1999年度日本オペレーションズ・リサーチ学会
春季研究発表会

制約充足問題に対するタブー探索における評価関数の重みの自動調整

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

制約充足問題 (Constraint Satisfaction Problem, CSP) は多くの組合せ問題を定式化できることが知られている. 我々はこれまでに, CSP の枠組みに基づいて, 様々な組合せ問題に対する汎用アルゴリズムの開発を行ってきた [3]. そのアルゴリズムには, メタ戦略と呼ばれる手法の一つであるタブー探索法を用いてきた. タブー探索法は, 局所探索法を基本とした近似解法であり, その基本要素である探索空間, 近傍, 解の評価関数などは, アルゴリズムの性能を大きく左右するため, 慎重に定義しなくてはならない. 本研究では特に, 解の評価関数について検討する.

本研究のタブー探索では, 実行不可能解の探索も利用するため, 実行可能領域と実行不可能領域をバランス良く探索することが重要となる. これを実現するため, 本研究では, 解の実行不可能度を表すペナルティ関数にかかる重みを, 探索状況に応じて動的に調整する方法を提案する. さらに, 計算実験によりその有用性を確かめる.

2 問題定義

CSP は, それぞれ有限離散領域 D_i を持つ n 個の変数 X_i $(i=1,2,\cdots,n)$ と, m 個の制約 C_l $(l=1,2,\cdots,m)$ で定義され, 全ての制約を満たすように, 各変数 X_i に値 $j\in D_i$ を割当てる問題である. 各制約 C_lは変数 $X_{l_1},X_{l_2},\cdots,X_{l_{t_l}}$ 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の集合である. ここで, 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.

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, 割当てを 0-1 ベクトル $\boldsymbol{x}=(x_{ij}\mid i=1,2,\cdots,n, j\in D_i)$ で表す. また, 全ての制約を満たすような値の割当てを実行可能解と呼ぶ.

通常 CSP は目的関数を持たないが, ここでは, m 個の制約 C_l $(l=1,2,\cdots,m)$ の下で, 目的関数 $f(\boldsymbol{x})$ を最小化する, 目的関数付き制約充足問題を考える.

3 タブー探索の適用

本研究では,

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

が成立, すなわち, 全ての変数 X_i に対して値がちょうど 1 つ割当てられている解 $\boldsymbol{x}$ 全てから成る解集合 $\mathcal{X}$ を探索空間とする. また, $\boldsymbol{x}\in\mathcal{X}$の近傍 $N(\boldsymbol{x})$ 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解の全てとする.

本タブー探索では, 各反復において, 値が割当て直された変数 X_i をタブーリストに保持し, ある期間 (tabu tenure と呼ばれる), それらの変数の値の変更を禁止する.

解の探索を効果的に実行するため, 各制約 C_l に対して, それが満たされるならば負, 満たされないならば正の値をとるような関数 $g_l(\boldsymbol{x})$ を導入し, ペナルティ関数

$$p_l(\boldsymbol{x})=\max(0,g_l(\boldsymbol{x}))$$

を定義する. さらに, 各制約 C_l に対する非負の重み $\boldsymbol{w}=(w_1,w_2,\cdots,w_m)$ を用いて, 全体としてのペナルティ関数

$$p(\boldsymbol{w},\boldsymbol{x})=\sum_l w_lp_l(\boldsymbol{x})$$

により解の実行不可能度を表す. そこで, このペナルティ関数 $p(\boldsymbol{w},\boldsymbol{x})$ を元の目的関数 $f(\boldsymbol{x})$ に加えた最小化問題

$$\begin{array}{ll}\text{minimize} & F(\boldsymbol{w},\boldsymbol{x})=f(\boldsymbol{x})+p(\boldsymbol{w},\boldsymbol{x})\\ \text{subject to} & \boldsymbol{x}\in\mathcal{X},\end{array}\tag{1}$$

を考える. 各ペナルティ関数に対する重み w_l は, 最終的に実行可能解を得るためには, 十分大きな値をとる必要がある. しかし, 重み w_l が大き過ぎると, 探索空間が実行可能領域に制限され, 本来の目的関数 $f(\boldsymbol{x})$ の最小化が達成されにくくなる. 本研究では, 次章で述べる方法により, 探索中, 重み $\boldsymbol{w}$ を調整することで効果的な探索の実現を図る.

4 アルゴリズム

本アルゴリズムは以下のように書ける:

Algorithm for CSP with objective function

ステップ 1: 初期解を生成する.

ステップ 2: 各重み w_l を十分大きな値に設定し, 実行可能解 $\boldsymbol{x}$ が一つ得られるまで, 近傍探索を繰り返す (初期ラウンド). $\tilde{f} := f(\boldsymbol{x})$ とする. 以下, $\tilde{f}$ は, それまでに得られた実行可能解 $\boldsymbol{x}'$ の目的関数値 $f(\boldsymbol{x}')$ の最良値とする.

ステップ 3: 一つ前のラウンドにおける近傍探索で得られた ($F(\boldsymbol{w},\boldsymbol{x})$ を最小化するという意味での) 最良解を $\bar{\boldsymbol{x}}$ とし, 以下の方法で重み $\boldsymbol{w}$ を更新する:

(i) $F(\boldsymbol{w},\bar{\boldsymbol{x}}) < \tilde{f}$ **の場合:**
直前のタブー探索で少なくとも一つの実行可能解が探索された (されなかった) 場合, $L = \{1,2,\cdots,m\}$ ($L = \{l \mid g_l(\bar{\boldsymbol{x}}) > 0\}$) とし, 各 $l \in L$ に対して,

$$w_l := \max\left(0, w_l + \frac{\tilde{f} - F(\boldsymbol{w},\bar{\boldsymbol{x}})}{\sum_{l\in L}|g_l(\bar{\boldsymbol{x}})|^2} g_l(\bar{\boldsymbol{x}})\right).$$

(ii) $F(\boldsymbol{w},\bar{\boldsymbol{x}}) \geq \tilde{f}$ **の場合:**
$w_l := 0, \quad l = 1,2,\cdots,m.$

ステップ 4: k 回の近傍探索を 1 ラウンドとし (k はプログラムパラメータ), これを実行する. $\tilde{f}$ を更新する. 終了条件が満たされれば終了. さもなければステップ 3 に戻る.

ステップ 3 での重み $\boldsymbol{w}$ の更新は, ラグランジュ緩和問題に対する劣勾配法における乗数の更新式を参考にしている. しかし, 場合 (i) において, 直前のラウンドにおける近傍探索で実行可能解が得られなかった場合, 重み w_l を更新する制約 C_l を, 解 $\bar{\boldsymbol{x}}$ により破られている制約に限定している (すなわち, 減少する重み w_l はない). その結果, 探索領域を実行可能領域に近付けるという効果が期待できる. また, 任意の重み $\boldsymbol{w}$ に対して, $F(\boldsymbol{w},\boldsymbol{x}) \leq \tilde{f}$ となる解 $\boldsymbol{x}$ が存在することから, (ii) の場合, 直前のラウンドで探索された領域には, (目的関数 $f(\boldsymbol{x})$ を $\tilde{f}$ より小さくするという意味での) 良い解が存在していないと思われる. そこで, 重みを全て 0 にし, 探索の多様化を実現している.

5 計算実験

上述のアルゴリズムを用いて, 集合カバー問題, 一般化割当て問題, 時間割問題など, 制約充足問題に定式化できる問題に対して計算実験を行った. ここでは, それらの中から一般化割当て問題に対する計算結果を示す.

一般化割当て問題

一般化割当て問題 (Generalized Assignment Problem, GAP) は, 各エージェントの資源制約を満たしつつ, コストを最小化するよう n 個の仕事を d 個のエージェントに割当てる問題である. GAP は, 仕事を変数, エージェントの集合を値の集合と見なすことで, 容易に CSP に定式化できる.

GAP のベンチマーク問題 12 問に対する計算結果を表 1に示す. ここで, GIY[1], LKGG[2] は GAP に対する専用アルゴリズムであり, old NI, new NI は, それぞれ, 重みを固定, および自動調整したときの我々のアルゴリズムを示す.

表 1 から分かるように, 重みを自動調整することにより, 探索能力を向上させることができた. また, GAP に対する専用アルゴリズムと比べても, あまり遜色のない解が得られており, 汎用アルゴリズムとしての我々のアルゴリズムの有用性が示されている.

表 1: GAP に対する計算結果.

type	n	d	GIY	LKGG	old NI	new NI
C	100	5	1931	1931	1934	1933
C	100	10	1402	1403	1408	1402
C	100	20	1245	1245	1244	1245
C	100	5	3456	3457	3466	3461
C	100	10	2807	2812	2812	2808
C	100	20	2393	2396	2397	2394
D	100	5	6353	6386	6636	6362
D	100	10	6360	6406	6705	6443
D	100	20	6234	6297	6674	6374
D	100	5	12744	12788	13464	12883
D	100	10	12445	12537	13328	12527
D	100	20	12273	12436	13194	12480

6 おわりに

ペナルティ関数に対する重みを自動調整することで, 効果的な探索の実現を図った. なお, さらに詳しい計算結果は当日発表させて頂く.

参考文献

[1] Glover, F., T. Ibaraki and M. Yagiura, "An ejection chain approach for the generalized assignment problem" (to be submitted).

[2] Laguna, M., J.P. Kelly, J.L. González-Velarde and F. Glover, "Tabu search for the multi-level generalized assignment problem," *European Journal of Operational Research* 82 (1995) 176-189.

[3] Nonobe, K. and T. Ibaraki, "A tabu search approach to the CSP (Constraint Satisfaction Problem) as a general problem solver", *European Journal of Operational Research* 106 (1998) 599-623.