

フェロモン更新に負のフィードバックを取り入れたACOによる 制約充足問題の解法

ACO with Pheromone Update by Negative Feedback Can Solve CSPs

増金拓弥 水野一徳
Takuya Masukane Kazunori Mizuno

拓殖大学大学院工学研究科情報・デザイン工学専攻
Department of Computer Science, Takushoku University

To solve large-scale constraint satisfaction problems, ant colony optimization (ACO) has recently been drawing attentions. In algorithms based on ACO, candidate assignments are constructed by taking account of pheromone trails, which are updated based on a candidate assignment with the least number of constraint violations. In this paper, we propose an ACO model with dual pheromone trails: usual pheromone trails and another pheromone trails. Another pheromone trails are updated based on a candidate assignment with the largest number of constraint violations. Also, usual pheromone trails are updated by considering another pheromone trails as well. We demonstrated that our model, which is applied to the cunning ant system, can be effective than other ACO-based methods for large-scale and hard graph coloring problems whose instance appears in the phase transition region.

1. はじめに

制約充足問題 (CSP) は、離散値をとるいくつかの変数について割当て可能な値の組合せのうち、与えられた全ての制約を満たす組合せを、探索によって発見する問題である。CSP は設計や計画問題などをはじめ、人工知能分野やパターン処理などの広い分野にわたって応用されている基盤的技術である。

CSP を解くための探索アルゴリズムは、系統的探索アルゴリズムと確率的探索アルゴリズムに大別される。大規模な CSP に対して、系統的探索アルゴリズムを用いて探索を行なう場合、実用的な時間内に解を得ることが困難である場合が多い。そのため、近年では確率的探索アルゴリズムが注目されている。しかし、確率的探索アルゴリズムには、局所最適解に陥ってしまう可能性があるという欠点がある。そこで、局所最適解に陥ることを避ける、または局所最適解から効率よく抜け出すためのメタヒューリスティクスについて研究がなされている。

蟻コロニー最適化 (ACO) は、蟻の採餌行動におけるフェロモンコミュニケーションをモデル化したメタヒューリスティクスである [Dorigo 96, Dorigo 99]。ACO では、探索の過程で発見された評価が高い解候補の情報を、フェロモンとして蓄積する。このフェロモンを解候補の生成時に参考にし、フェロモンが多く溜まっている値の組合せほど、解候補として選択されやすくなる。

従来の ACO アルゴリズムでは、フェロモン (評価が高い解候補) と制約違反という情報を参考にして探索を行なう。本研究ではこれらの情報に加え、“評価が低い解候補” の情報を探索の参考にする ACO モデルを提案する。特に、評価が低い解候補を、従来のフェロモンとは別のフェロモン (ネガティブフェロモンと呼ぶ) として蓄積させる ACO モデルを提案する。更に本手法では、ネガティブフェロモンが多く溜まっている値の組合せほど、通常のフェロモンの蓄積量を減少させる。これにより解候補が、評価が高い解候補へ誘引されつつ、評価が低い解候補から遠ざけられ、より効率的な探索が行なわれることが期待される。

2. 研究分野の概要

2.1 グラフ彩色問題

グラフ彩色問題 (COL) は制約充足問題に分類される問題の 1 つである。COL は、無向グラフにおいて隣接する頂点同士が同じ色に彩色されないような彩色パターンを探索によって発見する問題である。特に、色数を 3 色とした問題は 3COL と呼ばれ、アルゴリズムを評価するためのベンチマークとしてしばしば用いられる [Mizuno 11, Tayarani-N 15]。ここで、無向グラフの頂点数を n 、辺数を c とした際、制約密度 d を $d = c/n$ と定義する。3COL において、 $d = 2, 3 \sim 2.4$ の領域は相転移領域と呼ばれ、難しい問題が集中していることがわかっている [Hogg 96]。

2.2 蟻コロニー最適化

蟻コロニー最適化 (ACO) は、蟻の採餌行動におけるフェロモンコミュニケーションをモデル化したメタヒューリスティクスである [Dorigo 96, Dorigo 99]。現実世界における蟻は餌を見つけると、フェロモンを落としながら巣まで戻る。蟻にはこのフェロモンに誘引される性質がある。フェロモンの経路は蒸発や他の蟻による補強を繰り返し、徐々に経路の長さは短くなっていく。この過程から着想を得て、最適化問題の解法として提案されたものが ACO である。ACO をもとにしたアルゴリズムは、多くの組合せ最適化問題をはじめ、3COL を含む多くの CSP にも有効な手法であることがわかっている [Bui 08]。

2.3 cunning Ant System

ACO アルゴリズムの 1 つである cunning Ant System (cAS) [Tsutsui 06] のアルゴリズムを図 1 に示す。ACO アルゴリズムでは、1 匹の蟻が 1 つの解候補を持つ。複数の蟻が複数世代にわたって解候補を生成していくことで、探索が進められていく。解候補生成の手順としては、まず値 (COL における色) を割当てる変数 (COL における無向グラフの頂点) をランダムに 1 つ選択する。次に選択された変数に対して、フェロモンと制約違反から決定される確率に基づいて、割当てる値を選択する。以下に、解候補 A の変数 x_j に割当てられる値として v が選択される確率 $p_A(< x_j, v >)$ を示す。

連絡先: 増金拓弥, 拓殖大学大学院工学研究科情報・デザイン工学専攻, 東京都八王子市館町 815-1, 042-665-0519, phsl.masukane@gmail.com

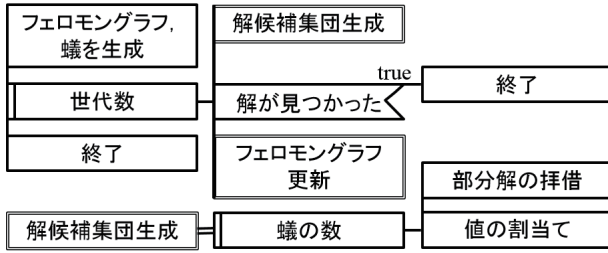


図 1: cAS のアルゴリズム

$$p_A(< x_j, v >) = \frac{[\tau_A(< x_j, v >)]^\alpha [\eta_A(< x_j, v >)]^\beta}{\sum_{w \in D_j} [\tau_A(< x_j, w >)]^\alpha [\eta_A(< x_j, w >)]^\beta}, \quad (1)$$

$$\tau_A(< x_j, v >) = \sum_{< x_k, u > \in A} \tau(< x_k, u >, < x_j, v >),$$

$$\eta_A(< x_j, v >) = \frac{1}{1 + \text{conf}(\{< x_j, v >\} \cup A) - \text{conf}(A)},$$

ただし, D_j は変数 x_j に割当て可能な値の集合 (3COL の場合は *red, green, blue*) を, $\tau(< x_k, u >, < x_j, v >)$ は “値 u を割り当てた変数 x_k ” と “値 v を割り当てた変数 x_j ” 間に溜まっているフェロモンの量を, $\text{conf}(A)$ は解候補 A の制約違反数をそれぞれ表している。また, α と β はそれぞれフェロモンと制約違反を重視する重みを表している。これを, 全ての変数に値が割当てられるまで繰り返すことで, 1 つの解候補が生成される。各蟻が解候補を生成した後, その集団内で最も評価が高い解候補を用いて, フェロモングラフが更新される。以下に $\tau(i, j)$ の更新式を示す。

$$\tau(i, j) = (1 - \rho) \times \tau(i, j) + \Delta\tau \quad (2)$$

$$\Delta\tau = \begin{cases} \frac{1}{\text{conf}(A_{\text{best}})}, & (i, j) \in A_{\text{best}} \\ 0, & \text{otherwise} \end{cases}$$

ただし, ρ はフェロモンの蒸発率, A_{best} は集団内で最も評価の高い解候補を表している。探索の過程で制約違反のない解候補が発見されれば探索成功となる。逆に, 指定した世代数で制約違反のない解候補が発見できなければ探索失敗となる。

ACO アルゴリズムでは解候補を生成する際, 全ての変数に対して値を割当てるために確率を算出する。そのため, 探索に時間がかかってしまうという欠点が挙げられる。このような欠点を改良するために提案された ACO アルゴリズムが cAS である。cAS では, 解候補を生成する前に, 前の世代の蟻 (donor ant) から解候補の一部を “拝借” する蟻 (cunning ant) を用いる。拝借された変数については確率計算が行なわれないため, cAS は探索にかかる時間を短縮できる。

3. 提案手法

3.1 基本方針

従来の ACO アルゴリズムは, 制約違反とフェロモングラフ (評価が高い解候補の情報) をもとに探索を行なう。この 2 つの情報に加え, 他の情報を ACO に取り入れることで, ACO を用いた探索をより効率的なものにすることができるのではないかと考えた。そこで我々は, 探索中に生成される “評価が

低い解候補” に着目した。解候補集団が評価が高い解候補に誘引されると同時に, 評価が低い解候補から遠ざかることで, 探索がより効率的に行なわれるということが期待される。

本研究では, 従来の (評価が高い解候補を用いて更新される) フェロモングラフに加えて, 評価が低い解候補を用いて更新されるフェロモングラフ (本研究ではこのフェロモングラフを “ネガティブフェロモングラフ” と呼ぶ) を探索に用いる ACO モデルを提案する。本提案モデルの基本方針は以下の 2 点である。

- 各変数に割当てる値を選択する際には, 従来の ACO モデルと同様に, 制約違反と通常のフェロモングラフのみを参考にする。
- 通常のフェロモングラフを更新する際には, 集団内で最も評価が高い解候補だけでなく, ネガティブフェロモングラフも参考にする。

3.2 アルゴリズム

本研究では, 2.3 節で述べた cAS に提案モデルを適用したアルゴリズムを提案アルゴリズムとし, ADUPT (Ant with DUal Pheromone Trails) と呼ぶ。ADUPT の手続きが図 1 に示した cAS の手続きと異なる点は, “ネガティブフェロモングラフの更新” と “通常のフェロモングラフの更新” である。

まず, ネガティブフェロモングラフの更新について説明する。本研究で用いるネガティブフェロモングラフは, 通常のフェロモングラフと同じ構造である。また, ネガティブフェロモングラフは集団内で最も評価が低い解候補に基づいて更新される。以下にネガティブフェロモングラフの更新式を示す。

$$N\tau(i, j) = (1 - \rho) \times N\tau(i, j) + \Delta N\tau \quad (3)$$

$$\Delta N\tau = \begin{cases} \text{conf}(A_{\text{worst}}), & (i, j) \in A_{\text{worst}} \\ 0, & \text{otherwise} \end{cases}$$

ただし $N\tau(i, j)$ は, (i, j) に溜まっているネガティブフェロモンの量を, A_{worst} は解候補集団内で最も評価の低い解候補を表している。通常のフェロモングラフの更新 (式 2) と比較すると, 最も評価が低い解候補を用いることの他に, “フェロモンの蓄積量” が異なっている。従来のフェロモンの蓄積量が解候補の制約違反数の逆数であったのに対して, ネガティブフェロモンの蓄積量は解候補の制約違反数そのものとする。これは, 評価が低い解候補情報を蓄積するためのネガティブフェロモンにおいて, 制約違反数が多い解候補ほど参考とする価値が高いためである。

次に, 通常のフェロモングラフの更新について説明する。従来の ACO アルゴリズムでは, 最も評価が高い解候補に含まれる全ての値の組合せについて, フェロモンの蓄積量は一定であった (式 2)。本研究では, ネガティブフェロモンが多く溜まっている, つまり, 低品質な割当てについて, 通常のフェロモンの蓄積量を小さくする。これにより, 低品質な割当てが確率的に生成されづらくなると考えられる。以下に通常のフェロモンの更新式を示す。

$$\tau(i, j) = (1 - \rho) \times \tau(i, j) + \Delta\tau \quad (4)$$

$$\Delta\tau = \begin{cases} \frac{1}{\text{conf}(A_{\text{best}})} \times w(i, j), & (i, j) \in A_{\text{best}} \\ 0, & \text{otherwise} \end{cases}$$

ここで, $w(i, j)$ はネガティブフェロモン量 $N\tau(i, j)$ によって決定される重み係数である。ただし $w(i, j)$ の値は, 対応するネガティブフェロモンの量が上限値に近いほど 0 に近い値をとり, 下限値に近いほど 1 に近い値をとるものとする。

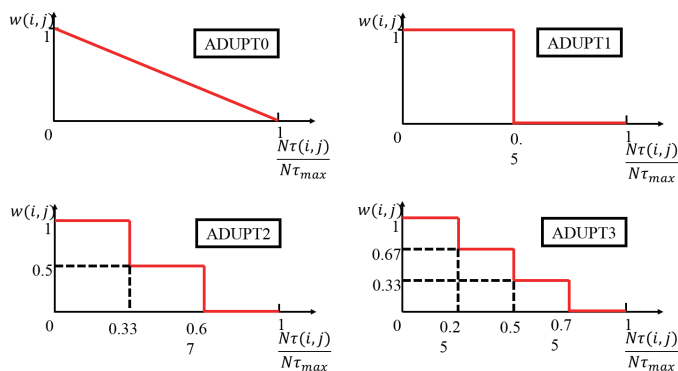
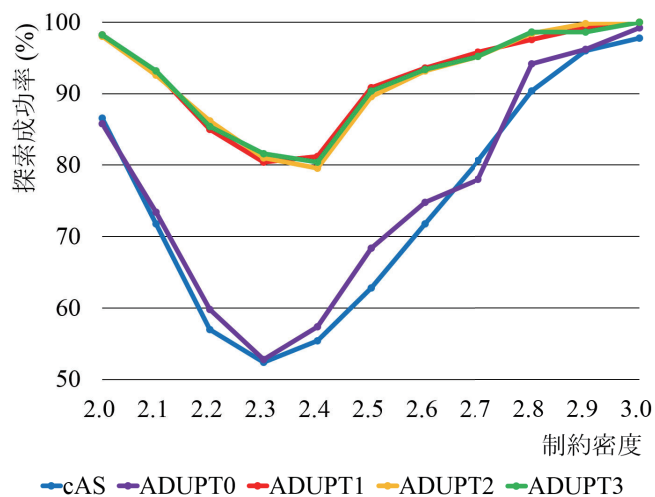
図 2: 重み $w(i, j)$ を決定するための関数

図 3: 各制約密度における探索成功率

4. 評価実験

4.1 実験条件

提案アルゴリズムである ADUPT の有効性を示すために、評価実験を行なった。ここでは、ADUPT と従来手法である cAS を比較する。本実験では頂点数 $n = 100$ である 3COL インスタンスを、制約密度 $d = 2.0 \sim 3.0$ の範囲で 0.1 毎に 11 のケースに対して、それぞれランダムに 100 問生成した。ここで生成した問題は全て解が存在する。生成した各問題について、それぞれ 5 回ずつ試行を行なった。本実験のパラメータは、蟻の数を 100、蟻の最大世代数を 2000、フェロモンの重み $\alpha = 5.0$ 、制約違反の重み $\beta = 10.0$ 、フェロモンの蒸発率 $\rho = 0.01$ とした。また、 $w(i, j)$ を決定するために、図 2 に示す 4 パターンの関数を用いる。それぞれの関数を用いた ADUPT をそれぞれ、ADUPT0 ~ ADUPT3 とする。これらの手法について、以下の 2 点を比較した。

- 探索成功率: 全体の試行数に占める制約充足解を発見できた試行数
- 探索コスト: 制約充足解を発見するまでに生成した解候補の数

なお、実験には PC/AT 互換機 (CPU: Intel Core i7 880 3.07GHz, RAM: 4GByte) を使用し、プログラムはすべて Java 言語で記述した。

4.2 実験結果

実験の結果を図 3 と図 4 に示す。図 3 は各手法の探索成功率をインスタンスの制約密度ごとに示したものである。図 3 から、提案手法である ADUPT0 の探索成功率は、従来手法である cAS の探索成功率と比較すると、差が小さい。一方で、ADUPT1, ADUPT2, ADUPT3 の探索成功率は、cAS よりも高くなっている。また、その探索成功率の差は、制約密度が相転移領域 ($d = 2.3 \sim 2.4$) に近づくにつれて大きくなっている。特に、 $d = 2.3$ のインスタンスについては、探索成功率の差が約 30% であり、最大となっている。

次に、図 4 は全ての試行における探索コストの平均を示している。ただし、探索に失敗した (制約充足解を発見できなかった) 試行については、蟻の数 $\times$ 世代数 $= 200,000$ の解候補を生成しているため、探索コストは 200,000 とした。図 4 よ

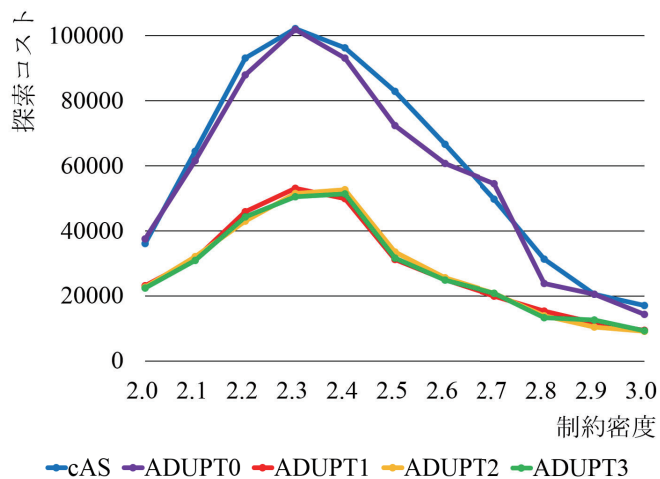


図 4: 各制約密度における探索コスト

り、ADUPT0 の探索コストは cAS とほとんど差がない。また、ADUPT1, ADUPT2, ADUPT3 は、cAS よりも小さい探索コストで解を発見できている。この探索コストの差は、相転移領域に近づくにつれて大きくなっている。特に、 $d = 2.3$ のインスタンスについては、ADUPT1, ADUPT2, ADUPT3 は cAS の約半分の探索コストで解を発見できている。ここで、今回の結果では、探索に失敗した試行における探索コストを 200,000 としたが、実際は 200,000 のコストをかけても解を発見できていない。つまり、解が見つかるまで探索を続けると、探索のコストの差は更に広がっていくと思われる。

4.3 考察

4.2 節より、ADUPT1, ADUPT2, ADUPT3 の 3 つの手法は、cAS よりも高い探索成功率を得られた。一方で、ADUPT0 は探索成功率と探索コストのどちらについても、cAS とほとんど差がなかった。本節ではこの実験結果について考察する。提案モデルでは、従来の ACO アルゴリズムにおけるフェロモンの蓄積量に変化を与えるものである。そのため、ここでは“フェロモンの溜まり方”に着目する。

図 5 は、探索に成功した 1 つの試行について調査したものである。縦軸はその試行で発見できた解の割当てに対応する

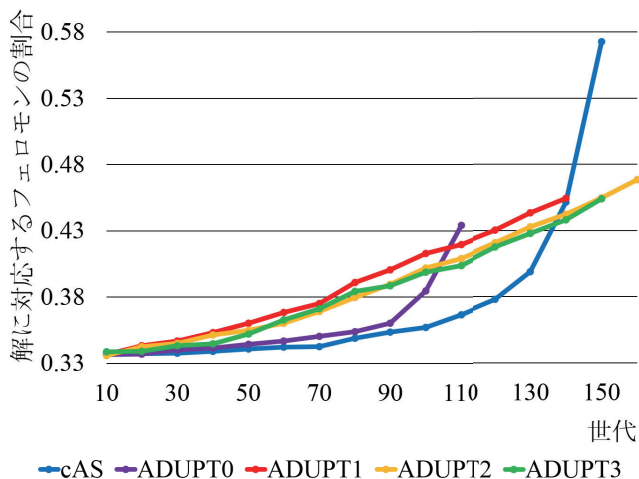


図 5: 解に対応するフェロモン量の割合の推移

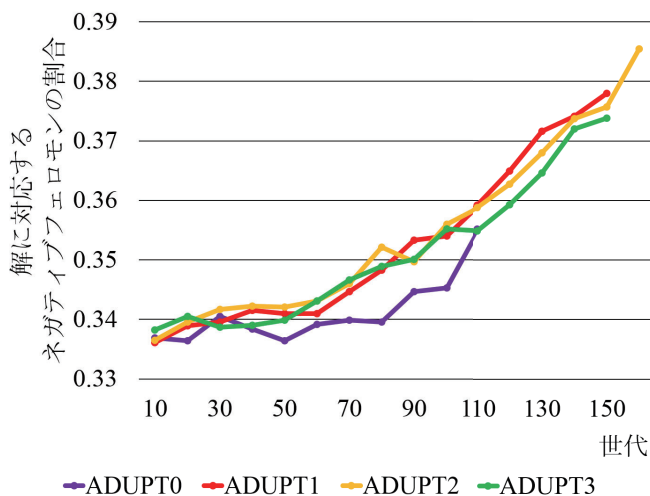


図 6: 解に対応するネガティブフェロモン量の割合の推移

フェロモンの量が、全ての割当てに対応するフェロモンの量に占める割合を、横軸は世代数を表している。図 5 から、cAS と ADUPT0 はおよそ 90 世代まで、解に対応するフェロモンの割合にほとんど変化が見られない。このことから、cAS と ADUPT0 についての実験結果がほとんど同じものとなったのは、cAS と ADUPT0 のフェロモンの溜まり方が似ているためではないかと考えられる。一方で、ADUPT1, ADUPT2, ADUPT3 は、解に対応するフェロモンの割合が徐々に増加していることがわかる。つまり、ADUPT1, ADUPT2, ADUPT3 は、後に発見される解に近い解候補が生成されやすくなっていく。このことが、探索成功率の大きな向上につながっているのではないかと考えられる。

図 6 は、図 5 と同じ試行について調査したものである。ただし、縦軸はその試行で発見できた解の割当てに対応する“ネガティブフェロモン”の量が、全ての割当てに対応するネガティブフェロモンの量に占める割合である。前述した通り、ネガティブフェロモンは集団内で最も評価が低い解候補に基づいて更新される。そのため、解に対応するネガティブフェロモン量の割合が上昇しているということは、最も評価が低い解候補が、後に発見される解に対応する割当てを多く含んでいるとい

うことであると思われる。つまり、集団内の解候補全体が、解に近づいていると考えられる。解に対応するネガティブフェロモン量の割合について、ADUPT0 がおよそ 90 世代までほとんど変化がないのに対して、ADUPT1, ADUPT2, ADUPT3 は徐々に上昇している。この結果から、ADUPT1, ADUPT2, ADUPT3 は集団全体が解に近づいていると思われる。結果として、ADUPT1, ADUPT2, ADUPT3 は解をより発見しやすくなっているのではないかと考えられる。

5. おわりに

本研究では、大規模な CSP を効率よく解くための手法として、複数種類のフェロモンを用いる ACO モデルを提案した。本モデルは、従来の ACO アルゴリズムで用いられるフェロモンを更新する際に、解候補集団内で最も評価が低い解候補に基づいて更新される新たなフェロモンを用いるものである。提案モデルを cAS という ACO アルゴリズムに適用し、単一のフェロモンのみを用いる cAS と実験的に比較を行なった。実験の結果から、提案モデルの有効性を示した。今後の課題として、他のメタヒューリスティクスとの比較や、他の CSP への適用が挙げられる。

参考文献

- [Bui 08] Bui, T. N., Nguyen, T. H., Petal, C. M. and Phan, K. T.: An ant-based algorithm for coloring graphs, *Discrete Applied Mathematics*, Vol. 156, pp. 190–200 (2008).
- [Dorigo 96] Dorigo, M., et. al.: The Ant System: Optimization by a Colony of Cooperating Agents, *IEEE Transaction on Systems, Man, and Cybernetics - Part B*, Vol. 26, pp. 26–41 (1996).
- [Dorigo 99] Dorigo, M. and Di Caro, G.: The Ant Colony Optimization Meta-Heuristics, *New Ideas in Optimization*, pp. 11–32 (1999).
- [Hogg 96] Hogg, T., Huberman, B. A. and Williams, C. P.: Phase transition and search problem, *Artificial Intelligence*, Vol. 81, pp. 1–16 (1996).
- [Mizuno 11] Mizuno, K., Hayakawa, D., Sasaki, H., and Nishihara, S.: Solving Constraint Satisfaction Problems by ACO with Cunning Ants, *The 2011 Conf. on Technologies and Applications of Artificial Intelligence (TAAI2011)* (2011).
- [Tayarani-N 15] Tayarani-N, M. H. and Prugel-Bennett, A.: Anatomy of the fitness landscape for dense graph-colouring problem, *Swarm and Evolutionary Computation*, Vol. 22, pp. 47–65 (2015).
- [Tsutsui 06] Tsutsui, S.: cAS: Ant Colony Optimization with Cunning Ants, *Proc. of the 9th Int. Conf. on Parallel Problem Solving from Nature (PPSN IX)*, pp. 162–171 (2006).