

## モラル付き囚人のディレンマの高速計算

1H-5

山口 佳樹（筑波大学）

丸山 勉（同）、星野 力（同）

### 1 はじめに

現在、複雑な社会現象のシミュレーションに、繰り返し囚人のディレンマが用いられている [1]。

本稿では FPGA を用いて、その拡張モデルであるモラル付き IPD の高速計算手法を提案する。このモデルは、非常に単純な計算を数多く繰り返し行なうという特徴を持っており、ハードウェア化することで飛躍的な速度向上が期待できる。

### 2 モラル付き IPD

IPD を拡張したモデルに、モラル付き囚人のディレンマがある [2]。このモデルは、個体と独立なパラメータであるモラルという概念が数値化され導入されており、解析をより具体的に行うことができる。このモデルを簡略に図式化したものが図 1 である。

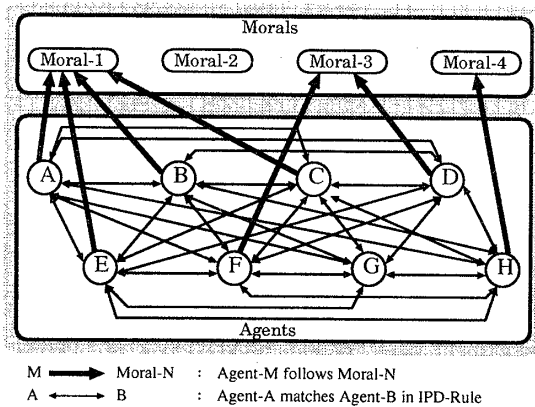


図 1: モラル付き IPD 概略図

個体とモラルはそれぞれ独立しているが、その関係の結び付きは個体側 (A, B, ..., H) の選択により決定される。例えば、図 1 において、個体 A → Moral-1、

となっている。これは、個体 A が Moral-1 を選択していることを意味している。

個体はあるモラルを選択することで、モラルに従ったより有利な得点を対戦時に得ることができる。個体がモラルを選択しない場合には、通常の IPD ルールで対戦が行われ得点を得る（図 1 のモラルに対して矢印を出していない個体 G がそれに該当する）。

個体は対戦で得た得点を評価値として、IPD モデルに基づいて増減される。モラルは自分を選択した個体数とその評価値から、GA を用いてより優れたモラルへと変化を繰り返す。

### 3 計算モデル

IPD の演算は、対戦 (Action) → 評価 (Evaluation) → 選択 (Selection) → 突然変異 (Mutation) → 対戦... の繰り返しである。ここで、個体数 4、モラル数 4 のときの計算モデル例を図 2 に示す。

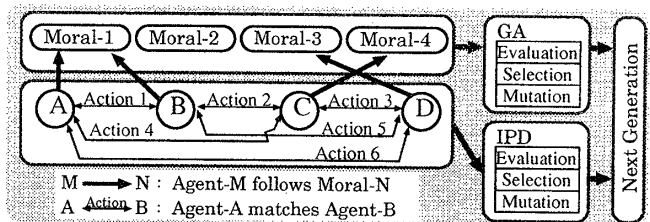


図 2: モラル付き IPD 計算図

このモデルでは、全個体は総当たり戦を行う。総当たり戦の各対戦を野球にあてはめると、野球のイニングは Action に相当する。野球では 1 試合 9 イニング繰り返されるが、Action も 1 回の対戦で複数回繰り返される。本稿ではこの回数を Turn とする (1 回の対戦は、Action を Turn 回行うことで終了する)。

全ての対戦が終了すると、評価、選択、変異を経て次の世代 (Next Generation) へ処理が移行する。

計算モデルにおいて、評価、選択、変異は 1 世代に N 回しか行われないので、そのオーダーは  $O(N)$  である。

次に、個体の総対戦数のオーダーについて考えると、個体数  $N$  の総当たり対戦回数  $Match$  は、 $N$  を用いて、以下の式で記述することができる。

$$Match = \frac{N(N-1)}{2}$$

各々の *Action* に対しては  $Turn$  回対戦が繰り返されるので、個体数  $N$  に対して総対戦数の計算オーダーは  $O(N^2 \times Turn)$  となる。

実際のシミュレーションでは、 $Turn = 1024$  で、個体数  $N$  はほとんど  $N > 128$  で実行される。対戦部分で必要とする演算時間は評価や選択などの演算時間に比べて非常に大きく、モデル全体の演算時間は対戦部分の演算時間にほぼ等しいものとなる。そこで、対戦部分を並列、高速化することでモラル付き IPD の高速化が実現できる。

#### 4 計算モデルのハードウェア化

本稿では個体の対戦部に重点をおき高速化を行った。対戦部分のハードウェアの概略図を図3に示す。

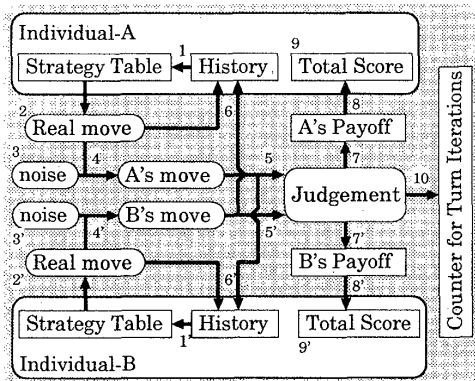


図3: Action の構造 概略図

対戦部分の主な処理を図3中の数字を用いて説明する。初めに、過去の履歴 (*History*) を利用して (1)、戦略テーブル (*Strategy*) から次の手 (*move*) を選び出す (2)。選んだ手を用いて対戦を行い (5)、対戦によって得た得点 (*Payoff*) (7) を総得点 (*Total Score*) (9) に加算する (8)。最後に、履歴を更新し (6)、1 回の対戦を終了する。但し、個体間の通信にノイズの混入があるモデルなので、乱数を用いてノイズを混入させている (3, 4)。

以上の処理 (1, 2, ...10) を FPGA を用いて 33MHz、1clock で実現できる。これは、ハードウェアでは戦略から次の手を選ぶような bit 操作 (1, 2) を高速に行えること、また対戦部分 (*Judgement*) で実行される得点行列の表引きなどにおいて (5, 7)、メモリを使用せずにレジスタを用いていることによるものである。

#### 5 計算結果

対戦部分の評価結果を表1に示す。使用デバイスは Altera 社製の 100Kgate 相当のものを使用し、動作速度は 33MHz である。その結果、SUN Ultra Sparc 200MHz に対して約 121 倍の高速化を実現できた。

表1: IPD の処理速度比

	処理時間	速度向上率
Sun Ultra Sparc 200MHz	119.02s	1.00
FPGA	0.98s	121.45

#### 6 おわりに

IPD の対戦部分の計算において、WorkStation に対し約 121 倍の高速化が得られた。現在、GA と IPD の共進化の部分を作成中であるが、FPGA を用いた GA モデル単体のハードウェア化において、約 80 倍の高速化が報告されている [3]。以上から、本稿のモデルは最終的に 100 倍以上の高速化が得られると考えられる。

#### 7 参考文献

[1] Kristian Lindgren and Mats G. Nordahl (1994): Cooperation and Community Structure in Artificial Ecosystems, *Artificial Life* 1, 15-37

[2] 山内 敦、星野 力 (1998): 囚人のディレンマゲームにおける社会的行動の進化、筑波大学理工学研究科修士論文

[3] 船津 輝宣、関 峰伸、丸山 勉、星野 力 (1998): FPGA による GA の計算の高速化、情報処理学会 第 56 回全国大会、2R-07

[4] 山口 佳樹、丸山 勉、星野 力 (1998): FPGA を用いた繰り返し囚人のジレンマの高速計算、情報処理学会 第 56 回全国大会、2R-05