

## 競争巡回ロボットゲームとその探索アルゴリズム

### Competitively Traveling Robot Game and its Search Algorithm

苫米地 宣裕\*

Nobuhiro Tomabechi\*

\* 八戸工業大学

\*Hachinohe Institute of Technology

キーワード：競争 (competition)、巡回 (traveling)、ロボット (robot)、ゲーム (game)、探索 (search)、アルゴリズム (algorithm)

連絡先： 〒031-8501 青森県八戸市大字妙字大開 88 1 八戸工業大学システム情報工学科  
苫米地宣裕、Tel. 0178-25-8051, Fax. 0178-25-1691, E-mail: tomabech@hi-tech.ac.jp

#### 1. はじめに

自律的に行動する知能ロボットに対する関心が高まっている<sup>(1)-(4)</sup>。

本研究では、相手(人、または、ロボット)があり、相手と争いながら何かを自律的に実行するようなロボット(競争ロボットという)を取り上げる。

相手のあるロボットの制御は、人工知能/マルチエージェントの課題として、これまでも研究されてきた<sup>(5)(6)</sup>。しかし、その多くは、ロボット同士の協調行動に焦点を当てたものであり、競争ロボットに関する報告はあまり見られないようである。例えば、追跡ロボットの研究は、競争の要素を含んでいるが、追跡者同士の協調行動が主題となっている<sup>(7)</sup>。しかし、近年、ロボットコンテスト、ロボカップの開催などを通して、競争ロボットに対する関心が、高まり

つつある。

本稿では、ある領域内に目標が散在しており、相手(人、または、ロボット)と競い合って、目標に接触するという形式の競争モデル<sup>(8)(9)</sup>(競争巡回ロボットゲームという)について検討する。

このような競争モデルは、例えば、荷物/物品の収集作業において行動に対する報酬が個別に支払われる場合、あるいは、価値のある無主物(廃棄物、海/山の産物)の取り合いなどに当てはまると考えられる。また、抽象化された数理モデルとして考えると、セールスマンが商行為をしながら多くの都市をめぐるいわゆる「巡回セールスマン問題」<sup>(10)</sup>において、競争者が存在する場合(言わば、競争形巡回セールスマン問題)と捉えることもできる。

本稿では、次の点を明らかにする。競争モデルの明確化、本モデル特有の最適経路探索

アルゴリズムの提案、本モデル特有の局面評価法の提案、シミュレーションプログラムの作成、対戦の実施によるアルゴリズム・局面評価法の検証、モデルの特性解析、有効性の評価。

競争は、一種のゲームとして、モデル化できる。本モデルにおいて、目標、および、相手に関する情報は、一般的には、事前にあるいは常時、全てを知り得るとは限らないが、第一段階として、常時、完全に知り得るものとする。すなわち、完全情報ゲームモデルとする。

最適経路探索アルゴリズムとして、思考ゲームにおいて用いられる先読み探索アルゴリズム<sup>(11)</sup>の適用を試みる。しかし、本ゲームモデルは、次の点で、通常の思考ゲームとは異なっている。

競争者の行動は、交互行動ではなく、同時並行の形で行われる。

動作はアナログ量である。

競争者の行動能力は同じであるとは限らない。

従って、通常先読み探索アルゴリズムは、そのままでは、適用することはできない。

本稿では、目標に直行する動作を基本とし、相手が有利な場合は中途まで前進するという先読み探索アルゴリズム<sup>(12)</sup>(並行探索法と言う)を提案する。本アルゴリズムによれば、競争者の行動は擬似的に交互行動として取り扱うことができる。ただし、一方が連続行動を取り得ること、行動の中断が有り得ること、競争者の能力が異なる可能性があることなどの点で、通常先読み探索アルゴリズムとは異なっている。

本稿では、また、勢力圏という概念に基づく本ゲームモデル特有の局面評価方法を提案する。

提案したアルゴリズム・局面評価法に従って、シミュレーションプログラムを作成した。プログラムを用いて多くの対戦を行い、アルゴリズム

・局面評価法の有効性を確認した。

## 2. ゲームモデルと用語の定義

競争のゲームモデルを以下のように設定する。

[ルール1] 競争者(以下、プレイヤー)の数は2とする。プレイヤーは、それぞれ、独立に行動する。双方の機能は等しい(必ずしも等しい必要はないが、とくに断らない限り、機能は等しいとする)。

[ルール2] プレイヤーはある一定の行動領域(以下、フィールドという)Dにおいて行動する。D内において、プレイヤーは任意の経路上を移動することができる。

[ルール3] D内には目標が散在する。目標の数は変数とする。目標は価値点を有する。目標の分布はランダムとする(1回の対戦ごとに乱数によって決定する)。

[ルール4] プレイヤーが目標の存在する座標に入るとその目標が取得されたと見なされる。目標の価値点が得点に加算され、目標は削除される。

[ルール5] 双方のプレイヤーが同一目標の座標に、同時に入ろうとすると、いずれか一方が優先権を有する(優先権を有する方を先手、その反対側を後手という)。

[ルール6] プレイヤーの位置、目標に関する情報は、常時、完全に知ることができる(完全情報ゲームである)。

[ルール7] ある回数の対戦を行い、勝敗を、次のように判定する。

先手の得点 > 後手の得点 + H、ならば先手勝

先手の得点 = 後手の得点 + H、ならば引き分け

先手の得点 < 後手の得点 + H、ならば後手勝。

ただし、Hは、先手/後手の相違、機能の相違がある場合に対するハンデキャップ値を表わし

ており、多くの対戦結果に基づいて定める。

また、次の用語を定義する。

[定義1] 単位距離：プレイヤーが単位時間に移動する距離をいう。

[定義2] 単位区画：1個の目標が存在し得る最小の領域をいう。

[定義3] 勢力圏：あるプレイヤーから一定の距離R以内にある領域をいう。

[定義4] 行動：ある目標に向かって最短距離で直行する行為をいう。

[定義5] 半行動：上で定義した「行動」を中途まで行う行為をいう。

フィールドの形状、目標の座標、目標の価値点、単位距離と単位区画の関係などは任意であるが、本稿では、人間が対戦するとき考え易いように、以下のように設定する。

目標の座標は、単位距離の整数倍とする（離散的な値とする）、価値点は整数とする。単位区画 = 10 単位距離 × 10 単位距離、フィールドの形状は、16 × 16 単位区画。

図1に、目標の数を5とした場合のフィールドの初期配置の例を示している。図において、Aは先手プレイヤー、Bは後手プレイヤー、数字は目標の存在と、その価値点を表わしている。

### 3. 最善行動探索アルゴリズム

#### 3.1 並行探索アルゴリズム

最適経路探索に、思考ゲームで用いられる先読み探索アルゴリズム<sup>(1)</sup>の適用を検討する。

しかし、本ゲームモデルは、次の点で、通常の思考ゲームとは異なっている。

競争者の行動は、交互行動ではなく、同時並行の形で行われる。

動作はアナログ量である。

競争者の行動能力は同じであるとは限らない。

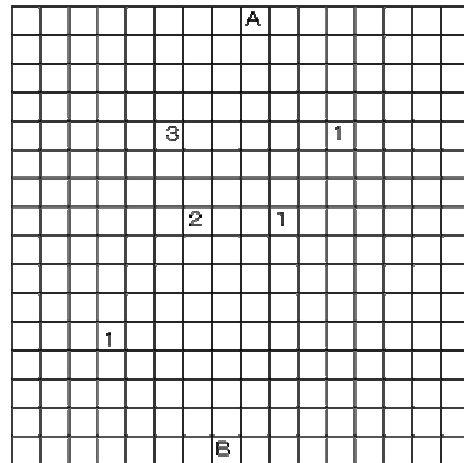


図1 ゲームの初期局面の例

A：先手プレイヤー、B：後手プレイヤー、  
数字：目標とその価値点

従って、通常の前読み探索アルゴリズムは、そのままでは、適用することはできない。

この場合でも、もし、プレイヤーの動作を最小移動単位で取り扱くと、擬似的に交互動作として取り扱うことができる。しかし、この最小移動単位の探索は、探索空間が膨大となり現実的でない。

本稿では、目標に最短距離で直行する動作(行動)を基本とし、相手が有利な場合は中途まで前進する(半行動)という探索方法(並行探索法という)を提案する。

本アルゴリズムによれば、競争者の行動は、擬似的に交互行動として取り扱うことができるので、通常の前読み探索方法が応用できる。ただし、次のような問題を含んでいる。先手、後手が常に交互に行動するとは限らない。後から行動した方が先に目標に到着する場合は、既行動を取り消さざるを得ない。

については、行動時間を計測しておき、行動時間の少ない方を手番とすることにより、解決することができる。については、取り消し

の可能性のある場合は、中途まで前進する（取り消しの生じない範囲で前進する：半行動）とすることにより、矛盾なく解決することができる。

深さ優先探索法<sup>(11)</sup>を修正した並行探索法の手順は次のようになる。まず、次の変数を設ける。

$L$ ：探索レベル（先読み探索の深さ）

$L_{max}$ ：探索レベルの限界値

$C$ ：2人のプレイヤーの走行時間の差

$M$ ：行動

$V$ ：局面評価値

$V_{MAX}$ ：局面評価の最大値

$M_b$ ：探索中途における優良行動

$V_b$ ： $M_b$ に対する局面評価値

$M_{BEST}$ ：最善行動

$V_{BEST}$ ： $M_{BEST}$ に対する局面評価値

### 【アルゴリズム1】並行探索法

（手順1） $L = 0$ 、 $C = 0$ 、手番 先手。

（手順2） $L = L_{max}$ 、または、目標がなくなったときは、 $V$  局面評価値。かつ、 $L = L - 1$ 、すなわち、探索レベルを1段階戻し、手順9へ再帰リターン。

さもなければ手順3へ行く。

（手順3） $C = 0$  ならば、手番 先手。 $C > 0$  ならば、手番 後手。

（手順4）探索候補をリストアップする。未取得の目標に対する行動は、すべてが候補となり得るが、優先順に整列し、順位の高いいくつかを探索候補とする。

相手がすでに同一目標に対して行動しており、かつ、相手の方が先に到着する場合はリストに加えない。

相手が同一目標に対して行動開始はしていないが、今、相手が行動開始すると相手の方が先に

到着できる場合は、目標に向かう中途の位置で停止するような行動（半行動）として取り扱う。

（手順5）先頭の探索候補ならば、次を行う。

$M_b$  先頭の探索候補

先手番ならば、 $V_b = V_{MAX}$

後手番ならば、 $V_b = V_{MAX}$ 。

（手順6） $M$  未探索候補の1つ。

到着した先の目標の価値点を得点に加算し、目標を削除する。ただし、半行動の場合は、中途の座標に移動するだけで、価値点の加算も目標の削除も行わない。

（手順7）先手番ならば、 $C = C +$  目標までの所要時間

後手番ならば、 $C = C -$  目標までの所要時間。

（手順8） $L = L + 1$ 、すなわち、探索のレベルを1段階上げる。手順2へ再帰ジャンプ。

（手順9）先手番ならば、 $V > V_b$ のとき、 $V_b = V$ 、かつ、 $M_b = M$

後手番ならば、 $V < V_b$ のとき、 $V_b = V$ 、かつ、 $M_b = M$ 。

（手順10） $M$ を戻す。削除した目標を戻す。得点を戻す。手番を戻す。 $C$ の値を戻す。

（手順11）未探索の候補があれば手順6へ、さもなければ手順12へ行く。

（手順12） $V_{BEST} = V_b$ 、 $M_{BEST} = M_b$ 。

（手順13） $L = 0$ ならば終了、 $M_{BEST}$ を最善行動とする。

さもなければ、 $L = L - 1$ 、すなわち、探索レベルを1段階戻して、手順9へ再帰リターン。

上記アルゴリズムにおいて、 $L$ 、 $L_{MAX}$ 、 $C$ は、全探索レベルを通して単一の変数とする。それ以外の変数は、すべて、再帰的に用いられる。すなわち、名称が同一でも、探索レベルが異なると、異なる変数として扱われる。

下線を記した部分、すなわち、手順3の走行

時間差によって手番を決定すること、手順4、手順6における「半行動」の導入、手順7における走行時間の計測が、通常の深さ優先探索法との主な相違点である。

本アルゴリズムは、プレイヤーの機能（単位時間当たりの走行距離）が異なっても適用することができる。なお、プレイヤーの機能が等しい場合は、走行時間を走行距離と言い換えても差し支えない。

### 3.2 半行動の取り扱い

行動を中断するタイミングとしては、次のような選択が考えられる。

ケース 1：同一目標に向かって相手が進行を開始した時。

ケース 2：同一目標に相手が到着した時。

ケース 3：上2つの中間地点。

ケース 1はどの目標を目指しているかを進行開始時点で正確に認識することが困難という問題点を有する。ケース 2は中断のタイミングが遅過ぎるきらいがあるが、互いの行動が一意に確定する利点がある。ケース 3は、互いが同一目標に向かうと見せて牽制し合う、「かけひき」の要素が生ずる。ここでは、ケース 3を採用する。

半行動は、中途の位置を目標とした行動とみなすことにより、通常の行動とほぼ同様に扱うことができる。通常の行動との相違は、単に、価値点を得られないだけである。半行動の後には、中途停止位置から、あらためて目標を探すこととなる。なお、行動の中断は、何度でも連続して生ずる可能性がある。また、相手が当該目標に向かって行動しなかった場合、当該目標に向かって行動する必然性はなく、停止位置から別の目標に向かうことも可能である。

### 3.3 並行探索法におけるゲームの木の表現

並行探索法に基づくゲームの木の書き方を以下のように定める。

#### 【ゲームの木の書き方】

初期局面を  $\text{A}$  で表す。

行動を矢印で表す。先手の行動は実線で、後手の行動は点線で表す。

到達目標を  $\text{C}$  で表す。  $\text{C}$  をノードと呼ぶ。

ノードの中に、目標の番号、あるいは、価値点を記入する。

必要に応じ、ノードの横に、その局面における走行時間差  $C$  の値を付記する。  $C = 0$  なら先手の手番、  $C > 0$  ならば後手の手番となる。

半行動は、小さな  $h$  で表す。ノードの番号は中断前の番号に半を意味する  $h$  を付けて表す。例えば、2番の目標に向かった行動を中断するときは、  $2h$  のように表す。価値点は記述しない。

図2に最も簡単な局面の例を、図3に、図2の局面に対応するゲームの木を示している。図において、実線/点線のいずれかが連続するのは、その側の連続行動を示している。

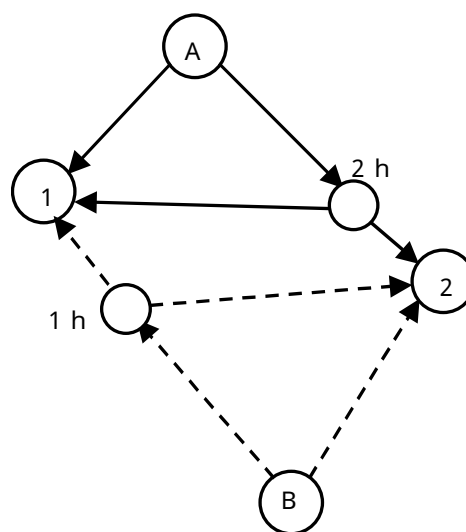


図2 配置例1

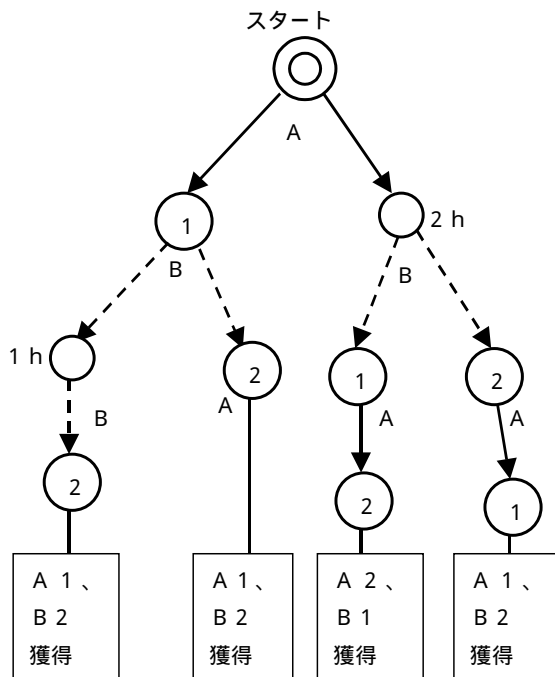


図3 配置例1に対するゲームの木

### 3.4 枝刈り

並行探索法では、通常の交互着手における枝刈り<sup>(13)</sup>と同様の手法により、探索数の削減を行うことができる。ただし、連続行動などの問題点を検討する必要がある。

枝刈りを付加した並行探索法は、次のようになる。下線部が、アルゴリズム1に付加した部分となる。

#### [アルゴリズム2] 枝刈り付並行探索法

(手順1)  $L = 0$ 、 $C = 0$ 、手番 先手。 MAX値、MIN値。

(手順2)  $L = L_{max}$ 、または、目標がなくなったときは、 $V$  局面評価値。かつ、 $L = L - 1$ 、すなわち、探索レベルを1段階戻し、手順9へ再帰リターン。

さもなければ手順3へ行く。

(手順3)  $C = 0$  ならば、手番 先手。  $C > 0$  ならば、手番 後手。

(手順4) 探索候補をリストアップする。未取得の目標に対する行動は、すべてが候補となり得るが、優先順に整列し、順位の高いいくつかを探索候補とする。

相手がすでに同一目標に対して行動しており、かつ、相手の方が先に到着する場合はリストに加えない。

相手が同一目標に対して行動開始はしていないが、今、相手が行動開始すると相手の方が先に到着できる場合は、目標に向かう中途の位置で停止するような行動(半行動)として取り扱う。

(手順5) 先頭の探索候補ならば、次を行う。

$M_b$  先頭の行動候補

先手番ならば、 $V_b = V_{MAX}$

後手番ならば、 $V_b = V_{MAX} - 1$ 。

(手順6)  $M$  未探索候補の1つ。

到着した先の目標の価値点を得点に加算し、目標を削除する。ただし、半行動の場合は、中途の座標に移動するだけで、価値点の加算も目標の削除も行わない。

(手順7) 先手番ならば、 $C = C +$  目標までの所要時間

後手番ならば、 $C = C -$  目標までの所要時間。

(手順8)  $L = L + 1$ 、すなわち、探索のレベルを1段階上げる。手順2へ再帰ジャンプ。

(手順9) 先手番ならば、 $V > V_b$  のとき、 $V_b = V$ 、かつ、 $M_b = M$

後手番ならば、 $V < V_b$  のとき、 $V_b = V$ 、かつ、 $M_b = M$ 。

(手順10-1)  $M$  を戻す。目標を戻す。得点を戻す。手番を戻す。 $C$  の値を戻す。

(手順10-2)

先手番ならば、

$V_b > =$  のとき、枝刈り(くり返しを停止し、手順13へ)。

さもなければ、手順10-3へ。

後手番ならば、

$V_b < =$  のとき、枝刈（くり返しを停止し、手順13へ）、さもなければ、手順10 - 3へ。

(手順10 - 3)

先手番ならば、 $V_b$

後手番ならば、 $V_b$ 。

(手順11) 未探索の候補があれば手順6へ、さもなければ手順12へ行く。

(手順12)  $V_{BEST}$ 、 $V_b$ 、 $M_{BEST}$ 、 $M_b$ 。

(手順13)  $L = 0$  ならば終了、 $M_{BEST}$  を最善行動とする。

さもなければ、 $L = L - 1$ 、すなわち、探索レベルを1段階戻して、手順9へ再帰リターン。

図において、右側の枝に後手側の連続行動が見られるが、後手側のノードに、 $>$  評価値が1つでも生ずると、図示した部分をすべて枝刈の対象とすることができる。本枝刈法を、以下、並行 法と呼ぶ。

図4に、アルゴリズム2に基づく枝刈の例を示している。

#### 4. 行動評価・局面評価

ゲームに勝利する力量は、同一の先読み探索アルゴリズムを用いたとしても、行動の価値判断、局面評価、作戦などによって支配される。ここでは、行動の価値判断、局面評価について論ずる。

##### 4.1 行動の優先順位

目標の数が多くなり全数探索が困難な場合は、未探索の目標に優先順位をつけて、順位の高い目標のみを探索する。優先順位は次のように決めるのが適当と考えられる。

相手との距離が遠い場合は、近い目標ほど優先する。

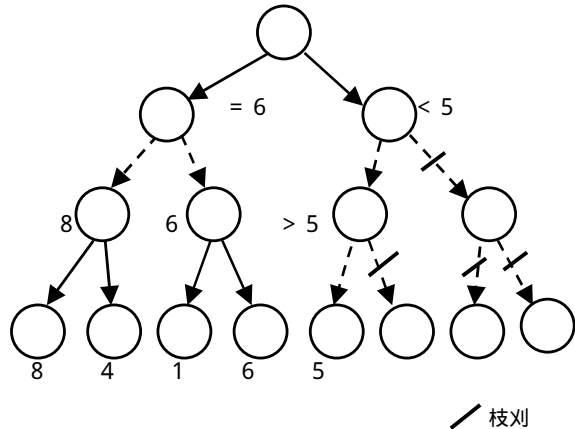


図4 並行 枝刈

相手との距離が接近した場合は、双方からの距離差が少なく、かつ、自分の方が相手よりも近い位置にある目標を優先する。

##### 4.2 局面評価

探索空間が広くて悉皆探索ができない場合は、先読みの最も深い局面（ゲームの木の葉）において局面評価を行い、評価値が自分に最も有利になるように、各ノードにおける行動の選択を行う<sup>(11)</sup>。

本ゲームモデルでは、局面評価に寄与する要因としては、次が考えられる。自分の既得点、相手の既得点、自分がこれから得点できる可能性（得点の期待値）、相手がこれから得点できる可能性。

本稿では、局面評価を次のように行う。

局面評価値 = 既得点差 + 得点の期待値の差

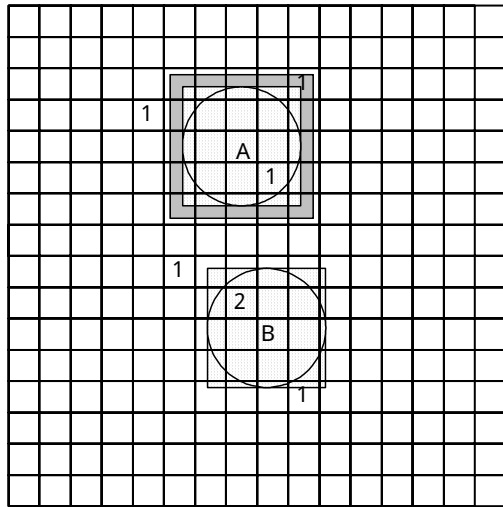
(1)

得点の期待値差 = (自分の勢力圏内にある目標の価値点の合計値 - 相手の勢力圏内にある目標の価値点の合計値) ×

(2)

以下、を局面評価係数とよぶ。その値は、多数回の対戦を実施して、経験的に決定する。

勢力圏は、[定義3]によれば、半径Rの円



- 勢力圏 (定義に基づく)
- 勢力圏 (近似計算:  $2R \times 2R$  分)
- 勢力圏 (走行時間差分)

図5 勢力圏の概念図

形となる。しかし、円形の内部か否かの計算は時間がかかるので、ここでは、方形で近似する。すなわち、プレイヤーの現在位置を中心とする  $2R \times 2R$  を勢力圏とする。

$R$  の値は、パラメータとして最適値を求める必要があるが、との相関が考えられるので、ここでは、一定とし、領域の寸法の  $1/8$  とする。

勢力圏の計算において、走行距離差をどのように反映させるかが、もう一つの課題となる。ここでは、次のように取り扱う。

走行距離差を  $d$  とすると、勢力圏の1辺を、 $2R + d$  とする。ただし、 $d > R$  のときは、 $d = R$  とする。

図5に、勢力圏の概念図を示す。ただし、 $R = (\text{領域の一辺の長さ}) / 8$  としている。

## 5 . シミュレーションプログラムの作成

提案したアルゴリズム、および、局面評価法に基づいて、プログラムの作成を行う。

### 5 . 1 プログラムの全体構成

通常の思考ゲームプログラムと基本構造は一致するが、本ゲームモデル特有の事項を列挙すると、次のようになる。

初期定義 :

フィールドの形状、目標の数・価値点、競争者の機能の定義、ハンデキャップ値。

フィールドの形状は、 $16 \times 16$  単位区画とする。目標数は、 $10 \sim 50$  個、目標の価値点は、 $1$  点  $\sim 3$  点とする。

対戦モード: 人対ロボット (先後の選択可能) 対ロボット対ロボット。

初期局面の生成: 擬似乱数を用いて目標の分布を決定する。擬似乱数発生関数は、松本によって開発された Twisted GFSR<sup>(14)</sup> を使用した。

情報の記憶 :

フィールド情報 (プレイヤー位置、目標の分布) メモリ、目標の未取得 / 取得に関するメモリ、行動の時系列を記憶するメモリを用意する。

各メモリは、1次元配列データ構造とする。

最善行動探索 :

提案した「並行探索法・枝刈付」を用いる。

探索候補は未探索候補の中から、距離の近い順に8個選択する。

相手の方が先に到着するときは、探索候補を半行動に修正する。

局面評価 :

提案した式 (1)、式 (2) を用いる。

局面表示: 次の2つを用意する。

DOS画面。

CGアニメーション。



上記 ~ が、そのまま、プログラムのゼネラルフローチャートとなる。

## 5.2 プログラムの作成結果

プログラムは、C++言語を用いて記述した。ただし、CGアニメーション部分はOpen GLを用いた。プログラム全体は、約1200ステップとなった。

図6に、図1の初期配置に対する実行結果を、CGアニメーション画面で示している。図において、下向きの三角形は先手プレイヤーを、上向きの三角形は後手プレイヤーを、方形は目標を、方形の寸法は目標の価値点を、目標間の直線はプレイヤーの軌跡を表している。

図6において、先手プレイヤーの第1行動は、自分の方が相手よりも近い距離にある目標の中で、自分から最も遠いものに向かっていて、相手との位置関係から見て、これが最善行動となる。この第1行動をとることによって、価値点の大きな2個の目標を獲得することができ、最終的に得点を最大にすることができる。この結果より、先読み探索・局面評価が正常に行われていることが分かる。

図7に、目標が41個の場合の結果を示す。図7の配置について、1回の行動決定に要する平均計算時間を約2秒と設定すると、枝刈なしの場合の限界探索深さ $L_{max}$ は6、並行枝刈付の場合の $L_{max}$ は10となった。なお、探索候補数は8個としている。

## 6. 対戦によるゲームの特性解析と評価

### 6.1 ゲームの特性解析

作成したプログラムを用いて、ロボット同士の多数の対戦を行い、ゲームを特徴づけるパラ

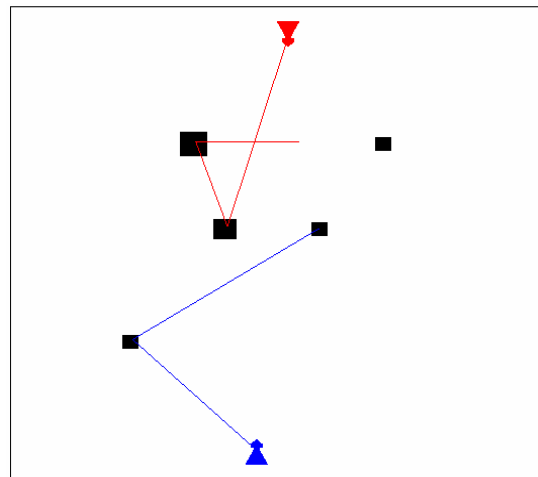


図6 初期配置1(図1)の場合の実行画面

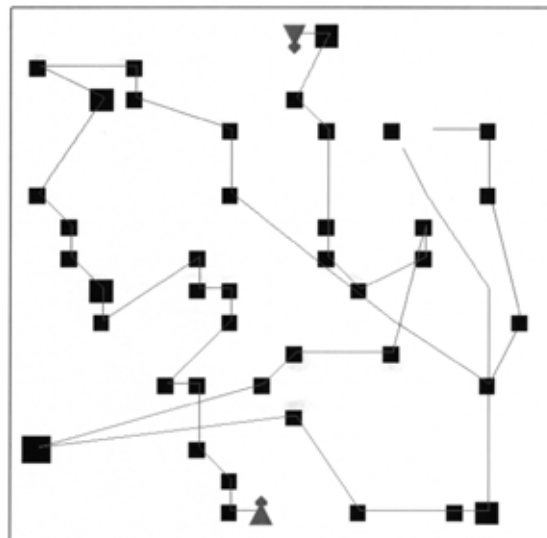
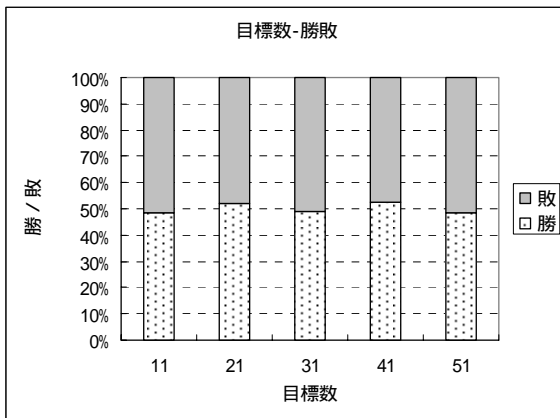


図7 目標数41個の場合の実行画面

メータの値を適切に決定する。本ゲームには、次のようなパラメータが存在する。フィールドの形状、目標の数・分布、プレイヤーの位置、1試合を何ゲームとするか。ハンデキャップ値H、局面評価係数、先読みの深さ、探索候補数。

フィールドの形状は、既述のように16×16単位区画とする。プレイヤーの位置は、図1の位置を基準とする。1試合は1ゲームとする。



(a) 目標の価値：全て1点

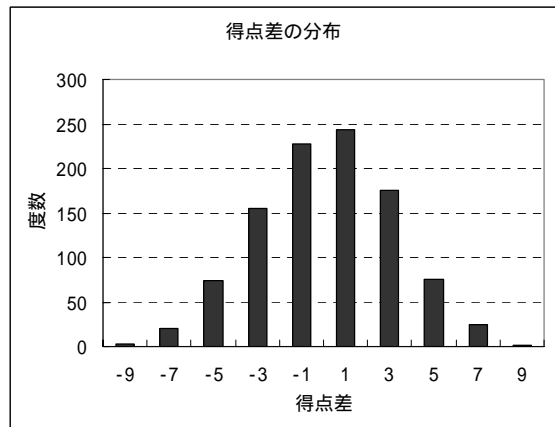


図9 得点差の分布1

先手より見た得点差、目標数：21、目標の価値：全て1点、先読み深さ：6、探索候補数：8、 $\beta = 0$

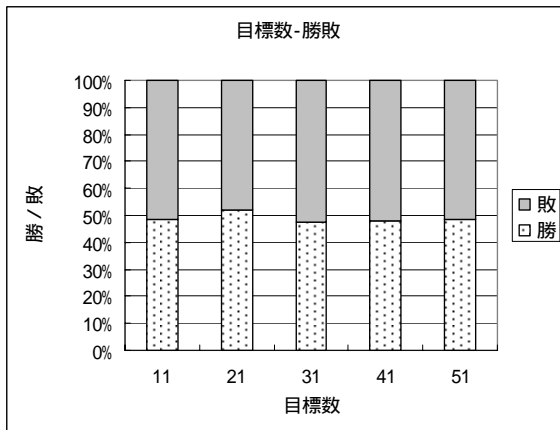
とし、先手、後手、同一条件としている。対戦回数は1000回とし、勝敗の表示は、先手側より見た値を示している。

図8より、勝敗が、 $\pm 2\%$ の範囲で、ばらつくことが分かる。勝敗のばらつきは、目標分布が同一でも、プレイヤーの位置、先読み深さなどの条件を変えると大きく変化する。従って、勝敗のばらつきは、目標の初期配置の偏りによるものではないことは明らかであり、目標を逐次的に収集していくというゲームの特性に起因すると考えられる。

図9は、1000回の対戦における得点差の分布の1例を示している。得点差は正規分布に類似の分布となることが分かる。なお、図9は、 $\beta = 0$ とした場合であるが、 $\beta$ を有効な値に設定すると、分布範囲が縮小されるという結果が得られている。

### (2) ハンデキャップHの設定

図10は、先手、後手の位置を基準位置とした場合と、位置交換した場合の結果を示す。図より、先手・後手の位置を交換すると、得点



(b) 目標の価値 3点：1個、2点：4個、1点：残りの全て

図8 目標の分布を変えた対戦結果

### (1) 目標の分布を変えた対戦

図8に、乱数を用いて様々な初期局面を生成し、対戦した結果を示す。

目標数は11～51個としている。奇数としたのは引き分けをなくすためである。目標の価値点は、次の2つのパターンとする。

1色パターン：全ての目標の価値点を1点とする。 3色パターン：3点の目標を1個、2点を4個、残り全てを1点とする。

以下、価値パターンは、1色、3色と簡略化して表記する。

先読みの深さは6、探索候補数は8、 $\beta = 0$

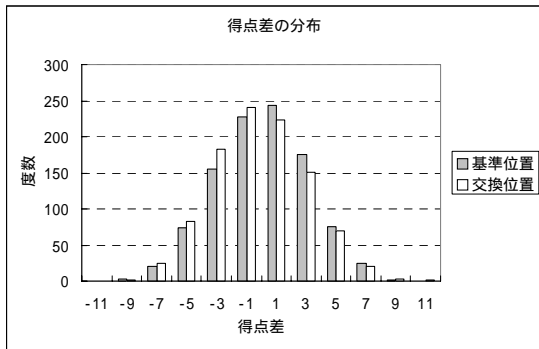


図10 得点差の分布2  
プレイヤーの位置を交換した場合

目標数：21、目標の価値：1色、先読み深さ：6

差も勝敗も入れ替わることが分かる。これより、先手・後手の間に優劣が存在しないと考えられる。従って、ハンデキャップ $H = 0$ が適当なことが分かる。

### (3) 局面評価係数 の設定

図11に、先手側の の値を変えて対戦した結果を示す。後手側は、常に、 $= 0$ としている。図11より、 の増大とともに勝率が向上することが分かる。このことは、提案した局面評価方法が有効なことを示している。また、の値は、1程度が適当であることが分かる。

## 6.2 評価

提案したアルゴリズム・局面評価法と他のアルゴリズムとの対戦を行い、評価を行う。

本ゲームモデル用に特別に開発された他のコンピュータアルゴリズムは存在しないので、対戦相手として、貪欲法<sup>(15)</sup>(本ゲームモデルでは、最も近い目標に向かう方法)、ヒューリスティックアルゴリズム(人間)を選ぶ。「貪欲法」は、経路探索において、最適解は得られないが、実用的な解を求めるのには有効とされている。

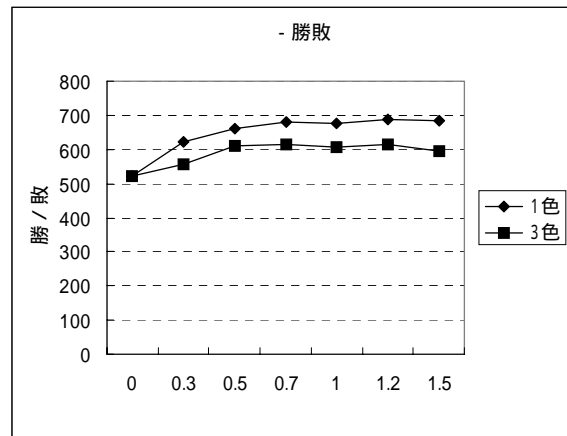


図11 局面評価係数 を変えた対戦結果  
先手側のみ を変化、後手側は  $= 0$ 、目標数：21、先読み深さ：6、探索候補数：8

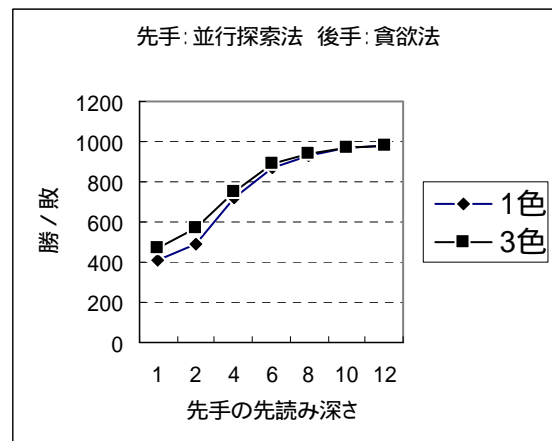


図12 貪欲法との対戦結果

先手：並行探索法、 $= 1$ 、探索候補数：8、  
後手：貪欲法、目標数：21、対戦数：1000回

### (1) 貪欲法との対戦

貪欲法のプログラムは、作成したプログラムにおいて、先読み深さ=1、探索候補数=1と設定することによって得られる。

なお、本研究の方法側は、相手が貪欲法を用いるとして先読みをすることとする。

図12に、対戦結果を示す。先手を本研究の方法、後手を貪欲法としている。

図12より、次が分かる。

[結果1] 本研究の方法、先読み深さ12とすると、勝率9割8分に達する。しかし、10割とはならない。

[結果2] 先読み深さ1~2では、貪欲法の方が、勝率が高い場合がある。

結果1は、本研究の方法が有効であることを示している。しかし、同時に、勝率10割が得られるように、局面評価法/作戦について検討の余地があることも示している。

結果2が得られる理由を調べると、先読み深さが浅い場合、勢力圏に基づく局面評価によって、かえって不適切な行動を選択する場合が生ずることが分かる。

## (2) ヒューリスティックアルゴリズム(人間)との対戦

対戦相手の人間は、工学部に在籍する大学生5名とした。本ゲームモデルは、ルールが簡単で誰でも即座にプレイできる。従って、対戦に当たって、数十回の予備的対戦以外は、特別な訓練は行っていない。

表1に、対戦結果を示す。表1より、次が分かる。

[結果3] 本研究の方法は、最強の条件下で、対戦相手の人間に対して、勝率6割~7割となる。

表1には、貪欲法と人間の対戦結果も示している。貪欲法との対戦では、人間に、相手が貪欲法であることを知らせている。表1より、人間の貪欲法に対する勝率は6割~8割となることが分かる。この数値を、図12に当てはめると、対戦相手の人間のプレイ能力は、本研究の方法の先読み深さ4のレベルに相当すると判定される。

表1 人間との対戦結果

本研究の方法対人間の場合

先手：本研究の方法、先読み深さ：12、

探索候補数：8、 $\beta = 1$

後手：人間(工学部大学生)

貪欲法対人間の場合、先手：人間、後手：貪欲法

目標数：21、3色、対戦数：20回。勝敗は、本研究の方法/貪欲法より見た値を示す。

		本研究の方法	貪欲法
対戦相手	人間I	16勝4敗	8勝12敗
	人間K	15勝5敗	6勝14敗
	人間S	13勝7敗	8勝12敗
	人間M	18勝2敗	14勝6敗
	人間Y	15勝5敗	7勝13敗

## 6.3 考察

アルゴリズムについて：

提案した並行探索法は有効に機能することが確認された。

行動の優先度評価を、予備的先読みを導入することにより、より適正化できる。これにより、探索能力の向上を図ることができる。

一人経路探索に有効とされる蟻コロニー探索が本モデルに適用できないか検討する価値がある。

局面評価/作戦について：

領域全体を見渡し、目標の集中する場所に先行するような局面評価法/作戦の導入が必要である。

ゲームモデルについて：

プレイヤー数を2としたが、プレイヤー数3以上の多人数ゲームモデルへの拡張も有意義と考えられる。提案した先読み探索アルゴリズムは、

多人数への拡張が比較的容易であると思われる。

局面評価を適切に修正することにより、競争を協調に変更できる可能性がある。競争/協調を組み合わせた多人数ゲームモデルへの発展も有意義であると考えられる。

本稿では完全情報ゲームモデルとしたが、センサ入力を導入した不完全情報ゲームモデルへと発展させる必要がある。情報収集過程と目標収集過程を混在させた探索アルゴリズムとなる。

## 7. むすび

本稿では、自律的に移動しながら、ある領域に散在する目標を争って収集するタイプの競争ロボットを取り上げ、本競争モデルに適した先読み探索アルゴリズムとして、目標に直行する動作を基本とし、相手が有利な場合は中途まで前進するという探索アルゴリズム(並行探索法)を提案した。また、勢力圏という概念に基づく局面評価法を提案した。アルゴリズム・局面評価法に従ってプログラム作成を行い、有効に機能することを確認した。

今後、次のような課題について検討する予定である。

目標分布の全体を把握する局面評価法の提案

先読み探索プロセッサの開発

多人数ゲームモデルへの発展

協調/競争混合ゲームモデルへの発展

不完全情報ゲームモデルへの発展

## 参考文献

(1) 亀山、樋口、"ロボットとVLSIコンピュータ"、日本ロボット学会誌、Vol. 6, No. 4, pp. 64-70, (1988-4).

(2) 辻、江尻、ロボット工学とその応用、電子情報通信学会、(1990).

(3) 太田、倉林他、知能ロボット入門、コロナ社、

(2001).

(4) 有本卓、知能科学 - ロボットの知と巧みさ -、コロナ社、(2007).

(5) 大須賀節雄監修、人工知能大辞典、丸善、(1991).

(6) 大沢、沼岡、石田、"分散人工知能における標準的小問題"、コンピュータソフトウェア、Vol. 10, No. 3, pp.1-21, (1993-03).

(7) 小林、細江、"マルチエージェントシステムの自律分散制御 - 群移動ロボットによる協調捕獲行動"、計測と制御、Vol.46, No. 3, pp. 178-184, (2007-03).

(8) 苫米地宣裕、"競争巡回知能ロボット制御用先読みエンジンの設計"、計測自動制御学会東北支部研究集会資料、175-4, (1998-6).

(9) 苫米地宣裕、"競争巡回知能ロボットとその先読み探索アルゴリズム"、八戸工業大学情報システム工学研究所紀要、Vol. 12, pp.1-5, (2000-3).

(10) 山本、久保、巡回セールスマン問題への招待、朝倉出版、(1997).

(11) 松原、竹内、ゲームプログラミング、共立出版、(1997).

(12) N. Tomabechi, T. Fujiieda, Y. Fujioka, "Design of a search processor for competitively traveling robots", Proc. of Mediterranean Conference on Control and Automation, MoM1-A-5, pp.1-6, (2004-6).

(13) 小谷、吉川他：コンピュータ将棋、サイエンス社、(1990).

(14) M. Matsumoto, Y. Kurita, "GFSR generators", ACM Trans. on Modeling and Computer Simulations 2, pp.179-194, (1992).

(15) 島内他、アルゴリズム辞典、共立出版、1998.