

ACO: アントコロニー最適化

筒井 茂義*

1. はじめに

アントコロニー最適化 (Ant Colony Optimization, ACO) は, 実際のアリの採餌行動の際の経路生成過程にヒントを得た探索手法であり, 巡回セールスマン問題 (Traveling Salesman Problem, TSP) など多くの組合せ最適化問題に適用され, 有効な結果が得られている. アリはフェロモンを介したコミュニケーションを行いながら群れで行動し, ある種の秩序を形成する. ACO では, この秩序形成過程を探索に用いる.

ACO の基本モデルは, Dorigo らによる Ant System (AS) [7] とよばれるアルゴリズムに代表されるが, その原型は, Dorigo の初期の研究の 1990 年代にさかのぼる [4,5]. しかし, AS の性能は他のメタヒューリスティックと比較して十分なものではなかった. そのため, その後多くの改良型 ACO アルゴリズムが提案されている [8]. これらは, 探索過程における多様化 (exploration) と集中化 (exploitation) のバランスをとることを考慮しており, Rank-based Ant System (AS_{rank}) [3], Ant Colony System (ACS) [6], Max-Min Ant System (MMAS) [20] や経路選択にランダムネスを導入するアルゴリズム [25] などの研究がある.

その中で優れた性能を有するアルゴリズムの代表例は ACS と MMAS である. また, 筆者らは, 先にカニングアントシステム (cunning Ant System, cAS) とよぶ新しい ACO アルゴリズムを提案し, TSP および 2 次割当て問題 (Quadratic Assignment Problem, QAP) を用いて, ACO アルゴリズムの中で最も優れた一つであることを示した [22,27,28].

本稿では, ACO の基本原理およびそれに基づく手法を述べ, また, 改良された ACO の特徴と ACO の応用について述べる. また筆者らの cAS の研究の紹介を行う. 以下, 2. において ACO の基本モデルである AS について述べる. つぎに 3. において代表的な拡張型 ACO アルゴリズムを紹介する. 4. では, 筆者らの cAS の研究紹介を行う. 5. では, ACO の QAP の解法への応用事例を紹介する. 最後に, 6. では, ACO の進化計算における位置づけや応用に当たった課題について考察する.

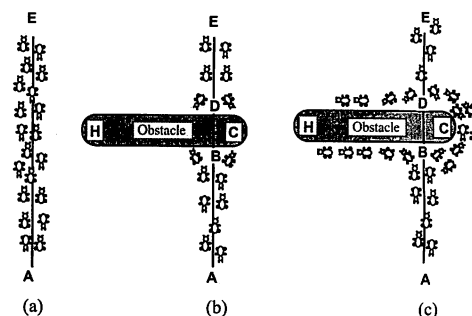
* 阪南大学 経営情報学部

Key Words: ant colony optimization, ant system, ant colony system, cunning ant system.

2. ACO の基本モデル: Ant System (AS)

AS[7] は, ACO の基本となるものである. ここでは代表的な組合せ最適化問題である TSP への応用を例にその動作原理をやや詳しく述べる. 実際のアリが, 餌場から巣までを短い距離で往復する経路を見つけるのにフェロモンを介した協調行動を用いていることが知られている. Dorigo らはアリのこの行動をヒントとするアルゴリズムとして ACO の基本スキームを提案した [4,5].

アリがフェロモンを介して, 2 点間の短い経路を見つける基本原理を文献 [7] をもとに第 1 図に示す. ここで, (1) アリは通過した経路にフェロモンを排出する, (2) アリはフェロモンに誘引されて経路を選択する, というアリの行動原理を前提にしている.



第 1 図 実際のアリが短い経路を見つける過程 [7] (1083-4419/96 ©1966 IEEE)

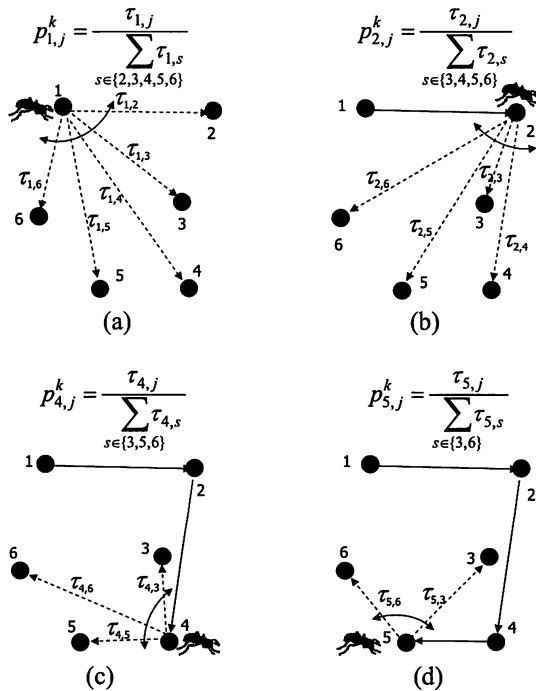
同図 (a) で, A-E 間にアリの群れの経路が形成され, A, E の双方向からアリが移動している状況である. 同図 (b) では, その経路に障害物が置かれた状況を示している. 各アリが障害物を避けてそれぞれの目的点 A, E に行くには, B, D 点で C 点を経由するか H 点を経由するかを選ばなければならない. もちろん, C 点を経由する方が経路長は短い, アリはどちらの経路の方が短いについて知る手立てを持っていない. そこで, 経路の選択はランダムとなる.

同図 (c) は, その後の経過である. 各アリの走行速度が同じであるとする, 経路長の長い H 点経由の経路を選んだアリはまだ障害物を越えていない. しかし経路長の短い C 点経由の経路を選んだアリはすでに障害物を越え, A 点から E 点に向かったグループは D 点に, E 点から A 点に向かったグループは B 点に, それぞれ到着して

いる。アリは経路にフェロモンを排出して移動するとしているので、この結果C点経由の経路の方がH点経由の経路よりもフェロモン濃度が高くなる。したがって、新たにA点からB点に到着したアリ、あるいはE点からD点に到着したアリは、C点経由の経路を選ぶ度合いが高くなる。その結果C点経由の経路のフェロモン濃度はさらに高まる。

フェロモンは蒸発する性質があるので、少数のアリしか通らなくなったH点経由の経路は最終的にフェロモンがなくなり、C点経由の経路が最短経路として最終的に確立する。

ASは、この基本原理を比較的忠実にモデル化したものである。TSPへの適用では、各アリは m 個からなるエージェントとして扱われる。各エージェントは、各都市に均等もしくはランダムに配置され、そこを出発点としてTSPの巡回路を形成する。このとき、各エージェントは、フェロモン濃度に比例して確率的に経路を選択する。二つの都市 i, j 間の経路（エッジ）のフェロモン濃度を τ_{ij} としよう。なお、ここで対称TSPの場合には $\tau_{ij} = \tau_{ji}$ であり、非対称TSPの場合には $\tau_{ij} \neq \tau_{ji}$ となる。このとき、一度行った都市は訪問しないというTSPの規則に従う。この経路選択の過程を第2図に示す。同図(a)は、都市1から出発するエージェント k を示している。都市にいるエージェント k がつぎに訪問できる都市の集合は $\{2, 3, 4, 5, 6\}$ である。そこで、これらの都市 j ($j \in \{2, 3, 4, 5, 6\}$)を選択する確率 $p_{1,j}^k$ は、(a)に示されているようになる。(b)は、この確率に基づいて都市2が選ばれた状況である。都市2にいるエージェント k がつぎに訪問できる都市の集合は $\{3, 4, 5, 6\}$ である



第2図 巡回路生成過程

ので、このエージェントがこれらの都市 j ($j \in \{3, 4, 5, 6\}$)を選択する確率 $p_{2,j}^k$ は(b)に示されているようになる。(c)は、この確率に基づいて都市4が選ばれた状況である。以下、同様に訪問する都市を順次決定してエージェント k はTSPの巡回路を完成させる。

各都市に配置された m 個のエージェントがTSPの巡回路を完成させる動作を1サイクルとする。このときフェロモン濃度は、次式によって更新される。

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

$$\Delta \tau_{ij}^k = \begin{cases} \frac{1}{C_k}, & \text{if } (i,j) \in T_k \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

ここで、 ρ は蒸発係数であり、 $(1-\rho)$ がサイクル t と $t+1$ の間にフェロモンが蒸発する割合を示している。また、 $\Delta \tau_{ij}^k$ はエージェント k により経路 (i, j) に新たに排出されるフェロモン濃度である。この値は、エージェント k の巡回路 T_k の長さ C_k が短いほど大きな値となるようにするために、 C_k の逆数とする。

初期状態ではすべての経路に同じ濃度 C のフェロモンが存在すると考える。すなわち、 $\tau_{ij}(0) = \tau_0$ とする。 ρ の値は試行錯誤法により決定しなければならないが、 $0.8 \sim 0.98$ の値が用いられることが多い。

第2図において説明した選択確率は、ASでは(2)式のように一般化される。すなわち、都市 i にいるエージェント k が都市 j に移動する確率 $p_{ij}^k(t)$ を次式で定義する。

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)] \cdot [\eta_{ij}]^\beta}{\sum_{s \in J_k(i)} [\tau_{is}(t)] \cdot [\eta_{is}]^\beta}, & \text{if } j \in J_k(i) \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

ただし、 $J_k(i)$ は、都市 i にいるエージェント k が訪問できる都市（まだ訪問していない都市）のリストであり、 β はフェロモン濃度 $\tau_{ij}(t)$ と η_{ij} との重要性の度合いを制御するパラメータである。ここで、 η_{ij} は(2)式を選択確率にバイアスを与えるものであり、TSPの場合には経路の選択で、より近隣の都市を選ぶことが好ましいというヒューリスティックがあり、 $\eta_{ij} = 1/d_{ij}$ （ただし、 d_{ij} は都市 i, j 間の距離）が使われる。どのように η_{ij} を決定するかは与えられた問題に依存し、そのようなヒューリスティックが存在しない場合には $\eta_{ij} = 1$ とする。

(2)式に基づいてすべてのエージェントが巡回路の生成を終了したとき、フェロモン濃度が(1)式に基づいて更新されるが、以下、 $t \leftarrow t+1$ として終了条件が満たされるまでこのサイクルが繰り返される。

3. ASの拡張

先に述べたように、ASは、ACOアルゴリズムの中で、アリの採餌行動における経路形成過程を比較的忠実にモデル化したものであるが、30都市程度のTSPを解く能

力しか有しない [7]. この原因の一つは, (1) 式から明らかなように, 短い巡回路を生成したエージェントがより高い濃度でフェロモンを排出するとしているが, 各経路のフェロモン濃度がある程度収束した状況を考えると, エージェント間でのフェロモン排出濃度の差はわずかとなり, 探索における集中化が弱くなるためである. 本章では, 代表的な拡張型 ACO である ACS および MMAS を紹介する. 筆者らの研究による cAS については章を改めて 4. で詳しく紹介したい.

3.1 Ant Colony System (ACS)

ACS[6] では, 巡回路生成を (2) 式に基づく確率に加えて, フェロモン濃度の高い経路を決定論的に選ぶ方法を加味した疑似確率的方法 (pseudo-random proportional rule) を用いて集中化の度合いを強化している. すなわち, 現在都市 i にいるエージェント k がつぎに行く都市 j の決定には, 次式を用いる.

$$j = \begin{cases} \arg \max_{x \in J_k(i)} ([\tau_{ix}(t)] \cdot [\eta_{ix}]^\beta), & \text{if } q \leq q_0 \\ S, & \text{otherwise} \end{cases} \quad (3)$$

ここで, q は $[0, 1]$ の一様乱数であり q_0 は制御パラメータである. また, S は (2) 式に基づいて確率に決定される都市である. このように確率 q_0 で $([\tau_{ix}(t)] \cdot [\eta_{ix}]^\beta)$ が最大となる経路が決定論的に選ばれるので集中化の度合いが強化される.

また, ACS では AS と異なり, 探索開始から現在までに得られた最良解を最良エージェント (以下, best-so-far) として保存する. 各サイクルにおけるフェロモン放出は, この best-so-far によってのみ行われ, (4) 式のように $\tau_{ij}(t)$ の更新が行われる. これにより集中化を促進する. この更新はグローバル更新 (global update) とよばれる.

$$\tau_{ij}(t+1) = \rho_g \cdot \tau_{ij}(t) + (1 - \rho_g) \Delta \tau_{ij} \\ \Delta \tau_{ij} = \begin{cases} (L_{gb})^{-1}, & \text{if } (i, j) \in \text{best-so-far} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

ここで, ρ_g ($0 < \rho_g < 1$) はグローバル更新に使われるパラメータである.

(4) 式は, AS の (1) 式とは異なり, 強化学習にヒントを得た形となっている. また更新される $\tau_{ij}(t)$ は経路 (i, j) が best-so-far に含まれるもののみである. なお, L_{gb} は, best-so-far の巡回路長である. 初期フェロモン濃度値 τ_0 には, $(n \times L_{nn})^{-1}$ を用いることが推奨されている. ただし, n は都市数である. L_{nn} は適当な都市から出発して, まだ訪問していない都市で現在都市から最も近い都市へ移動し, すべての都市を訪問したら出発都市へ戻ることによって巡回路を生成するという最も単純な TSP の近似解法 (Nearest Neighbor 法) による巡回路長である.

以上述べた工夫は, 探索における集中化を促進するが, これに対して多様化を図る機構がローカル更新 (local update) である. ローカル更新では, 各エージェントが (3) 式に基づいて一つの経路 (i, j) が選択されるごとに, 選択されたその経路のフェロモン濃度がある基準値 τ_{base} に近づけるよう次式により更新される.

$$\tau_{ij}(t+1) = \rho_l \cdot \tau_{ij}(t) + (1 - \rho_l) \tau_{base} \quad (5)$$

ここで, ρ_l ($0 < \rho_l < 1$) はローカル更新に用いるパラメータである.

ローカル更新では, (5) 式からわかるように, もし $\tau_{ij}(t)$ が τ_{base} よりも大きい場合には, $\tau_{ij}(t+1)$ が小さくなる方向に更新される. これにより, 他のエージェントがこの経路以外を選択することを促進する. 逆に, $\tau_{ij}(t)$ が τ_{base} よりも小さい場合には, $\tau_{ij}(t+1)$ が大きくなる方向に更新される. すなわち他のエージェントがこの経路を選択することを促進する. なお, τ_{base} には初期フェロモン濃度値 τ_0 を用いることが推奨されている.

ローカル更新は, 実際のアリによるフェロモン更新モデルとは離れた機構となっているが, フェロモン濃度の多様化を促進するという点で, 興味ある機構である. なお, ρ_g, ρ_l にはともに 0.9 が用いられることが多い.

3.2 Max-Min Ant System (MMAS)

MMAS[20] でも, ACS のグローバル更新と類似して, フェロモンの放出は各サイクルで新たに生成されたなかで最良のエージェント (以下, iteration-best) もしくは best-so-far のみとし, 集中化の度合いを高める. フェロモン濃度の更新式は, 以下の通りである.

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \Delta \tau_{ij}^{best} \\ \Delta \tau_{ij}^{best} = 1/C_t^{best-so-far}, \text{ or } 1/C_t^{iteration-best} \quad (6)$$

ただし $C_t^{best-so-far}, C_t^{iteration-best}$ はそれぞれ, best-so-far および iteration-best の巡回路長を示している. 蒸発係数 ρ には AS とほぼ同様の値が用いられる. MMAS では (6) 式の更新において $C_t^{best-so-far}$ と $C_t^{iteration-best}$ とが併用される. どのようにその併用戦略を決めるかが MMAS のチューニング項目の一つとなる.

MMAS の最大の特徴はその名称が示すとおり, フェロモン軌跡濃度を $[\tau_{min}, \tau_{max}]$ の区間に限定する点にある. 最小濃度 τ_{min} の導入は, 遺伝的アルゴリズム (Genetic Algorithm, GA) における突然変異と同様の効果があり, 多様化の度合いを高める機構となる. なお, τ_{min} は, (7) 式によるフェロモン濃度の理論的最大値 τ_{max} との相対値として, (8) 式のように決定される.

$$\tau_{max}(t) = 1/(1 - \rho) \times 1/C_t^{best-so-far} \quad (7)$$

$$\tau_{min}(t) = \frac{\tau_{max} \cdot (1 - \sqrt[p_{best}]{p_{best}})}{(n/2 - 1) \cdot \sqrt[p_{best}]{p_{best}}} \quad (8)$$

ここで、 $C_t^{best-so-far}$ はサイクル t における best-so-far の巡回路長、 n は都市数、 p_{best} は制御パラメータである。その他、MMAS では、フェロモン軌跡濃度の再初期化 (re-initialize) やフェロモン平滑機構 (pheromone trail smoothing; pts) を含ませることが提案されている。

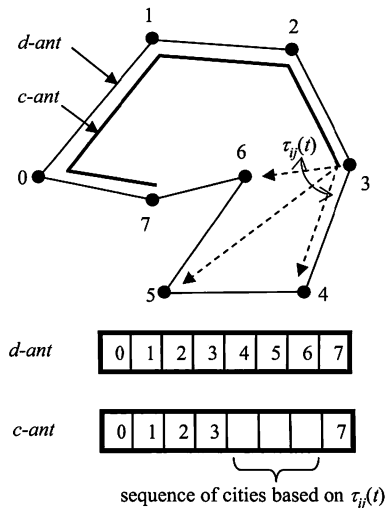
4. cAS：カニングアントシステム

cAS は筆者らが提案している ACO アルゴリズムであり、TSP および QAP で評価を行い、他の改良型 ACO アルゴリズムに比較して十分高い性能を有していることを明らかにした [22,27,23,28]。本章では、TSP への cAS の応用を例に、cAS の研究を紹介しよう。

4.1 カニングアント

cAS の特徴はカニングアント (cunning ant, $c\text{-ant}$) とよぶエージェントを導入している点にある。 $c\text{-ant}$ は、従来の ACO におけるエージェントと異なり、新しい解候補の生成の際に、過去の探索において存在する解の一部分 (部分解) を借用し、残りの部分の生成には従来と同様、フェロモン濃度を利用する。 $c\text{-ant}$ に借用を許すエージェントをドナーアント (donor ant, 以下 $d\text{-ant}$) とよぶ。

第3図は、TSP における $c\text{-ant}$ と $d\text{-ant}$ とを示している。この例では、 $c\text{-ant}$ は部分巡回路 $7 \rightarrow 0 \rightarrow 1 \rightarrow 2 \rightarrow 3$ を $d\text{-ant}$ から借用し、残りの都市 4, 5 および 6 の部分巡回路 $\tau_{ij}(t)$ を基に (2) 式により確率的に生成する。



第3図 TSP における $c\text{-ant}$ と $d\text{-ant}$

4.2 コロニーモデルとフェロモン濃度の更新

cAS では、 m 個のエージェント $s_t^k (k \in \{1, 2, \dots, m\})$ がアーカイブに保存され、それぞれが各 k の $d\text{-ant}$ として使われる。新しく生成された $c\text{-ant}$ は、それぞれの $d\text{-ant}$ と比較され、よい方が次のサイクルにおけるアーカイブのメンバー s_{t+1}^k となる。

cAS のフェロモン濃度更新は、AS の (1) 式と同様であるが、フェロモンの排出は、アーカイブの m 個のエー

ジェントによって行われるという点が異なる。また、フェロモン濃度は MMAS と同様 $[\tau_{min}, \tau_{max}]$ の区間に限定される。なお、蒸発係数 ρ の値は、AS とほぼ同様の値が適切である。

4.3 $c\text{-ant}$ 生成における借用部分の決定法

cAS の特徴は以下の2点に集約できる。まず、解候補の集合であるアーカイブにあるエージェントの部分解を利用するため、効果的に集中化が図られる。つぎに、 $c\text{-ant}$ の生成においては、新たに生成する解は、 $d\text{-ant}$ から借用する部分を除いた部分解である。したがって、 $d\text{-ant}$ から借用する割合を適切にすることで、収束速度を緩和し、多様性の維持を図ることができる。なお、 $c\text{-ant}$ の生成は、 $d\text{-ant}$ とフェロモン濃度による生成解との交叉のような形になっている。この点から、cAS は既存解の効果的な継承を行う GA と同様の特徴も有すると解釈できる。

$c\text{-ant}$ が $\tau_{ij}(t)$ に基づいてサンプルするノードの数を l_s 、 $d\text{-ant}$ から借用するノードの数を l_c とすると、 $l_c = n - l_s$ である。ここで、 l_s の平均値 $E(l_s)$ が $E(l_s) = n \times \gamma$ となる制御パラメータ γ を導入し、そのような条件を満たす l_s の分布を決めるものとして、以下の確率密度関数 $f_s(l)$ が用いられている。

$$f_s(l) = \begin{cases} \frac{1-\gamma}{n\gamma} \left(1 - \frac{l}{n}\right)^{\frac{1-2\gamma}{\gamma}} & \text{for } 0 < \gamma \leq 0.5 \\ \frac{\gamma}{n(1-\gamma)} \left(\frac{l}{n}\right)^{\frac{2\gamma-1}{1-\gamma}} & \text{for } 0.5 < \gamma < 1 \end{cases} \quad (9)$$

したがって、(2) 式に基づいてサンプルされる都市の数は、平均値は $n \times \gamma$ となるが、サンプルごとに確率的に変動する。(9) 式の意味の詳細については文献 [27] を参照していただきたい。cAS では、適切な γ 値 ($0 < \gamma \leq 1$) の設定が性能に大きな影響を与えるので、その設定が重要となる。

4.4 cAS の TSP での性能

4.4.1 小規模問題での結果

ここでは、cAS を ACO の中で優れた性能を持つ MMAS および ACS の結果と比較する。最大経路生成回数は、文献 [20] および [10] と同じとし、 $k \times n \times 10000$ とする。ここで、対称 TSP (STSP) のとき $k=1$ 、非対称 TSP (ATSP) のとき $k=2$ である。cAS の m の値も比較アルゴリズムと同様、 $m=n$ 、すなわち都市数と同じとし、また、 γ の値は 0.4 とする。また、 $\rho=0.98$ 、 $\beta=2$ である。結果の数値はいずれも 25 回の実行の平均とする。この結果を第1表に示す。なお、MMAS および ACS の結果は、それぞれ、文献 [20] および [10] で報告されたものである。同表において $Best_{avg}$ はベスト解の平均値であり、 $Error$ は、 $(Best_{avg} - \text{最適解}) / \text{最適解}$ を示す。同表で、最も良い結果 (最小の $Error$ 値) を示したアルゴリズムのデータを太字で示している。この結果から、

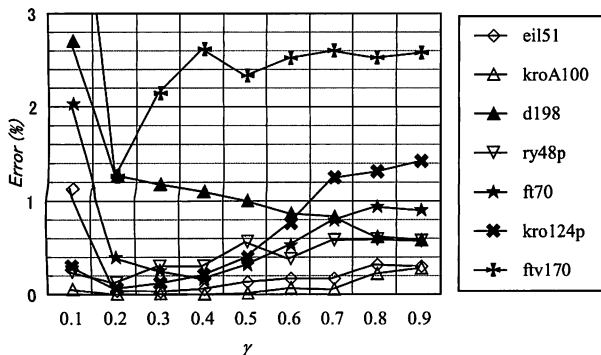
cAS は ftv170 問題を除くすべての問題で最も良い結果を示していることがわかる。

c-ant を用いる効果を見るために第1表に示した各問題において *c-ant* が τ_{ij} に基づいて解を生成する割合を示すパラメータ γ を 0.1 から 0.9 まで 0.1 刻みで変化させた場合の結果を第4図に示す。d198を除いて、 γ の値が [0.2, 0.6] の領域で良い結果を示している。大きい問題である d198 では、 γ の値が先の範囲よりも大きい領域でさらに良い結果を示している。第5図は、100都市対称 TSP である kroA100 問題における収束状況を各種 γ に対して *Error* の変化を用いて示したものである。

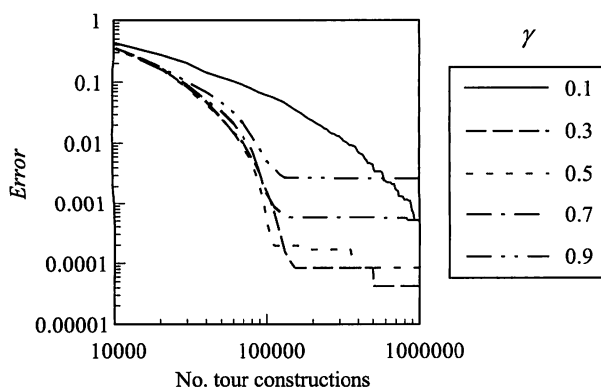
同図から明らかなように、 γ 値が 0.7 および 0.9, すなわち、借用するサブツアーが少ないときには初期段階で探索の停滞が起こっている。 γ 値が 0.3 および 0.5 のときには、探索の停滞はもっと後となっている。また、 γ 値が 0.1 のとき、すなわち、平均して 90% の経路を *d-ant* から引き継ぐ場合には、探索の停滞は起こらないが収束は非常に遅くなることがわかる。4.3 で予測したように、適切な γ 値を用いることにより、多様性の維持を図りつつ探索が進むという cAS の特徴の一端が確認できる。

4.4.2 cAS とローカルサーチとの結合

4.4.1 では比較的小規模な問題に cAS を適用して評価を行った。一般に大きな問題を解くには、ベースとするアルゴリズムにローカルサーチを組み合わせるのが一般



第4図 γ の変化に対する *Error* の影響



第5図 各種 γ における収束状況 (kroA100)

的である。これは ACO においても同様である [8]。ここでは、cAS に TSP のローカルサーチとして最も強力な一つである Lin-Kernighan 法 (以下、LK 法) [17] を組み合わせる。

LK 法の実装には各種のものがあるが [14]、その中で有力な方法は、LK 法の基本アルゴリズムを繰り返して適用する方法であり、それらは Chained LK [2] とよばれている。Chained LK では、基本 LK 法により得られた解に *kick* とよばれる変動を加え、基本 LK 法を繰り返して適用する。本研究では、Applegate らによって開発された *Concorde TSP solver* (Concorde) とよばれる Chained LK を用いる [1]。

本研究の cAS は Java で記述しており、また、Concorde は C 言語で記述されている。そこで、両者は Java Native Interface (JNI) を用いて結合した。cAS により各順回路を生成した後、それらを初期解として n 回 (n = 都市数) の基本 LK 法を適用する。

テスト問題は、数 100 ~ 数 1000 都市問題として att532 ($n=532$), d1291 ($n=1291$), vm1748 ($n=1748$), pr2392 ($n=2392$), fl3795 ($n=3795$) および rl5934 ($n=5934$) を用いる。最大実行時間 (T_{max}) は、各問題に対してそれぞれ 40, 80, 200, 240, 1400 および 3300 秒とした。なお、使用した計算機は Opteron 275×2 (2.4GHz, 2GB メモリ), OS は 32 ビット WindowsXP である。cAS のパラメータは、 $m=5$, $\rho=0.5$, $\gamma=0.4$ とした。

比較アルゴリズムとして、MMAS+LK および Chained LK 単独を用いる。なお、MMAS は筆者の実装による Java コードを用いた。

結果を第2表にまとめる。MMAS の結果には、比較的大きな問題である fl3795 および rl5934 で最も良い最適解発見回数 ($\#OPT$) を示した $\rho=0.8$, $m=5$, フェロモン濃度更新戦略 = *best-so-far* のものを示した。

Error の結果に着目すると、cAS, MMAS のいずれのアルゴリズムとも非常に小さい値であり、Chained LK 単独に比較して、これらのアルゴリズムと LK 法を結合する方法の効果は明白となっている。

つぎに、最適解発見回数 $\#OPT$ の結果に注目すると、MMAS は att532 を除くすべての問題において $\#OPT=25$ が得られていない。これに対して、cAS はすべての問題に対して $\#OPT=25$ が得られている。また、最適解を得るのに必要とした時間の平均 (T_{avg}) も比較対象のアルゴリズムの中で最小となっている。

第6図は、 γ 値を変えた場合の T_{avg} の変化を示したものである。cAS+LK において、 $\gamma \in [0.3, 0.5]$ で良い結果を示すことがわかる。この γ の値は、ローカルサーチを用いない場合の第4図の結果とほぼ一致している。

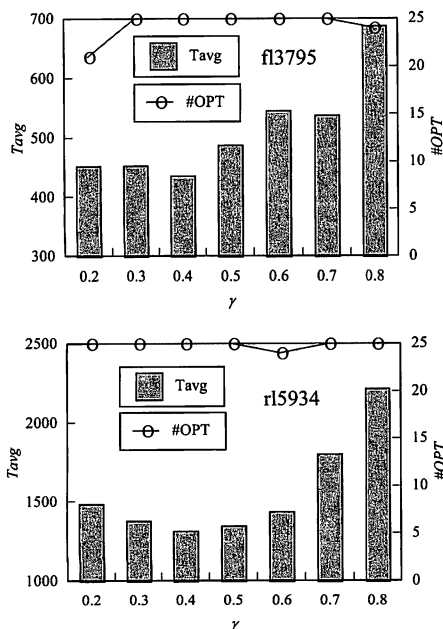
第 1 表 cAS の結果 $Best_{avg}$ は 25 回の実験の平均値, $Error$ は $Best_{avg}$ の最適解からの差の割合を示す.

TSP		opt	c AS ($\gamma=0.4$)			MMAS				ACS	
						MMAS+pts		MMAS			
			$Best_{avg}$	std^*	$Error$ (%)	$Best_{avg}$	$Error$ (%)	$Best_{avg}$	$Error$ (%)	$Best_{avg}$	$Error$ (%)
STSP	eil51	426	426.2	0.5	0.06	427.1	0.26	427.6	0.38	428.1	0.48
	kroA100	21282	21282.0	0.0	0.00	21291.6	0.05	21320.3	0.18	21420.0	0.65
	d198	15780	15954.1	35.6	1.10	15956.8	1.12	15972.5	1.22	16054.0	1.74
ATSP	ry48p	14422	14465.4	34.9	0.30	14523.4	0.70	14553.2	0.91	14565.5	0.99
	ft70	38673	38736.1	77.1	0.16	38922.7	0.65	39040.2	0.95	39099.1	1.10
	kro124p	36230	36303.2	120.3	0.20	36573.6	0.95	36773.5	1.50	36857.0	1.73
	ftv170	2755	2827.1	8.7	2.62	2817.7	2.28	2828.8	2.68	2826.5	2.59

* std : standard deviation of $Best_{avg}$

 第 2 表 ローカルサーチ (LK) を適用した cAS の結果

TSP	$cAS(\gamma=0.4)$				MMAS				Chained LK			T_{max}
	$LdO\#$	$Error$ (%)	I_{avg}	T_{avg} [min, max]	$LdO\#$	$Error$ (%)	I_{avg}	T_{avg} [min, max]	$LdO\#$	$Error$ (%)	T_{avg} [min, max]	
att532 ($n=532$)	25	0.00	1.8	7.8 [1.4, 27.8]	25	0.00	2.4	10.5 [1.4, 32.6]	17	0.02	6.11 [0.3, 28.5]	40
d1291 ($n=1291$)	25	0.00	5.7	27.4 [6.0, 54.4]	22	0.00	10.3	48.8 [6.1, 74.1]	6	0.12	17.0 [4.0, 61.3]	80
vm1748 ($n=1748$)	25	0.00	5.6	72.4 [8.4, 171.0]	21	0.00	5.6	78.4 [8.3, 173.0]	1	0.06	72.8 [-]	200
pr2392 ($n=2392$)	25	0.00	10.1	104.9 [33.7, 190.0]	12	0.00	20.4	211.3 [170.3, 233.2]	4	0.17	122.4 [40.2, 222.1]	240
fl3795 ($n=3795$)	25	0.00	9.8	435.1 [102.8, 1228.7]	17	0.00	17.6	770.7 [159.9, 1081.1]	0	0.57	-	1400
rl5934 ($n=5934$)	25	0.00	43.2	1336.1 [729.1, 1996.8]	10	0.00	82.8	2533.6 [1499.2, 2897.0]	0	0.27	-	3300


 第 6 図 γ の変化に対する T_{avg} の結果

5. ACO の応用事例：2 次割当て問題への応用

ACO と TSP とは, ACO のヒントとなったアリの行動モデルからわかるように直接的な関係があるので, 前章までは, TSP を用いて ACO の解説を試みた. 事実 ACO の改良の研究には TSP が用いられることが多い. しかし, ACO は, 多くの NP 困難な組合せ最適化問題に適用され, 有用な結果が報告されている [8].

その代表的な応用分野としては, 非対称 TSP の一般系であるシーケンシャル順序問題, 配送路問題, QAP, 通信における周波数割当て問題, フローショップ問題, ジョブショップ問題, 時間割作成問題, ネットワークルーティング問題など多岐にわたる. また, 近年実数最適化問題への適用も試みられている [19, 26]. 本章では, ACO の応用事例として Stützle らによる MMAS の QAP への応用の研究 [20] を紹介する.

QAP は, n 個からなる部門を n 個の場所に, 次式で定義される関数値が最小になるように割り当てるという組合せ最適化問題である.

$$f(\phi) = \sum_{r=1}^n \sum_{s=1}^n b_{rs} a_{\phi(r)\phi(s)} \quad (10)$$

ここで、 $A = (a_{pq})$ および $B = (b_{rs})$ はそれぞれ $n \times n$ のマトリックスであり、 ϕ は $\{1, 2, \dots, n\}$ の順列である。マトリックス A と B は、それぞれ、場所 p, q 間の距離、部門 r, s 間の流量を表している。 QAP は評価式が距離と流量の積となっており、両情報が相互に絡み合うため、TSP に比較して格段に困難な問題である。

TSP では、フェロモン濃度 τ_{ij} は、各都市間の経路 (i, j) の選択における好ましさの度合いとして定義され、巡回路生成は連続したシーケンスとして (2) 式に基づいて行われた。これに対して、QAP のフェロモン濃度 τ_{ij} は、部門 i を場所 j に割り当てる好ましさの度合いと定義される。また、 η_{ij} として適切なヒューリスティックがないので (2) 式において $\eta_{ij} = 1$ とされる。

研究 [20] では、ローカルサーチとして Taillard による Robust-Tabu Search (Ro-TS) [21] と 2-OPT が用いられている。Ro-TS との組合せでは、アルゴリズム実行時間が比較対象に使われている他のメタヒューリスティックの実行時間と同一となるように、MMAS により生成された各解に対して、 $4n$ 回繰り返す Ro-TS を適用し、MMAS による総個体生成数を 250、エージェント数を $m=5$ としている。

第3表にこの結果の一部を示す。各値は最適解からの誤差を % で表示したものであり 10 回の実験の平均値である。同表には、比較対象として Ro-TS 単独のほか、AS にサンプリング法などの改良やスワップベースのローカルサーチを付加した Hybrid AS (HAS-QAP)、GA にタブーサーチを組み合わせた Genetic Hybrid 法 (GH) の結果も記述されている。太字で示したものは最もよい結果を示している手法である、MMAS が優位な性能を有していることがわかる。

第3表 MMAS の QAP への応用の結果

Type	QAP	MMAS-TS	MMAS-2OPT	HAS-QAP	GH	Ro-TS
i	tai40a	0.794	1.509	1.989	0.884	0.990
	tai50a	1.060	1.795	2.800	1.049	1.125
	tai60a	1.137	1.882	3.070	1.159	1.203
	tai80a	0.836	1.402	2.689	0.796	0.900
ii	ske56	0.075	0.098	0.101	0.060	0.088
	ske64	0.071	0.099	0.129	0.092	0.071
	ske81	0.062	0.124	0.144	0.136	0.136
	ske90	0.114	0.140	0.231	0.196	0.128
iii	ste36a	0.061	0.126	n.a.	n.a.	0.155
	ste36b	0.000	0.000	n.a.	n.a.	0.081
iv	tai40b	0.402	0.000	0.000	0.211	0.531
	tai60b	0.005	0.005	0.048	0.291	0.417
	tai80b	0.591	0.266	0.667	0.829	1.031
	tai100b	0.230	0.114	n.a.	n.a.	0.512

6. 進化計算としての ACO

6.1 進化計算コミュニティは島モデル

生物進化にヒントを得た進化計算の源流は一般に、1960 年代に Rechenberg に提案され、その後 Schwefel らにより発展させられた進化戦略 (Evolutionary Strategy, ES)、同じく 1960 年代の L. J. Fogel に始まる進化プログラミング (Evolutionary Programming, EP) およ

び 1950 年代から研究が始まり 1975 年の Holland の研究 [13] で基本的な枠組みができた遺伝的アルゴリズム (Genetic Algorithm, GA) の三つがあげられる。これらは、いずれも生物進化をヒントとする探索、適応化手法であるがお互いに遭遇することなく 1980 年代まで研究されてきた。

1989 年に出版された Goldberg の著書、Genetic Algorithms in Search, Optimization and Machine Learning [11] は、GA の研究者に多大な影響を与えた名著である。同教授から直接聞いた話であるが、執筆時点では、EP や ES のことはまったく知らなかったとのことである。事実、同書には ES や EP の記述は見当たらない。EP に関してはお互いに無視しあっていたのかもしれない。ES に関しては、研究論文や報告がドイツ語で書かれていたことが大きな原因と思われる。

国際会議に関しては、GA の研究グループにより 1985 年に国際会議第 1 回 International Conference on Genetic Algorithms (ICGA) [9] が開催され、以降 2 年おきに開催されてきた (1999 年からより Genetic Programming Conference と統合して現在の毎年開催される GECCO となっている)。当初 ICGA は国際会議とはいえローカルな研究仲間の会議であったが、1989 年ごろから ES の研究者も加わりはじめ、1990 年代には GA、ES を含む進化計算の中心的国際会議となった。

一方、ヨーロッパにおいては、もう少し広い枠組みを狙いとした国際会議第 1 回 Parallel Problem Solving from Nature (PPSN) が Schwefel らの提唱により ES の発祥国ドイツで 1990 年に開催された。同会議には Goldberg ら GA の研究者も多く参加している。以降、2 年おきに開催され、現在では進化計算に関するハイレベルな国際会議としての地位を確立している。1994 年になって第 1 回 IEEE International Conference on Evolutionary Computation (ICEC) が開催された。1999 年から新たに、IEEE と Society of EP との共催の形で Congress on Evolutionary Computation (CEC) が開催されている。

このように 1990 年代になって従来のコミュニティが互いに本格的に遭遇し、研究交流が国際会議を通して始まった。この交流は、たとえば GA のコミュニティが重視している交叉が ES に取り入れられるというように方法論的イミグレーションを伴う。また、1990 年代に入り、従来の生物進化をベースとする手法に加え、群れの集団行動をモデルとする本解説の ACO や Particle Swarm Optimization (PSO) [15]、さらには進化計算に統計的手法を融合する分布推定型アルゴリズム (Estimation of Distribution Algorithm, EDA) [12,16] などの手法が、集団 (population based) 探索と選択プロセスを有することを特徴とする進化計算の仲間として加わってきた。

すべての問題に万能なアルゴリズムは存在しない [24] という事実がある。各コミュニティはときには強い自己主張を行い、またときには排他的になりつつも、こ

れらにより進化計算コミュニティの多様性が維持され、相互に影響しながら発展してきた。また、今後も発展し続けるであろう。このように、進化計算コミュニティは、分散 GA の島モデルのようにして進化し、まさに Evolutionary である。

さて、EDA は 2000 年代になり本格的な研究が始まっているが、このアルゴリズムでは、現世代の集団から有望な個体を選択する \Rightarrow この集団の分布推定モデルを作成する \Rightarrow この分布推定モデルに基づいて次世代の個体を生成する、というプロセスを繰り返す。ここで、分布推定にどのようなモデルを用いるかで、各種の研究が行われている [18]。

一方、ACO では、各エージェント（アリ）がその評価値に基づいたフェロモン濃度を放出し、フェロモン濃度マトリックスとしてモデル化する \Rightarrow このフェロモン濃度マトリックスを基につぎのサイクルのエージェントを生成する \Rightarrow これらからフェロモンを放出するエージェントを選択する、というプロセスが繰り返される。このフェロモン濃度マトリックスを分布推定モデルととらえれば、両者は基本的に類似のフレームワークと見なすことができる。もちろん ACO では、EDA のようにエージェントそのものの分布を統計量としてとりあげるのではなく、フェロモンという中間物質の分布を統計量として扱う点は大きく異なる。しかし、このように両者のコンセプトには密接な関係があるという解釈ができることは注目すべきであり、今後両者の相互イミグレーションが期待される。

6.2 ACO の応用における留意事項

本節では、ACO の適用にあたって、いくつかの考慮すべき点について述べたい。5. で紹介した ACO の QAP への適用では、MMAS は他のメタヒューリスティックスに比較して高い性能を有していることが示された。しかし、他の問題領域での組織的な比較研究は少ない。ACO は、GA など他の進化計算と同様、広い枠組みであり、実際の適用にあたってはいろいろな工夫が必要である。しかし、問題領域におけるフェロモンの定義が適切に行える場合、ACO は適用方法が比較的単純かつ直感的であるという点で有利な手法である。この意味で参入障壁の低い手法であり、実問題においてその適用を検討する価値は十分にあるといえる。

ACO の適用にあたっての留意事項として一般的に以下のようなことがあげられる。TSP におけるフェロモン濃度の定義はそのエッジとの関係で直接的なものであったが、一般にはその定義は対象問題に依存する。たとえば、5. の QAP では、フェロモン濃度 τ_{ij} は部門 i を場所 j に割り当てる好ましさの度合いと定義された。このように、問題に応じて適切にフェロモン濃度の定義を行う必要がある。

ACO における解候補生成法である (2) 式の計算量は

$O(n^2)$ であり、大きなサイズの問題では計算時間が問題となる。このような場合には候補リスト (candidate list) の利用や、3.1 で述べた擬似確率の方法の適用などの利用が必要となる。これにより、解候補生成の計算量は $O(n)$ となる。たとえば、TSP では各都市に対して近傍 20 都市程度からなる候補リストを作成し、(2) 式によるサンプリングに利用する。

GA などのメタヒューリスティックスによる探索ではローカルサーチと組み合わせることが一般的であるが、ACO による探索においてもまったく同様である。ACO にローカルサーチと組み合わせる場合、エージェント数 m を 1~10 と小さくすると効果的である。また、(2) 式におけるヒューリスティック関数 η の利用は好ましいが、適切な関数が定義できない問題も多いことに留意する必要がある。

7. おわりに

本稿では ACO について、基本原理ならびに具体的な各種アルゴリズムを紹介し、筆者らが提案している cAS の研究も紹介した。また ACO とその応用の一端について述べた。6.1 で述べたように、ACO は近年研究が活発に行われている EDA との関係がある。このような視点で ACO をみると、EDA での研究の成果を活用することも可能であり、EDA でのモデル学習の研究の成果を取り入れた ACO の研究など今後の発展が期待される。

最後に本稿を終わるにあたり、ACO に関する主要な情報ソースを紹介したい。まず、書籍では、Dorigo らによる書籍 [8] が ACO に関する全般を論述して、有益である。つぎに、国際学会に関しては、PPSN の開催時期 (西暦偶数年 9 月) に合わせて、ACO に関する国際会議 ANTS が開催され、会議録は Springer から LNCS として出版されている。本年度 (2008 年) は、9 月 22 日 - 24 日に 6th International Conference on Ant Colony Optimization and Swarm Intelligence (ANTS 2008) が Brussels で開催され、また、GECCO、CEC などの進化計算の会議では、ACO の Track や Session が組まれる。論文誌では、2007 年に Springer から、Swarm Intelligence が刊行された。

有用なサイトとしては、Dorigo らによる ACO サイト：<http://iridia.ulb.ac.be/~mdorigo/ACO/ACO.html> がある。

(2008 年 7 月 4 日受付)

参考文献

- [1] Ansi c code as gzipped tar file, concorde tsp solver, 2006. <http://www.tsp.gatech.edu/concorde.html>.
- [2] D. Applegate, W. Cook and A. Rohe: Chained linkernighan for large traveling salesman problems; *INFORMS J. on Computing*, Vol. 15, pp. 82-92 (2003)
- [3] B. Bullnheimer, R. F. Hartl and C. Strauss: A new

- rank based version of the ant system: a computational study; *Cent. Europ. J. for Op. Research and Economics*, Vol. 7, No. 1, pp. 25–38 (1999)
- [4] A. Coloni, M. Dorigo and V. Maniezzod: Distributed optimization by ant colonies; *Proc. of the First European Conference on Artificial Life*, No. 1, pp. 134–142 (1991)
- [5] A. Coloni, M. Dorigo and V. Maniezzod: An investigation of some properties of an ant algorithm; *Proc. of the Parallel Problem Solving from Nature Conference (PPSN 92)*, pp. 509–520 (1992)
- [6] M. Dorigo and L. M. Gambardella: Ant colony system: A cooperative learning approach to the traveling salesman problem; *IEEE Trans. on Evolutionary Computation*, Vol. 1, No. 1, pp. 53–66 (1997)
- [7] M. Dorigo, V. Maniezzo and A. Coloni: The ant system: Optimization by a colony of cooperating agents; *IEEE Trans. on SMC-Part B*, Vol. 26, No. 1, pp. 29–41 (1996)
- [8] M. Dorigo and T. Stützle: *Ant Colony Optimization*, MIT Press (2004)
- [9] J. J. Grefenstette (Ed): *Proceedings of the First International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates (1985)
- [10] L. M. Gambardella and M. Dorigo: Solving symmetric and asymmetric tps by ant colonies; *Proc. the IEEE Int. Conf. on Evo. Comp.(ICEC'96)*, pp. 622–627 (1996)
- [11] D. E. Goldberg: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley (1989)
- [12] H. Mühlenbein and G. Paaß: From recombination of genes to the estimation of distribution i. binary parameters; *Proc. of the Parallel Problem Solving from Nature (PPSN IV)*, pp. 178–187 (1996)
- [13] J. H. Holland: *Adaptation in Natural and Artificial Systems*, The University of Michigan Press (1975)
- [14] D. S. Johnson and L. A. McGeoch: The traveling salesman problem: a case study in local optimization; *Local Search in Combinatorial Optimization*, E. H. L Aarts and J. K. Lenstra (ed.), pp. 215–310 (1997)
- [15] J. Kennedy and R. C. Eberhart: Particle swarm optimization; *Proc. of the IEEE Int. Conf. on Neural Networks*, pp. 1942–1948 (1995)
- [16] P. Larranaga and J. A. Lozano (eds.): *Estimation of Distribution Algorithms*, Kluwer Academic Publishers (2002)
- [17] S. Lin and B. W. Kernighan: An effective heuristic algorithm for the traveling salesman problem; *Operations Research*, Vol. 21, pp. 498–516 (1973)
- [18] M. Pelikan, D.E. Goldberg and F. Lobo: A survey of optimization by building and using probabilistic models; *Computational Optimization and Applications*, Vol. 21, No. 1, pp. 5–20 (2002) Also IlliGAL Report No. 99018.
- [19] K. Socha: ACO for continuous and mixed-variable optimization; *Proc. of the 4th International Workshop on Ant Algorithms and Swarm Intelligence (ANTS-2004)*, pp. 25–36 (2004)
- [20] T. Stützle and H. Hoos: Max-min ant system; *Future Generation Computer Systems*, Vol. 16, No. 9, pp. 889–914 (2000)
- [21] É.D. Taillard: Robust taboo search for the quadratic assignment problem; *Parallel Computing*, Vol. 17, pp. 443–455 (1991)
- [22] S. Tsutsui: cAS: Ant colony optimization with cunning ants; *Proc. of the 9th Int. Conf. on Parallel Problem Solving from Nature (PPSN IX)*, pp. 162–171 (2006)
- [23] S. Tsutsui: Cunning ant system for quadratic assignment problem with local search and parallelization; *Proc. of the Second International Conference on Pattern Recognition and Machine Intelligence, Springer LNCS 4815*, pp. 269–278 (2007)
- [24] D. H. Wolpert and W. G. Macready: No free lunch theorems for optimization; *IEEE Trans. on Evolutionary Computation*, Vol. 1, No. 1, pp. 67–82 (1997)
- [25] 中道, 有田: Acoにおけるランダム選択に基づく多様性調節の効果; 情報処理学会論文誌, Vol. 43, No. 9, pp. 2939–2947 (2003)
- [26] 筒井: 集合フェロモンシステム (APS): 集合フェロモンの機能をモデルとする実数値探索アルゴリズムの一構成法の提案; 人工知能学会論文誌, Vol. 20, No. 1, pp. 76–83 (2005)
- [27] 筒井: cAS: カニングアントを用いたacoの提案; 人工知能学会論文誌, Vol. 22, No. 1, pp. 29–36 (2007)
- [28] 筒井, 劉, 小島: カニングアントを用いたacoの2次割当問題への適用とその並列化; 情報処理学会論文誌: 数理モデル化と応用, Vol. 49, No. SIG 4 (TOM 20), pp. 45–56 (2008)

著者略歴

筒井 茂義



1944年12月3日生。1969年3月大阪市立大学大学院工学研究科電気工学専攻修了。同年(株)日立製作所入社, 同社中央研究所, システム開発研究所を経て, 1987年より阪南大学に勤務。現在経営情報学部経営情報学科, 兼大学院 企業情報研究科教授。2000年4月–2001年3月イリノイ大学 IlliGAL 客員研究員。経営情報システム, 進化計算の基礎の研究に従事。工学博士。1993年度人工知能学会全国大会優秀論文賞, 1996年度人工知能学会論文賞受賞。人工知能学会, 情報処理学会, IEEE 各会員。