

## エージェント間の自動交渉に基づいた コラボレーティブな公共空間設計支援システムの試作

奥村 命<sup>†1</sup> 水谷 信泰<sup>†1</sup> 中川 裕揮<sup>†1</sup>  
藤田 桂英<sup>†1,†2</sup> 伊藤 孝行<sup>†1,†3,†4</sup>

近年、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、世界中から多くの人が同時に参加して議論をする必要がある場面が増えている。大規模な数の意見を集約し、合意を形成するためには、自動交渉エージェントによる効率的なコラボレーション支援が効果的に考えられる。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合意形成支援システムを提案する。特に、交渉対象となる設計モデルと属性空間を対応付ける手法と、ユーザーの効用空間を抽出する機構を扱う。一例として、公園デザインの作成を支援するシステムを取り上げ、ユーザーの好みを抽出する。

### An Implementation of A Collaborative Public Space Design Support System based on Multi-Agent Automated Negotiation

MIKOTO OKUMURA,<sup>†1</sup> NOBUYASU MIZUTANI,<sup>†1</sup>  
HIROKI NAKAGAWA,<sup>†1</sup> KATSUhide FUJITA<sup>†1,†2</sup>  
and TAKAYUKI ITO<sup>†1,†3,†4</sup>

Recently, there have been a lot of discussion among the many people is increasing. In this paper, we propose a collaborative park-design support system based on the multi-agent systems. In this system, agents elicit the utility information of users, collect many alternatives, and reach optimal agreements based on the automated negotiation protocol. Especially, we focus on the steps of deciding the attribute space and estimating the utility spaces of users in real world.

#### 1. はじめに

近年、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、世界中から多くの人が同時に参加して議論をする必要がある場面が増えている。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<sup>1)-5)</sup>。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。

本論文では、マルチエージェントの自動交渉機構<sup>1)-5)</sup>に基づいたコラボレーティブな公園設計支援システムを提案する。特に、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、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でから各ユーザーの好みを表す効用空間を形成する。

<sup>†1</sup> 名古屋工業大学 情報工学科/情報工学専攻/産業戦略工学専攻

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

<sup>†2</sup> マサチューセッツ工科大学スローン経営大学院

Sloan School of Management, Massachusetts Institute of Technology

<sup>†3</sup> 東京大学政策ビジョン研究センター

Todai Policy Alternatives Research Institute

<sup>†4</sup> 科学技術振興機構さきがけ

JST PREST

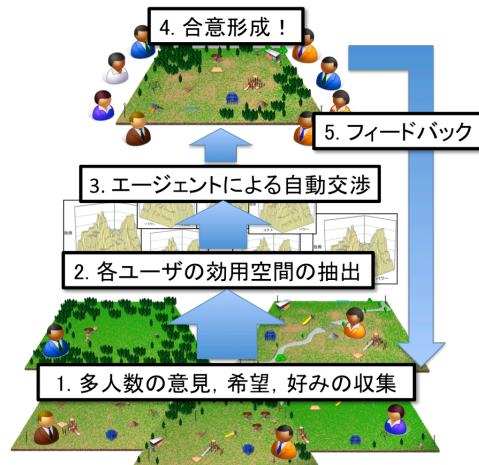


図 1 公共空間の共同設計プロセス

Fig. 1 A Process of Designing A Public Space Collaboratively

[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<sup>1)-5)</sup>を用い、ユーザー達にとって好ましい合意案を発見する。

[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。

### 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といことである。従って、合意案提案機構（2章で述べたシステムの [Step4]）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、ユーザーから多くの評価を得ることが可能である。

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す。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。

#### 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 ( $v_i, d_i \in \mathbb{R}^{++}$ )。

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

基本関数の山のような形状は、“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。上記関数はガウス関数であり、下記に示す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

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

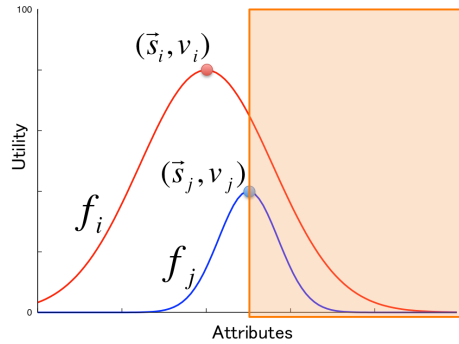


図2 基本関数が別の基本関数の下側に入り込んでしまっている状態  
Fig.2 A Problem that One Base Function is Under Another Base Function

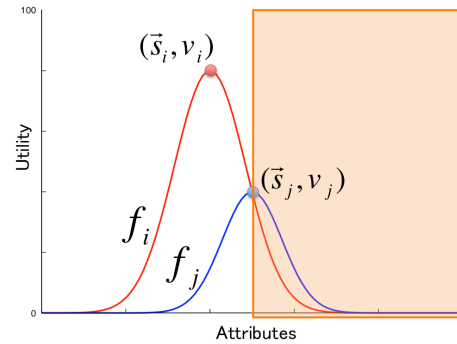


図3 基本関数が別の基本関数の下側に入り込んでしまっている状態 (図2) を解決した図  
Fig.3 A Resolution of The Problem that One Base Function is Under Another Base Function (Fig. 2)

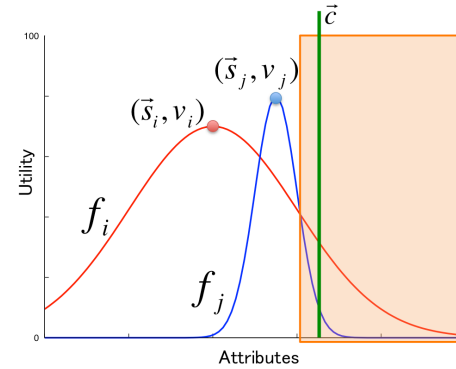


図4 基本関数の大部分が別の基本関数の下側に入り込んでしまっている状態  
Fig.4 A Problem that Many Parts of One Base Function are Under Another Base Function

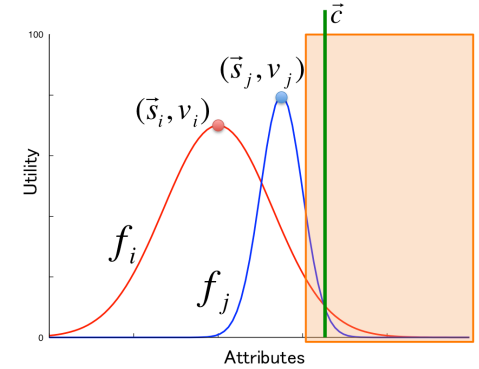


図5 基本関数の大部分が別の基本関数の下側に入り込んでしまっている状態 (図4) を解決した図  
Fig.5 A Resolution of The Problem that Many Parts of One Base Function are Under Another Base Function (Fig. 4)

### 3.3 効用空間の形成

#### 定義2 効用空間の定義

$N$  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、サンプリング点  $s_j$  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  $s_j$  に近い場合、 $f_j$  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、サンプリング点  $s_j$  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  $s_j$  に近い場合、 $f_j$  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.1].

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

**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A.2].

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

ここで、 $\vec{c} = s_j + k\sqrt{d_j/2} \cdot \vec{u}$ 、 $\vec{u} = (1/\|s_j - s_i\|) \cdot (s_j - s_i)$  で、 $\vec{u}$  は  $s_i$  から  $s_j$  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。

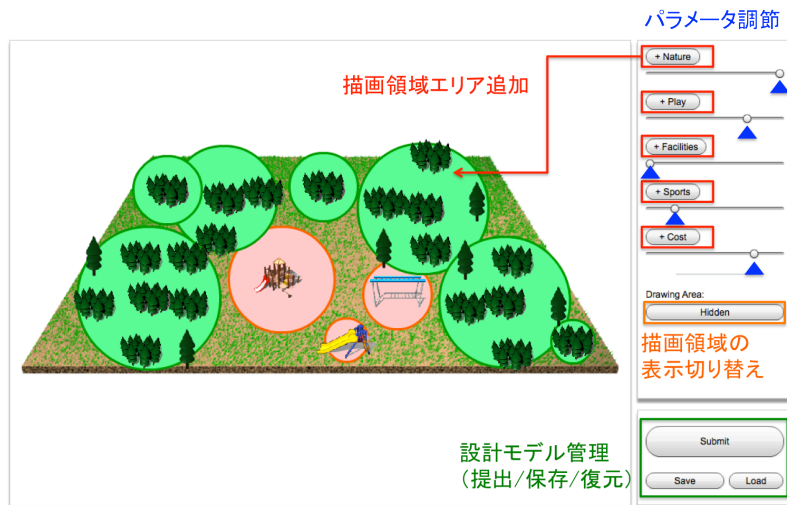


図 6 設計モデルのベース作成システム  
Fig.6 A System for Creating A Base Design

### 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 [Step1]~[Step4] で構成される。

#### [Step1] 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。効用空間抽出機構における設計モデルの自動生成ステップ ([Step3]) の際に、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。

#### [Step2] サンプルポイントの選出

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。

#### [Step3] 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、[Step1] 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。

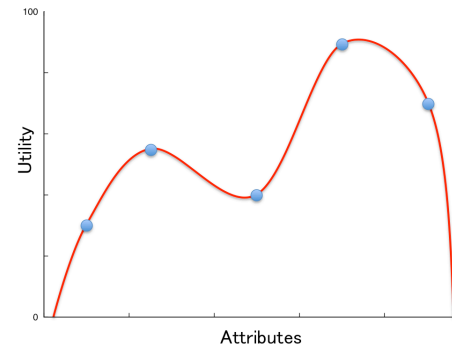


図 7 サンプル点を滑らかに繋ぐことで形成する効用関数のモデル

Fig.7 A Model of An Utility Function Estimated By Connecting Sampling Points Smoothly

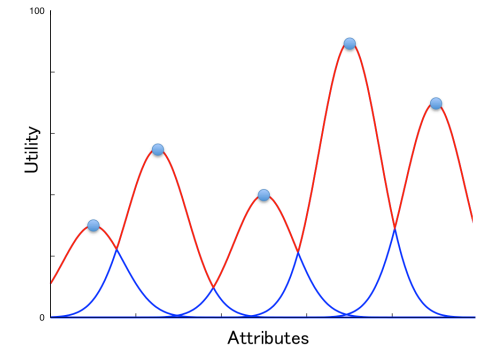


図 8 基本関数の最大値を取ることで形成する効用関数のモデル

Fig.8 A Model of An Utility Function Estimated By Maximum of Basic Functions

#### [Step4] 効用空間の形成

[Step3] 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。本論文で示すシステムでは、 $D_0 = 100$  としている。

以降、[Step2]~[Step4] 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 7 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 8 に示すように、基本関数の最大値を取ることで、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しており、情報が多い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。

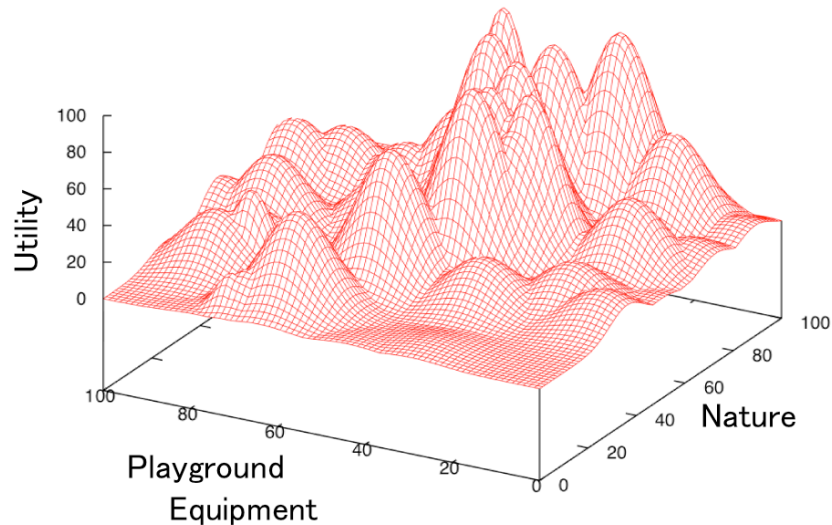


図9 抽出した効用空間  
Fig.9 An Estimated Utility Function

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9に示す。今回の抽出例では、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応することができる。

図9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。

今回は全体にほぼ均一にサンプリング点が分散している。しかし、より効率的な効用空間

抽出・合意形成のためには、ユーザーの効用値が高いことが判明した点付近や、他ユーザーの効用値が高い点のサンプリングを重要視すべきである。

## 5. 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<sup>1)~4)</sup>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。文献5)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。

効用空間の抽出機構として、文献6)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、文献6)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。

文献7), 8)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。

## 6. 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提案した。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。

今後の課題としては、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に、ユーザーからの評価を得るサンプリング点の選出方法を考える必要がある。大規模交渉のフレームワークを構成し、システムを実可動させ、評価実験を行う必要がある。

## 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) 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.
- 6) 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.
- 7) 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.
- 8) 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

## 付 録

### A.1 式 (3) の導出

式 (3) 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.

$$f_i(\vec{s}_j) = f_j(\vec{s}_j)$$

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

$$d_i = \frac{(\vec{s}_j - \vec{s}_i)^2}{\ln \frac{v_i}{v_j}}$$

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.

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

$$v_i > v_j$$

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.

### A.2 式 (4) の導出

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

$$f_i(\vec{c}) = f_j(\vec{c})$$

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

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

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

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

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.

$$d_i = \frac{(\vec{c} - \vec{s}_i)^2}{\frac{k^2}{2} - \ln \frac{v_j}{v_i}} > 0$$

$$\frac{k^2}{2} - \ln \frac{v_j}{v_i} > 0$$

$$\exp\left(\frac{k^2}{2}\right) > \frac{v_j}{v_i}$$

$$\frac{v_j}{v_j \cdot \exp\left(-\frac{k^2}{2}\right)} > \frac{v_j}{v_i}$$

$f_j(\vec{c}) = v_j \cdot \exp(-k^2/2)$  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.