

## 距離コミットメントによる 移動目標探索

3U-6

吉田智一 中西正和

慶應義塾大学 理工学研究科 計算機科学専攻

### 1. はじめに

目標の移動を追従する探索アルゴリズムとして移動目標探索 (MTS : Moving Target Search) アルゴリズム [Ishida 92] がある. Ishida は MTS アルゴリズムの性能改善のために, 目標の移動をある期間あえて無視する“コミットメント”を提案している. 本研究ではこの“コミットメント”を拡張した“距離コミットメント”を提案しその性能特性を評価する.

### 2. MTS

#### 2.1 問題の定義

MTS の問題空間は, すべての枝が同じコスト 1 を持つ連結グラフで表される. 問題解決器と目標は交互に動作し, 問題解決器は常に目標の状態 (位置) を知ることができる. ただし目標は時々移動しないことによって, 問題解決器の速度が目標のそれを上回るように設定されている. 問題解決器と目標が同じ状態に位置したとき, アルゴリズムの実行は完了する [Ishida 92].

#### 2.2 コミットメントを導入した MTS アルゴリズム

Ishida の提案は, “問題解決器が着実に推定距離を縮めながら移動する場合には目標に追従し, 問題解決器が推定距離の凹部に位置する場合には目標にコミット (目標の移動を無視) する” というものである. 問題解決器の状態を  $x$ 、目標の状態を  $y$  とすると, その間の推定距離は  $h(x, y)$  で表される. 推定距離が連続して減少する回数を  $D_c$ , 問題解決器を目標の変化に追従させる閾値を  $d_{com}$  としたとき, コミット

メントを導入した MTS アルゴリズムは問題解決器と目標が同じ状態になるまで以下の処理を繰り返す.

#### 問題解決器が移動する場合

1.  $h(x', y)$  を  $x$  に隣接するすべての  $x'$  に関して計算.
2.  $h(x, y) > \min_{x'} h(x', y)$  のとき,  $D_c \leftarrow D_c + 1$ .  
 $h(x, y) \leq \min_{x'} h(x', y)$  のとき,  $D_c \leftarrow 0$ .
3. 以下のように  $h(x, y)$  を更新.

$$h(x, y) \leftarrow \max \left\{ \begin{array}{l} h(x, y) \\ \min_{x'} \{h(x', y) + 1\} \end{array} \right\}$$

4. 最小の  $h(x', y)$  を持つ  $x'$  に移動. 最小のものが複数個あれば, そのなかでランダムに選択.

#### 目標が移動する場合

$D_c \geq d_{com}$  または  $x = y$  である場合に限り以下を行う.

1. 目標の新しい位置  $y'$  に対して  $h(x, y')$  を計算.
2. 以下のように  $h(x, y)$  を更新.  $y$  から  $y'$  までの移動回数 (問題解決器が目標を無視した回数) を  $t$  とする.

$$h(x, y) \leftarrow \max \left\{ \begin{array}{l} h(x, y) \\ h(x, y') - t \end{array} \right\}$$

3. 目標を再設定.

またこのアルゴリズムにおいて  $d_{com} = 0$  とすると従来の MTS アルゴリズムと同等のアルゴリズムとなる.

### 3. 距離コミットメント

Ishida の手法によるコミットメントでは性能が 4~8 倍程改善されることが知られている. 本研究ではこのコミットメントの特徴および問題点を考慮し“距離コミットメント”を提案する. これは  $d_{com}$  を大きめ

Distance Commitment Method for Moving Target Search problems

Tomokazu YOSHIDA

Masakazu NAKANISHI

Department of Computer Science, Graduate School of Science and Technology, Keio University 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa 223, Japan

にとり、問題解決器のコミットしている位置と実際の目標の位置との距離  $D_d$  が閾値  $d_{far}$  以上になると問題解決器を目標の変化に追従させる。また逆に距離が閾値  $d_{near}$  以下の間はコミットし続ける。さらに問題解決器と目標との距離  $D_a$  が閾値  $d_{act}$  以内になると Ishida の手法と同様の反動的な探索を行う。距離コミットメントを導入した MTS を以下に示す。問題解決器が移動する場合には、Ishida の手法と同じアルゴリズムである。

#### 目標が移動する場合

$D_a \leq d_{act}$  なら Ishida の手法を そうでない場合以下のように目標をコミットする。

1.  $x = y$  または  $D_d \geq d_{far}$  であれば目標の変化に追従。
2.  $D_d \leq d_{near}$  であれば目標の変化を無視。
3.  $D_c \geq d_{com}$  であれば目標の変化に追従。
4. 以上のどれでもなければ目標の変化を無視。

#### 4. 実験結果

各 MTS アルゴリズムの性能を同一の実験環境を用いて評価した。問題空間は両端をトーラス状に結合した  $100 \times 100$  の 2次元格子とし、格子点上に障害物を配置可能とした。図 1 は Ishida 手法によるコミットメント、図 2 は距離コミットメントによる MTS の性能を示す実験結果である。目標の動作はプログラムによって制御し以下の 3 種のモードを設けた。ただし目標の移動速度は問題解決器の 50%。つまりランダムに 2 回に 1 回は移動しないとす。

- Avoid: 目標は問題解決器から遠ざかる。
- Meet: 目標は問題解決器に近づく。
- Random: 目標はランダムに移動する。

#### 5. 評価

実験結果より距離コミットメントによる MTS アルゴリズムの特徴をまとめる。Ishida のコミットメントの性能を決定するのは閾値  $d_{com}$  である。この手法では  $d_{com}$  が小さすぎると性能が劣化する。これは大きな推定距離の凹部を目標をコミットしている

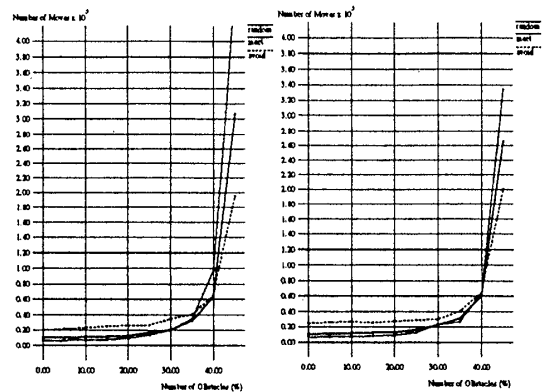


図 1:  $d_{com} = 10$

図 2:  $d_{far} = 20$

間では埋めつくすことができないからである。逆に  $d_{com}$  が大きくすぎても性能が劣化する。問題解決器が目標をコミットし続けている間に目標が大きく動き、目標の変化に敏感に反応できないからである。

距離コミットメントによる MTS アルゴリズムでは、 $d_{com}$  を適切にとった場合の Ishida のコミットメントより、障害物の比率が低いときは性能が数%劣化する。ただし、 $d_{com}$  を大きくとった場合のコミットメント程ではない。これは閾値  $d_{far}$  によって大きな目標の変化を感知しているからである。また障害物の比率が高いときには、 $d_{com}$  を適切にとった場合のコミットメントよりも性能を改善させている。これは障害物の比率が高いときには、目標も容易には移動することができず、問題解決器が目標の移動を無視し続けて目標に接近できるからである。

#### 6. おわりに

以上の評価から距離コミットメントによる MTS アルゴリズムの利点が明らかになった。問題点としては適切な閾値  $d_{com}$ ,  $d_{far}$ ,  $d_{near}$ ,  $d_{act}$  を容易に決定できないことであるが、他の移動目標探索においても有利な手法であると思われる。

#### 参考文献

- [Korf 90] Korf, R. E., Real-Time Heuristic Search, *Artificial Intelligence*, Vol. 42, No. 2-3, pp. 189-211, 1990.
- [Ishida 92] Ishida, T. and Korf, R. E., Moving Target Search with Intelligence, *Proceedings of the Tenth National Conference on Artificial Intelligence*, AAAI, pp. 525-532, 1992.