ハイブリッド型分散制約充足アルゴリズムの開発

安藤 雅彦 † 能登 正人 † 豊嶋 久道 †

神奈川大学 工学部 電気電子情報工学科 ††

1. はじめに

制約充足問題 (Constraint Satisfaction Problem: CSP) とは人工知能分野における様々な問題を定式化することができる一般的な枠組みである。しかしながら、より現実的な問題を表現しようとする場合には、いくつかの制限があり、必ずしも CSP の記述力は十分とは言えない。そのため、通常の CSP の枠組みを分散環境や動的環境などにも適応できるように拡張した研究が盛んに行われている。たとえば複数の自律した計算プロセス (エージェント) による種々なマルチエージェントシステムなどの応用問題も、分散 CSP として CSP を拡張することにより表現可能となる。このような現実的な分散環境下にある問題にも対処できるように、CSP の部分クラスである分散不完全 CSP を一部具体化した分散最大 CSP(Distributed Maximal CSP: DMCSP) という問題クラスがある [2]。文献 [1] では厳密解法として同期型分枝限定法、近似解法として反復分散ブレイクアウト法、双方のアルゴリズムを合わせて用いることにより、DMCSP の最適解を既存の厳密解法より高速に求めている。

本稿ではこの文献のアルゴリズムを考慮し近似解法から厳密解法へと切り替えるポイントについて焦点を当て、より高速なハイブリッド型のアルゴリズムを提案する。実験では、過制約な状態での分散グラフ色塗り問題を用いて、本手法の有効性を示す。

2. 提案アルゴリズム

DMCSP の解法は他の人工知能分野と同様に厳密解法と近似解法があるが、適応する問題クラスにより秀逸性が異なる。そこで、本研究では厳密解法である同期型分枝限定法と近似解法である反復分散ブレイクアウト法を組み合わせた解法により解を求める [2]。

探索規模が大きい問題に対して従来の厳密解法を単独で実施すると、アルゴリズムの完全性は保証されるが、実時間内に最適解が求まらない可能性が生じ、計算量は増大する。対照的に近似解法は分散ブレイクアウト法の利点がそのまま生かされるため、探索空間が膨大であっても短時間で準最適解を求めることが可能であるが、DMCSP の最適解を得たとしてもそれを最適解だと判定できないことがある。本研究においては、DMCSP の最適解を求めることを大前提としているため、アルゴリズムは同期型分枝限定法を用いなければ意味をなさない。そこで、探索空間が膨大であっても実行時間内に解を求めるためにはいかにして探索空間を狭めるかが最大の重要点となり、問題空間の縮小が可能であれば、自ずと計算量は限定的になる。

提案アルゴリズムの概略を以下に示す。

(1) 問題クラスの制約グラフの最大次数を、任意のエージェント i の必要値 N_i に代入し、十分値 S_i は 0 として反復分散ブレイクアウト法を実施する。

(2) 各エージェントが隣接しているエージェントに対して自分の情報をメッセージとして送信する。受信したエージェントは反復ブレイクアウト法に基づき自分の変数の値を決定し、自分の情報を送り返す。

(3) 1 サイクル経過後、各エージェントの制約違反数の最大値 n_g が減少したときには $counter = 0$ とし、減少が無いときには $counter$ を 1 つ上げる。

(4) $counter$ がある一定の値未満なら (2) の処理を行い、一定の値以上になったら代表となるエージェントは現時点の n_g を保持し、各エージェントに反復分散ブレイクアウト法を終了するようにメッセージを送信する。

(5) 同期型分枝限定法の必要値 N の初期値 N_0 に n_g を設定し、最終的な最適解を得る。

本研究では、同期型分枝限定法の前処理として反復分散ブレイクアウト法を第一手法で一定時間実行する。得られた準最適解の制約違反数を必要値 N の初期値として第二手法である同期型分枝限定法を実行し、最終的な最適解を得るという方法を提案する。なお、概略 (4) で記述してある**一定の値**とは、すなわち双方のアルゴリズムを切り替えるポイントを指している。

3. 切り替えポイント

3.1 切り替えポイントの重要性

近似解法である反復分散ブレイクアウト法から、厳密解法である同期型分枝限定法へアルゴリズムを切り替えるポイントが早過ぎると、反復分散ブレイクアウト法が十分に施行される前に同期型分枝限定法を開始することとなり、良い上界値が得られない。そのため、同期型分枝限定法においての有効的な枝刈りができず、広い範囲で同期型分枝限定法を行うことになる。また、反対に切り替えるポイントを誤って遅くすると、反復分散ブレイクアウト法を単独で実施する時間が長くなり、アルゴリズムを切り替えるまでに時間がかかる。どちらの場合においても、最終的にアルゴリズムを終了したときの実施時間が増し、高速性は得られない。

3.2 切り替えポイントの特定化

双方のアルゴリズムを切り替えるポイントを特定化するために、第一手法である反復分散ブレイクアウト法を単独でグラフ色塗り問題に適応した。

各エージェントのとりうる色の数 k を 3、リンクの個数 $m = n \times 2$ 本、$m = n \times 2.7$ 本、$m = n \times (n-1)/4$ 本の場合についてそれぞれ特定の問題 (エージェント数 $n = 10, 30, 50, 70, 100, 150, 200$) に関する評価を行った。なお、各々の問題クラスにおいてエージェント同士を結び、必要情報を送受信するリンクを定義した隣接行列が異なる問題を 5 種類用意し、それぞれのインスタンスについて初期値を変えて 100 回シミュレーションを行った。これら m のパラメータ設定は制約密度が疎な問題、中間的な難しい問題、密な問題である場合にそれぞれ相当する [3]。反復分散ブレイクアウト法

Development of Hybrid Distributed Constraint Satisfaction Algorithm

†Masahiko Ando, Masato Noto and Hisamichi Toyoshima

††Department of Electrical, Electronics and Information Engineering, Kanagawa University

表 1: エージェント間制約が “中間的” な問題についての評価

n	10		30		50		70		100	
	cycle	time	cycle	time	cycle	time	cycle	time	cycle	time
A	174.1	58.7	379.4	1107.0	453.5	5365.2	589.3	14855.1	636.1	31913.3
B	180.0	86.8	450.5	1470.3	547.9	6292.2	674.3	17744.3	779.4	49054.3

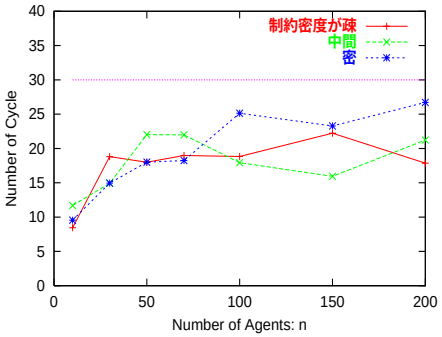

図 1: エージェント数と vc の関係

で得られる最適な制約違反数を pv と仮定すると、ある時点での制約違反数 (nv) が pv になる一つ前までの nv において、各エージェントが反復分散ブレイクアウト法を実施し、他のエージェントと通信した回数で最も多くなったサイクル数 (vc) を求め、そのサイクル数以上通信を行った場合には直ちにアルゴリズムを終了し、厳密解法を開始すれば双方のアルゴリズムを最も効率よく使用できることになる。反復分散ブレイクアウト法を先に述べたパラメータ設定で実行したときの最大通信回数を図 1 に示す。なお、エージェント数 n を横軸に、隣接行列が異なる 5 つの問題に 100 回試行し、制約違反数が減少するまでの最大通信サイクル数の最大値を縦軸にとり、プロットしてある。

図 1 より、特に n の値によらずほぼ一定の割合の値をとっていることがわかる。どの問題クラスにおいても最大通信回数が 30 サイクル以上の値にならないため、本研究においては初期の評価用のアルゴリズムとして 30 サイクルを切り替えポイントとして設定した。本研究ではシミュレーションしていないエージェント数 (n > 200) について考えると、vc が 30 サイクルで妥当かどうかは不明であるが、反復分散ブレイクアウト法の性質を考慮すると、そのようなエージェント数の場合については同期型分枝限定法が枝狩りを行い、探索空間を狭める 1 サイクル数と、反復分散ブレイクアウト法が施行されて得た N_0 の減少率を考慮しても、本手法の方が明らかな有効性が得られることが予想できる。

4. 実験結果と考察

本研究の目標は最適解を得ることにあるので、本手法と厳密解法である同期型分枝限定法との比較実験を行った。なお、シミュレーション評価には 3 章のシミュレーションで使用した分散グラフ色塗り問題を用いる。$k = 3$ として $n = 10, 30, 50, 70, 100$ に対して双方の手法を適応して最適解を得るまでのサイクル数と、アルゴリズムの実行時間を求める。変数間の制約の密度が疎である問題に対して、制約違反最少ヒューリスティックが効果的でないことが示されているため、リンク数 m の値はエージェント間制約が中間的なクリティカルな問題 ($m = n \times 2.7$) にのみ 100 回試行し、最適解を得た時のサイクル数、実行時間のそれぞれの平均を表 1 に示す。なお、制約密度が中間的な問題は最も難しいとされている問題クラスである [3]。表 1 において、マトリックスの列の割合はそれぞれの n におけるサイクル数と実行時間 (ms) を示し、‘A’ は提案手法を、‘B’ は同期型分枝限定法を示している。

まず、表 1 から明らかなように、提案した手法の方が問題クラスの大きさに関係なく、少ないサイクル数、実行時間で最適解を得ている。そのため、全体として比較的難しい問題クラスに対して、改善・向上が得られたと言える。これと同様のことは、この例のみならず、その他 4 つのランダムバイナリ問題の例についても観察された。同期型分枝限定法と提案するハイブリッド型の解法のそれぞれが最適解に達するまでを比較してみると、ほとんどすべての場合において提案する手法の方が早くそれに達することが確認された。また、最終的には同期型分枝限定法と同等の nv 値をとっており、提案した手法の最終的な値の精度は落ちていないことも分かる。

5. おわりに

本研究では、近似解法が準最適解を非常に速い段階で求められることに着目して、厳密解法の前処理として近似解法を実行させることにより探索空間を狭め探索効率の向上が可能であることを実験により示した。

しかしながら、本手法では同期型分枝限定法の探索空間を狭めることにより探索時間の縮小化を行っているため、すべての問題クラスに有効性が得られるということは考えにくい。つまり、探索空間の限定化に用いられた時間よりも反復分散ブレイクアウト法を実施している時間が長い場合には本手法は適さないと考えられる。本研究でのシミュレーション結果においては k=3、m=2.7、n=10、のクラスで、ランダム発生させた問題の一つにおいては有効性は得られなかった。これは先に述べた原因が考えられ、このような問題においてはユーザが適応するアルゴリズムを考慮する必要性があるのは避けられないことである。さらには、今回は DMCSP に焦点を当て、過制約な分散 CSP を取り扱ったが、DMCSP のみですべての過制約な問題をサポートできるわけではないため、さらに別の部分クラスについての検討、考察が今後の課題であると考えられる。

参考文献

[1] 安藤雅彦, 能登正人, 豊嶋久道: 分散最大制約充足アルゴリズムの分散グラフ色塗り問題による評価, 情報技術レターズ, Vol. 2, pp. 129–130 (2003).

[2] 平山勝敏, 横尾真: 分散不完全制約充足問題, 人工知能学会誌, Vol. 14, No. 4, pp. 636–645 (1999).

[3] S. Minton, D. D. Johnston, A. B. Philips and P. Laird: Minimizing Conflicts: A Heuristic Repair Method for Constraint Satisfaction and Scheduling Problems, *Artificial Intelligence*, Vol. 58, No.1–3, pp.161–205 (1992).