation among representative agents. In fact, the computational complexity for proposing solutions to unrepresentative agents only increases linearly and is almost negligible. Thus, the computational complexity is drastically reduced compared with the existing mechanism.

Finally, we describe the trade-off for an agent between revealing a large amount of utility space and being a representative agent. Representative agents have advantages since they can propose alternatives to other agents and dis-

Algorithm 2 ask_agent(*SC*)

select_representative() is a method for performing Step 1

Th: A reservation value of each agent ($Th = \{Th_0, Th_1, \dots, Th_n\}$)

```
1: while  $i < \text{the number of agents}$  do
2:   if  $SC'sUtility < Th_i$  then
3:     return false
4:   else
5:      $i := i + 1$ 
6:   end if
7: end while
8: return true
```

advantages because they need to reveal larger utility space. Unrepresentative agents have advantages in keeping their utility hidden and disadvantages in responding based on representatives' agreements.

3.3 Threshold Adjusting Mechanism

We extend our protocol to multi-round negotiation based on the threshold adjusting method [8] so that the number of times to be a representative agent is fair. The total amount of revealed utility space for each agent is almost the same by the threshold adjustment mechanism.

The main idea of the threshold adjusting mechanism is simple: if an agent reveals a larger area of his utility space, he should gain an advantage. On the other hand, an agent who reveals a smaller area of his utility space should adjust his threshold to agree with others. The threshold values are changed by each agent based on the amount of revealed area. If the agent decreases the threshold value, this means that he must reveal more of his utility space.

This mechanism is repeated until an agreement is achieved or all agents refuse to lower their thresholds. Agents can decide whether to lower the threshold based on their reservation value, i.e., the minimum threshold. This means that agents have the right to reject the request to decrease their threshold if the request decreases a threshold lower than the reservation value.

Figure 7 shows an example of the threshold adjusting process among three agents. The upper and bottom figures show the thresholds and the revealed areas before and after adjusting the threshold, respectively. In particular, in this case, agent 3 revealed a small amount of his utility space. The amount of agent 3's revealed utility space in this threshold adjustment is the largest among these three agents. The exact rate of the amount of revealed utility space and the amount of decreased threshold are defined by the mediator or the mechanism designer.

The threshold adjusting mechanism is shown as Algorithm 3:

In the threshold adjusting mechanism, agents can consider others' behaviors by adjusting the agent thresholds. In our definition, agents can reveal more

Algorithm 3 threshold_adjustment()

Ar : Area Range of each agent ($Ar = \{Ar_0, Ar_1, \dots, Ar_n\}$)

representative_protocol(): representative based protocol explained in previous section.

```
1: loop
2:    $i := 1, B := \emptyset$ 
3:   while  $i < |Ag|$  do
4:     bid_generation_with_SA( $Th_i, V, SN, T, B_i$ )
5:   end while
6:    $maxSolution := representative\_protocol(B)$ 
7:   if find  $maxSolution$  then
8:     break loop
9:   else if all agent can lower the threshold then
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
18:  end if
19: end loop
20: return  $maxSolution$ 
```

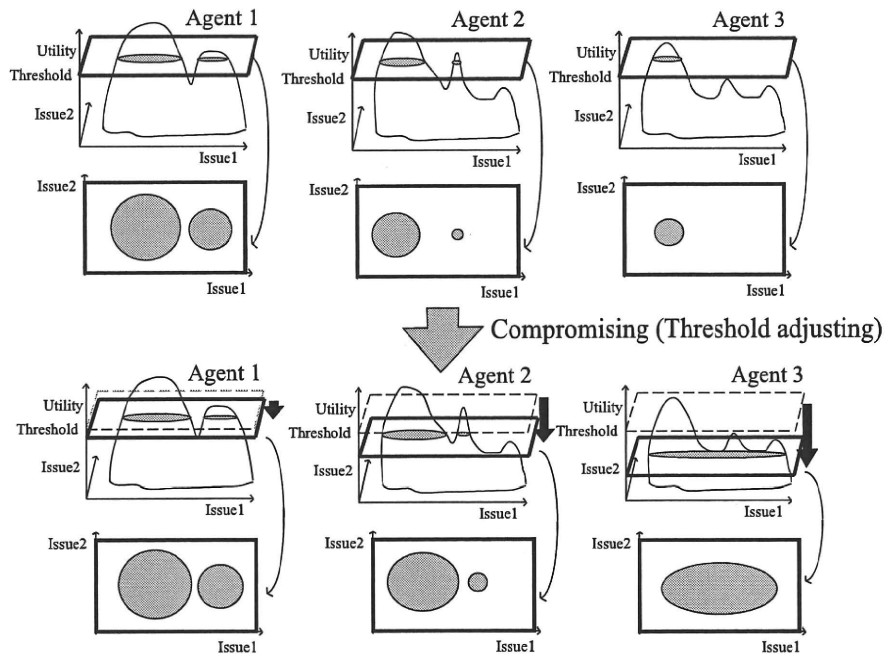


Figure 7: Threshold adjusting process

revealed area if they greatly lower their threshold. Additionally, the width of decreasing the threshold is decided based on a comparison of the others' revealed areas in the threshold adjusting mechanism. Therefore, they can take the behaviors of others into consideration in multi-round negotiation.

4 Experiment Results

4.1 Experiment Settings

We conducted several experiments to evaluate the effectiveness of our approach. In each experiment, we ran 100 negotiations between agents with randomly generated utility functions.

In the experiments on optimality, for each run, we applied an optimizer to the sum of all the agents' utility functions to find the contract with the highest possible social welfare. This value was used to assess the efficiency (*i.e.*, how closely optimal social welfare was approached) of the negotiation protocols. To find the optimum contract, we used simulated annealing (SA) because exhaustive search became intractable as the number of issues grew too large. The SA initial temperature was 50.0 and decreased linearly to 0 over the course of 2500 iterations. The initial contract for each SA run was randomly