1S-5

制約充足型 Artificial Bee Colony アルゴリズムの提案

荒津 裕子† 水野 一徳† 佐々木 整† 西原 清一‡
†拓殖大学工学部情報工学科 ‡筑波大学大学院コンピュータサイエンス専攻

1 はじめに

大規模な制約充足問題（Contraint Satisfaction Problem : 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．また近年，新しく提案されたメタヒューリスティクスの一つとして Artificial Bee Colony（ABC）アルゴリズムがある．ABC アルゴリズムは，ミツバチの群れによる知的な採餌行動に着想を得たもので，関数最適化を対象としたアルゴリズムとして 2005 年に D. Karaboga らによって初めて提案された．

本研究では，CSP に対応した ABC アルゴリズムを提案し，それを用いることによってより効率的に CSP を解くことを試みる．

2 研究分野の概要

2.1 CSP

CSP とは，離散値をとる複数の変数に割当て可能な値の組合せから，与えられた制約すべてを満たす組合せを探索する問題である．本研究では，すべての制約が二項制約である CSP (Binary CSP) を対象とする．図 1 は Binary CSP の例を示している．

$X = \{x_1,\ x_2,\ x_3,\ x_4\}$
$D = \{a,\ b,\ c\}\ (= D_1 = \cdots = D_4)$
$C = \{c_{12},\ c_{23},\ c_{14}\}$
$c_{12} = \{(a\ b),\ (b\ c)\}$
$c_{23} = \{(c\ b),\ (b\ a),\ (b\ b)\}$
$c_{14} = \{(a\ c)\}$
Solutions: $(x_1\ x_2\ x_3\ x_4) = \{(a\ b\ a\ c),\ (a\ b\ b\ c)\}$

図 1: BinaryCSP の例

2.2 ABC アルゴリズム

ミツバチの群れは，Employed Bees (雇用ハチ)，Onlooker Bees (見物ハチ), Scout Bees (偵察ハチ) の 3 つの特徴を持つグループから構成される [1]．各雇用ハチは特定の一つの食物源を担当し，その近傍を記憶し担当する．食物源が枯渇した場合，その雇用ハチは偵察ハチになる．各見物ハチは巣箱内に戻ってきた雇用ハチのダンスの様子から食物源の位置や豊かさの情報を取得し，食物源を一つ選びその近傍を探索する．各偵察ハチはランダムに新たな食物源を探索する．ABC アルゴリズムにおいて，食物源の位置は対象問題の解候補を，そして食物源の蜜の量はその解の質 (適応度) を表す．

Artificial Bee Colony for Constraint Satisfaction Problems
Yuko Aratsu†, Kazunori Mizuno†, Hitoshi Sasaki†, and Seiichi Nishihara‡
†Department of Computer Science, Takushoku University
‡Department of Computer Science, University of Tsukuba

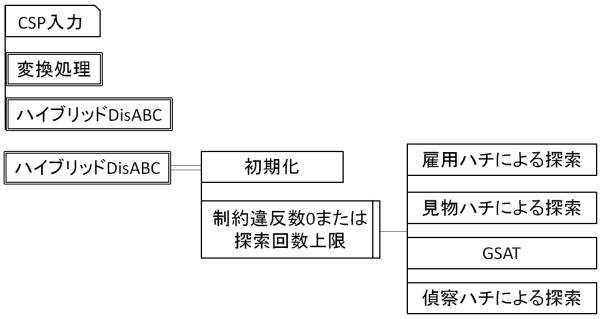

図 2: 本手法のアルゴリズム

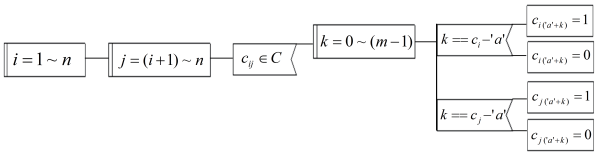

図 3: 変換処理

2.3 DisABC

ABC は，連続空間を対象としたアルゴリズムであるため，離散空間においてそのままの形では適用できない．そこで，離散空間に対応した手法を提案し CSP が解けるように適用させる必要がある．

DisABC は，連続空間を対象とした ABC をバイナリー最適化を対象とした ABC に改良したもので [2] において提案された．オリジナルの ABC では食物源の距離を測るために，マイナス演算子が使われている．DisABC では，マイナス演算子の代わりに集合の類似度を測る Jaccard 係数を利用した Dissimilarity を用いる．$X_i = (x_{i1}, x_{i2}, \cdots, x_{id}, \cdots, x_{iD})$，$X_j = (x_{j1}, x_{j2}, \cdots, x_{jd}, \cdots, x_{jD})$, x_{id} と x_{id} が 0，1 の値を取るとき, $Dissimilarity\,(X_i, X_j)$ は以下のように定義される．

$$Dissimilarity\,(X_i, X_j) = 1 - Similarity\,(X_i, X_j)$$

ただし，Similarity は以下のように定める．

$$Similarity\,(X_i, X_j) = \frac{M_{11}}{M_{11}+M_{10}+M_{01}},$$
$$M_{11} = \sum_{d=1}^{D} I(x_{id} = x_{jd} = 1),$$
$$M_{10} = \sum_{d=1}^{D} I(x_{id} = 1, x_{jd} = 0),$$
$$M_{01} = \sum_{d=1}^{D} I(x_{id} = 0, x_{jd} = 1),$$
$$M_{00} = \sum_{d=1}^{D} I(x_{id} = x_{jd} = 0).$$

3 提案する手法

3.1 基本方針

ABC を用いて CSP を解くために, 以下の 2 つの方針を立てた．

$X = \{(x_{1a}, x_{1b}, x_{1c}),$
$(x_{2a}, x_{2b}, x_{2c}),$
$(x_{3a}, x_{3b}, x_{3c}),$
$(x_{4a}, x_{4b}, x_{4c})\}$
$D = \{0, 1\}\ (= D_{1a} = \cdots = D_{4d})$
$C = \{c_{1*2*}, c_{2*3*}, c_{1*4*}\}$ (*: a, b, or c)
$c_{1*2*} = \{(1\,0\,0\,0\,1\,0), (0\,1\,0\,0\,0\,1)\}$
$c_{2*3*} = \{(0\,0\,1\,0\,1\,0), (0\,1\,0\,1\,0\,0), (0\,1\,0\,0\,1\,0)\}$
$c_{1*4*} = \{(1\,0\,0\,0\,0\,1)\}$
Solutions: $(x_{1*}\ x_{2*}\ x_{3*}\ x_{4*}) = \{(1\,0\,0\,0\,1\,0\,1\,0\,0\,0\,0\,1),$
$(1\,0\,0\,0\,1\,0\,0\,1\,0\,0\,0\,1)\}$

図 4: 変換後の BinaryCSP の例

i. DisABC に適用するため一般の CSP を割当てるべき値数が 2 つに限定された ($|D| = 2$) 問題に等価変換する.

ii. 局所探索能力を補うために GSAT[3] を組込む.

このように, 本手法では CSP を変換処理する方法を提案することによって, DisABC を用いることができるようにし, また, 局所探索能力を補うことで, 解への収束性を高めることができる.

3.2 アルゴリズム

本手法のアルゴリズムを図 2 に示す. また, 図 1 の CSP の例に変換処理を施した後の CSP の例を図 4 に示す. 変換処理のアルゴリズムを図 3 に示す. 図 2 では, 入力された CSP を変換後, ハイブリッド DisABC によって探索が行われる. ハイブリッド DisABC とは, DisABC に GSAT の処理を付け加えたものである.

4 実験

本手法を用いて実験を行った. 対象問題は, 変数 (n) を 30, 変域を (m)4, 変域間制約が発生する確率 (p_1) を 0.14, 値間制約に制約が発生する確率 (p_2) の各パラメータを, <30, 4, 0.14, 0.30~0.66> と <50, 4, 0.14, 0.14~0.50> とする. ランダムに生成した Binary CSP とした. ただし, 解を必ず含んでいるものとする. ABC のパラメータは食物源数を 50, 探索回数上限を 10000, 雇用ハチが担当する食物源を諦める断念反復回数を $n \times m$ とし, 以下の 2 つの条件で実験を行った.

a. 1 回の探索で 5 %の確率で GSAT が実行され, すべての食物源に対して 10 回フリップ処理が適用される.

b. 1 回の探索で 1 %の確率で GSAT が実行され, すべての食物源に対して 30 回フリップ処理が適用される.

また, 評価実験は次の 2 項目について行った.

- 探索成功率 (%) : 解を発見することができた割合
- 探索コスト (cycle) : 解を発見することができたとき探索にかかったサイクル数の平均値

図 5, 6 に実験結果を示す. 図 5, 6 より, 探索成功率が下がるとともに探索コストがかかっていることおよび相転移による問題の難易度の変化が確認できる. 図 5 においては, ほとんどの領域で b の条件よりの a の条件の方が探索成功率が高くなっており, 図 6 においては, b の条件よりも a の条件の探索成功率の方が難しい領域では探索成功率が下がっているが, 簡単な領域の問題になるにつれ, 成功率が上がっていることがわかる.

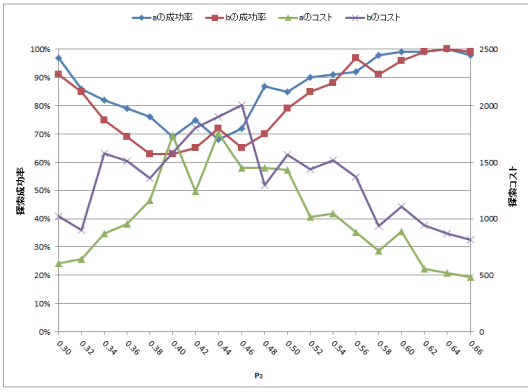

図 5: 変数 30 の場合

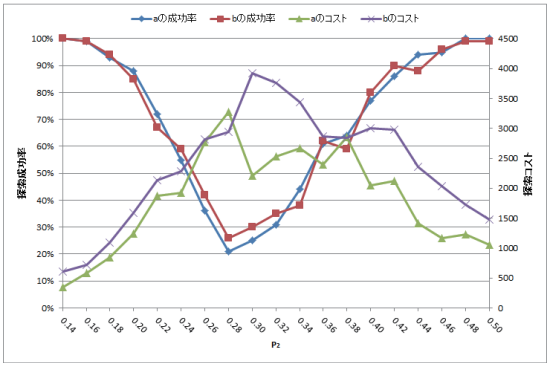

図 6: 変数 50 の場合

5 おわりに

本報告では, ABC アルゴリズムを用いて CSP を解くための方法として, CSP を変換処理する手法を提案し, 2 つの条件のもとで GSAT を動作させた実験を行った. 今後, さらに条件を増やして実験を重ねる予定である. また, 本手法の有効性をはかるため他のアルゴリズムとの比較実験を行う予定である.

参考文献

[1] 飯村伊智郎, 中山茂: 関数最適化を対象とした Artifitial Bee Colony アルゴリズムのロバスト性に関する研究, 進化計算シンポジウム 2010, pp. 42–46 (2010).

[2] Kashan, M.H., Nahavandi, N. and Kashan, A.H.: "DisABC: A new artificial bee colony algorithm for binary optimization, Applied Soft Computing 12, pp. 342–352 (2012).

[3] Selman, B., Levesque, H., and Mitchell, D.: A New Method for Solving Hard Satisfiability Problems, Proceedings of AAAI' 92, pp. 440-446 (1992).

[4] Aratsu, Y., Mizuno, K. *et. al.*: Artificial Bee Colony for Constraint Satisfaction Problems, SCIS-ISIS2012, pp. 2283–2286 (2012).