

改良型 Memory を用いた MAX-MIN Ant System

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。そこで提案手法では、MMAS 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、局所解からの脱出を目的とした近傍探索アリ、解の多様性の維持を目的とした複数のアリによるフェロモン更新を導入し、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評価実験で確認した。

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

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

Abstract: We construct a new ACO algorithm by the introduction of an improved memory of individual ant into the MMAS. And we confirm the great effectiveness of our algorithm by comparing with other ACO algorithms using the benchmark problems from the TSPLIB.

Keywords: Traveling Salesman Problem, Ant Colony Optimization, MAX-MIN Ant System

1. はじめに

本研究では、改良型 Memory を用いた MAX-MIN Ant System 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)に適用する。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、このような問題を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。短い経路ほど何度もアリが往復することになるので、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。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序付けし、順序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、局所更新と大局更新により解の集中化と多様性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 のの中では性能が高く、今までに 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持たせ、その経路と新たに見つけた経路の良いところをとって新たな解とする手法を用いた。また、従来の Memory の都市の入れ替え方を改良し、より収束速度と解の精度を向上させた。さらに、Memory を NN 法で初期化することで初期化を行わない手法よりも収束を早くすることを可能にした。

加えて、自然界にも存在すると言われているフェロモン

^{†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

情報に鈍感なアリを応用した近傍探索アリを導入して解の多様性を維持した。この近傍探索アリはフェロモン情報や距離情報を一切利用せず、Memory に記憶された経路の近傍解を後述するルールに従って生成するアリである。

評価実験では、提案モデルの有効性を調べるため、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、厳密解到達率も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 の最初のアルゴリズムである。AS は以下の 2.1.1~2.1.4 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。また、 m 匹のアリを N 個の都市にランダムに配置する。

2.1.2. 都市の探索

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

t ステップ目の都市 ij 間の経路のフェロモン量 $\tau_{ij}(t)$ 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、 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、アリが通った経路にのみフェロモンが付与される。

2.2. MAX-MIN Ant System (MMAS)

MMAS では、AS と同様に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 ではそのステップで最も良い成績を残したアリのみがフェロモンを分泌できるように(2)式を以下のように改良した。

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

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

ここで、 L_{ib} はそのステップの最良解である Iteration Best の解の長さである。そのステップまででの最良解である Global Best の解の長さ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ことが分かっている。また、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 τ_{max} と下限 τ_{min} を設けている。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。

2.3. Ant Colony Optimization with Memory (ACO with Memory)

ACO with Memory[5]では、AS に Memory というその時点での Global Best を記憶したスペースを持たせ、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 に記憶された都市と自身が選択した都市を比較し、異なっていれば Memory 上のこれらの 2 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続行し、短くなればそこで探索を打ち切り、Memory 上の巡回路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作業を繰り返す。

例えば、都市数 7 の TSP で Memory のパス TOURMS 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、アリのパス TOURPS は (1→5) であるが、Memory のパスは都市 1 の後に都市 3 を訪問しているので、Memory の都市 3 と都市 5 を入れ替え、アリが見つけた解に合わせ TOURMS (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、アリのパスは TOURPS (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、アリのパスは TOURPS (1→5→4→6) であるが、Memory のパスは都市 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。

この Memory によって、アリが良い経路を発見しても他の部分で遠回りをしてしまい、結果として悪い解となってしまうようなことがなくなり、新たに見つけた解と今までの解の良いとこ取りができる。

3. 提案手法

提案手法は MMAS をベースとして、それに改良した Memory とフェロモン情報および距離情報を用いない近傍探索アリを導入し、複数のアリがフェロモンを付与できるようにすることでより高精度な解が求まるようにした。提案手法は以下の 3.1~3.5 で成り立っている。3.1 で初期化を行い、3.2~3.5 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。

3.1. 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、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ことから、Memory の初期化による計算の増加はない。

3.2. 都市の探索

従来の ACO と同様に、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 (1) 式を用いて決定する。

全てのアリが探索時に Memory の解との比較を行う。従来の Memory とは異なり、対象の 2 都市のみの入れ替えを行うのではなく、対象の 2 都市間の都市の訪問順を逆転させる。これにより入れ替えを行った都市間の訪問順が維持され、より Memory の解に近い経路を得ることができる。

また、近傍探索アリを用いて Memory に記憶されている解の近傍解を生成した。近傍探索アリは、Memory 上の 2 都市間の全ての都市を入れ替えた近傍解を生成し、入れ替えによって巡回路長が短くなれば解として残し、そうでなければ破棄する。都市数が N のとき、近傍解の個数は ${}_NC_2$ 通りとなるので、 ${}_NC_2$ 匹の近傍探索アリが生まれる。

3.3. 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う。Iteration Best の解の長さが今までの Global Best の解の長さよりも短ければ Global Best を更新し、全てのアリの Memory を Global Best で上書きする。

3.4. 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。

3.5. フェロモン量の更新

t ステップ目の都市 ij 間の経路のフェロモン量 $\tau_{ij}(t)$ を以下の式に従って更新する。

$$\tau_{ij}(t+1) = \left[\rho \tau_{ij}(t) + \sum_{k=1}^{\sigma} \Delta \tau_{ij}^k(t) \right]_{\tau_{\min}}^{\tau_{\max}} \quad (6)$$

$$\Delta \tau_{ij}^k(t) = \begin{cases} \frac{1}{\sigma \cdot L_k} & \text{if } (i, j) \in \text{IterationBest} \\ 0 & \text{otherwise} \end{cases}$$

全ての経路のフェロモンは時間とともに蒸発していき、巡回路長が短いアリから順に σ 番目までが通った経路にフェロモンを分泌する。分泌量はアリの巡回路長 L_k にフェロモンを分泌するアリの数 σ をかけたものの逆数である。ただし、各経路に置かれるフェロモン量の上下限は(4)式および(5)式で決定された値となる。

4. 評価実験

提案手法の有効性を確認するため、TSPLIB 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。使用した問題は kroA100(都市数 $N=100$)で、予備実験により α , β の値をそれぞれ 1 刻みで 1~5, ρ の値を 0.01 刻みで 0.80~0.99, σ の値を 1 刻みで 1~10 まで変化させ、最適なパラメータとして $\alpha=1$, $\beta=2$, $\rho=0.98$, $\sigma=2$ とした。MMAS では最適な値とされている $\alpha=1$, $\beta=2$, $\rho=0.98$, AS + Memory も同様に最適な値とされている $\alpha=1$, $\beta=5$, $\rho=0.50$, アリの数は各手法とも都市数と同じ 100 匹とした。

既存の手法と提案手法を各 200 回ずつ計算し、それぞれの平均の解の長さを縦軸、ステップ数を横軸にとったものを図 1 に示す。従来の MMAS は収束が遅いことが欠点であったが、提案手法では NN 法で初期化した Memory を導入することによって AS + Memory に劣らない収束の早さを確認することができた。また、1000 ステップ目における探索性能の違いを表 1 に示す。提案手法は NN 法で初期化した Memory を MMAS に適用することで厳密解到達率や、それにかかるステップ数が既存の手法より著しく向上していることがわかる。さらに、改良型 Memory を用いることで従来の Memory 以上に探索性能が向上していることが分かる。近傍探索アリも、わずかではあるが厳密解到達率や収束速度を向上させている。加えて、成績の良い複数のアリにフェロモンを分泌させることで、より探索性能が向上したことが分かる。

5. おわりに

本研究では既存の MMAS に NN 法で初期化した改良型 Memory と近傍探索アリを導入し、成績の良い複数のアリにフェロモンを分泌させる手法を提案した。そして評価実験により高速でより高精度な探索を行えることを確認した。

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のが評価実験により確認できた。こ

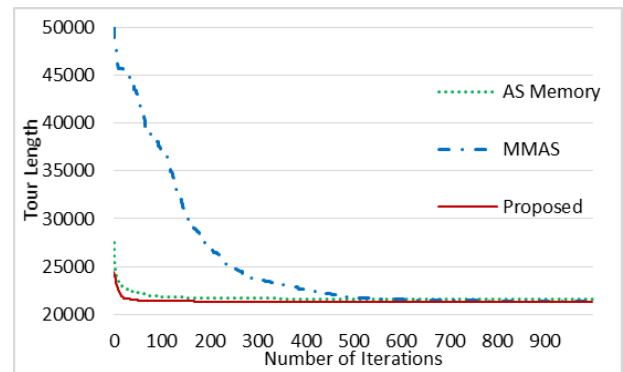


図 1. 解の精度と収束速度の関係
横軸にステップ数、縦軸に解の長さをとった際の既存手法と提案手法の性能。提案手法では収束速度と解の精度の両方が既存手法よりも優れていることが分かる。

れは、Memory が記憶している解とアリが新たに発見した解の良いとこ取りが可能となり無駄になる経路がなくなったからではないかと考えられる。Memory は良い経路を発見したときに、残りの解を今までの Global Best で補うことから、序盤に良い経路を発見したアリの解が Iteration Best となる可能性が上がり、アリが発見した経路を無駄にせず済んだと言える。

また、Memory の改良により従来のものよりも解の精度と収束速度の両方が向上した。従来の Memory では対象の 2 都市のみを入れ替えていたため、1 回入れ替えを行うごとに最大 4 本の経路が繋ぎ替わっていた。そのうち 1 本はアリの経路に合わせるように入れ替わるが、残りの経路はそうでないため、Memory の解が良くなることも多くあった。提案手法では対象の 2 都市間の都市を全て入れ替えるため、繋ぎ替わる経路は 2 本のみである。これによって Memory に記憶された良い経路を維持しやすくなり、従来手法以上 Global Best の解を有効利用して新たに良い解を発見できたのではないかと考えられる。

NN 法による初期化に関しては、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 Iteration Best は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 法で求めた解は厳密解に含まれる経路を多く含んでいるため、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。

近傍探索アリの導入については、Memory と組み合わせることによって、複数の近傍解の中から良いものを探索に活かすことができ、Memory と近傍探索アリを組み合わせた手法は Memory のみの手法よりも解の精度が向上したの

表 1. 1000 ステップ目における各手法の性能

	MMAS	MMAS +Memory	MMAS +新Memory	MMAS +新Memory +近傍アリ	MMAS +新Memory +近傍アリ +複数アリ
厳密解到達率	7.5%	56.5%	87.0%	89.0%	91.0%
平均到達ステップ	823.73	541.67	211.45	194.89	196.89
解の平均値	21390.54	21293.23	21286.47	21287.08	21284.44
解の標準偏差	72.78	19.09	15.05	18.48	9.28
エラー率	0.510%	0.053%	0.021%	0.024%	0.011%
エラー率1%到達率	99.0%	100.0%	100.0%	100.0%	100.0%
エラー率1%平均ステップ	659.37	279.78	76.17	73.38	73.85

ではないだろうか。

最後に、成績の良い複数のアリがフェロモン分泌を行うことで、解の精度の向上が確認できた。1位のアリのみがフェロモンを付与した場合、その解の周辺が探索されやすくなる。このとき、1位のアリが局所解にトラップされてしまうと、最終的な解も局所解にトラップされてしまう。複数のアリがフェロモンを残すことで、1つの解の周りにアリが集中し過ぎず解の多様性を維持できているのではないかと考えられる。

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、互いに影響を及ぼし合わ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の組み合わせが見つかる可能性がある。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効性を確認することが挙げられる。これまで既存手法では良い結果が得られなかった複雑な問題でも、提案手法では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。現実にかかる問題で、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コストの変化など、様々な制約条件が動的に変化する。そのような現実世界の問題に即した条件の下で、提案手法の有効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.ifi.uni-heidelberg.de/software/TSPLIB95/>