PROLOG を用いた選好集計ルールの生成[1]

Generating Preference Aggregation Rules by Using PROLOG

犬童 健良　　関東学園大学経済学部経営学科
Kenryo Indo　Department of Management, Faculty of Economics, Kanto Gakuen University.
kindo@kanto-gakuen.ac.jp, http://www.us.kanto-gakuen.ac.jp/indo/

Abstract

Preference aggregation is the goal of a planner who wants to integrate preference profile into a complete ordering. The classical impossibility or dictatorial results proved by social choice theorists, which were not computational, have been motivated many researchers of economics, political science, sociology, mathematics, and computer science. This paper presents a PROLOG implementation of the computational modeling of preference aggregation, including resolution principle based proofs for the Gibbard-Satterthwaite theorem, the oligarchy theorem, and the restricted domain of nondictatorial and strategy-proof social choice rules (i.e., the NDSP rules) mainly for three alternatives and two agents.

Keywords: PROLOG, preference aggregation, strategy-proofness, impossibility theorem, mechanism design

1. はじめに

選好集計は合理的選択に立脚し，エージェント個々の考える順序を，グループ全体としての順序に矛盾なく集計する問題である[Arrow 63, Sen 82; Moulin 88; Gaertner 01; 佐伯 80]．またこれに深く関る戦略的操作，すなわち虚偽報告を防ぐ目的では，戦略的ゲームとの境界においてオークションや投票などのメカニズムデザインへの応用を有する．

選好集計はエージェントシステムにおける意思決定と社会的行為のモデリング[Parsons 02]の一つの基礎となる．いうなれば社会的論理プログラミングの前駆としての選好集計および戦略的操作にかんする古典的成果は，K.J. Arrow らによって示された一連の不可能性定理（あるいは可能性定理）であった[Arrow 63, Sen 82, Gibbard 73, Satterthwaite 75]．まず代替案の集合上のエージェントの好みや評価を，完備的，反射的，推移的な 2 項関係（つまり選好順序）として表し，またいくつかのもっともらしい論理的条件（公理）を仮定して，その上で社会的な順序への集計が不可能であること，あるいは必然的に独裁的ないし戦略的操作可能になることが証明された．一方，その緩和，とくに領域制限によって不可能性は覆される[Arrow 63; Kalai 77; Sen 82; Gaertner 01].

選好集計の不可能性定理の論理的条件は，第 3 節で述べるように，3 代替案 2 人のケースに集約されることが知られている．しかし，社会的選択理論家による証明は計算的ではなく，本論文で示されるような不可能性定理のコンピュータプログラムによる証明はあまり知られていない．武隈(2000)は MATHEMETICA を用いてアローの定理を証明した．また将来のコンピュータの用途として，「数値では表現されない抽象概念」，とくに経済学における「抽象概念の思考」に対する適用を示唆している．PROLOG[Starling 94; Clocksin 03]は述語論理に基づく宣言的プログラミングと導出原理（と単一化）による定理自動証明を活用したプログラミング言語である．上記の抽象概念の思考のモデリングに適すると思われる．また筆者の知る限り，往年の投票背理，近年のエージェントシミュレーションを含め，非独裁的あるいは戦略的操作不可能な領域とそのルールの設計は，これまで具体的かつ体系的に計算されていなかった．

本論文の以降の部分では，第 2 節で選好集計を PROLOG によりモデリングし，第 3 節で Gibbard - Satterthwaite の定理および Arrow の定理の変種を計算的に再現する．第 4 節では不可能性を覆す許容的領域と非独裁的かつ戦略的操作不可能である集計ルール（NDSP ルール）および線形順序の最長価値制限領域を実験的に生成する．第 5 節で他の関連研究に触れ，まとめとする．

[1] 作成日：2006.10.16 最終改訂 2006.11.4 本論文はドラフトです．ご意見やコメントを頂けると幸いです．

2. エージェントの選好とその集計

選好集計はよく定式化された社会的相互作用の論理モデリングである．数値的な評価（効用）は扱わないが，効用からつねに選好順序を導くことができるという意味でより一般的である．

2.1. 代替案，選好，プロフィール

社会的選択の基本要素は以下のようである．有限な代替案の集合 A={a, b, …}，エージェント（個人）の集合 N＝｛1, 2, …, n｝とする．選好関係は以下のような性質を満たす代替案のペア比較，あるいはランキングである．

$R_i \subseteq A\times A,\quad i\in N.$

弱順序の場合は，完備的，反射的，かつ推移的である．

（$^\forall$a，b∈A）$aR_ib \vee bR_ia$（完備性）

（$^\forall$a∈A）aR_ia（反射性）

（$^\forall$a，b∈A）aR_ib ＆ $bR_ic \Rightarrow aR_ic$（推移性）

無差別（a I b）は，aR_ib ＆ bR_ia の場合である．全員の選好関係の組 $R_N=(R_1, \cdots, R_n)$をプロフィールと呼ぶ．なお社会全体としての選好は添え字をとって示す．

$R\subseteq A\times A.$

線形順序（強順序）の場合，非対称性を追加する．

（$^\forall$a，b∈A）aR_ib ＆ $bR_ia\Rightarrow a=b$　（非対称性）

```
possible_ranking( r(1), [a,b,c]).
possible_ranking( r(2), [a,c,b]).
possible_ranking( r(3), [b,a,c]).
possible_ranking( r(4), [b,c,a]).
possible_ranking( r(5), [c,a,b]).
possible_ranking( r(6), [c,b,a]).

prefer_x_to_y(A,B, R):-
   possible_ranking( R, O),
   append( _,[A|C],O),
   member(B,C).
```

図1　選好関係．3代替案の線形順序を6つのリストとして表した

その他，準順序はPの部分のみ推移性を，また擬順序はPの部分のみ非循環性を要請する．以上の選好関係を表1にまとめる．PROLOG を用いれば各種の選好関係を自在に生成できるが，とくに線形順序は，図1のようにリストによって表し，またそれぞれシンボル r(K), K=1, …, 6 によって名前付けしておくと便利であろう．今,個人や社会の選好はこれらの6つの順序のうちのいずれかをとるものと仮定する．そうすると，図1は，A={a, b, c}, N={1, 2}の場合の非制限的領域(unrestricted domain)として解釈でき，これを Σ と書く．またこれらのいくつかを削除した場合，つまり Ω⊆Σ を（共通の）許容的領域(admissible domain)と呼ぶ．より正確には，すべての可能な個人選好，プロフィール，社会的選好の集合をそれぞれ $\mathcal{R}_i$，$\mathcal{R}_N$，$\mathcal{R}$ と書くと，制限的領域は $\mathcal{R}_i=\mathcal{R}=\Omega$，$\mathcal{R}_N=\Pi\ \Omega$であり，非制限的領域は Ω＝Σ である．

表1 選好関係のバリエーション

選好関係	完備性	反射性	非循環性	推移性	反対称性
選好関係	Rのみ	○	—	—	—
擬順序	Rのみ	○	Pのみ	—	—
準順序	Rのみ	○	Pのみ	Pのみ	—
弱順序	○	○	○	○	Pのみ
線形順序	○	○	○	○	○

2.2. 選好集計

Arrow 流の社会厚生関数（social welfare function; SWF）は，社会的選好の推移性を要請する．すなわち任意のプロフィール$R_N=(R_1, \cdots, R_n)\in\mathcal{R}^N$に対し，SWF は一つの選好順序 f（$R_N$）∈$\mathcal{R}$ を割り当てる．

SWF　$f: \mathcal{R}_N\rightarrow\mathcal{R}$.

また社会選択ルール（social choice rule; SCR）は任意のプロフィールに対し，一つ以上の代替案を割り当てる．

SCR　$g: \mathcal{R}_N\rightarrow 2^A-\{\phi\}$.

社会選択関数（SCF）は SCR の一価の場合である．社会的決定関数（social decision function; SDF）は SWF の推移性の代わりに非循環性を要請し，その極大要素を抽出する SCR である[Sen 82, p.166].

選好集計の代表的な方法として，単純多数決原理（Condorcet ルール）は各個人 i に xR_iy や yR_ix を表明させ,票数により xRy または yRx を定める（以降，たんに多数決と言う）．多数決は推移性を満たすとは限らず，また勝者が存在しない場合が生じる（第4節参照）．図2にパレート条件（後述する条件 **P**）および戦略的操作不可能性（同じく条件 **S**）をそれぞれ満たす SCF を生成するコードを示す．SWF の場合は，値域を社会的順序に替えた同様の再帰的コードになる．

```
auto_scf( [],[], _).

auto_scf( [R-> X | H], [R|Q], pareto):-
  auto_scf( H,Q,pareto),
  alternative(X),
  is_Pareto_efficient_at_profile( R->X).

auto_scf( [R-> X | H], [R|Q], sp):-
  auto_scf( H,Q,sp),
  alternative(X),
  \+ is_manipulable_at_profile( H, R->X,_,_),
  \+ (
    member(V, H),
    is_manipulable_at_profile( [R->X|H], V,R->X,_)
  ).
```

図2　選好集計ルール（SCF）．第2引数に定義域のプロフィール集合を代入して用いる．

```
citizens_sovereignty(Scf):-
  forall(
    alternative(A),
    member( _->A, Scf)
  ).

is_Pareto_efficient_at_profile( (R1, R2)->Y):-
  \+ (
    prefer_x_to_y( X, Y, R1),
    prefer_x_to_y( X, Y, R2)
  ).
is_manipulable_at_profile( Scf,(R,R2)->X,(Q,R2)->Y,1):-
   member( (Q,R2)->Y, Scf),
   prefer_x_to_y( Y, X, R).
is_manipulable_at_profile( Scf,(R1,R)->X,(R1,Q)->Y,2):-
   member( (R1,Q)->Y, Scf),
   prefer_x_to_y( Y, X, R).
```

図3 市民主権，パレート原理，操作不可能性

2.3. 選好集計の制約条件

選好集計に課される制約条件をいくつか挙げる．

(条件 **U**) 非制限的領域．Ω＝Σ．定義域が制限されない．

(条件 **C**) 市民主権性．値域が制限されない．

(条件 **P**) 弱パレート原理．全員が強く一致した選好であるならば，社会も同じように選ぶ．

(条件 **D**) 独裁性．社会の決定がつねにある一人の好むもの一致する．

(条件 **I**) 無関係な代替案からの独立性(IIA)．SWF における社会の選好は当該の代替案についての個人選好だけによって決定される．

(条件 **S**) SCF の操作不可能性．誰も選好を偽ることによって己れの有利に導くことはできない．

(条件 **M**) SCF の単調性．あるプロフィールで選ばれた案は，誰かにとって順位を下げない限り，別のプロフィールでも選ばれる．

なお **M** は **P** を含意し，**S** と同等である[Muller 77]．

３つの条件，**C** と **P** と **S** について，それぞれのコードを図 3 に示した．また図 4 は **U** と **P** および **C** を満たす SCF を生成し，**S** への違反を検査する実験を示している．

3. 不可能性定理の自動証明

不可能性定理の論理的条件は 3 代替案 2 人のケースに集約される．しかしこの最も基本的な 36 通りのプロフィールに対して，SWF の候補は 6^{36}＝約 10^{28} 通り，SCF では 3^{36} 通りある．素朴な生成検査法では，1 億分の 1 秒で１つの SWF を処理するとして，Arrow の不可能性定理の証明に 32.7 兆年かかることになる．この計算量の問題は，図 2 や図 3 に示す SCF や SWF の生成実験において制約条件(**I** や **S**)を再帰的に累積することよって解決される．

また本論文では基本的に線形順序を仮定するが，弱順序としても以下の諸定理は基本的に変わらない．また一部の実験では，準順序を用いる．

```
?- create_auto_scf_with_check.
制約条件のリストを入力してください.
|: [pareto, cs].
領域を制限しますか.
|: n.
社会的選択関数を生成します.
|: y.

col=r(#) [1, 2, 3, 4, 5, 6]
---------------------------
row=r(1) [a, a, a, a, a, a]
row=r(2) [a, a, a, a, a, a]
row=r(3) [a, a, b, b, a, b]
row=r(4) [a, a, b, b, b, b]
row=r(5) [a, a, a, b, c, c]
row=r(6) [a, a, b, b, c, c]

検証する条件のリストを入力してください.
|: [sp].

戦略的操作可能かどうか調べますか.
|: y.

エージェント:1はプロフィール: (r(3), r(5))において案:aから案:bに戦略的操作できます.
エージェント:2はプロフィール: (r(4), r(5))において案:bから案:aに戦略的操作できます.
エージェント:2はプロフィール: (r(4), r(5))において案:bから案:aに戦略的操作できます.
エージェント:2はプロフィール: (r(5), r(3))において案:aから案:bに戦略的操作できます.
エージェント:1はプロフィール: (r(5), r(4))において案:bから案:aに戦略的操作できます.
エージェント:1はプロフィール: (r(5), r(4))において案:bから案:aに戦略的操作できます.
Yes
?-
```

図4 制約を満たすSCFを生成し，事後チェックする

3.1. 非制限的領域における不可能性定理

SWF や S 条件 **I** や条件 **S** は，これらはそれ自体で値を決定しないが，SCF / SWF に対する制約としてはたいへん強力であり，独裁性を論理的に帰結する．

命題 A．[Arrow 63] ３つ以上の代替案があるとき，任意の SWF について，**U**&**P**&**I**⇒**D**.

命題 G．[Gibbard 73, Satterthwaite 75] ３つ以上の代替案があるとき，任意の SCF について，**U**&**C**&**S**⇒**D**.

次節で述べるように両定理は論理的に同値であり，また PROLOG による実験を通じて，これを確認できる．

図 5 の左側は命題 *G* の自動証明の結果を示したものである．例えば左の各表における行と列は，2 人の選好の組を表す．それぞれ各表上側は列(2)の、また下側は行(1)の最良選択肢から成り立っているのが分かる．

なお命題 *A* における条件 **P** を条件 **C** に緩和した場合，反独裁者が新たに発生する．

(条件 **A**) 反独裁性．社会の決定がつねにある一人の好まない選択になっている．

命題 W．[Wilson 72] ３つ以上の代替案があるとき，任意の SWF について，**C**&**U**&**I**⇒**D**∨**A**.

命題 *W* は該当条件を外せば命題 *A* と同様に証明できる

3.2. 値域の緩和と寡頭性

SWF は推移性を要請するが，値域が準推移的順序である SDF に緩和することによって可能性定理が得られる．しかし限られた条件内で独裁性がグループ支配（寡頭性）に変わるにすぎない[Sen 82, pp. 166-167; Moulin 88, p. 292]．またその必要十分条件は後述する価値制限と循環的依存性である[Salles 76; Gaertner 01, pp. 44--45]．

(条件 **O**) 寡頭性．つねにある一つのグループについて，各人が拒否権を有する．すなわち，その内 1 人でも x を y より強く好めば，社会はその反対の関係を選べない．

```
?- auto_scf( F,sp),
citizens_sovereignty(F),
show_scf(F),fail.

col=r(#) [1, 2, 3, 4, 5, 6]
---------------------------
row=r(1) [a, a, b, b, c, c]
row=r(2) [a, a, b, b, c, c]
row=r(3) [a, a, b, b, c, c]
row=r(4) [a, a, b, b, c, c]
row=r(5) [a, a, b, b, c, c]
row=r(6) [a, a, b, b, c, c]
col=r(#) [1, 2, 3, 4, 5, 6]
---------------------------
row=r(1) [a, a, a, a, a, a]
row=r(2) [a, a, a, a, a, a]
row=r(3) [b, b, b, b, b, b]
row=r(4) [b, b, b, b, b, b]
row=r(5) [c, c, c, c, c, c]
row=r(6) [c, c, c, c, c, c]

No
?-
```

```
?- r_x(K,A,B,C,E),
nl,write([K]:A),fail.

[1]:[+, +, +]
[2]:[0, +, +]
[3]:[-, +, +]
[5]:[0, 0, +]
[6]:[-, 0, +]
[9]:[-, -, +]
[10]:[+, +, 0]
[11]:[0, +, 0]
[13]:[+, 0, 0]
[14]:[0, 0, 0]
[15]:[-, 0, 0]
[17]:[0, -, 0]
[18]:[-, -, 0]
[19]:[+, +, -]
[22]:[+, 0, -]
[23]:[0, 0, -]
[25]:[+, -, -]
[26]:[0, -, -]
[27]:[-, -, -]

No
?-
```

```
?- show_swf_0_max([p,i],F),nl,member(P:G,F),
is_decisive_swf(P,F,J),write(d(P:J)),nl,fail.

[+, +, +]:1:[[x], [y], [y], [x], [z], [z]]
[-, +, +]:3:[[x], [y], [y], [x], [z], [z]]
[-, -, +]:9:[[x], [y], [y], [x], [z], [z]]
[+, +, -]:19:[[x], [y], [y], [x], [z], [z]]
[+, -, -]:25:[[x], [y], [y], [x], [z], [z]]
[-, -, -]:27:[[x], [y], [y], [x], [z], [z]]
d((x, y):2)
d((x, z):2)
d((y, z):2)

[+, +, +]:1:[[x], [x, y], [x, y], [x], [x, z], [x, y, z]]
[-, +, +]:3:[[x, y], [y], [y], [x, y], [x, y, z], [y, z]]
[-, -, +]:9:[[x, y], [y], [y], [x, y, z], [y, z], [y, z]]
[+, +, -]:19:[[x], [x, y], [x, y, z], [x], [x, z], [x, z]]
[+, -, -]:25:[[x, z], [x, y, z], [y, z], [x, z], [z], [z]]
[-, -, -]:27:[[x, y, z], [y, z], [y, z], [x, z], [z], [z]]

[+, +, +]:1:[[x], [x], [x], [x], [x], [x]]
[-, +, +]:3:[[y], [y], [y], [y], [y], [y]]
[-, -, +]:9:[[y], [y], [y], [y], [y], [y]]
[+, +, -]:19:[[x], [x], [x], [x], [x], [x]]
[+, -, -]:25:[[z], [z], [z], [z], [z], [z]]
[-, -, -]:27:[[z], [z], [z], [z], [z], [z]]
d((x, y):1)
d((x, z):1)
d((y, z):1)

No
?-
```

図5　不可能性定理の自動証明. Gibbard-Satterthwaite定理(左). 改良されたコードにおける選好モデル（中）. 値域を準推移的順序とした場合のArrowの定理のバージョン（右）. ただし右図はSDFを示している. 図1の線形順序の番号1～6はそれぞれ, 中と右の2つの図では1, 19, 3, 9, 25, 27に対応する. また＋や－の記号はそれぞれ(x,y), (x,z), (y,z)の選好関係を示す.

また，また全員一致で強く好めば社会もそうである．

(条件 **SD**) 循環的依存性. 任意の 3 代替案 a, b, c∈A について, 線形順序をもつ個人がいないか, または(a, b), (b, c) ∈P_iであって,かつ以下のいずれにも該当しないこと.

[（∃j∈N） c P_j a, b P_j c ＆（∃k∈N） c R_k a, a R_k b]∨

[（∃j’∈N） c $R_{j'}$a, b $R_{j'}$c ＆（∃k’∈N） c $P_{k'}$a, a $P_{k'}$b]∨

[（∃j”∈N） c $P_{j''}$a , b $I_{j''}$c ＆（∃k” ∈N） c$I_{k''}$a, a $P_{k''}$b].

ただし k と j’ は 3 代替案のすべてに無差別でない(concerned agents)とする．

命題 S. [Sen 82 ; Salles 76]　任意の SDF が準推移的順序を帰結する⇔**V**（後述）＆**S D**⇒（**U**&**P**&**I**⇒**O**）.

図 5 の右側は若干改良を加えたプログラムを用いて命題 *S* を実験した結果である．各人の各代替案ペアに対する指令性(decisivenes)についても同時に調べてある. 上と下はそれぞれ列（2）と行(1)の独裁である．また中はグループ全体の寡頭であり，その社会的順序は，ちょうど両者の選好パタンを加算したものになっている．

一方，弱順序では独裁的ルールが多数存在するため，残念ながら，図 5 のような鮮やかな結果は得られない．しかし次節で述べるように，不可能性の結果の必要十分条件が知られており，それに基づき可能的結果を得る条件 **U** を緩和する方法を具体的に計算できる．

4．非独裁的制限領域の設計

不可能性を回避するため選好を制限する方法は，分解可能性[Kalai 77]や価値制限[Inada 69; Sen 69]などとして解明されている．しかしその具体的な計算例は示されておらず計算的なメカニズム設計としては不十分であった．本節では PROLOG を用いて 3 代替案の非独裁的かつ戦略的操作不可能な SCF(NDSP ルール)が存在する許容的領域を具体的に計算し，また 2 人の場合のルールと 3 人の多数決の場合を吟味する．

4.1.　制限的領域における可能性定理

命題 *A* における条件 **U** は，3 代替案の非制限的領域の埋め込まれた制限的領域に緩められることや，2 代替案の場合や単峰性領域については，多数決 SWF の可能性定理は[Arrow 63]が示している．また多数決がうまくいかなくても，分解可能性（条件 **B**）を満たせば，またそのときに限り非独裁的かつ操作不可能な意思決定ルールを設計できる．

命題 K. [Kalai 77]　許容的領域Ω⊆Σが以下の条件 **B** を満たす⇔n≧2 のとき，ある SWF が命題 *A* の条件下で¬**D** を満たす⇔n≧2 のとき，ある SCF が命題 *G* の条件下で¬**D** を満たす.

(条件 **B**）分解可能性．許容的領域Ω⊆Σにおいて，自明なペア集合（すなわち対立する選好を持たない代替案ペアの集合）に，自明でない相異なる代替案ペアを一つ以上追加して得られるある集合が作れて，しかしペア全体にはならず，かつ以下の条件 **CDI** を満たすこと．

(条件 **CDI**）代替案ペアの部分集合 R が指令性含意において閉じていること．すなわち，

（∀ (x,y), (x,z) ∈非自明ペア集合）

[x P y P z & y Q z Q x， P，Q∈Ω

⇒（(x,y) ∈R⇒(x,z)∈R）&（(y,x) ∈R⇒(z,x)∈R）]

& [　x P y P z， P∈Ω

⇒（(x,y) ∈R&(y,z)∈R⇒(x,z)∈R)）

```
violates_DI(V,(X,Y,Z),L):-
  nontrivial_ordered_pair((X,Y)),
  nontrivial_ordered_pair((X,Z)),
  (violates_DI(I,(X,Y,Z),C,L)->V=(I,C);fail).

violates_DI(1, (X,Y,Z),[(R1,R2)],L):-
  possible_ranking(R1,[X,Y,Z]),
  possible_ranking(R2,[Y,Z,X]),
  violates_DI_1(_, (X,Y,Z),L).

violates_DI_1( a, (X,Y,Z),L):-
  member( (X,Y), L),
  \+ member( (X,Z), L).

violates_DI_1( b, (X,Y,Z),L):-
  member( (Z,X), L),
  \+ member( (Y,X), L).

is_cdi_pairs(D,Tb):-
  findall(B,distinct_ordered_pair(B),Tb),
  ntr_projection(D,Tb,_),
  \+ violates_DI(_,_,D),
  is_nontrivial_and_consistent_cdi(D,Tb).
```

```
col=r(#) [1, 2, 3, 4, 5]
---------------------------
row=r(1) [a, a, b, b, a]
row=r(2) [a, a, b, b, a]
row=r(3) [b, b, b, b, b]
row=r(4) [b, b, b, b, b]
row=r(5) [a, a, b, b, c]cdi

col=r(#) [1, 2, 3, 4, 6]
---------------------------
row=r(1) [a, a, a, a, a]
row=r(2) [a, a, a, a, a]
row=r(3) [a, a, b, b, b]
row=r(4) [a, a, b, b, b]
row=r(6) [a, a, b, b, c]cdi

col=r(#) [1, 2, 3, 5, 6]
---------------------------
row=r(1) [a, a, a, a, a]
row=r(2) [a, a, a, a, a]
row=r(3) [a, a, b, a, b]
row=r(5) [a, a, a, c, c]
row=r(6) [a, a, b, c, c]cdi
```

```
col=r(#) [1, 2, 4, 5, 6]
---------------------------
row=r(1) [a, a, a, a, a]
row=r(2) [a, a, a, a, a]
row=r(4) [a, a, b, c, c]
row=r(5) [a, a, c, c, c]
row=r(6) [a, a, c, c, c]cdi

col=r(#) [1, 3, 4, 5, 6]
---------------------------
row=r(1) [a, b, b, a, b]
row=r(3) [b, b, b, b, b]
row=r(4) [b, b, b, b, b]
row=r(5) [a, b, b, c, c]
row=r(6) [b, b, b, c, c]cdi

col=r(#) [2, 3, 4, 5, 6]
---------------------------
row=r(2) [a, b, b, c, c]
row=r(3) [b, b, b, b, b]
row=r(4) [b, b, b, b, b]
row=r(5) [c, b, b, c, c]
row=r(6) [c, b, b, c, c]cdi
```

図6 長さ5の許容的領域での非独裁かつ操作不可能なSCFを生成し，分解可能性を検証する．左は分解可能性の判定に使う条件CDIのコードの一部

& ((z,x) ∈R⇒ (y,x)∈R ∨ (z,y)∈R)].

図 6 は，**C**&**S**&¬**D** を満たす SCF をもつ長さ 5 の領域を生成し，また分解可能性を検証する実験結果を，条件 **CDI** のコードの一部と共に示したものである．ちなみにこの実験では以下のようにゴール節を入力した．

?- auto_restricted_domain(L),length(L,5),
¥+ ¥+ (auto_scf(F,sp),citizens_sovereignty(F),
is_non_dictatorial_scf(F),nl,show_scf(F)),
(¥+ ¥+ is_cdi_pairs(_,_)->write(cdi);true),fail.

なお長さの制限をはずせば命題 K の必要性についての証明となる．また十分性についても同様に実験で確かめられる．

分解可能性は人数によらない領域設計のツールであり，理論上は任意のサイズの代替案の場合に使える．しかし 2 人 3 代替案のとき，3 条件 **C**，**S**，¬**D** を同時に満たす選好集計ルール（領域と SCF の組合せ）は，290 通りあることが実験により確かめられる．これらの非独裁的かつ戦略的操作不可能な SCF とその許容的領域のペアを，NDSP ルール(nondictatorial and strategy-proof rules)と呼ぶことにする．

4.2. 穏当性と埋め込み極大性

NDSP ルールを絞り込むため以下の２条件，穏当性と極大性を提案する．

（穏当性） $x\,P\,y\,P\,z\ \&\ z\,Q\,x\,Q\,y \Rightarrow x\,S\,z$

（極大性）NDSP ルール $f: \Omega \to A$，$\Omega \subseteq \Sigma$ に対して，それを一部として含む NDSP ルール $g: \Theta \to A$，$\Omega \subseteq \Theta \subseteq \Sigma$，$f = g \mid \Omega$ が存在しない．

直観的に言うと，穏当性の解釈は，「皆がベストかそれに次ぐと同意できるのに，なぜ私にとって最悪の案を選ぶのでしょう」といったクレイムがだれからもつかないことである．穏当性は条件 **P** の緩和であるが，これが有効なのは領域が制限されているためである．

また穏当性の条件が適用されるプロフィールは，少なくとも２人のエージェントは自分自身が自由に決められる代替案ペアをもつべしとする最小自由主義の条件とパレート条件 **P** の両立を不可能とする．これはいわゆる自由主義背理と呼ばれる[Sen 82, chapter 3]．穏当性は，したがってその一つの解決法を示している．

２人３代替案の場合，穏当かつ極大な選好集計ルールは長さ４の領域に６つと，長さ３の領域に２つ，計８つある．後者の２つはラテン方格（次節参照）であり，ともに多数決に一致する．長さ４の領域はいずれも第２位にならない代替案を持ち，したがって価値制限（次節参照）を満たすが，その NDSP ルールは多数決ではない．なお長さ５の領域は，図５中の ６つを含む 24 の NDSP ルールすべてが穏当性に違反する．

4.3. 価値制限

多数決によって社会的選好を矛盾なく定めるための必要十分条件は，1960 年代の終わりに[Inada 69]と[Sen 69]によって解明された．

(条件 **T**) 多数決による社会の選好 R^M が推移的である．

(条件 **W**) コンドルセ勝者．（$^\forall y \in A$）$x\,R^M\,y$ なる $x \in A$.

(条件 **V**) 価値制限．任意の３代替案について，ある代替案がその中のある順位をとらない場合がある．すなわち，

$(^{\forall}x, y, z \in A)$
$[\ (^{\forall}i \in N)\ aR_i b \vee aR_i c] \vee$
$[\ (^{\forall}i \in N)\ bR_i a \vee cR_i a] \vee$
$[(^{\forall}i \in N)\ (aR_i b\ \ \&\ \ aR_i c) \vee (bR_i a\ \ \&\ \ cR_i a)].$

ただしこの条件はエージェント i が 3 代替案のすべてに無差別である場合は適用除外される.

命題 M. [Sen 69, Inada 69]　個人の選好が線形順序とする.このとき，条件 **V**⇔条件 **W**．また人数が奇数ならば，条件 **V**⇔条件 **T**.

すなわち奇数人数のとき，条件 **V** に違反すると，社会的選好が非推移的となるが，そうでない場合には，SWF になる．また社会的選好が非推移的であってもコンドルセ勝者を選べる場合があり，線形順序では条件 **V** がその必要十分条件である．なお Sen の定義では 3 代替案すべてが無差別でないエージェントの選好にかんしてのみ要求されるが，線形順序のため省いた.

図 7 に条件 **V** に対応する PROLOG コードの一例を示す. 3 人の社会で条件 **B** を満たす領域とその多数決 NDSP ルールを生成したところ，長さ 4 と長さ 3 の各領域でそれぞれ 6 つ，計 12 パタンが見つかった．図 8 に長さ 4 の結果を示す．表の各行は一人目の選好，各列は二人目と 3 人目の選好の組に対応する．3 種類の価値制限のうち, 3 位抜け(not worst)すなわち単峰性領域と 2 位抜け(not medium)が 3 つずつ，計 6 つあり，1 位抜け(not best)すなわち単谷性領域はない．**C** をはずして実験しなおすと，長さ 4 で 9 つ，長さ 3 で 12，長さ 2 で 3 つ，計 24 になる．長さ 4 で増えた 3 つは 1 位抜けである．また，条件 **B** を満たさない領域には多数決に一致する NDSP ルールはない．しかし，条件 **C** を外すと，長さ 3 の 6 領域と長さ 2 の 9 領域で見つかった.

```
value_restriction( C):-
  member(C,[not_best, not_medium, not_worst]),
  forall(
    triple_of_alternatives((X,Y,Z)),
    (
     alternative(W),
     member(W,[X,Y,Z]),
     forall(
       profile_of_rankings( RL),
       is_agreed_on( C, W,(X,Y,Z), RL)
     )
    )
  ).

is_agreed_on( C, W, (X,Y,Z), RL):-
  member((K,C),[
    (1, not_best),
    (2, not_medium),
    (3, not_worst)
  ]),
  member(W, [X,Y,Z]),
  \+ (
    member(R, RL),
    rank_in_list( K, W, [X,Y,Z], R)
  ).
```

図7　価値制限

4.4.　価値制限を満たす最長領域

ところで条件 **V** を満たす最長の線形順序集合（最大非

```
scf:
  #r2: 111111222222333333444444555555666666
  #r3: 123456123456123456123456123456123456
r1=r(1)aaa-a-aaa-a-aab-a-------aaa-c-------
r1=r(2)aaa-a-aaa-a-aab-a-------aaa-c-------
r1=r(3)aab-a-aab-a-bbb-b-------aab-c-------
r1=r(4)------------------------------------
r1=r(5)aaa-c-aaa-c-aab-c-------ccc-c-------
r1=r(6)------------------------------------
domain:[r(1), r(2), r(3), r(5)];not_worst
scf:
  #r2: 111111222222333333444444555555666666
  #r3: 123456123456123456123456123456123456
r1=r(1)aa-a-aaa-a-a------aa-b-b------aa-b-c
r1=r(2)aa-a-aaa-a-a------aa-b-c------aa-c-c
r1=r(3)------------------------------------
r1=r(4)aa-b-baa-b-c------bb-b-b------bc-b-c
r1=r(5)------------------------------------
r1=r(6)aa-b-caa-c-c------bc-b-c------cc-c-c
domain:[r(1), r(2), r(4), r(6)];not_medium
scf:
  #r2: 111111222222333333444444555555666666
  #r3: 123456123456123456123456123456123456
r1=r(1)a-aa-a------a-bb-ba-bb-b------a-bb-c
r1=r(2)------------------------------------
r1=r(3)a-bb-b------b-bb-bb-bb-b------b-bb-c
r1=r(4)a-bb-b------b-bb-bb-bb-b------b-bb-c
r1=r(5)------------------------------------
r1=r(6)a-bb-c------b-bb-cb-bb-c------c-cc-c
domain:[r(1), r(3), r(4), r(6)];not_worst
```

```
scf:
  #r2: 111111222222333333444444555555666666
  #r3: 123456123456123456123456123456123456
r1=r(1)a-a-aa------a-b-ab------a-a-cca-b-cc
r1=r(2)------------------------------------
r1=r(3)a-b-ab------b-b-bb------a-b-ccb-b-cc
r1=r(4)------------------------------------
r1=r(5)a-a-cc------a-b-cc------c-c-ccc-c-cc
r1=r(6)a-b-cc------b-b-cc------c-c-ccc-c-cc
domain:[r(1), r(3), r(5), r(6)];not_medium
scf:
  #r2: 111111222222333333444444555555666666
  #r3: 123456123456123456123456123456123456
r1=r(1)------------------------------------
r1=r(2)-------aaaa--abba--abbc--aacc-------
r1=r(3)-------abba--bbbb--bbbb--abbc-------
r1=r(4)-------abbc--bbbb--bbbb--cbbc-------
r1=r(5)-------aacc--abbc--cbbc--cccc-------
r1=r(6)------------------------------------
domain:[r(2), r(3), r(4), r(5)];not_medium
scf:
  #r2: 111111222222333333444444555555666666
  #r3: 123456123456123456123456123456123456
r1=r(1)------------------------------------
r1=r(2)-------a-aaa-------a-bcc-a-ccc-a-ccc
r1=r(3)------------------------------------
r1=r(4)-------a-bcc-------b-bbb-c-bcc-c-bcc
r1=r(5)-------a-ccc-------c-bcc-c-ccc-c-ccc
r1=r(6)-------a-ccc-------c-bcc-c-ccc-c-ccc
domain:[r(2), r(4), r(5), r(6)];not_worst
```

図8　3人の社会で多数決SCFが存在する分解可能性を満たす長さ4の領域とその価値制限

循環的集合）を求める問題は組合せ数学における難問の一つとして知られる[Fishburn 04]. 現在まで確実に知られているのは 7 代替未満のケースである（表 2 参照）. PROLOG を用いた実験により，6 代替案以下の既知の最長領域を確認できた．しかし再帰的な制約累積だけでは 6 代替案以上の最長領域生成は困難であった．Fishburn の交代スキーマ(alternating schema)を組み込み，まず 6 代替案までは成功した．7 代替案ではその下界（表 2 最下行参照）を用いて初期解として見つけ，順序を追加する方法を用いた．図 9 にその結果を示す.

表 2 線形順序に対する価値制限領域の最大長

代替案数（M）	最大長
3	4
4	9
5	20
6	45
7	≧100
8 以上	$\geqq 3\times 2^{(M-2)}-4$

5. まとめ

本論文では非常に小さい領域ではあるが，従来机上で証明されていた社会的選択の基本的定理を，導出原理を用いて自動証明することに成功した．また許容的領域とその非独裁的かつ戦略的操作不可能な社会選択ルール（NDSP ルール）を実験によって生成し，穏当性や極大性の条件を吟味した．筆者の知る限り，この結果は，従来，具体的に計算されていなかった．本論文で紹介した実験はノート PC (Windows XP Professional SP2, Celeron 1.40GHz, 1.24GB RAM) 上で，SWI-PROLOG 5.0.9 を用いたが，一部（4. 4 節）の実験を除き，数秒以内で処理される．素朴なシミュレーション実験では，2 人 3 代替案の場合ですら，事実上，不可能であるが，図 2 に示す PROLOG コードの再帰において，選好集約の局所的制約(図 3)が累積されているからである.

ところで従来，選好集計のシミュレーション研究は数値的なものが中心だった．また投票背理の確率のように

1	[1, 2, 3, 4, 5, 6, 7]	26	[2, 4, 1, 6, 7, 3, 5]	51	[4, 6, 2, 1, 3, 5, 7]	76	[6, 4, 7, 2, 5, 1, 3]
2	[1, 2, 3, 4, 6, 5, 7]	27	[2, 4, 1, 6, 7, 5, 3]	52	[4, 6, 2, 1, 3, 7, 5]	77	[6, 4, 7, 2, 5, 3, 1]
3	[1, 2, 3, 4, 6, 7, 5]	28	[2, 4, 6, 1, 3, 5, 7]	53	[4, 6, 2, 1, 7, 3, 5]	78	[6, 4, 7, 5, 2, 1, 3]
4	[1, 2, 4, 3, 5, 6, 7]	29	[2, 4, 6, 1, 3, 7, 5]	54	[4, 6, 2, 1, 7, 5, 3]	79	[6, 4, 7, 5, 2, 3, 1]
5	[1, 2, 4, 3, 6, 5, 7]	30	[2, 4, 6, 1, 7, 3, 5]	55	[4, 6, 2, 7, 1, 3, 5]	80	[6, 4, 7, 5, 3, 2, 1]
6	[1, 2, 4, 3, 6, 7, 5]	31	[2, 4, 6, 1, 7, 5, 3]	56	[4, 6, 2, 7, 1, 5, 3]	81	[6, 7, 4, 2, 1, 3, 5]
7	[1, 2, 4, 6, 3, 5, 7]	32	[2, 4, 6, 7, 1, 3, 5]	57	[4, 6, 2, 7, 5, 1, 3]	82	[6, 7, 4, 2, 1, 5, 3]
8	[1, 2, 4, 6, 3, 7, 5]	33	[2, 4, 6, 7, 1, 5, 3]	58	[4, 6, 2, 7, 5, 3, 1]	83	[6, 7, 4, 2, 5, 1, 3]
9	[1, 2, 4, 6, 7, 3, 5]	34	[2, 4, 6, 7, 5, 1, 3]	59	[4, 6, 7, 2, 1, 3, 5]	84	[6, 7, 4, 2, 5, 3, 1]
10	[1, 2, 4, 6, 7, 5, 3]	35	[2, 4, 6, 7, 5, 3, 1]	60	[4, 6, 7, 2, 1, 5, 3]	85	[6, 7, 4, 5, 2, 1, 3]
11	[2, 1, 3, 4, 5, 6, 7]	36	[4, 2, 1, 3, 5, 6, 7]	61	[4, 6, 7, 2, 5, 1, 3]	86	[6, 7, 4, 5, 2, 3, 1]
12	[2, 1, 3, 4, 6, 5, 7]	37	[4, 2, 1, 3, 6, 5, 7]	62	[4, 6, 7, 2, 5, 3, 1]	87	[6, 7, 4, 5, 3, 2, 1]
13	[2, 1, 3, 4, 6, 7, 5]	38	[4, 2, 1, 3, 6, 7, 5]	63	[4, 6, 7, 5, 2, 1, 3]	88	[6, 7, 5, 4, 2, 1, 3]
14	[2, 1, 4, 3, 5, 6, 7]	39	[4, 2, 1, 6, 3, 5, 7]	64	[4, 6, 7, 5, 2, 3, 1]	89	[6, 7, 5, 4, 2, 3, 1]
15	[2, 1, 4, 3, 6, 5, 7]	40	[4, 2, 1, 6, 3, 7, 5]	65	[4, 6, 7, 5, 3, 2, 1]	90	[6, 7, 5, 4, 3, 2, 1]
16	[2, 1, 4, 3, 6, 7, 5]	41	[4, 2, 1, 6, 7, 3, 5]	66	[6, 4, 2, 1, 3, 5, 7]	91	[7, 6, 4, 2, 1, 3, 5]
17	[2, 1, 4, 6, 3, 5, 7]	42	[4, 2, 1, 6, 7, 5, 3]	67	[6, 4, 2, 1, 3, 7, 5]	92	[7, 6, 4, 2, 1, 5, 3]
18	[2, 1, 4, 6, 3, 7, 5]	43	[4, 2, 6, 1, 3, 5, 7]	68	[6, 4, 2, 1, 7, 3, 5]	93	[7, 6, 4, 2, 5, 1, 3]
19	[2, 1, 4, 6, 7, 3, 5]	44	[4, 2, 6, 1, 3, 7, 5]	69	[6, 4, 2, 1, 7, 5, 3]	94	[7, 6, 4, 2, 5, 3, 1]
20	[2, 1, 4, 6, 7, 5, 3]	45	[4, 2, 6, 1, 7, 3, 5]	70	[6, 4, 2, 7, 1, 3, 5]	95	[7, 6, 4, 5, 2, 1, 3]
21	[2, 4, 1, 3, 5, 6, 7]	46	[4, 2, 6, 1, 7, 5, 3]	71	[6, 4, 2, 7, 1, 5, 3]	96	[7, 6, 4, 5, 2, 3, 1]
22	[2, 4, 1, 3, 6, 5, 7]	47	[4, 2, 6, 7, 1, 3, 5]	72	[6, 4, 2, 7, 5, 1, 3]	97	[7, 6, 4, 5, 3, 2, 1]
23	[2, 4, 1, 3, 6, 7, 5]	48	[4, 2, 6, 7, 1, 5, 3]	73	[6, 4, 2, 7, 5, 3, 1]	98	[7, 6, 5, 4, 2, 1, 3]
24	[2, 4, 1, 6, 3, 5, 7]	49	[4, 2, 6, 7, 5, 1, 3]	74	[6, 4, 7, 2, 1, 3, 5]	99	[7, 6, 5, 4, 2, 3, 1]
25	[2, 4, 1, 6, 3, 7, 5]	50	[4, 2, 6, 7, 5, 3, 1]	75	[6, 4, 7, 2, 1, 5, 3]	100	[7, 6, 5, 4, 3, 2, 1]

図9 PROLOGで生成した7代替案のときの非循環的な領域（線形順序の価値制限領域）の一つ

近似式が改良されるにつれ,その意義は薄らいでいく(例えば [佐伯 80; Gaertner 02]を参照).一方，社会選択理論家による定理は計算的でないまま残されていた．例えば 4 節で述べた価値表現は，高々不可能性が覆るための条件にすぎず，具体的にどのようなルールがいくつあるのかを明らかにするものでない（投票背理の確率計算ではそれゆえ別の指標が用いられている）．また，エージェントシステムと合理的選択の交流は，不確実性下の意思決定の側面[Parsons 02]で先行し，社会的選択の抽象的かつ決定論的モデリングは，エージェントの結託形成や集合的推論の観点から，最近になって研究成果が報告され始めている．エージェントアプローチ[Axelrod 06, Epstein 05]では AI 手法の社会科学的応用を開拓し，それを通じて，AI（認知科学）や経済学（意思決定科学）などの諸分野の境界を越えた新理論･新手法の開発が期待される．本論文は公理的アプローチとエージェントアプローチの両立可能性を示しており，またそれぞれ前者に対する計算化･可視化と後者に対する厳密な基礎付けを提供するという点で両分野に貢献しうる．もちろんその評価は今後の発展次第と思われる．最後に，自由，権利，公平などの制度設計をめぐる応用では，選好集計のモデリングはより一般化されることになるが，今後の研究の目標としたい．

参考文献

[Arrow 63] Arrow, K.: *Social Choice and Individual Values*, Yale University Press (1963)

[Axerlod 06] Axelrod, R. and Tesfatsion, L.: "A Guide for Newcomers to Agent-Based Modeling in the Social Sciences," In L. Tesfatsion and K. L. Judd (eds.), *Handbook of Computational Economics, Vol. 2: Agent-Based Computational Economics*, Elsevier (2006)

[Clocksin 03] Clocksin, W. F. and Mellish, C. S.: *Programming in Prolog: Using the ISO Standard*, 5th edition, Springer (2003)

[Epstein 05] Epstein, J. M.: “Remarks on the foundations of agent-based generative social science,” In L. Tesfatsion and K. L. Judd (eds.), *Handbook of Computational Economics, Vol. 2: Agent-Based Computational Economics*, Elsevier (2006)

[Fishburn 02] Fishburn, P.: “Acyclic sets of linear orders: A progress report,” *Social Choice and Welfare*, Vol.19, pp. 431-447 (2002)

[Gaertner 01] Gaertner,W.: Domain Conditions in Social Choice Theory, Cambridge University Press (2001)

[Gibbard 73] Gibbard, A.: “Manipulation of voting schemes: A general result,” *Econometrica*, Vol. 41, pp. 587-602 (1973)

[Kalai 77] Kalai, E. and Muller, E.: “Characterization of domains admitting nondictatorial social welfare functions and nonmanipulable voting procedures,” *Journal of Economic Theory*, Vol. 16, pp. 457-469 (1977)

[Inada 69] Inada, K.: “On the simple majority decision rule,” *Econometrica*, Vol. 36, pp. 490-506 (1969)

[Moulin 88] Moulin, H.: *Axioms of Cooperative Decision Making*, Cambridge Univesity Press (1988)

[Muller 77] Muller, E. and Satterthwaite, M. A.: “The equivalence of strong positive association and strategy-proofness,” *Journal of Economic Theory*, Vol. 14, pp. 412-418 (1977)

[Parsons 02]Parsons, S., Gmytrasiewicz, P., and Wooldridge, M.: *Game Theory and Decision Theory in Agent-Based Systems*, Kluwer Academic Publishers (2002)

[Saeki 80] 佐伯胖:『決め方の論理』東京大学出版会(1980)

[Salles 76] Salles, M.: “Characterization of transitive individual preferences for quasi-transitive collective preference under simple games,” *International Economc Review*, Vol. 17, pp. 308-318 (1976)

[Satterthwaite 75] Satterthwaite, M. A.: “Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions,” *Journal of Economic Theory*, Vol. 10, pp. 187-217 (1975)

[Sen 82] Sen, A.: *Choice, Welfare and Measurement*, MIT Press (1982)

[Sen 69] Sen, A. and Pattanaik, P. K.: “Necessary and sufficient condition for rational choice under majority decision,” *Journal of Economic Theory*, Vol. 1, pp. 178-202 (1969)

[Starling 94] Starling, L. and Shapiro, E.: *The Art of Prolog: Advanced Programming*, 2nd edition, MIT Press (1994)

[Takekuma 00] 武隈愼一: 数理経済学とコンピュータ，特集金融・数理経済学とコンピュータ：数理モデルで読み解く経済現象, *Computer Today*, Vol. 17: pp. 10-15 (2000)

[Wilson 72] Wilson, R.: “Social choice theory without the Pareto principle,” *Journal of Economic Theory* , Vol. 5: pp. 478-486 (1972)