

RA-001

アントコロニー最適化におけるモデル化に依存する探索バイアスの抑制 Inhibition of Modeling-Dependent Search Bias in Ant Colony Optimization

大谷 隆浩† 越野 亮‡ 木村 春彦†
Takahiro Otani Makoto Koshino Haruhiko Kimura

1 まえがき

アントコロニー最適化 (Ant Colony Optimization: ACO) [1] は自然界の蟻が行う、フェロモンを利用した食料採集行動に着想を得たメタヒューリスティック手法であり、組合せ最適化問題を効率的に解く手法として注目されている。ACOには様々なバージョンがあり、AS (Ant System) [2] に始まり、ACS (Ant Colony System) [3]、 AS_{rank} (Rank based Ant System) [4]、 $MMAS$ ($MAX-MIN$ Ant System) [5] などが提案されている。また、これまでに巡回セールスマン問題 (TSP) [5]、2次割当問題 (QAP) [5]、配送計画問題 (VRP)[6] など様々な組合せ最適化問題に応用され、これまでのメタヒューリスティック手法を上回る精度が得られている [7, 8]。

このように ACO は様々な問題において高い性能を示しているが、特定の制約付き最適化問題においては、フェロモンのモデル化に依存した探索バイアスが生じることが知られている。この探索バイアスによる性能低下は2次騙し (second order deception) 効果と呼ばれる [9, 10]。[11] では k -部分木問題 (KCT) において AS を適用する場合、探索を続けるごとに生成される解の評価値が減少していくことが示されている。さらに、[9, 10, 12, 13] ではスケジューリング問題においても、[14] ではナップサック問題や一般化割当問題においても同様の現象が発生することを示している。

2次騙し効果を抑制する方法としては次の2つが考えられる。1つは問題ごとに2次騙し効果が発生しないような適切なフェロモンのモデルを用いることである。[9] においてはジョブショップスケジューリング問題 (JSS) において適切なフェロモンのモデル化を行うことで、2次騙し効果を抑制することに成功している。また、[14] では各種の組合せ最適化問題における適切なフェロモンのモデルを示している。しかしながら、これら方法では問題ごとに適切なフェロモンのモデルを設計しなければならず、また設計したモデルは他の問題においても有効であるとは限らない。

2つ目に考えられる方法は、2次騙し効果を抑制するようなフェロモン量更新ルールを用いる方法である。しかしながら、こちらの方法についてはこれまでに十分な研究がなされているとは言いがたい。KCTにおいては、1回の反復内で見つかった最良解のみがフェロモン量増加に寄与するような更新ルールを用いればこの効果を抑制することが出来る [11]。しかしながら、JSS

においてはこのルールを用いたとしても、効果的な探索が行えないことがわかっている [9, 10]。

本研究では2次騙し効果を抑制するための、新しいフェロモン量更新ルールを提案する。この更新ルールは任意の制約付き組合せ最適化問題に対して容易に適用することが可能である。そして、2次騙しの効果が発生することが知られているジョブショップスケジューリング問題に適用し、提案手法の性能を評価する。

2 アントコロニー最適化 (ACO)

ACO は組合せ最適化問題に対するメタヒューリスティック手法の一種である [1]。ACO の処理手順は文献によって異なった表記によって説明されているが、ここでは [9, 10] での表記に従って説明する。

2.1 組合せ最適化問題のモデル化

まず、本研究で対象とする組合せ最適化問題のモデルを次のように定義する。この定義は [9, 10, 15] など で用いられているものと同様である。

定義 1. 組合せ最適化問題のモデル $\mathcal{P} = (S, \Omega, f)$ は次に示す要素で構成される。

- 離散的な決定変数の有限集合で定義される探索空間 S と変数間の制約の集合 Ω
- 最小化する目的関数 $f: S \rightarrow \mathbb{R}^+$

実行可能解 $s \in S$ は n 個の決定変数 $X_i, i = 1, \dots, n$ からなり、制約集合 Ω に含まれる全ての制約を満たす。決定変数 X_i のとり得る値 (ドメインと呼ぶ) は $D_i = \{v_i^1, \dots, v_i^{|D_i|}\}$ で与えられる。制約集合 Ω が空集合であれば、決定変数 X_i には他の変数の値に依存せず、ドメイン D_i の中から任意の値を割り当てられる。この場合、 \mathcal{P} を制約無しの問題モデルと呼び、そうでない場合は制約有りの問題モデルと呼ぶ。実行可能解 $s^* \in S$ が $f(s^*) \leq f(s), \forall s \in S$ である場合、これを大域的最適解と呼ぶ。大域的最適解の集合は $S^* \subseteq S$ で表される。

組合せ最適化問題のモデルに基づき、次のようにフェロモンのモデル化を行う。まず決定変数 X_i とその値 $v_i^j \in D_i$ の組合せを、解を構成するコンポーネントと呼び、 c_i^j と表す。ここで、コンポーネント c_i^j に対応するフェロモンパラメータを T_i^j と定義する。フェロモンパラメータ T_i^j の値をフェロモン量と呼び、 τ_i^j と表す。また、全てのフェロモンパラメータの集合を \mathcal{T} と表す。

† 金沢大学 自然科学研究科

‡ 石川工業高等専門学校 電子情報工学科

```

input: モデル  $\mathcal{P} = (\mathcal{S}, \Omega, f)$  の具体例  $P$ 
InitializePheromoneValues( $T$ )
 $s_{bs} \leftarrow \text{NULL}$ 
while 終了条件が満たされない do
   $\mathcal{G}_{iter} \leftarrow \emptyset$ 
  for  $j = 1, \dots, n_a$  do
     $s \leftarrow \text{ConstructSolution}(T)$ 
     $s \leftarrow \text{LocalSearch}(s)$  {optional}
    if ( $f(s) < f(s_{bs})$ ) or ( $s_{bs} = \text{NULL}$ ) then  $s_{bs} \leftarrow s$ 
     $\mathcal{G}_{iter} \leftarrow \mathcal{G}_{iter} \cup \{s\}$ 
  end for
  ApplyPheromoneUpdate( $T, \mathcal{G}_{iter}, s_{bs}$ )
end while
output:  $s_{bs}$ 

```

図 1: ACO アルゴリズム

2.2 ACO アルゴリズムの構成

基本的な ACO アルゴリズムの構成を図 1 に示す。各反復において、 n_a 個の解をフェロモンパラメータに基づく確率的な構築処理によって生成する。その後、生成された解を局所探索によって改善する。最後に、次の反復が開始する前に、生成されたいくつかの解を用いてフェロモン量の更新を行う。以下ではこれらの処理の詳細について説明する。

InitializePheromoneValues(T). アルゴリズムの開始時にフェロモン量を定数 $c > 0$ に設定する。

ConstructSolution(T). コンポーネントの有限集合 \mathcal{C} から順にコンポーネントを取り出し、部分解 s^P に追加していくことで解を構築する。まず、 $s^P = \langle \rangle$ とし、その後 s^P に実行可能なコンポーネントを追加していく。このとき追加されるコンポーネントは、解の構築手法によって定義される集合 $\mathfrak{N}(s^P) \subseteq \mathcal{C} \setminus s^P$ の要素である。この処理によって生成される解の集合を \mathcal{G} とする。

部分解 s^P に追加するコンポーネントの選択は確率的に行われる。多くの ACO アルゴリズムではコンポーネント c_i^j の選択確率は次式で定義している。

$$p(c_i^j | s^P) = \frac{[\tau_i^j]^\alpha \cdot [\eta(c_i^j)]^\beta}{\sum_{c_k^j \in \mathfrak{N}(s^P)} [\tau_k^j]^\alpha \cdot [\eta(c_k^j)]^\beta} \quad (1)$$

ここで、 η は適用する問題に関するヒューリスティック情報に基づいて、選択確率に重み付けを行う関数である。 $\alpha \geq 0$ と $\beta \geq 0$ はフェロモン情報とヒューリスティック情報の重要性を決めるパラメータである。

ApplyPheromoneUpdate($T, \mathcal{G}_{iter}, s_{bs}$). 生成されたいくつかの解を用いて、フェロモン量の更新を行う。多くの ACO アルゴリズムは次のようなフェロモン量更新ルールを用いている。右辺の第 1 項は既存のフェロモンの減少を表し、第 2 項は新たなフェロモンの増加を表す。

$$\tau_i^j \leftarrow (1 - \rho) \cdot \tau_i^j + \frac{\rho}{n_a} \cdot \sum_{\{s \in \mathcal{G}_{upd} | c_i^j \in s\}} F(s) \quad (2)$$

ただし、 $i = 1, \dots, n$, $j \in \{1, \dots, |D_i|\}$ である。 $\rho \in (0, 1]$ は蒸発率と呼ばれるパラメータである。 $F: \mathcal{G} \mapsto$

\mathbb{R}^+ は評価関数と呼ばれる関数であり、 $f(s) < f(s') \Rightarrow F(s) \geq F(s')$ とする。 \mathcal{G}_{upd} はフェロモン量の更新処理に用いる解の集合であり、 $\mathcal{G}_{iter} \cup \{s_{bs}\}$ の部分集合である。ただし、 \mathcal{G}_{iter} は 1 回の反復中に生成された解の集合であり、 s_{bs} はアルゴリズムの実行中に見つかった最良の解である。

2.3 フェロモン量更新処理に用いる解集合の設定法

フェロモン量の更新処理に用いる解の集合 \mathcal{G}_{upd} の設定法によって、様々な更新ルールが定義できる。例えば AS[2] で用いられている更新ルール (AS-更新ルールと呼ぶ) では

$$\mathcal{G}_{upd} \leftarrow \mathcal{G}_{iter} \quad (3)$$

とする。実用上よく用いられるルールは

$$\mathcal{G}_{upd} \leftarrow \operatorname{argmax}\{F(s) | s \in \mathcal{G}_{iter}\} \quad (4)$$

とするものである。これは、1 回の反復中に見つかった最良の解を \mathcal{G}_{upd} に設定することを意味する (IB-更新ルールと呼ぶ)。IB-更新ルールは AS-更新ルールに比べて良い解を見つけやすいが、初期収束に陥る危険性が高まる。

3 アントコロニー最適化における騙し

ACO においては探索を続けるごとに、1 回の反復中に生成される解の平均評価値が上昇することが望まれる。ACO がそのように振る舞うか調べるために、[9, 10, 15] では以下に説明する ACO のモデルを用いて解析を行っている。このモデルでは蟻の数 n_a を無限大と仮定することでモデルの簡略化を行っている。なお、より実際のモデルは [16] などに示されている。

モデルの振る舞いは反復ごとに生成される解の期待評価値の変化によって明らかにする。以下ではこの期待評価値を $W_F(T)$, または $W_F(T|t)$ と表す。ただし、 $t > 0$ は反復回数である。期待評価値 $W_F(T)$ は次式で与えられる。

$$W_F(T) = \sum_{s \in \mathcal{G}} F(s) \cdot p(s|T) \quad (5)$$

ただし、 $p(s|T)$ は、解 s が発生する確率である。

さらに、 $n_a = \infty$ と仮定することで、AS-更新ルールは以下ようになる。

$$\tau_i^j(t+1) \leftarrow (1 - \rho) \cdot \tau_i^j(t) + \rho \cdot \sum_{\{s \in \mathcal{G} | c_i^j \in s\}} F(s) \cdot p(s|T) \quad (6)$$

ただし、 $i = 1, \dots, n$, $j = 1, \dots, |D_i|$ であり、 t は反復回数である。このようにして得られる更新ルールを期待更新ルールと呼ぶ。

ACOにおける1次, および2次騙しは[10]で定義されている。1次騙しは進化的計算における騙し構造による局所最適解への収束に対応し, 2次騙しは解の表現や構築方法に依存した探索バイアスに対応する。それぞれの定義は, 次に示す局所最適化手法 (*local optimizer*) の定義に基づいている。

定義 2. 組合せ最適化問題のモデル P が与えられ, P の具体例 P に ACO のモデルを適用したとする。このとき任意の初期フェロモン量の設定に対して, 期待更新ルールによるフェロモン量の更新が $W_F(T|t+1) \geq W_F(T|t), \forall t \geq 0$ となるように行われる場合, この ACO のモデルを局所最適化手法 (*local optimizer*) と呼ぶ。

なお, 制約無しの問題モデルに ACO を適用する場合, 局所最適化手法となることが証明されている [10, 15]。この定義に基づいて, 1次騙し系が定義される。

定義 3. 組合せ最適化問題のモデル P が与えられ, P の具体例 P に局所最適化手法を適用したとき, 大域的最適解に収束しないことが見込まれるような初期フェロモン量の設定が存在する場合, この局所最適化手法を1次騙し系 (*First order deceptive system: FODS*) と呼ぶ。

これは ACO が局所最適化手法であったとしても, 複数の不動点を持つ問題に適用された場合は1次騙し系となることを意味する。

次に, 2次騙し系は次のように定義される。

定義 4. 組合せ最適化問題のモデル P が与えられ, P の具体例 P に ACO のモデルを適用したとする。このとき, 期待評価値の変化の中に, $W_F(T|t+1) < W_F(T|t), \forall t \in \{i, \dots, i+l-1\}$ となるような時間窓 $[i, i+l]$ ($i > 0, l > 0$) がある場合, このモデルを2次騙し系 (*Second order deceptive system: SODS*) 呼ぶ。これは, もし P に対する ACO のモデルが局所最適化手法とならなければ, その2つの組合せが *SODS* であることを意味する。

2次騙しの例は [11, 12, 13, 16] などに示されている。[16] では, 置換問題に対して簡略化した ACO のモデルを適用すると2次騙しが発生することが示されている。[11] では, k -部分木問題において2次騙しが発生することが示されている。しかしながら, この場合は AS-更新ルールを用い, かつ局所探索を行わない場合のみこの現象が見て取れる。[12, 13] ではジョブショップスケジューリング問題での例が示されている。この問題においては, IB-更新ルールを用い, 局所探索を行ったとしても2次騙しが発生する。

4 ジョブショップスケジューリング問題

4.1 問題定義

本研究では2次騙しが発生することが知られているジョブショップスケジューリング問題 (JSS) を用いて

アルゴリズムの性能評価を行う。この問題では, 工程 $O = \{o_1, \dots, o_n\}$ が与えられ, この工程は部分集合の集合 $M = \{M_1, \dots, M_{|M|}\}$ に分割される。 $M_i \in M$ 内の工程は同じ機械で処理される。 O は別の集合 $J = \{J_1, \dots, J_{|J|}\}$ にも分割され, 工程の集合 $J_j \in J$ をジョブと呼ぶ。さらに, 処理時間 $p(o) \in \mathbb{N}$ が各工程 $o \in O$ に対して与えられる。また, 各ジョブ $J_j \in J$ に対して順列 π^{J_j} が与えられる。この順列は同一ジョブ内の工程の処理順序 (技術的順序と呼ばれる) を決定する。以下では順列 π が与えられたとき, $\pi(l)$ を π の l 番目の位置にある工程とする。2つの工程 $o_i = \pi(l)$ と $o_j = \pi(k)$ について $l < k$ であるとき, $o_i \preceq o_j$ と書く。工程 $o_i \in O$ の先行工程は次式で与えられる。

$$\text{pred}(o_i) \leftarrow \{o_j \in O | o_j \preceq o_i\} \quad (7)$$

各機械は同時に複数の工程を処理することは出来ないものとし, また, 工程の割り込みは発生しないものとする。さらに, 同じジョブに属する工程は技術的順序を守って処理しなければならない。

解 $s = (\pi^{M_1}, \dots, \pi^{M_{|M|}})$ は $M_i, \forall i \in \{1, \dots, |M|\}$ 内の工程の順列 π^{M_i} によって与えられる。これらの順列は各機械 M_i における工程の処理順序を決定する。なお, 特定の順列においては処理順序の循環が生じるため, 実行可能解にならない順列も存在する。

解のコストを評価するにはいくつかの方法があるが, 本研究ではメイクスパンの最小化を目的とする。解 s に対して, 全ての工程 $o \in O$ に最早開始時刻 $t_{es}(o, s)$ が定義される。全てのジョブは時刻0から開始できるとする。解 s に対して, 工程 $o \in O$ の最早完了時刻 $t_{ec}(o, s)$ は $t_{es}(o, s) + p(o)$ と定義される。このとき, 解 s の目的関数値は次式で定義される。

$$f(s) \leftarrow \max \{t_{ec}(o, s) | o \in O\} \quad (8)$$

4.2 モデル化

全ての工程の順列を考えると, そのうちいくつかは JSS の解を表現する。これは, そのような順列には機械ごとの順列も含まれているからである*。Colomni らはこの表現に基づき, 以下のようなモデル P_{JSS}^{JSS} を提案している [17]。まず, 工程の集合 O に処理時間が0のダミー工程 o_0, o_{n+1} を加える。工程 o_0 は全てのジョブの起点となり, 工程 o_{n+1} は終点となる。よって, 工程の集合は $O = \{o_0, o_1, \dots, o_n, o_{n+1}\}$ となる。そして, 各工程 o_i に対して決定変数 X_i を定義する。ただし, $i \in \{0, \dots, n\}$ である。決定変数 X_0 のドメインは $D_0 = \{1, \dots, n\}$ であり, 他の決定変数 X_i のドメインは $D_i = \{1, \dots, n+1\} \setminus i$ である。決定変数 X_i に値 $j \in D_i$ を割り当てることは, 全工程の順列において o_i のすぐ後に o_j が位置することを意味する。また, 決定変数 X_i とその値 $j \in D_i$ の組合せをコンポーネント c_i^j とする。よってフェロモン・モデル T は各コンポーネ

*ただし, 一般的には実行可能な順列から解への写像は多対1の関係にある。

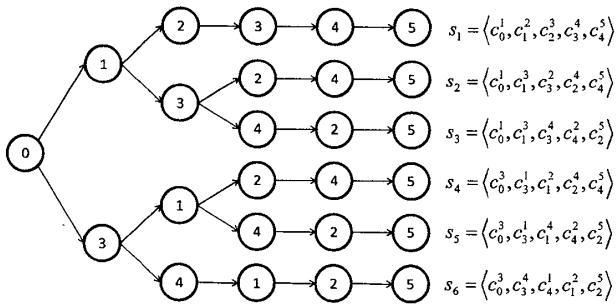


図 2: jss_simple_inst の探索木

ント c_i^j に対応するフェロモンパラメータ T_i^j で構成される。

スケジュールの構築にはリストスケジューラアルゴリズムを用いる。このアルゴリズムでは、割り当て候補となる工程の集合 O_t から1つの工程を取り出し、その工程を割り当てる処理を n 回繰り返すことによって、全工程の順列を左から右に向かって構築する。 O は次のように定義される。

$$O_t \leftarrow \{o \in O^+ | \text{pred}(o) \cap O^+ = \emptyset\} \quad (9)$$

ここで、 O^+ は割り当てられていない工程の集合である。

ACO の解の構築は以下のように行う。 i_c を直前に割り当てた工程のインデックス番号とする。まず、部分解 $s^P = \langle \rangle$ とし、 $i_c = 0$ とする。その後、コンポーネント $c_{i_c}^j \in \mathfrak{N}(s^P)$ を s^P へ追加していく。ただし、

$$\mathfrak{N}(s^P) = \{c_{i_c}^k | o_k \in O_t\} \quad (10)$$

とする。 s^P にコンポーネント $c_{i_c}^j$ を追加したときは、 $i_c \leftarrow j$ として引き続き処理を行う。追加するコンポーネントは次式の確率分布に従って決定する。

$$p(c_{i_c}^j | T) = \frac{[T_{i_c}^j]^\alpha}{\sum_{c_{i_c}^k \in \mathfrak{N}(s^P)} [T_{i_c}^k]^\alpha}, \forall c_{i_c}^j \in \mathfrak{N}(s^P) \quad (11)$$

これは式 (1) の分布と同じであるが、ヒューリスティック情報は用いていない。

P_{JSS}^{SS} モデルではIB-更新ルールを用いたとしても、2次騙しの効果による評価値の減少を防ぐことが出来ない [9, 10]。このため、別のモデル P_{JSS}^{POS} [18], P_{JSS}^{REL} [9] が提案されており、このモデルを用いることで2次騙しの効果を抑制することが出来る。しかしながら、本研究の目的はフェロモン更新ルールを改良することによって、モデルの変更をせずに2次騙しの効果を防ぐことであるため、これらのモデルは用いず P_{JSS}^{SS} モデルを用いて性能の評価を行う。

4.3 2次騙しの例

[9] では以下の小規模な問題 (jss_simple_inst) における2次騙しの発生例を示している。

$$\begin{aligned} O &= \{o_1, o_2, o_3, o_4\} \\ \mathcal{J} &= \{\mathcal{J}_1 = \{o_1, o_2\}, \mathcal{J}_2 = \{o_3, o_4\}\} \\ \mathcal{M} &= \{\mathcal{M}_1 = \{o_1, o_4\}, \mathcal{M}_2 = \{o_2, o_3\}\} \\ p(o_1) &= p(o_4) = 10 \\ p(o_2) &= p(o_3) = 20 \end{aligned}$$

この問題において、前述のリストスケジューラアルゴリズムを用いて解を構築する際の探索木を図2に示す。生成される解は s_1, \dots, s_6 の6つである。目的関数値は、 $i = 1, 6$ のとき $f(s_i) = 60$ であり、 $i = 2, 3, 4, 5$ のとき $f(s_i) = 40$ である。評価値は $F(s_i) = 1/f(s_i)$ として与える。

この問題で2次騙し効果が発生する仕組みは以下のように説明できる。部分解が $s^P = \langle c_0^1 \rangle$ のとき (図2において $0 \rightarrow 1$ と辿ったとき) を考えると、次に加えられるコンポーネントの候補は c_1^2 と c_1^3 である。ここで、 c_1^2 は3つの解 s_1, s_4, s_6 に含まれるのに対し、 c_1^3 は s_2, s_3 の2つの解にしか含まれない。よって、フェロモン量 T_1^2 には1つの最適解 (評価値 $1/40$) と2つの準最適解 (評価値 $1/60$) から新たなフェロモンが加えられるのに対し、 T_1^3 には2つの最適解からしか新たなフェロモンが加えられない。つまり、 T_1^2 には T_1^3 よりも多くのフェロモンが加えられやすいと言える。また、対称的な構造により、 T_3^4 と T_3^1 にも同様のことが言える。従って、反復を繰り返すごとに、準最適解 s_1, s_6 の生成確率が上昇し、最適解 s_2, s_3, s_4, s_5 の生成確率が減少していく。

実際にこの問題に対して、AS-更新ルールを用いるACOを適用した結果を図3に示す。図3(a)には、3節で示した簡略化モデルにおける理論的な期待更新ルールを用いたときの期待評価値の推移を、図3(b)には通常のAS-更新ルールを適用したときの平均評価値の推移を示す。これは $n_a = 10$ とし、100回の試行を行ったときの平均であり、エラーバーは標準偏差を表す。反復を繰り返すごとに、評価値が減少していることがわかる。なお、通常の更新ルールを用い、 $\rho = 0.05$ としたときに標準偏差が増加しているのは、高い学習率によって良い解にも悪い解にも収束し得るためである。

5 提案手法

5.1 概要

探索バイアスは適切なフェロモンのモデル化を行うことで抑制することができる [9, 14, 18]。しかしながら適切なモデルの設計には経験的な知識が必要であり、容易にそのようなモデルが得られない場合もあると考えられる。そこで、適切でないモデルを用いたとしてもある程度の性能が得られるように、フェロモン量更

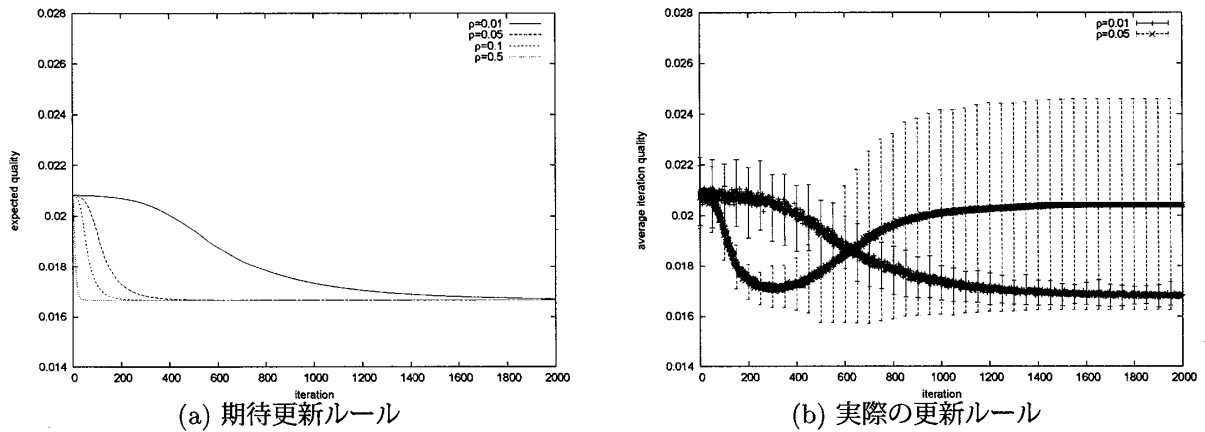


図 3: AS-更新ルールを用いたときの評価値の推移. 同様の結果が [9] に示されている.

新ルールの改良によりバイアスを抑制することを検討する.

4.3節で示したように, ACO ではより多くの実行可能解に含まれるコンポーネントほどフェロモン量が増加しやすい. よって, 多くの解に含まれるコンポーネントのフェロモン増加量を減らし, 少ない解にしか含まれないコンポーネントのフェロモン増加量を増やすことで, このバイアスを抑制できると考えられる. そこで, 以下のフェロモン更新ルールを提案する.

$$\tau_i^j \leftarrow \begin{cases} (1 - \rho)\tau_i^j + \frac{\rho}{|\mathcal{G}_i^j|} \sum_{s \in \mathcal{G}_i^j} F(s) & |\mathcal{G}_i^j| > 0 \\ \tau_i^j & \text{otherwise} \end{cases} \quad (12)$$

ただし, $\mathcal{G}_i^j = \{s \in \mathcal{G}_{upd} | c_i^j \in s\}$ とする.

この更新ルールでは, 1回の反復でコンポーネント c_i^j を含む解が複数生成された場合は, その評価値の平均をフェロモンの増加量とする. 多くの解に含まれるコンポーネントには, 1回の反復で多くの解からフェロモン量の増加が行われるため, その偏りを抑制するために平均をとる. また, コンポーネント c_i^j が含まれる解が生成された場合にフェロモン量の更新を行い, 1つも生成されない場合は更新を行わない. 少ない解にしか含まれないコンポーネントにはフェロモン量の増加が起りにくい, 通常ルールでは蒸発は毎回の反復で行われるため, 急速にフェロモン量が減少してしまう. これを防ぐために, フェロモンの蒸発は分泌が発生するときのみ行う.

なお, フェロモン量の更新処理に用いる解の集合 \mathcal{G}_{upd} の設定法によって, 次の二つの更新ルールを定義する. 式 (3) の設定法を用いるルールを AS-proposal ルールと呼び, 式 (4) の設定法を用いるルールを IB-proposal ルールと呼ぶ.

5.2 提案ルールの性質

3節と同様に, $n_a = \infty$ と仮定したときに期待されるフェロモン量更新について考える. $n_a = \infty$ とすると,

大数の法則により AS-proposal ルールは以下のようになる.

$$\tau_i^j(t+1) \leftarrow (1 - \rho)\tau_i^j(t) + \rho \sum_{s \in \mathcal{G}_i^j} F(s) \cdot \mathbf{p}(s|T, X_i = v_i^j) \quad (13)$$

ただし, $\mathcal{G}_i^j = \{s \in \mathcal{G} | c_i^j \in s\}$ となる. $\mathbf{p}(s|T, X_i = v_i^j)$ は $X_i = v_i^j$ と固定したときに解 s が生成される確率である. ここで,

$$W_F(T|X_i = v_i^j) := \sum_{s \in \mathcal{G}_i^j} F(s) \cdot \mathbf{p}(s|T, X_i = v_i^j) \quad (14)$$

は $X_i = v_i^j$ としたときに生成される解の期待評価値である. よって提案する更新ルールでは, 期待評価値の高いコンポーネントほどフェロモンの増加量が多くなる.

フェロモン量の上限值, 下限値は以下のように得られる. 提案する更新ルールにおいてフェロモン量 τ_i^j の増加量が最大となるのは, c_i^j を含む解の中で, 評価値が最大となる解のみが \mathcal{G}_i^j に含まれる場合である. この場合の評価値を $F_i^{j,max} := \max_{s \in \mathcal{G} | c_i^j \in s} \{F(s)\}$ とすると, 提案する更新ルールは

$$\tau_i^j(t+1) \leftarrow (1 - \rho)\tau_i^j(t) + \rho F_i^{j,max} \quad (15)$$

となる. この一般項は,

$$\tau_i^j(t) = (1 - \rho)^t \cdot c + \rho \sum_{k=0}^t (1 - \rho)^k F_i^{j,max} \quad (16)$$

となり, フェロモン量の極限值は,

$$\lim_{t \rightarrow \infty} \tau_i^j(t) = F_i^{j,max} \quad (17)$$

となる. よってフェロモン量の上限值は $\max\{F_i^{j,max}, c\}$ となる. 下限値は上限値の場合と同様に, c_i^j を含む解

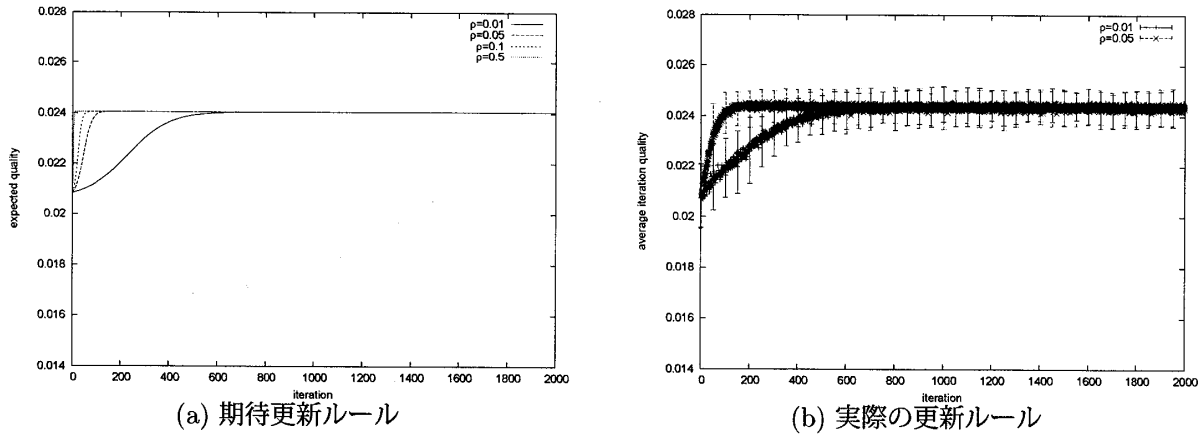


図 4: AS-proposal ルールを用いたときの評価値の推移.

の中で、評価値が最小となる解のみが \mathcal{G}_i^j に含まれる場合を考えればよい. この場合の評価値を $F_i^{j \min} := \min_{s \in \mathcal{G}_i^j \cap \mathcal{E}_s} \{F(s)\}$ とすると、このときのフェロモン量の極限值は

$$\lim_{t \rightarrow \infty} \tau_i^j(t) = F_i^{j \min} \quad (18)$$

であり、下限値は $\min\{F_i^{j \min}, c\}$ として得られる. 以上より、提案するフェロモン量更新ルールでは、暗黙的にフェロモン量の上下限值が設定されることになる. このような上下限値は ACS[3] においては暗黙的に、MMAS[5] においては明示的に設定されている. しかしながら、MMAS においては全てのコンポーネントにおいて同じ上下限値が設定され、ACS では上限値はコンポーネントごとに異なる値が設定されるが、下限値は同じ値が設定される. これに対して、提案手法では上限値、下限値ともにコンポーネントごとに異なる値が設定されることになる.

5.3 従来手法との比較

4.3 節と同様に、jss_simple_inst に対して AS-proposal ルールを用いる ACO を適用した結果を図 4 に示す. 図 4 (a) には、簡略化モデルにおける理論的な期待更新ルールを用いたときの期待評価値の推移を、図 4 (b) には通常の更新ルールを適用したときの平均評価値の推移を示す. なお、ACO のパラメータは $\alpha = 10.0$, $c = 0.5$ とした. 提案ルールを用いる場合、反復を繰り返すごとに評価値が上昇していることがわかる.

6 性能評価実験

[9] と同様に、JSS のベンチマーク問題である ft10[19] と orb08[20] を用いて性能評価を行う. ここでは平均評価値の推移とともに、以下の式で定義される sequencing

factor の推移も調べる.

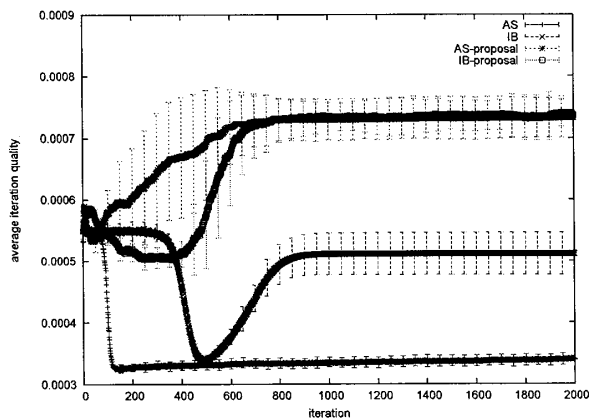
$$f_{seq}(\pi) \leftarrow \frac{\sum_{i=1}^{n-1} \delta(\pi, i, i+1)}{\sum_{J_j \in \mathcal{J}} (J_j - 1)} \quad (19)$$

ここで、 $\delta(\pi, i, i+1)$ は順列 π の i 番目にある工程と、 $i+1$ 番目にある工程が同じジョブに属する場合に 1 となり、そうでなければ 0 となる関数である. P_{JSS}^{seq} モデルを用いる場合、探索バイアスによって f_{seq} の高い解が生成されやすいことがわかっている [9].

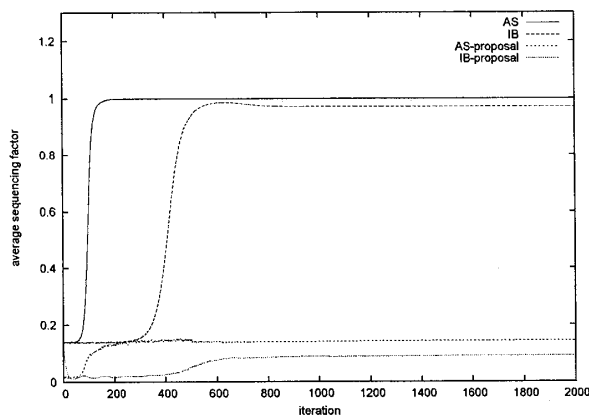
各更新ルールを適用した結果を図 5 に示す. なお、ACO のパラメータは AS-更新ルールでは蒸発率 $\rho = 0.1$, 初期フェロモン量 $c = 0.5$ とし、IB-更新ルールでは $\rho = 0.03$, $c = 0.5$ とした. これらは [9] と同様の結果が得られるように設定した. 提案するルールを適用する場合、AS-proposal ルールでは $\alpha = 80$, $\rho = 0.3$, $c = 0.001$ とした. IB-proposal ルールでは $\rho = 0.4$ とし、他のパラメータは AS-proposal ルールと同様の値を用いた. これらは事前実験により良好な結果を示した値である.

提案する更新ルールを用いることで、平均評価値が通常の更新ルールに比べ向上している. しかしながら、探索の初期段階では性能の低下が見られる. 提案ルールでは、あるコンポーネントを含む解が生成されない場合、そのコンポーネントのフェロモン量は変化しない. また、前述のパラメータ設定では初期フェロモン量に比べてフェロモン増加量は小さい値となる. よって、探索の初期では更新処理によりフェロモン量が減少し、対応するコンポーネントが選ばれにくくなる. P_{JSS}^{seq} モデルを用いると f_{seq} の高い解が生成されやすいため、この段階では f_{seq} が急激に減少することになる. その後、他のコンポーネントにも更新処理が行われ、全体のフェロモン量が均一化するため、 f_{seq} も上昇する. 有効な探索はこの初期段階の後から始まり、平均評価値が上昇していく.

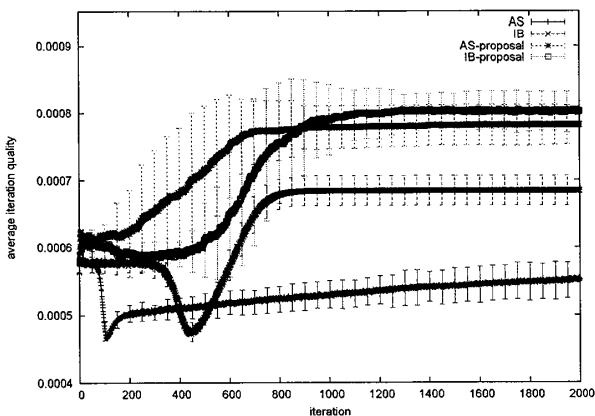
次に、[9] で提案されている問題である patho_1 における性能評価を行う. この問題は 10 ジョブ、10 機械か



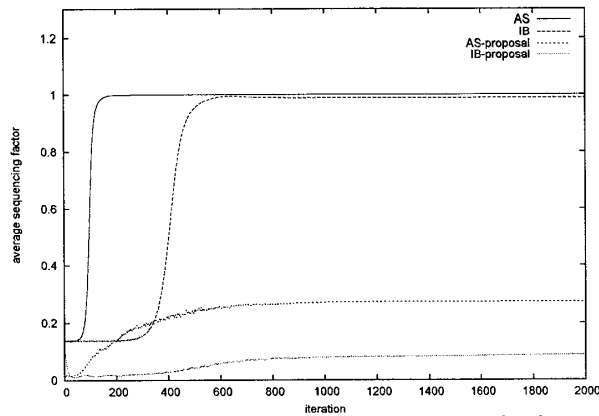
(a) ft10 における平均評価値の推移



(b) ft10 における sequencing factor の推移



(c) orb08 における平均評価値の推移



(d) orb08 における sequencing factor の推移

図 5: ベンチマーク問題における各手法の性能

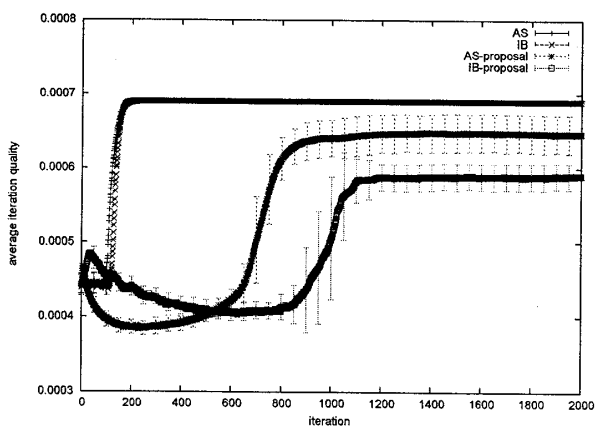
らなり, 1つのジョブには 10 工程が含まれる. ジョブ内の工程は第 1 工程は機械 M_1 , 第 2 工程は機械 M_2 というように, 各機械で順に処理する. 機械 M_j で処理する工程 o の処理時間は $p(o) = 110 - (10 \times j)$ とする. この問題では f_{seq} が大きい解ほど評価値も高くなるため, 通常の AS-更新ルール, IB-更新ルールを用いれば最適解を簡単に求めることが出来る. なお, 最適解は f_{seq} が 1 (最大値) となる解である. しかしながら, このような問題の特性は JSS の一般的な問題とは異なる [9].

各更新ルールを適用した結果を図 6 に示す. ACO のパラメータは AS-更新ルール, IB-更新ルールでは $\rho = 0.1$, $c = 0.5$ とした. 提案するルールを適用する場合は, AS-proposal ルールでは $\alpha = 80$, $\rho = 0.05$, $c = 0.001$ とした. IB-proposal ルールでは $\rho = 0.3$ とし, 他のパラメータは AS-proposal ルールと同様に設定した. 提案するルールを適用した場合, 従来手法に比べ平均評価値は減少しているが, 最終的には比較的良好な解に収束している. 図 5 の結果と合わせると, 提案するルールを適用することで問題の特性に対する頑健性が向上すると考えられる.

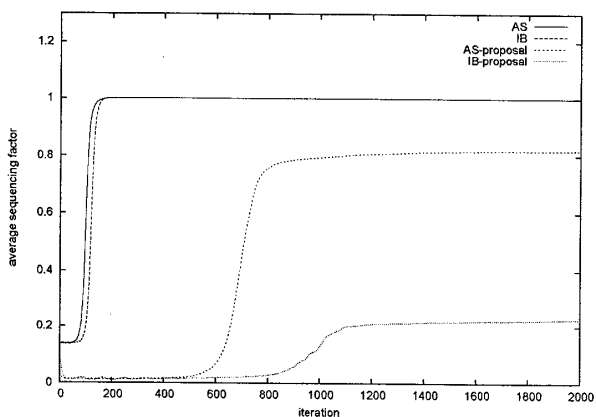
7 むすび

本研究では 2 次騙し効果を軽減するための新しいフェロモン量更新ルールを提案した. 提案するルールでは, 簡略化したモデルを `jss_simple_inst` 問題に適用した場合に期待評価値が上昇していくことを示し, また代表的なベンチマーク問題である ft10, orb08 を用いた性能評価実験の結果, 提案する更新ルールを用いることで通常の更新ルールに比べ良好な性能が得られることがわかった.

今後の課題としては, まず ACO のパラメータの設定が挙げられる. 特に α パラメータについては, 一般的には 1.0 とすることが良いとされているが, 提案手法ではそれに比べて非常に大きい値を設定しなければならないなど, 従来手法とは設定法が異なる場合もある. より広範な実験によって適切な設定法を確立することとともに, パラメータに依存せず良好な性能が得られるような改良法を検討する必要がある. また, ヒューリスティック情報を表す η 関数や, 局所探索を適用した場合の性能を調べる必要がある. これらの処理は ACO の探索に人為的な偏りを与えるため, フェロモン情報



(a) 平均評価値の推移



(b) sequencing factor の推移

図 6: patho_1 における各手法の性能

のみを用いる場合とは異なると考えられる。

さらに、探索バイアスの抑制による性能低下についても改良の余地がある。4.3 節の例では適切でないコンポーネントが多く解に含まれていたため、提案する更新ルールによりバイアスを抑制することで性能が向上した。反対に、多くの解に含まれるコンポーネントが適切なものであるような問題では、探索バイアスを抑制すると適切なコンポーネントのフェロモン量が増加しにくくなるため、性能が低下すると考えられる。また、6 節で示した patho_1 問題の例では f_{seq} の大きい解ほど評価値が高くなるため、探索バイアスが好ましい影響を与える。よって、バイアスを抑制する提案ルールは従来のルールに比べ性能が低下している。

参考文献

[1] M. Dorigo and T. Stutzle, "Ant Colony Optimization," MIT Press, 2004.
 [2] M. Dorigo, V. Maniezzo, and A. Colorni, "The Ant System: Optimization by a colony of cooperating agents," IEEE Trans. Syst. Man Cybern.B, vol.26, no.1, pp.29-41, 1996.

[3] M. Dorigo and L. M. Gambardella, "Ant Colonies for the Traveling Salesman Problem," BioSystems, vol.43, pp.73-81, 1997.
 [4] B. Bullnheimer, R. F. Hartl, and C. Strauss, "A new rank-based version of the ant system," a computational study, Central European Journal for Operations Research and Economics, vol.7, No.1, pp.25-38, 1999.
 [5] T. Stutzle and H. H. Hoos, "MAX-MIN Ant System," Future Generation Computer Systems, vol.16, no.8, pp.889-914, 2000.
 [6] B. Bullnheimer, R. F. Hartl, and C. Strauss, "An improved ant system algorithm for the vehicle routing problem," Annals of Operations Research, vol.89, pp.319-328, 1999.
 [7] E. Bonabeau, M. Dorigo, and G. Theraulaz, "Inspiration for Optimization from Social Insect Behavior," Nature, vol.406, pp.39-42, 2000.
 [8] M. Dorigo, G. Di Caro, L.M. Gambardella, "Ant Algorithms for Discrete Optimization," Artif. Life, vol.5, no.2, pp.137-172, 1999.
 [9] C. Blum and M. Dorigo, "Search Bias in Ant Colony Optimization: On the Role of Competition-Balanced Systems," IEEE Trans. Evolutionary Computation, vol.9, no.2, pp.159-174, 2005.
 [10] C. Blum and M. Dorigo, "Deception in Ant Colony Optimization," Proc. of the ANTS 2004, Lecture Notes in Computer Science, vol.3172, pp.118-129.
 [11] C. Blum, M. Sampels, and M. Zlochin, "On a particularity in model-based search," Proc. of the Genetic and Evolutionary Computation Conf. (GECCO-2002), pp.35-42.
 [12] C. Blum and M. Sampels, "Ant Colony Optimization for FOP shop scheduling: A case study on different pheromone representations," Proc. of the 2002 Congress on Evolutionary Computation, vol.2, pp.1558-1563.
 [13] C. Blum and M. Sampels, "When model bias is stronger than selection pressure," Proc. of the Seventh Int. Conf. on Parallel Problem Solving from Nature, Lecture Notes in Computer Science, vol.2439, pp.893-902, 2002.
 [14] J. Montgomery, M. Randall and T. Hendtlass, "Solution bias in ant colony optimisation: Lessons for selecting pheromone models," Computers & Operations Research, vol.35, pp.2728-2749, 2008.
 [15] C. Blum and M. Dorigo, "The hyper-cube framework for ant colony optimization," IEEE Trans. Syst. Man Cybern.B, vol.34, no.2, pp.1161-1172, 2004.
 [16] D. Merkle and M. Middendorf, "Modelling the dynamics of ant colony optimization algorithms," Evolutionary Computation, vol.10, no.3, pp.235-262, 2002.
 [17] A. Colorni, M. Dorigo, V. Maniezzo, and M. Trubian, "Ant system for job-shop scheduling," Belgian J. Oper. Res. Statist. Comput. Sci. (JORBEL), vol.34, no.1, pp.39-53, 1994.
 [18] D. Merkle and M. Middendorf, "An ant algorithm with a new pheromone evaluation rule for total tardiness problems," Proc. of the EvoWorkshops 2000, Lecture Notes in Computer Science, vol.1803, pp.287-296.
 [19] J. F. Muth and G. L. Thompson, "Industrial Scheduling," Englewood Cliffs, NJ: Prentice-Hall, 1963.
 [20] D. Applegate and W. Cook, "A computational study of the job-shop scheduling problem," ORSA J. Comput., vol.3, pp.149-156, 1991.