

蜜蜂の採餌行動を模した解の選別基準を 自律的に調整する BCO

古川まき † 鈴木泰博 †

Bee Colony Optimization(BCO)は蜜蜂の採餌行動を模した最適化手法であり、巡回セールスマン問題(Travelling Salesman Problem TSP)に適用されている。蜜蜂を模したエージェントが各1個の解を持ち、巣内での8の字ダンスによる仲間への解情報の伝搬と、巣外の採餌による解の局所的最適化によって最適解を探索する。このBCOに対し、本研究では解情報の伝搬を制御するための閾値を導入した。各エージェントに対し閾値によって解の選別を行い、より優良な解を持つエージェントのみ情報伝搬を行うよう改良することによって、解の探索精度の向上を図った。この閾値は蜜蜂の採餌基準の調整を模したものであり、状況に応じて自動調整される。これをTSPのいくつかの問題に適用し、結果を示す。

BCO with Autonomous Threshold Regulation to Select Solutions Mimicking Honey Bee Foraging

Maki Furukawa † and Yasuhiro Suzuki †

Bee Colony Optimization (BCO) is a meta-heuristics for Travelling Salesman Problem (TSP). In BCO, agents mimicking the honeybee search shorter tour by waggle dance and foraging. Waggle dance means advertising and referring each other's tour and foraging means the local optimization of their own tour. We introduced the threshold which controls the number of advertising agents to improve the BCO. Only agents who have shorter path-length than the threshold can advertise their tour to others. The threshold is adjusted in the way of honeybees' adjusting their foraging threshold according to the situations. Experimental results comparing the simple BCO model with the adding threshold model on a set of benchmarks problems are presented.

1. はじめに

社会性昆虫の一種である蜜蜂は高度に組織化された社会構造を持っている。その生態についての基礎研究は Seeley が行っており「ミツバチの知恵」1)に詳しくまとめられている。それらの行動を倣って、たとえば8の字ダンスなどの独特の採蜜行動の特徴を模したマルチエージェントシステムによる最適化手法などが各種提案されている。

代表的なものに Bee Colony Optimization (BCO)2)3)4) がある。BCO は巡回セールスマン問題(Traveling Salesperson Problem, TSP)を解くため Wong らによって提案された最適化アルゴリズムである。蜜蜂を模したエージェントを使用し、巣内の8の字ダンスと巣外の採餌によって解探索することになり、いくつかの問題に適用されている。5)6)。

しかし、8の字ダンスを行う基準が実際の蜜蜂の行動に即しておらず、本来自律的に行われる採蜜の基準値(閾値)の調整を取り入れる部分に対し、BCOの使用者が適当な値を設定するという方法で調整しているため、蜜蜂の群知能の最も重要な部分が十分に利用されていない。

そこで本研究では、蜜蜂の閾値自動調整機能を用いた改良を提案する。各エージェントの持つ解情報が他のエージェントへ伝える価値があるものであるかどうかの選別に使用する閾値を自律調整する機能を付与したBCOについて比較を行う。

2. Bee Colony Optimization

巡回セールスマン問題(TSP)とは何ヶ所かの都市とその都市間距離が与えられたとき、すべての都市を一巡する最短の巡回経路を求める問題である。

解は全ての都市を一回ずつ含んだ都市の巡回リストであり、BCOでは蜜蜂を模したエージェントが各自1個の解を持ち、エージェントはこのリストに従って各都市を巡回する。

BCOの基本的なアルゴリズムを図1に示す。

エージェントによる最適解の探索は巣内での8の字ダンスの観察と、巣外での採餌によって行われる。ダンスの観察で、他のエージェントの優れた解を参照し、巣外の採餌で都市リストの局所的最適化を行う。

都市リストの更新が終わるとエージェントは巣にもどり、経路長の記憶を更新する。ダンスの観察前と比べ経路長が短くなっていたエージェントは次回の更新にてダンスを踊る。

この一連の操作を1サイクルとし、規定回繰り返すことによって最適解を探索する。

† 名古屋大学 情報科学研究科
Science Graduate School of Information Science, Nagoya University,

1. 初期化
 - 解とエージェントの生成
 2. 終了条件を満たすまでループ
 - 2.1 全てのエージェントに対して (サイクル)
 - 2.1.1 巣内で8の字ダンスの観察
 - 2.1.2 巣外で採餌
 - 2.1.3 帰巣し8の字ダンス
 - 2.2 解の評価
 3. 最良解を出力し終了

図 1 BCO の基本的なアルゴリズム

2.1 初期化

TSP における BCO の主な構成要素は以下の通りである

- エージェント
- 巣
- 都市

都市情報には TSPLIB^{a)}のデータを用いる。エージェントの数は実験に使用する都市の数と等しい。エージェントの初期位置は巣であり、巣から都市への移動は考慮しないものとする。巣には8の字ダンスを行うためのダンスフロアがある。

2.2 8のダンスによる解の伝搬

蜜蜂は花などから蜜を採取して巣に貯蔵する。効率よく採蜜するため、価値の高い蜜源に多数の蜂を集めるため8の字ダンスによって他の蜂を誘導する。

採蜜において、蜂は蜜の糖度や位置などから採蜜コストを算出し、巣の貯蜜状態や必要量を考慮し蜜源の重要度を判断しており、この判別基準を閾値と呼ぶ。各採蜜蜂は一つの蜜源を選んで採蜜を行い、その蜜源の評価を行う。閾値を超える場合は帰巣後に他の蜜蜂を蜜源に誘導するため8の字ダンスを行い、蜜源の評価が高いほど採蜜蜂は長くダンスを続ける。蜜源の評価が低ければ放棄し、有用な蜜源を知る他の蜂を待機する。十分多くの採蜜蜂が存在するとき、各蜜源を宣伝する蜂の数は、全ての蜜源を比較評価したときの各蜜源の評価値に比例する。

巣で待機する他の蜂はダンスする採蜜蜂のうちから一匹を無作為に選び情報を得る。新しく蜜源に誘導される蜂の数は各蜜源についてダンスする蜂の数に比例する。これはルーレット選択であり、有用な蜜源にはより多くの採蜜蜂が集まる。

これと同じように、BCO では優良な解を持つエージェントは他のエージェントに対し8の字ダンスを踊り、解を宣伝する。宣伝時間の長さは式(1)(2)(3)で与えられる。

$$D_i = K \frac{Pf_i}{Pf_{colony}} \quad (1)$$

$$Pf_i = \frac{1}{L_i} \quad (2)$$

$$Pf_{colony} = \frac{1}{N_{bee}} \sum_{i=1}^{N_{bee}} Pf_i \quad (3)$$

D : ダンス時間 (サイクル数)

K : 比例定数

Pf_i : 経路長の逆数 (収益率)

Pf_{colony} : Pf_i の平均値

L_i : 各エージェントの経路長

N_{bee} : エージェント数

前サイクルの探索で経路長の自己最短記録を更新したエージェントは、 D の期間だけがダンスを踊ることができる。このエージェントをダンサーと呼ぶ。 D はダンスを続けるサイクルの数であり、経路長に反比例する。ダンスの期間中もエージェントは採餌行動を行い、期間中に自己最短記録が更新された場合は D の値を更新する。

巣から採餌に向かうエージェントはまずこのダンス群を観察し、優良な解を参照する。経路長が長いほど他の優良な解を参照しなければならないため、ダンスを観察する確率は3)より、Table1 のように与える。ダンスを観察するエージェントはダンサーのうち無作為に選択した1個体の都市リストを丸ごと写し取る。

探索が進み解の更新が少なくなるとダンサーがいなくなる場合がある。この時は適当な期間を設け、それを過ぎてもダンサーが現れない場合には各エージェントの経路長の自己最短記録をある程度長くすることで再びダンスを行うようにする。たとえば、2)では10サイクル以上の期間ダンサーがいなるときは全エージェントの経路長の自己最短記録を10%長い値と置き換えている。

このようなエージェントによるダンスと観察は、局所的な最適化を避け、全体として最適解を探索するためのものである。

初期化直後はダンスをするエージェントがいないため、直接採餌を行う。

a) TSPLIB : <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>

表 1 P_{follow} を調整するためのルックアップテーブル
 P_{follow} : 他のエージェントのダンスを観察する確率

Profitability Scores	P_{follow}
$Pf_i < 0.5Pf_{colony}$	0.80
$0.5Pf_{colony} < Pf_i < 0.725Pf_{colony}$	0.20
$0.725Pf_{colony} < Pf_i < 0.99Pf_{colony}$	0.02
$0.99Pf_{colony} < Pf_i$	0.0

2.3 採餌

巢内でダンスを観察した後、巢外での採餌において各エージェントは TSP に適する方法で解を修正する。

巣を出たエージェントは都市リストの最初の都市に移動する。次に行く都市からはリストに従わず、未訪問の都市全てに対し式(4)(5)に従って重みづけしたルーレット選択によって経路を再決定する。各都市への移動確率は $P_{ij,n}$ で与えられる。

$$P_{ij,n} = \frac{[\rho_{ij,n}]^\alpha \left[\frac{1}{d_{ij}}\right]^\beta}{\sum_{j \in A_{i,n}} [\rho_{ij,n}]^\alpha \left[\frac{1}{d_{ij}}\right]^\beta} \quad (4)$$

i, j : 都市

$P_{ij,n}$: 都市 i から j への移動確率

d_{ij} : ij 間距離

$\rho_{ij,n}$: ij 間の部分経路(弧)の適合度

α, β : 定数

$A_{i,n}$: 都市 i からの未訪問都市群

$$\rho_{ij,n} = \begin{cases} \lambda & , j \in F_{i,n}, |A_{i,n}| > 1 \\ \frac{1 - \lambda |A_{i,n} \cap F_{i,n}|}{|A_{i,n} - F_{i,n}|} & , j \notin F_{i,n}, |A_{i,n}| > 1 \\ 1 & , |A_{i,n}| = 1 \end{cases} \quad (5)$$

λ : 定数, $0 < \lambda < 1$

θ : 変更前の都市リスト

θ_i : θ の i 番目の都市

$F_{i,n}$: n サイクル目の θ_{i+1}

式(4)のように、都市間の距離 d_{ij} と式(5)で得られる適合度 $\rho_{ij,n}$ を用いて、都市 i から都市 j への移動確率を定める。都市間距離が近く、適合度が高い都市ほど移動確率は高くなる。この確率に従い、エージェントは全ての都市を一巡する経路を新しく得るまで各都市を移動する。

適合度は採餌前の都市リスト θ によって定める。 θ の i 番目の都市を θ_i とする。エージェントが都市 θ_i に到達し、未訪問の都市が ℓ 個あるとき、 $\rho_{ij,n}$ は θ_i から θ_{i+1} ($\theta_{i+1} = \theta_j$) への移動において λ とし、それ以外の移動では $1 - \lambda / \ell - 1$ とする。ただし、すでに θ_{i+1} を訪れており、未訪問都市の中に θ_{i+1} が含まれていない場合は全ての移動に対し $1 - \lambda / \ell$ とする。

ここで適合度とは、他のエージェントのダンスより参照した経路が保存されやすくなるための重みづけである。採餌の直前にエージェントは巢内で他のエージェントのダンスを参照し、 θ はダンスによって推奨された経路であることが多い。自然界の蜜蜂と同じように、エージェントは仲間のダンスから得られた経路を基にして部分修正を行う。

新しく全ての都市を一周する経路で θ を更新した後、エージェントは巣に戻る。巣に戻ったエージェントは解の評価を行う。

初期化にて解を生成するときには、最初の都市のみランダムに選択し、変更前の都市リスト θ は空集合として扱う。

2.4 BCO の問題点

BCO において、優良な解を見つけたエージェントはダンスによって他のエージェントに伝え、全体としての解の改善を速めている。ただし、解の収束が進んで解の更新が少なくなるとダンスを踊るエージェントは減少するため、もし適当な期間を過ぎてもダンスが見られない場合には各エージェントの経路長の自己最短記録をある程度改悪するとしている。この設定について2)では、ダンサーの数を増やすためにエージェントの持つ経路長の自己最短記録を長くすることは、人間の体のホルモン調整のような恒常性機能におけるネガティブフィードバックと同じようなメカニズムと考えられると述べており、実際の蜜蜂でも同様に状況の変化とともに採餌の基準が調整されることはよく知られているため、BCO に何らかの基準値の調整機能を導入することは必要な操作であると考えられる。しかし、ダンスを踊るエージェントの数は図2のように遷移する。

探索初期では解の更新が続くため、順位の低いエージェントも全てダンスを行っている。また、終盤の解の収束が進みほとんど改善が見られなくなるとダンスを行うエージェントは激減する。

しかし、初期は解の絞り込みが必要な時期であり、全てのエージェントの解を優良として参照するのは不適切である。逆に、終盤では解が収束しやすいため微小な差異

を持つ多数の解を参照するのが望ましい。特にエージェントは採餌によって個別に新しい経路を獲得するため、あるエージェントが単独で異なる解を見つけても、それがダンスによって他のエージェントに伝わらなければ全体としての解の改善は行われな
 い。経路が収束に向かう終盤では局所解に陥ることを避けるためにもダンスによる新しい解の伝搬が重要となるはずであり、現在の BCO の設定である「解が改良されたエージェントのみ宣伝を行う」という設定では困難である。

この問題は、TSP に特化した基準の調整を行っていることに原因があると思われる。そこで、実際の蜜蜂の行動に即したダンサーになる基準を調整するための手法を以下のように提案する。

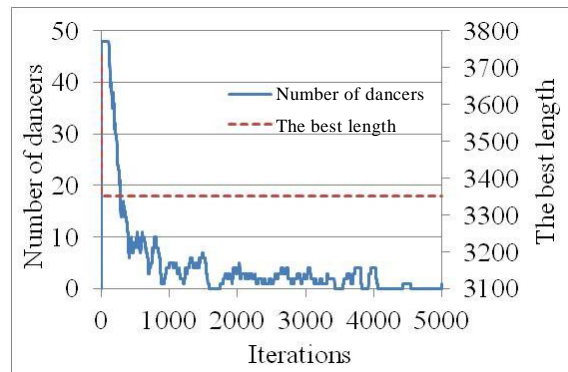


図 2 ダンサー数の遷移

3. 閾値の自動調整

3.1 採蜜蜂の閾値

実際の蜜蜂は採蜜する蜂と巣に蜜を貯蔵する蜂とに分かれ、その連携によって調整を行っていることが明らかになっている。蜜蜂にはダンスを行うかどうかの判断基準として個々に閾値があり、蜜源の価値が閾値を上回るとダンスを踊る。この閾値は、採蜜蜂が巣に帰った直後に貯蔵蜂を探し、見つかるまでの時間によって変動する。貯蔵蜂がすぐ見つかった場合は巣内の蜜の需要が高く、貯蔵蜂の数に対し採蜜蜂の数が少ないということであり、閾値を下げてダンスを行いやすくし、採蜜箇所を増やす必要がある。逆のときは閾値を上げる。このようにして、貯蔵蜂と採蜜蜂の数の比が適切になるようダンスを行う閾値に対しフィードバックをかける。

このように、ダンサーの数の推移に対し適切なフィードバックをかけることを貯蔵蜂のすべき役割とみなし、それに対応する指標と閾値を設定する。

3.2 閾値の定義

ダンスを行うエージェントは経路長の自己最短記録ではなく閾値 T と現在の経路長を比較し、短い経路長を持つエージェントとする。閾値は、図 3 に示すように、全エージェントの経路長の最小値 L_{min} (最適解の経路長とは限らない) に、平均値と最小値の差の r 倍を加えたものとする。 r は閾値指標と呼ぶ。以下、図 3 と式(6)に詳細を示す。

$$T = L_{min} + (L_{ave} - L_{min}) * r \quad (6)$$

- T : ダンサー選別のための閾値
- L_{min} : 全エージェントの最短経路長
- L_{ave} : 全エージェントの平均経路長
- r : 閾値指標

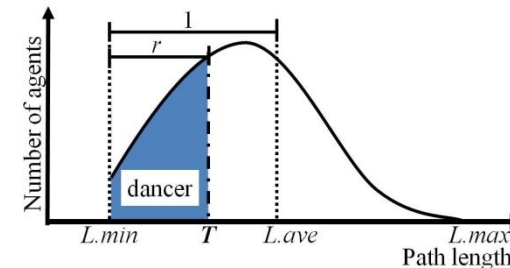


図 3 閾値の定義

次に、ダンサーが適正な数に保たれるよう、数の推移を推定する機能を付与する。各サイクルにおいて、ダンサー数の多寡を判断するのは、「今までのダンサー数」と「今のダンサー数」の比較を行ったときである。ダンサー数の履歴を考慮し、ダンサー数の平均を式(7)のように設定した。

$$E_n = \frac{E_{n-1} \times (n-1) + e_n}{n}, E_0 = 0 \quad (7)$$

E_n : n サイクル目までのダンサー数の平均
 n : サイクル数($n > 2$)
 e_n : n サイクル目のダンサー数

ただし、この平均値 E_n はサイクル数が増えると e_n の項の影響が無視される。そこで、都市数に対して式(8)のように平均を取るように改めた。

$$E_n = \frac{E_{n-1} \times (c-1) + e_n}{c} \quad (8)$$

c : 都市数

この平均の値を用いて閾値指標 r_n の値を制御する。 r_n は、式(9)のようにその増減が E_n の増減によって抑制されるように調整する。従って、閾値はサイクルごとに式(10)で決め直される。

$$\Delta r_n = - \frac{E_n - E_{n-1}}{E_n} \quad (0.0 \leq r_{n-1} \leq 1.0) \quad (9)$$

$$r_0 = 1.0$$

$$T_n = L_{min} + (L_{ave} - L_{min}) * r_n \quad (10)$$

r_n : 閾値指標
 T_n : ダンサー選別のための閾値

この指標を用いて実験を行う。

4. 実験

閾値の自動調整による効果を数値実験によって明らかにする。

実験には、2-opt を用いた BCO 3) と、それに閾値の自動調整機能を加えたもの(+T) を比較に用いる。初期化にて都市と同じ数のエージェントを生成する。都市のデータは TSPLIB より att48, eil76, lin105 を使用する。変数の値は、 $K=100$, $\alpha=1$, $\beta=10$, $\lambda=$

0.95 とする。エージェントはまず初期生成時のリストに従って採餌を行い、これを 1 サイクル目とする。2 サイクル目からは採餌前に解の評価を行い、経路長が閾値を下回ったエージェントはダンスを行う。表 1 の値に従ってダンスを観察する割合を算出し、該当するエージェントはダンスを観察する。その後採餌を行い、以降規定回数 (5,000 回) まで繰り返す。これを 10 試行した結果を表 2 に示す。

4.1 実験結果

表 2 に実験結果を比較したものを示す。

使用するアルゴリズムは、比較対象を BCO、閾値の自動調整を付与したものを +T と表記した。それぞれの試行について、以下のものを計測した。

- Finding Optimal: 10 試行中最適解を発見した回数
- The Best: 10 試行中の最短経路長の最適解に対する誤差
- Average length: 各試行での最短経路長の平均の最適解に対する誤差
- Average Iterations to the best: 最短経路を発見するまでにかかったサイクル数の平均
- Average Iterations of latest renew: 最後に最短経路長が更新されたサイクル数の平均

最短経路長と平均経路長の最適解に対する誤差は式(11)で表される。

$$\text{error} = \frac{x - \text{opt}}{\text{opt}} \quad (11)$$

x : 経路長

opt: 最適解の経路長

表 2 実験結果比較

Problem Instance: 都市データ名 (および最短経路長)

Problem Instances		Finding Optimal	The Best	Average length	Average Iterations to the best	Average Iterations of latest renew
att48 (3351)	BCO	10/10	0.0	0.0	884	884
	+T	10/10	0.0	0.0	1995	1995
eil76 (5454)	BCO	0/10	1.06E-3	2.90E-3	4055	2785
	+T	0/10	8.50E-4	2.71E-3	1707	2002
lin105 (14383)	BCO	9/10	0.0	1.18E-2	2597	2837
	+T	10/10	0.0	0.0	1896	1896

2-opt とは、経路に含まれる 2 つの隣り合わない弧を選び、それぞれの頂点をつなぎ換えることで、より経路長が短くなるよう部分修正を繰り返す手法である。TSP に対して非常に強力であり、実験時間の短縮のためこれを導入した BCO を用いた。

BCO は、att48, lin105 では常に最適解を発見し、最短経路（最適解とは限らない）との誤差(The Best)は全試行で 0.0 であった。しかし、eil76 では最適解を一度も発見できず、最短経路を発見するまでにかかった平均サイクル数(Average Iterations to the best)も多い。

これに対し、+T では eil76 と lin105 において、BCO よりも早い段階で最適解を発見している。よって、閾値の自動調整により適切な数のダンサーを確保し続けることは解探索に対し有効であると考えられる。

しかし、eil76 では最適解が発見できていないにも関わらず、最後に最短経路長が更新されたサイクル数の平均(Average Iterations of latest renew)が BCO よりも短くなっている。これは、解探索の早い段階で探索がそれ以上の改善を行うことができなかったことを示しており、探索の終盤においてもダンサーを確保し、解の探索を続けるという当初の構想を外れている。

また、att48 では最適解は発見しているが、平均サイクル数が大幅に増えている。これは、ダンサーの供給過多によって解の収束が遅くなったためと考えられる。

このように、都市数によって解探索の傾向が大きく異なることが明らかになったため、さらに状況に合わせた自動調整を行うことが今後の課題である。

5. まとめ

BCO に対し、蜜蜂の採蜜行動における閾値調整機能を導入した結果を比較した。

eil76, lin105 においては速やかに最短経路を発見することができ、閾値の自動調整は解探索に対し有効であることを確認した。

しかし、都市数によって解探索の傾向が大きく異なることが明らかになったため、さらに状況に合わせた自動調整を行うことが今後の課題である。

謝辞 本研究にあたり直接の御指導を頂いた名古屋大学情報科学研究科複雑系科学専攻森敏彦名誉教授に深謝する。

参考文献

- 1) トーマス・D・シーラー：ミツバチの知恵 - 蜜蜂コロニーの社会生理学, 青土社 (1998)
- 2) Li-Pei Wong, Malcolm Yoke Hean Low, and Chin Soon Chong : A Bee Colony Optimization Algorithm for Traveling Salesman Problem, AICMS, In: Second Asia international conference on modeling and simulation, AICMS 08, pp 818-823 (2008)

3) Li-Pei Wong, Malcolm Yoke Hean Low, and Chin Soon Chong : Bee Colony Optimization with Local Search for Traveling Salesman Problem, IEEE, in Proc. of 6th IEEE International Conference on Industrial Informatics(INDIN 2008), IEEE, pp. 1019-1025 (2008)

4) Li-Pei Wong Malcolm Yoke Hean Low, and Chin Soon Chong : An efficient bee colony optimization algorithm for traveling salesman problem using frequency-based pruning, IEEE, in Proc. of the 7th IEEE International Conference on Industrial Informatics(INDIN 2009), IEEE, pp. 775-782 (2009)

5) Arvinder Kaur and Shivangi Goyal : A Survey on the Applications of Bee Colony Optimization Techniques, International Journal on Computer Science and Engineering (IJCSSE), ISSN : 0975-3397, Vol. 3 No. 8 August 2011, pp. 3037-3046 (2011)

6) DUSAN TEODOROVIC, TATJANA DAVIDOVIC and MILICA SELMI: Bee Colony Optimization: The Applications Survey, copyrights@ 2011 ACM 1529-3785/2011/0700-0001