

# 共進化モデルの解析とその高速化

山口 佳樹<sup>†</sup> 丸山 勉<sup>††</sup> 星野 力<sup>††</sup>

集団に所属する個体は、集団に特有の共通のモラル（文化、道徳、習慣など）に従って行動を決定するが、モラルの発生過程を現実の世界から考察することは非常に難しい。本稿では繰り返し囚人のジレンマモデル (Iterated Prisoners Dilemma game) の得点行列を進化させることで、その解明を試みた。しかし、このモデルは非常に多くの計算時間を要求するため、十分なシミュレーションを行うことが非常に難しい。そこで、書換え可能なハードウェアを用いた高速計算によりこの問題を解決した。本稿で用いた書換え可能なハードウェアである Field Programmable Gate Arrays (FPGAs) は、計算機やメモリから回路情報をダウンロードすることで、問題毎に最適な回路を提供することができる LSI である。この LSI を用いて、本稿では複雑適応系の計算において SUN Workstation (Ultra Sparc 200MHz) に対し 120 倍の高速化を実現した。ハードウェアによる高速計算は、複雑適応系のモデルを様々な制約から解放する有力な手段であり、本稿では共進化モデルの解析結果と共に FPGA を用いて行った高速化手法について示す。

## High Speed Hardware Computation of Co-evolution Models

YOSHIKI YAMAGUCHI,<sup>†</sup> TSUTOMU MARUYAMA<sup>††</sup>  
and TSUTOMU HOSHINO<sup>††</sup>

It is difficult to understand the emergence of social morals clearly. As the first step, we observed how the scores in IPD games are evolved. In general, the scores of IPD games are fixed, and not changed during the computation. In our model, however, the scores are evolved using a simple Genetic Algorithm. Our model requires a lot of computation time, from several days to a week, for only one simulation. High speed hardware computation is required for our model. Field Programmable Gate Arrays (FPGAs) can provide the most suitable circuits for given problems by reconfiguring its circuits. In this paper, we show that a FPGA chip can achieve about 120 times of speedup compared with a workstation (Ultra-Sparc 200 MHz) in the computation of a co-evolution of strategies and scores in Iterated Prisoner's Dilemma game. This speedup makes it possible to challenge more complex problems beyond the limitation by software.

### 1. はじめに

様々な要因が複雑に絡み合う現実社会において集団（国家、民族、宗教など）に特有のモラル（文化、道徳、習慣など）の発生過程や発生原因を解析することは非常に難しい。そこで、本稿ではまず非常に単純なモデルを利用し価値観の発生過程を解析する。本稿で使用するモデルは、2層構造になっており、上層は価値観層、下層は個体層となっている。個体層では、繰り返し囚人のジレンマゲーム (Iterated Prisoner's Dilemma

game: IPD) を用いて各個体を進化させていく。価値観層では、個体層で行われる IPD ゲームに使用する対戦条件を遺伝的アルゴリズム (Genetic Algorithm: GA) を用いて進化させる。個体層と価値観層を共進化させることで価値観の発生過程をより詳細に調べることができる。しかし、このモデルの進化計算には非常に多くの時間が要求されるため、十分なシミュレーションを行うことができない。

そこで、本稿ではハードウェア化による共進化モデルの高速化を行った。Field Programmable Gate Arrays (FPGAs) は、計算機やメモリ上から回路情報をダウンロードすることで回路を容易に変更することができる書換え可能な LSI である。そのため FPGA を組み込んだシステムは、与えられたアプリケーションに対してより最適な回路を提供することができる (1)。本稿では、共進化モデルの高速化において FPGA 1 チッ

<sup>†</sup> 筑波大学理工学研究科

Master's Program in Science and Engineering, University of Tsukuba

<sup>††</sup> 筑波大学機能工学系

Institute of Engineering Mechanics Systems, University of Tsukuba

ブで、ワークステーション (Ultra Sparc 200MHz) に対して 120 倍の高速化を実現した。この高速化により、数日を要するシミュレーションをわずか 1 時間で終了させることができる (10)。以降、本稿では第 2 章で価値観と個体の共進化モデルとその解析結果を、第 3 章で FPGA を用いた高速化手法について、第 4 章で以上のまとめをおこなう。

## 2. 価値観と個体の共進化

### 2.1 価値観と個体

繰り返し囚人のジレンマ (Iterated Prisoner's Dilemma: IPD) は、非常に単純なモデルでありながら複雑な振る舞いを示すことから、複雑適応系の計算モデルとして、非常に多くの分野で使用されている。しかし IPD は、シミュレーションの過程で発生する小集団レベルの融合、分化、対立、協調などの解析を行うにはモデルとして不十分なものがあつた。観測者によらず、小集団レベルの挙動を解析するには、集団行動そのものを個体の解析より切り放して考える必要がある (6)。

集団ごとに特有のルールを価値観とした場合、その価値観を独立に記述し観察を行うことができれば、価値観の発生を具体的に観測できる。そこで我々は、共進化モデルを提案する。共進化モデルは遺伝的アルゴリズム (Genetic Algorithm: GA) により進化する価値観層と、IPD により進化する個体層の 2 つの層より構成されているモデルである。第 1 層である価値観層は、GA を用いて進化を行うことで、小集団の融合、分化、対立行動の解析を具体的かつ容易に行うことができるようにしたことで、モデル全体の振る舞いを観測者依存に陥らないようにしている。第 2 層である個体層は IPD による進化を行うことで、観測者は現在までの研究を踏まえたモデルのより細密な解析を行うことができる。

### 2.2 共進化モデル

共進化モデルは、価値観層と個体層の 2 層構造になっている。このモデルを簡単に示すと図 1 のように表現できる。

個体層 (Agents' Layer) において、個体は IPD を用いて進化を行う。しかし、得点の計算方法は従来の IPD とは多少異なったものを使用している。共進化モデルでは、IPD の基本得点行列に価値観として 2 変数 ( $\alpha_i, \beta_i$ ) を加えている (図 1)。価値観として定義されている各得点行列の 2 変数に色々に変化を起こすことで様々な振る舞いを引き起こすことが目的である。価値観は、Simple GA を用いることで進化が行われ、その評価は、価値観を採用した IPD の個体数を利用して行われる。

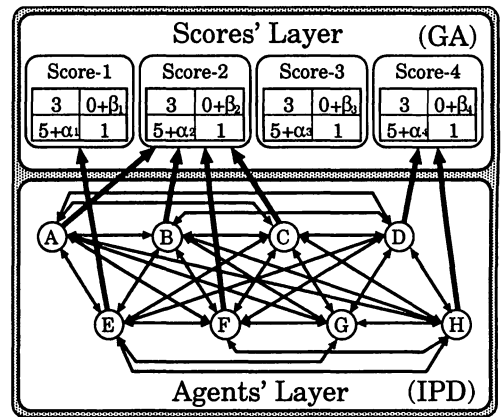


図 1 価値観と個体の共進化モデル

具体的に示すと、図 1 において、個体 -A、B、C、F、から価値観 2 (Score-2) に対して矢印が出ている。逆に、価値観 3 (Score-3) に対しては矢印は一つも出していない。この矢印の数 (得点行列として採用した個体数) が評価関数として利用されるので、価値観 2 は、4 個体の支持 (4 本の矢印) を持つことで他の個体に比べ次の世代により多くの個体を残すことができる。逆に、価値観 3 は矢印が 1 本もないため、次の世代では減少 (もしくは絶滅) するかもしれない。

各個体は、初期状態として乱数を用いてランダムに価値観を決定する。もちろん価値観を指示しない個体も存在しその個体は Lindgren の提唱した IPD モデルに従い対戦を繰り返す (1), (2)。突然変異などで新しい個体が生じるときや、指示する価値観が淘汰された個体も同様に乱数を用いることで代替りの新しい価値観 (もしくは Lindgren's Model で使用される基本得点行列) を選択しゲームを進めていく。

この価値観という概念を使用することで、我々は主観的でないモデルの振る舞いの解析を狙っている。

### 2.3 価値観の設定と進化

表 1 共進化モデルの得点行列

		個体-2	
		協調 (C)	裏切り (D)
個体-1	協調 Cooperate	Reward $3+\gamma_i$	Sucker $0+\beta_i$
	裏切り Defect	Temptation $5+\alpha_i$	Punishment $1+\delta_i$

価値観を採用した得点行列を表 1 に示す。得点行列中で使用される変数 ( $\alpha_i, \beta_i, \gamma_i, \delta_i$ ) は、変化を繰り返すことで、以下のような色々なゲームルールに対応する得点行列に進化させることができる。

今回、我々は IPD の得点行列を満たす範囲で解析を

- $T > P > R > S$  行き詰まりゲーム
- $T > R > P > S$  囚人のジレンマ
- $T > R > S > P$  チキンゲーム
- $R > T > P > S$  鹿狩りゲーム

行うという制約を設定したので、上の IPD の条件式に加え共進化モデルの各変数の変域を以下のように決定した。

- $0 \leq \alpha_i < 1$  (1)
- $-1 < \beta_i \leq 0$  (2)
- $S + T < 2R$  (3)
- $\gamma_i = 0$  (4)
- $\delta_i = 0$  (5)

IPD の対戦において計算される得点は、各個体の指示する価値観により決定されるため、 $\alpha$  が 1 に近いような得点行列を持つ個体は、他の個体との対戦に際し、非常に大きい得点を得るかもしれない。逆に、 $\beta$  が -1 に近いような非常に小さい得点行列を持つ個体は他の個体との対戦の時に不利な状況をつくるかもしれない。

### 2.4 個体の戦略の進化

本稿では、IPD の進化の基本的モデルは Lindgren のモデルを使用している 1), 2)。このモデルでは、次手は、32 ビットの戦略テーブルに記述されており、次の対戦で出す手の選択は、過去の対戦履歴 (3 対戦前 (相手), 2 対戦前 (自分, 相手), 1 対戦前 (自分, 相手)) より決定される (図 2)。戦略テーブルは、世代毎にある一定確率で突然変異をおこし様々に変化する。

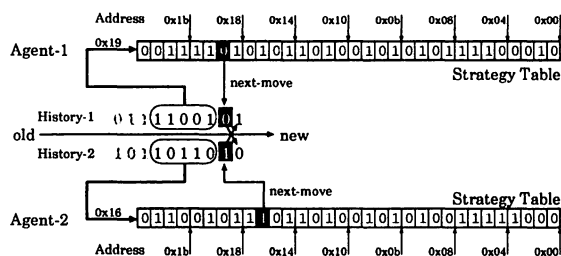


図 2 次手の選択方法

### 2.5 共進化モデルの解析例

図 3 に、共進化モデルのシミュレーションにより得られた全個体の平均得点のグラフを示す。初めより 20,000 世代までは、色々な個体が発生しては消えていき、目立った変化はグラフ上より観察されなかった。

20,000 世代周辺において、急激に裏切り戦略を取る個体がモデル世界に浸透し得点が減少している。そして、27,500 世代周辺で再び協調戦略を選択した個体がモデル世界を占拠する。

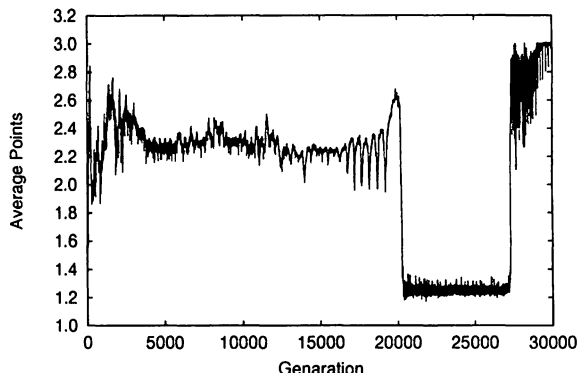


図 3 全個体の平均得点

図 4 に、価値観の変動を世代ごとに表したグラフを示す。横軸に世代、縦軸はモデル内に存在する全価値観 (20 価値観) の距離の総和 (分布) を示している。

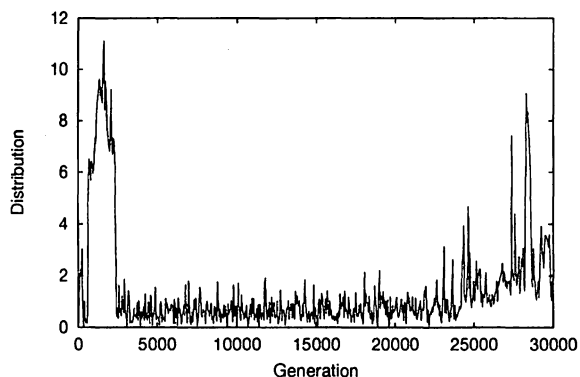


図 4 価値観の世代変動

2つの価値観の距離は、 $\alpha$ 、 $\beta$  の値を用いて表した 2 次元空間のユークリッド距離で示されている。よって、式 (1)、(2) から 2つの価値観の距離の最大値は  $\sqrt{2}$  と表される。分布は、20 個体全ての距離の総和で示されるため、価値観が一点に収束すればするほど (ある決まった価値観に収束すればするほど) 分布は小さく表現され、逆に分布が大きいということは、その世代は様々な価値観が存在していることを表している。

分布図からモデルの振る舞いを解析すると、0 世代目から 2500 世代目までの初期世代は、分布は大変大きなものとなっている (図 4)。この初期世代では、個体の戦略が十分進化していないため、価値観の選択や淘汰というものが行われず様々な価値観が存在し多様性を生じるため分布が大きいものとして表れている。その後、個

体が持つ戦略の進化とともに  $\alpha$  は 1 に、 $\beta$  は 0 に近づき始める。この価値観は、支持する個体に対しその個体が対戦相手を裏切るときに相手に対し非常に有利な得点 (5+1) を与え、逆に相手に裏切られたときも失う点数を少なくするため (0-0) 裏切り戦略が浸透し始めたモデル世界において非常に有利な状況を生む価値観としてモデル世界に浸透していく。そのため、価値観が収束するため価値観の分布は非常に小さいものとなり 24000 世代くらいまでこの状態が続く (図 4)。その後、価値観の分布は再び大きくなり多様性を生じ始める。

このときのモデル社会は、裏切り戦略が浸透している状態で安定している (図 3)。表 1 に示されるように完全に裏切りあいの対戦では、各個体は 1 点しか手に入れることができない。裏切りあいの対戦では、 $\alpha$ 、 $\beta$  に対して制約条件が働かないため価値観が色々と変化しても淘汰されることがない (価値観の 2 変数がモデル社会の得点と独立に変化するため多様性を生じても制約を受けない)。

27500 世代でモデル世界が裏切り社会から協調社会に急激に変化している (図 3)。このとき、裏切り戦略と協調戦略を取る個体がモデル世界に同時に存在するが、モデル世界の変化が非常に急激であるため (この変化は 80 ~ 100 世代で完了する) 価値観の分布はほとんど減少することなくそのまま保存されている。そのため、価値観の分布の目立った減少はグラフからは見られない (図 4)。その後、再び分布が増大している (図 4)。これは、協調社会でも裏切り社会と同様に、 $\alpha$ 、 $\beta$  は独立に変化を繰り返すため多様性が増加し続けている。

30000 世代周辺では、裏切り戦略がモデル内に生じ始めるため価値観の淘汰が生じ収束を始めるため、価値観の分布が減少しはじめている (図 4)。

図 5、図 6、にそれぞれの状況において特徴的な価値観の分布図を示す。図 5 (20326 世代) は、モデル社会の平均得点が激減し、価値観の分布が 0 に近づいている状態を取り出したものである。図 6 は、価値観に多様性が生じている協調社会における価値観の分布を取り出したものである。協調社会は、表 1 の *Reward* に相当し、 $\alpha$ 、 $\beta$  の影響がない中立状態である。そのため、価値観の多様性が進み価値観の分布として様々な値をとる図となっている。

### 3. ハードウェアによる高速計算

一般に複雑適応系の計算は、膨大な繰り返し計算から予測不可な振る舞いを発生させるものであり、本稿の共進化モデルも 1 回のシミュレーションを実行するために大量の計算時間を必要とする。このため、時間的な制

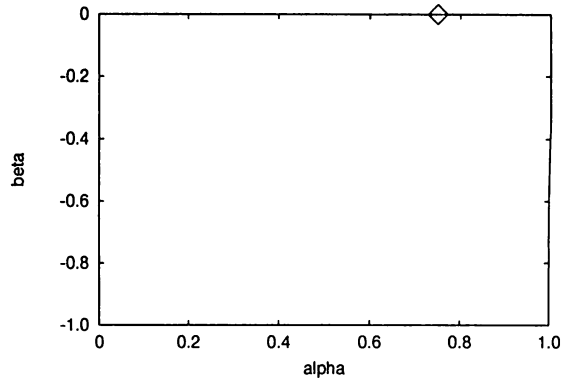


図 5 価値観の分布 (20326 世代)

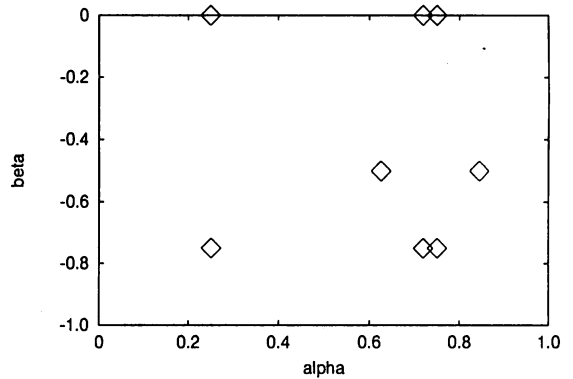


図 6 価値観の分布 (28343 世代)

約からモデルに対して厳しい制約を付けざるをえない状態にある。そこで本稿では、モデルのシミュレーションを高速化する方法として、書換え可能なハードウェアである Field Programmable Gate Array (FPGA) を用いた専用ハードウェアの開発を行った。これにより本稿では、SUN Workstation (Ultra Sparc 200MHz) に対し FPGA 1 チップで 120 倍の高速化を実現した。

本稿では FPGA を選択したが、ハードウェア化という点においては、従来から利用されている ASIC などの LSI を用いてれば FPGA より高い周波数で高速なシステムを実現できるかもしれない。しかしシステム設計した後、対象とするモデルの細かい変更、モデルの拡張に対して ASIC などでは柔軟な対応が難しく、LSI の持つ性能を十分に活用することが難しい。そのため、複雑適応系のモデルのようなモデルの拡張や変更が頻繁に起こりえる高速処理にはあまり適していない。FPGA を用いることで、モデルの変更や変数の変更などに対して柔軟に対応できるシステムを容易に作成することが可能である。以上から、本稿ではモデルの変更に柔軟に対

処ができる FPGA を採用した。

### 3.1 複雑適応系の高速化

進化計算は、膨大な量の個体が同様の処理を数多く繰り返すことで、予測不可分な結果を導き出すものである。この流れをプログラムの表すと図 7 のように表現できる。

```
for(j=0;j<agent(M);j++){
  for(i=0;i<iteration;i++){
    stage-1 (agent(j)) ;
    stage-2 (agent(j)) ;
    .
    .
    stage-M (agent(j)) ;
  }
}
```

図 7 複雑系の計算 (プログラム)

多重ループにより同じ処理が数多く繰り返し行われるため、ハードウェア化することで高速な処理を容易に実現することができる。また、各個体の処理は基本的に独立に実行されるため、並列化も容易である。以上のプログラムの処理を図示すると図 8 のように表現できる。

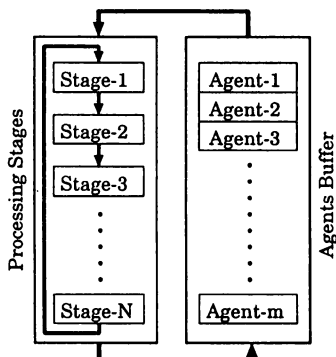


図 8 複雑系の計算

複雑系の計算は、基本的な処理 (Processing Stages) と処理に必要なデータ (Agents) から構成されている。個体 (Agents) は短い bit-string で表されるため、多くのレジスタを必要としない点でハードウェア化が行いやすい。また、処理も bit 操作、加減算などの簡単な操作が多いため、高速化を行うのに非常に適した構造になっている。

そこで、各処理 (Processing Stages) をパイプライン的に処理を行うことで高い周波数での動作を実現し、

データ (Agents) の処理を可能な限り並列にすることで飛躍的な速度向上が可能になる。

### 3.2 FPGA への実装

ハードウェア化において、ASIC などの LSI は、設計後のモデルの細かい変更などに対してハードウェア的な変更が難しく、複雑適応系の解析のような、モデルの条件やモデル自体が頻繁に変更される高速処理にはあまり適していない。

本研究では、モデルの変更に対して柔軟に対応する LSI として、書き換え可能なハードウェアである Field Programmable Gate Array (FPGA) を用いている。FPGA は、ハードウェア記述言語 (HDL) を用いることで、ソフトウェアの変更に対応したハードウェアの記述を容易に行うことができ、リアルタイムに回路の書き換えを行うことができる。また、最近の LSI 技術の進歩と共に、FPGA の回路量は飛躍的に増大し、動作周波数も非常に速いものになっている。そのため、様々なシステム LSI として利用されはじめて (3), 4), 7)。

複雑系の高速計算において、FPGA の回路量は重要な問題である。これは、どれだけ多くの並列処理を実現可能かは FPGA の回路量により決定されてしまうからである。しかし、この問題は FPGA における LSI 技術の進歩と共に解決される問題であるように思われる。

### 3.3 並列化と回路容量

我々の共進化モデルを全て回路上に実装するためには、現在の FPGA の大きさでは不十分であり、目的とする飛躍的な高速化を実現できない。そのため、我々はモデルの中で一番時間がかかる部分を切り出すことで共進化モデルを高速化することを試みた。モデルの拡張も考慮に入れ総計算時間に占める割合が多いものを全てハードウェアで処理をおこなうようにした。

共進化モデルにおいて、最も計算時間がかかる部分は IPD モデルを使用した個体の対戦部分であるということがわかった。総当たり戦の計算時間のオーダーは、個体数の 2 乗 ( $N_{agents}^2$ ) で表現されるが、他の部分の計算時間のオーダーは、個体数 ( $N_{agents}$ ) もしくは価値観数 ( $N_{Scores}$ ) のオーダーで表現されるため計算時間は IPD の対戦部分と比較し非常に少ない。本稿のモデルは 1024 個体、価値観数 20、1 ゲーム 1024 対戦の総当たり戦、30000 世代でゲームが行われるため、価値観の全ての処理の計算時間は多めに見積もっても総計算時間の 2% にも満たない。そこで、IPD の総当たり戦に的を絞った高速化を行った。

本稿では、FPGA 1 チップ (ALTERA EPF 10K100) を使用し、24 対戦 (48 個体) が同時に対戦できるハードウェア化を作成した。モデルの拡張と

共に、ハードウェアに外部メモリを使用することでモデル内の個体数を増やす必要が生じるかも知れない。しかし、個体を取り替えるために必要とする通信時間も全体の計算時間と比較すると非常に小さいものであるため高速化における問題にはならない。

### 3.4 IPD モデルのハードウェア化

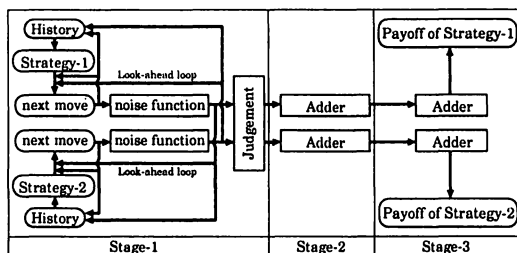


図9 個体の対戦部 (x24/chip)

図9に、FPGAに実装したIPDの対戦部分の構成を示す。この回路が1chip上に24組実現されている。この回路は、stageが3つに分かれているが、各stageは1 clock cycleで処理を終了する。各ステージをパイプライン化したことにより33MHzという高い周波数で回路を動作させることができた。

また、この手法により計算に  $N_{repetition} \times 3$  かかる処理を、 $N + 2$  に減らすことができる(図10)。

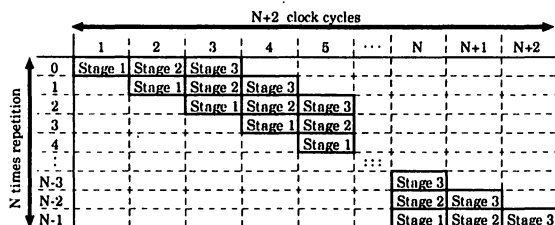


図10 Pipelining of IPD games

### 3.5 個体の対戦と次手の決定

個体の対戦において、ハードウェア的には次のような処理を経て次手が決定される(図11)。まず、対戦履歴、戦略はそれぞれデータを保持する必要があるためレジスタ上に格納される。次に、レジスタに格納されている対戦履歴から戦略テーブルのアドレスが計算され、そのアドレスをもとに戦略テーブルから次手が決定される。対戦履歴は、自分の次手と相手の次手を参照することで対戦終了毎にその都度更新される。

このモデルは、伝達路のノイズも考慮するモデルなので、上記の次手をテーブルから参照する間に、乱数を用

いてノイズを同時に発生させる。ノイズが発生した時のみ、乱数発生器からEX-ORゲートに1が流れ次手を反転させて相手に伝達させる。以上の処理を本稿では、1 clock cycle (33MHz)で終了させている。

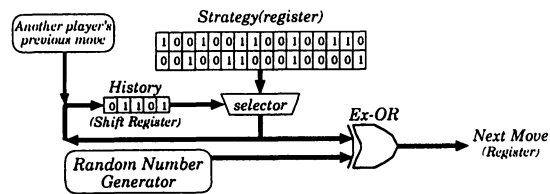


図11 次手決定回路

### 3.6 配線容量の削減

回路の設計を行う時、重要な問題として実行回路の大きさと同時に配線容量の問題である。本稿では、可能な限りIPDを並列処理することで高速化を図るため回路は効率良く使用しなければならない。しかし、各個体の総当たり戦を実現するとき従来の方法では、並列化数(個体数)の2乗のオーダー ( $Order(N_{agents}^2)$ ) で配線領域が増加してしまうため、実装する回路よりも先に配線領域が飽和してしまい高速化の障害となってしまう。

そこで、本稿では新しいアルゴリズムを提唱することでこの問題を解決した(図12)。このアルゴリズムは、各個体の総当たり戦を個体の移動のみで実現するものであり、配線量は個体数オーダー ( $Order(N_{agents})$ ) でしか増大しないため、回路量に比例した線形な速度向上を実現でき、複数のLSIにまたがるような大規模なシステムや、新しく市場に出てくる次世代の大規模なLSIに適用することで、数千倍の速度向上も可能である。

### 3.7 並列化アルゴリズム

プレイヤーの対戦をいかにスムーズに、無駄なく対戦させるか、が、並列化アルゴリズムの重要ポイントである。シリアルにプレイヤーを動かすということは、言い換えれば回路をベルトコンベヤに見立て、それに乗っているプレイヤーの処理をどのように実行していくか、である。そこで、プレイヤーがシリアルに処理されるということは、時系列の方向にもシリアルにデータが処理されているはずである(図12)。このことに着目しプレイヤーの対戦の組合せの重なりを防ぎ、アルゴリズム化したものが「いりこアルゴリズム」である。

この方法ならば、偶数であれば任意の数のプレイヤーに対しシリアルに、かつ重複せず総当たり戦をするのに適した回路を提供することができる。

例として、16個体について考えてみる。図12を見

Deadlock  
 12のX-カ  
 100% - 1対

ている。

Agents combination of palying IPD games

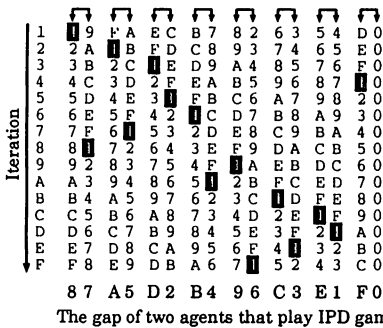


図 12 いりこアルゴリズムによるプレーヤーの配置

てみると、時間軸を縦に取ってあるが、プレーヤーはその時間軸に対しシリアルに動いている（番号がつかっている）。この状態を維持して、全対戦行うには「隣り合う対戦相手のプレーヤーの時間のずれ」が問題になってくる。すなわち、対戦が1つずれた形で実行される列と、2つずれた形で実行される列では、異なった対戦を行っていることを表している。同じ時間のずれの対戦が存在すると、その組合せは同じ対戦しか行わないことが図 12 からわかる。

そこで、図 12 の列の下の数字が鍵となる。この数字は、時間のずれを表していて 4 と書いてあれば 4 ターン時間がずれていることを表している。この時間のずれを用いて、最適な回路を設計したのが図 13 である。

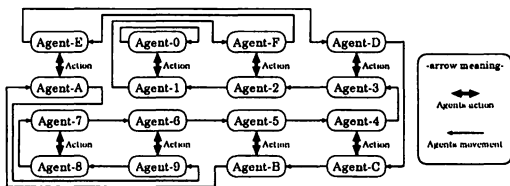


図 13 いりこアルゴリズム適用例 (16 個体)

総当たり戦の様子 (16 個体の場合) を、図 13 で説明すると、図 13 において、各個体の番号は初期値を表しており、対戦が終了するごとに、矢印に従って各個体が移動することを示している。このアルゴリズムを利用することで特別な捜査を必要とせず個体の移動のみで全ての総当たり戦を実現できる。

### 3.8 実行結果

表 2 に共進化モデル (48 個体、総当たり戦モデル) をハードウェア化した実行結果を示す。本稿では、FPGA を使用することによりワークステーション (Ultra Sparc 200MHz) に対し、120 倍の高速化を実現した。IPD の並列化とパイプライン処理によりこの高速化は実現し

表 2 実行結果

	時間(sec)	速度向上率
Ultra-Sparc (200 MHz)	119.02	1.00
FPGA (33 MHz)	0.98	121.45

## 4. おわりに

我々は、共進化モデルを利用することで価値観の発生の解析を試みている。これからの課題として、 $\alpha$ 、 $\beta$ 、の制限を緩めると共に、 $\delta$ 、 $\gamma$ 、の 2 変数を新しく導入することで IPD という枠組みをはずし、Deadlock、Stag Hunt、Chicken、などの様々なゲームルールを混合したより複雑なモデルの解析を進めていく。

また、我々は、共進化モデルに対して FPGA を用いることで Sun ワークステーション (Ultra-Sparc 200MHz) に対し、120 倍の高速化を実現した。この速度向上率は、5 日かかるシミュレーションを約 1 時間で終了させることのできるものであり、更に複雑で計算時間を必要とするようなモデルの解析も可能になる。現在、FPGA のサイズは急速に大きくなっており我々が使用した FPGA の 10 倍近いものも販売される。本稿の高速化に使用したアルゴリズムは回路量にほぼ線形的な速度向上を期待できるため、最先端の LSI を複数個使用することで、数千倍の速度向上も可能である。今後は、複雑適応系のシステムとして共進化モデルの全モデルをハードウェアに実装することを進めていく。

最後に、本稿で提唱するアルゴリズムが、現実の世界で発生する社会現象を解明する糸口になることを我々は信じている。

## 参考文献

- 1) K. Lindgren, "Evolutionary Phenomena in Simple Dynamics", *Artificial Life II*, pp.295-312, 1991.
- 2) K. Lindgren and M. G. Nordah, "Cooperation and Community Structure in Artificial Ecosystems" *Artificial Life 1*, pp.15-37, 1994.
- 3) P. Graham and B. Nelson, "Genetic Algorithm In Software and In Hardware - A Performance Analysis of Workstation and Custom Computing Machine Implementations", *FPGAs for Custom Computing Machines*, 1996 pp.216-225.
- 4) Stephen D.Scott, Ashok Samal and Shared Seth, HGA: "A Hardware-Based Genetic Algorithm" *ACM/SIGDA 1995*, pp.53-59
- 5) David B.Fogel, "Evolving Behaviors in the

Iterated Prisoner's Dilemma", Evolutionary Computation Vol.1:1, pp.77-97, 1993.

- 6) 星野 力 著, "進化論は計算しないとわからない - 人工生命白書 -", 共立出版, 1998
  - 7) T.Maruyama, T.Funatsu and T.Hoshino, "A Field Programmable Gate-Array System for Evolutionary Computation" FPL'98 pp.356-365
  - 8) 丸山 勉、船津 輝宣、関 峰伸、山口 佳樹、星野 力, "Field-Programmable Gate-Array による進化的計算の高速化", 情報処理学会論文誌 Vol.40 No.5, May 1999
  - 9) Yoshiki Yamaguchi, Tsutomu Maruyama and Tsutomu Hoshino, "High Speed Hardware Computation of Co-evolution Models", European Conference on Artificial Life V, September 1999
  - 10) 山口 佳樹、丸山 勉、星野 力, "モラル付き囚人のディレンマの高速計算", 情報処理学会 第 58 回全国大会, 1H-05 (1999)
  - 11) 情報処理学会誌 Vol.40 No.8: 特集 やわらかいハードウェア (1999).
-