

# ルールベース蟻コロニー最適化に基づく 自律行動型ロボットの知能化

M2008MM032 山本勇作

指導教員：高見勲

## 1 はじめに

本研究の目的は、自律行動型ロボット知能化である。知能化とは、知識がない状態から学習を開始し、行動過程から得た経験を基に知識を蓄積しながら、より正確な判断を行うことである。その検証として迷路問題を例にとり、Ant Colony Optimization(ACO)[1] にウィルス感染[2] や Particle Swarm Optimization(PSO)[3] などの学習機能を融合させたアルゴリズムを提案する。本研究の特徴は以下の点である。

1. ACO にルールを適用し、探索回数の低減を図る。
2. ACO に PSO, ウィルス感染の概念を融合させ、局所解の回避を図る。
3. エントロピー概念を導入し、収束性を評価する。

有用性は以下の点である。

1. 探索領域のマップを作成しながら学習的に最短経路問題を解く。
2. 進行方向を決定する手段として、確定的要素(ルール)と確率的要素のハイブリッド進路決定法の提案により、ACO を環境認識し、最短経路を学習する過程に適用する。
3. 局所解に陥らず、大域的最適解を求める手段としてウィルス感染による部分空間探索を適用する。

ダイクストラ法などの確定的方法に比べて、ACO が必要となる問題は以下の点である。

1. 地図情報がない状態からスタートして、地図情報を獲得しながら、最短経路を求める問題。
2. 一旦得られた地図情報が時間と共に変化する条件が動的な問題。
3. 経路長だけでなく、目的地方向のベクトルや、進路の変更回数、価値観(例えばカーナビゲーションにおいて、運転手にとって右折より左折の方がストレスが少ない)などの複数の評価を考慮する必要がある問題。

## 2 本研究での蟻コロニー最適化手法

### 2.1 ルールベース蟻コロニー最適化

本研究では図1のような迷路を想定する。1マス座標  $(x, y)$  としてとらえ、その座標をノードとし、白いマスは通行可能なノード、黒いマスは通行不可なノードとし、この集合をマップとする。マップを探索する蟻の行動では、前後左右のノードにのみ移動できるものとする。また蟻のフェロモンは、従来法ではパスにフェロモンを与えているが、本研究ではノードに与える。スタート座標  $S$  から蟻  $m$  匹を放ち、ゴール座標  $G$  に全ての蟻が到

達する過程を1試行とし、それを  $n$  回繰り返す。

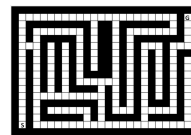


図1 横25縦17の迷路

ACO を図1のような迷路に適用するために、以下のようルールを制定する。

ルール1 通過可能なノードを判別し、式(1)によりマップを作成する。

ルール2 進行方向の選択として、各ノードごとに通過した回数をカウントし、式(2)により、同じ経路を何回も巡回する事を禁止する。

図2は、本研究で採用している基本的なアルゴリズムの流れである。

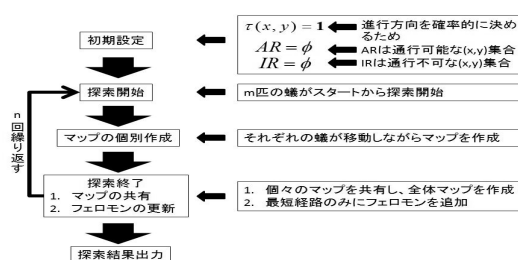


図2 アルゴリズムの流れ

通行可能な  $(x, y)$  集合を  $AR$ (Admissible Region) とし、通行不可な  $(x, y)$  集合を  $IR$ (Inhibited Region) とする。

式(1)はマップを作成する際、通行可否なノードであるか判定し、学習する式である。 $(v, w)$  は進行方向を操作するための変数であり、取りうる値は  $v, w = -1, 0, 1$  で、 $|v| + |w| = 1$  を満たす。

$$\begin{cases} \text{if } (x + v, y + w) \text{ が通行可能なノード} \\ \text{then } (x + v, y + w) \text{ を } AR \text{ の要素に加える} \\ \text{if } (x + v, y + w) \text{ が通行不可なノード} \\ \text{then } (x + v, y + w) \text{ を } IR \text{ の要素に加える} \end{cases} \quad (1)$$

式(2)により、ループ探索を禁止している。また、式(2)は各試行回数ごとの各蟻全てに対して別々に定義する。式(2)において、最小値が複数ある場合は、式(3)で確率的に進行方向を決定する。移動方向  $(v, w)$  の選択は、移動先のフェロモン濃度に比例する。

$$\min_{\substack{v, w \\ (x+v, y+w) \in AR}} PC(x + v, y + w) \quad (2)$$

$PC(x, y)$  は、ノード  $(x, y)$  を通過した回数である。ノード  $(x, y)$  から  $(v^*, w^*)$  の方向に行く確率  $p_{v^*, w^*}(x, y)$  は式 (3) で表す。

$$p_{v^*, w^*}(x, y) = \begin{cases} \frac{\tau(x + v^*, y + w^*)}{\sum_{v, w} \tau(x + v, y + w)} & (x + v^*, y + w^*) \in AR \\ 0 & (x + v^*, y + w^*) \in IR \end{cases} \quad (3)$$

$\tau_{x, y}$  は、ノード  $(x, y)$  のフェロモン濃度を示す。

試行回数  $t$  回目において、全ての蟻がゴールまで辿り着いたら、式 (4) によってフェロモンの更新を行う。試行回数  $t$  回目の最短経路を発見した蟻だけにフェロモンを与える。これにより収束の高速化を促進している。

$$\tau_{x, y}(t + 1) = \rho \tau_{x, y}(t) + \Delta \tau_{x, y} \quad (4)$$

$\rho$  は蒸発に関するパラメータである。 $\tau_{x, y}$  はフェロモン増量であり、式 (5) のように定義する。

$$\Delta \tau_{x, y} = \begin{cases} 1/L & (x, y) \in BR \\ 0 & (x, y) \notin BR \end{cases} \quad (5)$$

$L$  は試行回数  $t$  回目の最短経路を見つけた蟻の経路長である。 $BR$  は試行回数  $t$  回目で得られた最短経路上の  $(x, y)$  集合であり、最短経路を発見した蟻だけにフェロモンを与えている事を示している。

## 2.2 ウィルス感染による部分空間探索

スタート地点からゴール地点までの二地点間最短経路問題を解く場合、探索領域が大きくなると二地点間のノードが多数になり、局所解に陥りやすい。ACO の場合、大域的最適解を発見する前に局所解のフェロモン濃度が高くなってしまふことが問題である。局所解に収束することを回避するために、ウィルス感染の考え方を ACO に適用する方法を提案する。特徴は、部分空間をフェロモン濃度に関係なく徘徊するフェロモン鈍感期を設けた点である。以下にアルゴリズムの流れを示す。

step1 蟻の集団をウィルス感染集団と非感染集団に分け、試行回数  $z$  回ごとにウィルスに感染させる。

step2 非感染集団は通常通りの探索を続け、ウィルス感染集団は部分空間をそれ以前のフェロモン濃度に関係なく探索する。

step3 部分空間の形成方法は、それまでの最短経路を利用する。経路上の 2 点をランダムで選択し、その 2 点を対角とした 4 角形を形成する。部分空間でデッドロックになってしまった場合は、その試行に関する情報を破棄する。

## 2.3 PSO による学習機能強化

フェロモンの与え方について、ACO の基本モデル Ant System(AS) では全ての蟻の経路にフェロモンを増加させ、本研究では試行回数  $t$  回目の最短経路を発見した蟻だけにフェロモンを与える方法を提案した。さらなる学習機能の向上を考えた場合、過去の行動履歴に従って動的に

フェロモンを調整することが考えられる。そこで、PSO との融合アルゴリズムを提案する。

ACO において、Particle とは各蟻を示し、各蟻が試行回数  $t$  までに学習した最短経路を  $pbest$ 、群全体が試行回数  $t$  までに学習した最短経路を  $gbest$  とし、式 (6)(7)(8) を定義する。

$$\tau_{x, y}(t + 1) = \rho \tau_{x, y}(t) + C_1 \cdot rand_1 \sum_{k=1}^m \Delta \tau_{x, y}^{k-pbest} + C_2 \cdot rand_2 \Delta \tau_{x, y}^{gbest} \quad (6)$$

$C_1, C_2$  は重み、 $rand_1, rand_2$  は 0~1 の一様乱数、 $k-pbest$  は蟻  $k$  が試行回数  $t$  までに発見した最短経路である。

$$\Delta \tau_{x, y}^{k-pbest} = \begin{cases} 1/L^{k-pbest} & (x, y) \in k-pbest-route \\ 0 & (x, y) \notin k-pbest-route \end{cases} \quad (7)$$

$L^{k-pbest}$  は蟻  $k$  が試行回数  $t$  までに発見した最短経路長、 $k-pbest-route$  は蟻  $k$  が試行回数  $t$  までに発見した最短経路の座標集合である。

$$\Delta \tau_{x, y}^{gbest} = \begin{cases} 1/L^{gbest} & (x, y) \in gbest-route \\ 0 & (x, y) \notin gbest-route \end{cases} \quad (8)$$

$L^{gbest}$  は群全体が試行回数  $t$  までに発見した最短経路長、 $gbest-route$  は群全体が試行回数  $t$  までに発見した最短経路の座標集合である。

## 3 エントロピーの定義と解の収束性評価

進化計算法の一般的な評価方法は、求めた解と最適解の誤差率、適度の比較や、最適解を求めるまでの時間などを比較している場合が多い。進化計算法において、多様性と収束性のバランスを保つことが必要不可欠であり、それによって誤差率や計算時間も大きく改善される。そこで、エントロピーを観測することで ACO の収束特性を解析する。

巡回セールスマン問題における ACO の収束特性については、フェロモン濃度情報のエントロピー変化を観測して解析している [4]。本研究では、学習経過の情報はフェロモン濃度  $\tau_{x, y}(t)$  に保持されている。 $\tau_{x, y}(t)$  のエントロピーを式 (9) 式 (10) と定義する。

$$E(t) = - \sum_x \sum_{y \neq x} p_{x, y}(t) \log p_{x, y}(t) \quad (9)$$

$$p_{x, y}(t) = \frac{\tau_{x, y}(t)}{\sum_x \sum_{y \neq x} \tau_{x, y}(t)} \quad (10)$$

$E(t)$  の最大は、フェロモン濃度が一様に分布している  $\tau_{x, y}(0)$  のときで、その値は  $\log n$  である。 $n$  は通過可能なノードの数である。そこで、式 (11) のように正規化エントロピー  $E_N(t)$  を用いる。 $E_N(t)$  は、 $[0, 1]$  の区間の値となる。

$$E_N(t) = \frac{E(t)}{\log n} \quad (11)$$

エントロピーの解析方法について、フェロモンが均一に分布している時はエントロピーが高く、多様性がある状態である。また、フェロモン分布が片寄っている時はエントロピーが低いことを示し、解が収束している状態である。さらに、エントロピーの減少勾配を観測することで収束の速さを解析することができる。

#### 4 シミュレーション解析

シミュレーションには Intel(R)Core(TM)2 Duo CPU T8300 2.4GHz と 4GB のメモリを持つ計算機を使用した。

##### 4.1 マップ作成の検証

蟻がマップを作成しながら経路探索を進める過程を図 3 に示す。本来の ACO は、多くの蟻に経路探索させ、群体行動で最短経路を導くが、マップ作成過程を把握するために 1 匹の蟻で検証する。検証環境は、図 1 において、左下のスタート座標  $S$  からゴール座標  $G$  までを探索するまでを 1 試行とし、100 回繰り返す。最短経路長 60 である。黒い点は、それまでの全試行の中での最短経路である。アルゴリズムは本研究が提案するルールベース蟻コロニー最適化である。図 3 を解説する。図 3(A) は 1 匹の蟻が 1 回目の試行を終えた図である。通過可能な領域を判別して徘徊し、通過した場所のみのマップを出力している。行き止まりの経路に進入し、戻って来ている。最短経路ではない。図 3(B) は 2 回目の結果表示である。(A) とは異なる経路を通り、(A) よりもマップ作成は進んでいる。しかし、最短経路は発見できていない。図 3(C) は 30 回目の結果表示である。マップの作成は終了している。経路の色が濃いほどフェロモンの濃度が高い事を示している。フェロモン濃度が最も高い経路は最短経路ではない。図 3(D) は 100 回目の結果表示である。最短経路に収束していることが分かる。

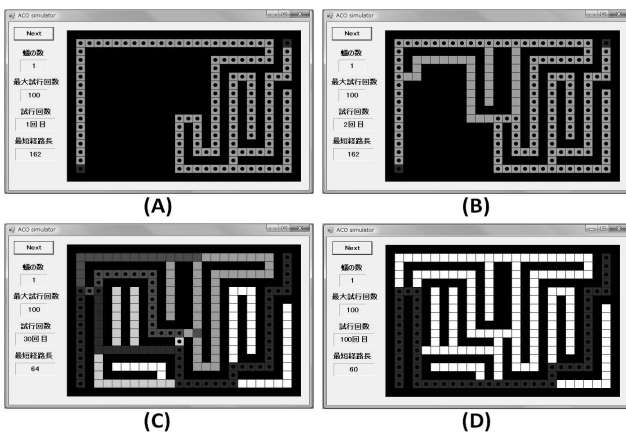


図 3 マップ作成検証

##### 4.2 ルールベース蟻コロニー最適化の検証

確定的要素(ルール)と確率的要素のハイブリッド進路決定法の有効性を検証する。検証環境は、図 1 のような迷路を縦 130 横 82 に拡大した迷路を使用する。最短経路は 1 通りしかなく、最短経路長は 412 である。大きなマップで検証する理由は、探索領域が広がっても実用的な時間内で解を求めることができるか検証することであ

る。蟻 10 匹を試行回数 100 回繰り返し、それぞれの試行の最短経路を出力したグラフが図 4 である。

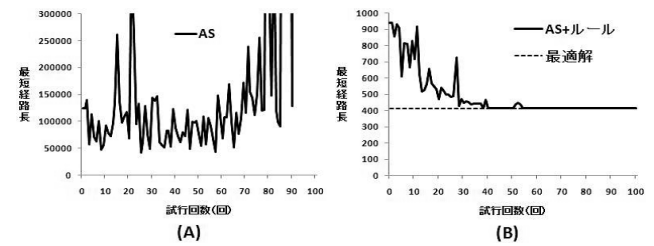


図 4 ルールの有用性検証

図 4(A) は基本モデル AS のみのグラフである。計算時間は 4 分 11.615 秒で、全体の最短経路長でも 41998 で最適解 412 には程遠く、試行回数 70 回目辺りから発散する傾向も見られる。経路長が長くなる理由として、均一に分布された初期状態から進行方向を確率式だけで決める、前進後退を繰り返すような移動になり、またフェロモンの増加量は距離に逆数になるので微小で、最短経路への収束の速度が小さいことが原因である。図 4(B) は基本モデル AS に確定的要素(ルール)を適用させたときのグラフである。計算時間は 0.658 秒で、確定的要素の適用で計算時間の改善が見られる。初めは大域的に探索し、試行回数 40 回目辺りから最適解に収束していることがわかる。ルールが有効に働き、前進後退を繰り返すような移動を回避し、さらに通路の通過回数が少ない方向に移動するため、大域的探索を可能にしている。

図 5 は、図 4 におけるエントロピーの変化である。

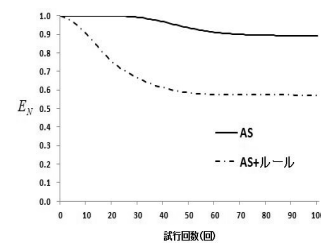


図 5 収束特性の比較

基本モデル AS は試行回数を重ねてもエントロピーが 0.9 で収束してしまっているので、解の収束性が欠如し何回繰り返しても最適解への収束は見られない。本研究のアルゴリズムは、初期から徐々にエントロピーが収束し始め、最適解を発見した後もある程度高い値を保持している。これにより、探索領域がリアルタイムに変化する場合にも対応できる。

##### 4.3 ウィルス感染による部分空間探索の検証

ウィルス感染による部分空間探索の有用性を検証する。検証環境は図 1 のような迷路を縦 130 横 82 に拡大した迷路を使用する。スタートからゴールまでの経路が多数存在する迷路であり、最短経路長は 208 である。このようなマップを使用することにより、局所解に陥りやすい環境でもウィルス感染による部分空間探索が有効的に働くことを示す。蟻 100 匹を試行回数 100 回繰り返し、その過程を 10 回行い、平均値を取り、最適解との誤差率を出力させたグラフが図 6 である。

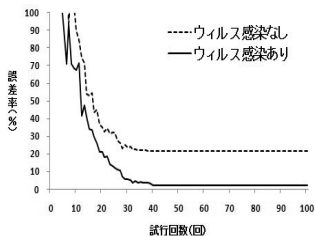


図 6 最適解との誤差率

ウィルス感染がない場合は局所解に陥っていることが多かったため、最終的な誤差率が高くなっているが、ウィルス感染ありの場合は、局所解に陥る場合があるが、前半の試行回数から効率的に学習を行い、素早く最適解付近に収束している事が分かる。改善された理由は、局所解に陥った場合、ウィルス感染による部分空間探索によって、ある区間を集中的に探索できるため、局所解を回避できたためであると考えられる。

#### 4.4 リアルタイム変化迷路に対する検証

あらかじめマップの情報が得られており、マップが変化しない場合の最短経路を求めるならば、ダイクストラ法などの確定的方法が有効である。しかし、リアルタイムにマップが変化する場合、最初から計算し直す必要がある。知識の蓄積による利点として、探索領域が変化した場合でも学習の継続ができる点である。

図 7 は迷路の拡大図で、左下から右上までのパスについて、マップを変化させる前後の出力表示である。試行回数 50 回、70 回を終えた時点で迷路を変化させる。図 7(A) は 50 回目、図 7(B) は 55 回目、図 7(C) は 75 回目の結果表示である。

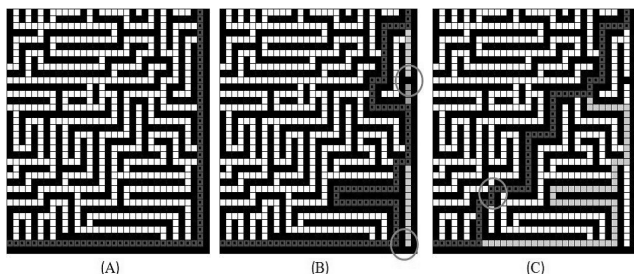


図 7 リアルタイム変化迷路の検証

図 7(A) は 50 回目までにフェロモンが最短経路上に収束している。図 7(B) は 55 回目の結果表示である。50 回目終了時点で最短経路上の 2 箇所を通行不可にする。様々なパターンの迂回路を探索し、最短迂回路にフェロモンが収束している。図 7(C) は 75 回目の結果表示である。70 回目終了時点で、それまでの最短経路を更新するように離れた場所を 1 箇所通行可能にする。最短経路上を通行不可にして迂回路を探索するよりも、こちらの変化の方がフェロモン収束後の多様性を必要とされる。この検証により、試行回数が多くても多様性を維持し、常に新しい経路と最短経路を探索していることが分かる。

#### 4.5 ACO と PSO の融合アルゴリズム検証

図 6 において、本研究が提案するウィルス感染による部分空間探索でも全て最適解が発見できず、最適解との

誤差が生じている。これは、ウィルス感染で部分空間に区切る際、ランダムで最短経路上の 2 点を選択するため、局所解に陥っている空間を部分空間探索できないためであると考えられる。そこで、ACO と PSO の融合アルゴリズムを同じ迷路に適用し、ウィルス感染による部分空間探索と性能を比較する。蟻 100 匹を試行回数 200 回繰り返す、その過程を 10 回行い、平均値を取り、誤差率とエントロピーを図 8 に示す。

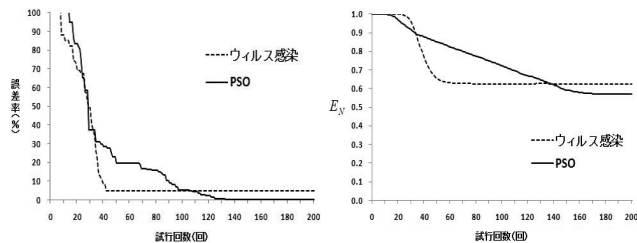


図 8 PSO とウィルス感染の比較

PSO を適用した場合、多様性を維持したまま学習が進み、誤差なく最適解に全て収束している。ウィルス感染に比べ、PSO を適用した場合はエントロピーの減少が緩やかで、収束速度はウィルス感染に劣るが、多様性の維持は十分見てとれる。

## 5 おわりに

本研究で得られた成果は以下のとおりである。

1. 蟻コロニー最適化手法を迷路探索する行動過程に適用した。
2. マップを作成しながら学習的に最短経路を求める方法を提案し、通路や壁などの初期値を与えなくても探索を可能にした。
3. 蟻の進行方向の選択法として、確定的要素(ルール)と確率的要素のハイブリッド進路決定法を提案し、迷路を効率的に探索することを可能にした。
4. ウィルス感染による部分空間探索法を提案し、大域的探索と最適解への素早い収束性を実現した。
5. ACO と PSO の融合アルゴリズムを提案し、過去の情報を学習機能に取り入れ、誤差なく最適解に収束することを示した。
6. フェロモン濃度のエントロピーを観測することで、解の収束性を評価した。

今回は最短経路探索手法を主体に研究したが、今後はカーナビゲーション等への応用を検討する。

## 参考文献

- [1] M. Dorigo, V. Maniezzo and A. Coloni: The Ant System: Optimization by a Colony of Cooperating Agents, IEEE Trans. on SMC-Part B, **26**-1, 29/41 (1996)
- [2] 安藤貴彦: GA を用いた極配置による PID 制御, 高速信号処理応用技術学会誌, **10**-2, 74/80 (2007)
- [3] 石亀篤司: Particle Swarm Optimization-群れでの探索-, 計測と制御, **47**-6, 459/465 (2008)
- [4] 筒井茂義: カニグアントシステムの収束特性について, 情報処理学会研究会報告 数理モデルと問題解決 2008-MPS-70, **2008**-65, 31/34 (2008)