

Memory とフェロモンに関するルールの改良を行った MAX-MIN Ant System with Memory

磯崎敬志^{†1} 穴田一^{†2}

概要: 本研究では新たなアントコロニー最適化技法(ACO)を提案する。ACO はアリの採餌行動をモデル化したメタヒューリスティクスで、巡回セールスマン問題(TSP)などの組み合わせ最適化問題の近似解を求めることができる。ACO の一種である MAX-MIN Ant System (MMAS)は高い精度で近似解を求めることができるが、収束が遅いなどの欠点がある。そこで本研究では、MMAS に解を記憶させておくスペースである Memory を改良したものを持たせ、フェロモンの感じ方や付与ルールに関する改良を行い、新たなアルゴリズムを構築した。評価実験では従来手法と比べて解の精度と収束速度の両方が向上したことを確認した。

キーワード: 巡回セールスマン問題, アントコロニー最適化技法, MAX-MIN Ant System

MAX-MIN Ant System with Improved Memory and Pheromone Rule

TAKASHI ISOZAKI^{†1} HAJIME ANADA^{†2}

1. はじめに

本研究では、Memory とフェロモンに関するルールの改良を行った MAX-MIN Ant System with Memory という新たなアントコロニー最適化技法(Ant Colony Optimization, 以下 ACO)を提案し、それを巡回セールスマン問題(Traveling Salesman Problem, 以下 TSP)に適用した。TSP とは、複数の都市が与えられたとき、全ての都市を1度ずつ訪問し、最初の都市へ戻ってくる際の最短経路を求める組み合わせ最適化問題である。都市数が N である TSP の巡回路の総数は $(N-1)!/2$ 通りあり、都市数の増加に伴い列挙法による計算量が爆発的に増えてしまうことが知られている。このように有効時間内に計算することが困難な問題を NP 困難(NP-hard)といい、効率的に解く方法が研究されている。

ACO は、アリの採餌行動に着想を得た群知能アルゴリズムで、TSP などの組み合わせ最適化問題を解くためのメタヒューリスティクスである。アリの集団は、餌場から自分の巣へ揮発性のフェロモンを自分の通った経路に分泌しながら帰る。他のアリも、フェロモン量のより多い経路を好んで選択しながら自らもフェロモンを分泌しながら巣へ帰る。長い経路は時間当たりに行き来するアリの数が短い経路に比べて少ないため、フェロモンが短い経路より薄くなってしまふ。従って、よりフェロモン量が多く残っている経路は距離が短い経路となり、アリの集団はその経路に行列を作る。

ACO は以上のようなアリの採餌行動をモデル化し、TSP における経路選択をフェロモン量と都市間の近さによって

確率的に行い、得られた巡回路によってフェロモン情報を更新し、再び経路選択を行っていくことで近似解を生成する手法である。ACO の中には複数のベースとなるアルゴリズムが存在する。最も基本となるアルゴリズムは Ant System (AS)[1] で、1996 年に Dorigo らによって提唱された。その後、アリを解の長さによって順位付けし、順位によって各アリが付与するフェロモン量に重みを付けた Rank Based Ant System (ASrank)[2] や、フェロモンの局所更新と大局更新により解の集中化と多様化のバランスをとった Ant Colony System (ACS)[3]、フェロモン量に上限値と下限値を設定することで解の多様性を維持させた MAX-MIN Ant System (MMAS)[4] などが提唱されてきた。特に ACS や MMAS は ACO の中では性能が高く、TSP 以外の様々な問題にも適用されている。また、これらのアルゴリズムを基にパラメータを動的に変化させたり、フェロモン付与に関するルールなどを変更するなどの改良モデルも提案されている。

本研究では、MMAS をベースとした新たな ACO を提案する。MMAS はフェロモン量に上限と下限を設けているため解の多様性が維持されるが、一方で収束が遅いという欠点がある。そこで、Memory[5]という今までに見つけた最も良い解を記憶しておくスペースをそれぞれのアリに持たせ、その経路と新たに見つけた経路の良いところを組み合わせで新たな解とする手法を用いた。また、アリが経路を選択した際に行う Memory 内での都市の入れ替え方法を改良し、従来の Memory よりも収束速度を大幅に向上させ、解の精度の向上も図った。

さらにアリによる探索を行う前に Memory を Nearest Neighbor 法(以下 NN 法)で初期化することによって探索初

^{†1} 東京都市大学 大学院工学研究科
Graduate School of Engineering, Tokyo City University

^{†2} 東京都市大学 知識工学部
Faculty of Knowledge Engineering, Tokyo City University

期からある程度良い解を持った Memory の参照を可能にし、収束速度を向上させた。

また、フェロモン情報および距離情報をフェヒナーの法則に従って実際の値に対して対数比例させることで、探索によって得られる解に多様性を持たせた。

加えて、アリが経路を選択するにつれてフェロモンの付与率を下げることで、有用な経路にのみフェロモンを残すことに成功した。

評価実験では、提案モデルの有用性を調べるため、TSPLIB[6]に掲載されているベンチマーク問題を用いて従来手法との比較を行った。その結果、MMAS の欠点であった収束の遅さの改善とともに、探索によって求めた解の経路長や厳密解への到達率が大幅に向上したことを確認した。

2. 関連研究

2.1. Ant System

Ant System (AS)は、1997年に Dorigo らによって提案された TSP を解くための ACO の最初のアゴリズムである。AS は以下の 2.1.1 で初期化を行い、2.1.2~2.1.4 を 1 ステップとし、それを一定回数繰り返すことにより探索を行う。

2.1.1. 初期化

全ての経路を一律のフェロモン量で初期化する。

2.1.2. 都市の探索

m 匹のアリを N 個の都市にランダムに配置する。2 都市目以降に訪問する都市は、フェロモン情報と距離情報に基づき確率的に決定する。アリが t ステップ目に都市 i から都市 j へ移動する確率 $p_{ij}(t)$ は次式で表される。

$$p_{ij}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N'} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta} & \text{if } j \in N' \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

ここで、 $\tau_{ij}(t)$ は都市 ij 間の t ステップ目におけるフェロモン量、 η_{ij} は都市 i から都市 j への距離の逆数、 N' は未訪問都市の集合で、 α 、 β はそれぞれフェロモン情報の重みと距離情報の重みである。アリは α 、 β によってバランスがとられたフェロモン情報と距離情報を用いて未訪問都市の中から次に訪問する都市を選択する。

2.1.3. 解の評価

全てのアリが探索を終えたら解の評価を行う。そのステップで最も短い距離で探索を終えたアリの解を Iteration Best とする。また、探索開始からそのステップまでで最も短い距離で探索を終えたアリの解を Global Best とする。

2.1.4. フェロモン情報の更新

都市 ij 間の経路のフェロモン量 τ_{ij} を以下の式に従って更新する。

$$\tau_{ij}(t+1) = \rho \tau_{ij}(t) + \sum_{k=1}^m \Delta \tau_{ij}^k \quad (2)$$

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{if } (i, j) \in TOUR_k \\ 0 & \text{otherwise} \end{cases}$$

ここで、 ρ はフェロモンの残存率を表す定数で $0 < \rho < 1$ の範囲をとる。 Q は付与するフェロモン量の重みを表す定数であり、 L_k はアリ k が持っている解の長さである。全ての経路のフェロモン量は時間とともに蒸発し、アリ k が通った経路 $TOUR_k$ にのみフェロモンが付与される。

2.2. MAX-MIN Ant System

MAX-MIN Ant System (MMAS) では、AS と同様のルールで初期化と都市の探索を行うが、フェロモン付与に関するルールが異なる。AS では全てのアリがフェロモンの分泌を行うが、これによって悪い解を持ったアリがフェロモンを分泌したり、1 箇所の経路にフェロモンが集中し過ぎてしまい、局所解にトラップされてしまうことがあった。そこで、MMAS では Iteration Best の解を持つアリのみがフェロモンを分泌できるよう(2)式を以下のように改良した。

$$\tau_{ij}(t+1) = [\rho \tau_{ij}(t) + \Delta \tau^{best}(t)]_{\tau_{min}}^{\tau_{max}} \quad (3)$$

$$\Delta \tau^{best}(t) = \begin{cases} \frac{1}{L_{ib}} & \text{if } (i, j) \in IterationBest \\ 0 & \text{otherwise} \end{cases}$$

ここで、 L_{ib} はそのステップの Iteration Best の解の長さで、 τ_{max} と τ_{min} はそれぞれフェロモン量の上限値と下限値である。MMAS では各経路のフェロモン量に後述する式を用いて上限と下限を設けており、それらを超える、または下回る場合はその経路のフェロモン量を τ_{max} と τ_{min} で置き換える。これによって、フェロモン量が特定の経路に集中し過ぎることで特定の経路しか選ばれないことや、フェロモン量が 0 になってしまい選ばれない経路ができることを防いでいる。

解の評価を行った後、 τ_{max} と τ_{min} の更新を次式に従って行う。

$$\tau_{max} = \frac{1}{1-\rho} \cdot \frac{1}{L_{gb}} \quad (4)$$

$$\tau_{min} = \frac{1 - \sqrt[N]{0.05}}{(N/2 - 1)\sqrt[N]{0.05}} \tau_{max} \quad (5)$$

ここで、 L_{gb} は Global Best の解の長さである。

MMAS は、AS と比較して収束速度は遅くなるものの、精度の高い解が求まることが確認されている。また、(3)式で Iteration

Best の代わりに Global Best のアリを用いてフェロモンを分泌するよりも解の精度が良くなる事が分かっている。

2.3. Ant Colony Optimization with Memory

Ant Colony Optimization with Memory (ACO with Memory) [5]では、AS に Memory というその時点での Iteration Best を記憶するスペースを持たせることによって、AS や ACS などの ACO のベースモデルよりも収束が早く、精度の高い解を求めることに成功している。

全てのアリは都市を選択する度に Memory の経路と比較し、巡回順が異なっていれば Memory の経路をそのアリが選択した経路と同じになるよう、アリが k 番目に選択した都市と Memory 上の k 番目の都市を入れ替える。入れ替えによって巡回路長が長くなれば探索を継続し、短くなればそこで探索を打ち切り、残りの経路は Memory に記憶しているものを用いてそのアリの解とする。例えば、7都市 TSP で Memory の解が(3→4→7→5→6→2→1)で、その解の長さが 28 であったとする。アリは 1 都市目をランダムに選択するので、ここでは都市 1 が選ばれたとする。その後、(1)式に従って 2 都市目以降の都市を決定していく。2 番目に訪問する都市が都市 5 であったとすると、アリの巡回路は(1→5)であるが、Memory の巡回路は(1→3)であるので、アリが見つけた巡回路に合わせるように Memory の都市 3 と都市 5 を入れ替え、その解を(5→4→7→3→6→2→1)とする。この作業によって解の長さが増えるので再計算を行う。ここで、解の長さが 32 になったとすれば、Memory の最初の解の長さである 28 よりも長くなってしまったことになるので、探索を続行する。次に、3 番目に訪問する都市が都市 4 であったとすると、アリの巡回路は(1→5→4)となり、Memory のパスと一致している。この場合は入れ替えや解の長さの再計算などは行わない。4 番目に訪問する都市が都市 6 であったとすると、アリの巡回路は(1→5→4→6)であるが、Memory の巡回路では都市 4 の後に都市 7 を訪問しているので、Memory の都市 6 と都市 7 を入れ替え解の長さを再計算する。ここで、長さが 26 になったとすれば、Memory の最初の解の長さである 28 よりも短くなっているため、そこで探索を終了し、Memory の解をそのアリの解とする。

全てのアリが同一ステップ内でこの方法を用いて解の探索を行う。

この Memory によって、アリが部分的に良い解を発見しても他の部分で遠回りをしてしまい、結果として悪い解になってしまうようなことが減り、新たに見つけた解と今までの解の良いところ取りができる。

3. 提案手法

3.1. 改良型 Memory

本研究では、2.3.節で述べた Memory の改良を行った。従

来の Memory は、入れ替えの対象となった 2 都市のみが入れ替わるため、その都市と繋がっている経路が 2 本ずつ計 4 本繋ぎ替わることになる。この操作によって、本来の Memory の巡回経路が大きく崩れてしまうことになり、Memory による解の更新の可能性が低くなっていた。

そこで本研究では、入れ替え対象となった 2 都市だけでなく、その 2 都市に挟まれた都市全てを逆順に入れ替えることでこの問題を解決した。

例えば、2.3.節と同様に 7 都市 TSP で Memory のパスが(3→4→7→5→6→2→1)であったとする。アリは 1 都市目に都市 1 を訪問し、(1)式に従って 2 都市目以降の都市を決定していく。2 番目に訪問する都市が都市 5 であったとすると、アリのパスは(1→5)であるが、Memory のパスは(1→3)であるので、従来の Memory では Memory の都市 3 と都市 5 を入れ替えるが、提案手法ではそれだけでなく、その都市に挟まれた都市 4 と都市 7 も逆順になるように入れ替える。すると(5→7→4→3→6→2→1)となり、入れ替えを行った都市間での経路順が維持される。これによってどのような状況においても Memory 内の都市の入れ替えによって繋ぎ変わる経路の本数は 2 本のみとなり、従来よりも Memory の解を崩すことなく都市の入れ替えを行うことが可能になる。

3.2. 提案手法の流れ

提案手法は MMAS をベースとし、それに改良した Memory を導入し、フェロモンおよび距離の感じ方や、フェロモン付与に関するルールの改良を行った。提案手法は以下の 3.2.1~3.2.5 で成り立っている。3.2.1 で初期化を行い、3.2.2~3.2.5 を 1 ステップとして、それを繰り返すことで解の探索を行う。

3.2.1. Memory およびフェロモン量の初期化

従来の Memory は 1 ステップ前の最良解(Iteration Best)を記憶していた。この方法では、探索序盤には Memory に参照価値の高い解が入っていない可能性が高かった。そこで、提案手法では ACO による解の探索を行う前に、NN 法によって求めた解を Memory に記憶させる。これにより、探索序盤から参照価値の高い解が Memory に記憶されている状態になり、探索が効率的に行えることが期待できる。

NN 法は最初に訪問する都市をランダムに決定し、以降未訪問都市の中で最も距離の近い都市を順番に選択していく。この手法は常に厳密解が求まる保証はないが、非常に高速にある程度良い解が求まるため、Memory の初期解として用いることとした。NN 法 1 回の計算時間は ACO の総探索時間と比べて十分無視できるレベルである。また、全経路のフェロモン量の初期値は(4)式の L_{gb} に NN 法で求めた解の長さを代入して求めた値を用いる。

従来の MMAS も τ_{max} の初期化に NN 法で求めた解の長さを利用していただけから、Memory の初期化による計算量の増加はない。

3.2.2. 都市の探索

アリは次に訪問する都市をフェヒナーの法則によって求めたフェロモン量と距離の知覚量によって決定する。フェヒナーの法則とは人間の五感に関する法則で、知覚量が実際の刺激量に対して対数比例するといったものである。

はじめに全てのアリをランダムに都市に配置し、それ以降に訪問する都市は次式を用いて決定する。

$$p_{ij}(t) = \begin{cases} \frac{[\tau'_{ij}(t)]^\alpha [\eta'_{ij}]^\beta}{\sum_{i \in N'} [\tau'_{il}(t)]^\alpha [\eta'_{il}]^\beta} & \text{if } j \in N' \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

$$\tau'_{ij}(t) = \log \left(k \cdot \frac{\tau_{ij}(t)}{\tau_{\max}} + 1 \right)$$

$$\eta'_{ij} = \left[\log \left(k \cdot \frac{d_{ij}}{d_{\max}} + 1 \right) \right]^{-1}$$

ここで $\tau'_{ij}(t)$ および η'_{ij} は(1)式中の $\tau_{ij}(t)$ および η_{ij} をフェヒナーの法則に従って変換したものである。k は各情報の値域を制御するパラメータで、各情報をその最大値で割りスケールしたものに掛けることで、問題に依らず値域が $[0, \log(k+1)]$ となるようにしている。

また、全てのアリが探索時に 3.1 節で述べた改良型 Memory を用いて探索を行う。

3.2.3. 解の評価と Memory の更新

全てのアリが探索を終えたら解の評価を行い、Iteration Best および Global Best を記憶する。そして、Global Best の解の長さが NN 法で求めた解の長さよりも短ければ、全てのアリの Memory を Global Best で上書きする。

3.2.4. フェロモン上限値と下限値の更新

従来の MMAS と同様に、フェロモン量が特定の経路に集中し過ぎることで特定の経路しか選ばれないことや、フェロモン量が 0 になってしまい選ばれない経路ができることを防ぐため、フェロモン量の上限と下限を設定し、(4)式および(5)式で更新する。

3.2.5. フェロモン量の更新

アリが経路を選択していくにつれて、残りの選択可能な経路は減っていくため、有用な経路が選択できていない可能性がある。そこで都市を選ぶにつれて付与するフェロモン量を減らし、有用でない可能性がある経路へのフェロモン付与率を低く設定した。

都市 ij 間の経路のフェロモン量 τ_{ij} を以下の式に従って更新する。

$$\tau_{ij}(t+1) = [\rho \tau_{ij}(t) + f(s) \Delta \tau_{ij}(t)]_{\tau_{\min}}^{\tau_{\max}} \quad (7)$$

$$f(s) = -\frac{2(1-c)}{1 + e^{-a(N-s)}} + 1$$

$$\Delta \tau_{ij}(t) = \begin{cases} \frac{1}{L_{ib}} & \text{if } (i, j) \in \text{IterationBest} \\ 0 & \text{otherwise} \end{cases}$$

ここで $f(s)$ はアリが経路を選択した順番 s を引数とするシグモイド関数で、 s の値が大きくなるほど $f(s)$ の値が小さくなる。a はシグモイド関数のカーブの強さを表す定数、c は最後に選択した経路へのフェロモン付与率を表す定数で、 $0 \leq c \leq 1$ の値をとる。

4. 評価実験

提案手法の有用性を確認するため、TSPLIB [6] に掲載されている TSP のベンチマーク問題を用いて実験を行った。比較に用いた手法は従来の MMAS、MMAS に改良型 Memory を導入したもの、その Memory を NN 法で初期化したもの、さらに対数感度とフェロモン付与率の減衰を導入したものの 4 手法である。使用した問題は kroA100 (都市数 100, 厳密解の長さ 21282) である。予備実験により $\alpha=1$, $\rho=0.98$ と固定した状態で、 β と s の値をそれぞれ 1 刻みで 1~5 と動かしパラメータを決定した。さらに上記の中で最も良かったパラメータを用いて a の値を 0.2~1.0 の 0.2 刻み、c の値を 0.1~0.9 の 0.2 刻みで動かし最も良いパラメータを決定した。本稿では、MMAS と提案手法のパラメータは $\alpha=1$, $\beta=2$, $\rho=0.98$, アリの数は各手法とも都市数と同じとし、提案手法のみで用いるパラメータは $k=3$, $a=0.4$, $c=0.3$ とした。

各手法を 200 回ずつ試行した際の平均値について、横軸にステップ数、縦軸に解の長さをとったグラフを図 1 に示す。MMAS に改良型 Memory を導入することで、収束が大幅に早くなっていることが確認できる。Nearest Neighbor 法で初期化した手法については、してない手法と比べて探索開始直後から経路長が短く、序盤から効率の良い探索が行えていることが確認できた。

表 1 は各手法を 200 回ずつ計算した際の 1000 ステップ目における評価を表にしたものである。改良型 Memory を導入することで厳密解到達率や到達までの平均ステップ数、解の平均値や標準偏差が向上しており、提案手法の Memory が優れていることが確認できた。さらに、対数感度や付与ルールの変更を行うことでさらに厳密解へ到達するまでの速度が速くなり、MMAS の 1/4 以下のステップ数ですべての試行について厳密解に到達したことが確認できた。

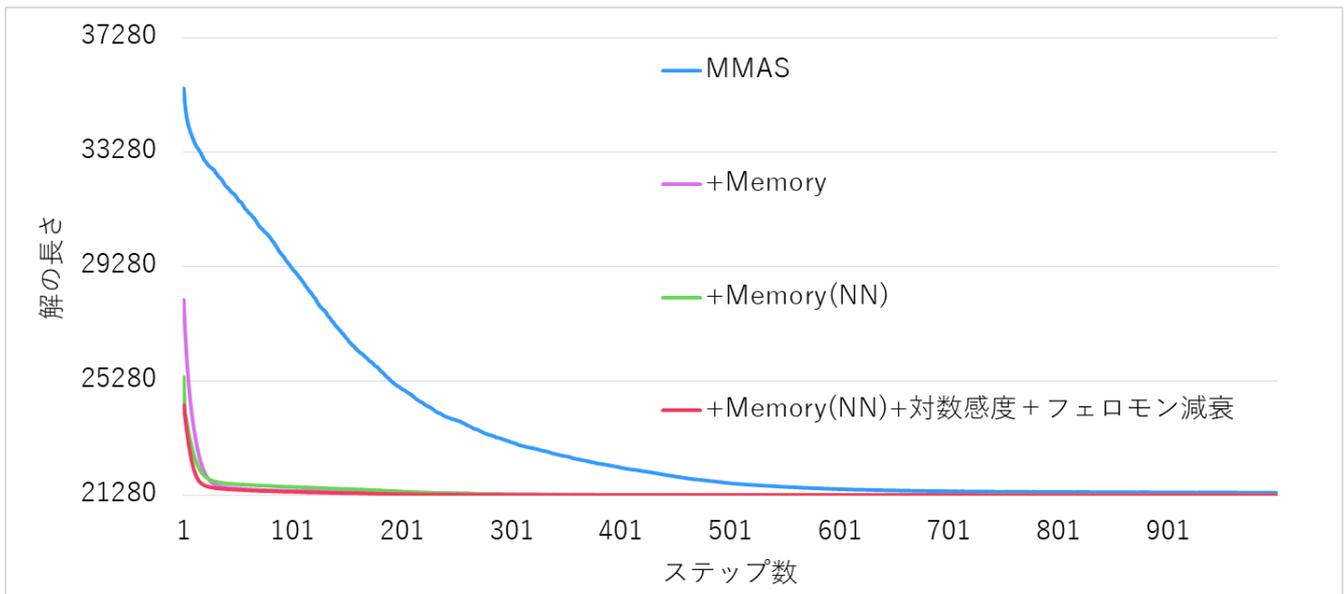


図 1. 解の精度と収束速度の関係

横軸にステップ数，縦軸に解の長さをとった際の既存手法と提案手法の性能．提案手法では収束速度と解の精度の両方が既存手法よりも優れていることが分かる．

表 1. 1000 ステップ目における各手法の性能

	MMAS	+Memory	+Memory(NN)	+Memory(NN) +対数感度 +フェロモン減衰
厳密解到達率	17.0%	81.0%	96.0%	100.0%
平均到達ステップ	832.5	202.3	291.3	195.2
解の平均値	21369.6	21287.9	21283.2	21282.0
解の標準偏差	56.3	15.6	7.7	0.0
エラー率5%到達率	100.0%	100.0%	100.0%	100.0%
エラー率5%平均ステップ	383.4	18.6	14.4	9.6

5. 考察

MMAS に Memory を導入することによって，導入しない手法よりも収束速度が上がるということが評価実験により確認できた．これは，Memory が記憶している解とアリが新たに発見した経路を上手く組み合わせることができたからである．Memory は部分的に良い解を発見したときに，残りの解を今までの Global Best で補うことから，部分的に良い解を発見したアリが Iteration Best となる可能性が上がり，そのアリが発見した経路を無駄にせずに済んだと言える．

また，Memory を NN 法で初期化することによって，厳密解到達率や標準偏差，収束速度の向上が確認できた．既存の Memory は 1 ステップ前の Iteration Best を代入していたため，1 ステップ目では Memory が利用できないという

欠点があった．さらに，2 ステップ目以降においても，探索初期においては Iteration Best が良い解であるとは言えないため，Memory の参照値が低く，Memory で残りの解を置き換えてもあまり良い解にならないことがあった．NN 法による Memory の初期化は，このような欠点をうまく補えたのではないかと考えられる．NN 法で求めた解は部分的には厳密解に含まれる経路を多く含んでいるため，最終的なアリの解も厳密解に近く安定した探索が行えていた．

さらに，Memory を改良することで厳密解到達率と収束速度の向上が確認できた．従来型の Memory では，入れ替えた 2 都市と繋がっている各 2 経路が繋ぎ替わるため最大で 4 経路が繋ぎ替わっていた．これによってアリが新たに選択した経路以外の経路が意図せず繋ぎ替わり解が悪くな

ってしまうことがあった。提案手法では都市の入れ替え方法を見直すことにより繋ぎ替わる経路の本数を最小の2本に減らし、解が悪くなりにくくなったことが向上の要因と考えられる。

フェロモン情報や距離情報を実際の値に対して対数比例させたことについては、最も値の大きい経路と最も値の小さい経路の値の比が小さくなったことで、解の多様性が生まれ、優れた探索性能を発揮できたと考えられる。

最後にシグモイド関数を用いたフェロモン付与率の減衰については、アリが経路を選択していくにつれて、残りの選択可能な経路が減っていくため、本来アリが選択しなかった経路が選択できなくなってしまっている可能性がある。そのような経路へのフェロモン付与率を減らすことで有用な経路のみにフェロモンが残り、効率的な探索が行えたものと考えられる。

6. むすび

本研究では既存の MMAS に NN 法で初期化した Memory を導入、さらにその Memory 上の都市の入れ替え方法を改良、加えてアリが経路を選択するにつれてフェロモンの付与率を下げることで、有用な経路にのみフェロモンを残すアルゴリズムを提案した。そして評価実験により従来の MMAS よりも高速で高精度な探索を行えることを確認した。

今後の課題として、パラメータの設定方法の確立が挙げられる。提案手法では設定するパラメータの数が既存のモデルより多くなったため、互いに影響が少ないと思われるパラメータを固定した状態で複数のパラメータを動かして予備実験を行った。しかし、全く他のパラメータに影響を与えないパラメータがある訳ではないので、より良いパラメータが存在する可能性がある。しかしながら、パラメータ数に比例して組み合わせの数も増えてしまう為、パラメータが互いに及ぼす影響について深く考察し、容易にパラメータを設定できるような方法を確立したいと考えている。

また、より多くの TSP のベンチマーク問題を用いて提案手法の有用性を確認することが挙げられる。都市の配置が複雑な問題ほど、既存の MMAS よりも提案手法が優れた結果を出すのではないかと期待できる。

さらに、TSP 以外の問題への適用も検討したい。現実世界に存在する問題で、TSP の制約条件とそっくりそのまま同じものはほとんどない。実際は時間による制約や1度に運べる荷物の量など、様々な制約条件が加わる。そのような現実世界の問題に即した制約条件の下で、提案手法の有用性を確認することも重要であると考えている。

参考文献

[1] Dorigo M., Maniezzo V., Coloni A., Ant system: optimization by a colony of cooperating agents, Systems,

- Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, Volume 26 Issue 1, Pages 29-41, 1996.
- [2] Bernd Bullnheimer, Richard F. Hartl, Christine Strauß, A New Rank Based Version of the Ant System - A Computational Study, Central European Journal for Operations Research and Economics, Volume 7, Pages 25-38, 1997.
- [3] Dorigo M., Gambardella L.M., Dorigo M., Gambardella L.M., Ant colony system: a cooperative learning approach to the traveling salesman problem, Evolutionary Computation, IEEE Transactions on, Volume 1 Issue 1, Pages 53-66, 1997.
- [4] Thomas Stützle, Holger H. Hoos, MAX-MIN Ant System, Future Generation Computer Systems, Volume 16 Issue 9, Pages 889-914, 2000.
- [5] Rong-Long WANG, Li-Qing ZHAO, Xiao-Fan ZHOU, Ant Colony Optimization with Memory and Its Application to Traveling Salesman Problem, IEICE TRANSC. FUNDAMENTALS, Volume E95-A No.3, Pages 639-645, 2012.
- [6] TSPLIB,
<http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/>