1E3-OS-4-2

ハミルトン閉路問題のSAT符号化に関する研究

Study of a SAT Encoding of Hamiltonian Cycle Problem

船越 泰輔[*1] 番原 睦則[*2] 田村 直之[*2]
Taisuke Funakoshi Mutsunori Banbara Naoyuki Tamura

[*1]神戸大学大学院システム情報学研究科情報科学専攻
Graduate School of System Informatics, Kobe University

[*2]神戸大学情報基盤センター
Information Science and Technology Center, Kobe University

In this paper, we propose a method of solving Hamiltonian Cycle Problem with SAT technology. A feature of the proposed method is an incremental solving of the problem. A condition of connectivity is not added at the first stage, therefore its solution may consist of multiple non-spanning cycles. Then, conditions of excluding those cycles are added as constraints, and the modified problem is solved again to find the next solution candidate. This process is repeated until finding a Hamiltonian Cycle or proving its unexistence. The proposed method solved 97 instances out of 119 instances of COLOR04 benchmark set. Compared with the previous work by Velev and Gao, the number of solved instances is more than twice, and the average CPU time is 25 times faster.

1. はじめに

命題論理の充足可能性判定問題 (SAT; Satisfiability Testing) は，与えられた命題論理式を真にする値割り当てが存在するか否かを判定する問題である [Biere 09, 井上 10, 梅村 10]．近年になって，SAT 問題を解くための非常に高速な SAT ソルバーが実現されている [鍋島 10, Sörensson 09, 鍋島 11]．このことを背景として，プランニング，スケジューリング，システム検証，制約充足等の問題について，それらを SAT 問題に変換して解くための **SAT 符号化** (SAT encoding) の研究が注目を集めている [田村 10b, Inoue 06, Tamura 09, Soh 10, 田村 10a, Banbara 10]．

本稿では，与えられたグラフの全頂点をちょうど一度だけ通る閉路が存在するかどうかを判定する**ハミルトン閉路問題**を対象として，SAT 符号化を用いて求解する新しい方法を提案する．提案手法の特徴は，本問題をインクリメンタルに求解する点である．まず最初の段階では，連結性の条件を追加せずに解候補を求める．したがって，得られる解候補には複数の全域的でない閉路が含まれる可能性がある．そこで，それらの閉路を排除する条件を追加し，再び他の解候補を求める．この処理をハミルトン閉路が見つかるか存在しないことが証明されるまで続ける．

以下では，ハミルトン閉路問題および既存の SAT 符号化の手法について説明した後，提案方法および評価結果について述べる．

2. ハミルトン閉路問題

グラフにおいて全頂点をちょうど一度だけ通る道は**ハミルトン道** (Hamiltonian Path) と呼ばれ，閉路は**ハミルトン閉路** (Hamiltonian Cycle) と呼ばれる．また，与えられたグラフがハミルトン道およびハミルトン閉路を持つかどうか判定する問題は，それぞれ**ハミルトン道問題** (HPP; Hamiltonian Path Problem) および**ハミルトン閉路問題** (HCP; Hamiltonian Cycle Problem) と呼ばれる．HPP と HCP のどちらも NP-完全な問題である．なお，HPP は全頂点と隣接した新しい頂点を追加したグラフ上での HCP に帰着でき，HCP はある頂点から出る辺を削除したグラフ上での HPP に帰着できる．

HPP および HCP は無向グラフと有向グラフのどちらについても定義される問題であるが，本稿では無向グラフ上の HCP を対象とする．ただし，SAT 符号化の段階では，与えられた無向グラフの各辺 u — v に対して 2 つの弧 $u \longrightarrow v$ と $v \longrightarrow u$ を対応させることで有向グラフ化する．変換した有向グラフ上のハミルトン閉路は元の無向グラフ上のハミルトン閉路となり，また逆も成り立つ．

有向グラフ (Directed Graph) $D(V, A)$ は**頂点**の集合 V と**弧** (有向辺) の集合 A からなる．ここで弧は頂点の対 (u, v) で表され u を弧の始点，v を弧の終点と呼ぶ．頂点 v について，v を始点とする弧の個数を v の**出次数** (outdegree)，v を終点とする弧の個数を v の**入次数** (indegree) という．頂点 u から頂点 v への道が存在する時，u から v へ**到達可能** (reachable) であるという．また，任意の 2 頂点についてどちらか一方から他方へ到達可能である時，その有向グラフは**連結** (connected) であるという．

有向グラフ $D(V, A)$ にハミルトン閉路が存在する時，そのハミルトン閉路を構成する弧の集合を A' とすると，部分グラフ $D'(V, A')$ の各頂点の出次数および入次数は 1 となり，またこの部分グラフは連結である．逆に，これらの条件を満たす部分グラフ $D'(V, A')$ が存在する時，A' はハミルトン閉路を構成する．したがって以下の命題が成り立つ．

命題 1. 有向グラフ $D(V, A)$ にハミルトン閉路が存在する必要十分条件は，以下の 2 条件を満たす部分グラフ $D'(V, A')$ が存在することである．

(次数の条件) D' の各頂点の出次数および入次数が 1

(連結の条件) D' は連結

連絡先: 船越泰輔，神戸大学大学院システム情報学研究科情報科学専攻，兵庫県神戸市灘区六甲台町 1-1 神戸大学情報基盤センター田村研究室，078-803-5364，funakoshi@stu.kobe-u.ac.jp

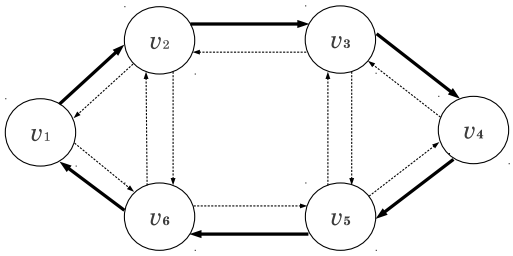

図 1: ハミルトン閉路の例

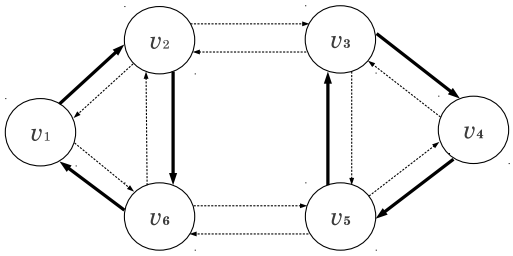

図 2: 次数の条件のみを満たす例

図 1 にハミルトン閉路の例を示す (実線の矢印がハミルトン閉路を構成する弧)．この例では次数の条件，連結の条件の双方が満たされている．図 2 に次数の条件のみを満たしている例を示す．頂点を共有しない 2 つの閉路が得られており，連結の条件を満たしていない．たとえば，頂点 v_2 と v_3 は双方向とも到達可能でない．

3. ハミルトン閉路問題の SAT 符号化

HCP の SAT 符号化に関する既存研究としては，Prestwich の方法 [Prestwich 03] および Velev & Gao による方法 [Velev 09] がある．

3.1 Prestwich の方法

Prestwich の元々の論文 [Prestwich 03] では HPP の SAT 符号化について述べられているため，ここでは論文 [Velev 09] を元に HCP に対する Prestwich の方法を説明する．

与えられた有向グラフ $D(V, A)$ の頂点を $v_1, v_2, \ldots, v_n$ とする．D のハミルトン閉路を構成する部分グラフ $D'(V, A')$ を表すために，D の各弧 (v_i, v_j) に対応させ命題変数 s_{ij} を導入する．s_{ij} は**後者変数** (successor variable) と呼ばれ，$(v_i, v_j) \in A'$ を表している．この時，次数の条件は以下のようになる．

- 各頂点 v_i に対し $\bigvee_{(v_i,v_j)\in A} s_{ij}$．この条件は v_i を始点とする弧 (v_i, v_j) の少なくとも 1 つが A' に含まれていること，すなわち D' における v_i の出次数が 1 以上であることを表す．
- 各頂点 v_i に対し，異なる 2 つの弧 $(v_i, v_j), (v_i, v_l) \in A$ のすべてについて $\neg s_{ij} \vee \neg s_{il}$．この条件は v_i を始点とする 2 つ以上の弧が A' に含まれないこと，すなわち D' における v_i の出次数が 1 以下であることを表す．
- 各頂点 v_j に対し $\bigvee_{(v_i,v_j)\in A} s_{ij}$．この条件は v_j を終点とする弧 (v_i, v_j) の少なくとも 1 つが A' に含まれていること，すなわち D' における v_j の入次数が 1 以上であることを表す．
- 各頂点 v_j に対し，異なる 2 つの弧 $(v_i, v_j), (v_k, v_j) \in A$ のすべてについて $\neg s_{ij} \vee \neg s_{kj}$．この条件は v_i を終点とする 2 つ以上の弧が A' に含まれないこと，すなわち D' における v_j の入次数が 1 以下であることを表す．

次に，連結の条件を表すために全頂点の順列を考える．まず，任意に頂点 v_f を 1 つ選び (**第 1 頂点**と呼ぶ) v_f を順列の先頭とする．次に，任意の異なる 2 頂点 v_i, v_j に対応させ命題変数 o_{ij} を導入する．o_{ij} は**順序変数** (ordering variable) と呼ばれ，順列において頂点 v_i より後に v_j があることを表す．また $(v_i, v_j) \in A'$ の時，順列において v_i より後に v_j が現れるようにする．以上の条件は，次のような節で表される．

- 各弧 $(v_i, v_j) \in A$ (ただし $j \neq f$) に対し，$s_{ij} \Rightarrow o_{ij}$ すなわち $\neg s_{ij} \vee o_{ij}$．
- 互いに異なる 3 頂点 v_i, v_j, v_k に対し，$o_{ij} \wedge o_{jk} \Rightarrow o_{ik}$ すなわち $\neg o_{ij} \vee \neg o_{jk} \vee o_{ik}$．これは順序変数に関する推移律の条件である．
- 異なる 2 頂点 v_i, v_j に対し，$\neg(o_{ij} \wedge o_{ji})$ すなわち $\neg o_{ij} \vee \neg o_{ji}$．これは順序変数に関する非対称律の条件である．
- 各頂点 v_i (ただし $i \neq f$) に対し，o_{fi}．これは第 1 頂点が順列の先頭であることを表す．

以上により o_{ij} は v_f を先頭とする頂点の順列を表し，s_{ij} が真の時 v_i より後に v_j が現れる．

ここまでの条件で v_f を始点とするハミルトン道の条件が得られている．最後に，v_f を終点とする弧の条件を加える．

- v_f を終点とする各弧 (v_j, v_f) および各頂点 v_i (ただし $i \neq f, i \neq j$) に対し，$s_{jf} \Rightarrow o_{ij}$ すなわち $\neg s_{jf} \vee o_{ij}$．

3.2 Velev & Gao の方法

Velev & Gao の方法 [Velev 09] では，前節の Prestwich の方法と同様に後者変数と順序変数を用いているが，以下に述べる手法の導入により，いくつかのベンチマーク問題について 10^4 倍以上の速度向上を実現している．

- **順序変数と推移律条件の削減:** 任意の順序変数について $o_{ij} \equiv \neg o_{ji}$ であることを利用し，順序変数を半分に減らし，また推移律の条件を表す節数を 3 分の 1 に減らす．
- **弦グラフ化による順序変数と推移律条件の削減:** 元のグラフ D に弧を追加することで弦グラフ化し，得られた弦グラフの弧に対してのみ順序変数を導入する．また弦グラフ中の三角頂点についてのみ推移律の条件を表す節を加える．このことにより符号化で生成される SAT 問題のサイズは，Prestwich の方法と比較して大幅に削減される．また，弦グラフ化するヒューリスティックスとしては $t1$ から $t12$ の 12 通りが提案されている．
- **第 1 頂点選択ヒューリスティックス:** 第 1 頂点 v_f を選択するヒューリスティックスとして $f1$ から $f11$ の 11 通りが提案されている．
- **逆推移律の導入:** 各弧 $(v_i, v_j) \in A$ および各頂点 v_k について，$s_{ij} \wedge o_{ik} \Rightarrow o_{jk}$ すなわち $\neg s_{ij} \vee \neg o_{ik} \vee o_{jk}$，および $s_{ij} \wedge o_{kj} \Rightarrow o_{ki}$ すなわち $\neg s_{ij} \vee \neg o_{kj} \vee o_{ki}$ の条件を追加し，探索の高速化を図っている．

4. 提案方法

Prestwich の方法，Velev & Gao の方法のどちらも，与えられた有向グラフにハミルトン閉路が存在するための必要十分条件のすべてを SAT 問題の節に符号化し，その後 SAT ソルバーによって求解する方法である．この時，特に連結の条件の符号化により生成される節数が大きく，与えられた有向グラフの頂点数を n とした時 n^3 に比例する節が生成される．

Velev & Gao の方法は，弦グラフ化の手法等を用いて連結の条件に必要な節数を大幅に削減しているが，400 頂点以上のグラフのハミルトン閉路を求めることは困難である．

そこで本稿では，連結の条件を事前に符号化することなく求解する方法を提案する．提案方法では，最初に次数の条件だけからなる HCP の緩和問題を考える．この問題から得られた解候補は複数の全域的でない閉路を含んでいる可能性がある．そこで，それらの閉路を排除する条件を追加し，再び他の解候補を求める．この処理をハミルトン閉路が見つかるか存在しないことが証明されるまで続ける．この方法は，条件が順次追加されるという点で一種のインクリメンタル解法となっている．

以下に提案方法のアルゴリズムの概要を示す．なお命題変数としては後者変数 s_{ij} だけが必要である．

(S1) 与えられた有向グラフ D について次数の条件を符号化した SAT 問題を生成する．

(S2) SAT 問題の解を求める．SAT 問題が充足不能であればハミルトン閉路は存在しない．

(S3) 充足可能なら，解がいくつの閉路から成るかを調べる．

(S4) 閉路が 1 つならば，それがハミルトン閉路である．

(S5) 閉路が 2 つ以上ならば，各々の閉路を排除する条件を SAT 問題に追加する．

(S6) (S2) に戻る．

上の (S5) で，たとえば v_1, v_2, v_6, v_1 が閉路として得られた，すなわち s_{12}, s_{26}, s_{61} が真だったとすると，その閉路を排除する条件の節は $\neg s_{12} \vee \neg s_{26} \vee \neg s_{61}$ となる．無向グラフを有効グラフ化した問題の場合，逆向きの閉路を排除する条件として $\neg s_{16} \vee \neg s_{62} \vee \neg s_{21}$ も追加でき，より短いステップでハミルトン閉路を求めることが期待できる．

上記のアルゴリズムでは，何度も SAT ソルバーを起動するため，オーバーヘッドが大きくなる可能性がある．しかし，SAT ソルバーを起動したまま複数の SAT 問題の求解を繰り返し行う手法である**インクリメンタル SAT 解法** (incremental SAT solving) の導入によりこのオーバーヘッドを減らすことが可能である [Eén 03b, Nabeshima 06].

5. 性能評価

Velev & Gao の方法と提案方法について，COLOR04 の Web サイト*1 で公開されている無向グラフのデータ 119 問を対象とし比較実験を行った．頂点数の最小値，最大値，平均値はそれぞれ 11, 10000, 671 であり，辺数の最小値，最大値，平均値はそれぞれ 20, 990000, 38893 である．また，実験に使用した計算機の CPU は Xeon 3.16GHz でメモリ容量は 32GB，バックエンドの SAT ソルバーとしては MiniSat 2.2 (core) [Eén 03a] を用いた．

Velev & Gao の方法については論文 [Velev 09] に記載されてるアルゴリズムを元に独自に実装した．弦グラフ化および第 1 頂点選択のヒューリスティックスとしてはそれぞれ $t4$ と $f4$ を用いた．これは，COLOR04 ベンチマーク中の 6 問について良い結果を示す組合せとして上記論文で述べられている方法である．SAT 符号化の処理および SAT ソルバーによる求解処理の CPU 時間の制限を各々300 秒に設定し計測を行った．

提案方法については，最初の SAT 符号化の処理および SAT ソルバーによる求解処理の CPU 時間の制限を各々300 秒に設定し計測を行った．提案方法の場合，(S3) から (S5) の処理も必要になるが，その CPU 時間はほぼ無視できる．

表 1 に，全 119 問のうち各方法で時間制限内に解けた問題数，符号化でタイムアウトした問題数，SAT ソルバーでタイムアウトした問題数を示す．なお，ここで「解けた」とは，ハミルトン閉路を発見できた (充足可能すなわち `SAT` の場合)，あるいはハミルトン閉路が存在しないことが証明できた (充足不能すなわち `UNSAT` の場合) ことを意味する．

表 1: 各方法で解けた問題数

	Velev&Gao の方法	提案方法
時間制限内に解けた問題数	43 問	97 問
符号化で T.O. した問題数	57 問	15 問
SAT ソルバーで T.O. した問題数	19 問	7 問

提案方法は全問題の 80%以上にあたる 97 問を解くことができた (うち `UNSAT` の場合は 16 問)．これは Velev & Gao の方法の 43 問の 2 倍以上であり，これら 43 問をすべて含んでいる．双方で解けた 43 問についての SAT ソルバーの CPU 時間の平均は，提案方法で約 0.25 秒，Velev & Gao の方法で約 6.74 秒と 25 倍以上の速度向上結果だった．

また，提案方法でハミルトン閉路を発見できたグラフの最大頂点数は 5231 (問題名 `wap04a`)，最大辺数は 294902 (`wap04a`) である．Velev & Gao の方法の場合の最大頂点数 281 (`3-Insertions_4`)，最大辺数 10396 (`miles1500`) と比較し，提案手法は約 20 倍のサイズの問題でも取り扱えることがわかった．なお，両方法で解けた 43 問について提案方法で生成された節数の平均は 13 万であり，Velev & Gao の方法の約 12%となっていた．

提案手法は SAT ソルバーを繰り返し何度も起動するが，制限時間内に解けた 97 問のうち SAT ソルバーを起動した最大回数は 233 回 (`2-FullIns_4`)，平均回数は 17.13 回だった．しかし，最大の 233 回の場合でも SAT ソルバーの総 CPU 時間は 6.63 秒であり何度も起動することによるオーバーヘッドはそれほど大きくなく，提案方法の有効性が確認できた．

6. おわりに

本稿では，ハミルトン閉路問題を SAT 符号化し SAT ソルバーを用いて解く新しい方法を提案した．提案方法では，次数の条件のみを最初の制約として用いる．そこで得られた解候補から全域的でない閉路を排除する条件を追加し，再び他の解候補を求める．この処理を繰り返すことによりインクリメンタルにハミルトン閉路問題を解いている．

COLOR04 のベンチマーク問題 119 問について比較実験を行った結果，提案方法は既存の Velev & Gao の方法よりも 2 倍以上の 97 問を解くことができ，平均 CPU 時間も 25 倍以上だった．また Velev & Gao の方法よりも約 20 倍大きなサイズの問題を取り扱うことができ，生成される節数も約 12%と

*1 `http://mat.gsia.cmu.edu/COLOR04/`

かなり小さくなっていた．

以上のように提案方法は既存方法と比較して，大幅な性能向上を実現できていると考えられる．

後者変数に関する次数の条件は，いわゆる**基数制約** (cardinality constraint) である．本稿では基数制約の SAT 符号化に単純なペア・ワイズ法を用いたが論文 [Bailleux 03, Sinz 05, Eén 06, Codish 10] などで提案されているより優れた符号化や擬ブール制約ソルバー [平山 10] を用いることが考えられる．また 4. 節でも述べたように，SAT ソルバーを起動したまま複数の SAT 問題について繰り返し求解を続ける手法であるインクリメンタル SAT 解法 [Eén 03b, Nabeshima 06] の導入も今後の課題である．

参考文献

[Bailleux 03] Bailleux, O. and Boufkhad, Y.: Efficient CNF Encoding of Boolean Cardinality Constraints, in *Proceedings of the 9th International Joint Conference on Principles and Practice of Constraint Programming (CP 2003), LNCS 2833*, pp. 108–122 (2003)

[Banbara 10] Banbara, M., Matsunaka, H., Tamura, N., and Inoue, K.: Generating Combinatorial Test Cases by Efficient SAT Encodings Suitable for CDCL SAT Solvers, in *Proceedings of the 17th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-17), LNCS 6397*, pp. 112–126 (2010)

[Biere 09] Biere, A., Heule, M., Maaren, van H., and Walsh, T. eds.: *Handbook of Satisfiability*, Vol. 185 of *Frontiers in Artificial Intelligence and Applications (FAIA)*IOS Press (2009)

[Codish 10] Codish, M. and Zazon-Ivry, M.: Pairwise Cardinality Networks, in *Proceedings of the 17th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-17), LNCS 6397*, pp. 154–172 (2010)

[Eén 03a] Eén, N. and Sörensson, N.: An Extensible SAT-solver, in *Proceedings of the 6th International Conference on Theory and Applications of Satisfiability Testing (SAT 2003), LNCS 2919*, pp. 502–518 (2003)

[Eén 03b] Eén, N. and Sörensson, N.: Temporal Induction by Incremental SAT Solving, *Electronic Notes in Theoretical Computer Science*, Vol. 89, No. 4 (2003)

[Eén 06] Eén, N. and Sörensson, N.: Translating Pseudo-Boolean Constraints into SAT, *Journal on Satisfiability, Boolean Modeling and Computation*, Vol. 2, No. 1-4, pp. 1–26 (2006)

[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)

[Inoue 06] Inoue, K., Soh, T., Ueda, S., Sasaura, Y., Banbara, M., and Tamura, N.: A Competitive and Cooperative Approach to Propositional Satisfiability, *Discrete Applied Mathematics*, Vol. 154, No. 16, pp. 2291–2306 (2006)

[井上 10] 井上 克巳, 田村 直之：SAT ソルバーの基礎, 人工知能学会誌, Vol. 25, No. 1, pp. 57–67 (2010)

[Nabeshima 06] Nabeshima, H., Soh, T., Inoue, K., and Iwanuma, K.: Lemma Reusing for SAT based Planning and Scheduling, in *Proceedings of the International Conference on Automated Planning and Scheduling 2006 (ICAPS 2006)*, pp. 103–112 (2006)

[鍋島 10] 鍋島 英知, 宋 剛秀：高速 SAT ソルバーの原理, 人工知能学会誌, Vol. 25, No. 1, pp. 68–76 (2010)

[鍋島 11] 鍋島 英知, 岩沼 宏治：リテラルブロック距離に基づく良い学習節の評価と獲得による SAT ソルバの性能改善 (特集 「AI の基本問題 SAT と応用技術」および一般), 人工知能基本問題研究会, Vol. 81, pp. 1–6 (2011)

[Prestwich 03] Prestwich, S. D.: SAT problems with chains of dependent variables, *Discrete Applied Mathematics*, Vol. 130, No. 2, pp. 329–350 (2003)

[Sinz 05] Sinz, C.: Towards an Optimal CNF Encoding of Boolean Cardinality Constraints, in *Proceedings of the 11th International Joint Conference on Principles and Practice of Constraint Programming (CP 2005), LNCS 3709*, pp. 827–831 (2005)

[Soh 10] Soh, T., Inoue, K., Tamura, N., Banbara, M., and Nabeshima, H.: A SAT-based Method for Solving the Two-dimensional Strip Packing Problem, *Fundamenta Informaticae*, Vol. 102, No. 3-4, pp. 467–487 (2010)

[Sörensson 09] Sörensson, N. and Biere, A.: Minimizing Learned Clauses, in *Proceedings of the 12th International Conference on Theory and Applications of Satisfiability Testing (SAT 2009), LNCS 5584*, pp. 237–243 (2009)

[Tamura 09] Tamura, N., Taga, A., Kitagawa, S., and Banbara, M.: Compiling Finite Linear CSP into SAT, *Constraints*, Vol. 14, No. 2, pp. 254–272 (2009)

[田村 10a] 田村 直之, 丹生 智也, 番原 睦則：SAT 変換に基づく制約ソルバーとその性能評価, コンピュータソフトウェア, Vol. 27, No. 4, pp. 183–196 (2010)

[田村 10b] 田村 直之, 丹生 智也, 番原 睦則：制約最適化問題と SAT 符号化, 人工知能学会誌, Vol. 25, No. 1, pp. 77–85 (2010)

[梅村 10] 梅村 晃広：SAT ソルバ・SMT ソルバの技術と応用, コンピュータソフトウェア, Vol. 27, No. 3, pp. 24–35 (2010)

[Velev 09] Velev, M. N. and Gao, P.: Efficient SAT Techniques for Relative Encoding of Permutations with Constraints, in *Proceedings of the 22nd Australian Joint Conference on Artificial Intelligence, LNCS 5866*, pp. 517–527 (2009)