3J1-OS7-6

値変更コスト付き動的CSPの定式化とその解法

Dynamic CSP with Decision Change Costs :Formalization and Solutions

波多野大督　　平山勝敏
Daisuke HATANO　Katsutoshi HIRAYAMA

神戸大学大学院海事科学研究科
Graduate School of Maritime Sciences, Kobe University

We introduce Dynamic CSP with decision change costs, in which an assignment of each CSP and penalty changing assignments between two neighboring CSPs can be represented as a decision and some costs respectively. The purpose of this problem is to find a sequence of assignments minimizing total of the costs for changing variable. For this problem, we provide two solutions. The first uses a Weighted CSP solver after we encode the entire problem as a Weighted CSP. The second uses the Lagrangian decomposition technique that divides the entire problem into sub-problems, each of which can be separately solved by an exact Weighted CSP solver, and produces both lower and upper bounds on the optimal in an anytime manner. To compare the performance of these solvers, we experimented on the target tracking problem. The experimental results show that a solver based on Lagrangian decomposition performs better.

1. はじめに

制約充足問題（Constraint Satisfaction Problem: CSP）は高い定式化能力を持つことから，その解法は汎用的な問題解決法となりうる．

一方，動的 CSP は CSP の系列として表され，系列内のすべての CSP を満たすような解の系列を求めることが目的となる [Dechter and Dechter 88]．動的 CSP の解法は，２つのアプローチに分類される．

１つは proactive アプローチで，将来の情報が既知である状態でのアプローチである．そのため，求めた解は起こりうる変化に対して頑健である．robust 解 [Walsh 02] と flexible 解 [Ginsberg *et al.*98] がこの例となる．

もう１つは，reactive アプローチで，将来の情報が未知である状態でのアプローチである．このアプローチでは，情報の再利用が重要であり，解の再利用 [Verfaillie and Schiex 94] と推論の再利用 [Bessière 91] はその典型的な方法である．

動的 CSP は各 CSP の解を意思決定とすると，動的な意思決定問題と捕えられる．この問題では変化が少ない意思決定を行うことが望ましいと考えられる．本論文では，この意思決定の変化をコストとして表現し，コストの最小化を行う．意思決定変化のコストは様々な動的意思決定問題で見られるにも関わらず，CSP でこの問題を扱った研究はあまり見られない．

そこで，値変更コスト付き動的 CSP を導入する．値変更コスト付き動的 CSP は CSP の系列と値変更コストを入力として与え，値変更コストの合計が最小となるような解の系列を出力させる問題である．この問題の解法は動的 CSP に対する proactive アプローチの１つと言える．

先行研究として，値変更コスト付き動的 SAT [Hatano and Hirayama 11] がある．この研究は与えられた SAT の系列に対して，コストが最小になるモデルの系列を求めることが目的である．本研究は，値変更コスト付き動的 SAT を CSP に一般化したものと見なすことができる．

連絡先: 波多野大督，daisuke-hatano@stu.kobe-u.ac.jp,
神戸大学大学院海事科学研究科海事科学専攻,
〒 658-0022　神戸市東灘区深江南町５－１－１

本論文では，値変更コスト付き動的 CSP に対する２つの解法を提案する．１つ目の解法は全体の問題を重み付き CSP (Weighted CSP : WCSP) に変換し，WCSP ソルバーで解く方法である．２つ目はラグランジュ分解という方法を用いる解法である．この方法は全体の問題を部分問題に変換し，各部分問題を WCSP ソルバーで解くことで，上界と下界を得る方法である．

以下，本論文は次のように構成される．まず，2 節で CSP と動的 CSP を定義し，3 節で値変更コスト付き動的 CSP を提案する．4 節では値変更コスト付き動的 CSP に対する２つの解法について記述し，5 節では２つの解法に基づいた様々なソルバーの性能を評価する．最後に 6 節で本論文の結論を述べる．

2. CSP と動的 CSP

CSP とは，変数集合 $X = \{x_1, \ldots, x_n\}$，値域集合 $D = \{D_1, \ldots, D_n\}$，制約集合 C が与えられた時，制約集合 C を全て満たす値の割り当てを探索する問題である．値域 D_i は変数 x_i が取りうる値の集合である．動的 CSP は CSP の拡張で，現実問題の動的性質をモデル化することが目的である．ここでは動的 CSP の定義を述べる．

定義 1 (動的 CSP) 動的 *CSP* の問題例は (X, ϕ) で与えられる．ϕ は関数 $\phi : T \rightarrow \mathrm{CSP}(X)$，$T$ は非負整数の集合である．$\mathrm{CSP}(X)$ は起こりうる全ての *CSP* で，X の変数のみで構成されている．

すなわち，動的 CSP の問題例は X の変数で構成された CSP の系列で表現され，stage s における CSP は関数 ϕ によって与えられる．また，k-stage 動的 CSP とは stage $(k-1)$ 以降は変化しない動的 CSP である．

定義 2 (k-stage 動的 CSP) *k-stage* 動的 *CSP* は (k, X, ϕ) で与えられる．ここで，$\forall s \geq k : \phi(s) = \phi(k-1)$ である．

動的 CSP には 2 種類あり，1 つは決定問題としての動的 CSP で，任意の stage s において $\phi(s)$ が解 $A(s)$ を持つかどうかを決定する問題である．もし，すべての $\phi(s)$ に解があれば，ϕ

は充足可能で，そうでなければ ϕ は充足不可能である．もう 1 つは解追跡としての動的 CSP で，ϕ が充足可能と仮定して具体的な解の系列を求める問題である．本論文では解追跡に焦点を当てる．

3. 値変更コスト付き動的 CSP

解が時間を経て変化すれば意思決定者もそれに応じて意思決定を変えることは自然なことである．もし，実世界で意思決定が変化すれば，それに対してコストを支払わなければならないと仮定する．このコストを一般に意思決定変更コストと呼ぶ．

意思決定変更コストはアプリケーションに依存して定義されるべきだが，ここでは変数の値を変更するコストの合計として定義する．まず，ある stage において変数の値が変化した時のコストを定義する．

定義 3 (値変更コスト) *stage s における変数 x_i が変化* したときのコストは関数 $f : T\backslash\{0\} \times X \times D \times D \to \mathcal{R}^+$ によって与えられる．ここで，T は 非負整数の集合で X は変数の集合，D は値域の集合，$\mathcal{R}^+$ は正の実数の集合である．

例えば，変数 x_i の値域が $D_i = \{R, G, B\}$ であるとき，$f(s, x_i, R, G)$ は stage s において変数 x_i が R から G に変化した時にかかるコスト $c^{i,s}_{R,G}$ を返す．

2 つの連続した解 $A(s-1)$ と $A(s)$ が与えられたとする．この 2 つの解間で変数 x_i の値が変化した時のコストは $cost(x_i, A(s-1), A(s))$ と記述され，コスト関数 $f(s, x_i, \_, \_)$ で求められる．そして，各変数のコストを足し合わせることにより，$A(s-1)$ と $A(s)$ の間の変化に対するコストを求めることができる．これを $cost(A(s-1), A(s))$ と記述する．すなわち，

$$cost(A(s-1), A(s)) \equiv \sum_{x_i \in X} cost(x_i, A(s-1), A(s)).$$

さらに，A を集合 T 上の解の系列，すなわち $A = \{A(s) \mid s \in T\}$ とすると，この解の系列 A のコストを次式のように定義する．

$$cost(A) \equiv \sum_{s \in T\backslash\{0\}} cost(A(s-1), A(s)). \quad (1)$$

値変更コスト付き動的 CSP は以下のように定義できる．

定義 4 (値変更コスト付き動的 CSP) 値変更コスト付き動的 *CSP* の問題例は *5* つ組 $(X, \phi, f, +, +)$ で 与えられる．

動的 CSP と同様に，値変更コスト付き k-stage 動的 CSP を次のように定義できる．

定義 5 (値変更コスト付き k-stage 動的 CSP) 値変更コスト付き *k-stage* 動的 *CSP* の 問題例は 6 つ組 $(k, X, \phi, f, +, +)$ で与えられる．ここで，$\forall s \geq k : \phi(s) = \phi(k-1)$ である

各変更コストを $+$ 以外の演算子で集計することも考えられるが，ここでは，6 つ組 $(k, X, \phi, f, +, +)$ で表現される値変更コスト付き k-stage 動的 CSP に注目する．この枠組みは自然な問題設定の 1 つであると考えられる．値変更コスト付き k-stage 動的 CSP の問題では，(1) 式で与えられるコストを最小化するような解の系列を求めることが目的となる．この最小のコストを最適値と呼び，もし最適値が存在する場合，それを与える解が最適解となる．

4. 解法

この節では，この問題に対する 2 つの解法を提案する．

4.1 WCSP

1 つ目の解法では，まず与えられた $(k, X, \phi, f, +, +)$ で表される問題を WCSP に変換し，TOULBAR2 などの既存の WCSP ソルバーなどを用いて解く．WCSP には必ず満たさなければならない hard 制約と必ずしも満たす必要はないが満たされない場合は重みと呼ばれるコストがかかる soft 制約がある．

問題の変換は以下の手順で行う．$\phi(s)$ の各制約に対して，stage s でラベル付けされた変数を用いて hard 制約を導入する．例えば，$\phi(2)$ の制約 $(x_1 \neq x_2)$ は $(x_{1,2} \neq x_{2,2})$ という hard 制約に変換される．soft 制約は f で定義されたコストを用いて導入される．例えば，$f(2, x_1, R, G) = 7$ は stage 2 において変数 x_1 の値が R から G に変化したとき，コストを 7 支払う必要があることを示している．これを soft 制約で表すと重み 7 を持つ soft 制約 $(x_{1,1} \neq R \vee x_{1,2} \neq G)$ となる．

4.2 ラグランジュ分解

2 つ目はラグランジュ分解を使用した解法で，最適値の上界と下界を求めることができる．

4.2.1 分解

まず，コスト関数 f を整数計画問題に変換する．今，変数 x_i の値域 $D_i = (R, G, B)$ であるとき，$f(s, x_i, \_, \_)$ はコスト行列 F を用いて，

$$F_{i,s} = \begin{pmatrix} 0 & c^{i,s}_{R,G} & c^{i,s}_{R,B} \\ c^{i,s}_{G,R} & 0 & c^{i,s}_{G,B} \\ c^{i,s}_{B,R} & c^{i,s}_{B,G} & 0 \end{pmatrix},$$

と表現できる．コスト行列の n 行目は変数 $x_{i,s-1}$ が値 $D_i(n)$ を取るときのコストベクトルを表し，m 列目は変数 $x_{i,s}$ が値 $D_i(m)$ を取るときのコストベクトルを表す．コスト行列の対角線上の成分は変数の値に変化がなければコストがかからないことを示している．ここで，変数 x_i の取りうる値 R, G, B をそれぞれ e_1，e_2，e_3 とする．e_j は j 番目の要素が 1 で，それ以外の要素が 0 である列ベクトルで，サイズは D_i の要素数に等しい．このとき，変数 $x_{i,s-1}$ と $x_{i,s}$ に関わるコスト関数は以下の整数計画問題の最適値に等しいことがわかる．

$$\begin{aligned} \min.\quad & \alpha^T_{i,s} \cdot F_{i,s} \cdot \beta_{i,s} \\ \text{s. t.}\quad & x_{i,s-1} - \alpha_{i,s} = 0,\ x_{i,s} - \beta_{i,s} = 0, \\ & x_{i,s-1}, x_{i,s}, \alpha_{i,s}, \beta_{i,s} \in \{e_1\ e_2\ e_3\}, \end{aligned}$$

ここで $x_{i,s-1}$ と $x_{i,s}$ はそれぞれ stage $s-1$ と s における変数 x_i で $\alpha_{i,s}$ と $\beta_{i,s}$ は e_1，e_2，e_3 の値をとる補助変数である．例えば，$x_{i,s-1} = e_2$，$x_{i,s} = e_1$ の場合，上記の問題の最適値は $c^{i,s}_{G,R}$ になる．この変換は任意の隣り合う 2 つの変数 $x_{i,s-1}$，$x_{i,s}$ に関して適用できる．stage 毎のコストと stage 全体のコストは足し合わすため，$(k, X, \phi, f, +, +)$ の全体の問題は CSP を含んだ整数計画問題として，以下のように定式化できる．なお，以下では決定変数 $x_{i,s}, \alpha_{i,s}, \beta_{i,s}$ の制約を省略する．

$$\begin{aligned} \mathcal{P}\ :\ \min.\quad & \sum_{s=1}^{k-1} \sum_{i=1}^{n} (\alpha^T_{i,s} \cdot F_{i,s} \cdot \beta_{i,s}) \\ \text{s. t.}\quad & x_{i,s-1} - \alpha_{i,s} = 0,\ x_{i,s} - \beta_{i,s} = 0, \\ & i = 1, ..., n,\ s = 1, ..., k-1, \quad (2) \\ & \phi(s), \quad s = 0, ..., k-1, \quad (3) \end{aligned}$$

問題 $\mathcal{P}$ は CSP をすべて制約 (3) として持っているため，$\mathcal{P}$ の実行可能解は解の系列となっており，このときの実行可能解の目的関数値は値変更コストの和となっている．したがって，$\mathcal{P}$ を解くことで全体の問題に対する最適解を求めることが可能である．正確には，$\mathcal{P}$ の最適解を各変数 $x_{i,s}$ に写像したものが最適解となる．

しかし，$\mathcal{P}$ を直接解くことは困難なので，$\mathcal{P}$ を緩和し，扱いやすい問題に変換する．ここでは，各 stage の変数毎に定義された (2) の制約を緩和したラグランジュ緩和問題を導入する．

$$\begin{aligned}\mathcal{L} \ : \ L(\mu) \ = \ \text{min.} & \sum_{s=1}^{k-1}\sum_{i=1}^{n}(\alpha_{i,s}^{T}\cdot F_{i,s}\cdot\beta_{i,s}) \\ & + \sum_{s=1}^{k-1}\sum_{i=1}^{n}\mu_{\alpha,i,s}(x_{i,s-1}-\alpha_{i,s}) \\ & + \sum_{s=1}^{k-1}\sum_{i=1}^{n}\mu_{\beta,i,s}(x_{i,s}-\beta_{i,s}), \\ \text{s. t.} \quad & \phi(s), \quad s=0,...,k-1,\end{aligned}$$

ここで，$\mu_{\alpha,i,s}$，$\mu_{\beta,i,s}$ はラグランジュ乗数ベクトルと呼ばれ，その要素は実数値を取る．また，$\mu_{\alpha,i,s}$，$\mu_{\beta,i,s}$ は要素数が変数 x_i の値域 D_i の要素数と同数の行ベクトルである．

この問題は以下のように $k+1$ 個の部分問題に分解することが可能である．

$$\begin{aligned}L^{aux}(\mu) \ = \ \text{min.} & \sum_{s=1}^{k-1}\sum_{i=1}^{n}(\alpha_{i,s}^{T}\cdot F_{i,s}\cdot\beta_{i,s}) \\ & - \sum_{s=1}^{k-1}\sum_{i=1}^{n}\{\mu_{\alpha,i,s}\alpha_{i,s}+\mu_{\beta,i,s}\beta_{i,s}\}, \quad (4)\end{aligned}$$

$$L^{0}(\mu) \ = \ \text{min.}\sum_{i=1}^{n}\mu_{\alpha,i,1}x_{i,0}, \quad \text{s. t.} \quad \phi(0), \qquad (5)$$

各 stage $s\in\{1,...,k-2\}$ について

$$L^{s}(\mu) \ = \ \text{min.}\sum_{i=1}^{n}(\mu_{\beta,i,s}+\mu_{\alpha,i,s+1})x_{i,s}, \ \text{s. t.}\ \phi(s), \quad (6)$$

$$L^{k-1}(\mu) \ = \ \text{min.}\sum_{i=1}^{n}\mu_{\beta,i,k-1}x_{i,k-1}, \ \text{s. t.}\ \phi(k-1). \quad (7)$$

ここで，各部分問題が実際には WCSP であることに注意すべきである．さらに， (4) は補助変数の soft 制約のみからなるが，計算量は値域のサイズを d とするとたかだか $O(d^2)$ である．一方，他の部分問題には hard 制約 $\phi(s)$ と stage s の変数の単位 soft 制約が含まれている．

μ の任意の値に対して，$\mathcal{L}$ の最適値である $L(\mu)$ は，$\mathcal{P}$ の最適値の下界となる．従って，その下界を最大化するような μ を探すラグランジュ双対問題

$$\mathcal{D} \ : \ \text{max.} \quad L(\mu) \quad \text{s. t.} \quad \mu\in\Re,$$

を定義でき，一般に，これを解くことにより $\mathcal{P}$ の最適値を求めるという解法を構成できる．

このラグランジュ双対問題では，μ が決定変数であり，しかも，μ に関する制約はなく目的関数のみ考慮すればよい．また，前述のように $\mathcal{L}$ を部分問題に分解できることから，上記のラグランジュ双対問題は，

$$\mathcal{D} \ : \ \text{max.} \quad L^{aux}(\mu)+\sum_{s=0}^{k-1}L^{s}(\mu) \quad \text{s. t.} \quad \mu\in\Re.$$

と書き直すことができる．つまり，(4) から (7) の部分問題の最適値の和を目的関数として，μ の空間を山登り型で探索することで，原問題 $\mathcal{P}$ の解を求めることができる．以下，この解法に基づく操作の概要を示す．

4.2.2 操作の概要

ここでは，$\mathcal{P}$ の実行可能解に加え，上界と下界を求める操作の概要を記述する．

step 1: μ の全ての要素に 0 を代入する;

step 2: (4) を列挙法を用いて解き，(5) から (7) の部分問題を既存の WCSP ソルバーを用いて解く;

step 3: 最も高い下界 *LB*，最も低い上界 *UB*，そして，下界を与える実行不可能解から $\mathcal{P}$ の実行可能解 *A* を求める;

step 4: もし終了条件を満たすならば，*LB*，*UB*，*A* を返す．それ以外の場合は μ を更新し，step 2 へ戻る;

この操作は step 1 から始まり，終了条件を満たすまで step 2 から step 4 を繰り返す．以下，step 3 と step 4 について詳述する．

4.2.3 下界と上界

前述のように，$\mathcal{P}$ の下界は (4) から (7) の部分問題の最適値を合計することで求められる．さらに，各部分問題の最適解からラグランジュ緩和問題を生成する時に緩和した (2) 式の制約がそれぞれ満たされるように補助変数の値を変更することにより $\mathcal{P}$ の実行可能解とそのときの目的関数値，つまり上界が得られる．したがって，step 2 から step 4 の 1 度の繰り返し操作で実行可能解と上界，下界を求めることができる．step 3 においては，これまでの繰り返しで得られた下界と上界のうちで最も良い値をそれぞれ *LB*，*UB* に保存する．

4.2.4 終了条件

$\mathcal{P}$ の最適値が求められたかどうかを判断する方法が 2 つある．1 つ目は部分問題の最適解と $\mathcal{P}$ の最適解の関係について記述した下記の定理を用いる．

定理 1 μ のある値のもとで，もし，*(4)* から *(7)* の部分問題のすべての最適解が，緩和された *(2)* の制約をすべて満たしている場合，各部分問題の最適解は $\mathcal{P}$ の最適解になっている．

これは部分問題の最適解が下界だけでなく，上界も与えるためである．従ってすべての部分問題の最適解が (2) の制約を満たす場合，操作を終了させる．

2 つ目は単純である．step 3 で得られた $\mathcal{P}$ の実行可能解の目的関数値が LB と同じ値を持つ場合，この実行可能解は最適解である．この事実もまた終了条件として使用される．

4.2.5 ラグランジュ乗数の更新

$\mathcal{P}$ の最適解が見つからない場合，step 4 で μ を更新し，$\mathcal{P}$ の最適値により近い下界を求める．この手続きはラグランジュ双対問題の探索アルゴリズムに相当する．本論文では，この問題を解く方法として劣勾配法 [Bertsekas 99] を使用する．

しかし，劣勾配法は $\mathcal{P}$ の最適値に収束するとは限らない．従って，step 4 の終了条件が満たされないことがあり，その場合は繰り返し操作を強制的に打ち切る必要がある．このときその時点での最良の上界と下界だけが得られることになる．

5. 実験

前節の 2 つの解法に基づいた以下のソルバーをターゲット追跡問題を用いて評価する．

- WCSP に基づくソルバー群

  TOULBAR2 ：深さ優先の分枝限定法による厳密解法を実装したソルバーである．

  LOCALSEARCH ： 山登り法とランダム選択を確率的に選択するを用いる局所探索法を実装している．

- ラグランジュ分解法に基づくソルバー群

  LD ：ラグランジュ分解に基づくソルバーで，各部分問題を TOULBAR2 で解く．

実験の目的は限られた時間内でどれぐらい質の高い実行可能解が得られるかを知ることである．実行時間を制限した場合，WCSP による方法では上界しか得られないがラグランジュ分解法では上界と下界の両方を求めることが可能である．

次に実験に用いるターゲット追跡問題について説明する．この問題では，格子上に並べられた 25 個のセンサステーションがあり，各センサステーションは 4 つのセンサーを持っている．これらのセンサステーションにはターゲットを追うための 2 つの制約がある．1 つは，領域内にターゲットがいる場合，少なくとも 2 つのセンサーを ON にしなければならない制約である．もう 1 つは，領域内にターゲットがいない場合，どのセンサーも ON にしてはならない制約である．ある stage におけるターゲットの状態を与えられた時，この状態は制約に基づき CSP として定式化される．この時，各センサステーションは変数として表現でき，値域はセンサーをつける位置により，{OFF，右上，右下，左上，左下} の 5 つの値を取る．同様に，k 個の連なるターゲットの状態を与えられたとき，これらの状態は動的 CSP として定式化できる．さらに動的 CSP をセンサーの切り替え回数を最小にするため，値変更コスト付き動的 CSP として定式化する．具体的に，各 $k \in \{10, 15, ..., 35\}$ に対する 30 個の $(k, X, \phi, f, +, +)$ の問題例を次のように生成する．

- k の範囲は $\{10, 15, ..., 35\}$ である．
- X は 25 個の変数の集合 $\{x_1, ..., x_{25}\}$ である．
- ϕ は各 stage s のターゲットの状態から得られる充足可能な CSP である．
- f はある stage の変更コストを表し，10^6 で固定する．

実験では，ある決められた時間内に得られた実行可能解の質 UB/LB を測定する．明らかにこの値は 1 に近いほど良い．各実験は Intel Corei7-2600@3.4GHz，4 コア，メモリ 8GB で行った．ラグランジュ分解法の基本部分は JAVA で実装し，Ubuntu 10.10(64bit) の JDK1.6.0_20 でコンパイルした．TOULBAR2 は作者のページから最新バージョンをダウンロードし，デフォルトの設定で利用した．ラグランジュ分解法に関しては並列化が容易であるため，なぜなら，μ の値が固定されれば，分解された部分問題は互いに独立な問題と見なせるからである．この実験では 4 コアを用いて部分問題を並列に解くこととした．さらに，平等な比較のため，他のソルバーでも 4 コアを用いてポートフォリオ型の並列処理を行った．

図 1(a)，図 1(b) はそれぞれ系列数と制限時間を変化させたときの上界の質を表した図である．図 1(b) では系列数 35 の

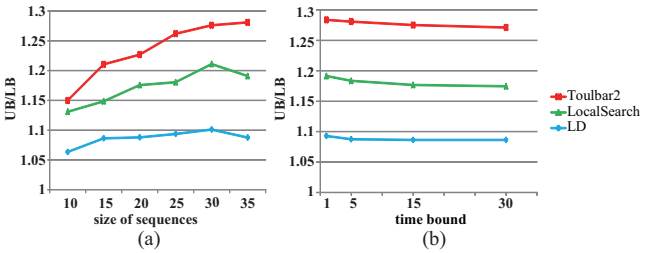

図 1: ターゲット追跡問題に対する，(a) 上界の質 対 系列数 (制限時間 5 分)．(b) 上界の質 対 制限時間 (系列サイズ 35)

30 種類の問題に対する 1 分，5 分，15 分，30 分間の実行結果を表す．図 1(a)，図 1(b) ともに LD の方が TOULBAR2 と LOCALSEARCH よりも解の質が良いことを示している．

6. おわりに

本論文では，値変更コスト付き動的 CSP の枠組みを導入し，それに対する 2 つの解法として WCSP に基づく解法とラグランジュ分解に基づく解法を提案した．これらの解法のうち，ラグランジュ分解に基づく解法では上界だけでなく下界も求めることができる．また，今回提案したラグランジュ分解法は高い拡張性があり，値変更コスト付き動的 SAT にもそのまま転用可能であるさらに，ラグランジュ分解に基づくソルバーである LD は制限時間を厳しくしても，質の良い実行可能解が得られるという実験結果が得られた．実世界の問題の動的性質を考えれば，この性質は重要であると考えられる．

参考文献

[Bertsekas 99] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, second edition, (1999).

[Bessière 91] Christian Bessière. Arc-consistency in dynamic constraint satisfaction problems. *AAAI-91*, pages 221–226, (1991).

[Dechter and Dechter 88] Rina Dechter and Avi Dechter. Belief maintenance in dynamic constraint networks. *AAAI-88*, pages 37–42, (1988).

[Ginsberg *et al.*98] Matthew L. Ginsberg, Andrew J. Parkes, and Amitabha Roy. Supermodels and robustness. *AAAI-98*, pages 334–339, (1998).

[Hatano and Hirayama 11] Daisuke Hatano and Katsutoshi Hirayama. Dynamic SAT with Decision Change Costs :Formalization and Solutions. *IJCAI-11*, to appear, (2011).

[Verfaillie and Schiex 94] Gérard Verfaillie and Thomas Schiex. Solution reuse in dynamic constraint satisfaction problems. *AAAI-94*, pages 307–312, (1994).

[Walsh 02] Toby Walsh. Stochastic constraint programming. *ECAI-02*, pages 111–115, (2002).