

蟻の群行動を模擬した最適化手法 ACO の高速化

渡邊 勇 (isamu@criepi.denken.or.jp)

(財) 電力中央研究所 情報研究所

〒 201-8511 東京都狛江市岩戸北 2-11-1

Phone: (03) 3480-2111, Fax: (03) 5497-0318

概要

近年、組合せ最適化問題に対するメタ解法の1つとして、蟻の採餌行動を模擬した最適化手法 ACO (Ant Colony Optimization) が注目されている。しかし、大規模な問題に対しては計算時間が長くなるなどの問題があり、実用時間での適用には課題が残されている。本発表では、ACO アルゴリズムにフェロモン情報に基づく候補集合の概念を導入し、解の探索過程で動的に変化する候補集合の計算を効率化することで高速化を図る方法について報告する。代表的な組合せ最適化問題であるグラフ彩色問題に本解法を適用した結果、大規模な問題 (節点数 1500) に対し既存の ACO アルゴリズムと同精度の解を最短で約 30% の計算時間で求めることができる。

1 はじめに

企業活動における計画、設計、運用に関わる問題の多くは組合せ最適化問題 (Combinatorial Optimization) として定式化することができる。組合せ最適化問題の多くは、現実的な計算時間で厳密解を求めることが極めて困難であることが知られている。しかし、最適性の保証は無くても、十分精度の高い解が求めれば実用上問題がない場合が多い。このような目的に用いられる発見的解法の1つとして、蟻が採餌活動を行なう際に観察される組織行動を模した最適化手法 ACO (Ant Colony Optimization) が提案されている [1]。これまでに ACO は多くの組合せ最適化問題に適用され、既存の発見的解法と同精度かそれ以上の最良解を獲得している [2, 3]。しかし、大規模な問題では考慮すべき状態数が多くなるため、次候補を選択する際の計算量が大きくなるなどの課題が残されている。本発表では、代表的な組合せ最適化問題であるグラフ彩色問題に対する ACO アルゴリズム [4] に候補集合の概念を導入し、解の探索過程で動的に変化する候補集合の計算を効率化することで高速化を図る方法について報告する。本解法を用いることで、大規模なグラフ彩色問題に対し既存の ACO アルゴリズムと同精度の解をより短時間で得ることができる。

2 協調型最適化手法 ACO

2.1 蟻の群行動のアナロジー

蟻は採餌活動の際に、他の個体との情報交換の手段としてフェロモン (pheromone) と呼ばれる芳香性の物質を地表に分泌することが観察されている。各個体は、餌を探索する際や巣に持ち帰る際に、他の個体が過去に分泌したフェロモンの分布にしたがって確率的な経路選択を行ない、その意思決定の結果を再びフェロモンとして地表に分泌する。その結果、多くの蟻が選択した経路には多くのフェロモンが分泌され、さらに多くの蟻をその経路に誘引するという正のフィードバック効果がもたらされる。

この蟻の群行動を組合せ最適化問題の解の探索に応用した最適化手法 ACO が提案されている [1]。すべての決定変数の値が未確定な状態 (初期状態) にある蟻エージェントは、隣接する部分解の中から確率的に1つを選択し、その部分解へと移動する (図 1 参照)。これは、一部の決定変数の値を確定することを意味している。エージェントは隣接する部分解への移動を繰り返し、最終的に実行可能解を獲得する。この時点で

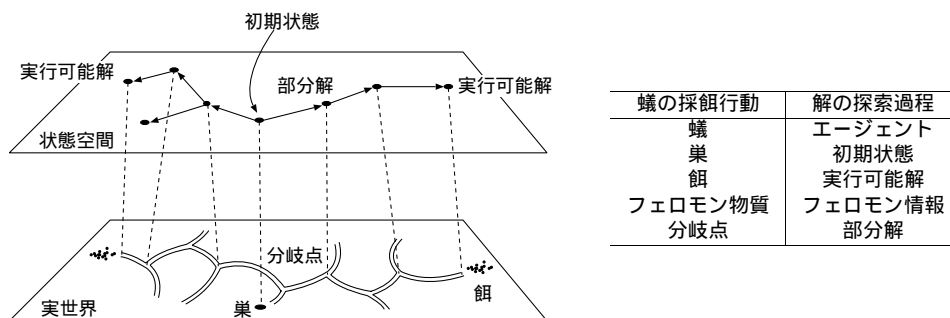


図 1: 蟻の採餌行動と ACO における解の探索過程との対応関係

エージェントは獲得した実行可能解を評価し、自身が辿ってきた経路上にその評価値に相当するフェロモンを残す。実行可能解を獲得したエージェントは最良解を記録し、再び初期状態から解の探索を開始する。

経路上に分泌されるフェロモンの量は、獲得した実行可能解の評価値やその解を獲得したエージェントの数や頻度によって異なる。したがって、評価値が最適値により近い実行可能解を獲得したエージェントはより多くのフェロモンを分泌し、また多くのエージェントによって選択された経路にはより多くのフェロモンが蓄積される。各エージェントは、これら蓄積されたフェロモンの分布にしたがって確率的に経路の選択を行なうため、蟻の採餌行動における正のフィードバック効果の発現と同様、過去のエージェント群の行動によって好ましいと判断された経路に多くのエージェントが集まり、良い評価値を示す実行可能解の近傍を集中的に探索することで、最適解を獲得する可能性を高めている。

2.2 基本アルゴリズム

ACOの基本的なアルゴリズムを図 2 に示す。ACOでは、解の探索を行なうエージェントに対し定義されるのは、フェロモン情報の更新、確率的な経路選択、内部状態の更新(行動の短期的な記憶)のみである。また、これらの基本機能を拡張した様々な ACO アルゴリズムがこれまでも提案されている [2, 3]。以下では、エージェントに定義される各基本機能について概説する。

(a) フェロモン情報の更新

状態 i から状態 j への経路 (i, j) 上に蓄積されたフェロモンの時点 t での総量 $\tau_{ij}(t)$ が、時点 $(t+1)$ までの間にその $\rho \times 100\%$ が蒸発し、エージェント k ($k = 1, \dots, m$) の分泌によって新たに $\Delta\tau_{ij}^k(t)$ だけ追加される。これより、時点 $(t+1)$ でのフェロモン量 $\tau_{ij}(t+1)$ は次式のように計算できる。

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

フェロモンを時間経過にしたがって蒸発させる効果としては、最近の行動に対しより大きな重要度を与えることによって、過去の行動に過度に縛られずに新たな領域を探索できる点が挙げられる。この機構により、探索早期に局所最適解へ陥ることを回避でき、より広い範囲の解の探索が可能になる。

(b) 確率的な経路選択

ACOでは、エージェントが意志決定を行なう時点での各経路の好ましさを定量化し、その総和に占める割合として選択確率を定義する。その際、フェロモン量の他に問題から得られる先見的な情報 η_{ij} も考慮する。この先見的な情報はフェロモン量とは独立に定義されるものであり、解の探索過程において不変な情報である。時点 t で状態 i にあるエージェント k が、推移可能な状態集合 J_i^k の中から状態 $j \in J_i^k$ を選択する確率を次式で与える。ただし、 α, β は $\tau_{ij}(t)$ と η_{ij} の相対的な重要度を表すパラメータである。

1. 初期化：エージェントの配置と初期フェロモンの付与
 2. ループ (終了条件を満たすまで)
 - 2.1 ループ (実行可能解を獲得するまで)
 - 2.1.1 ループ (すべてのエージェントについて)

確率的な経路選択 $\leftarrow p_{ij}^k(t) = [\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta / \sum_{\ell \in J_i^k} [\tau_{i\ell}(t)]^\alpha [\eta_{i\ell}]^\beta$

内部状態の更新
(フェロモン情報の更新)
 - 2.2 評価：解の評価と暫定解の記録
 - 2.3 更新：フェロモン情報の更新 $\leftarrow \tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t)$
3. 終了：最良解の出力

図 2: ACOの基本アルゴリズム

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{s \in J_i^k} [\tau_{is}(t)]^\alpha [\eta_{is}]^\beta} \quad (2)$$

先見的な情報 η_{ij} は、解の探索の初期段階において緩い方向づけを行なう。これによって初期段階での無駄な探索を省き、最適解の早期発見の可能性を高める効果がある。

(c) 行動の短期的な記憶

エージェントは、生成した解の実行可能性を保証するために、自身の過去の行動を記憶するための領域を保持している。この記憶領域はタブーリスト (tabu list) と呼ばれ、エージェントが推移可能な状態集合 J_i^k を求める際にこの記憶領域を参照する。つまり、解の探索途中で得られる部分解の実行可能性を常に検証することで、最終的に得られる解が実行可能解であることを保証する。

3 グラフ彩色問題への適用

本節では、代表的な組合せ最適化問題であるグラフ彩色問題に対する ACO アルゴリズム [4] の概要を示す。さらに、より大規模なグラフ彩色問題への適用を考え、アルゴリズムに候補集合の概念を導入し、解の探索過程で動的に変化する候補集合の計算を効率化することで高速化を図る方法について述べる。

3.1 問題の定義

有限個の節点からなる集合 V と節点对の集合 $E \subseteq V \times V$ からなる無向グラフ $G = (V, E)$ が与えられた時、グラフ G の彩色 (coloring) とは V を k 個の互いに素な部分集合に分割することをいう。ただし、隣接する2節点を異なる部分集合に割り当てることを制約条件とする。各部分集合に異なる色を対応させれば、これは節点对 $e \in E$ の両端は異なる色で塗りつつ、グラフ全体を k 色で塗り分けることに相当する。色数 k として採用可能な最小の値をグラフ G の彩色数 (chromatic number) といい、与えられたグラフ G の彩色数 $\chi(G)$ を求める、またはその色数で彩色する問題をグラフ彩色問題 (Graph Coloring Problem: GCP) という。GCP は代表的な組合せ最適化問題の1つであり、プログラム最適化におけるレジスタ割当て問題など工学的に有用な問題の多くはその特殊形として GCP を含んでいる。

3.2 ACO アルゴリズムの概要

ACO を実際の組合せ最適化問題 (GCP) に適用する際には、2 節で示した基本アルゴリズムで未定義のまま残されていた以下の3項目について明確にする必要がある。

3.2.1 推移可能な状態集合 J_i^k の決め方

推移可能な状態集合 J_i^k を決めることは、ACO アルゴリズムにおける実行可能解の構築方法を決めることを意味している。これまでに GCP に対し様々な構築法が提案されているが、ここでは解の精度の面で優れている再帰的 LRF 法 (RLF 法) を採用する。いま第 $(i-1)$ 色までの色分け $\{V_1, \dots, V_{i-1}\}$ が完了していると仮定する。さらに 2 種類の部分集合 \hat{V}, U を定義する。

$$\hat{V} = V - \bigcup_{j=1}^i V_j \quad (3)$$

$$U = \{u \in \hat{V} \mid (v, u) \in E, v \in V_i\} \quad (4)$$

\hat{V} は色の割当てが決まっていない節点集合、 U は第 i 色では塗られないことが既に分かっている節点集合を意味する。節点 $v \in \hat{V} - U$ に対し $E_v = \{(v, u) \in E \mid u \in U\}$ と定義すると、RLF 法は $|E_v|$ の値が最も大きい節点 $v^* \in \hat{V} - U$ を次候補として選択する。

$$v^* = \arg \max_{v \in \hat{V} - U} |E_v| \quad (5)$$

できるだけ多くの節点に第 i 色を割り当てるが、それが不可能になった時に初めて第 $(i+1)$ 色を追加する。RLF 法の構築手順をもとに、ACO アルゴリズムでは推移可能な状態集合 J_i^k として $\hat{V} - U$ を採用する。

3.2.2 エージェントが分泌するフェロモン量 $\Delta\tau_{ij}^k$

RLF 法は、できるだけ多くの節点に第 i 色を割り当て、それが不可能になった時に初めて第 $(i+1)$ 色を追加する。この規則にしたがうと、同じ色で塗られる節点の数を増やせば使用される色数が少なくなることが期待できる。したがって、ACO は同じ色で塗るべき節点の間にフェロモンがより多く蓄積されるような機構を備える必要がある。ACO では、与えられたグラフ $G = (V, E)$ に対する補グラフ¹ $\bar{G} = (V, \bar{E})$ を考え、補グラフの節点对上にフェロモンを蓄積する。分泌するフェロモンの量は、獲得した実行可能解の評価値 z_k の逆数とし、実行可能解を獲得したすべてのエージェントがフェロモンを分泌する。したがって、エージェント k が節点对 $(i, j) \in \bar{E}$ 上に分泌するフェロモン量 $\Delta\tau_{ij}^k$ は次式のとおりである。

$$\Delta\tau_{ij}^k = \begin{cases} 1/z_k & i \text{ と } j \text{ が同じ色} \\ 0 & \text{その他} \end{cases} \quad (6)$$

3.2.3 問題固有の先見的な情報 η_{ij}

RLF 法が次候補の選択に使用する (5) 式を先見的な情報として用いることにする。節点 i にいるエージェントが次に彩色する節点 j を選択する際の、節点 j に対する先見的な情報 η_{ij} として $|E_j|$ を用いる。

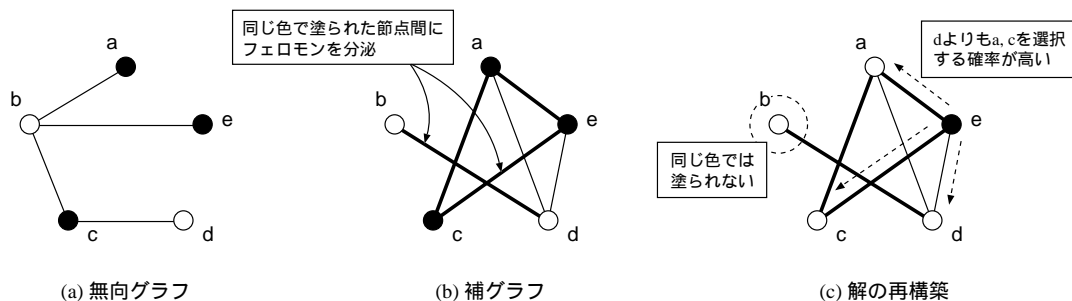


図 3: GCP に対する ACO アルゴリズムの解の構築手順

¹ グラフ G で節点对が存在しない節点間のみ節点对が存在するグラフ

3.3 大規模問題に対する高速化法

3.3.1 候補集合の限定

3.2 節で示した ACO アルゴリズムでは、次に第 i 色を割り当てる節点の候補として、色の割当てが決まっている節点集合 \hat{V} から、第 i 色では塗られないことが既に分かっている節点集合 U を除いた節点集合 $\hat{V} - U$ を候補集合とする。これには次候補を選択する際の計算量の面で次の 2 つの欠点がある。

- 1) 実行可能解の構築の早い段階に、非常に多くの節点を次候補として考慮しなければならない
- 2) フェロモンの蓄積(と蒸発)が進んだ段階に、フェロモン量が微小で選択される確率が 0 に近い節点も次候補として考慮しなければならない

グラフの節点数が少ない場合には問題にならないが、大規模な問題を扱う場合には計算量が膨大になることが予想される。提案する解法は、次に色を割り当てる節点の候補集合を限定することでアルゴリズムの高速化を図る。従来、他の組合せ最適化問題では候補集合を限定する際に、問題の構造から容易に計算できる情報を用いていた。例えば巡回セールスマン問題 (Traveling Salesman Problem) の場合、都市間の距離に基づき候補集合を限定している [2]。しかし、GCP では候補集合を限定するための適切な情報(どの 2 節点と同じ色で塗られやすいかを示す指標)をグラフ構造から得ることは非常に難しい。

本解法では、新たな基準として節点間に蓄積されるフェロモン量に注目し、その値が大きい節点のみを次候補として採用する。候補集合の大きさを一律 ℓ とすると、節点 i の候補集合 L_i は、節点をフェロモン量 τ_{ij} にしたがって降順に並び換えた時の上位 ℓ 個の節点からなる集合となる。したがって、本解法では確率的な経路選択を 2 段階で行なうことになる。まず、節点 i の候補集合 L_i 内の節点だけで確率的な経路選択を行なう。

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{s \in J_i^k \cap L_i} [\tau_{is}(t)]^\alpha [\eta_{is}]^\beta} \quad (7)$$

もし、節点集合 $dist_i$ 内の節点で推移可能な節点集合 J_i^k に含まれる節点が無い場合は、 J_i^k 内のすべての節点を候補として、(2) 式にしたがって確率的な経路選択を行なう。

3.3.2 挿入ソート法による計算の効率化

フェロモン量は各反復の最後で更新するため、解の探索が進むにつれてその値が動的に変化する。したがって、値が更新される度にすべての節点について候補集合を再計算する必要がある。これは膨大な計算量になる。しかし、フェロモンは蓄積と蒸発がある程度進むとその値は比較的安定し、急激な増加や減少はほとんど起こらない(図 4 参照)。この性質を利用して、節点をフェロモン量にしたがって並び換える際に挿入ソート法 (insertion sort) を用いる。挿入ソート法は、既にある程度整列された配列を並び換えるのに適したソート法であり、変動の少ないフェロモン量に対するソート法として適している。

具体的には、節点 i の次数を $\deg(i)$ で表すと、各節点に長さ $|V| - \deg(i) - 1$ の配列を用意する。各反復の最後に $\tau_{ij}(t)$ の値にしたがって配列の要素をソートする。その際、0 番目から $(\ell - 1)$ 番目までの要素が整列されれば良いので、第 k 要素 ($0 < k < \ell$) については通常の挿入ソート法を適用するが、それ以降の要素については第 $(\ell - 1)$ 要素より前の要素とのみ比較をすれば良い(図 4 参照)。

4 数値実験

提案する解法の有効性を数値実験 (PentiumII 300MHz) を用いて示す。本解法で設定すべきパラメータは、(7) 式で用いる 2 つのパラメータ α, β と、フェロモンの蒸発係数 ρ 、候補集合の大きさ ℓ の 4 種類がある。このうち ℓ を除く 3 種類のパラメータについては、予備実験より $(\alpha, \beta, \rho) = (2, 4, 0.5)$ に設定する。 ℓ は問題ごとに設定を変更した。また、エージェント数は 10、反復回数は 100 回とする。

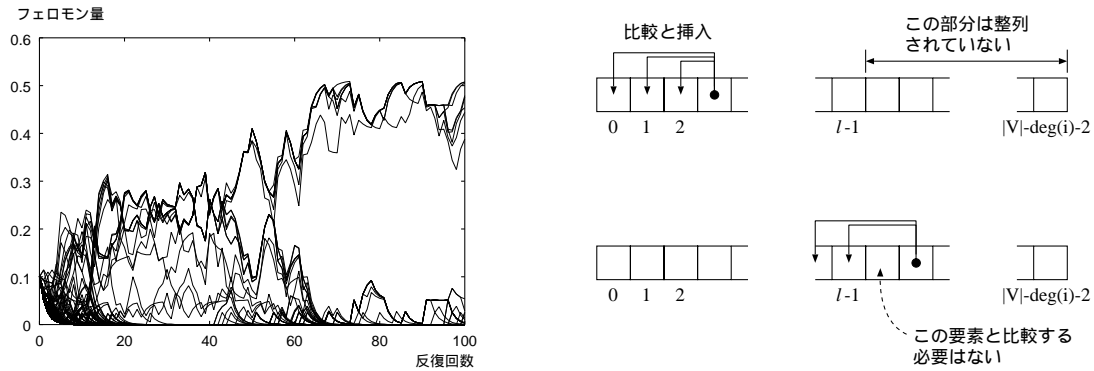
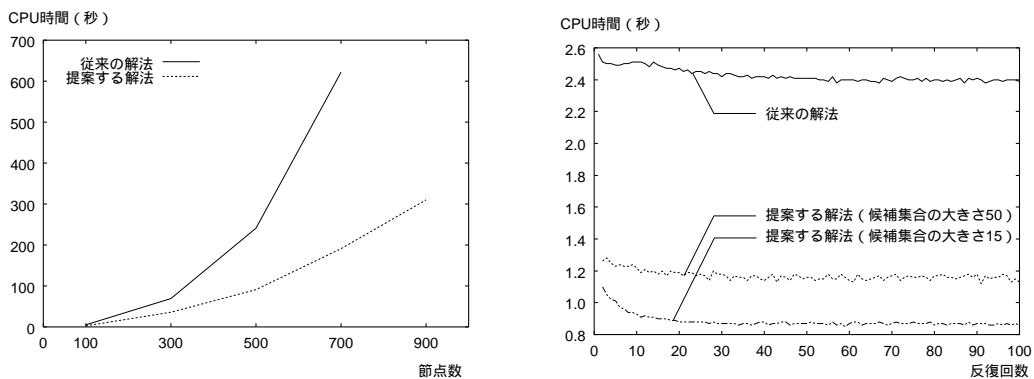


図 4: フェロモン量の変動(左)と候補集合の計算手順(右)

4.1 ランダムグラフを用いた平均的な挙動

数値実験に用いるランダムグラフは、節点数 n と節点对の存在確率 p を変化させて生成したランダムグラフを使用する。まず、節点对の存在確率 p を 0.5 に固定したランダムグラフを用いて、節点数 n の増加にともなう 2 つの ACO アルゴリズムの計算時間の違いを図 5 (a) に示す。候補集合の大きさ ℓ は 15 とする。提案する解法は従来の ACO アルゴリズムと比較して大幅に計算時間を短縮することが可能になる。特に、節点数を多くした際の計算時間に高速化の効果が顕著にあらわれる。

次に各反復ごとの計算時間の変化を図 5 (b) に示す。従来の解法に比べ、提案する解法は半分以下の計算時間で 1 反復を終えていることが分かる。また、候補集合の大きさ ℓ の違いによる計算時間の差も示されている。候補集合の大きさを小さくすることで、各反復ごとの計算時間は短縮できる。しかし、過度に候補集合を限定するとエージェントの解の探索範囲を狭めてしまう危険がある。



(a) 節点数による計算時間の変化

(b) 1 反復あたりの計算時間の変化(節点数 500)

図 5: 提案する解法による高速化の効果

4.2 ベンチマーク問題への適用結果

GCP のベンチマーク問題として広く用いられている DIMACS の問題群² に対し提案する解法を適用し、その有効性を検証する。ここでは、ベンチマーク問題として公開されている問題のうち、最適値が既知の 52 種類の問題を対象とする。適用結果を表 1 に示す。表中の値は各問題に対し最適解を獲得した回数を示

² [ftp://dimacs.rutgers.edu/pub/challenge/graph/benchmarks/color/](http://dimacs.rutgers.edu/pub/challenge/graph/benchmarks/color/)

表 1: ベンチマーク問題への適用結果 (エージェント数 n)

問題	(V , E)	最適値	RLF法	既存の解法*			提案する解法*		
				$n = 10$			$n = 10$		
flat300_20_0	(300, 21375)	20	38	1	1	5	1	1	5
flat300_26_0	(300, 21633)	26	39	0	0	0	0	0	0
flat300_28_0	(300, 21695)	28	39	0	0	0	0	0	0
le450_5a	(450, 5714)	5	8	5	5	5	5	5	5
le450_5b	(450, 5734)	5	7	5	5	5	3	5	5
le450_5c	(450, 9803)	5	5	5	5	5	5	5	5
le450_5d	(450, 9757)	5	7	5	5	5	5	5	5
le450_15a	(450, 8168)	15	17	0	0	0	0	0	0
le450_15b	(450, 8169)	15	16	0	0	0	0	0	0
le450_15c	(450, 16680)	15	23	1	0	5	0	5	5
le450_15d	(450, 16750)	15	24	0	1	4	1	4	4
le450_25a	(450, 8260)	25	25	5	5	5	5	5	5
le450_25b	(450, 8263)	25	25	5	5	5	5	5	5
le450_25c	(450, 17343)	25	28	0	0	0	0	0	0
le450_25d	(450, 17425)	25	28	0	0	0	0	0	0
queen5_5	(25, 160)	5	5	5	5	5	5	5	5
queen6_6	(36, 290)	7	8	1	2	3	2	3	3
queen7_7	(49, 476)	7	9	0	1	2	1	2	2
queen8_8	(64, 728)	9	10	0	0	2	0	2	2
queen8_12	(96, 1368)	12	13	4	3	5	3	5	5
queen9_9	(81, 2112)	10	11	0	0	1	0	1	1
queen11_11	(121, 3960)	11	14	0	0	0	0	0	0
queen13_13	(169, 6656)	13	16	0	0	0	0	0	0

* 値は最適解を獲得した回数 (実験回数 5 回)

している (最大 5 回)。ただし、表 1 で示した 27 種類を除く 25 種類の問題については、5 回の適用すべてにおいて非常に短い計算時間で最適解を探索することができたので省略した。同じパラメータ設定での数値実験では、提案する解法は従来の解法とほぼ同精度の解を得ることができる ($\ell = 30$)。さらに、エージェント数を 30 にした場合では最適解を獲得する割合が増加し、エージェント数が 10 の場合では最適解が得られなかった問題に対しても最適解を獲得している。

また、最適値が分かっていない 3 種類のベンチマーク問題 (節点数 1000) に対しては、従来の ACO アルゴリズムとほぼ同精度の解を最短で約 30% の計算時間で求めることができる (表 2 参照)。

表 2: 大規模なベンチマーク問題への適用結果 ($\ell = 100$ 、適用回数 5 回の平均)

問題	(V , E)	従来の ACO アルゴリズム			提案する高速解法		
		最良値	平均値	CPU 時間	最良値	平均値	CPU 時間
DSJC1000.1	(1000, 49629)	27	27.0	1933.4 秒	24	24.6	591.4 秒
DSJC1000.5	(1000, 249826)	102	102.6	1721.5 秒	102	103.4	617.9 秒
DSJC1000.9	(1000, 449449)	255	257.0	903.6 秒	255	256.6	627.8 秒

5 おわりに

本発表では、代表的な組合せ最適化問題であるグラフ彩色問題に対する ACO アルゴリズム [4] に候補集合の概念を導入し、解の探索過程で動的に変化する候補集合の計算を効率化することで高速化を図る方法について示した。また、数値実験により提案する解法の有効性を検証した。本解法を用いることで、大規模なグラフ彩色問題に対し既存の ACO アルゴリズムと同精度の解をより短時間で得ることができる。しかし、候補集合の大きさ cl に対する解の精度の振舞いについては十分な分析ができていない。今後はこれらの点を含め、提案した解法が持つ特性を明らかにする。

参考文献

- [1] Alberto Coloni, Marco Dorigo, and Vittorio Maniezzo. Distributed optimization by ant colonies. In *Proceedings European Conference on Artificial Life*, pages 134 – 142. MIT Press, 1991.
- [2] Eric Bonabeau, Marco Dorigo, and Guy Théraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, 1999.
- [3] David Corne, Marco Dorigo, and Fred Glover. *New Ideas in Optimization*. McGraw-Hill, 2000.
- [4] Daniele Costa and Alain Hertz. Ants can colour graphs. *Journal of the Operational Research Society*, 48:295–305, 1997.