Kobe University Repository : Kernel

PDF issue: 2025-12-04

値推移コスト付き動的制約充足問題とその解法

波多野, 大督
平山, 勝敏

(Citation)
人工知能学会論文誌,28(1):34-42

(Issue Date)
2013

(Resource Type)
journal article

(Version)
Version of Record

(URL)
https://hdl.handle.net/20.500.14094/90002958

原著論文 Original Paper

値推移コスト付き動的制約充足問題とその解法

Dynamic CSP with Decision Transition Costs and its Solutions

波多野 大督 Hatano Daisuke 神戸大学大学院海事科学研究科 Graduate School of Maritime Sciences, Kobe University daisuke-hatano@stu.kobe-u.ac.jp

平山 勝敏 Hirayama Katsutoshi （同 上） hirayama@maritime.kobe-u.ac.jp

keywords: CSP, Dynamic CSP, Weighted CSP, Lagrangian decomposition, planning

Summary

The dynamic constraint satisfaction problem (DynCSP) is a sequence of CSP instances. By introducing a notion of *decision transition costs*, one natural optimization problem results, where we search for a sequence of solutions that minimizes a total sum of decision transition costs. We will refer to this problem as the *dynamic constraint satisfaction problem with decision transition costs* (DynCSP-DTC). Previously, Hatano and Hirayama have presented an integer linear programming formulation to apply *Lagrangian decomposition* to the SAT-version of the problem called Dynamic SAT with decision change costs (DynSAT-DCC). However, since their linear encoding of decision change costs was specially designed for DynSAT, a new encoding method is required when we try to extend Lagrangian decomposition to solve general DynCSP-DTC. In this paper, we will introduce the *quadratic encoding* of decision transition costs that enables Lagrangian decomposition to work on general DynCSP-DTC including DynSAT-DCC. Furthermore, we empirically show that, even on DynSAT-DCC, Lagrangian decomposition with quadratic encoding performs more efficiently than other methods.

1. は じ め に

動的制約充足問題 (Dynamic Constraint Satisfaction Problem: 動的 CSP) は，実世界の動的な性質をモデル化するために提案された問題である．入力は，時間軸上に並べられた CSP の系列であり，解法は，過去に解いた問題例の情報から次の問題例の解を求める *reactive* アプローチ [Verfaillie 94, Bessière 91, Schiex 93] と，問題例の系列が予め得られている状態から解の系列を求める *proactive* アプローチ [Fargier 96, Wallace 98, Walsh 02, Freuder 91, Ginsberg 98, Bofill 10, 波多野 11] に大別できる．

近年，動的 SAT（動的 CSP の一種）に対して値変更コストを導入した，値変更コスト付き動的 SAT（DynSAT with decision change costs）という新たな問題が提案された [波多野 11]．この問題は，値変更コストの総和が最小となるような解の系列を求めることを目的とする．波多野と平山は，値変更コスト付き動的 SAT に対して，重み付き部分 MaxSAT に変換して解く方法とラグランジュ分解法を用いた方法を提案した．特に後者のラグランジュ分解法は，値変更コストを線形の整数計画問題に符号化（線形符号化）して緩和することにより，全体の問題を部分問題の集合に分解することを特徴とする．評価実験によれば，ラグランジュ分解を用いた方法は 系列の各部分問題の実行可能領域が比較的小さい密な問題例に対して経験的に良い結果を与えることが分かっている．このように，値変更コスト付き動的 SAT とその解法は，ターゲット追跡問題 [波多野 11]，貨物の積み付け問題，ナーススケジューリング問題 [服部 05] などのある種の動的環境におけるプランニング／スケジューリング問題をモデル化し，良質な解を与えるものとして期待されている．

一方で，値変更コスト付き動的 SAT とその解法には，多値変数を扱うことができない，また，値を維持するコストを扱うことができないという 2 つの課題がある．そこで本研究では，これらの課題を解決するために値推移コスト付き動的 CSP とその解法を提案する．

解法としては，値変更コスト付き動的 SAT の場合と同様に，ラグランジュ分解法を用いた解法を提案する．本解法の特徴は，多値変数における値推移コスト（値変更コストと値維持コストを一般化したもの）を記述する新たな符号化法として 2 次符号化法を用いる点である．2 次符号化法では，ある多値変数（値域サイズは $|D|$）の値を $|D|$ 次元単位ベクトル，値を変更／維持するコストを $|D| \times |D|$ の行列で表現し，値推移コスト付き動的 CSP を離散 2 次計画問題として定式化する．これにより，値変更コスト付き動的 SAT の場合と同様に，値推移コスト付き動的 CSP に対してもラグランジュ分解が適用可能となる．

評価実験によれば，2 次符号化法を用いたラグランジュ

分解法は，従来の線形符号化法を用いたラグランジュ分解法よりも値変更コスト付き動的 SAT に対して質の良い解を発見できることがわかった．また，値推移コスト付き動的 CSP に対しても，2 次符号化法を用いたラグランジュ分解法は重み付き CSP に対する最新の厳密解法と比較して質の良い解を発見できることがわかった．

以下，本論文は次のように構成される．まず，2 章では関連研究を概観し，3 章で本研究の動機を例題を用いて説明する．その後，4 章で値推移コスト付き動的 CSP の定義を述べ，5 章では 2 次符号化法の詳細とラグランジュ分解を用いた解法を説明する．そして，6 章では実験によりラグランジュ分解を用いた方法と他の解法の性能を比較し，最後に 7 章で本研究をまとめる．

2. 関 連 研 究

動的 CSP に対する解法は，過去に解いた問題例の情報から次の問題例の解を求める *reactive* アプローチ [Verfaillie 94, Bessière 91, Schiex 93] と，問題例の系列が予め得られている状態から解の系列を求める *proactive* アプローチ [Fargier 96, Wallace 98, Walsh 02, Freuder 91, Ginsberg 98, Bofill 10, 波多野 11] に大別できる．

proactive アプローチに関して，[Walsh 02] では，確率が付与された変数が存在する決定問題が提案されている．また，[Wallace 98] では，割り当てが一時的に変化しても制約を満たすような解（安定解）を効率的に求める方法が提案され，[Fargier 96] では，操作可能な変数と操作不可能な変数が存在する mixed CSP が提案されている．さらに，[波多野 11] では，動的 SAT に値変更コストを導入した問題である値変更コスト付き動的 SAT が提案されている．筆者らの知る限り，*proactive* アプローチの既存研究で，[波多野 11] 以外に意思決定の変化に対するコストという概念を導入した事例はない．

一方，本研究に類似した既存研究として動的重み付き最大制約充足問題 [服部 05] がある．そこでは，意思決定の変更をなるべく少なくするために，与えられた重み付き最大制約充足問題に対して，変数の値を変更する場合にコストを課す暫定制約が導入される．本研究で提案する値推移コスト付き動的 CSP と動的重み付き最大制約充足問題は，系列内の問題が前者は CSP，後者は重み付き最大制約充足問題（Weighted MaxCSP）という違いはあるが，基本的な考え方は類似している．しかしながら，[服部 05] では，暫定制約を導入する効果に焦点を当てており，新たな独自解法の提案には至っていない．また，値推移コスト付き動的 CSP に対する解法は，環境の変化に関する情報を十分集めた後に解の系列を求める *proactive* アプローチに分類されるが，動的重み付き制約充足問題に対して [服部 05] で提案されている解法は，環境の変化に対して即座に解を求める *reactive* アプローチに近い．

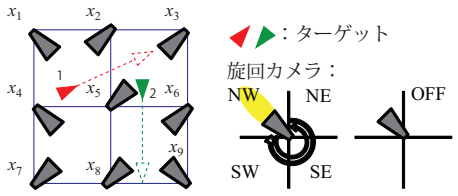

図 1 ターゲット追跡問題

3. 動 機

値変更コスト付き動的 SAT では，値変更コストを新たに定義することによりある種の動的環境におけるプランニング／スケジューリング問題を表現できるが，値変更コストだけでは表現できない問題設定が存在する．例えば，図 1 のような旋回カメラを用いたターゲット追跡問題に対して，カメラの旋回度に応じてコストが変わるような問題を考えてみよう．このターゲット追跡問題では，グリッド上に配置された旋回カメラを用いて幾つかの動くターゲットを追跡する．各カメラは 4 つの領域，北東 (NE) と北西 (NW), 南西 (SW), 南東 (SE) を監視できる．各ターゲットは少なくとも 2 つのカメラに捕捉される必要があり，そのターゲットが位置する領域の四隅にあるカメラのみがそのターゲットを捕捉可能となる．例えば，図 1 のターゲット 1 を捕捉するためには，x_1，x_2，x_4，x_5 の 4 つのカメラのうち少なくとも 2 つがターゲット 1 が位置する領域を監視する必要がある．また，この問題にはエネルギーを節約するための制約がある．それは，各カメラはターゲットがいない領域に対して監視しないという制約である．これらの制約を満たした上で，予測されたターゲットの位置情報の系列に対して，カメラの切り替えにかかるコストが最小となるようなスケジュールを作ることが目的である．

従来の値変更コスト付き動的 SAT では，4 つの各領域 {NE, NW, SW, SE} に対してカメラを向ける (真) か否か (偽) の 2 値を取る変数を割り当てていた．そのため，カメラを NE から SW へ変更するコストは NE の値を ON から OFF に変更するコストと SW の値を OFF から ON に変更するコストで表される．しかし，このコストにはカメラの旋回にかかるコストは考慮されていない．この問題点を解決するためには，センサーステーションを値 {OFF, NE, NW, SW, SE} を持つ 1 つの変数とする必要がある．そうすることにより，カメラの旋回コストを設定できる．このように多値変数を扱うことで表現できる問題設定がある．また，カメラを ON にし続けた場合は維持コストがかかるが，OFF にし続けた場合は維持コストがかからないと考えられるが，従来の方法では，ON と OFF を切り替えた場合にしかコストを定義できない．上記のような問題設定を表現するために，多値変数と維持コストを扱えるように値変更コスト付き動的 SAT を拡張したい．そこで，維持コストと変更コストを含む新たな

概念である値推移コストを動的 CSP に導入し，値推移コスト付き動的 CSP を定式化する．

4. 値推移コスト付き動的 CSP

CSP の問題例は 3 つ組 $(\mathcal{X},\mathcal{D},\mathcal{C})$ で書ける．$\mathcal{X}$ は変数の集合 $\{x_1,\ldots,x_n\}$ で，$\mathcal{D}$ はこれらの変数集合に対する値域の集合 $\{D_1,\ldots,D_n\}$ である．また，$\mathcal{C}$ は制約の集合 $\{C_1,\ldots,C_m\}$ を示す．各制約 C_j には変数のある部分集合に対して，選択可能もしくは選択不可能な値の組み合わせが記述されている．CSP の目的はすべての制約が充足可能か否かを決定し，もし可能であれば，少なくとも 1 つの値の組み合わせを返すことである．

動的 CSP の問題例は時間軸に対する CSP の系列として表現でき，動的な環境における決定問題を表現したい場合に使用される．動的 CSP の問題例は 2 つ組 $(\mathcal{X},\phi)$ で表現できる．$\mathcal{X}$ は変数の集合で，ϕ は非負整数値 T 上の関数である．関数 ϕ はある時刻 $s \in T$ を与えると，時刻 s の CSP 問題例 $(\mathcal{X},\mathcal{D}_s,\mathcal{C}_s)$ を返す．本論文では，系列のサイズを k に制限した k-stage 動的 CSP に焦点をあてる．k-stage 動的 CSP は $(k,\mathcal{X},\phi)$ で表記でき，その目的は k 個の CSP 問題例すべてが制約を満たす解を持つかどうかを判定し，もし解を持つならばその時の解の系列を少なくとも 1 つ見つけることである．

動的 CSP に導入する値推移コストは，値を変更したときのコストだけでなく維持したときのコストも含んでおり，次のように定義される．

【定義 1】（値推移コスト） ある時刻 s における変数 x_i の値推移コストは関数 $f_{i,s} : D_i \times D_i \rightarrow \mathcal{R}^+$ で与えられる．D_i は変数 x_i の値域を表し，$\mathcal{R}^+$ は非負実数の集合を表す．

例えば，変数 x_i の値域 D_i が $\{1,2,3\}$ とすると，$f_{i,s}(1,2)$ は変更コスト $c^{i,s}_{1,2}$ を返す．この変更コスト $c^{i,s}_{1,2}$ は時刻 $s-1$ で 1 の値をもつ変数 x_i の値を時刻 s で 2 に変更したときのコストである．同様に $f_{i,s}(1,1)$ は変数 x_i が時刻 $s-1$ と s の間で値を 1 に維持した場合にかかる維持コスト $c^{i,s}_{1,1}$ を返す．ここで，値変更コスト付き動的 SAT[波多野 11] とは異なり，維持コストを 0 にする必要はない．

次に，時刻 $s-1$，s における $\mathcal{X}$ への値の割り当てをそれぞれ $A(s-1)$，$A(s)$ とすると，$\mathcal{X}$ の要素である変数 x_i の値推移コストは関数 $f_{i,s}$ を使用して計算できる．このようなコストを $cost_i(A(s-1),A(s))$ と書くものとする．2 つの連続した解 $A(s-1)$ と $A(s)$ の間の値推移コストは以下のように定義される．

$$cost(A(s-1),A(s)) \equiv \sum_{i=1}^{n} cost_i(A(s-1),A(s)).$$

さらに，$\{A(0),A(1),\ldots,A(k-1)\}$ を k-stage 動的 CSP 問題例の解の系列 A とすると，この系列の値推移コストは以下のとおりとなる．

$$cost(A) \equiv \sum_{s=1}^{k-1} cost(A(s-1),A(s)). \tag{1}$$

最後に，値推移コスト付き k-stage 動的 CSP を次のような最適化問題として定義する．

【定義 2】（値推移コスト付き k-stage 動的 CSP） 値推移コスト付き k-stage 動的 CSP の問題例は $(k,\mathcal{X},\phi,\mathcal{F})$ で定義される．ここで，$\mathcal{X}$ は変数集合で，ϕ はある時刻における変数集合に対する CSP 問題例を返す関数である．さらに，$\mathcal{F}$ は関数の集合 $\{f_{i,s}|i=1,...,n,s=1,...,k-1\}$ である．ここで，$f_{i,s}$ はある時刻 s における変数 x_i の値推移コストを返す関数である．

ある値推移コスト付き (k-stage) 動的 CSP の問題例が与えられたとき，式 (1) で定義される値推移コストが最小となるような解の系列を探すことが目的となる．以後，値推移コスト付き動的 CSP の最適な解の系列を A^* と表現し，そのとき得られる最適な値推移コストの合計値を $cost(A^*)$ とする．

5. 解　　法

波多野と平山は値変更コスト付き動的 SAT に対して 2 つの解法を提案している．1 つは重み付き部分 MaxSAT 解法で，もう 1 つはラグランジュ分解を用いた解法である．これらの解法のうち，ラグランジュ分解を用いた解法は 系列の各部分問題の実行可能領域が比較的小さい密な問題例に対して経験的に良い性能をもつことが示されている [波多野 11]．彼らはラグランジュ分解法を適用するために，変数がもつ値変更コストを 0-1 整数計画問題に符号化している．しかし，この線形符号化法は動的 SAT のような 2 値変数を持つ問題に特化されたものであるため，値推移コスト付き動的 CSP に直接適用することはできない．したがって，本論文では，値推移コストに対する新たな符号化法である 2 次符号化法を導入する．

5·1 線形符号化法

次のような値変更コスト付き動的 SAT の問題例を考える．変数の値域がすべて真 (1) と偽 (0) の 2 値であり，時刻 s において，変数 x_i の値を 1 から 0 に変えたときの変更コストを $c^{i,s}_{1,0}$，0 から 1 に変更した時の変更コストを $c^{i,s}_{0,1}$ とする．波多野と平山はこれらのコストを以下のような 0-1 整数計画問題に符号化した [波多野 11]:

$$\begin{aligned} \text{min.} \quad & c^{i,s}_{1,0}y^{i,s}_{1,0} + c^{i,s}_{0,1}y^{i,s}_{0,1} \\ \text{s. t.} \quad & x^{s-1}_i - x^s_i - y^{i,s}_{1,0} + y^{i,s}_{0,1} = 0, \\ & x^{s-1}_i, x^s_i, y^{i,s}_{1,0}, y^{i,s}_{0,1} \in \{0,1\}. \end{aligned} \tag{2}$$

ここで，x^{s-1}_i と x^s_i はそれぞれ時刻 $s-1$ と s の変数 x_i で変数 $y^{i,s}_{1,0}$ と $y^{i,s}_{0,1}$ はともに 0 か 1 の値をとる補助変数

である．例えば，$x_i^{s-1}=1$ かつ $x_i^s=0$ であるとき，補助変数の値は $y_{1,0}^{i,s}=1$ と $y_{0,1}^{i,s}=0$ となり，その時の目的関数値は $c_{1,0}^{i,s}$ となる．

線形符号化法には 2 つの欠点がある．まず，この符号化法を用いると変数の変更コストは記述できるが，維持コストは記述できない点である．なぜなら，0-1 整数計画問題上で変更コストを表現するために x^{s-1} と x^s の差が一意であることを利用しているからである．つまり，維持コストがかかる場合である $(x_i^{s-1}=1, x_i^s=1)$ と $(x_i^{s-1}=0, x_i^s=0)$ は共に差が 0 となり区別ができない．2 つ目の欠点は線形符号化法を値推移コスト付き動的 CSP のように多値変数を持つ問題に適用することは難しいという点である．その理由は，1 つ目の欠点と同様である．例えば，変数 x_i の値域 D_i が $\{1,2,3\}$ のように 3 つ以上ある場合，$(x_i^{s-1}=1, x_i^s=2)$ と $(x_i^{s-1}=2, x_i^s=3)$ は共に差が 1 となり区別できない．この問題点を回避するための単純な方法は，どの 2 つの値の差を取っても他の値と重ならないように値域を設定することである．例えば，ゴロム定規や指数関数を値域に用いる方法は解決方法の 1 つであるが，その性能は経験的にはあまり良くない．また，この解決方法では維持コストを表現することはできない．

5·2 2 次符号化法

これらの欠点を補うために，2 次符号化法を新たに導入する．

時刻 s の変数 x_i の値推移コストをコスト行列

$$\boldsymbol{F}_{i,s}=\begin{pmatrix} c_{1,1}^{i,s} & c_{1,2}^{i,s} & \cdots & c_{1,|D_i|}^{i,s} \\ c_{2,1}^{i,s} & c_{2,2}^{i,s} & \cdots & c_{2,|D_i|}^{i,s} \\ \vdots & \vdots & \ddots & \vdots \\ c_{|D_i|,1}^{i,s} & c_{|D_i|,2}^{i,s} & \cdots & c_{|D_i|,|D_i|}^{i,s} \end{pmatrix},$$

として表現する．このコスト行列では，例えば，コスト行列の $(1,2)$ の要素 $c_{1,2}^{i,s}$ は変数 x_i の値を 1 から 2 に変えたときにかかるコストと一致する．一方，コスト行列の $(2,2)$ の要素 $c_{2,2}^{i,s}$ は変数 x_i の値を 2 に維持した時に得られるコストと一致する．このコスト行列を用いて以下の 2 次計画問題を生成する．

$$\begin{aligned} \text{min.} \quad & \boldsymbol{x}_{i,s-1}^{\mathrm{T}} \cdot \boldsymbol{F}_{i,s} \cdot \boldsymbol{x}_{i,s} \\ \text{s. t.} \quad & \boldsymbol{x}_{i,s-1}, \boldsymbol{x}_{i,s} \in \{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_{|D_i|}\}. \end{aligned}$$

ここで，$\boldsymbol{e}_j$ は $|D_i|$ 次元の単位列ベクトルで，j 番目の要素が 1，他の要素がすべて 0 である．$\boldsymbol{x}_{i,s}$ と $\boldsymbol{x}_{i,s-1}$ はそれぞれ時刻 $s, s-1$ の決定変数で，値 $\boldsymbol{e}_j$ を変数 $\boldsymbol{x}_{i,s}$ に割り当てることは変数 x_i^s に値 j を割り当てることに相当する．また，各決定変数の右肩にある'T' はベクトル転置を表す．得られた問題の変数 $\boldsymbol{x}_{i,s}$ と $\boldsymbol{x}_{i,s-1}$ にある値を代入すると，その値の時のコストを計算できる．例えば，変数 x_i が時刻 $s-1$ において値 1 を選択した時のコストは，$\boldsymbol{x}_{i,s-1}=\boldsymbol{e}_1$ かつ $\boldsymbol{x}_{i,s}=\boldsymbol{e}_2$ を代入した結果得られる目的関数値 $c_{1,2}^{i,s}$ である．

この表現を用いると，値推移コスト付き k-stage 動的 CSP$(k, \mathcal{X}, \phi, \mathcal{F})$ は以下の通りになる．

$$\begin{aligned} \text{min.} \quad & \sum_{s=1}^{k-1}\sum_{i=1}^{n}(\boldsymbol{x}_{i,s-1}^{\mathrm{T}} \cdot \boldsymbol{F}_{i,s} \cdot \boldsymbol{x}_{i,s}) \\ \text{s. t.} \quad & \boldsymbol{x}_{i,s-1}, \boldsymbol{x}_{i,s} \in \{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_{|D_i|}\}, \\ & \quad i=1,\ldots,n,\ s=1,\ldots,k-1, \\ & \phi(s), \quad s=0,\ldots,k-1. \end{aligned} \tag{3}$$

ここで，式 (3) の $\phi(s)$ は時刻 s の CSP 問題例を返す関数で，この問題例に含まれる決定変数は $\boldsymbol{x}_{i,s}(i=1,...,n)$ のみで構成される．また，この関数 $\phi(s)$ が返す値である時刻 s の CSP 問題例は制約式に置き換わることに注意されたい．

この問題を解くことで値推移コスト付き k-stage 動的 CSP$(k, \mathcal{X}, \phi, \mathcal{F})$ の最適解を得られるが，直接解くことは難しいのでラグランジュ緩和・分解を適用することを考える．目的関数の値推移コストに関する項には異なる時刻における変数 $x_{i,s-1}$ と $x_{i,s}$ が含まれているため，この項を分解できれば，問題を時刻毎の部分問題に分解できることがわかる．そこで，変数 $\boldsymbol{x}_{i,s-1}, \boldsymbol{x}_{i,s}$ のコピーとしてそれぞれ補助変数 $\boldsymbol{\alpha}_{i,s}, \boldsymbol{\beta}_{i,s}$ を新たに導入し，これら 2 つの変数の値が一致するようにコピー制約を追加する．これらの補助変数 $\boldsymbol{\alpha}_{i,s}, \boldsymbol{\beta}_{i,s}$ が持つ値域は，それぞれ対応する変数 $\boldsymbol{x}_{i,s-1}, \boldsymbol{x}_{i,s}$ が持つ値域と同じである．さらに，目的関数の値推移コストに関する項の決定変数 $\boldsymbol{x}_{i,s-1}, \boldsymbol{x}_{i,s}$ をそれぞれ補助変数 $\boldsymbol{\alpha}_{i,s}, \boldsymbol{\beta}_{i,s}$ に置き換えると以下の問題が得られる．

$$\begin{aligned} \mathcal{P}\ :\ \text{min.} \quad & \sum_{s=1}^{k-1}\sum_{i=1}^{n}(\boldsymbol{\alpha}_{i,s}^{\mathrm{T}} \cdot \boldsymbol{F}_{i,s} \cdot \boldsymbol{\beta}_{i,s}) \\ \text{s. t.} \quad & \boldsymbol{x}_{i,s-1} - \boldsymbol{\alpha}_{i,s} = \boldsymbol{0}, \\ & \quad i=1,\ldots,n,\ s=1,\ldots,k-1, && (4) \\ & \boldsymbol{x}_{i,s} - \boldsymbol{\beta}_{i,s} = \boldsymbol{0}, \\ & \quad i=1,\ldots,n,\ s=1,\ldots,k-1, && (5) \\ & \phi(s), \quad s=0,\ldots,k-1, \\ & \boldsymbol{x}_{i,s}, \boldsymbol{x}_{i,s-1}, \boldsymbol{\alpha}_{i,s}, \boldsymbol{\beta}_{i,s} \in \{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_{|D_i|}\}, \\ & \quad i=1,\ldots,n,\ s=1,\ldots,k-1. \end{aligned}$$

コピー制約 (4)，(5) の導入により，値推移コストに関する項と各時刻の CSP 問題例の決定変数が互いに独立するため，コピー制約をラグランジュ緩和すれば値推移コストに関する項と各時刻の CSP 問題例を分解できる．以下では，各決定変数の値域に関する下 2 つの制約を省略する．

5·3 分　　解

問題 $\mathcal{P}$ を解くために，波多野，平山の方法 [波多野 11] と同様にラグランジュ分解法を適用する．

まず，コピー制約 (4) と (5) を緩和し，以下のようなラグランジュ緩和問題を生成する．

$$\begin{aligned}\mathcal{L} : L(\boldsymbol{\mu}) \;=\; \min. &\sum_{s=1}^{k-1}\sum_{i=1}^{n}(\boldsymbol{\alpha}_{i,s}^{\mathrm{T}}\cdot\boldsymbol{F}_{i,s}\cdot\boldsymbol{\beta}_{i,s})\\ &+\sum_{s=1}^{k-1}\sum_{i=1}^{n}\boldsymbol{\mu}_{\alpha,i,s}^{\mathrm{T}}(\boldsymbol{x}_{i,s-1}-\boldsymbol{\alpha}_{i,s})\\ &+\sum_{s=1}^{k-1}\sum_{i=1}^{n}\boldsymbol{\mu}_{\beta,i,s}^{\mathrm{T}}(\boldsymbol{x}_{i,s}-\boldsymbol{\beta}_{i,s}),\\ \text{s. t.}\quad &\phi(s),\quad s=0,\ldots,k-1.\end{aligned}$$

ここで，$\boldsymbol{\mu}_{\alpha,i,s}$ と $\boldsymbol{\mu}_{\beta,i,s}$ は緩和問題を生成するときに導入されるラグランジュ乗数ベクトルで，ともに $|D_i|$ 次元の実数値を持つ列ベクトルである．

また，$\boldsymbol{\mu}$ は $\boldsymbol{\mu}_{\alpha,1,1},\ldots,\boldsymbol{\mu}_{\alpha,n,k-1},\boldsymbol{\mu}_{\beta,1,1},\ldots,\boldsymbol{\mu}_{\beta,n,k-1}$ の集合とする．ラグランジュ緩和問題 $\mathcal{L}$ の最適値 $L(\boldsymbol{\mu})$ は，$\boldsymbol{\mu}_{\alpha,i,s}$ と $\boldsymbol{\mu}_{\beta,i,s}$ の値により変化する．さらに，$L(\boldsymbol{\mu})$ は，原問題 $\mathcal{P}$ に対する双対問題であるため，弱双対定理により，任意の $\boldsymbol{\mu}_{\alpha,i,s}$ と $\boldsymbol{\mu}_{\beta,i,s}$ の値に対して，常に $\mathcal{P}$ の最適値に対する下界値となる [Bertsekas 99].

次に，ラグランジュ緩和問題 $\mathcal{L}$ を以下の $k+1$ 個の部分問題に分解する．その結果，まず，補助変数 $\boldsymbol{\alpha}_{i,s}$ と $\boldsymbol{\beta}_{i,s}$ に関する部分問題

$$\begin{aligned}L^{aux}(\boldsymbol{\mu}) \;=\; \min. &\sum_{s=1}^{k-1}\sum_{i=1}^{n}(\boldsymbol{\alpha}_{i,s}^{\mathrm{T}}\cdot\boldsymbol{F}_{i,s}\cdot\boldsymbol{\beta}_{i,s})\\ &-\sum_{s=1}^{k-1}\sum_{i=1}^{n}(\boldsymbol{\mu}_{\alpha,i,s}^{\mathrm{T}}\boldsymbol{\alpha}_{i,s}+\boldsymbol{\mu}_{\beta,i,s}^{\mathrm{T}}\boldsymbol{\beta}_{i,s}),\end{aligned}\tag{6}$$

が得られ，次に，時刻 0 の決定変数に関する部分問題

$$\begin{aligned}L^{0}(\boldsymbol{\mu}) \;=\; \min. &\sum_{i=1}^{n}\boldsymbol{\mu}_{\alpha,i,1}^{\mathrm{T}}\boldsymbol{x}_{i,0}\\ \text{s. t.}\quad &\phi(0),\end{aligned}\tag{7}$$

が得られる．さらに，$s\in\{1,...,k-2\}$ の各時刻の決定変数に関する部分問題

$$\begin{aligned}L^{s}(\boldsymbol{\mu}) \;=\; \min. &\sum_{i=1}^{n}(\boldsymbol{\mu}_{\beta,i,s}^{\mathrm{T}}+\boldsymbol{\mu}_{\alpha,i,s+1}^{\mathrm{T}})\boldsymbol{x}_{i,s}\\ \text{s. t.}\quad &\phi(s),\end{aligned}\tag{8}$$

がそれぞれ得られ，最後に，時刻 $k-1$ の決定変数に関する部分問題

$$\begin{aligned}L^{k-1}(\boldsymbol{\mu}) \;=\; \min. &\sum_{i=1}^{n}\boldsymbol{\mu}_{\beta,i,k-1}^{\mathrm{T}}\boldsymbol{x}_{i,k-1}\\ \text{s. t.}\quad &\phi(k-1),\end{aligned}\tag{9}$$

が得られる．

ある $\boldsymbol{\mu}$ の値における (7) から (9) の部分問題は，対応する重み付き CSP を解くことで最小化できる．例えば，式 (9) では，$\phi(k-1)$ 内のすべての制約を満たしつつ，$\boldsymbol{\mu}_{\beta,i,k-1}$ の重みをもつ単項制約（unary constraint）のコストの合計値を最小とするような値を変数 $x_{1,k-1},\ldots,x_{n,k-1}$ に割り当てる．一方，式 (6) の最小化は $\boldsymbol{\alpha}_{i,s}$ と $\boldsymbol{\beta}_{i,s}$ の値を i と s に関してすべて列挙するだけである．値域のサイズを d とするとき，列挙にかかる計算コストは $O((k-1)nd^2)$ であるが，他の部分問題を解くコストと比較すると取るに足らない計算量である．

5·4 ラグランジュ双対問題

5·3 節で議論したように，ある $\boldsymbol{\mu}$ の値のもとで部分問題 $L^{aux}(\boldsymbol{\mu}),L^{0}(\boldsymbol{\mu}),\ldots,L^{k-1}(\boldsymbol{\mu})$ の最適値が得られる．明らかに，これらの最適値の合計が全体のラグランジュ緩和問題 $\mathcal{L}$ の最適値 $L(\boldsymbol{\mu})$ と一致している．つまり，

$$L(\boldsymbol{\mu})=L^{aux}(\boldsymbol{\mu})+\sum_{s=0}^{k-1}L^{s}(\boldsymbol{\mu}),$$

といえる．

$L(\boldsymbol{\mu})$ はいかなる $\boldsymbol{\mu}$ の値においても原問題 $\mathcal{P}$ の最適解に対する下界値を与えることから，その下界値を最大化し，$\mathcal{P}$ の最適値へと近づくような $\boldsymbol{\mu}$ の値を探索することが次の目的となる．この問題はラグランジュ双対問題と呼ばれ，以下のように定義される．

$$\mathcal{D} : \max.\quad L^{aux}(\boldsymbol{\mu})+\sum_{s=0}^{k-1}L^{s}(\boldsymbol{\mu}).\tag{10}$$

この問題は無制約の最大化問題で，その決定変数は $\boldsymbol{\mu}$ である．

ラグランジュ双対問題を解く一般的な方法として劣勾配法 [Bertsekas 99] がある．この方法は山登り法の一種であり，(10) の目的関数値を利用する．以下は，ラグランジュ双対問題に対する劣勾配法の概要である．

Step 1: $\boldsymbol{\mu}$ のすべての要素に 0 を代入する．

Step 2: すべての部分問題 (6) から (9) を解く．特に部分問題 (7) から (9) に関しては既存の重み付き CSP ソルバーを用いて解く．

Step 3: 原問題 $\mathcal{P}$ の最適値に対する最も低い上界値 *BestUB* と最も高い下界値 *BestLB* および *BestUB* を与える $\mathcal{P}$ の実行可能解 M を求める．

Step 4: もし終了条件を満たすなら，*BestLB*，*BestUB*，M を返す．満たしていなければ，$\boldsymbol{\mu}$ の値を更新し，Step 2 へ戻る．

この手順は終了条件を満たすまで Step 2 から Step 4 を繰り返す．また，各繰り返しにおいて，求めた実行可能解に対応した上界値と下界値を求めることができる．

Step3 では，ある $\boldsymbol{\mu}$ の値のもとでの部分問題 $L^{aux}(\boldsymbol{\mu})$，$L^{0}(\boldsymbol{\mu}),\ldots,L^{k-1}(\boldsymbol{\mu})$ の最適値の合計を計算することによ

り下界値が得られる．一方，上界値は原問題 $\mathcal{P}$ の実行可能解 M から得られる．この実行可能解は，緩和された (4) と (5) の制約式を満たすように $\boldsymbol{\alpha}_{i,s}$ の値を $\boldsymbol{x}_{i,s-1}$ の値に，$\boldsymbol{\beta}_{i,s}$ の値を $\boldsymbol{x}_{i,s}$ の値に置き換えることにより得られる．この上界値は系列 $\{\phi(0), \phi(1), \ldots, \phi(k-1)\}$ に対する解から得られる値推移コストの合計値と一致する．上界値，下界値がそれ以前の値よりもよい場合はこれらの値をそれぞれ *BestLB*，*BestUB* として記録する．

Step 4 では，$\mathcal{P}$ の最適解が発見された場合，手順を終了する．明らかに，*BestLB* と *BestUB* が共に同じ値に収束する場合，その *BestUB* を与える実行可能解が最適解である．一方，最適解が発見できない場合でも，ある制限時間を超えた場合は手順を終了することができる．制限時間を超えた場合でも，上界の質である $BestUB/BestLB$ を与える実行可能解を得ることができる．動的な問題では，一般的にある制限時間内に困難な問題を解くことが求められているため，この性質は重要であると考えられる．

Step 4 において，もし終了条件を満たさない場合は $\boldsymbol{\mu}$ の値を更新し，Step 2 へと戻る．$\boldsymbol{\mu}$ の更新時には，以下のような慣習的な計算式を用いる．この方法では，Step 2 の部分問題の解を用いて，まず全ての i と s に対する劣勾配ベクトル $\boldsymbol{G}_{\alpha,i,s}$ と $\boldsymbol{G}_{\beta,i,s}$ を以下のように求める．

$$\boldsymbol{G}_{\alpha,i,s} = \boldsymbol{x}_{i,s-1} - \boldsymbol{\alpha}_{i,s},$$
$$\boldsymbol{G}_{\beta,i,s} = \boldsymbol{x}_{i,s} - \boldsymbol{\beta}_{i,s}.$$

これらの値はそれぞれ $\mathcal{L}$ の目的関数にある $\boldsymbol{\mu}^{\mathrm{T}}_{\alpha,i,s}$ と $\boldsymbol{\mu}^{\mathrm{T}}_{\beta,i,s}$ の係数と一致する．求めた劣勾配ベクトルを用いて，S 任意の i と s に対する $\boldsymbol{\mu}_{\alpha,i,s}$ と $\boldsymbol{\mu}_{\beta,i,s}$ を以下のように更新する．

$$\boldsymbol{\mu}_{\alpha,i,s} \leftarrow \boldsymbol{\mu}_{\alpha,i,s} + D \cdot \boldsymbol{G}_{\alpha,i,s},$$
$$\boldsymbol{\mu}_{\beta,i,s} \leftarrow \boldsymbol{\mu}_{\beta,i,s} + D \cdot \boldsymbol{G}_{\beta,i,s}.$$

ここで，D はステップ長と呼ばれるスカラ値であり，以下の式で求める．

$$D = \frac{\pi(BestUB - BestLB)}{\sum_{i,s} \boldsymbol{G}^{\mathrm{T}}_{\alpha,i,s} \cdot \boldsymbol{G}_{\alpha,i,s} + \sum_{i,s} \boldsymbol{G}^{\mathrm{T}}_{\beta,i,s} \cdot \boldsymbol{G}_{\beta,i,s}}.$$

上式の π もまたスカラ値であり，初期値を 2 として繰り返しが進む毎に徐々に減少していく．

これらの式は，劣勾配ベクトルが正（負）であれば，次回の $\boldsymbol{\mu}_{\alpha,i,s}$，$\boldsymbol{\mu}_{\beta,i,s}$ を増加（減少）させることを意味する．つまり，劣勾配ベクトルの値は $\mathcal{L}$ の目的関数内の $(\boldsymbol{x}_{i,s-1} - \boldsymbol{\alpha}_{i,s})$，$(\boldsymbol{x}_{i,s} - \boldsymbol{\beta}_{i,s})$ と一致するため，この値がもし正（負）であれば，対応する $\boldsymbol{\mu}_{\alpha,i,s}$，$\boldsymbol{\mu}_{\beta,i,s}$ を増加（減少）させることにより，次回に得られる目的関数値が増加することが期待できる．このように $\boldsymbol{\mu}_{\alpha,i,s}$ と $\boldsymbol{\mu}_{\beta,i,s}$ の値を更新することで，Step2 における (6) から (9) の部分問題の単項制約にかかる重みが $\boldsymbol{\mu}_{\alpha,i,s}$ から $\boldsymbol{\mu}_{\alpha,i,s} + D \cdot \boldsymbol{G}_{\alpha,i,s}$，また，$\boldsymbol{\mu}_{\beta,i,s}$ から $\boldsymbol{\mu}_{\beta,i,s} + D \cdot \boldsymbol{G}_{\beta,i,s}$ へと変化する．

6. 実　　験

提案手法を評価するために，幾つかの実験を行った．

6·1 線形符号化法との比較

まず，線形符号化後にラグランジュ分解を適用する方法 (LD(L) と表記) と 2 次符号化後にラグランジュ分解を適用する方法 (LD(Q) と表記) の 2 つを比較した．比較に使用する問題は [波多野 11] で使用されている問題であるランダムな値変更コスト付き動的 SAT を使用し，その問題例 $(k, \mathcal{X}, \phi, \mathcal{F})$ を以下のように生成した．

- 系列サイズ k の範囲は $\{10, 15, \ldots, 35\}$ である．
- $\mathcal{X}$ は 100 個のブール変数の集合 $\{x_1, x_2, \ldots, x_{100}\}$ である．
- 各時刻 s における ϕ は，SATLIB[Hoos 00] の `uf100-430` からランダムに選択された部分問題を返す．この部分問題は 100 変数，430 節を持つ充足可能な 3SAT の CNF 論理式である．
- $\mathcal{F}$ の各要素 $f_{i,s}$ は，時刻 s における変数 x_i の値変更コストとして $\{1, 2, \ldots, 10^6\}$ からランダムに選択された整数値を返し，維持コストとして 0 を返す．

つまり，系列サイズが k の問題例は，変数の数が $100k$，ハード節の数が $430k$，ソフト節の数が $100(k-1)$ となる．この問題は 2 値変数からなり，維持コストが 0 であるため，その問題例はどちらの符号化法でも解くことができる．なお，実験では各系列サイズ k に対して，30 個の問題例を生成した．

2 つの符号化法では，部分問題を解くために SAT4J [Berre 10] の SAT 2011 competition バージョンを用いた．各部分問題の節はハード節であるため，SAT4J のような”SAT ベース”のソルバーが適切である．さらに，線形符号化法においては，ラグランジュ分解法と局所探索法を組み合わせた．つまり，Step 3 で発見した実行可能解を局所探索法を用いて 10 万回変数の値を変化させることにより改良した．この局所探索法による改善は，下界値が改善したときに IROTS[Smyth 03] を用いて行った．IROTS は確率的局所探索法に基づくソルバーでタブーサーチを実装している．実験では，IROTS を含む幾つかの SAT に関する発見的解法が実装された UBCSAT のバージョン 1.1[Tompkins 05] をダウンロードし，重みつき最大 SAT を解くためのオプションである’-w’ をつけて IROTS を実行した．

この実験は Intel Corei7-2600@3.4GHz (4 cores, 8GB memory) の Ubuntu 10.10 (64bit) を用いて行われた．また，ソースコードは JAVA で書かれ，JDK1.6.0_20 でコンパイルされた．

実験では，評価指標として，制限時間と系列サイズを変化させたときの解の質（$BestUB/BestLB$）を用いた．プランニング等の時系列に関するアプリケーションが多い動的 CSP では，どの程度の時間でどの程度の解が得

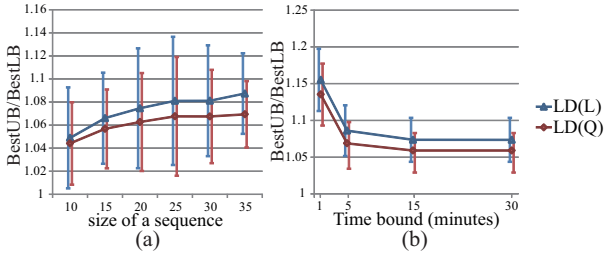

図 2 ランダムな値変更コスト付き動的 SAT に対する LD(L) と LD(Q) の比較: (a) 系列サイズを変化させ，制限時間を 5 分に固定したときの解の質の平均．(b) 制限時間を変化させ，系列のサイズを 35 に固定したときの解の質の平均

られるかを測ることは重要であるため，評価指標として，制限時間を用いることは理にかなっていると考えられる．また，問題サイズに対して頑健性を測る指標として，系列サイズを用いた．

図 2 は，(a) 系列サイズを変化させ，制限時間を 5 分に固定した場合と (b) 系列サイズを 35 に固定し，制限時間を変化させた場合のエラーバー付きの解の質 (*BestUB*/*BestLB*) の平均値を表している．明らかに，解の質は 1 に近いほど，性能が良いことを示している．この図からわかるように，局所探索法による解の改善を行わなくても，2 次符号化法は線形符号化法よりも平均的に質の良い解が得られていることがわかる．その理由として，ラグランジュ乗数の数の増加が挙げられる．一般的には，ラグランジュ乗数の数の増加は緩和した制約の数の増加を意味するため，下界値の精度を下げてしまうと考えられる．しかし，線形符号化法と 2 次符号化法を比較すると，緩和される制約 (2) と (4)，(5) は表記が異なるが本質的には同じである．一方，ラグランジュ乗数の数を比較すると，線形符号化法は各時刻間の変数に対して 1 つのラグランジュ乗数が導入される．例えば，k 系列で各時刻が n 変数，値域サイズが d である値変更コスト付き動的 SAT であれば，導入されるラグランジュ乗数の数は $n(k-1)$ となる．それに対して，2 次符号化法では，各変数の値域の各値ごとに 1 つのラグランジュ乗数が導入される．つまり，先ほどの例だとその数は $2n(k-1)d$ となる．このラグランジュ乗数の数の増加は，値域の各値に対して，劣勾配法における細かな方向決めを可能とするため，より良い上界値と下界値を求めることに繋がると考えられる．

6·2 一般的な重み付き CSP ソルバーとの比較

次に，2 次符号化法を用いたラグランジュ分解法と一般的な重み付き CSP ソルバーとの比較を行った．[波多野 11] で提案された重み付き部分 MaxSAT に変換する方法と同様に，値推移コスト付き動的 CSP の問題例は，ある変数の値推移コストをその変数のコピー変数との間のソフト制約とみなすことにより，1 つの大きな重み付き CSP に変換できる．実験では，厳密解法のソルバーとして CSP competition 等で活躍している TOULBAR2 を使用した．一方，Weighted CSP の発見的解法は問題の構造に依存しやすいことから，DynCSP の構造に合わせて自身で作成した LOCALSEARCH を使用した．このうち，TOULBAR2 は，深さ優先の分枝限定法とロシアンドール探索法を用いている．開発者のページから TOULBAR2 のバージョン 0.9.5.0 をダウンロードし，1 つ目の実験と同様の実験環境においてデフォルトの設定で実験した．さらに，ラグランジュ分解法における (6) から (9) の部分問題を解く場合にも同様のソルバーを用いた．LOCALSEARCH は WALKSAT[Selman 95] と同様の発見的解法を実装した確率的局所探索ソルバーで，初期解として系列の各 CSP を別々に解くことにより得られた解を用いているため，制限時間内に初期解が得られれば必ず実行可能解が得られる．

比較に使用した問題例は，ランダムバイナリ値推移コスト付き動的 CSP と，実問題に近いターゲット追跡問題の 2 つである．ランダムバイナリ 値推移コスト付き動的 CSP では，各問題例 $(k, \mathcal{X}, \phi, \mathcal{F})$ を以下のように生成した．

- 系列のサイズ k の範囲は $\{10, 15, \ldots, 35\}$ である．
- $\mathcal{X}$ は 30 個の変数の集合 $\{x_1, x_2, \ldots, x_{30}\}$ とし，各変数の値域は 15 である．
- 各時刻 s における ϕ は Model RB [Xu 07] に基づいて生成した充足可能な部分問題で，各部分問題は 30 変数，値域サイズは 15，284 のハード制約を持つ．
- $\mathcal{F}$ の各要素 $f_{i,s}$ は $\{1, 2, \ldots, 10^6\}$ からランダムに選択された整数値を推移コストとして返す．

つまり，系列サイズが k の問題例は，変数の数が $30k$，ハード制約の数が $284k$，ソフト制約の数が $30(k-1)$ となる．ランダムバイナリ値推移コスト付き動的 CSP に関しても，系列サイズ k 毎に 30 の問題例を生成した．

重み付き CSP の問題例は Model RB の問題例を部分問題として一部含んでいるため，TOULBAR2 が解くのに時間がかかることから，TOULBAR2 の各実行毎の制限時間を 15 分に延長した．図 3(a) からこの問題においては，LD(Q) は一般的に TOULBAR2 よりも質の良い解が得られていることがわかる．図 3(b) では，TOULBAR2 の 1 分，5 分の結果がないが，これは全 30 問のうち，ある問題例に対して，制限時間内に実行可能解が得られなかったことを意味する．

一方，LD(Q) は，1 分の時点で実行可能解を発見している．この違いは系列サイズが増加するほど顕著に表れる．なぜなら，LD(Q) は系列の各部分問題を個別に解くことができ，かつその解が実行可能解となっているが，TOULBAR2 は全体の問題を解く必要があるためである．

次に，3 章で説明したターゲット追跡問題を使用する．この問題は容易に値推移コスト付き動的 CSP にモデル化できる．具体的には，ターゲット追跡問題の問題例 $(k, \mathcal{X}, \phi, \mathcal{F})$ を以下のように生成した．

- 系列サイズ k の範囲は $\{10, 15, \ldots, 35\}$ である．
- $\mathcal{X}$ は 25 個の変数の集合 $\{x_1, x_2, \ldots, x_{25}\}$ とし，各変

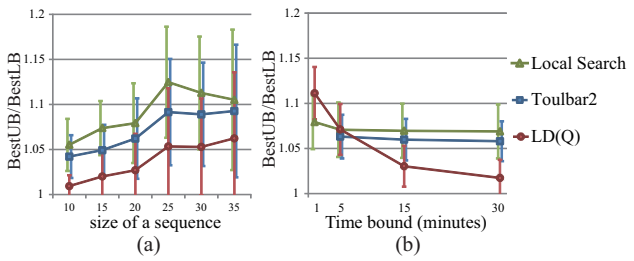

図 3 ランダムバイナリ値推移コスト付き動的 CSP に関する LD(Q) と TOULBAR2 の比較：(a) 系列サイズを変化させ，制限時間を 15 分に固定したときの解の質の平均．(b) 制限時間を変化させ，系列のサイズを 35 に固定したときの解の質の平均

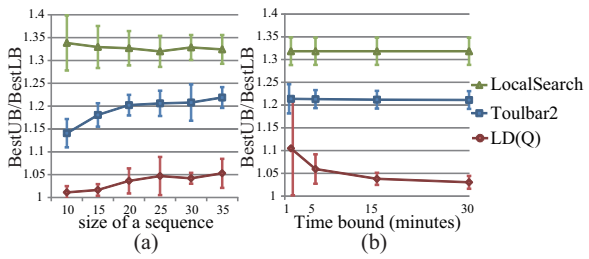

図 4 ターゲット追跡問題における LD(Q) と，TOULBAR2 と LOCALSEARCH との比較: (a) 系列サイズを変化させ，制限時間を 5 分に固定したときの解の質の平均． (b) 系列サイズを 35 に固定し，制限時間を変化させたときの解の質の平均．

数の値域は $\{\mathrm{NE}, \mathrm{NW}, \mathrm{SW}, \mathrm{SE}, \mathrm{OFF}\}$ である．

- 各時刻 s における ϕ はターゲットのスナップショットから得られる位置をもとに生成した部分問題で，各部分問題は 25 変数，値域サイズ 5，約 75 のハード制約を持つ．
- $\mathcal{F}$ の各要素 $f_{i,s}$ を以下のように設定する．
  - もしセンサが次の時刻で反対側の領域に移動した場合，変更コスト 200,000 を返す．
  - もしセンサが次の時刻で隣接する領域に移動した場合，変更コスト 100,000 を返す．
  - もしセンサを ON にし続けた場合，維持コスト 50,000 を返す．
  - もしセンサを OFF にし続けた場合，維持コスト 0 を返す．
  - もしセンサを ON から OFF に変更した場合，変更コスト 0 を返す．
  - もしセンサを OFF から ON に変更した場合，変更コスト 100,000 を返す．

つまり，系列サイズが k の問題例は，変数の数が $25k$，ハード制約の数が約 $75k$，ソフト制約の数が $25(k-1)$ となる．ターゲット追跡問題に関しても同様に，系列サイズ k 毎に 30 問の問題例を生成した．

また，ターゲット追跡問題では，ラグランジュ分解法の部分問題を解くソルバーとして TOULBAR2 のバージョン 0.9.5.0 を用いた．

実験結果を図 4 に示した．明らかに，ターゲット追跡問題においても LD(Q) は他のソルバーよりも質の良い解を与えていることがわかる．特に図 4(b) からわかるように，短い時間で質の良い解に収束している．この結果は，動的な性質を持つ問題を解く上で重要である．

一方，TOULBAR2 はランダムな問題に強いソルバーであるが，ターゲット追跡問題だけではなく，部分問題がランダムな問題であるランダムバイナリ CSP においても，制限時間の増加に対してあまり解の質が改善されていない．なぜなら，動的 CSP は各部分問題が値推移コストにより緩く結合された問題であるため，全体としては構造を持つ問題となるためである．次に，2 つの問題の実験結果を比較すると，ランダムバイナリ CSP とターゲット追跡問題で同等程度の解の質が得られている．しかし，ランダムバイナリ CSP の作成に用いた Model RB は相転移領域の問題であるため，求解に時間がかかるということを考慮すると，提案手法も値変更コスト付き動的 SAT と同様，系列の個々の部分問題が相転移領域である問題に対して強いといえる．一方，LD(Q) は部分問題が相転移領域から離れた問題，つまり，実行可能領域が小さい問題に対して弱いと考えられる．

7. お わ り に

本論文では，まず値変更コスト付き動的 SAT に対して，実世界の動的な問題の問題設定を表現するために CSP への拡張と維持コストの導入を提起し，これらの問題提起をふまえて値変更コスト付き動的 SAT を拡張した値推移コスト付き動的 CSP を定式化した．次に，値推移コスト付き動的 CSP に 2 次符号化法を導入した．この 2 次符号化法は，値推移コストを 2 次計画問題へ符号化し，ラグランジュ分解の適用を可能とした．実験では，2 次符号化法の導入により，値変更コスト付き動的 SAT に関して解の質を改善できた．また，2 次符号化法を用いたラグランジュ分解法は他の重み付き CSP ソルバーより良質な解をより早く得られており，特に実世界の問題に近いターゲット追跡問題に関しては短い時間で質の良い解に収束していることがわかった．この性質は動的な問題において，非常に重要であると考えられる．

今後は，提案方法を CSP 問題例の系列のみならず，CSP 問題例の”木”も解くことができるように拡張したい．例えば，ターゲット追跡問題において，ターゲットの動きがいくつか予測される場合，その CSP 問題例は時刻 1 の CSP を根とした複数の系列を持つ”木”として表現できる．このような問題では，各系列（予測）に対して確率が付与されており，その目的は値推移コストの合計値の期待値を最小化することであると考えられる．この問題に対してラグランジュ分解法に基づいた解法を適用するためには，確率への対応と，CSP の木を解くことが可能な符号化の導入を考える必要がある．

◇ 参 考 文 献 ◇

[Berre 10] Berre, D. L., Parrain, A., Lille, U., and France, N.: The Sat4j library, release 2.2 system description, in *Journal on Satisfiability, Boolean Modeling and Computation*, pp. 59–64 (2010)

[Bertsekas 99] Bertsekas, D. P.: Nonlinear Programming, Athena Scientific, 2nd edition (1999)

[Bessière 91] Bessière, C.: Arc-consistency in dynamic constraint satisfaction problems, in *Proceedings of the ninth National conference on Artificial intelligence*, pp. 221–226, AAAI Press (1991)

[Bofill 10] Bofill, M., Busquets, D., and Villaret, M.: A declarative approach to robust weighted Max-SAT, in *Proceedings of the 12th international ACM SIGPLAN symposium on Principles and practice of declarative programming*, pp. 67–76 (2010)

[Fargier 96] Fargier, H., Lang, J., and Schiex, T.: Mixed constraint satisfaction: a framework for decision problems under incomplete knowledge, in *Proceedings of the thirteenth national conference on Artificial intelligence*, pp. 175–180 (1996)

[Freuder 91] Freuder, E. C.: Eliminating interchangeable values in constraint satisfaction problems, in *Proceedings of the ninth National conference on Artificial intelligence*, pp. 227–233 (1991)

[Ginsberg 98] Ginsberg, M. L., Parkes, A. J., and Roy, A.: Supermodels and robustness, in *Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, pp. 334–339 (1998)

[波多野 11] 波多野 大督, 平山 勝敏：値変更コスト付き動的 SAT の定式化とその解法, 人工知能学会論文誌, Vol. 26, No. 6, pp. 682–691 (2011)

[服部 05] 服部 宏充, 磯村 厚誌, 伊藤 孝行, 大囿 忠親, 新谷 虎松：動的重み付き最大制約充足問題に基づくナーススケジューリングシステム- 暫定制約の導入に基づく界の安定性の実現-, 人工知能学会論文誌, Vol. 20, No. 1, pp. 25–35 (2005)

[Hoos 00] Hoos, H. H. and Stutzle, T.: SATLIB: An Online Resource for Research on SAT, pp. 283–292, IOS Press (2000)

[Schiex 93] Schiex, T. and Verfaillie, G.: Nogood Recording for Static and Dynamic Constraint Satisfaction Problems, *International Journal of Artificial Intelligence Tools*, Vol. 3, pp. 48–55 (1993)

[Selman 95] Selman, B., Kautz, H., and Cohen, B.: Local Search Strategies for Satisfiability Testing, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, Vol. 26 pp. 521–532 (1995)

[Smyth 03] Smyth, K., Hoos, H. H., and Stützle, T.: Iterated robust tabu search for MAX-SAT, in *Proceedings of the 16th Canadian society for computational studies of intelligence conference on Advances in artificial intelligence*, pp. 129–144 (2003)

[Tompkins 05] Tompkins, D. A. D. and Hoos, H. H.: UBCSAT: An Implementation and Experimentation Environment for SLS Algorithms for SAT and MAX-SAT, in *Revised Selected Papers from the Seventh International Conference on Theory and Applications of Satisfiability Testing*, Vol. 3542 of *Lecture Notes in Computer Science*, pp. 306–320 (2005)

[Verfaillie 94] Verfaillie, G. and Schiex, T.: Solution reuse in dynamic constraint satisfaction problems, in *Proceedings of the twelfth national conference on Artificial intelligence*, pp. 307–312 (1994)

[Wallace 98] Wallace, R. J. and Freuder, E. C.: Stable Solutions for Dynamic Constraint Satisfaction Problems, in *Proceedings of the 4th International Conference on Principles and Practice of Constraint Programming*, pp. 447–461 (1998)

[Walsh 02] Walsh, T.: Stochastic Constraint Programming, in Proceedings of the 15th European Conference on Artificial Intelligence, pp. 111–115 (2002)

[Xu 07] Xu, K., Boussemart, F., Hemery, F., and Lecoutre, C.: Random constraint satisfaction: Easy generation of hard (satisfiable) instances, *Artificial Intelligence*, Vol. 171, No. 8-9, pp. 514–534 (2007)

〔担当委員：服部 宏充〕

2012 年 7 月 7 日 受理

著 者 紹 介

波多野 大督（学生会員）

2008 年 神戸大学海事科学部海上輸送システム学課程卒業．2010 年 神戸大学大学院海事科学研究科海事科学専攻博士課程前期課程修了．2010 年 同研究科海事科学専攻博士課程後期課程に進学．2012 年 JAWS-2012 学生優秀論文賞受賞．制約充足，組み合わせ最適化に興味を持つ．

平山 勝敏（正会員）

1990 年 大阪大学基礎工学部制御工学科卒．1992 年 同大大学院基礎工学研究科博士前期課程修了．1995 年 同大大学院基礎工学研究科博士後期課程修了．1995 年 神戸商船大学助手．1997 年 同講師．2001 年 同助教授．2003 年 神戸大学海事科学部助教授（神戸大学と神戸商船大学の統合による）．2007 年 神戸大学大学院海事科学研究科准教授．1999 年–2000 年 カーネギーメロン大学ロボティクス研究所客員研究員（文部省在外研究員）．マルチエージェントシステム，制約充足/SAT，組合せ最適化に関する研究に従事．博士（工学）．2010 年 IFAAMAS Influential Paper Award 受賞．2011 年 CP-2011 Best Application Paper Award 受賞．電子情報通信学会，情報処理学会，人工知能学会，日本オペレーションズリサーチ学会，AAAI 各会員．