

自動交渉マルチエージェント技術に基づく コラボレーティブなデザイン支援システムの試作

Implementation of Collaborative Design Support System based on Automated Multi-Agent Negotiation

奥村 命^{1*} 藤田 桂英¹ 吉村 卓也¹ 伊藤 孝行^{1,2}
Mikoto Okumura¹ Katsuhide Fujita¹ Takuya Yoshimura¹ Takayuki Ito^{1,2}

¹ 名古屋工業大学 情報工学専攻/産業戦略工学専攻/情報工学科

¹ Computer Science and Engineering/Master of Techno-Business Administration/
Dept. of Computer Science, Nagoya Institute of Technology

² 東京大学政策ビジョン研究センター

² Todai Policy Alternatives Research Institute

1 はじめに

近年、インターネットインフラの整備や計算機性能の向上から、世界中から多くの人々が議論に参加し、協調することが可能な場が提供されつつある。具体的には、ビデオ会議システム、Twitter、ソーシャルネットワークなどのシステムが実装されている。大規模な議論を行う場が提供可能になったことで、地球温暖化問題対策、多国籍企業のグローバルな商品開発などの、グローバルで大規模な議論を行うことへのニーズが高まっている。通常の間人同士の議論では、参加者数が増えるほど合意案を形成することは困難になる。大規模な数の意見を集約し、合意を形成するためにはマルチエージェントシステムによる効率的なコラボレーション支援が必要と考えられている。

マルチエージェントによる自動交渉機構に関する既存研究は数多く存在する [1, 2, 3, 4, 5, 6, 7, 8, 9]。しかし、多くの既存研究では、エージェントの完全な効用関数が仮定されていた。現実世界において、ユーザーの効用空間を完全に取得するには膨大な時間がかかる。従って、これまでにマルチエージェントに基づく交渉支援システムの具体的な実装は存在しなかった。

本論文では、マルチエージェントの自動交渉機構 [1, 2, 3, 4, 5, 6, 7, 8, 9] に基づいた交渉支援システムの一例として、コラボレーティブな公園設計支援システムを試作する。特に、実装の少ない交渉対象となる設計モデルを属性値で表す機構と、ユーザーによる評価をベースとした効用空間抽出機構に焦点を当てる。本シ

ステム試作の目的は、現実世界においてマルチエージェントによる自動交渉支援システムが有用であることを実際にシステムを試作し、評価、確認することである。

本論文の構成を以下に示す。2章でシステム全体の構成について説明し、3章では効用空間の抽出機構を提案し、4章で効用空間の抽出例を示す。5章では、システムの有用性を確認するために、評価者実験を行い考察した。最後にまとめと今後の課題を示す。

2 コラボレーティブな公園設計支援システム

図1のような仕組みで共同公園設計を支援する共同設計支援システムを試作開発する。本システムの基本ステップは以下の [Step1] ~ [Step5] からなる。

[Step1] ユーザーの意見、好みの収集機構
本ステップでは、サンプリング点を選出し、対応する設計モデルをプログラムにより生成する。生成した設計モデルに対するユーザー評価を得ることで、ユーザーの好みを抽出する。

[Step2] ユーザーの効用空間形成機構
3章で述べる効用空間形成手法を用い、[Step1] で得たサンプリング点から各ユーザーの好みを表す効用空間を形成する。

[Step3] エージェントによる自動交渉機構
[Step2] で抽出した各ユーザーの効用空間をユーザーの代理となるエージェントに与え、自動交渉技術 [1, 2, 3, 4, 5, 6, 7, 8, 9] を用い、ユーザー達にとって好ましい合意案を発見する。

*連絡先：名古屋工業大学 産業戦略工学専攻
〒466-8555 愛知県名古屋市昭和区御器所町
E-mail: okumura@itolab.nitech.ac.jp

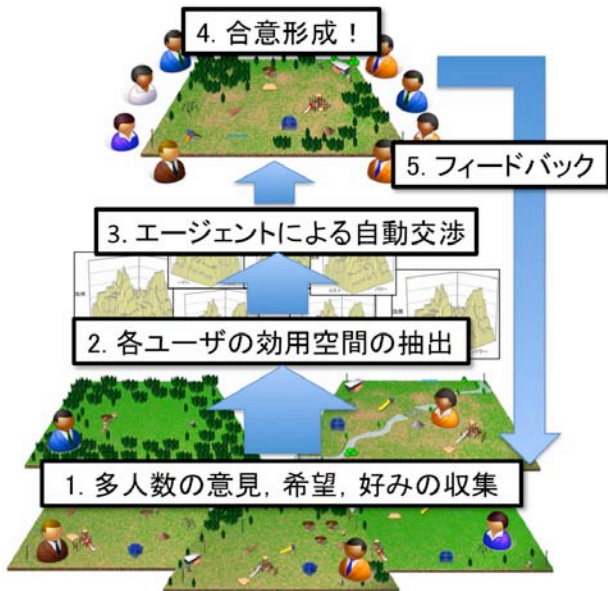


図1: 公共空間の共同設計プロセス

[Step4] 合意案提案機構

得られた合意案（属性空間上の点）から設計モデルを生成する。

[Step5] フィードバック機構

ユーザーへ合意案をフィードバックし、多くのユーザーが納得する合意案が形成できるまで [Step1]～[Step5] を繰り返す。本来は、より良い合意案を形成するために重要な機構であるが、今回は省略した。なぜなら、本機構の実現は容易ではなく、本研究では現実世界におけるマルチエージェントによる自動交渉支援システムの有用性を示すことを目指しており、本機構が無くても、交渉支援システムとして十分成立するため、本研究目標を達成することが可能であるからである。

3 効用空間抽出機構

3.1 効用空間抽出機構の概要

本章で提案する効用空間抽出機構は、2章で説明した設計支援システムの [Step1], [Step2] に相当する。本機構では設計モデルを自動生成し、生成された設計モデルに対するユーザー評価を得ることにより、ユーザーの好みを表す効用空間を抽出する。

本論文ではパラメトリックな設計モデルを扱い、設計モデルを表現するために必要なパラメータをそのまま属性として採用することにより、設計モデルと属性空間上の点を1対1に対応付ける。つまり、属性値を与えることで対応する設計モデルを得ることができる。従って、合意案提案機構（2章で述べたシステムの [Step4]）に

において、エージェントによる自動交渉の結果として得られる合意案（属性空間上の点）を設計モデルへ変換することが可能であり、効用空間抽出機構のユーザーによる設計モデル評価ステップにおいて、設計モデルをプログラムにより自動生成することで、ユーザーから数多くの評価を得ることを実現する。

本論文で提案する効用空間の形成手法では、ベースとなる基本関数を用いて複雑な効用空間を表す関数を表現する。サンプリング点におけるユーザー評価値を効用値とし、サンプリング点から離れるほど効用値が減少するような形状をした基本関数を用意する。サンプリング点で発生した基本関数の山を組み合わせることで、効用空間全体の複雑な凹凸形状を表現する。本手法は情報の確かさを考慮し、サンプリングデータに近く、より確かな点で合意が形成されることを促進する。

本論文におけるサンプリングとは、2章で説明した設計支援システムの [Step1] で行なわれるサンプリングに相当する。各サンプリング点は設計モデル（公園デザイン）を表す属性値のベクトルである。ユーザーは各サンプリング点（設計モデル）に対し、評価値を付ける。

3.2 基本的な効用関数の定義

定義1 基本関数の定義

\mathbb{R}^+ を0以上の実数集合、 \mathbb{R}^{++} を正の実数集合とする。 i 番目のサンプリング点が s_i で、 s_i に対するユーザー評価点が v_i であるとき、 s_i に対応する基本関数 f_i を以下のように定義する。ここで、 d_i は f_i の広がり具合を表すパラメータである ($v_i, d_i \in \mathbb{R}^{++}$)。

$$f_i(\vec{x}) = v_i \cdot \exp\left(-\frac{(\vec{x} - \vec{s}_i)^2}{d_i}\right) \quad (1)$$

基本関数の形状は、“効用空間を表す関数は、属性空間上の距離の近い2点は似通った効用値をとる”という仮定の下に設計されている。基本関数の条件を下記に示す。また、上記関数はガウス関数であり、下記基本関数の条件を満たす。

- 値域が0以上の多次元空間上の関数

$$f_i : \mathbb{R}^{+n} \rightarrow \mathbb{R}^+$$

- 最大値はユーザーによる評価点に等しい

$$\max f_i(\vec{x}) = v_i$$

- サンプリング点で最大値をとる

$$\arg \max_{\vec{x}} f_i(\vec{x}) = \vec{s}_i$$

- サンプリング点から離れるほど、値が減少する

$$\|\vec{x}_1 - \vec{s}_i\| > \|\vec{x}_2 - \vec{s}_i\| \rightarrow f_i(\vec{x}_1) < f_i(\vec{x}_2)$$

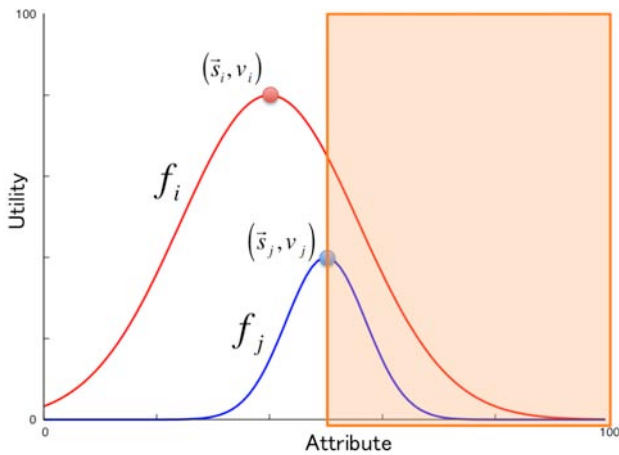


図 2: 基本関数が別の基本関数の下側に入り込んでしまっている場合

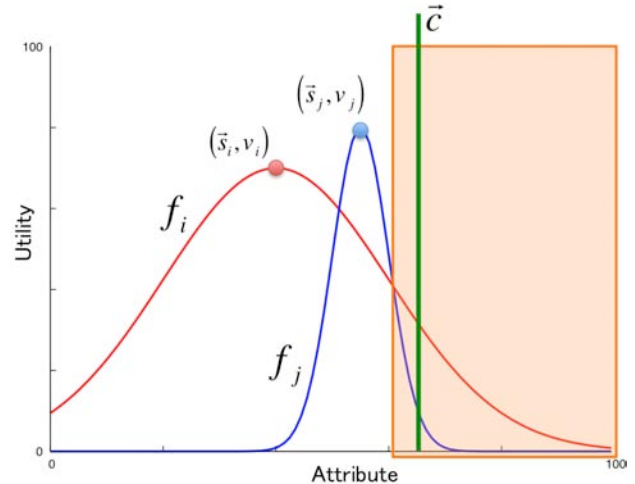


図 4: 基本関数の大部分が別の基本関数の下側に入り込んでしまっている場合

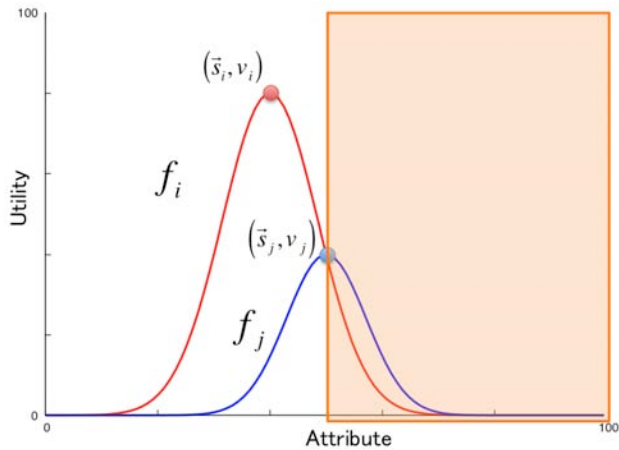


図 3: 基本関数が別の基本関数の下側に入り込んでしまっている状態 (図 2) を解決した図

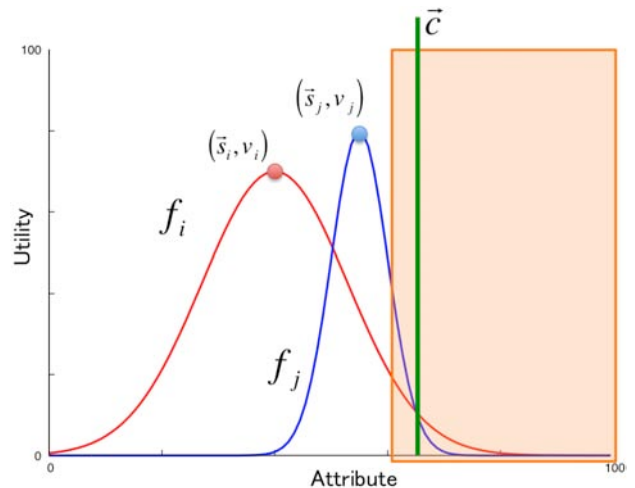


図 5: 基本関数の大部分が別の基本関数の下側に入り込んでしまっている状態 (図 4) を解決した図

3.3 効用空間の形成

定義 2 効用関数の定義

N 個のサンプリング点 s_1, \dots, s_N が得られたとき, 効用関数 U を以下のように定義する.

$$U(\vec{x}) = \max_{i=1, \dots, N} f_i(\vec{x}) \quad (2)$$

しかし, 単純に上記定義によって効用関数 U を求めた場合, 図 2, 4 に示す問題が発生してしまう. 図 2 では, 関数 f_j が f_i の下側にあるため, サンプリング s_j によって得られた情報を全く活用できていない. 実際, 図中のオレンジ色の領域は, サンプリング点 s_i よりも s_j に近いため, f_j の値を採用すべきであるが, 効用関数では f_i が採用されてしまう. 図 4 では, 裾の狭い関数 f_j の大部分が, 裾の広い関数 f_i の下側にあるため, サンプリング s_j によって得られた情報を正しく反映で

きていない. 具体的には, 図中のオレンジ色の領域は, サンプリング点 s_i よりも s_j に近いため, f_j の値を採用すべきであるが, 効用関数では f_i が採用されてしまう. 以下に示す 2 手法では, 基本関数 f_i の広がり具合を表す変数 d_i を変更することで, 本問題を解決する.

手法 1 基本関数が別の基本関数の下側に入り込んでしまっている状態 (図 2) を解決する.

本手法では, 図 2 のような状況では, d_i を調節することで, 図 3 のように修正する. 具体的には, 2 つの異なるサンプリング点 s_i, s_j について, $f_i(s_j) > f_j(s_j)$ ならば, 式 (3) によって $f_i(s_j) = f_j(s_j)$ となるように d_i を変更する ($i \neq j, \max f_i(\vec{x}) \geq \max f_j(\vec{x})$) [付録 A].

$$d_i = \frac{(s_j - s_i)^2}{\ln \frac{v_i}{v_j}} \quad (3)$$



図 6: 設計モデルのベース作成システム

手法 2 基本関数の大部分が別の基本関数の下側に入り込んでしまっている状態 (図 4) を解決する。

本手法では、図 4 のような状況では、 d_i を調節することで、図 5 のように修正する。具体的には、2 つの異なるサンプリング点 s_i, s_j について、 $f_i(\vec{c}) > f_j(\vec{c})$ ならば、式 (4) によって $f_i(\vec{c}) = f_j(\vec{c})$ となるように d_i を変更する ($i \neq j, d_i > d_j$) [付録 B]。

$$d_i = \frac{(\vec{c} - \vec{s}_i)^2}{\frac{k^2}{2} - \ln \frac{v_i}{v_j}} \quad (4)$$

ここで、 $\vec{c} = \vec{s}_j + k\sqrt{\frac{d_j}{2}}\vec{u}$ 、 $\vec{u} = \frac{1}{\|\vec{s}_j - \vec{s}_i\|}(\vec{s}_j - \vec{s}_i)$ で、 \vec{u} は \vec{s}_i から \vec{s}_j へ向かう方向を表す単位ベクトルであり、 \vec{c} は変更の基準点を指す ($k \in \mathbb{R}^{++}$)。基準点 \vec{c} はパラメータ k によって変化し、 k が大きいほど s_j から離れた点で本手法を実行する。例えば、図 5 の場合、 k が大きくなると \vec{c} が右方向へ移動する。本論文で示すシステムでは、 $k = 2$ としている。

3.4 効用空間の抽出

本論文で提案する効用空間の抽出機構は、以下に示す [Step A] ~ [Step D] で構成される。

[Step A] デザインのベース作成

図 6 に示すベース作成システムを用いて、交渉問題の管理者が公園デザインのベースを作成する。本ベース作成システムでは、木、遊具、設備等のおおまかな配置を決定する。[Step C] における設計モデルの自動生成時に、プログラムによって決定されるパラメータを自由に変化させ、対応する設計モデルを確認することができる。

[Step B] サンプリングポイントの選出

ユーザーによる評価を行う属性空間上の点を決定する。ユーザーインタラクションを伴うため、サンプリング可能数は限られている。従って、少ないサンプリング数

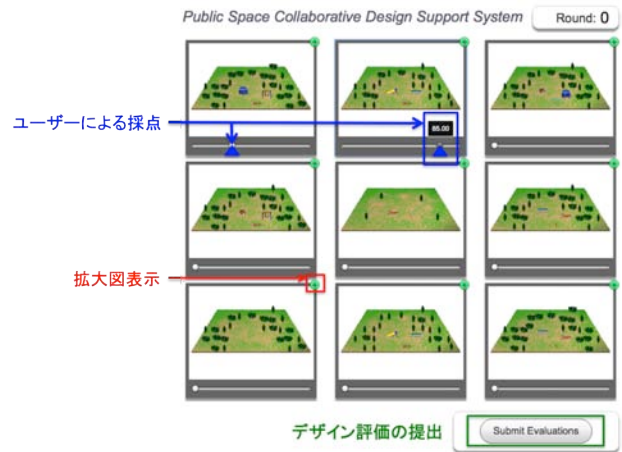


図 7: デザイン評価システム

で効率良く効用空間を形成しなければならないが、今回はランダムにサンプリング点を選出している。

[Step C] ユーザーによる評価

選出した属性空間上の点に対応する設計モデルを、[Step A] で作成されたベースデザインを用いて、プログラムによって生成し、ユーザーから評価を得る。評価方法としては、投票方式や採点方式などが考えられるが、今回は採点方式を採用する。ユーザーは図 7 に示したデザイン評価システムを用いて生成されたデザインを評価し、提出する。

[Step D] 効用空間の形成

[Step C] により i 番目に新しく得られた評価点から、基本関数 f_i を生成する。この時、 $d_i = D_0$ とする。新規基本関数 f_i と既存基本関数 f_j に対して、手法 1, 2 をこの順で適用し、 d_i, d_j を調節する ($0 \leq j < i$)。 D_0 は基本関数の広がり具合を表すパラメータの初期値である。基本関数 f_i の広がり具合を表すパラメータ d_i は、手法 1, 2 によって徐々に小さい値に修正される。従って、 D_0 を効用空間に対して、大きな値とすることで、サンプリング数が少ない段階では効用空間の概形を表わし、サンプリング数が増加するに従って、効用空間の詳細な形状を表現することができる (サンプリング数が増えるに従い、効用関数が多峰性になる)。本論文で示すシステムでは、 $D_0 = 100$ としている。以降、[Step B] ~ [Step D] を繰り返す。

サンプリング点から効用空間を形成する最も単純な方法は、図 8 のようにサンプリング点を滑らかに繋ぐ方法である。しかし、サンプリング点間の距離が大きい場合、効用値を高く評価してしまう可能性がある。本手法では、図 9 に示すように、基本関数の最大値を取ることによって、本問題を解決している。

本手法ではサンプリング点から離れた点、即ち情報が不足している点では、効用値が低くなるように設計

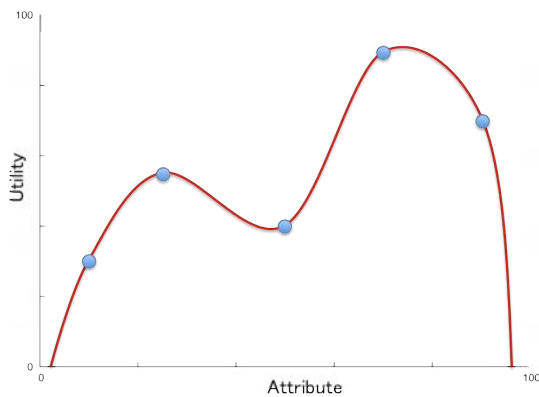


図 8: サンプル点を滑らかに繋ぐことで形成する効用関数のモデル

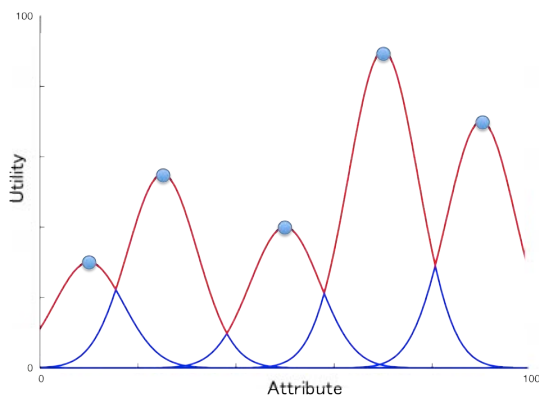


図 9: 基本関数の最大値を取ることで形成する効用関数のモデル

しており、情報が多点で合意し易くなっている。一方、サンプリングが不足している段階では合意形成が難しい。しかし、多くの交渉相手が合意を求める点をサンプリングすることで本問題は解決する。

4 効用空間の抽出例

本手法の正しさを評価するために、一定の方針をもって設計モデルを評価し、抽出された効用空間に設定した方針が現れているか確認する。

今回は、“緑が豊かで、遊具がある程度充実した公園を最も高く評価し、遊具が多過ぎるか或いは少な過ぎる公園は自然度に関係無く低く評価する”という方針で設計モデルを評価し、著者の効用空間を抽出した。実際に抽出した効用空間を図 10 に示す。今回の抽出例で

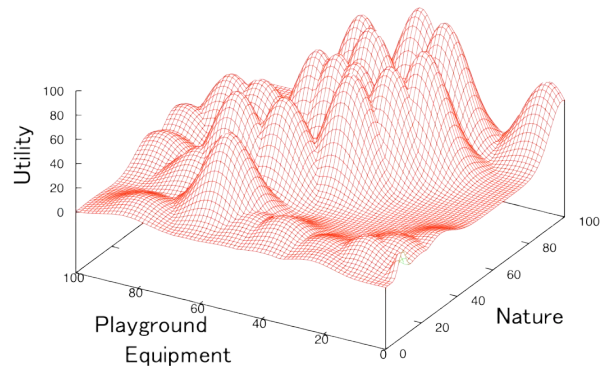


図 10: 抽出した効用空間

は、サンプリング数を 30、属性数を 2 とした。今回は視覚的に評価するために属性数を 2 としたが、本手法は属性数が 3 以上の多属性な効用空間抽出に適用することができる。

図 10 中の軸“Nature”は公園の自然度を表し、値が大きいほど緑豊かな公園であることを示す。軸“Playground Equipment”は遊具充実度を表し、値が大きいほど遊具が充実した公園であることを示す。軸“Utility”は効用値を表し、値が大きいほど対応する設計モデルをユーザーが好む。

“Playground Equipment”が 50 に近く、“Nature”が高いほど効用値が高い効用空間が抽出され、設定した方針が効用空間に正しく現れていることが分かる。今回の抽出結果から、ユーザーの好みに一定の規則がある場合、好みが抽出した効用空間に正しく表れることが分かる。

“Nature”が 60、“Playground Equipment”が 50 の点付近でサンプリング点が密集してしまっている。効率的な効用空間の抽出のためには、サンプリング点の選出方法を考慮する必要がある。

5 評価者実験

本システムの評価者実験を行った。属性は自然度と遊具充実度の 2 つを設定した。各属性は 0~100 の実数値を取る。交渉は被験者 11 人で 2 度行った。交渉時間をそれぞれ 1 度目は 10 分、2 度目は 5 分に設定した。2 度目の交渉時間が 1 度目よりも短い理由は、被験者がシステムの操作に慣れたためである。

図 11 に交渉結果として得られたデザインを示し、表 1 に交渉結果に対する効用値に関する情報をまとめた。表 1 中の U1、U2 は、交渉結果に対するユーザーの効用値である。本値は交渉終了後、ユーザーがシステムを利用する際と同様にして交渉結果に対して採点した

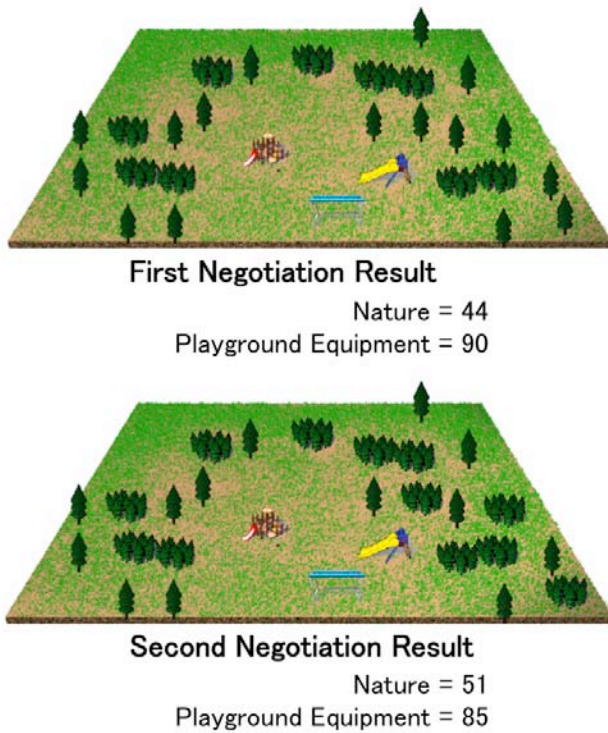


図 11: 交渉結果

値である。Est1, Est2 は抽出した効用関数を用いて交渉結果を評価した値である。Err1, Err2 はユーザーの実際の効用値と抽出した効用関数で交渉結果を評価した値との誤差である。また、U1, Est1, Err1 は 1 回目の交渉に関する情報で、U2, Est2, Err2 は 2 回目の交渉に関する情報である。

U1 の平均が 79.09 で、U2 の平均が 79.36 であることから、両交渉結果で高い評価値平均を得ており、多くのユーザーが納得する合意案を形成できたことが分かる。Est1 の平均が 15.78 で、Est2 の平均が 16.60 であることから、効用空間抽出機構の精度は良くないものの、ユーザーの好みの傾向を十分抽出できていることが分かる。また、U1, U2 の平均と Est1, Est2 の平均からも分かる通り、本効用空間抽出手法が全体的に効用値を低く推測していることが分かる。本効用空間抽出手法が情報の不確かさを考慮し、サンプリングが不足している点で合意形成する可能性を抑制していることが原因と考えられる。本方針はサンプリングが不足している領域では有用であるが、サンプリングが十分に得られた領域では、サンプリング点から離れても効用値の減少しない効用空間抽出手法を適応するなど、属性空間上のサンプリング点の分布を考慮した効用空間抽出手法を考える必要がある。

また、表 1 が示すように、多くのユーザーについて、User5 のように誤差 (Err1, Err2) が非常に低く、精度

	U1	U2	Est1	Est2	Err1	Err2
User1	80	85	94	61	14	24
User2	80	80	-	64	-	16
User3	60	50	19	74	41	24
User4	80	85	-	-	-	-
User5	85	75	80	73	5	2
User6	90	97	62	82	28	15
User7	70	80	57	51	13	29
User8	90	86	67	58	23	28
User9	80	80	90	87	10	7
User10	90	90	86	75	4	15
User11	65	65	69	71	4	6
Average	79.09	79.36	69.30	69.60	15.78	16.60

表 1: 効用値に関するデータ

良く効用関数を抽出できているが、User3 のように一部のユーザーについて、うまく効用関数を抽出できていない場合が存在する。効用関数をうまく抽出できた例として、User5 から抽出した効用関数を図 12 に示し、うまく抽出できなかった例として、User3 から抽出した効用関数を図 13 に示した。User5 の効用関数に比べ、User3 の効用関数には尖った山が多く発生している。本状況は、似たデザインに対して、大きく異なる評価をした場合に生じる。似たデザインとは、属性空間上で近いデザインのことをいう。実際、評価者実験後にアンケートを行った所、「交渉の途中で評価基準が変化した」という回答を User3 から得られた。つまり、今回の場合は選好が時間的に変化していることが原因と考えられる。本効用空間抽出機構では、一貫した選好を想定しており、選好が変化することを想定していない。しかし、現実世界では時間経過や他者の発言によって選好が変化することは多々存在する。本問題は、Preference Elicitation に関する多くの既存研究 [10, 11, 12] でも考慮されておらず、また効用理論に潜在的に存在する問題といえる。

6 関連研究

複数論点交渉問題の合意形成支援に関する多くの既存研究 [1, 2, 3, 4] では、論点の独立性が仮定されているが、現実的な交渉問題では、論点間に依存性がある場合が多い。[5, 6, 7, 8, 9] では、複数論点交渉問題における論点間の依存関係を考慮した合意形成支援について提案している。しかし、多くの既存研究では効用空間を仮定しているため、効用空間の抽出機構が必要である。本論文では論点間に依存関係の存在する複数論点から成る効用空間を抽出する機構を提案する。

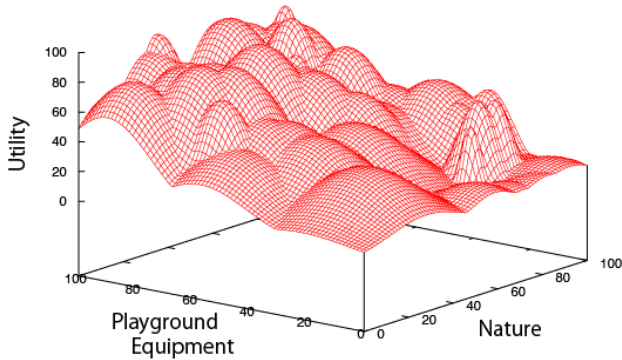


図 12: 効用関数をうまく抽出できた例 (User5 から抽出した効用関数)

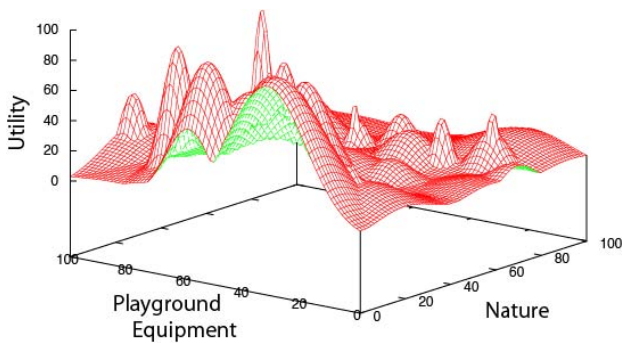


図 13: 効用関数をうまく抽出できなかった例 (User3 から抽出した効用関数)

効用空間の抽出機構として、文献 [10] では、トレードオフとなる属性を取り出す方法を提案している。システムは異なる2つの属性間に存在するトレードオフ関係をユーザーと対話することで抽出し、数値化する。しかし、[10]の手法では、3つ以上の複数の属性間に依存関係が存在する場合や、属性間依存が非線形である場合などの複雑な効用空間の抽出を行うことができない。本手法では、このような複雑な効用空間の抽出を行うことができる。

[13, 14]では、インタラクティブGAを用いた設計支援システムを提案し、ユーザーの好みを反映させた設計案を生成する。しかし、一度に利用可能なユーザーは1人に限られてしまう。本論文で提案する自動交渉エージェントを用いたコラボレーティブ設計支援システムでは、大人数での共同設計が可能であり、多くのユーザーが納得する合意案を形成する。

7 まとめ

本論文では、マルチエージェントの自動交渉機構に基づいたコラボレーティブな公共空間設計支援システムを試作した。特に、交渉対象となる設計モデルを属性空間上の点と対応付ける手法と、ユーザーによる評価をベースとした効用空間の抽出機構に焦点を当てた。また、評価者実験を行い、多くのユーザーが納得する合意案の形成に成功し、システムの有用性を確認した。

今後の課題としては、効用理論の拡張、或いは新しい選好記述モデルを考案し、選好の変化を考慮する必要がある。また、より良い合意案の形成や選好の変化に対応するためにも、フィードバック機構を実装する必要がある。効用関数抽出のためにはユーザーからのサンプリングが欠かせないが、ユーザーの負担軽減のためには、少ないサンプリング数で効率的に効用空間を形成できるようなサンプリング点の選出方法を考える必要がある。

参考文献

- [1] Peyman Faratin, Carles Sierra, and Nicholas R. Jennings. Using similarity criteria to make issue trade-offs in automated negotiations. In *Artificial Intelligence*, pages 142:205-237, 2002.
- [2] S. Shaheen Fatima, Michael Wooldridge, and Nicholas R. Jennings. Multi-issue negotiation with deadlines. *J. Artif. Intell. Res. (JAIR)*, 27:381-417, 2006.
- [3] S. Shaheen Fatima, Michael Wooldridge, and Nicholas R. Jennings. Approximate and online multi-issue negotiation. In *AAMAS*, page 156, 2007.
- [4] S. Shaheen Fatima, Michael Wooldridge, and Nicholas R. Jennings. An analysis of feasible solutions for multi-issue negotiation involving nonlinear utility functions. In *AAMAS (2)*, pages 1041-1048, 2009.
- [5] Ito, T., Hattori, H., and Klein, M.: Multi-issue Negotiation Protocol for Agents : Exploring Nonlinear Utility Spaces, in *Proc. of IJCAI-2007*, pp. 1347-1352 (2007).
- [6] 藤田 桂英, 伊藤 孝行, and Mark Klein. 複数論点交渉問題における論点間の依存関係を考慮した合意形成機構. In *合同 エージェントワークショップ&シンポジウム (JAWS2009)*, pages 202-209, 2009.

- [7] Katsuhide Fujita, Takayuki Ito, Mark Klein: Approximately fair and secure protocols for multiple interdependent issues negotiation. AAMAS (2) 2009: 1287-1288.
- [8] Katsuhide Fujita, Takayuki Ito, Mark Klein: Secure and efficient protocols for multiple interdependent issues negotiation. Journal of Intelligent and Fuzzy Systems 21(3): 175-185 (2010).
- [9] Nobuyasu Mizutani, Katsuhide Fujita, Takayuki Ito: Issue Clustering and Distributed Genetic Algorithms for Multi-issue Negotiations. ACIS-ICIS 2010: 593-598.
- [10] X Luo, N. R. Jennings, and N. Shadbolt. Acquiring user strategies and preferences for negotiating agents: A default then adjust method. International Journal of Human Computer Studies, 64(4):304-321, 2006.
- [11] Kuriyama Koichi and Ishii Yutaka. Estimation of the Environmental Value of Recycled Wood Wastes : A Conjoint Analysis Study. Journal of forest research 5(1), 1-6, 2000-02-16.
- [12] Tuomas Sandholm and Craig Boutilier. Preference Elicitation in Combinatorial Auctions. Proceedings of the Third ACM Conference on Electronic Commerce, 2001.
- [13] 難波 政佳, 小部 昌史, 蔡 東生. インラクティブ GA を用いた日本庭園の設計. 情報処理学会研究報告. 人文科学とコンピュータ研究会報告 2001(96), 55-62, 2001-10-12.
- [14] 難波 政佳, 小部 昌史, 蔡 東生. インラクティブ GA による仮想日本庭園設計. 情報処理学会研究報告. グラフィクスと CAD 研究会報告 2003(86), 53-58, 2003-08-18

A 式3の導出

(3) 式では, $f_i(\vec{s}_j) = f_j(\vec{s}_j)$ となるように d_i を変更する.

$$\begin{aligned} f_i(\vec{s}_j) &= f_j(\vec{s}_j) \\ v_i \cdot \exp\left(-\frac{(\vec{s}_j - \vec{s}_i)^2}{d_i}\right) &= v_j \\ d_i &= \frac{(\vec{s}_j - \vec{s}_i)^2}{\ln \frac{v_i}{v_j}} \end{aligned}$$

以上より, d_i が導出された.

また, f_i がガウス関数であるためには, $d_i > 0$ でなければならない.

$$\begin{aligned} d_i &= \frac{(\vec{s}_j - \vec{s}_i)^2}{\ln \frac{v_i}{v_j}} > 0 \\ v_i &> v_j \end{aligned}$$

手法1の適応条件 $f_i(s_j) > f_j(s_j)$ より,

$$v_i = f_i(s_i) > f_i(s_j) > f_j(s_j) = v_j$$

以上より, $d_i > 0$ が示された.

B 式4の導出

(4) 式では, $f_i(\vec{c}) = f_j(\vec{c})$ となるように d_i を変更する.

$$\begin{aligned} f_i(\vec{c}) &= f_j(\vec{c}) \\ v_i \cdot \exp\left(-\frac{(\vec{c} - \vec{s}_i)^2}{d_i}\right) &= v_j \cdot \exp\left(-\frac{(\vec{c} - \vec{s}_j)^2}{d_j}\right) \end{aligned}$$

$\vec{c} = \vec{s}_j + k\sqrt{\frac{d_j}{2}}\vec{u}$ だから,

$$\exp\left(-\frac{(\vec{c} - \vec{s}_i)^2}{d_i}\right) = \frac{v_j}{v_i} \cdot \exp\left(-\frac{k^2}{2}\vec{u}^2\right)$$

\vec{u} は単位ベクトルだから, $\vec{u}^2 = 1$ なので,

$$\begin{aligned} \exp\left(-\frac{(\vec{c} - \vec{s}_i)^2}{d_i}\right) &= \frac{v_j}{v_i} \cdot \exp\left(-\frac{k^2}{2}\right) \\ d_i &= \frac{(\vec{c} - \vec{s}_i)^2}{\frac{k^2}{2} - \ln \frac{v_j}{v_i}} \end{aligned}$$

以上より, d_i が導出された.

また, f_i がガウス関数であるためには, $d_i > 0$ でなければならない.

$$\begin{aligned} d_i &= \frac{(\vec{c} - \vec{s}_i)^2}{\frac{k^2}{2} - \ln \frac{v_j}{v_i}} > 0 \\ \frac{k^2}{2} - \ln \frac{v_j}{v_i} &> 0 \\ \exp\left(\frac{k^2}{2}\right) &> \frac{v_j}{v_i} \\ \frac{v_j}{v_j \cdot \exp\left(-\frac{k^2}{2}\right)} &> \frac{v_j}{v_i} \end{aligned}$$

$f_j(\vec{c}) = v_j \cdot \exp\left(-\frac{k^2}{2}\right)$ と手法2の適応条件 $f_i(\vec{c}) > f_j(\vec{c})$ より, $f_i(\vec{s}_i) > f_i(\vec{c}) > f_j(\vec{c})$ だから,

$$\frac{v_j}{v_j \cdot \exp\left(-\frac{k^2}{2}\right)} = \frac{v_j}{f_j(\vec{c})} > \frac{v_j}{f_i(\vec{c})} > \frac{v_j}{f_i(\vec{s}_i)} = \frac{v_j}{v_i}$$

以上より, $d_i > 0$ が示された.