

部分的形質遺伝に基づく ACO による制約充足問題の解法

On the Efficiency of Ant Colony Optimization with Cunning Ants
for Binary Constraint Satisfaction Problems

早川大貴*1 水野一徳*2 長澤圭孝*1 小野智司*3 西原清一*4 佐々木整*2
Daiki Hayakawa Kazunori Mizuno Yoshitaka Nagasawa Satoshi Ono Seiichi Nishihara Hitoshi Sasaki

*1 拓殖大学大学院工学研究科電子情報工学専攻

Department of Electronics and Information Science, Graduate School of Engineering, Takushoku University

*2 拓殖大学工学部情報工学科

Department of Computer Science, Faculty of Engineering, Takushoku University

*3 鹿児島大学理工学研究科情報生体システム工学専攻

Department of Information Science and Biomedical Engineering, Graduate School of Science and Engineering, Kagoshima University

*4 筑波大学大学院コンピュータサイエンス専攻

Department of Computer Science, University of Tsukuba

To solve large-scale constraint satisfaction problems (CSPs) that are NP-complete, meta-heuristics for stochastic search approaches has been recently made remarkable progress. Ant colony optimization (ACO), which is one of typical meta-heuristics, has been effective for many combinatorial search and optimization problems, where the behavior of real ants is modeled as a metaphor.

In this paper, we discuss the performance of cunning Ant System (cAS) in solving CSPs. We describe the method where, artificial ants construct a candidate solution by partially using building blocks, or useful partial solutions, of the candidate solution constructed at the previous cycle, instead of constructing the candidate solution entirely. The method is applied to large-scale binary CSPs that are in the phase transition region, whose experimental simulations demonstrated that how effective our method can be by variation of using the previous partial solution.

1. はじめに

大規模な制約充足問題 (CSP) に対して, 確率的探索アルゴリズムにおける局所最適解からの脱出のためのメタ戦略に関する研究が注目されている [Mizuno 01]. これは, アルゴリズムの完全性は保証されていないものの, 実用的な時間内になるべく高い確率で解を発見する, 実用可能な近似解を高速に求めようとするといったアプローチである. その中でも, アリの餌探しの際の行動を模倣した Ant Colony Optimization (ACO) [Dorigo 96, Dorigo 96] は, 新しいメタヒューリスティクスの一つとして近年活発に研究がなされ, これまで多くの組合せ最適化問題に応用されてその有効性が報告されている. Ant System (AS) は, ACO の仕組みを実現した探索アルゴリズムである. 一般にメタヒューリスティックを用いた探索では, 探索過程における多様化と集中化のバランスが重要であり, ACO においても同様である [Dorigo 04]. また, ACO の拡張研究としては, ASrank [Bullnheimer 99], ASHC [Mizuno 09], 多集団型 AS [Mizuno 10] などの研究がある.

本稿では, エージェントが解候補を構築する際, 他のエージェントの解候補の一部を生成に利用する (形質遺伝) という性質を持つエージェント (カニングアント, cunning ant; c-ant) を導入した ACO として cAS (cunning Ant System; カニングアントシステム) [Tsutsui 07] を Binary CSP [Prosser 96] に適用し, この方法の効果を実験的に確認する.

$$\begin{aligned} X &= \{x_1, x_2, x_3, x_4\} \\ D &= \{a, b, c\} (= D_1 = \dots = D_4) \\ C &= \{c_{12}, c_{23}, c_{14}\} \\ c_{12} &= \{(a, b), (b, c)\} \\ c_{23} &= \{(c, b), (b, a), (b, b)\} \\ c_{14} &= \{(a, c)\} \\ \text{解: } (x_1, x_2, x_3, x_4) &= \{(a, b, a, c), (a, b, b, c)\} \end{aligned}$$

図 1: Binary CSP の例

2. 研究分野の概要

2.1 制約充足問題

CSP とは, 問題の構成要素に対して制約が与えられたとき, すべての制約を一斉に満足させる問題である. 本研究ではランダムに生成された, 制約を二項間に限定した CSP (Binary CSP) を対象とする. 問題 P を 3 つ組 (X, D, C) で定義し, $X (= \{x_1, \dots, x_n\})$ は, n 個の変数の有限集合を表す. $D_i \in D$ は, 変数 x_i に割当て可能な値の有限集合を表している. 本研究では, $D = D_1 = \dots = D_n$ とし, $|D| = m$ とする. C は二項制約の集合を表す. その要素 $c_{ij} \in C$ は変数 x_i と x_j の間に制約が存在することを示しており, その変数間に割当て可能な値組の集合で記述される. 図 1 に Binary CSP の例を示す.

本研究では, $\langle n, m, p_1, p_2 \rangle$ の 4 つパラメータを用いて, ランダムな Binary CSP インスタンスを生成する. n は変数の数, m は変数に割当て可能な値の変域, p_1 は変数間に制約が存在する確率, p_2 は変数間に割当て可能な値組に関する確率を表す. また, 生成した問題が相転移領域 [Prosser 96] に近い難しい問題であるかを判断するために, パラメータ κ を

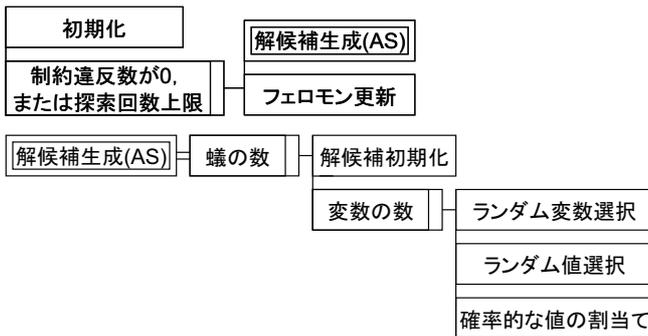


図 2: AS のアルゴリズム

用いてこれを (1) 式のように定義する [Clark 96]. この κ の値が 1 に近づくほど制約の強さがクリティカルな領域に近づくと報告されている.

$$\kappa = \frac{n-1}{2} p_1 \log_m \left(\frac{1}{1-p_2} \right) \quad (1)$$

κ の値が 1 未満ならば制約不足, 1 以上ならば制約過多, 約 1 ならば相転移領域に属していることを意味する.

2.2 ACO

ACO とは, 最近注目を浴びてきたメタヒューリスティクスで, アリのフェロモン軌跡の機能をモデルとしている [Solnon 02]. Ant System (AS) は Dorigo らによって開発された最初の ACO である. 図 2 に, AS のアルゴリズムを示す. 図 2 では, CSP に適用するため, フェロモン軌跡の蓄積量を以下に示されるグラフ構造 $G = (V, E)$ を用いている [Solnon 02].

$$V = \{ \langle x_i, v \rangle \mid x_i \in X \text{ and } v \in D_i \},$$

$$E = \{ (\langle x_i, v \rangle, \langle x_j, w \rangle) \in V^2 \mid x_i \neq x_j \}.$$

本論文では, 上記のグラフをフェロモングラフと呼び, $\langle x_i, v \rangle, \langle x_j, w \rangle$ に蓄積されるフェロモンの蓄積量を $\tau(\langle x_i, v \rangle, \langle x_j, w \rangle)$ と表記する.

AS では, アリ 1 匹が 1 つの解候補を生成していくことで解探索を進める. 解候補の生成では, 未割当ての変数からランダムに 1 つ選び (x_j), その変数に割当て可能な値 (v) を (2) 式の確率で選び出すという処理を, すべての変数に値が割当てられるまで繰り返す.

$$p_A(\langle x_j, v \rangle) = \frac{[\tau_A(\langle x_j, v \rangle)]^\alpha [\eta_A(\langle x_j, v \rangle)]^\beta}{\sum_{w \in D_j} [\tau_A(\langle x_j, w \rangle)]^\alpha [\eta_A(\langle x_j, w \rangle)]^\beta} \quad (2)$$

ただし,

$$\tau_A(\langle x_j, v \rangle) = \sum_{\langle x_k, w \rangle \in A} \tau(\langle x_k, w \rangle, \langle x_j, v \rangle),$$

$$\eta_A(\langle x_j, v \rangle) = \frac{1}{1 + \text{conf}(\{\langle x_j, v \rangle\} \cup A) - \text{conf}(A)}.$$

ここで, $\text{conf}(A)$ は解候補 A の制約違反数を表す. (2) 式におけるパラメータ α と β はそれぞれ, 大域性の重みと局所性の重みを表わしていて, この 2 つのパラメータのバランスをとることが重要である.

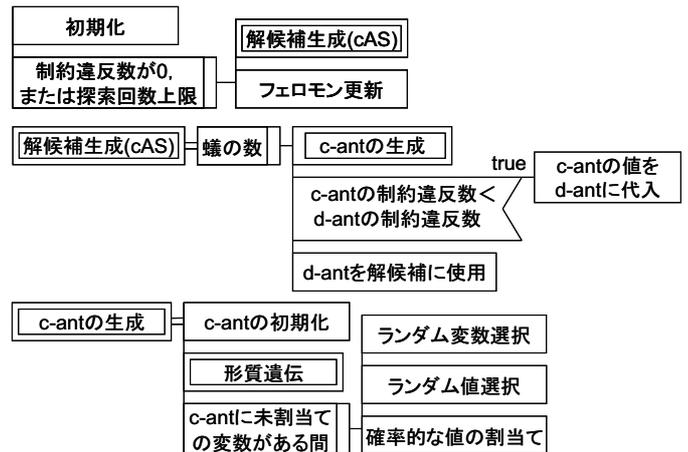


図 3: 本手法のアルゴリズム

AS では, 1 サイクルごとにすべてのアリが解候補を生成する. しかし, 生成した解候補はサイクルごとに初期化され, 次サイクルでは新たに解候補を生成するため解候補生成の処理に手間がかかってしまう [Mizuno 09].

2.3 cAS

cunning Ant System (cAS) とは, エージェントが解候補を構築する際に, 他のエージェントの解候補の一部を形質遺伝するという性質を持つエージェント (カニングアント, cunning ant; c-ant) を導入した ACO である. また, カニングアントに形質遺伝を許すエージェントをドナーアント (donor ant; d-ant) と呼ぶ.

従来の ACO では, 各エージェントの解候補生成はフェロモン濃度に基づき一から解を見つけなければならない. これに対し, 文献 [Tsutsui 07] で提案しているカニングアントというエージェントは, 解候補を生成する際, 他のエージェントの解候補の一部を形質遺伝して利用するため探索時間の短縮が期待できる. 残りの部分の解候補生成については, 従来と同様フェロモン濃度を利用する [Tsutsui 07].

3. cAS による Binary CSP の解法

3.1 基本方針

本研究 [Hayakawa 10] では, 文献 [Tsutsui 07] の cAS を Binary CSP に適用する. その基本方針は以下の 3 点である.

- (i) 初期化の際に AS により解候補を生成する.
- (ii) 各サイクルでは cAS により解候補を生成する.
- (iii) c-ant と d-ant により生成された解候補のうち, 制約違反数の少ない方を次サイクルに残す.

これにより, 従来の ACO では優れた解が見つかったとしても各サイクルで解候補を初期化し, 排除していたものをカニングアントを用いることで解候補の一部を生成に利用することができ, 探索時間を短縮することができる.

3.2 アルゴリズム

図 3 に本手法のアルゴリズムを示す. 本手法は cAS によって解候補の構築を行う. cAS ではまず, 形質遺伝をするエージェントである c-ant を用いて解候補を生成する. 形質遺伝を

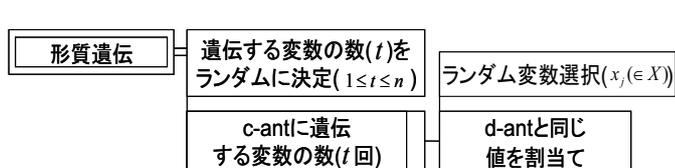


図 4: (a) のアルゴリズム

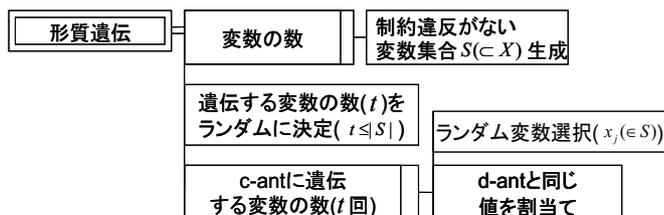


図 5: (b) のアルゴリズム



図 6: (c) のアルゴリズム

しない残りの部分については、従来と同様にフェロモン濃度を利用して解候補の生成を行う。その後、c-ant と d-ant の合計制約違反数を比較する。比較した制約違反の多い方を排除し、残った方を次のサイクルの d-ant にする。

今回は形質遺伝する部分解の選択の方法 (図 3 における形質遺伝) として以下の 3 つを考えた。

- (a) ランダム
- (b) 制約違反のないものの中からランダム
- (c) 制約違反のないものすべて

図 4, 図 5, 図 6 にそれぞれ (a), (b), (c) のアルゴリズムを示す。(a) では、まず c-ant に遺伝する変数の数をランダムに決定する。次に、その数の中から c-ant に遺伝する変数をランダムに決め、c-ant に値を代入する。(b) では、cAS により生成された解候補のうち制約違反のない変数を選び出す。次に、その中からランダムに変数を選び出し、c-ant に値を代入する。(c) では、cAS により生成された解候補のうち制約違反のない変数を選び出し、c-ant に値を代入する。

4. 評価実験

4.1 実験条件

今回の実験では、本手法における形質遺伝する部分解の選択の 3 つの方法の性能を比較するため解探索を試みた。ここでは、 $\langle n, m, p_1, p_2 \rangle = \langle 150, 4, 0.14, p_2 \rangle$ と $\langle n, m, p_1, p_2 \rangle = \langle 200, 4, 0.14, p_2 \rangle$ の Binary CSP を対象とした。探索空間のサイズはそれぞれ $4^{150} (\approx 10^{90})$, $4^{200} (\approx 10^{120})$ となる。 p_2 の値は、 $n=150$ では $p_2 = 0.10 \sim 0.20$ で 0.02 おきの 6 ケースに対して 100 問ずつ (計 600 問)、 $n=200$ では $p_2 = 0.08 \sim 0.20$ で 0.02 おきの 7 ケースに対して 100 問ずつ (計 700 問) ランダムに生成した可解なインスタンスを使用した。これらは式 1 に基づいて、解くのが非常に難しい (相転移) 領域の問題である。すべての方法において、アリの数 $nbAnts=50$ 、蒸発係数 $\rho=0.01$ 、重み $\langle \alpha, \beta \rangle = \langle 5, 5 \rangle$ とし、探索回数 (解候補生成数) 上限は 400 とした。

また、実験での評価項目は、

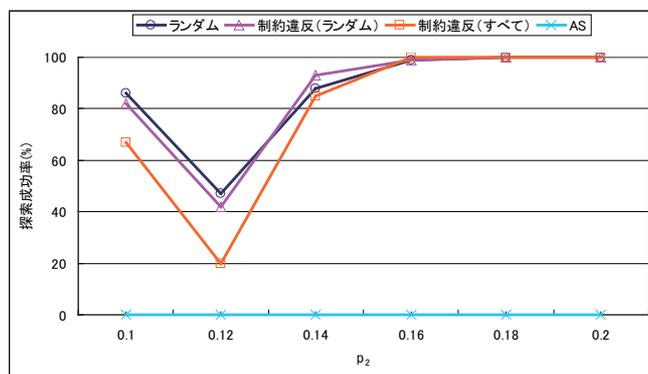


図 7: (a)(b)(c) の探索成功率

- (1) 探索成功率 (%): 解探索を行なった CSP のうち解を発見することができたものの割合。
- (2) 平均探索時間 (sec.): すべての問題に対して探索が終了するまでにかかった平均 CPU 時間。

とした。

4.2 実験結果と考察

図 7 は $\langle 150, 4, 0.14, p_2 \rangle$ の Binary CSP の実験結果 (探索成功率) を示している。図 8 は $\langle 150, 4, 0.14, p_2 \rangle$ の Binary CSP の実験結果 (平均探索時間) を示している。図 7 より、(c) に比べると (a), (b) は全体的に探索成功率が向上している。また、解くのが難しい領域であると思われる $p_2=0.12$ でも探索成功率が向上していることが分かる。図 8 より、(a) に比べると制約違反のないものの中から形質遺伝を行っている (b), (c) は探索時間が改善できている。特に、 $p_2=0.12$ ではその幅が大きいことが分かる。

図 9 は $\langle 200, 4, 0.14, p_2 \rangle$ の Binary CSP の実験結果 (探索成功率) を示している。図 10 は $\langle 200, 4, 0.14, p_2 \rangle$ の Binary CSP の実験結果 (平均探索時間) を示している。図 9 より、(c) は解くのが難しい領域であると思われる $p_2 = 0.08$ では、ほとんど解が発見できなかった。しかし、図 8 と同様に図 10 では、(a), (b) に比べて探索時間が改善できていることが分かる。これは、変数選択の際に制約違反のないものの中から選んでいるため、最初は制約違反が多くても探索が進んでいくにつれて制約違反が少なくなってくるためだと考えられる。

5. おわりに

本稿では、大規模な制約充足問題を効率よく解くための方法として、cAS を用いてランダムに生成した可解な Binary CSP を解き、形質遺伝の方法の違いによる効果を実験的に示した。実験結果より、変数選択の際にランダムに選ぶよりも制約違反のないものの中から選ぶ方が探索時間は改善できるということ

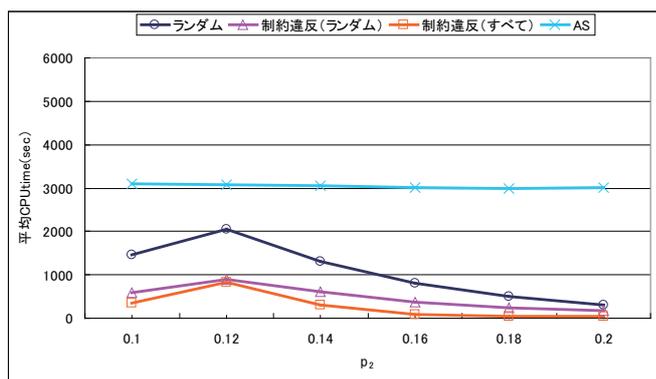


図 8: (a)(b)(c) の平均探索時間

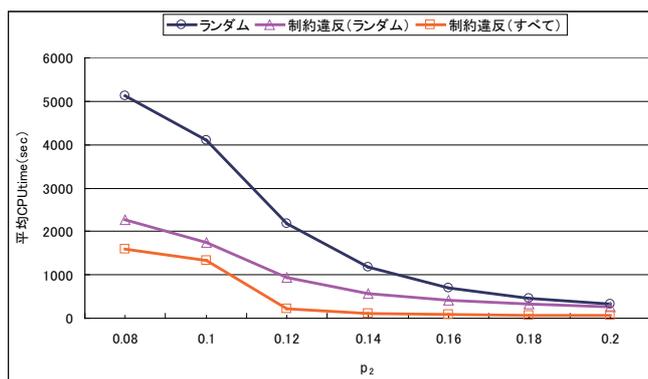


図 10: (a)(b)(c) の平均探索時間

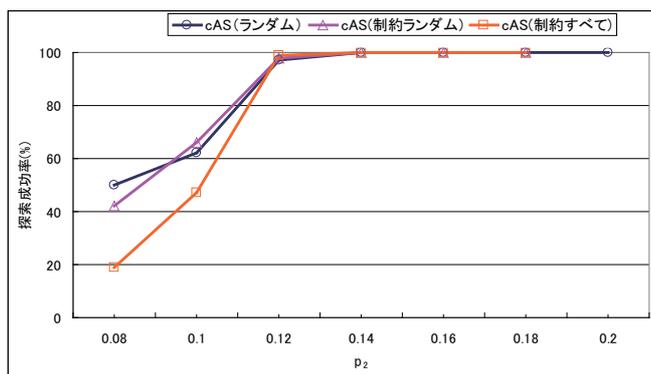


図 9: (a)(b)(c) の探索成功率

を実験で確認した。しかし、単に制約違反のないものをすべて遺伝してしまうと探索成功率は低下するという事も確認できた。今後は、今回考えた変数選択の方法 3 つを並列化するなどして効率化のためのさらなる改善を図る。

参考文献

[Mizuno 01] Mizuno, K., Nishihara, S. et al.: Population migration: a meta-heuristics for stochastic approaches to constraint satisfaction problems, *Informatika*, Vol. 25, No. 3, pp. 421–429 (2001).

[Dorigo 96] Dorigo, M., Di Caro, G.: The Ant Colony Optimization meta-heuristic, *New Ideas in Optimization*, pp. 11–32 (1996).

[Dorigo 96] Dorigo, M., et al.: The Ant System: Optimization by a Colony of Cooperating Agents, *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, Vol. 26, pp. 26–41 (1996).

[Dorigo 04] Dorigo, M., Stutzle, T.: *Ant Colony Optimization*, MIT press, Massachusetts (2004).

[Tsutsui 07] 筒井 茂義: 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, 人工知能学会 (2007).

[Prosser 96] Prosser, P.: An empirical study of phase transitions in binary constraint satisfaction problems, *Artificial Intelligence*, Vol. 81, pp. 81–109 (1996).

[Clark 96] Clark, D.A., et al.: Local Search and the Number of Solutions, *Proc. CP'96*, pp. 119–133 (1996).

[Solnon 02] Christine Solnon: Ants Can Solve Constraint Satisfaction Problem, *IEEE transactions on evolutionary computation*, Vol. 6, No. 4, pp. 347–357 (2002).

[Mizuno 09] 水野一徳, 小野智司, 西原清一, 佐々木整: 制約充足型 Ant System における局所探索による解候補育成の効果, *進化計算シンポジウム 2009*, (2009).

[Bullnheimer 99] 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*, 7(1), 25–38 (1999).

[Hayakawa 10] 早川大貴, 水野一徳, 長澤圭孝, 小野智司, 西原清一, 佐々木整: 制約充足型 ACO における部分的形質遺伝に基づく解候補育成, *進化計算シンポジウム 2010*, (2010).

[Mizuno 10] Mizuno, k., et.al.: Ant Population Meta-Heuristics for Binary Constraint Satisfaction Problem, *Proc, TAAI 2010*, pp. 314–321(2010).