蟻コロニー最適化と粒子群最適化の融合による巡回セールスマン問題解法の効率化

2006MI183 多和田　泰子
指導教員：高見　勲

1　はじめに

本研究は，Ant Colony Optimization(蟻コロニー最適化手法，以下 ACO) に Particle Swarm Optimization(粒子群最適化手法，以下 PSO) を融合し，効率的な Traveling Salesman Problem(巡回セールスマン問題，以下 TSP) 解法アルゴリズムの作成を目的としている．

厳密な最適解を求める必要性が少ない TSP において，大域的な情報を保持でき，かつ動的な変化に対応できる ACO は，優れた近似解解法の 1 つとして挙げられる [1]．しかし，ACO は計算時間が長く局所解の誤差が大きい．そのため「今までの最良解を考慮して行動を決定する」という考えを持つ PSO を融合し，より短い経路を選択する確率を増やし，多様性を損なわず収束性を向上させ，TSP 解法の効率化を目指す．

本研究では提案した PSO 融合アルゴリズムの検証を行い，元のの ACO アルゴリズムや，ACO の代表的な TSP 解法アルゴリズムである $\mathcal{MAX}-\mathcal{MIN}$ Ant System(以下 $\mathcal{MM}$AS) と比較をする．

2　Ant Colony Optimization

ACO とは，蟻がコロニー (群れ) から餌までの経路を見つける際の行動方法を元に作られた手法である．図 1 に TSP 解法の ACO アルゴリズムを示す．

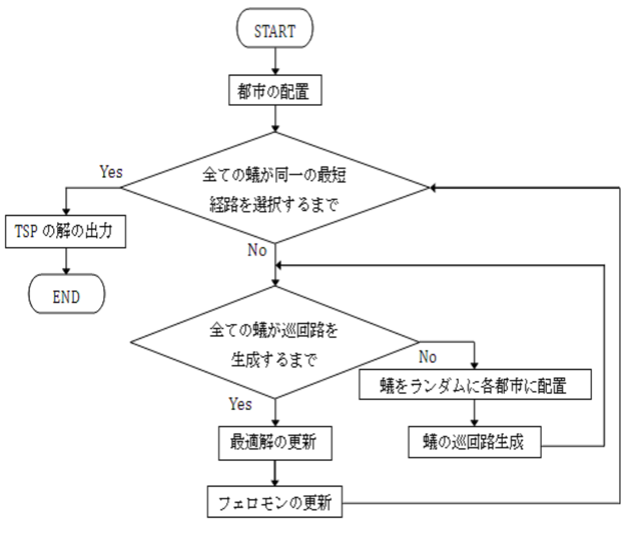

図 1　ACO フローチャート

3　Particle Swarm Optimization

PSO とは，鳥や魚，昆虫の群れといった集団での探索行動に基づいた多点型最適化手法である．今回は，PSO の「自己最良解と全体の最良解の情報を利用する」といった考えを用いる．図 2 に使用する PSO アルゴリズムを示す．

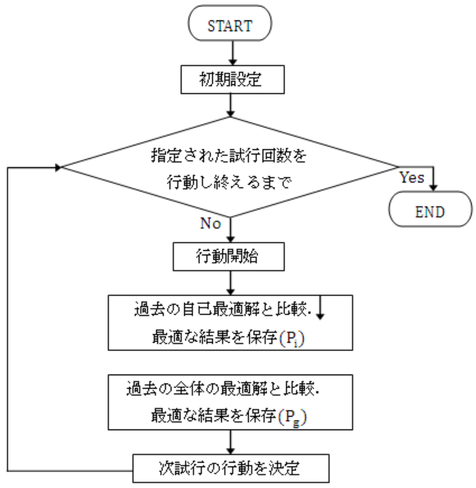

図 2　PSO フローチャート

4　ACO・PSO の融合方法の提案

4.1　最良解のフェロモンに重みを与える更新法

この融合方法は，最良解の経路のフェロモンに重みを与える計算法である．この時，自己最良解は常に更新し，次の試行のみ有効とする．図 3 にフローチャートを示す．

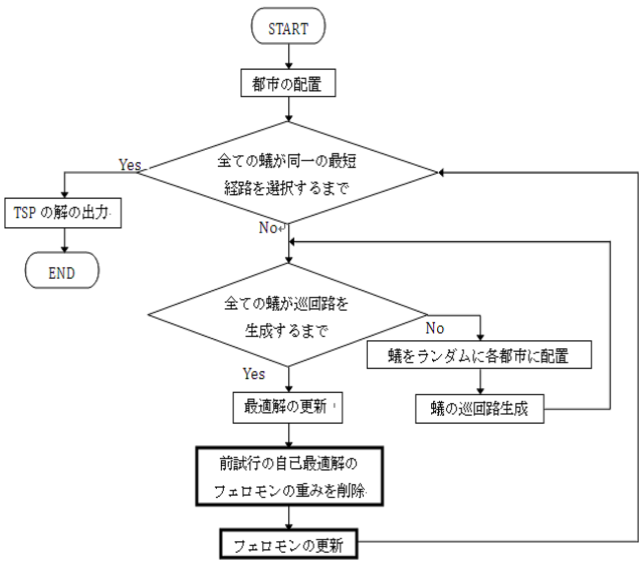

図 3　フェロモンに重みを与える更新法フローチャート

フェロモン濃度更新の計算式は以下である．

$t=1$ の時

$$\tau_{ij}(t+1)=\rho\cdot\tau_{ij}(t)+\Delta\tau_{nij} \tag{1}$$

$t>1$ の時

$$\tau_{ij}(t+1)=\rho\cdot[\tau_{ij}(t)-\sum_{k=1}^{K}P^{k}(t-1)]+\Delta T(t) \tag{2}$$

$$\Delta T(t) = \begin{cases} \Delta\tau_{gij}(t) = (\frac{1}{d_g})^{\gamma} & if(D_{ij} \in G_{best}) \\ \Delta\tau^k_{pij}(t) = (\frac{1}{d^k_p})^{\beta} & elseif(D_{ij} \in P^k_{best}) \\ \Delta\tau_{nij}(t) = \sum_{k \in A_{ij}} (\frac{1}{d_n}) & else \end{cases} \tag{3}$$

τ_{ij}:都市 i から j までの経路のフェロモン量　t:試行回数
k:蟻の番号 (1,2,$\cdots$, K)　d:経路の距離　ρ:フェロモン蒸発係数
g:全体の最良解　p:自己最良解　D_{ij}:都市 i から j までの経路
G_{best}:全体の最良解の経路　P^k_{best}:蟻 k の自己最良解の経路
A_{ij}：都市 i から j への経路を通った全ての蟻の集合
$P^k(t)$:蟻 k の t 回目の自己最良解フェロモンの増加量
β, γ：フェロモン増加量の指数パラメータ　ただし $\beta \geq \gamma$

$D_{ij} \in G_{best}$ とは経路 d_{ij} が自己最良解の経路であるということである．$\beta \geq \gamma$ とし，自己最良解より全体の最良解により多くのフェロモンが分泌されるようにした．

4.2　経路を選択する確率を優遇させる計算法

この融合方法は，探索時に自己最良解と全体の最良解の経路を選択する確率を増加させる探索法である．自己最良解の情報は自分のもののみ利用するため，蟻はそれぞれ独立した探索を行う．

確率計算時の式を式 (4)(5) に変更する．

$$p^k_{ij}(t) = \frac{\tau_{ij}(t)\eta^{\alpha}_{ij} \times \omega}{\sum_{h \in J^k_i} \tau_{ij}(t)\eta^{\alpha}_{ij} \times \omega} \tag{4}$$

$$\omega = \begin{cases} \delta & if(D_{ij} \in G_{best}) \\ \epsilon & elseif(D_{ij} \in P^k_{best}) \\ 1 & else \end{cases} \tag{5}$$

p^k_{ij}：蟻 k が都市 i から都市 j に進む確率
η_{ij}：ヒューリスティック値　距離の逆数 $\frac{1}{d_{ij}}$
J^k_i：蟻 k が都市 i から進めるすべての都市の集合
α：ヒューリスティック値の指数パラメータ
δ，ϵ：確率増加量の指数パラメータ ただし $\delta \geq \epsilon$

$\delta \geq \epsilon$ とし，自己最良解よりも全体の最良解の経路に行く確率を増加させた．

5　検証

元の ACO アルゴリズムと融合後のアルゴリズムで TSP を解き，結果を比較する．対象問題は TSPLIB[1] の dantzig42.tsp(都市数 42)，スタート座標を (170, 85)(1 つ目の要素) に設定する．　解の精度がよい $\mathcal{MMAS}$ でステップ数 120000 の検証により求めた最良解の距離は 678.783678 である．

今回はステップ数 200 回，蟻の数は都市数と同様に 42 匹で 10 回検証を行う．各パラメータは $\alpha = 1$，$\beta = 0.75$，$\gamma = 0.5$，$\delta = 2$，$\epsilon = 1.5$，$\rho = 0.8$ とする．経路のランダム選択を行うための乱数の種 (seed) を 0 から 9 に変化させた結果を示す．左側がグラフの全体図，右側が左のグラフの縦軸の範囲を 650 から 900 に拡大した図である．

図 4，5，6 より融合後のアルゴリズムは ACO より収束時間が早く，誤差も少ないことが分かる．しかし，図 6 左側より，第 4.2 節の確率計算時の融合アルゴリズムでは収束が早いが誤差も大きい．以上より，第 4.1 節「最良解のフェロモンに重みを与える更新法」が最も誤差が少なく，かつ収束も早いといえる．なお，平均計算時間は 3 つとも約 1.33 秒であった．

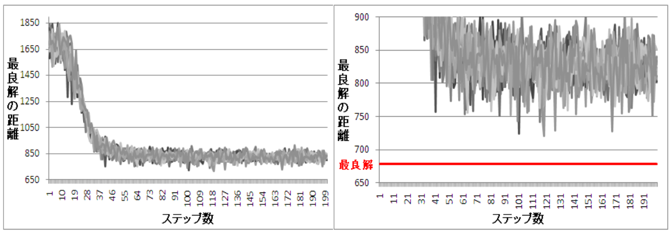

図 4　ACO　検証 10 回分のグラフと拡大図

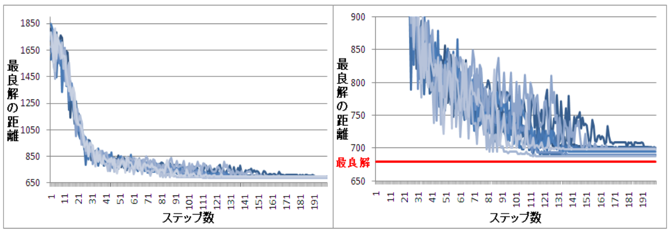

図 5　ACOPSO4.1　検証 10 回分のグラフと拡大図

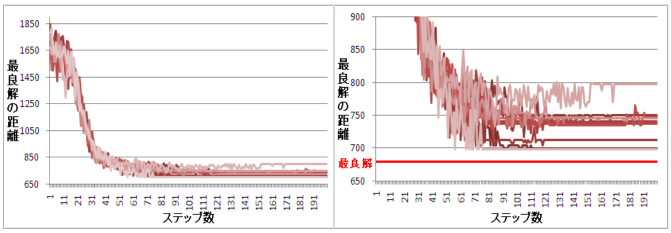

図 6　ACOPSO4.2　検証 10 回分のグラフと拡大図

次に $\mathcal{MMAS}$ との比較を行う．$\mathcal{MMAS}$ で同じ条件で dantzig42.tsp を解いた結果を図 7 左に，最良解の距離を 670 から 750 に拡大した図を右に示す．ただし，最良解に辿り着くよう 120000 回の試行で検証を行った．

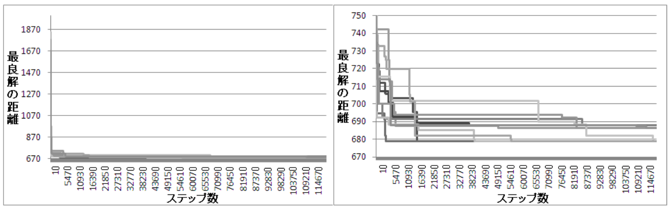

図 7　MMAS　検証 10 回分のグラフと拡大図

図 7 より収束した解の精度は良いが，完全に収束するまでのステップ数が 100000 回前後であることが分かる．また，検証 10 回分の平均計算時間は 408 秒であった．これより $\mathcal{MMAS}$ は最適解を求める上で性能は優秀であるが，計算時間が大きい事が分かる．したがって厳密な最適解を求める必要がない TSP の場合は，本研究の融合アルゴリズム ACOPSO4.1「最適解のフェロモンに重みを与える更新法」の方が計算時間が短く，かつ誤差も少ないため有効であると言える．

6　おわりに

本研究の成果は以下である．

- 融合アルゴリズムを提案・検証できた
- 融合アルゴリズムにより，TSP の解の収束性を高め，収束時間と誤差を減らす事が出来た．

参考文献

[1] 伊庭斉志：『知の化学　進化論的計算手法』．オーム社，東京，2003.

[1] http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/