

制約充足問題 (CSP) に対するタブー探索における プログラムパラメータの自動調節

02401594 京都大学 *野々部 宏司 NONOBE Koji
01001374 京都大学 茨木 俊秀 IBARAKI Toshihide

1 はじめに

近年、組合せ問題に対して多くの発見的手法が提案されてきたが、個々の問題に対してその問題専用のアルゴリズムを構築することは必ずしも容易なことではない。そこで我々は、多くの組合せ問題に対して手軽に利用することができ、しかも、どの問題に対しても満足のいく良質の近似解を得ることができるような、いわゆる“汎用アルゴリズム”の開発を行ってきた [3]。そのために、多くの組合せ問題を定式化することができる制約充足問題 (Constraint Satisfaction Problem, CSP) [4] に着目し、CSP に対するタブー探索 [2] を基本的枠組みとした近似アルゴリズムを構築し、その有用性と将来性を確かめるため、グラフの彩色問題等のベンチマーク問題の他、時間割問題やスケジューリング問題といった現実問題に対しても計算実験を行ってきた。その結果、これまでの研究でも指摘されてきたことではあるが、*tabu tenure* と呼ばれるプログラムパラメータが、計算結果に大きな影響を及ぼすため、その調整に多大な時間を要することが分かってきた。汎用アルゴリズムを目指す上で、この調節手間を軽減することは最も重要な課題の 1 つである。そこで本報告では、タブー探索中、探索状況をもとに *tabu tenure* を自動的に調節する方法を提案し、その有効性を確かめる。

2 定義

CSP は、それぞれ有限離散領域 D_i を持つ n 個の変数 X_i ($i = 1, 2, \dots, n$) と、 m 個の制約 $C_l(X_{l_1}, X_{l_2}, \dots, X_{l_{t_l}})$ ($l = 1, 2, \dots, m$) で定義され、全ての制約を満たすように、各変数 X_i に値 $j \in D_i$ を割り当てる問題である。ここで、各制約 C_l は変数 $X_{l_1}, X_{l_2}, \dots, X_{l_{t_l}}$ に対する t_l -項制約であり、それらの変数が同時にとることのできる値の組全てから成る直積 $D_{l_1} \times D_{l_2} \times \dots \times D_{l_{t_l}}$ の部分集合である。全ての制約を満たすような値の割り当てを実行可能解と呼ぶ。

ここで、変数 X_i とその値 j ($\in D_i$) の組それぞれに対して、値変数 x_{ij} を

$$x_{ij} = \begin{cases} 1, & \text{変数 } X_i \text{ が値 } j \text{ をとる,} \\ 0, & \text{その他,} \end{cases}$$

と定義し、割り当てを $\sum_{i=1}^n |D_i|$ 次元の 0-1 ベクトル $\mathbf{x} = (x_{ij} \mid i = 1, 2, \dots, n, j \in D_i)$ で表す。このとき、

$$\sum_{j \in D_i} x_{ij} = 1, \quad i = 1, 2, \dots, n, \quad (1)$$

が成立すれば、全ての変数 X_i に対して値が 1 つ割り当てられていることになる。なお、各制約 C_l の表現方法は一意ではなく、線形不等式、論理関数、値の組の集合等、問題に応じて適当な表現方法を用いることができる。

3 タブー探索の適用

タブー探索中、式 (1) を満たす割り当てから成る探索空間 X^* を考える。また、 $\mathbf{x} \in X^*$ の近傍 $N(\mathbf{x})$ を、ある 1 つの変数 X_i の値 j を他の値 j' に変えることによってできる割り当ての全てとする。また、各制約 C_l に対して、それが満たされるならば 0、満たされないならば正の値をとるようなペナルティ関数 $p_l(\mathbf{x})$ を適当に定義し、全体としてのペナルティ関数 $p(\mathbf{x}) = \sum_l p_l(\mathbf{x})$ を考えることで、本来決定問題である CSP を最小化問題

$$\begin{aligned} & \text{minimize } p(\mathbf{x}) \\ & \text{subject to } \mathbf{x} \in X^*, \end{aligned} \quad (2)$$

として扱う。すなわち、ペナルティ関数値が 0 であるような解 $\mathbf{x} \in X^*$ が存在することと、CSP が実行可能解を持つことは同値である。

また、タブー探索では、局所探索において改悪解への移動を許しているため、通常、解の移動の属性をタブーリストと呼ばれるリストに記憶しておき、ある期間 (*tabu tenure* と呼ばれる) その逆向きの移動を禁止するという方法がとられる。タブーリストに置くべき属性としては、値が 0 から 1 に変わった値変数 x_{ij} と値が割り当て直された変数 X_i との 2 つが考えられる。前者の方が解の移動を忠実に表現しているとも言えるが、これまでの計算結果では、後者の方が良い結果を出していることが多いため、以下では値が割り当て直された変数 X_i を用いる。

4 Tabu tenure の自動調節

探索中 tabu tenure を自動調節する方法の1つとして、過去に探索した解を記憶しておき、サイクリングが生じたら tabu tenure を増加させ、逆にサイクリングがある程度の期間生じなかつたら減少させるという、Reactive tabu search [1] があり、幾つかの組合せ問題に対してその有用性が示されている。しかし、この方法では過去に探索した解を再び探索しない限りサイクリングとは見なされないため、大規模な問題において、サイクリングではないが、解空間のある一部を克明に調べてしまうような探索過程を効果的に防ぐことはできない。そこで本報告では、Reactive tabu search とは別の tabu tenure の自動調節方法を提案する。以下にその概略を述べる。

4.1 Tabu tenure の増加

まず、tabu tenure をどのような場合に増加させるかを考える。第 k 反復において、解 $x^{(k-1)}$ から $x^{(k)}$ に移動する際に値を割当て直した変数を $X^{(k)}$ とし、第 k 反復までに値を割当て直した変数集合を $A^{(k)}$ とする。すなわち、 $A^{(0)} := \phi$, $A^{(k)} := A^{(k-1)} \cup \{X^{(k)}\}$ とする。このとき、ある反復 t において、変数 $X^{(t)}$ が t 以前に第 $t' (< t)$ 反復で値が割当て直されており、しかも $A^{(t')} = A^{(t)}$ であったとすると、この間の探索において、限られた変数集合 $A^{(t)}$ の値のみを変化させているだけであり、探索が偏っていると考えられる。そこで、そのような場合には多様化が必要であると判断し、tabu tenure を1増加させる。但しこのままでは、 $A^{(k)}$ が単調増加であるため、ある程度反復が進むと毎回 tabu tenure が増加してしまうという結果になるので、それを防ぐため、暫定値が更新されたり、多様化が行われたと判断されたときには、 $A^{(k)} := \phi$ として過去の情報を消去する。

多様化が行われたと判断する基準としては、tabu tenure の増加(多様化)に伴って初めて値が割当て直された変数で、しかもその際、解の移動が改悪であった(ペナルティ関数値が増えた)ような変数 X' に着目し、変数 X' が、タブーリストから出た直後に値を割当て直されなかった場合、多様化が行われたとする。すなわち、初めに変数 X' の値が割当て直されたときは改悪であったので、その直後は、変数 X' の値を元に戻せばペナルティ関数値を減らすことができる。しかし、しばらくの間、タブーリストの働きによりその移動は禁止されるわけであり、もし変数 X' がタブーリストから出た直後に値を元に戻されなかったとしたら、その間に解は充分変化したと考えることができ、多様化が行われたと判断するわけである。

以上の操作により、解空間のある一部分に限定された探索を察知し、tabu tenure を増加させることによってそれを解消することができると思

られる。

4.2 Tabu tenure の減少

次に tabu tenure の減少について考える。タブー探索では、ある解の移動がタブーリストにより禁止されていたとしても、*aspiration criteria* と呼ばれるある基準を満たせば、その移動を許すという手法が通常用いられる。その基準としては、“これまでの暫定値が更新される場合”、とすることが多い。ここでは、その基準をサイクリングが起こらない程度に緩めると共に、探索中、タブーリストにより禁止されている移動が *aspiration criteria* を満たすことによって実行された場合には、tabu tenure が長すぎると判断し、それを1減少させるものとする。

aspiration criteria としては、タブーリストはサイクリングを防止する役目も担っていることを考慮すると、サイクリングが生じる程に緩くしてしまうのは好ましくない。そこで本実験では、ある変数がタブーリストに入るとき解の移動が改善であり、しかも、今その変数の値を割当て直すことで、タブーリストに入ったときと比べてペナルティ関数値がさらに小さくなるような場合に、*aspiration* として認めるという方法をとった。

この方法により、tabu tenure が適正值より大きい場合でも、タブーリストによる、効果的な移動の過剰な禁止を防ぐことができ、しかも tabu tenure の調整ができることが、実験により確認された。

5 おわりに

上述の方法を用いることで、パラメータ調節という手間を省くことができるかどうか、探索中の tabu tenure の値や得られた解の質等を、tabu tenure を固定した場合と比較することで評価した。なお、詳しい計算結果は当日発表させて頂く。

参考文献

- [1] Battiti, R., and Tecchiolli, G., “The reactive tabu search”, *ORSA Journal on Computing* 6 (1994) 126-140.
- [2] Glover, F., “Tabu search fundamentals and uses”, Technical Report, University of Colorado (1995).
- [3] Nonobe, K. and Ibaraki, T., “Tabu search Approach to CSP (Constraint Satisfaction Problem) as a general problem solver”, Technical Report, Kyoto University (1996).
- [4] Tsang, E., *Foundations of constraint satisfaction*, Academic Press, London, 1993.