

F-017

# Binary CSP のための制約違反最小化戦略の ハイブリッド型 Ant System の効果

## Toward Effect of a Hybrid Ant System with Min-Conflict Heuristics for Binary Constraint Satisfaction Problems

水野 一徳 † 奥住 雅仁 † 佐々木 整 † 西原 清一 ‡  
Kazunori Mizuno Masahito Okuzumi Hitoshi Sasaki Seiichi Nishihara

### 1 はじめに

近年, 大規模な制約充足問題 (CSP) に対して, メタヒューリスティクスを用いた確率的解探索アルゴリズムで効率的に解こうという研究が盛んに行なわれている [1]. Ant System (AS) [2] は, 代表的なメタヒューリスティクスの 1 つである Ant Colony Optimization (ACO) に基づいて開発された探索アルゴリズムであり, 多くの組合せ最適化問題に対して有効な手法として研究されている. しかし, CSP を解くために必要な局所探索能力が十分でないために探索時間がかかってしまう場合がある.

本報告では, AS の局所探索能力を補うために, CSP の確率的探索としてしばしば引用される制約違反最小化戦略による山登り法を組み合わせた探索アルゴリズムを提案し, これにより探索時間が改善されることを実験的に示す.

### 2 研究分野の概要

#### 2.1 対象問題

CSP とは, 離散値をとるいくつかの変数に割当て可能な値の組合せのうち, 与えられた制約をすべて満たすものを探索することによって発見する問題である. 本研究では, 二項制約のみからなる CSP (Binary CSP) を対象とし, 実験には,  $\langle n, m, p_1, p_2 \rangle$  で特徴づけられたランダムに生成されたインスタンス (具体的問題) を用いる. ただし,  $n$ : 変数の数,  $m$ : 値の数,  $p_1$ : 変数間制約が発生する確率,  $p_2$ : 値間の制約 (部分解) が発生する確率を表わす.

#### 2.2 Ant System とその問題点

ACO は, 蟻の餌探しの行動を模したメタヒューリスティクスであり, AS は ACO を基にした最初の探索アルゴリズムである. 図 1 に AS のアルゴリズムを示す [2]. AS では, 蟻一匹が 1 つの解候補を生成するサイクルを繰り返す. 解候補の生成では, ランダムに選ばれた変数 ( $X_j$ ) と値 ( $v$ ) を (1) 式で示される確率  $P(X_j, v)$  で割当てられるかどうかをすべての変数に値が割当てられるまで繰り返す.

$$P(X_j, v) = \frac{[\tau(X_j, v)]^\alpha [\eta(X_j, v)]^\beta}{\sum_{w \in D(X_j)} [\tau(X_j, w)]^\alpha [\eta(X_j, w)]^\beta} \quad (1)$$

† 拓殖大学 工学部 情報工学科, Department of Computer Science, Takushoku University

‡ 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

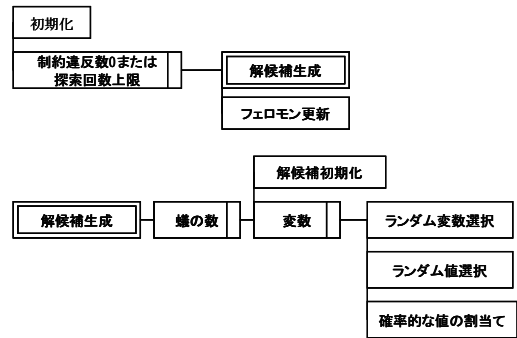


図 1 AS のアルゴリズム

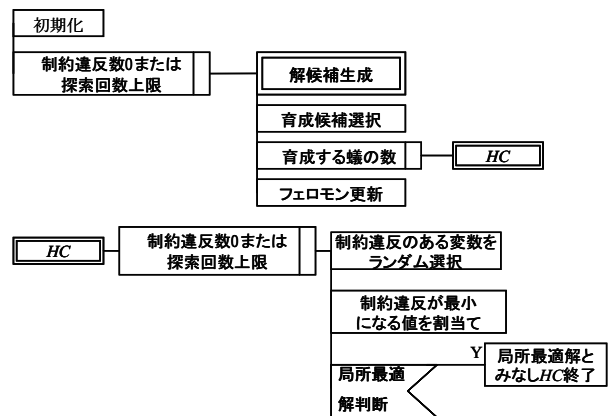


図 2 本手法のアルゴリズム

ただし,  $\tau$  はフェロモン蓄積量,  $\eta$  は制約違反数の変化量,  $D(X_j)$  は, 変数  $X_j$  に割当て可能な値の集合を表わす.

AS では, 1 サイクルごとにすべての蟻が解候補を生成する. しかし, 生成した解候補は初期化され, 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. また, フェロモン更新では, 最も制約違反数の少ない解候補を用いて行なわれるため, 破棄された解候補の中にも有望な解候補が存在する可能性もあるため解探索が非効率になる場合がある.

### 3 提案するハイブリッド型 Ant System

#### 3.1 基本方針

前節の問題点を踏まえ, 本研究では AS によって生成された一部の解候補に対して局所探索を施し, 全体の解探索効率を向上させることを試みる. 本手法の基本方針

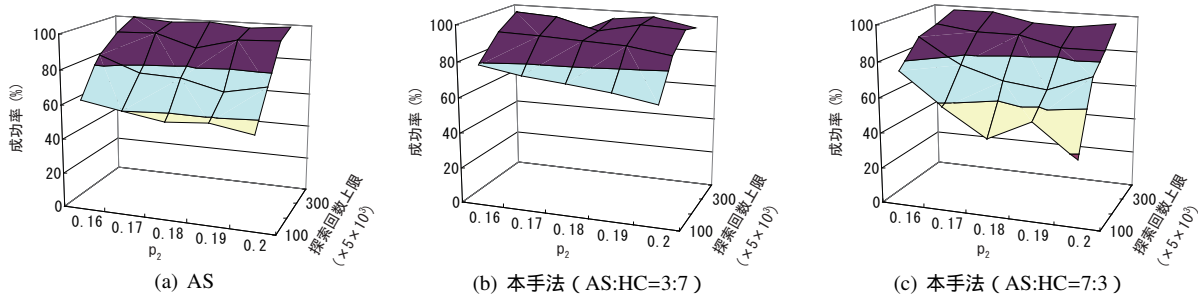


図3 実験結果 (探索成功率)

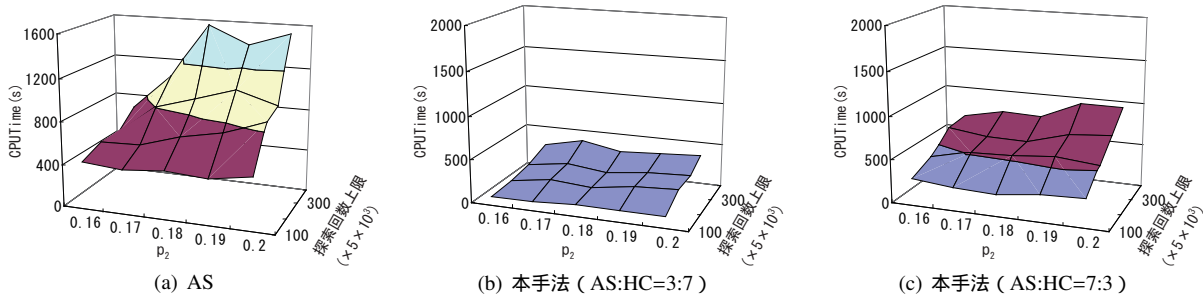


図4 実験結果 (CPU 時間)

は以下の2点である。

- i) 各サイクルでは AS により解候補を生成する。
- ii) 生成された解候補のうちの一部を選択して、それらに対して制約違反最小化戦略による山登り法 (MCHC) を適用する (育成処理)。

### 3.2 アルゴリズム

本手法のアルゴリズムを図2に示す。本手法は、ASと同様の方法で解候補を生成した後、一部の解候補を育成候補として選択し、その選ばれた解候補に対してMCHCを適用する。

## 4 評価実験

ASと本手法の性能を比較するため解探索を試みた。ここでは、 $\langle n, m, p_1, p_2 \rangle = \langle 100, 4, 0.14, p_2 \rangle$ のBinary CSPを対象として、 $p_2 = 0.16 \sim 0.20$ で0.01おきの5ケースに対して100問ずつランダムに生成したインスタンスを使用した。このインスタンスはいわゆる相転移が発生する難しい問題の領域に属する[3]。AS、本手法ともに蟻の数=50、蒸発係数 $\rho=0.01$ 、 $\alpha=1$ 、 $\beta=10$ とし、両手法の総探索回数がほぼ同等になるように、本手法では、ASにおける解候補生成(サイクル数)とMCHCの適用回数を設定した。

図3, 4に実験結果を示す。ここで、これらの図の本手法におけるAS:HCは、総探索回数のうちそれぞれのアルゴリズムに設定した探索回数の割合を表わしている。図3より、両手法とも総探索回数が増えるにしたがって解を発見できる割合(探索成功率)が向上していることが分かる。しかし、図4より、本手法では、ASと探索成功率が同等にもかかわらず探索時間が大幅に改善できていることが分かる。本手法においてASとMCHCの探

索回数の割合の変化が解探索の性能に影響を与えているものの、いずれの場合においてもASと比較するとより高速な解探索が実現できている。これは、通常のASでは破棄されていた解候補の中に有望な解候補が存在しており、それらから育成候補として選んだ解候補にMCHCを適用したことにより解探索の効率化が図ることができているものと考えられる。

## 5 おわりに

本報告では、大規模な制約充足問題を効率的に解くための方法として、ASに局所探索を組み合わせた方法を提案してその効果を実験的に示した。本手法は、ASによって生成された解候補のうちの一部を育成候補として制約違反最小化戦略による山登り法を適用するものである。また、二項制約のみからなるCSPに適用し、ASと比較して探索成功率をほぼ維持したまま、大幅に探索時間を改善できていることを実験で確認した。今後は、育成候補の選択方法の検討や他のCSPへの適用などにより本手法の適用範囲を明確にすることが重要である。

## 参考文献

[1] Mizuno, K. et. al.: Population migration: a meta-heuristics for stochastic approaches to constraint satisfaction problems, *Informatica Journal*, No. 25, pp. 421-429 (2001).  
 [2] Solnon, C.: Ants Can Solve Constraint Satisfaction Problem, *IEEE transactions on evolutionary computation*, Vol.6, No.4, pp.347-357(2002).  
 [3] Clark, D. A., et. al.: Local search and the number of solutions, *Proc. CP '96*, pp. 119-133 (1996).