



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

黒田, 陽之
平山, 勝敏

(Citation)

電子情報通信学会論文誌. D, 情報・システム, 92(1):51-60

(Issue Date)

2009-01-01

(Resource Type)

journal article

(Version)

Version of Record

(Rights)

copyright©2009 IEICE 本文データは学会の許諾に基づきCiNiiから複製したものである。

(URL)

<https://hdl.handle.net/20.500.14094/90002954>



論文

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

黒田 陽之^{†*} 平山 勝敏^{††a)}

Multi-MaxSAT: Weighted Max-SAT Solving with the Lagrangian Decomposition and Coordination Method

Akiyuki KURODA^{†*} and Katsutoshi HIRAYAMA^{††a)}

あらまし Weighted Max-SAT 問題とは、SAT 問題における各節に重みを与え、満足できない節の重み和を最小にするような論理変数への真偽値の割当を求める最適化問題である。SAT 問題に関する研究が近年劇的に進展したことを受けて、SAT 問題の最適化版である Weighted Max-SAT 問題への関心も最近高まりつつある。本論文では、複数の部分問題（節集合）が互いに緩く結合することにより構成される Weighted Max-SAT 問題に対して特に有効な新しい解法として Multi-MaxSAT を提案する。評価実験によれば、部分問題間で共有される変数の数が比較的少ないとき、Multi-MaxSAT は、最新の代表的な厳密解法や発見的解法よりも処理時間や発見できる解の質の面で優れている。

キーワード Weighted Max-SAT, 疎結合な問題, ラグランジュ分解

1. ま え が き

充足可能性判定問題（Satisfiability Problem：SAT 問題）とは、与えられた CNF 形式の命題論理式を真にするような論理変数への真偽値の割当が存在するか否かを問う問題であり、計算機科学の分野でも古くから研究され、NP 完全であることが示された最初の問題としても有名である。

SAT 問題を厳密に解く効率的なアルゴリズムの存在はほぼ絶望視されていたが、1990 年代にバックトラック探索法に学習メカニズムを組み込んだ手法 [4] や局所探索法に基づく手法 [24] が提案されたことをきっかけに、その後、活発に研究が行われ、現在ではかなりの規模の問題でも現実的な時間で解けることが示されている。更に、プランニングやスケジュー

リング [5], [13], [20], [26], 回路やソフトウェアの設計検証 [23] などの実用的な問題を SAT 問題に変換し、SAT アルゴリズムで解くというアプローチが成功を収めており、SAT 問題に関する研究は近年劇的な進展を見せている。

一方、Max-SAT 問題は SAT 問題の最適化版とみなすことができる。SAT 問題の目的が、与えられた節集合のすべての節を満足することであるのに対し、Max-SAT 問題の目的は、満足できない節の数を最小にすることである。また、各節に重みを与え、満足できない節の重み和を最小にするように Max-SAT 問題を一般化したものを Weighted Max-SAT 問題と呼ぶ。

Weighted Max-SAT 問題の厳密解法に関する研究には、長らく目立った進展がなかったが、国際会議 SAT-2006 において Max-SAT アルゴリズムに関する第 1 回競技会（First Max-SAT evaluation）、続く SAT-2007 において第 2 回競技会（Second Max-SAT evaluation）が開催され、分枝限定法に基づく洗練された厳密解法が最近提案されつつある [2], [7]～[9], [15], [16]。

本論文では、複数の部分問題（節集合）が互いに緩く結合することにより構成される Weighted Max-SAT

[†] 神戸大学大学院自然科学研究科, 神戸市

Graduate School of Science and Technology, Kobe University, Kobe-shi, 658-0022 Japan

^{††} 神戸大学大学院海事科学研究科, 神戸市

Graduate School of Maritime Sciences, Kobe University, 5-1-1 Fukaeminami-machi, Higashinada-ku, Kobe-shi, 658-0022 Japan

* 現在, 富士通関西中部ネットテック株式会社

a) E-mail: hirayama@maritime.kobe-u.ac.jp

問題（疎結合な Weighted Max-SAT 問題）に対して特に有効な新しい解法として Multi-MaxSAT を提案する．評価実験によれば，部分問題間で共有される変数の数が比較的少ないとき，Multi-MaxSAT は，最新の代表的な厳密解法や発見的解法よりも処理時間や発見できる解の質の面で優れている．

Multi-MaxSAT の意義は次の 2 点である．

- 従来の Weighted Max-SAT アルゴリズムの多くが分枝限定法や局所探索法に基づいて設計されているのに対し，Multi-MaxSAT は，ラグランジュ分解・調整法というアイデアに基づく解法である．筆者の知る限り，Weighted Max-SAT 問題に対してこのアイデアを適用した既存研究は存在しない．

- また，疎結合な SAT 問題や Weighted Max-SAT 問題は実用上有用な問題でしばしば現れる．例えば，複数の異なる知識ベースを用いた分散推論エンジンを設計する際，[1] では，各知識ベースがもつ知識を節集合で表現した上で，系全体の問題を，一部の変数を共有する節集合の集合としてモデル化している．また，時間とともに変動する問題を扱う Dynamic SAT 問題 [12] や Dynamic Max-SAT 問題 [22] では，時間軸上で連続する二つの SAT 問題/Max-SAT 問題は，その時点で変更不可能な一部の変数を共有する節集合の集合とみなすことができる．このように空間的あるいは時間的に分割され，互いに緩く結合した節集合からなる問題の充足可能性を判定したり，あるいは，最適な真偽値の割当を求めたいという要求は少なからず存在する．このような特別な構造をもつ問題に Multi-MaxSAT は非常に有効である．

以下，本論文は次のように構成される．2. で Weighted Max-SAT 問題とその節分割記述を導入し，3. で提案手法である Multi-MaxSAT の詳細を述べる．4. では今回行った評価実験の結果を報告し，5. で本論文のまとめを示す．

2. Weighted Max-SAT 問題

2.1 SAT 問題

SAT 問題とは，例えば次のような CNF 形式の命題論理式

$$(x_1 \vee x_3) \wedge (\bar{x}_2 \vee \bar{x}_3) \wedge (x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1)$$

が与えられたとき，これを真にする論理変数 x_i ($i = 1, 2, 3$) への真偽値の割当が存在するか否かを判定する問題である．ここで，各 x_i は真 (1) または偽 (0)

を値にとる変数で， $\bar{x}_i$ は x_i の否定を表す．なお， x_i 及び $\bar{x}_i$ を総称してリテラルと呼ぶ．また，リテラルの選言を節と呼ぶ．上の命題論理式は四つの節からなり，この命題論理式を真にするとは，この四つの節すべてを真にする（充足する）ことにほかならない．

2.2 Weighted Max-SAT 問題

一方，Max-SAT 問題とは，節集合が与えられたとき，充足されない節の数を最小にするような変数への真偽値の割当を求める問題である．また，各節に重みを与え，充足されない節の重み和を最小にするような変数への真偽値の割当を求める問題を Weighted Max-SAT 問題と呼ぶ．先の SAT 問題の例で，例えば次のような重みを各節に与えると，Weighted Max-SAT 問題の例となる．

[例 1] Weighted Max-SAT 問題

- 1: $x_1 \vee x_3$ (weight = 2)
- 2: $\bar{x}_2 \vee \bar{x}_3$ (weight = 3)
- 3: $x_1 \vee x_2 \vee \bar{x}_3$ (weight = 1)
- 4: $\bar{x}_1$ (weight = 2)

ここで，各節の先頭の数字は節に付けられた番号である．例 1 では，変数への真偽値の割当が $(x_1 \ x_2 \ x_3) = (0 \ 0 \ 1)$ のとき，充足されない節の重み和が最小値 1 となる．したがって，この割当がこの問題例の最適解となる．

容易に分かるとおり，Weighted Max-SAT 問題を解くアルゴリズムは，各節の重みを 1 とすることにより Max-SAT 問題を解くことができる．また，各節の重みを任意の正の値とし，充足されない節の重み和の最小値が 0 かそれより大かを判定することにより SAT 問題も解くことができる．

以下，各節のリテラルの数がたかだか k 個である Weighted Max-SAT 問題を特に Weighted Max- k SAT 問題と呼ぶ．例 1 は Weighted Max-3SAT 問題の例である．

2.3 節分割記述

本論文では，Weighted Max-SAT 問題における節集合が複数の部分節集合に分割された状況を記述するために，Weighted Max-SAT 問題の節分割記述を与える．例えば，例 1 において，節 1 と節 2 はサイト 1 に，節 3 と節 4 はサイト 2 に存在するとすれば，次のような Weighted Max-SAT 問題の節分割が得られる．

[例 2] Weighted Max-SAT 問題の節分割記述

At Site 1

$$1: x_1 \vee x_3 \text{ (weight = 2)}$$

$$2: \bar{x}_2 \vee \bar{x}_3 \text{ (weight = 3)}$$

At Site 2

$$3: x_1 \vee x_2 \vee \bar{x}_3 \text{ (weight = 1)}$$

$$4: \bar{x}_1 \text{ (weight = 2)}$$

1. で述べたとおり, このような状況は, 分散推論 [1] や Dynamic SAT 問題 [12], Dynamic Max-SAT 問題 [22] など実用上有用な問題においてしばしば現れる.

この節分割記述では, 一般に, 複数の部分節集合同士で変数が一部共有されることに注意されたい. 例えば, 例 2 では変数 x_1, x_2, x_3 (すなわちすべての変数) が両サイトの節集合間で共有されている. なお, 共有されている変数には, 通常, 同じ値を割り当てなければならないという制約が存在する. 本論文では, これを共有変数に関する一致制約と呼ぶ. このような節分割された Weighted Max-SAT 問題の目的は, すべての一致制約を満たし, かつ, 充足されない節の重み和を最小にする変数への真偽値の割当を求めることである.

3. Multi-MaxSAT

本論文では, 節分割された Weighted Max-SAT 問題を効率良く解くことを目指して, Multi-MaxSAT という新しい解法を提案する. Multi-MaxSAT は, 後述するパラメータを調整しながら部分節集合を順に繰り返し解く逐次型解法である. 言い換えると, 節分割で得られた部分節集合を一つの操作単位とみなした解法であり, その点が従来の Weighted Max-SAT アルゴリズムにない Multi-MaxSAT の大きな特徴である.

Multi-MaxSAT では, ラグランジュ分解と呼ばれるアイデアを利用するため, 本章ではまず, 節分割された Weighted Max-SAT 問題のラグランジュ分解について例を用いて説明する.

3.1 ラグランジュ分解

節分割された Weighted Max-SAT 問題の目的は, 共有変数に関するすべての一致制約を満たし, かつ, 充足されない節の重み和を最小にすることである. 例 2 において, このことを明確に記述すると以下の最小化問題 P が得られる.

$$P \left\{ \begin{array}{ll} \min. & 2C_1(\mathbf{x}^1) + 3C_2(\mathbf{x}^1) \\ & + C_3(\mathbf{x}^2) + 2C_4(\mathbf{x}^2) \\ \text{s.t.} & x_1^1 - x_1^2 = 0, \\ & x_2^1 - x_2^2 = 0, \\ & x_3^1 - x_3^2 = 0, \\ & \mathbf{x}^1 = (x_1^1 \ x_2^1 \ x_3^1) \in \{0, 1\}^3, \\ & \mathbf{x}^2 = (x_1^2 \ x_2^2 \ x_3^2) \in \{0, 1\}^3. \end{array} \right.$$

ここで, 変数ベクトル $\mathbf{x}^k$ はサイト k における変数の組で, この場合, その要素は x_1^k, x_2^k, x_3^k の三つであり, それぞれ変数 x_1, x_2, x_3 に対応する. 共有変数 x_i に関する一致制約は, この例の場合, $x_i^1 - x_i^2 = 0$ という等式制約で表現されている. また, $C_j(\mathbf{x}^k)$ は, 変数ベクトル $\mathbf{x}^k$ がある値をとるとき, 節 j が充足されるなら 0, 充足されないなら 1 を返す関数とする.

この最小化問題 P のすべての等式制約を緩和したラグランジュ緩和問題を作る. その結果, 以下の最小化問題 $P(\boldsymbol{\mu})$ が得られる.

$$P(\boldsymbol{\mu}) \left\{ \begin{array}{ll} \min. & 2C_1(\mathbf{x}^1) + 3C_2(\mathbf{x}^1) \\ & + C_3(\mathbf{x}^2) + 2C_4(\mathbf{x}^2) \\ & + \mu_1^{12}(x_1^1 - x_1^2) \\ & + \mu_2^{12}(x_2^1 - x_2^2) \\ & + \mu_3^{12}(x_3^1 - x_3^2) \\ \text{s.t.} & \mathbf{x}^1 = (x_1^1 \ x_2^1 \ x_3^1) \in \{0, 1\}^3, \\ & \mathbf{x}^2 = (x_1^2 \ x_2^2 \ x_3^2) \in \{0, 1\}^3. \end{array} \right.$$

なお, $\boldsymbol{\mu} = (\mu_1^{12} \ \mu_2^{12} \ \mu_3^{12})$ は, サイト 1 と 2 の間の各共有変数の一致制約に対するラグランジュ乗数を要素とする実数値ベクトルであり, ラグランジュ乗数ベクトルと呼ばれる. $\boldsymbol{\mu}$ の任意の値に対して, ラグランジュ緩和問題 $P(\boldsymbol{\mu})$ の最適値は原問題 P の最適値の下界となる [19].

問題 $P(\boldsymbol{\mu})$ は, 実質的には無制約の最小化問題である. よって目的関数のみに注目し, その項をサイト 1 に由来するものと, サイト 2 に由来するものに分け, 次の二つの最小化問題 $P^1(\boldsymbol{\mu})$ と $P^2(\boldsymbol{\mu})$ を作る.

$$P^1(\boldsymbol{\mu}) \left\{ \begin{array}{ll} \min. & 2C_1(\mathbf{x}^1) + 3C_2(\mathbf{x}^1) \\ & + \mu_1^{12}x_1^1 + \mu_2^{12}x_2^1 + \mu_3^{12}x_3^1 \\ \text{s.t.} & \mathbf{x}^1 = (x_1^1 \ x_2^1 \ x_3^1) \in \{0, 1\}^3. \end{array} \right.$$

$$P^2(\boldsymbol{\mu}) \left\{ \begin{array}{ll} \min. & C_3(\mathbf{x}^2) + 2C_4(\mathbf{x}^2) \\ & - \mu_1^{12}x_1^2 - \mu_2^{12}x_2^2 - \mu_3^{12}x_3^2 \\ \text{s.t.} & \mathbf{x}^2 = (x_1^2 \ x_2^2 \ x_3^2) \in \{0, 1\}^3. \end{array} \right.$$

なお、 μ の任意の値について、 $P^1(\mu)$ と $P^2(\mu)$ の最適値の和は $P(\mu)$ の最適値であり、かつ、原問題 P の最適値の下界となる。ここで、 $P^1(\mu)$ と $P^2(\mu)$ は、それぞれ次のような Weighted Max-SAT 問題と解釈できる。

At Site 1

$$1: x_1^1 \vee x_3^1 \text{ (weight} = 2\text{)}$$

$$2: \bar{x}_2^1 \vee \bar{x}_3^1 \text{ (weight} = 3\text{)}$$

$$x_1^1 \text{ (weight} = \mu_1^{12}\text{)}$$

$$x_2^1 \text{ (weight} = \mu_2^{12}\text{)}$$

$$x_3^1 \text{ (weight} = \mu_3^{12}\text{)}$$

At Site 2

$$3: x_1^2 \vee x_2^2 \vee \bar{x}_3^2 \text{ (weight} = 1\text{)}$$

$$4: \bar{x}_1^2 \text{ (weight} = 2\text{)}$$

$$x_1^2 \text{ (weight} = -\mu_1^{12}\text{)}$$

$$x_2^2 \text{ (weight} = -\mu_2^{12}\text{)}$$

$$x_3^2 \text{ (weight} = -\mu_3^{12}\text{)}$$

なお、各 Weighted Max-SAT 問題には、緩和された一致制約に対応する単位節が含まれており、その重みはラグランジュ乗数の値により決まる（一方は +1 倍、他方は -1 倍された値が重みとなる）。したがって、ラグランジュ乗数の値が決まれば、 $P^1(\mu)$ と $P^2(\mu)$ の最適解と最適値は任意の Weighted Max-SAT アルゴリズムにより求められる。

以上の要領で、原問題 P から部分問題 $P^1(\mu)$ と $P^2(\mu)$ を作る操作は、一般にラグランジュ分解 [17] と呼ばれている。

ラグランジュ分解後の部分問題群 $\{P^k(\mu) \mid k = 1, \dots, m\}$ と原問題 P の関係に関する以下の命題は、原問題 P に対応する Weighted Max-SAT 問題を解く上で重要な役割を果たす。本論文で提案する Multi-MaxSAT は、この命題を基本設計原理とする解法である。

[命題 1] ラグランジュ乗数ベクトル μ のある値のもとでラグランジュ分解後のすべての部分問題群 $\{P^k(\mu) \mid k = 1, \dots, m\}$ の最適解がすべての一致制約を満たしているとき、それらの最適解は原問題 P に対応する Weighted Max-SAT 問題の最適解を構成している。

なお、最適化問題に対しラグランジュ分解を適用し

た研究が、近年、いくつか報告されている。例えば、非線形計画問題 [3]、スケジューリング問題 [14]、サプライチェーン最適化問題 [21] に対し、同様の解法が提案されている。また、[10], [11] では、ラグランジュ分解と分散制約充足の手法を用いて、一般化相互割当問題に対するピアツーピア型の分散型解法が提案されている。

3.2 基本的なアイデア

前節で示したとおり、ラグランジュ分解後の各部分問題 $P^k(\mu)$ は、ラグランジュ乗数ベクトル μ の値が決まれば Weighted Max-SAT 問題となり、任意の Weighted Max-SAT アルゴリズムで解くことができる。一方、 μ の値が変われば、 $P^k(\mu)$ は節の重みが異なる別の Weighted Max-SAT 問題となり、その最適解や最適値が変わる。Multi-MaxSAT の基本的なアイデアは、最適解がすべての一致制約を充足するまで、ラグランジュ乗数ベクトル μ の値を変化させながら部分問題群 $\{P^k(\mu) \mid k = 1, \dots, m\}$ を Weighted Max-SAT アルゴリズムで繰り返し解くというものである。

μ のある値のもとで、部分問題群の最適解がすべての一致制約を満たしていれば、先の命題 1 より原問題 P に対応する Weighted Max-SAT 問題の最適解が得られている。一方、その最適解が一部の一致制約を満たしていない場合でも、そのとき得られている最適値の和は、前述のとおり、原問題 P の最適値の下界となる。また、この場合、ある共有変数の一致制約が満たされていない（すなわち、ある部分問題の最適解ではその共有変数の値は 0、別の部分問題の最適解では 1 になっている）わけだが、例えば多数決などの何らかの規準でその共有変数の値を 0 または 1 に強制的に揃えることが可能である。このヒューリスティックな操作により、一致制約を満たしていない実行不可能解から、最適解ではないかもしれないが一致制約をすべて満たしている実行可能解を生成することができる。

すなわち、Multi-MaxSAT では、ラグランジュ乗数ベクトル μ の任意の値に対して、常に、原問題 P に対応する Weighted Max-SAT 問題の最適値の上界値と下界値、及び、上界値に対応する実行可能解を生成することができる。一般に、WalkSAT [25] などのヒューリスティックな解法では、上界値とそれに対応する実行可能解が得られるが、Multi-MaxSAT ではそれらに加えて下界値が得られる。

3.3 詳細

Multi-MaxSAT の手続きの詳細を以下に示す。

(Multi-MaxSAT の手続き)

INPUT: 原問題 P , 部分問題群 $\{P^k(\mu) \mid k = 1, \dots, m\}$, 上限時間 TM

OUTPUT: P の実行可能解 F

VARIABLES: ラグランジュ乗数ベクトル μ , (最小の) 上界値 UB , (最大の) 下界値 LB

Step1: $\mu = 0$, $UB = \infty$, $LB = -\infty$ と初期化する.

Step2: 部分問題群 $\{P^k(\mu) \mid k = 1, \dots, m\}$ を Weighted Max-SAT 問題に対する厳密解法ですべて解く.

Step3: Step2 で得られた最適解が P のすべての一致制約を満たしていれば, それを P の最適解として出力し終了する.

Step4: Step2 で得られたすべての最適値の和が LB より大きければ, それを新しい LB とする.

Step5: Step2 で得られた最適解に対しヒューリスティックスを用いて一致制約を充足させ, P の実行可能解を生成する.

Step6: Step5 の実行可能解の目的関数値が UB 未満であれば, それを新しい UB とし, その実行可能解を F とする.

Step7: LB と UB の値が等しければ, F を P の最適解として出力し終了する.

Step8: 経過時間が TM を越えれば, F を P の実行可能解として出力し終了する. そうでなければ μ の値を更新して Step2 に戻る.

以下, Multi-MaxSAT の主なステップを説明する.

Step2 で, 各部分問題 $P^k(\mu)$ を解く際には, Lazy [2] や Toolbar [7], [8], [15], [16] など既存の任意の厳密解法を利用する. なお, 後述する評価実験では, 実装が容易な Lazy を利用した.

Step5 で, 一致制約を充足させるためのヒューリスティックスには多数決を用いる. すなわち, 一致制約を満たしていないある共有変数について, 1 が割り当てられた部分問題の数と 0 が割り当てられた部分問題の数を求め, 多い方の値に一致させる. ここで, 部分問題の数が同数となる場合には, 1 でそろえた場合と 0 でそろえた場合のそれぞれについて原問題 P の目的関数値を計算し, その値が小さい方を採用する.

Step7 では, 上記のヒューリスティックな操作で得られた実行可能解の目的関数値 (すなわち上界値) が下界値に一致すれば, 得られた実行可能解は原問題 P の最適解であるため, 結果を出力して終了する.

Step8 で, μ の各ラグランジュ乗数の値更新には劣こう配法を用いる. 劣こう配法とは, ラグランジュ乗数の初期値から組織的に値を生成する繰返し手続きである. その目的は, ラグランジュ乗数の値を適切に選択することにより, 下界値を最大化することである. 具体的には, 次の手順でラグランジュ乗数の値を更新する.

部分問題 $P^k(\mu)$ と部分問題 $P^{k'}(\mu)$ が変数 x_i を共有している場合, まず, その一致制約 $x_i^k - x_i^{k'} = 0$ に対する劣こう配 $G_i^{kk'}$ を次式で計算する.

$$G_i^{kk'} = x_i^k - x_i^{k'}$$

すなわち, 劣こう配 $G_i^{kk'}$ は, 変数 x_i に関する両問題間の一致制約が満たされていれば 0, 満たされていない場合は +1 または -1 となる. この計算を, 各共有変数について, その一致制約が明示的に定義されたすべての部分問題のペアについて行う.

次に, これらの劣こう配と, 現時点での上界値 UB , 下界値 LB , 及び, 初期値が 2 で時間経過とともに次第に減少する外部パラメータ π を用いて, ステップ長 (ラグランジュ乗数の更新幅) T を次式で計算する.

$$T = \frac{\pi(UB - LB)}{\sum_{k,k',i} (G_i^{kk'})^2}$$

直感的には, 一般に上界値と下界値の差が大きい探索の序盤ではステップ長を大きくしてラグランジュ乗数を「大胆に」更新し, その差が小さい終盤ではステップ長を小さくしてラグランジュ乗数を「慎重に」更新する. なお, ステップ長をこの式で計算した場合, 最適解に収束する理論的な保証はないが, 実際には多くの問題で非常に有効であることが知られている [17].

最後に, 劣こう配 $G_i^{kk'}$ とステップ長 T を用いて, ラグランジュ乗数 $\mu_i^{kk'}$ ($P^k(\mu)$ と $P^{k'}(\mu)$ の間の共有変数 x_i の一致制約に対応するラグランジュ乗数) を, 次のように更新する.

$$\mu_i^{kk'} \leftarrow \mu_i^{kk'} - T \cdot G_i^{kk'}$$

すなわち, 互いに共有変数の値が一致している場合には劣こう配 $G_i^{kk'}$ は 0 であり, ラグランジュ乗数の値はそのまま維持されるが, $G_i^{kk'}$ が +1 のとき ($P^k(\mu)$ では $x_i^k = 1$, $P^{k'}(\mu)$ では $x_i^{k'} = 0$ と割り当てられたとき), ラグランジュ乗数の値はステップ長 T だけ減少する. 3.1 のラグランジュ分解後の各サイトの Weighted Max-SAT 問題の例で示したとおり, ラグランジュ乗

数は一致制約に由来する単位節の重みに相当するため、この結果、 $P^k(\mu)$ の単位節 x_i^k の重みが減少し、 $P^{k'}(\mu)$ の単位節 $x_i^{k'}$ の重みが増加する。その結果、次の部分問題を解く際には、 $P^k(\mu)$ では $x_i^k = 1$ とすることが相対的に重視されなくなり、一方、 $P^{k'}(\mu)$ では $x_i^{k'} = 0$ とすることが相対的に重視されなくなる。すなわち、「互いに歩み寄る方向」に調整される。一方、 $G_i^{kk'}$ が -1 のとき ($P^k(\mu)$ では $x_i^k = 0$ 、 $P^{k'}(\mu)$ では $x_i^{k'} = 1$ と割り当てられたとき) には、ラグランジュ乗数の値はステップ長 T だけ増加し、結果、 $P^k(\mu)$ の単位節 x_i^k の重みが増加し、 $P^{k'}(\mu)$ の単位節 $x_i^{k'}$ の重みが減少する。その結果、次の部分問題を解く際に、先と同様の調整が行われることになる。

なお、ラグランジュ乗数ベクトル μ のサイズは、共有変数に対する一致制約を部分問題間でどう定義するかによる。例えば、ある共有変数が m 個の部分問題に共有されている場合、すべての部分問題のペア間に一致制約を定義すれば $(m(m-1)/2)$ 個のラグランジュ乗数が必要であり、一方、最小限のペア間で定義することにすれば $(m-1)$ 個の乗数が必要になる。今、問題が m 個の部分問題からなり、共有変数が n 個あるとすると、 μ のサイズは、前者の定義方法では $O(m^2n - mn)$ であり、後者の定義方法では $O(mn - n)$ となる。共有変数の値をできるだけ一致させるためには前者の定義方法が望ましいと思われるが、後者に比べ μ のサイズが大きくなるという欠点がある。

4. 評価実験

Multi-MaxSAT の性能を評価するために、本論文では 2 種類の実験を行う。まず、最初の実験では、問題の他の設定は固定したまま、部分問題間で共有される変数の割合を変化させた場合の Multi-MaxSAT 及び比較対象の解法の性能を計測する。また、次の実験では、同じく問題の他の設定は固定したまま、部分問題の数を変化させた場合のそれぞれの解法の性能を計測する。なお、今回の実験では、簡単化のため、各共有変数はすべての部分問題間で共有されるものとし、その一致制約はすべての部分問題のペア間で定義されるものとした。

比較対象は、2006 年の First Max-SAT evaluation の Weighted Max-SAT 部門で最も良い成績をあげた厳密解法である Toolbar [7], [8], [15], [16], 及び、代表的なヒューリスティック解法である WalkSAT [25]

である。Toolbar については、開発者のサイト^(注1)より上記の evaluation で利用された Version 3.0 の C ソースコードを入手し、Athlon64 3500+, メモリ 1 GB, Linux という環境で gcc4.1.0 を用いてコンパイルし実行した。一方、Multi-MaxSAT と WalkSAT については、それぞれ Java で実装し、同じ環境で JDK1.5.0_12 を用いてコンパイルし実行した。なお、Multi-MaxSAT では、各部分問題を解く際に任意の Weighted Max-SAT アルゴリズムを利用できるが、今回の実験では、実装が比較的容易な Lazy [2] を利用した。

4.1 共有変数の割合に関する性能比較

この実験では、次の二つのケース

- ケース 1
 - － 部分問題の数 10
 - － 各部分問題に含まれる変数の数 10
 - － 各部分問題に含まれる節の数 100
- ケース 2
 - － 部分問題の数 10
 - － 各部分問題に含まれる変数の数 10
 - － 各部分問題に含まれる節の数 50

を考え、それぞれのケースについて、各部分問題の 10 個の変数のうち、1 個がすべての部分問題で共有されている (共有変数の割合が 0.1)、2 個がすべての部分問題で共有されている (共有変数の割合が 0.2)、... という要領で、共有変数の割合が 0.1 から 1.0 まで計 10 個の問題設定を用意し、各設定について節分割された Weighted Max-3SAT の問題例をランダムに 10 問作成した。ケース 1 では、各部分問題は 10 個の変数とそれに関する 100 個の節からなり、一方、ケース 2 では、各部分問題は 10 個の変数とそれに関する 50 個の節からなる。したがって、ケース 1 の方が、部分問題ごとの「節のクラスター」がより密な問題とみなすことができる。

なお、この問題設定では、共有変数の割合が多くなるほど全体の Weighted Max-3SAT 問題における実質的な変数の数が少なくなる。したがって、節分割記述を用いずに全体の問題を一つの Weighted Max-3SAT 問題として解く Toolbar や WalkSAT にとっては、共有変数の割合が多くなるほど探索空間は小さくなる。

図 1 は、ケース 1 の問題例に対して、各解法によ

(注1) : <http://mulcyber.toulouse.inra.fr/gf/download/frsrelease/113/155/toolbar.tar.gz>

り 30 分以内に得られた実行可能解の質を示している。解の質とは得られた実行可能解の目的関数値を最適値で割ったものである。横軸は共有変数の割合であり、割合の各値ごとにランダムに生成した 10 問の平均値をプロットした。また、データのばらつきを示すために平均値の上下に最悪値と最良値を表すエラーバーを

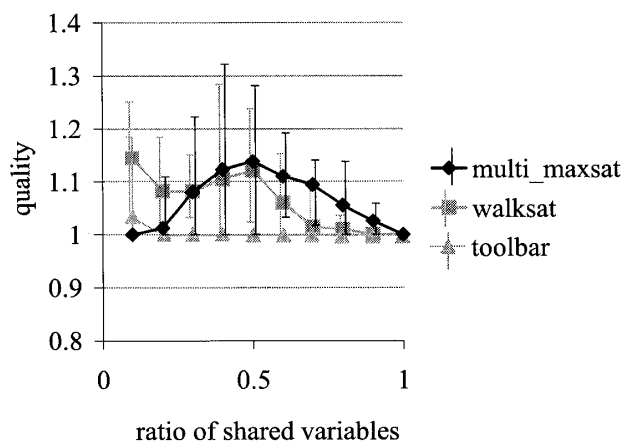


図 1 解の質の平均 (ケース 1)

Fig. 1 Average solution quality for case 1.

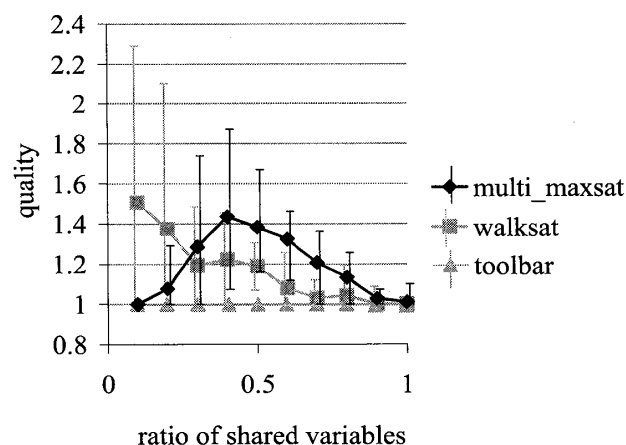


図 2 解の質の平均 (ケース 2)

Fig. 2 Average solution quality for case 2.

表示している。図に示すとおり、共有変数の割合が小さいときには、Multi-MaxSAT では最適解あるいはそれに近い解が求まっている。一方、共有変数の割合が大きときには Multi-MaxSAT の解の質はやや劣る。なお、厳密解法である Toolbar は共有変数の割合が 0.1 のとき以外はすべて 30 分以内に最適解を発見した。

一方、図 2 はケース 2 の結果である。図 1 と定性的には同様の結果だが、部分問題ごとの節のクラスターがあまり密でない場合、共有変数の割合が大きときの Multi-MaxSAT の性能はあまり良くない。しかし、共有変数の割合が小さいときには最適解かそれに近い解が得られている。

表 1 は、ケース 1 の問題例に対して、Multi-MaxSAT と Toolbar のそれぞれで最適解が得られるまでにかかった処理時間 (単位は ms) である。Multi-MaxSAT については、30 分以内に最適解の発見に成功した問題例の割合を ratio 欄に示し、その成功した問題例のみの処理時間を示している。なお、すべての問題例で最適解の発見に失敗したところは N/A とした。この表より明らかとなっており、Multi-MaxSAT は、常に最適解を発見できるとは限らないが、共有変数の割合が小さいときには Toolbar に比べ非常に効率良く最適解を発見している。特に共有変数の割合が 0.1 のときには、Toolbar よりも平均で約 8000 倍高速に最適解を発見している。一方、共有変数の割合が大きときには、実質的な変数の数が減って探索空間が小さくなるため、Toolbar では効率良く最適解が求まるが、Multi-MaxSAT では最適解の発見に失敗する機会が多くなる。

表 2 は、ケース 2 での Multi-MaxSAT と Toolbar の処理時間を示している。部分問題ごとの節のクラスターが相対的に疎な問題では、Toolbar の性能が劇的

表 1 処理時間 [ms] (ケース 1)
Table 1 Processing time [ms] for case 1.

共有変数の割合	Toolbar			Multi-MaxSAT			
	best	mean	worst	best	mean	worst	ratio
0.1	1492250	16732737	93831100	1780	2114	3209	1.0
0.2	42580	225750	1059380	1938	6121.9	12122	0.7
0.3	2060	7996	28850	16780	22249.5	27719	0.2
0.4	130	312	650	2570	6586.5	10603	0.2
0.5	110	141	190	N/A			0
0.6	100	126	150	N/A			0
0.7	110	124	140	N/A			0
0.8	90	103	120	N/A			0
0.9	80	99	110	N/A			0
1.0	70	112	150	N/A			0

表 2 処理時間 [ms] (ケース 2)
Table 2 Processing time [ms] for case 2.

共有変数の割合	Toolbar			Multi-MaxSAT			ratio
	best	mean	worst	best	mean	worst	
0.1	250	1210	5890	822	2005.6	4090	1.0
0.2	350	2909	7650	1427	4186.4	5554	0.5
0.3	100	611	11310	2466	2722	2978	0.2
0.4	80	140	390	N/A			0
0.5	50	81	150	N/A			0
0.6	50	71	90	N/A			0
0.7	50	59	80	N/A			0
0.8	40	63	80	N/A			0
0.9	50	60	70	N/A			0
1.0	50	61	100	N/A			0

に改善されることが分かる。Toolbar は、ランダムで節の数が少ない問題に対して特に有効であると報告されているが、ケース 2 の設定はそのような問題に近い。一方、Multi-MaxSAT では、節のクラスタが疎になると最適解の発見に失敗することが多くなり性能は劣化する。しかし、共有変数の割合が小さいときには Toolbar と同オーダの処理時間で最適解を発見できている。

4.2 部分問題の数に関する性能比較

次の実験では、問題の他の設定を固定したまま、部分問題の数を変化させた場合の各解法の性能を比較する。今回の実験では、

- ケース 3
 - 共有変数の割合 0.1
 - 各部分問題に含まれる変数の数 10
 - 各部分問題に含まれる節の数 100
- ケース 4
 - 共有変数の割合 0.2
 - 各部分問題に含まれる変数の数 10
 - 各部分問題に含まれる節の数 100

の二つのケースについて、それぞれの部分問題の数を 5 から 50 まで 5 刻みで変化させて、計 10 個の問題設定を用意し、各設定について節分割された Weighted Max-3SAT の問題例をランダムに 10 問作成した。

図 3 は、ケース 3 の各設定に対応する問題例において、それぞれの解法により 30 分以内で得られた実行可能解の質の平均を表している。図より明らかとなり、共有変数の割合が 0.1 のときには、部分問題の数が増加しても、Multi-MaxSAT では最適解を発見することができる。一方、他の二つの解法では、部分問題の数が増加すると得られる解の質が徐々に劣化する。

図 4 はケース 4 の結果である。なお、共有変数の割

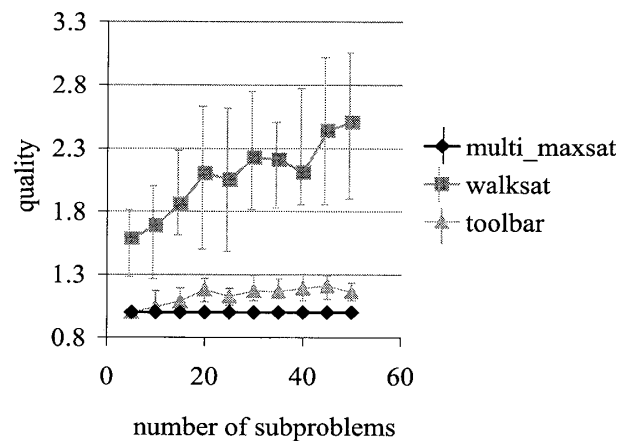


図 3 解の質の平均 (ケース 3)
Fig. 3 Average solution quality for case 3.

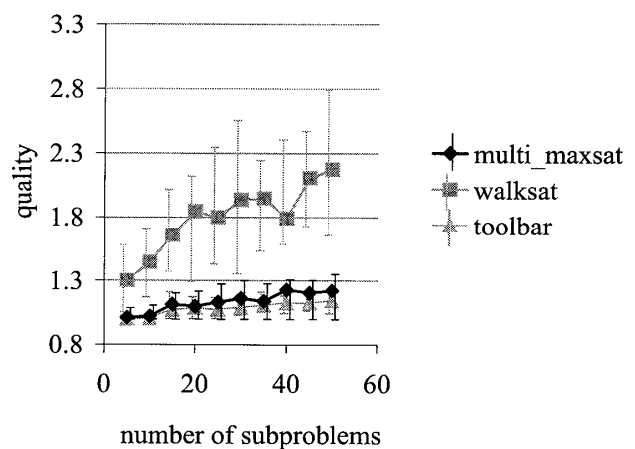


図 4 解の質の平均 (ケース 4)
Fig. 4 Average solution quality for case 4.

合が 0.2 のときには、問題例によってはいずれの解法でも制限時間内に最適解が得られない場合がある。その場合には、それぞれの解法による実行可能解の質を計算する際に、Multi-MaxSAT で得られた最良の下界値を最適値の代わりに用いた。前述したとおり、共有変数の割合が大きくなると実質的な変数の数が減少し探索空間が小さくなるため、Toolbar や WalkSAT ではケース 3 に比べ解の質が少し改善される。一方、Multi-MaxSAT では逆にケース 3 よりも解の質が劣化する。

5. む す び

本論文では、節分割された Weighted Max-SAT 問題を解く新しい解法として Multi-MaxSAT を提案した。Multi-MaxSAT は、ラグランジュ分解・調整法に基づく解法であり、Weighted Max-SAT 問題に対する従来のヒューリスティック解法と異なり、最適値の

上界を与える実行可能解だけでなく下界も求めることができる。実験による評価によれば、個々の部分問題内の節数が多く、かつ、部分問題間で共有される変数の割合が小さい場合、既存の解法である Toolbar や WalkSAT よりも Multi-MaxSAT は圧倒的に効率良く質の高い解を発見することができる。一方で、共有変数の割合が大きくなると、Toolbar や WalkSAT では探索空間が実質的に縮小するため性能が大幅に改善されるが、Multi-MaxSAT では逆に性能が劣化する。

クラスタ度の高い密な部分問題が互いに疎結合された SAT 問題や Weighted Max-SAT 問題は、分散推論や Dynamic SAT 問題、及び、Dynamic Max-SAT 問題など実用上有用な問題においてしばしば現れる。Multi-MaxSAT は、このような問題に対し非常に有効な解法であるといえる。今後の課題は、Multi-MaxSAT を分散型の解法に拡張すること、及び、値変更コストを伴う Dynamic weighted Max-SAT 問題に Multi-MaxSAT を適用することである。

文 献

- [1] P. Adjiman, P. Chatalic, F. Goasdoue, M.C. Rousset, and L. Simon, "Distributed reasoning in a peer-to-peer setting: Application to the semantic Web," *J. Artificial Intelligence Research*, vol.25, pp.269–314, 2006.
- [2] T. Alsinet, F. Manyà, and J. Planes, "A max-SAT solver with lazy data structures," *9th Ibero-American Conf. on Artificial Intelligence*, pp.334–342, 2004.
- [3] I.P. Androulakis and G.V. Reklaitis, "Approaches to asynchronous decentralized decision making," *Comput. Chem. Eng.*, vol.23, pp.341–355, 1999.
- [4] R.J. Bayardo, Jr., and R.C. Schrag, "Using CSP look-back techniques to solve real-world SAT instances," *AAAI-97*, pp.203–208, 1997.
- [5] J.M. Crawford and A.B. Baker, "Experimental results on the application of satisfiability algorithms to scheduling problems," *AAAI-94*, pp.1092–1097, 1994.
- [6] R. Dechter, *Constraint Processing*, Morgan Kaufmann Publishers, 2003.
- [7] S. de Gibry, J. Larrosa, P. Meseguer, and T. Schiex, "Solving max-SAT as weighted CSP," *CP-03*, pp.363–376, 2003.
- [8] S. de Givry, F. Heras, J. Larrosa, and M. Zytnicki, "Existential arc consistency: Getting closer to full arc consistency in weighted CSPs," *IJCAI-05*, pp.84–89, 2005.
- [9] F. Heras, J. Larrosa, and A. Oliveras, "MiniMaxSAT: A new weighted max-SAT solver," *SAT-07*, pp.41–55, 2007.
- [10] K. Hirayama, "A new approach to distributed task assignment using lagrangian decomposition and distributed constraint satisfaction," *AAAI-06*, pp.660–665, 2006.
- [11] K. Hirayama, "An α -approximation protocol for the generalized mutual assignment problem," *AAAI-07*, pp.744–749, 2007.
- [12] H.H. Hoos and K. O'Neill, "Stochastic local search methods for dynamic SAT—An initial investigation," *AAAI-2000 Workshop on Leveraging Probability and Uncertainty in Computation*, pp.22–26, 2000.
- [13] H.A. Kautz and B. Selman, "Pushing the envelope: Planning, propositional logic, and stochastic search," *AAAI-96*, pp.1194–1201, 1996.
- [14] E. Kutanoglu and S.D. Wu, "On combinatorial auction and lagrangian relaxation for distributed resource scheduling," *IIE Trans.*, vol.31, no.9, pp.813–826, 1999.
- [15] J. Larrosa, "Node and arc consistency in weighted CSP," *AAAI-02*, pp.48–53, 2002.
- [16] J. Larrosa and T. Schiex, "In the quest of the best form of local consistency for weighted CSP," *IJCAI-03*, pp.239–244, 2003.
- [17] L.S. Lasdon, *Optimization Theory for Large Systems*, Dover Publications, 2002.
- [18] P.J. Modi, W.M. Shen, M. Tambe, and M. Yokoo, "An asynchronous complete method for distributed constraint optimization," *AAMAS-03*, pp.161–168, 2003.
- [19] 森 雅夫, 松井知己, オペレーションズ・リサーチ, 朝倉書店, 2004.
- [20] H. Nabeshima, T. Soh, K. Inoue, and K. Iwanuma, "Lemma reusing for SAT based planning and scheduling," *ICAPS-06*, pp.103–113, 2006.
- [21] 西 竜志, "サプライチェーンにおける分散協調型最適化技術," *人工知能誌*, vol.19, no.5, pp.571–578, 2004.
- [22] P.C. Pinto, T.A. Runkler, and J.M.C. Sousa, "Ant colony optimization and its application to regular and dynamic MAX-SAT problems," *Advances in Biologically Inspired Information Systems*, pp.283–302, 2007.
- [23] M.R. Prasad, A. Biere, and A. Gupta, "A survey of recent advances in SAT-based formal verification," *Software Tools for Technology Transfer*, vol.7, no.2, pp.156–173, 2005.
- [24] B. Selman, H. Levesque, and D. Mitchell, "A new method for solving hard satisfiability problems," *AAAI-92*, pp.440–446, 1992.
- [25] B. Selman, H.A. Kautz, and B. Cohen, "Noise strategies for improving local search," *AAAI-94*, pp.337–343, 1994.
- [26] N. Tamura, A. Taga, S. Kitagawa, and M. Banbara, "Compiling finite linear CSP into SAT," *CP-06*, pp.590–603, 2006.

(平成 20 年 4 月 7 日受付, 9 月 9 日再受付)



黒田 陽之

2004 国立明石工業高等専門学校電気情報工学科卒. 2006 神戸大・海事科学部卒. 2008 同大大学院自然科学研究科博士前期課程了. 2008 富士通関西中部ネットテック(株)入社.



平山 勝敏 (正員)

1990 阪大・基礎工卒. 1992 同大大学院基礎工学研究科博士前期課程了. 1995 同大大学院基礎工学研究科博士後期課程了. 1995 神戸商船大学助手. 1997 同講師. 2001 同助教授. 2003 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 神戸大学大学院海事科学研究科准教授. 1999~2000 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. 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.