値変更コスト付き動的ＳＡＴの定式化とその解法

波多野，大督
平山，勝敏

(Citation)
人工知能学会論文誌,26(6):682-691

(Issue Date)
2011

(Resource Type)
journal article

(Version)
Version of Record

(URL)
https://hdl.handle.net/20.500.14094/90002957

原著論文　　Original Paper

値変更コスト付き動的SATの定式化とその解法

Dynamic SAT with Decision Change Costs: Formalization and Solutions

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

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

keywords: SAT, Dynamic SAT, Weighted Partial MaxSAT, Lagrangian decomposition, planning

Summary

We address a dynamic decision problem in which decision makers must pay some costs when they change their decisions along the way. We formalize this problem as Dynamic SAT (DynSAT) with decision change costs, whose goal is to find a sequence of models that minimize the aggregation of the costs for changing variables. We provide two solutions to solve a specific case of this problem. The first uses a Weighted Partial MaxSAT solver after we encode the entire problem as a Weighted Partial MaxSAT problem. The second solution, which we believe is novel, uses the Lagrangian decomposition technique that divides the entire problem into sub-problems, each of which can be separately solved by an exact Weighted Partial MaxSAT 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 random problem and the target tracking problem. The experimental results show that a solver based on Lagrangian decomposition performs better for the random problem and competitively for the target tracking problem.

1. は じ め に

SAT(充足可能性問題) は理論と実践において疑いなく重要な決定問題の１つである．すべての NP 問題が SAT 問題に変換できることはよく知られている．さらに，MiniSAT [Een 03] のような最先端のソルバーを使用することで，何 100 万もの変数を含む問題をほんの数分で解くことができる．近年の SAT ソルバーの成果により，実世界に即した問題を扱えるような SAT の拡張の必要性が大きくなってきている [平山 10]．動的 SAT [Hoos 00a] はそのような拡張の１つで，実世界の動的性質をモデル化することを目的としている．動的 SAT は動的 CSP [Dechter 88] の一種と考えられる．動的 SAT と動的 CSP では問題例の系列が与えられ，その系列に対する解の系列を求める．

動的 CSP に対するアプローチは大きく 2 つに分類される [Verfaillie 05]．

1 つは reactive アプローチで，起こりうる未来の変化に対する知識を使用しない方法である．目的は現在の問題例からある変化が起こることによって得られる新しい問題例の解を求めることである．reactive アプローチの重要なアイデアは何を再利用するかである．解の再利用 [Verfaillie 94] と推論の再利用 [Bessiere 91, Schiex 93] はその典型的な方法である．reactive アプローチの利点は，問題にどんな変化が起こっても反応できる点であるが，欠点は将来に起こりうる変化についての知識を使用しないために，求めた解が一般的に頑健ではないことである．

一方，もう 1 つは proactive アプローチで，将来起こりうる変化についての知識を使用する方法である．この将来の知識を使用することで，起こりうる変化に強い解が求められる可能性が大きい．robust 解 [Wallace 98, Fragier 96, Walsh 02] と flexible 解 [Freuder 91, Ginsberg 98, Bofill *et al.*10] がこの例となる．しかし，これらの解を求める計算コストが問題で，一般に通常の CSP を解くよりもコストが高い可能性がある．

本論文では，新しい動的な意思決定問題を提案する．この問題では意思決定者は意思決定が変化したとき，いくらかのコストを払わなければならない．実世界ではこのようなコストはプランニングやスケジューリングの段取りコストとしてよく見られる．例えば，病院のスタッフの一ヶ月のスケジュールを決めなければならないと仮定する．来週に重要な外科手術があるなど，起こりうる将来の出来事の情報を知っている場合，毎日の仕事が順調にいくような，すなわち段取りコストが最小となるようなスケジュールを組めるかもしれない [服部 05]．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が，CSP や SAT のような一般的な問題でこれを扱った研究はあまり見られない．

例外として，動的 CSP に対する最小変化コストを求め

る研究がある [Roos 00, Ran 02]．この研究では値が変化する変数の個数を最小にすることが目的で，現在の解からのハミング距離が最小となるような新しい解を求めるアルゴリズムが提案されている．また，CSP に対するより変化の多い解やより変化の少ない解を求める方法 [Hebrand 05] や与えられた部分割り当てが高々ある特定の数しか違わないような解を求める distance-SAT [Bailleux 06] も提案されている．なお，これらの研究は基本的に reactive アプローチに分類される．

本論文では新しい proactive アプローチの 1 つとして，値変更コスト付き動的 SAT を提案する．これは，SAT 問題例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求める問題である．CSP ではなく SAT である理由としては，SAT には効率的なソルバーが多く，また効果的な符号化方法が多いことが挙げられる．値変更コスト付き動的 SAT は SAT 問題例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求める問題である．

さらに本論文では，値変更コスト付き動的 SAT に対する 2 つの解法を提案する．1 つ目の解法は問題例を重み付き部分 MaxSAT に変換し，重み付き部分 MaxSAT 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．この方法は問題例を複数の部分問題に分解し，各部分問題を重み付き部分 MaxSAT ソルバーで解くことで，上界と下界を得る方法である．なお，ラグランジュ分解を制約問題に適用した先行研究としては，重み付き制約充足問題を扱う離散 MRF 最適化 [Komodakis 07]，分散制約最適化 [Vinyals 10]，重み付き MaxSAT 問題を解く Multi-MaxSAT[黒田 09] などがある．特に，Multi-MaxSAT は，提案方法と同様に，ラグランジュ分解を用いて問題例を複数の部分問題に分解する方法であるが，Multi-MaxSAT が共有変数に関するハード節を緩和するのに対して，提案方法はソフト節を緩和するという違いがある．

以下，本論文は次のように構成される．まず，2 節で SAT と動的 SAT をそれぞれ定義し，3 節で値変更コスト付き動的 SAT を提案する．4 節では値変更コスト付き動的 SAT 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. SAT と動的 SAT

SAT とは，与えられた CNF 論理式がモデルを持つか否かを判定する問題である．CNF 論理式は節の連言で表された論理式で，各節はリテラルの選言で表される．リテラルはブール変数もしくはその否定で表される．各ブール変数は真 (1) または偽 (0) の値をとり，モデルとは CNF 論理式を真にするようなブール変数への真偽値の割当てである．

動的 SAT は SAT の拡張で，その目的は現実問題の動的性質をモデル化することにある．ここでは動的 SAT の定義を述べる [Hoos 00a].

【定義 1】（動的 SAT） 動的 SAT の問題例は (X,ϕ) で与えられる．$X=\{x_1,\ldots,x_n\}$ はブー ル変数の集合で，ϕ は関数 $\phi: T \to \mathrm{CNF}(X)$，$T$ は非負整数の集合である．$\mathrm{CNF}(X)$ は 起こりうる全ての CNF 論理式で，X のブール変数のみで構成されている．

すなわち，動的 SAT の問題例は X のブール変数で構成された CNF 論理式の系列で表現され，時刻 $t \in T$ における CNF 論理式は関数 ϕ によって与えられる．また，k-stage 動的 SAT とは時刻 $(k-1)$ 以降は変化しない動的 SAT である．

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

動的 SAT には 2 種類あり，1 つは決定問題としての動的 SAT で，任意の時刻 t において $\phi(t)$ がモデル $M(t)$ を持つかどうかを決定する問題である．もし，すべての $\phi(t)$ にモデルがあれば，ϕ は充足可能で，そうでなければ ϕ は充足不能である．もう 1 つはモデル追跡としての動的 SAT で，ϕ が充足可能と仮定して具体的なモデルの系列を求める問題である．本論文ではモデル追跡に焦点を当てる．

3. 値変更コスト付き動的 SAT

モデル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．もし，実世界で意思決定が変化すれば，それに対して（プランニングやスケジューリングにおける段取りコストのような）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．

本論文では，意思決定変更コストを変数の値を変更するコストの合計として定義する．まず，ある時刻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） 時刻 t において変数 x_i が変化 したときのコストは関数 $f: T\backslash\{0\} \times X \times \{1,0\} \times \{1,0\} \to \mathcal{R}$ によって与えられる．ここで，T は 非負整数の集合で X はブール変数の集合，$\mathcal{R}$ は非負実数の集合である．

例えば，$f(t,x_i,1,0)$ は時刻 t において変数 x_i が 1 から 0 に変化した時にかかるコスト $c_0^{i,t}$ を返す．同様に $f(t,x_i,0,1)$ は変数 x_i が 0 から 1 に変化したときにかかるコスト $c_1^{i,\,t}$ を返す．$f(t,x_i,1,1)$ と $f(t,x_i,0,0)$ はどの時刻 t のどの変数 x_i に対してもコスト 0 を返す．これは値が変化しない場合はコストが 0 であることを意味する．

2 つの連続したモデル $M(t-1)$ と $M(t)$ が与えられたとする．この 2 つのモデル間で変数 x_i の値が変化した

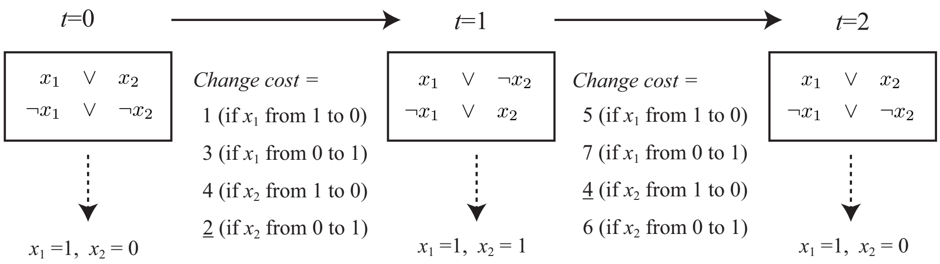


図 1　値変更コスト付き 3-stage DynSAT の問題例

時のコストは $cost(x_i, M(t-1), M(t))$ と記述され，コスト関数 f を参照することにより決定できる．すなわち，$cost(x_i, M(t-1), M(t))$ の値は $f(t, x_i, \_, \_)$ で求められる．そして，各変数の値変更コストをある演算子 $\oplus$ で集めることにより，$M(t-1)$ から $M(t)$ に変化するコストを求めることができる．これを $cost(M(t-1), M(t))$ と記述する．すなわち，

$$cost(M(t-1), M(t)) \equiv \bigoplus_{x_i \in X} cost(x_i, M(t-1), M(t)).$$

例えば，$\oplus$ を $'+'$ にすると次式が得られる．

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

さらに，M を集合 T 上のモデルの系列，すなわち $M = \{M(t) \mid t \in T\}$ とすると，このモデルの系列 M のコストを次式のように定義する．

$$cost(M) \equiv \bigodot_{t \in T \setminus \{0\}} cost(M(t-1), M(t)), \qquad (1)$$

ここで，$\odot$ は各モデル間の変更コストを集める演算子である．例えば，$\odot$ を $'+'$ にすると次式が得られる．

$$cost(M) \equiv \sum_{t \in T \setminus \{0\}} cost(M(t-1), M(t)).$$

値変更コスト付き動的 SAT 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 SAT） 値変更コスト付き動的 SAT の問題例は 5 つ組 $(X, \phi, f, \oplus, \odot)$ で 与えられる．ここで，X と ϕ は定義 1 ，f は定義 3 で定義されたものを 使用し，$\oplus$ と $\odot$ はそれぞれあるモデルの各変数の値変更コストを集 める演算子と各モデル間の変更コストを集める演算子である．

動的 SAT と同様に，値変更コスト付き k-stage 動的 SAT を次のように定義する．

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

値変更コスト付き（k-stage）動的 SAT の目的は，(1) 式で与えられるコストを最小化するようなモデルの系列 M^* を求めることである．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，それを与えるモデルの系列 M^* を最適解と呼ぶ．

以上のように，一般的な $(\oplus, \odot)$ を扱うことで，値変更コスト付き k-stage 動的 SAT をより広い文脈をカバーできるような問題の枠組みとして表現できる．そのメリットは，問題の本質的な部分と自由度のある部分を明確にできることである．

〔例 1〕（値変更コスト付き 3-stage 動的 SAT） 図 1 に $(3, \{x_1, x_2\}, \phi, f, +, +)$ の問題例を示す．ここで，

- $\phi(0) = (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)$,
- $\phi(1) = (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2)$,
- $\phi(2) = (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)$,
- $f(1, x_1, 1, 0) = 1$, $f(1, x_1, 0, 1) = 3$,
- $f(1, x_2, 1, 0) = 4$, $f(1, x_2, 0, 1) = 2$,
- $f(2, x_1, 1, 0) = 5$, $f(2, x_1, 0, 1) = 7$,
- $f(2, x_2, 1, 0) = 4$, $f(2, x_2, 0, 1) = 6$.

とすると，各 CNF 論理式 $\phi(t)$ に対し以下のような 2 つのモデルが得られる．

- $M(0) \in \{(x_1 = 0, x_2 = 1), (x_1 = 1, x_2 = 0)\}$,
- $M(1) \in \{(x_1 = 0, x_2 = 0), (x_1 = 1, x_2 = 1)\}$,
- $M(2) \in \{(x_1 = 0, x_2 = 1), (x_1 = 1, x_2 = 0)\}$.

この問題例の最適解 M^* は $M^*(0) = (x_1 = 1, x_2 = 0), M^*(1) = (x_1 = 1, x_2 = 1), M^*(2) = (x_1 = 1, x_2 = 0)$ となり，そのコストは $f(1, x_2, 0, 1) + f(2, x_2, 1, 0)$ で 6 となる．

4. 解　　法

本論文では，自然な問題設定の 1 つであると考えられる $(k, X, \phi, f, +, +)$ に対して 2 つの解法を提案する．

4·1　重み付き部分 MaxSAT

1 つ目の解法では，与えられた $(k, X, \phi, f, +, +)$ で表される問題を重み付き部分 MaxSAT 問題に変換し，SAT4J [Berre and Parrain 10]，WMAXSATZ [Li 07]，あるいは確率的局所探索である UBCSAT [Tompkins 05] などの既存ソルバーを用いて解く．重み付き部分 MaxSAT 問題には必ず満たさなければならないハード節と必ずしも満たす必要はないが満たさない場合は重みと呼ばれるコストがかかるソフト節がある．

問題の変換は以下の手順で行う．CNF 論理式 $\phi(t)$ の各節に対して，時刻 t でラベル付けされたブール変数を用

いてハード節を導入する．例えば，$\phi(2)$ の節 $x_1 \vee x_2$ は $x_1^2 \vee x_2^2$ というハード節に変換される．一方，ソフト節は関数 f で定義されたコストを用いて導入される．例えば，$f(2, x_1, 0, 1) = 7$ は時刻 2 において変数 x_1 の値が 0 から 1 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．これをソフト節で表すと重み 7 を持つソフト節 $x_1^1 \vee \neg x_1^2$ となる．一般的に $f(t, x_i, 1, 0) = c_0^{i,t}$ は重み $c_0^{i,t}$ を持つソフト節 $\neg x_i^{t-1} \vee x_i^t$ となり，$f(t, x_i, 0, 1) = c_1^{i,t}$ は重み $c_1^{i,t}$ を持つソフト節 $x_i^{t-1} \vee \neg x_i^t$ 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 [Bertsekas 99] 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．

§1 分 解

まず，コスト関数 f を 0-1 整数計画問題に変換する．今，f の値が $f(t, x_i, 1, 0) = c_0^{i,t}$，$f(t, x_i, 0, 1) = c_1^{i,t}$，$f(t, x_i, 1, 1) = 0$，$f(t, x_i, 0, 0) = 0$ だとする．後者の 2 つの値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．これらのコストは以下の 0-1 整数計画問題の最適値に等しい．

$$\begin{aligned} \text{min.} \quad & c_0^{i,t} y_0^{i,t} + c_1^{i,t} y_1^{i,t} \\ \text{s.t.} \quad & x_i^{t-1} - x_i^t - y_0^{i,t} + y_1^{i,t} = 0, \\ & x_i^{t-1}, x_i^t, y_0^{i,t}, y_1^{i,t} \in \{0, 1\}, \end{aligned}$$

ここで x_i^{t-1} と x_i^t はそれぞれ時刻 $t-1$ と t における変数 x_i で，$y_0^{i,t}$ と $y_1^{i,t}$ は 0 または 1 の値を取る補助変数である．例えば，$x_i^{t-1} = 1$，$x_i^t = 0$ の場合，上記の問題の最適値は $c_0^{i,t}$ になる．この変換は任意の x_i と t の組み合わせに対して適用できる．

コストの集計に加法を用いているため $(k, X, \phi, f, +, +)$ の全体の問題は CNF 論理式を含んだ 0-1 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 $x_i^{t-1}, x_i^t, y_0^{i,t}, y_1^{i,t}$ の 0-1 制約を省略する．

$$\begin{aligned} \mathcal{P} \,:\, \text{min.} & \sum_{t=1}^{k-1} \sum_{i=1}^{n} (c_0^{i,t} y_0^{i,t} + c_1^{i,t} y_1^{i,t}) \\ \text{s.t.} \quad & x_i^{t-1} - x_i^t - y_0^{i,t} + y_1^{i,t} = 0, \qquad (2) \\ & i = 1, ..., n,\ t = 1, ..., k-1, \\ & \phi(t), \quad t = 0, ..., k-1, \qquad (3) \end{aligned}$$

問題 $\mathcal{P}$ は CNF 論理式をすべて制約 (3) として持っているため，$\mathcal{P}$ の実行可能解はモデル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^t に写像したものが最適解となる．

しかし，$\mathcal{P}$ を直接解くことは一般に困難なので，$\mathcal{P}$ を緩和し，扱いやすい問題に変換する．ここでは，異なる時刻の変数間にまたがって定義された (2) の制約を緩和したラグランジュ緩和問題を導入する．

$$\begin{aligned} \mathcal{L} \,:\, L(\mu) \,=\, & \text{min.} \sum_{t=1}^{k-1} \sum_{i=1}^{n} (c_0^{i,t} y_0^{i,t} + c_1^{i,t} y_1^{i,t}) \\ & + \sum_{t=1}^{k-1} \sum_{i=1}^{n} \mu^{i,t} (x_i^{t-1} - x_i^t - y_0^{i,t} + y_1^{i,t}) \\ & \text{s.t.} \quad \phi(t), \quad t = 0, ..., k-1, \end{aligned}$$

ここで，μ はラグランジュ乗数ベクトルと呼ばれ，その要素 $\mu^{i,t}$ は実数値を取る．この問題の目的関数を整理すると以下のようになる．

$$\begin{aligned} \mathcal{L} \,:\, L(\mu) \,=\, \text{min.} & \sum_{t=1}^{k-1} \sum_{i=1}^{n} (c_0^{i,t} - \mu^{i,t}) y_0^{i,t} \\ & + \sum_{t=1}^{k-1} \sum_{i=1}^{n} (c_1^{i,t} + \mu^{i,t}) y_1^{i,t} \\ & + \sum_{i=1}^{n} \mu^{i,1} x_i^0 \\ & + \sum_{t=1}^{k-2} \sum_{i=1}^{n} (\mu^{i,t+1} - \mu^{i,t}) x_i^t \\ & + \sum_{i=1}^{n} (-\mu^{i,k-1}) x_i^{k-1}, \\ \text{s.t.} \quad & \phi(t), \quad t = 0, ..., k-1. \end{aligned}$$

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

$$\begin{aligned} L^{aux}(\mu) \,=\, \text{min.} & \sum_{t=1}^{k-1} \sum_{i=1}^{n} (c_0^{i,t} - \mu^{i,t}) y_0^{i,t} \\ & + \sum_{t=1}^{k-1} \sum_{i=1}^{n} (c_1^{i,t} + \mu^{i,t}) y_1^{i,t}, \qquad (4) \end{aligned}$$

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

各時刻 $t \in \{1, ..., k-2\}$ について

$$\begin{aligned} L^t(\mu) \,=\, & \text{min.} \sum_{i=1}^{n} (\mu^{i,t+1} - \mu^{i,t}) x_i^t, \qquad (6) \\ & \text{s.t.} \quad \phi(t), \end{aligned}$$

$$\begin{aligned} L^{k-1}(\mu) \,=\, & \text{min.} \sum_{i=1}^{n} (-\mu^{i,k-1}) x_i^{k-1}, \qquad (7) \\ & \text{s.t.} \quad \phi(k-1). \end{aligned}$$

ここで μ が決まれば，各部分問題は実際には重み付き部分 MaxSAT 問題であることに注意すべきである．すなわち，(4) は補助変数の単位ソフト節のみからなる重み付き部分

MaxSAT 問題，(5) は最初の時刻 0 におけるハード節 $\phi(0)$ と単位ソフト節からなる重み付き部分 MaxSAT 問題，(6) は最初と最後の時刻を除いた時刻 $t \in \{1, ..., k-2\}$ におけるハード節 $\phi(t)$ と単位ソフト節からなる重み付き部分 MaxSAT 問題，(7) は最後の時刻 $k-1$ におけるハード節 $\phi(k-1)$ と単位ソフト節からなる重み付き部分 MaxSAT 問題である．

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

$$\mathcal{D} : \max. \quad L(\mu),$$

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，$\mu^{i,t}$ が決定変数であり，しかも，$\mu^{i,t}$ 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，

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

となるため，上記のラグランジュ双対問題は，

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

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という原問題 $\mathcal{P}$ に対する解法が可能になる．

§2 操作の概要

ここでは，$\mathcal{P}$ の実行可能解に加え，最適値の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: 既存の重み付き部分 MaxSAT ソルバーを用いて (4) から (7) の部分問題を解く;

step 3: 下界を与える実行不可能解から $\mathcal{P}$ の実行可能解を求め，これまでに得られた最も高い下界 *BestLB*，最も低い上界 *BestUB* を求める．さらに *BestUB* に対応する実行可能解を M とする;

step 4: もし終了条件を満たすならば，*BestLB*，*BestUB*，M 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．

§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，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各部分問題の最適解における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下界 *LB*，上界 *UB* 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 *BestLB*，*BestUB* とする．

§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 つ目の方法では，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次の定理を用いる．

［定理 1］ 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，それらは $\mathcal{P}$ 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．

一方，step 3 で得られた $\mathcal{P}$ の実行可能解 M の目的関数値が下界を与えると判断できる場合，明らかに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．

§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，$\mathcal{P}$ の最適解により近い下界を求めるために step 4 で μ の値を更新する．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．

劣勾配法ではラグランジュ乗数 $\mu^{i,t}$ を以下のように更新する．

step 4.1: 劣勾配 $G^{i,t}$ を次式で計算する

$$G^{i,t} = x_i^{t-1} - x_i^t - y_0^{i,t} + y_1^{i,t},$$

これは $\mathcal{L}$ の目的関数に含まれる $\mu^{i,t}$ の係数であり，step 2 で得た部分問題の最適解を用いて計算する．

step 4.2: $\mu^{i,t}$ を次のように更新する．

$$\mu^{i,t} \leftarrow \mu^{i,t} + D \cdot G^{i,t},$$

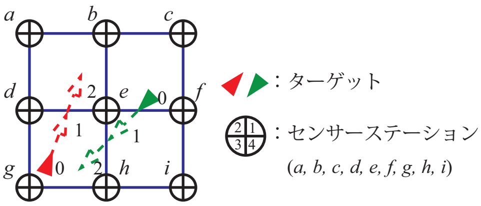
D はステップ長と呼ばれ，次のように計算される

$$D = \frac{\pi(BestUB - BestLB)}{\sum_{i,t}(G^{i,t})^2},$$

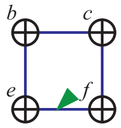
π は初期値 2 より単調減少する正のスカラー値である．

この規則は $\mathcal{L}$ の目的関数内の $G^{i,t}$ が正（負）であれば $\mu^{i,t}$ を増加（減少）させることで，次回の $\mathcal{P}$ の下界 $L(\mu)$ が上昇することを期待している．

しかし，劣勾配法では単調性は保証されず，下界は各繰り返し操作毎に上下する．さらに，この方法は $\mathcal{P}$ 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場合があり，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

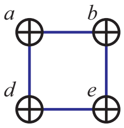


(a) ターゲット追跡問題の全体像



$b_4 \vee c_3 \vee e_1 \vee f_2$　←(b_4, c_3, e_1, f_2) が OFF のとき違反
$\neg b_4 \vee c_3 \vee e_1 \vee f_2$　←b_4 が ON, (c_3, e_1, f_2) が OFF のとき違反
$b_4 \vee \neg c_3 \vee e_1 \vee f_2$　←c_3 が ON, (b_4, e_1, f_2) が OFF のとき違反
$b_4 \vee c_3 \vee \neg e_1 \vee f_2$　←e_1 が ON, (b_4, c_3, f_2) が OFF のとき違反
$b_4 \vee c_3 \vee e_1 \vee \neg f_2$　←f_2 が ON, (b_4, c_3, e_1) が OFF のとき違反

(b) 区域内にターゲットがいる場合に生成される節集合（時刻 0，右上の区域の場合）



$\neg a_4$　←a_4 が ON のとき違反
$\neg b_3$　←b_3が ON のとき違反
$\neg d_1$　←d_1が ON のとき違反
$\neg e_2$　←e_2が ON のとき違反

(c) 区域内にターゲットがいない場合に生成される節集合（時刻 0，左上の区域の場合）

$\neg a_1 \vee \neg a_2$　←a_1と a_2が ON のとき違反
$\neg a_1 \vee \neg a_3$　←a_1と a_3が ON のとき違反
$\neg a_1 \vee \neg a_4$　←a_1と a_4が ON のとき違反
$\neg a_2 \vee \neg a_3$　←a_2と a_3が ON のとき違反
$\neg a_2 \vee \neg a_4$　←a_2と a_4が ON のとき違反
$\neg a_3 \vee \neg a_4$　←a_3と a_4が ON のとき違反

(d) センサーステーション内の制約から生成される節集合（センサーステーション a の場合）

図 2　ターゲット追跡問題

の最良の上界と下界，および実行可能解のみ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ランダム問題とターゲット追跡問題を用いて評価する．

- 重み付き部分 MaxSAT に基づくソルバー群

Sat4j：SAT 符号化と SAT ソルバーを使用した厳密解法を実装したソルバーである．経験的には，構造を持った問題に強いと言われている [Berre and Parrain 10]．

WMaxSatz：分枝限定法に基づく厳密解法を実装したソルバーである．経験的には，ランダムな問題に強いと言われている [Li 07]．

WalkSat：確率的局所探索に基づく代表的なソルバーである [Selman 96]．

Irots：確率的局所探索に基づくソルバーでタブーサーチを実装している [Smyth 03]．

- ラグランジュ分解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 Sat4j で解く．実行可能解は，各繰り返し操作毎に求められる部分問題の最適解から，各緩和制約を満たすように補助変数の値を変更することにより得られる．

LdWalk：ラグランジュ分解に基づくソルバーで，同じく各部分問題を Sat4j で解く．さらに得られた実行可能解を一定時間（100,000 フリップ）WalkSat を実行することにより改善する．この改善は下界が更新された場合のみ実行する．

LdIrots：ラグランジュ分解に基づくソルバーで LdWalk と同様である．違いは WalkSat の代わりに Irots を用いて実行可能解を改善する点である．

実験の目的は，限られた時間内でどの程度質の高い実行可能解が得られるかを知ることである．制限時間が厳しい動的な環境では，この実験設定は理にかなっていると考えられる．実行時間を制限した場合，重み付き部分 MaxSAT では上界しか得られないが，ラグランジュ分解に基づくソルバーでは上界と下界の両方を求めることが可能である．

以下では，実験に用いる 2 つの問題について説明する．1 つ目はランダム問題である．$(k, X, \phi, f, +, +)$ 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 100 個のブール変数の集合 $\{x_1, ..., x_{100}\}$ である．
- 各 t における ϕ は，SATLIB [Hoos 00b] の `uf100-430` からランダムに選択された CNF 論理式を返す．`uf100-430` は 100 変数，430 節を持つ CNF 論理式の集合である．
- f の値は，ある時刻における変数の値変更コストを表し，$\{1, 2, ..., 10^6\}$ からランダムに選択される．

つまり，系列サイズが k の問題例は変数の数が $100 \times k$，ハード節の数が $430 \times k$，ソフト節の数が $100 \times (k-1)$ の問題例となる．なお，実験では k の各値について 30 個の問題例をランダムに生成した．

もう 1 つはターゲット追跡問題である．この問題では図 2(a) のように領域内にいるターゲットをセンサーを用いて追跡する．領域には複数の区域が存在し，各区域にはセンサーステーションが配置されている．なお，各センサーステーションは 4 つのセンサーを持っている．ターゲットの移動経路は予め予想できるものとし，図 2(a) は時刻 0，1，2 におけるターゲットの位置情報を示している．このとき，各センサーは以下の 3 種類の制約を満たしつつターゲットを追跡する必要がある．1 つ目の制約は，区域内にターゲットがいる場合，区域内の少なくとも 2 つのセンサーを ON にするという制約である（図

2(b)）．2 つ目は，区域内にターゲットがいない場合，区域内のすべてのセンサーを OFF にするという制約である（図 2(c)）．3 つ目は，同じセンサーステーション内のセンサーは同時に 1 つしか ON にできないという制約である（図 2(d)）．3 種類の制約を満たすように，各センサーは時刻毎の状態を ON か OFF に決定する．このとき，予想されたターゲットの移動経路に対して，各センサーの状態の切り替え回数の合計を最小にすることがターゲット追跡問題の目的である．

次に，ターゲット追跡問題を値変更コスト付き動的SATとして定式化する．まず，各時刻におけるターゲットの位置情報から上述の 3 種類の制約と SAT の節集合として記述する．なお，ここでは各センサーに 1 つのブール変数を定義し，その値はセンサーが ON のときは 1，OFF のときは 0 とする．例えば，時刻 0 の右上の区域の制約は図 2(b) の節集合で記述できる．この節集合は，全てのセンサーが OFF になる場合と 3 つのセンサーが OFF になる場合を排除している．また，時刻 0 の左上の区域の制約は，図 2(c) の単位節の集合で記述できる．一方，3 つ目の制約は，ターゲットの位置情報に依存しない各センサーステーションに関する静的な制約である．例えば，センサーステーション a に対しては，図 2(d) のような節集合が得られる．この節集合は，センサーステーション a 内のどの 2 組のセンサーも同時に ON にならないことを要求している．以上のようにして得られた各時刻ごとの SAT 問題に対し，各センサーを ON から OFF，または，OFF から ON に切り替えるコストを値変更コストとすることでターゲット追跡問題を値変更コスト付き動的 SAT として定式化できる．

実験に使用する問題例では，センサーステーションの個数を 25 個とする．つまり，4×4 の区域が用意され，その上をターゲットが移動する．ターゲットの個数は 10 個で，各ターゲットは時刻が進むごとにある方向に，ある距離だけ直進する．ここで使用する方向と距離はターゲット毎に決められている．このターゲット追跡問題を CNF 論理式として記述すると，100 変数，約 260 節の CNF 論理式となる．具体的に，$(k, X, \phi, f, +, +)$ 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 100 個のブール変数の集合 $\{x_1, ..., x_{100}\}$ である．
- 各 t における ϕ は，ターゲットの位置情報から得られる充足可能な CNF 論理式を返す．
- f の値は，ある時刻における変数の値変更コストを表し，10^5 の固定値を取る．

つまり，系列サイズが k の問題例は変数の数が $100 \times k$，ハード節の数が $260 \times k$，ソフト節の数が $100 \times (k-1)$ の問題例となる．ランダム問題と同様に，k の各値について 30 個の問題例をランダムに生成した．

実験では，ソルバー性能を比較するために，比較項目として実行可能解の質を使用する．実行可能解の質は各ソルバーが求めた *BestUB* を，LD 系のソルバーが求めた *BestLB* のうち，最大の値である *BestLB** で割った値である．言うまでもなくこの値は 1 に近いほど良い．

ラグランジュ分解に基づくソルバーの基本部分はJAVAで実装し，RedHat Enterprise Linux 5(64bit) の JDK1.6.0_11 でコンパイルした．SAT4J と WMAXSATZ は Web 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．WALKSAT と IROTS は UBCSAT バージョン 1.1 [Tompkins 05] をダウンロードし，'-w' オプションをつけて利用した．実験のマシン環境は Intel Xeon X5460@3.16GHz，4 コア，メモリ 32GB である．なお，ラグランジュ分解に基づくソルバーでは，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ため，4 コアを用いて部分問題を並列に解いた．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．

ラグランジュ分解に基づくソルバーは実装するうえで気を付ける点がある．ラグランジュ分解を用いた解法では，ラグランジュ乗数の値によっては重みが実数である部分問題が発生する可能性がある．しかし，既存の重み付き部分 MaxSAT ソルバーの多くは実数値を扱えない．この問題を解決するために，ラグランジュ乗数の更新時にステップ長 D の小数点以下を切り捨てることで，D の値を整数値化する．この方法により，ラグランジュ乗数が整数値となり，分解後の重み付き部分 MaxSAT 問題の節の重みも整数値となる．

表 1 はランダム問題に対して制限時間 5 分で実行可能が得られた問題例の割合を表している．表を見てわかるように，SAT4J，LD，LDWALK，LDIROTS は全ての問題例に対して 5 分間で少なくとも 1 つの実行可能解が得られた．一方，WMAXSATZ は実行可能解が得られないときがあり，サイズ 35 においては 1 つも実行可能解を得られていない．WMAXSATZ は，2009MaxSAT Evaluation の結果からもわかるようにランダムな問題には強いが，構造をもつ問題には弱い傾向がある．今回の実験で扱ったランダム問題では各 SAT 問題例はランダムであるが，それらを k 個連ねた問題例は互いにソフト節で接続されているため，全体としては特殊な構造をもつ問題になる．そのため，WMAXSATZ の結果が悪くなったと考えられる．一方，確率的局所探索法に関して，WALKSAT は特にサイズが大きい問題に対してはあまり良い結果が得られなかった．IROTS に関してはすべての問題例で実行可能解の発見に失敗した．

なお，LD 系のソルバーの 1 つの特徴として最適解を保証できる解を発見できることが挙げられるが，今回の実験ではそのような解を発見した実行例はなかった．

図 3 は 5 分間の実験で全ての問題例で実行可能解が得られたソルバーの解の質の平均値と標準偏差である．図

表 1 ランダム問題に対して制限時間 5 分で実行可能が得られた問題例の割合

Size k	10	15	20	25	30	35
Sat4j	1.00	1.00	1.00	1.00	1.00	1.00
WMaxSatz	0.63	0.27	0.13	0.07	0.03	0.00
WalkSat	1.00	1.00	0.87	0.70	0.50	0.47
IRots	0.00	0.00	0.00	0.00	0.00	0.00
Ld	1.00	1.00	1.00	1.00	1.00	1.00
LdWalk	1.00	1.00	1.00	1.00	1.00	1.00
LdIRots	1.00	1.00	1.00	1.00	1.00	1.00

表 2 ターゲット追跡問題に対して制限時間 5 分で実行可能が得られた問題例の割合

Size k	10	15	20	25	30	35
Sat4j	1.00	1.00	1.00	1.00	1.00	1.00
WMaxSatz	1.00	1.00	1.00	0.97	1.00	0.93
WalkSat	1.00	1.00	1.00	1.00	1.00	1.00
IRots	1.00	1.00	1.00	1.00	1.00	1.00
Ld	1.00	1.00	1.00	1.00	1.00	1.00
LdWalk	1.00	1.00	1.00	1.00	1.00	1.00
LdIRots	1.00	1.00	1.00	1.00	1.00	1.00

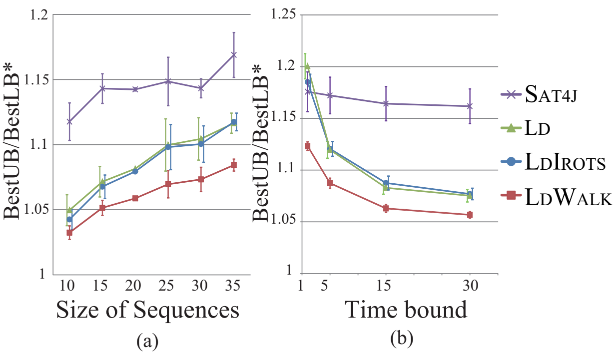


図 3 ランダム問題に対する，(a) 実行可能解の質 対 SAT 問題例の系列サイズ (制限時間 5 分) と (b) 実行可能解の質 対 制限時間 (系列サイズ 35)

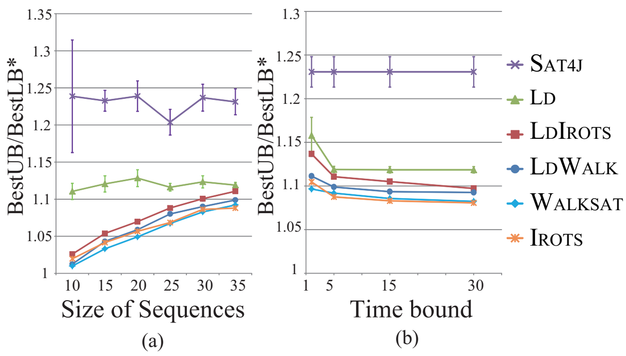


図 4 ターゲット追跡問題に対する，(a) 実行可能解の質 対 SAT 問題例の系列サイズ (制限時間 5 分) と (b) 実行可能解の質 対 制限時間 (系列サイズ 35)

3(a) の x 軸は系列サイズ k である．明らかに，どの系列サイズ k に対してもラグランジュ分解に基づくソルバー群の方が Sat4J よりも解の質が良いことが分かる．図 3(b) は系列サイズを 35 に固定した 30 個の問題例に対して，1 分，5 分，15 分，30 分間で得られた解の質の平均値と標準偏差である．この図は制限時間を延ばしてもラグランジュ分解に基づくソルバー群の方が Sat4J よりも良いことを示している．WMaxSatz，WalkSat，IRots に関しては制限時間を 30 分に延ばしても実行可能解を得られない問題例が存在した．

今回の実験では，uf100-430 からランダムに選択した k 個の CNF 論理式を用いて問題例を作成したが，1 つ前の時刻の CNF 論理式から例えば 2 割の節を入れ替えるという要領で作成した問題例についても同様の実験を行なった．その結果，ソルバー間の性能比較については前述のものと大差はなかった．文献 [Wallace 09] では，制約充足問題では制約が少し変化するだけでも解が大きく変わることが指摘されているが，今回扱った SAT 問題でも事情は同じだと推測できる．すなわち，CNF 論理式を少しだけ変えたとしても解は大きく変わるので，CNF 論理式をランダムに変えた場合の結果と大差ないと推測できる．

表 2 はターゲット追跡問題に対して制限時間 5 分で実行可能が得られた問題例の割合を表している．表からわかるように，WMaxSatz を除いた全てのソルバーが全ての問題例に対して 5 分以内に実行可能解を発見できた．

なお，全ての問題例において，各ソルバーが求めた実行可能解の内，最適解であると保障された実行可能解はなかった．

図 4 はターゲット追跡問題に対する解の質の平均と標準偏差である．図 4(a) より，全ての系列サイズ k に対して，LdIRots と LdWalk は IRots や WalkSat とほぼ同等の解の質が得られている．図 4(b) は，系列サイズが 35 である 30 個の問題例に対して 1 分，5 分，15 分，30 分間で得られた解の質の平均と標準偏差を示した図である．制限時間を延ばしても LdIRots と LdWalk は IRots や WalkSat とほぼ同等であることがわかる．ターゲット追跡問題は一般的にハード節が少ない問題であるため，IRots や WalkSat のような確率的なソルバーでも比較的良い解が得られる．しかし，ランダム問題の結果と合わせると，IRots や WalkSat には頑健性があるとは言えない．

6. お わ り に

本論文では，値変更コスト付き動的 SAT を提案し，それに対する 2 つの解法として重み付き部分 MaxSAT 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．さらに，ラ

グランジュ分解に基づくソルバーである LD，LDWALK，LDIROTS 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．

これらの解法と結果は値変更コストを集める演算子 $(\oplus,\odot)$ に $(+,+)$ を用いた時のものであり，他の有用な組み合わせも考えることができる．例えば，モデル $M(t-1)$ と $M(t)$ の間での値変更コストの最大値を両モデル間のコストとすることは妥当な集計方法の 1 つであると考えられるが，これは，$\oplus$ を'max' とすることで記述できる．また，近い未来よりも遠い未来のコストを割り引いて集計するような場合，$\odot$ を'$+\delta^s$'（時刻 s だけ先の未来のコストは δ^s 倍 $(0<\delta<1)$ して加算するという演算子) とすることで記述できる．

以上のようにコストを集計する演算子を変えると，その解法も個別に考案する必要がある．$(+,+\delta^s)$ は今回提案した重み付き部分 MaxSAT とラグランジュ分解の 2 つの解法を適用できる．また，$(max,+)$ や $(max,+\delta^s)$ に対しては，SAT ソルバーを繰り返し適用するという解法が考えられる．

今後の課題は，双対問題における μ の更新方法に劣勾配法以外の方法を検討し，下界値をさらに改善することである．また，本研究と同様のアイデアを動的 CSP に適用することである．

◇ 参 考 文 献 ◇

[Bailleux 06] Bailleux, O., Marquis, P.: *Some computational aspects of distance-SAT,* Journal of Automated Reasoning 37(4), pp. 231–260 (2006)

[Berre and Parrain 10] Berre, D.L. and Parrain, A.: *The Sat4j library, release 2.2, system description,* Journal on Satisfiability, Boolean Modeling and Computation 7, pp. 59–64 (2010)

[Bertsekas 99] Bertsekas, D.P.: *Nonlinear Programming,* Athena Scientific, second edn. (1999)

[Bessiere 91] Bessière, C.: *Arc-consistency in dynamic constraint satisfaction problems,* In: Proceedings of the 9th National Conference on Artificial Intelligence (AAAI-1991). pp. 221–226 (1991)

[Bofill *et al.*10] Bofill, M., Busquets, D. and Villaret, M. *A Declarative Approach to Robust Weighted Max-SAT,* 12th Intl. ACM SIGPLAN Symposium on Principles and Practice of Declarative Programming, pp. 67–76 (2010)

[Dechter 88] Dechter, R., Dechter, A.: *Belief maintenance in dynamic constraint networks,* In: Proceedings of the 7th National Conference on Artificial Intelligence (AAAI-1988). pp. 37–42 (1988)

[Een 03] Eén, N., Sörensson, N.: *An extensible SAT-solver,* In: Proceedings of 6th International Conference on Theory and Applications of Satisfiability Testing (SAT-2003). pp. 502–518 (2003)

[Fragier 96] Fargier, H., Lang, J., Schiex, T.: *Mixed constraint satisfaction: a framework for decision problems under incomplete knowledge,* In: Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-1996). pp. 175–180 (1996)

[Freuder 91] Freuder, E.C.: *Eliminating interchangeable values in constraint satisfaction problems,* In: Proceedings of the 9th National Conference on Artificial Intelligence (AAAI-1991). pp. 227–233 (1991)

[Ginsberg 98] Ginsberg, M.L., Parkes, A.J., Roy, A.: *Supermodels and robustness,* In: Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-1998). pp. 334–339 (1998)

[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-, 人工知能学会誌, Vol. 12, No 3, pp. 25–35 (2005)

[Hebrand 05] Hebrard, E., Hnich, B., O'Sullivan, B., Walsh, T.: *Finding diverse and similar solutions in constraint programming,* In: Proceeding of the 20th National Conference on Artificial Intelligence (AAAI-2005). pp. 372–377 (2005)

[平山 10] 平山勝敏，横尾真: *-SAT: SAT の拡張, 人工知能学会誌, Vol.25, No.1, pp.105–113 (2010)

[Hoos 00a] Hoos, H.H., O'Neill, K.: *Stochastic local search methods for dynamic SAT – an initial investigation,* In: AAAI-2000 Workshop: Leveraging Probability and Uncertainty in Computation. pp. 22–26 (2000)

[Hoos 00b] Hoos, H.H., Stützle, T.: *SATLIB: An online resource for research on SAT,* In: I.P.Gent, H.v.Maaren, T. (ed.) SAT 2000. pp. 283–292. IOS Press (2000), available online at www.satlib.org.

[Smyth 03] Smyth, K., Hoos, H.H., Stutzle, T.: *Iterated robust tabu search for MAX-SAT,* In: Proceedings of the 16th Conference of the Canadian Society for Computational Studies of Intelligence (AI-03). pp. 129–144 (2003)

[Komodakis 07] Komodakis, N., Paragios, N., Tziritas, G.: *MRF optimization via dual decomposition: Message-passing revisited,* In: Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV-2007). pp. 1–8 (2007)

[黒田 09] 黒田 陽之 平山 勝敏: Multi-MaxSAT：ラグランジュ分解・調整法を用いた Weighted Max-SAT 問題の解法: 電子情報通信学会論文誌 D Vol.J92-D No.1 pp. 51–60 (2009).

[Li 07] Li, C.M., Manyà, F., Planes, J.: *New inference rules for max-sat,* Journal of Artificial Intelligence Research 30, pp. 321–359 (2007)

[Ran 02] Ran, Y., Roos, N., van den Herik, J.: *Approaches to find a near-minimal change solution for dynamic CSPs,* In: Proceedings of the 4th International Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CP-AI-OR-2002) (2002)

[Roos 00] Roos, N., Ran, Y., van den Herik, J.: *Combining local search and constraint propagation to find a minimal change solution for a dynamic CSP,* In: Proceedings of the 9th International Conference on Artificial Intelligence: Methodology, Systems, and Applications (AIMSA-2000). pp. 272–282 (2000)

[Schiex 93] Schiex, T., Verfaillie, G.: *Nogood recording for static and dynamic constraint satisfaction problems,* In: Proceedings of the 5th IEEE International Conference on Tools with Artificial Intelligence (ICTAI-1993). pp. 48–55 (1993)

[Selman 96] Selman, B., Kautz, H., Cohen, B.: *Local search strategies for satisfiability testing,* In: Proceedings of the 2nd DIMACS Implementation Challenge on Cliques, Coloring, and Satisfiability 26, pp. 521–532 (1996)

[Tompkins 05] Tompkins, D.A.D., Hoos, H.H.: *UBCSAT: An implementation and experimentation environment for SLS algorithms for SAT and MAX-SAT,* In: Theory and Applications of Satisfiability Testing: Revised Selected Papers of the 7th International Conference (SAT 2004). Lecture Notes in Computer Science, vol. 3542, pp. 306–320. Springer Verlag (2005)

[Verfaillie 05] Verfaillie, G., Jussien, N.: *Constraint solving in uncertain and dynamic environments,* A survey. Constraints 10, pp. 253–281 (2005)

[Verfaillie 94] Verfaillie, G., Schiex, T.: *Solution reuse in dynamic constraint satisfaction problems,* In: Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-1994). pp. 307–312 (1994)

[Vinyals 10] Vinyals, M., Pujol, M., Rodriguez-Aguilar, J., Cerquides, J.: *Divide-and-coordinate: DCOPs by agreement,* In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2010). pp. 149–156 (2010)

[Wallace 98] Wallace, R.J., Freuder, E.C.: *Stable solutions for dynamic constraint satisfaction problems,* In: Proceedings of the 4th International Conference on Principles and Practice of Constraint Programming (CP-1998). pp. 447–461 (1998)

[Wallace 09] Wallace, R.J., Grimes, D. and Freuder, E.C.: *Solving dynamic constraint satisfaction problems by identifying stable features,* In: Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09). pp. 621–627 (2009)
[Walsh 02] Walsh, T.: *Stochastic constraint programming,* In: Proceedings of the 15th European Conference on Artificial Intelligence (ECAI-2002). pp. 111–115 (2002)

〔担当委員：岸本 章宏〕

2011 年 7 月 1 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．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，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．電子情報通信学会，情報処理学会，日本オペレーションズリサーチ学会，AAAI 各会員．