社団法人 情報処理学会　研究報告
IPSJ SIG Technical Report

2008－AL－119　(11)
2008／9／12

双対モデリングを用いた充足可能性問題のCNF encoding

薗部知大*1　稲葉真理*1　上田和紀*2

*1東京大学大学院情報理工学系研究科　*2早稲田大学理工学術院

充足可能性問題 (SAT) とは、与えられた論理式に対して充足可能な変数割り当てがあるかどうか証明する問題である。近年の研究の成果により、問題を解決するソルバが著しく発展し、回路検証やモデル検査などの実アプリケーションに応用が期待されている。本研究では、この SAT ソルバに入力として与える乗法標準形 (CNF) に着目し、効率よく解くことのできる CNF の作成手法の確立を目指す。具体的には、SAT と関連の深い制約充足問題 (CSP) で発案された双対モデリングの手法を SAT の CNF encoding に適用し様々な評価実験を行った。巡回セールスマン問題を含む三つの問題を取り上げ、大きな探索空間が必要な問題に対して効果的であり、SAT 問題における双対モデリングの有効性や特性を捉えることができた。

SAT CNF Encoding with Dual Modeling

Tomohiro SONOBE*1　Mary INABA*1　Kazunori UEDA*2

*1Graduate School of Information Science and Technology, The University of Tokyo
*2Faculty of Science and Engineering, Waseda University

This paper describes the CNF encodings of boolean satisfiability problems (SAT) with dual modeling. The aim of our research is to specialize the CNF encoding to speed up SAT solver activities. To achieve this aim, we have adapted dual modeling, which was proposed for Constraint Satisfaction Problems (CSP) and enables the extra level of pruning and propagation activities, to CNF encoding. We have evaluated this modeling method with three problems including the traveling salesman problem. We could grasp empirical evidences about the effectiveness of dual modeling in SAT.

1　はじめに

充足可能性問題 (SATisfiability problem, SAT 問題) は古典的な NP 完全問題であり、多項式時間での解決は難しいとされている。また、他の NP 問題の帰着対象として多くのアルゴリズムが研究されてきた。その一方で、近年、急速にこの SAT 問題を効率よく解くための研究が進み、様々な SAT ソルバが考案され、その効率を競い合っている。これにより、論理回路やソフトウェア検証等の実アプリケーションへの応用が可能になった。

しかし、問題を解く SAT ソルバが発展する一方で、入力として与える乗法標準形 (Conjunction Normal Form, CNF) にはまだまだ改良や実験の余地があると思われる。特に、SAT ソルバは与えられた CNF を解くだけであり、この CNF によって探索処理の効率性が大きく左右されることが多々ある。本研究ではこの CNF に着目し、SAT ソルバが効率よく解くことのできる CNF の作成手法の確立を目指した。

我々は、SAT と関連の深い制約充足問題 (CSP) で発

案されその有効性が確かめられている、双対モデリング [2, 3] と呼ばれる手法をSAT に応用し様々な評価実験を行った。この手法は、変数とその値を異なる視点からとらえ、それぞれのモデルを併合して一つのモデルを作成する手法である。これを巡回セールスマン問題を含む三つの問題に適用し、大きな探索空間を必要とする問題に対して効果的であり、SAT 問題における双対モデリングの有効性や特性を捉えることができた。

本論文の構成は、2 節で SAT の概要、3 節で CNF の概要と作成方法、4 節で双対モデリングの概要、5 節で実験内容と方法、6 節でハミルトン路問題のモデリングと結果、7 節で knight's tour 問題のモデリングと結果、8 節で巡回セールスマン問題のモデリングと結果、9 節で関連研究、10 節でまとめを述べる。

2 SAT とは

充足可能性問題 (SAT 問題) とは、与えられた論理式を充足させる変数の真偽値割り当てが少なくとも一つ存在するか判定する問題である。変数は真 (true) か偽 (false) のどちらかを値として取る。問題を表す論理式の形式は乗法標準形 (CNF) が一般的である。

- 乗法標準形 (CNF):
  クローズ (節) が $\wedge$ (and) で連結された論理式
- クローズ (節 clause):
  リテラルが $\vee$ (or) で連結された論理式
- リテラル: 変数の正負の出現を表す

問題例

$$P = (a \vee \neg b) \wedge (\neg a \vee c \vee d) \wedge (b \vee c)$$

CNF を充足可能 (SATisfiable) にするためには、すべてのクローズの評価を真にする必要があり、各クローズに必ず真となるリテラルが一つ存在しなければならない。それができない問題を充足不可能 (UNSATisfiable) な問題という。上記の問題例では、a,b,c に true を割り当てることで与式 P が充足される。

SAT 問題を解くシステムとして SAT ソルバがあり、与えられた論理式に対して充足可能な変数割り当てを探索し解を示す。SAT ソルバは充足不可能な問題を判定できるかどうかで大別される。充足不可能な問題を正しく判定することができる SAT ソルバを体系的 SAT ソルバと言い、本研究ではこちらを対象とする。

体系的 SAT ソルバの基本アルゴリズムは DPLL アルゴリズム [1] であり、以下の三手順から構成される。

- Decision: 変数に真偽値を割り振る。
- Deduce: Decision による推論を行う。
- Backtrack: 充足不可能 (UNSAT) な変数割り当てが生じたら、割り当てをやり直す。

最初に Decision をして変数に値を割り当て、それによって展開される論理的推論を進めることで深さ優先探索を行う。推論の結果、衝突 (conflict) が生じた場合は Backtrack を行い、直前までの変数割り当てをやり直す。衝突が生じない場合は再び Decision をし、充足解が発見できるまで探索を続ける。

3 CNF encoding

CNF を充足させるためにはすべてのクローズを充足させる必要がある。つまり、問題に対する満たすべき制約一つ一つをクローズに変換すれば、その問題の性質を記述することができる。制約の多くは含意 ($\Rightarrow$, implies) を用いて記述することが可能であるため、

- $A \Rightarrow B \Leftrightarrow \neg A \vee B$

上記のようなクローズになる。また、中間変数と呼ばれる新たな変数を導入することで、リテラル数が多いクローズの削減や複雑な制約の記述の簡便化ができる。このように、問題の性質を表す様々な制約をクローズに変換し、最低限必要な制約を揃えれば問題を解決することが可能である。

SAT 問題は古典的な NP 完全問題であり、他の NP 問題の帰着対象として様々な多項式時間変換アルゴリズムが考案されてきた。しかし、それらの多くは変換効率のみに着目されており、SAT 問題として解く場合の効率までは考慮されていないことが多い。

SAT ソルバで解くことを考えたとき、CNF は必ずしも短ければ良いというものではなく、「冗長なクローズ」を追加することで探索空間の剪定を行うことができ、効率よく解ける場合が多々ある。それによって、SAT 問題への変換効率の悪化は避けられなくなるが、処理時間が百倍向上することも珍しくない。近年の SAT ソルバには、探索中の衝突から新たなクローズを学習する

アルゴリズムが備わっており、効率よく解くためのクローズの追加をSATソルバが行うことで近年の飛躍的な発展を遂げている。しかし、与えられたCNFから元の問題を類推することは困難であり、SATソルバで学習するのでは限界がある。本研究では、この冗長ではあるが効率よく解くためのクローズを問題の構造から引き出すことに取り組んでいる。問題をCNFに変換する際に、可能な限り探索に有益なクローズを記述することで、SATソルバの探索処理を促進するCNFが期待できる。

4 双対モデリング

制約充足問題 (Constraint Satisfaction Problems, CSP) とは、SAT問題が状態を二値 (真偽) で表現するのに対して、CSPでは有限範囲内の値をとることが可能であり、それに対する制約を満たす値を探す問題である。変数の取り得る値が二つかそれ以上かという点を除けば、制約を加えて問題を解くという性質は類似していると言える。事実、SATソルバのBacktrackがCSPで提案された技術をもとにしているように、両者の関係は深い。

問題をモデリングする際に、まず変数とその値がどのような関係で、何を表すのかを考え、それから必要な制約を加えることで問題を作成する。ここで、変数とその値の関係を入れ替えても、問題をモデリングすることが可能である。たとえば、ノードに順番を割り振るような問題において、変数がノード、その値が順番を表すモデルが作成できれば、変数が順番、その値がノードであるモデルも同様に作成可能である。そして、どちらの視点からモデリングをしても一般的に問題を解決することが可能である。双対モデリングは、この異なる視点から作成した二つまたは複数のモデルを合成して、一つのモデルを作成する手法である。

単に二つのモデルを併合するだけでは意味がなく、互いのモデルに対する探索が反映されあうことで、双対モデリングの効果が発揮される。そこで、片方のモデルにおいて変数iの値がjであるという状態に対して、もう片方のモデルにおいて変数jの値がiである状態が必ず一意に存在するため、両者の条件が等価であることを述べる制約を加えれば良い。つまり、「x_i=j $\Leftrightarrow$ y_j=i」という制約を新たに加えることで、互いのモデルの探索が反映されることを保証できる。この制約をchanneling constraintと呼ぶ。双対モデリングの概念図を図1に示す。

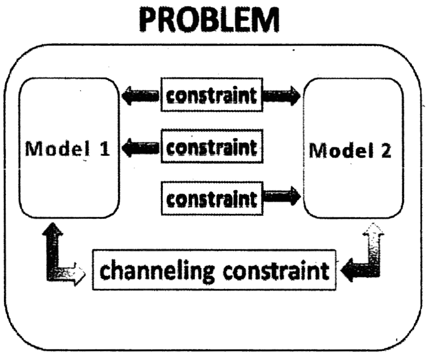

図 1: 双対モデリングの概念図

CSPにおける双対モデリングの効果は、高次の探索空間の剪定と制約伝播 (propagation) が実現できることである。channeling constraintを通じて、一視点のみのモデルでは実現できなかった先読み (look-ahead) が進み、その結果探索木の剪定に大きく貢献する。我々は、この双対モデリングをSATのCNF作成に応用することで、探索の効率化につながるのではないかと考えた。

互いのモデルのすべての制約を使用することは、多くの場合探索の妨げになる。つまり、同じ意味をなす制約であるなら、どちらか一方のモデルにのみ加えれば問題を解決することが可能であり、両者のモデルに重複させる必要はない。しかし、どちらのモデルに加えるべきかの判断は困難であり、また、同義な制約の重複が必ずしも探索を阻害するというわけではない。それらを含めて、SAT問題における双対モデリングの効果を実験で確かめる。

制約伝播と変数のドメインが異なるため、CSPと完全に同一の効果は期待できない。しかし、入力として与えられたCNFに対するSATソルバの探索はひとつの探索木に対する探索であることに変わりないものの、仮想的には二つのモデルに対する探索が並行に行われていると考えられ、互いの探索が反映しあうことでより強力な剪定が見込める。また、channeling constraint等の新たな制約の追加によるクローズ数の増加も同様の効果が期待できる。

なお、変数とその値の役割を視点、ある視点に基づい

てモデルを作成することをモデリングと定義する。また、双対モデリングを用いて作成するモデルを双対モデル、双対モデリングを用いずに作成するモデルを一視点モデルと呼ぶことで区別する。

5　実験

ハミルトン経路・閉路問題、knight's tour 問題、巡回セールスマン問題に対して、双対モデリングを用いた場合と用いない場合のモデルを作成し、それぞれに対する処理時間を計測する。なお、処理時間の上限は 1200 秒とし、結果は秒単位で表記する。制限時間を超えてしまった結果は「—」で示している。実験に使用するマシンは、CPU が Pentium M 1.5GHz、メモリが 1GB である。SAT ソルバは体系的ソルバの代表である MiniSat version1.14 を使用する。

6　ハミルトン経路・閉路問題

ノードの集合とそれらを結ぶエッジで構成されたグラフに対して、すべての頂点を一度だけ訪れるような経路・閉路が存在するかという問題である。閉路は経路の中で始点と終点を結ぶことができるものである。

6.1　モデリング

変数と値の関係は、ノードと順番で考えることができる。一つ目の視点として、変数がノード、その値が順番を表すモデルが作成でき、同様に二つ目の視点として、変数が順番、その値がノードを表すモデルが作成できる。そしてノード間の移動に関する制約は二つの表現方法がある。一つ目は、あるノードの順番が n であれば、そのノードに隣接するノードの中で、いずれかの順番が n+1 になる、というものである。もう一つは、あるノードの順番が n であれば、そのノードに隣接しないノードはすべて n+1 番目になりえない、というものである。どちらで表現しても問題を解決することができるため、それぞれで実験を行う。加える制約の詳細は、

- 各ノードは最低一回訪れられる
- 各ノードは二回以上訪れられることはない
- 各順番に最低一個のノードが割り振られる
- 各順番に二個以上のノードが割り振られない

これらに加え、移動に関する制約

- p: 次のノードは隣接する中の一つ
- n: 次のノードは隣接しないノードではない

のどちらかを加える。なお、前者を p、後者を n と表記する。

上述の四つにノード間の移動に関する制約 p と n のどちらか一方を加えてモデルを作成する。また、変数がノード、値が順番を表す一視点モデルを nodeview、変数が順番、値がノードのモデルを turnview と便宜上呼ぶことにし、この二つのモデルには四つの制約すべてを加える。それに対して、双対モデル dualview には四つの制約を二つのモデル間に分散させて配置する。また、移動に関する制約 p と n を二つのモデルにそれぞれ加える。結果の表記で、nodeview+p は nodeview モデルに移動制約 p を加えたモデルを表し、dualview+p+n は dualview モデルの nodeview モデル側に p、turnview モデル側に n を加えたモデルとなる。なお、問題のグラフはプログラムでランダムに作成したものであり、n-e という表記はそれぞれノード数とエッジ数を表す。

6.2　結果と考察

経路の結果では、n のみが組み込まれたモデルを除くと、それぞれ一長一短な結果となっている。dualview に対する制約 n は制約 p と組み合わせることで、n のみの結果と比べるとだいぶ速く解けていて、p+p のモデルと比べても遜色ない。しかし、dualview モデルが nodeview モデルや turnview モデルより必ずしも効率が良いとは言えない。逆に閉路の結果では、dualview モデルが良い結果を残していると言える。特に p+p と p+n モデルはすべてのグラフに対して制限時間以内に解けており、二つの一視点モデルでは解けなかった n80-e287 が解けたことも評価できる。

制約 n だけが組み込まれているモデルは総じて遅いと言える。しかし、dualview では p と組み合わせることで効果的に働いている結果も見られ、特に閉路問題においてはマイナスに働くというよりもプラスに働くことが多くなっている。p+p のように、二つのモデルに対して完全に同一な制約を与えるよりも、異なる制約を

与えることで探索領域の分散が生じていると考えられる。それにより、互いの探索がより効果的に反映しあうことで、探索域の大きな剪定が実現されたと推測する。解の個数が経路問題より少ない閉路問題で dualview が良い結果を残せたのはこの性質によるところが大きいと言える。

表 1: ハミルトン経路結果:処理時間 (秒)

	n75-e277	n80-e316	n85-e324	n90-e333
nodeview+p	5.58	322.43	74.91	88.01
nodeview+n	—	—	—	—
turnview+p	1.78	—	21.06	—
turnview+n	—	—	—	—
dualview+p+p	1.12	387.71	171.97	—
dualview+n+n	187.16	—	—	—
dualview+p+n	1.72	8.09	318.84	—
dualview+n+p	246.76	712.91	37.67	—

「—」は制限時間 (1200 秒) 超過

表 2: ハミルトン閉路結果:処理時間 (秒)

	n75-e231	n80-e287	n90-e400	n100-e450
nodeview+p	116.58	—	73.89	491.32
nodeview+n	—	—	—	—
turnview+p	—	—	4.38	—
turnview+n	—	—	171.80	—
dualview+p+p	128.14	176.91	67.89	229.22
dualview+n+n	—	—	167.85	—
dualview+p+n	993.69	687.74	30.46	3.23
dualview+n+p	53.18	—	65.87	—

7 knight's tour 問題

チェスで使用される駒の一種のナイトを使用して、任意の初期位置から盤面上のすべてのマスを一度だけ訪れる経路を探すという問題である。

7.1 モデリング

ハミルトン路問題のノードをマスに置き換えれば、ほぼ同様のモデルとなる。マスの接続関係は、そのマスに対してナイトが移動できるマスとの関係を記述する。作成するモデルは、

- sview: 変数がマス、値が順番を表すモデル
- tview: 変数が順番、値がマスを表すモデル
- dview: sview と tview の双対モデル

である。制約は、

- f: あるマスの順番が i なら，次に移動できるマスの中でどれかが i+1 番になる
- b: あるマスの順番が i なら，次に移動できるマスの中でどれかが i-1 番になる
- t: 各マスは最低一回訪れられる
- v: 各マスは二回以上訪れられることはない
- o: 各順番には最低一個のマスが割り振られる
- n: 各順番に二個以上のマスが割り振られない

それぞれを略称で表記する。dview+fvt+bon という表記であれば、dview モデルの sview モデル側に f,v,t を、tview 側のモデルに b,o,n の制約をそれぞれ組み込んだモデルを表す。

対象となる盤面のサイズは 8 × 8、10 × 10、12 × 12 の三種類を用意して、それぞれに対して CNF を作成する。また、組み込む制約の種類を変えての実験も行う。この問題は移動に関する規則が単純であり、盤面のサイズが決定した時点で経路の数も決定するため、ナイトの初期位置によって大きく処理時間が左右される。よって、盤面のすべてのマスを初期位置として、その処理時間の平均を比較する。なお、盤面のサイズ、初期位置にかかわらず、すべて解の存在する充足可能な問題である。

表 3: knight's tour 結果:平均処理時間 (秒)

	size8	size10	size12
sview+fbtvon	0.82	3.65	10.07
sview+fbtvn	1.10	2.88	8.61
tview+fbtvon	1.14	3.87	10.61
tview+fbvon	0.80	2.63	8.00
dview+ftv+bon	1.39	3.27	9.81
dview+btv+fon	1.70	3.18	9.56
dview+fbtv+fn	0.76	3.12	8.27
dview+btv+fn	0.88	3.26	8.38
dview+fbtv+fbn	0.93	3.11	8.98

7.2 結果と考察

盤面サイズによらず、各モデルの平均処理時間は押し並べて等しいと言えるが、注目すべき点が二つある。まず、このようにモデル間で大きな差が生じなかったことである。平均処理時間であるため、個々の初期位置の結果を比べればもう少し明確になるが、ハミルトン路問題のようなばらつきは見受けられない。特に dview モデルの結果が sview モデル及び tview モデルとほとんど差がないと言える。これは問題の規則が単純でかつ解が多いという性質に起因していると考えられる。

もう一つは、dview に対する制約の重複の効果である。dview+ftv+bon と dview+btv+fon は今回用意したすべての制約を備えており、制約が重複していないモデルである。しかし、結果をみると、dview+fbtv+fn や dview+fbtv+fbn のような、双対モデル内の二つのモデルに同義な制約が組み込まれているモデルが良い成果を残している。推察するに、双対モデルに組み込まれた二つのモデルに対して均等に制約を分散させると、互いのモデルに対する探索が類似してしまい、探索の反映効果が少なくなると考えられる。それに対して、制約を不均一に分散させることで異なる探索処理が反映しあい、結果的に探索空間の効果的な剪定が実現できたのではないかと思われる。

8 巡回セールスマン問題

複数の都市があり、セールスマンが各都市を一回ずつ通って巡回する最短の閉路を見つける問題である。都市をノード、都市間の距離をコストとして考える。

8.1 モデリング

SAT 問題は充足可能か不可能かのどちらかしか示せないため、巡回セールスマン問題を決定性問題に変換する必要がある。つまり、上限コストに対して、そのコスト以内で巡回できる閉路が存在するかしないかを問えば良い。

問題を解決するために必要な要素は、ノード、コスト、順番の三つである。あるノードに対して、そこまで到達するのにかかった小計コストとそのノードの順番があり、それらを元に、次の移動先のノードの順番と小計コストを決定する。要素が三つあるため、そのうちのどれを変数にするか選択し、残りの二つを値として考えることができる。そのため三つの一視点モデルが作成可能である。その三つのモデルから二つを選んだ双対モデルが三種類できる。さらに、この三つをすべて組み合わせて一つのモデルを作成することも可能である。この手法は multi modeling と呼ばれる。合計七種類のモデルで実験を行う。

表 4: 巡回セールスマン問題結果:ノード数 40:一視点モデルの処理時間 (秒)

	nodeview	costview	turnview
55 unsat	—	614.61	—
56 unsat	—	—	—
58 unsat	—	—	—
60 unsat	—	—	—
...	—	—	—
160 sat	—	—	41.49
165 sat	—	—	—
170 sat	—	—	—

「—」は制限時間 (1200 秒) 超過

表 5: 巡回セールスマン問題結果:ノード数 40:双対・multi モデルの処理時間 (秒)

	node+cost	node+turn	turn+cost	all
55 unsat	167.85	—	263.75	514.22
56 unsat	205.96	—	446.80	800.49
58 unsat	692.58	—	—	—
60 unsat	909.69	—	—	—
...	—	—	—	—
160 sat	1180.33	407.71	65.83	—
165 sat	—	996.39	1154.37	285.46
170 sat	—	—	108.77	228.29

- nodeview: 変数をノードとして表すモデル
- costview: 変数をコストとして表すモデル
- turnview: 変数を順番として表すモデル
- node+cost: nodeview と costview の双対モデル
- node+turn: nodeview と turnview の双対モデル
- turn+cost: turnview と costview の双対モデル
- all: 三つのモデルをすべて組み合わせた multi model

この問題では各モデルに加える制約について詳しく追及せず、また加える制約を変化させての実験も行わない。なぜなら、制約の数が多くその内容も複雑で、どのような組み合わせ方をすべきか判断が難しいためである。そのため、三つの一視点モデルには必要な制約をすべて組み込み、双対モデルと multi model ではなるべく重複を避けるように制約を配置している。しかし、制約の配置の類似性が結果に対しても同様の類似性を招く恐れがあるため、敢えて重複を許している部分も存在する。

CNF を作成する際に巡回の上限コストを指定して、そのコスト以内にグラフを巡回できれば充足可能 (SAT)、できなければ充足不可能 (UNSAT) になる。結果の表記で、「40 unsat」ならば、上限コストを 40 に指定した UNSAT な問題ということを示す。また、本論文ではノード数 40 のグラフに対する結果を載せるが、ノード数の異なるグラフでも同様の結果が確認できている。

8.2 結果と考察

この問題は他の問題と違い、充足不可能な問題が容易に作成できるため、大きな発見ができた。まず、全体的に双対・multi モデルが良い結果を残したと言える。充足可能、不可能な問題どちらに対しても制限時間以内で解くことができており、その数も一視点モデルと比べて圧倒的に多い。しかし、個々のモデルには長短があり、node+cost モデルと node+turn モデルでは充足可能、不可能な問題に対する傾向が反対になっている。上述の制約分配による影響と考えられる。

充足可能な問題は、CNF を充足させる変数割り当てが見つかり次第そこで探索を終了してその変数割り当てを解として示す。そのため、偶然に変数割り当てがうまくいく例も少なくない。turnview モデルの上限値 160 の問題に対する結果が好例である。それに対して充足不可能な問題は、すべての変数割り当てを試みる必

要があるため、速く解くためには効率的な探索が必須になる。双対・multi モデルはまさにそれを実践していると言える。

今回の実験では、上限コスト 61 以上から 160 未満の問題はすべてのモデルにおいて制限時間以内に解くことができなかった。その範囲の問題は、より解が少ないまたは解がなくても探索空間の広い難題である。巡回セールスマン問題のような最適解を求める問題では、ある境界値を越えると急激に解が少なくなるため、SAT で厳密な解を求めることは困難である。しかし、双対モデリングを用いることで、一視点モデルでは解くことのできないその境界値近傍の問題を多く解くことができた。

9 関連研究

SAT ソルバの発展に伴い実用的な応用が増え、ソルバで速く解けることを目指した CNF を得ること、例えばもとの CNF からより良い CNF への変換の研究も行われている。[4] では、検証問題の CNF を短時間で容易に変換し、より良い CNF への変換が SAT ソルバの処理時間の高速化に大きく貢献すると結論付けている。目的は同じだが、本研究では問題の構造からその性質を可能な限り引き出し、クローズ数は増えるが、効率よく解ける CNF の作成手法の確立を目指している。

10 まとめ

制約充足問題で提唱された双対モデリングを SAT 問題の CNF encoding に応用し、その有用性を確認することができた。本論文で扱った問題に対する結果だけを見れば、双対モデリングを使用した CNF encoding を行う価値は十分にある。二つの独立したモデルを channeling constraint で結ぶことで、互いのモデルに対応する探索が反映しあい、探索域の剪定が促進されると考えられる。特に、全探索域への処理が必要である充足不可能な問題や、解の個数が比較的少ない充足可能な問題のように、広大な探索域を対象とする問題には剪定が効くことで双対モデリングの活躍が期待できる。ハミルトン閉路問題と巡回セールスマン問題の結果はその一例であると言える。

しかし、制約の配分によって効果が表れなかったモデルの存在も考慮せねばならない。ハミルトン経路問題では一視点モデルに劣ることが多々あり、knight's tour 問題ではモデル間の制約の重複を避けたモデルが良い結果を残せていない。特にモデル間の制約の分配方法には不明な点が多く、さらなる調査が必要である。その他にも、衝突の回数や学習したクローズ数などの、処理時間以外のパラメータに着目した評価や、双対モデリングによる効果の理論的な証明を今後の課題とする。

CNF の作成方法は一意に定められず、その形式次第で SAT ソルバの探索効率が左右される。そのため、我々は様々な視点からとらえたモデルを組み合わせることで、一視点モデルでは実現できない有効な剪定が見込める双対モデリングを適用した。新たな制約や変数の追加による CNF サイズの拡大は避けられないものの、探索空間が大きい解の少ない、あるいは解の無い問題に対して、SAT ソルバの探索処理が促進され、優れた結果を残せた。モデル間の制約配分方法で長短が現れるものの、一視点のモデルでは解くのが困難な問題に適用させる価値は高い。

参考文献

[1] Lintao Zhang, Sharad Malik: "The Quest for Efficient Boolean Satisfiability Solvers", Proceedings of the 14th International Conference on Computer Aided Verification, pp.17-36, 2002.

[2] B. M. W. Cheng, K. M. F. Lee, and J. C. K. Wu: "Increasing Constraint Propagation by Redundant Modeling: an Experience Report", Constraints Vol.4 pp.167-192, 1999.

[3] Brahim Hnich, Barbara M. Smith, Toby Walsh: "Dual Modeling of Permutation and Injection Problems" Journal of Artificial Intelligence Research, vol.21, pp.357-391, 2004.

[4] D. Sheridan: "The Optimality of a Fast CNF conversion and its use with SAT" Proceedings of the 7th International Conference on Theory and Applications of Satisfiability Testing, 2004.