特　集　「プランニング技術の進展と新たな応用展開」

プランニンググラフとSATプランニング

Plannig Graph and SAT Planning

鍋島　英知　山梨大学工学部コンピュータ・メディア工学科
Hidetomo Nabeshima　Department of Computer Science and Media Engineering, Yamanashi University.
nabesima@media.yamanashi.ac.jp

井上　克巳　神戸大学工学部電気電子工学科
Katsumi Inoue　Department of Electrical and Electronics Engineering, Kobe University.
inoue@eedept.kobe-u.ac.jp

Keywords: planning, Graphplan, planning graph, SAT planning.

1. はじめに

プランニングは，AIの初期から研究されている重要な分野の一つであるが，ここ数年，その研究は大きな進展を見せている．その発端にあるのが，それぞれ独立して研究されてきた二つのプランニングアルゴリズム，BlumとFurstによるGraphplan [Blum 97]およびKautzとSelmanによるSATPLAN[Kautz 96]である*1．そして，これら二つのアルゴリズムを融合させたBlackbox [Kautz 98a] は，現在世界最速のプランナ（planner）の一つである．

初期のプランニング研究においては，Greenの手法[Green 69] に見られるように，プランニングは論理的演繹の形式として定式化され，汎用の定理証明器により解くことでプランを求めることが試みられた．しかし現実的なサイズの問題を解くことは困難であったため，研究は，プランニングに特化したアルゴリズムの探求へと焦点を移していった．

Graphplanは，こうした流れから生まれてきたプランナである．その特徴は，プランニング問題を，いったん**プランニンググラフ**（planning graph）と呼ばれるデータ構造に変換することにある．プランニンググラフは，プラン探索空間の非常にコンパクトな表現であり，プランが明らかに存在しない空間が除外されている．プランニンググラフの作成後，体系的な探索アルゴリズムによりグラフからプランを抽出する．つまり，プラン探索空間をプランニンググラフにより荒く絞り込んだ後，体系的な探索によりプランを求めるのである．その結果，Graphplanは，それまでのプランナよりも何桁も速くプランを求めることができた．

一方，SATPLANは，プランニング問題を命題論理の**充足可能性問題**（Satisfiability; SAT）に変換し，それを汎用のSATソルバで解くことによりプランを求めるプランナである．このアプローチは，非常に大きくて解くことが困難なSAT問題を高速に解くことができるSATソルバの誕生が動機となっている [Kautz 92]．プランニング問題をSAT表現に変換することで，プラン探索のための手続きの流れにとらわれることなく各種制約を自由に伝搬させ，探索空間を素早く減少させることが可能となる．

そしてKautzとSelmanは，これら二つのアプローチを融合させた新しいプランナBlackboxを提案した[Kautz 98a]．彼らは，プランニンググラフがSAT表現に自動変換できることを示し [Kautz 96]，それを既存の高速なSATソルバにより解くことにより，より速くプランを求めることができることを示した．例として，Blackboxは，状態数が10^{16}，プランが14ステップ・105個のアクションからなる問題（“logistics. d” [Kautz 98b]）を約6分で解くことができる*2 [Kautz 98a]．

本稿では，これら三つのアプローチを紹介する．ここで対象とするのは，古典的なSTRIPS風のプランニング問題（すなわち（1）世界の初期状態に関する完全な記述と，（2）目標条件，（3）アクションの集合から構成される）である．まず2章でプランニンググラフを用いたプラン探索を行うGraphplanのアルゴリズムを紹介し，3章で，プランニング問題をSAT表現に変換する際の一般的な符号化方法について述べる．そして4章で，二つのアプローチの長所を融合させたBlackboxのアルゴリズムを示す．5章では，プランニング効率の改善に関する話題を紹介し，6章で，アクションの記述形式の拡張に関する話題を紹介する．

*1 [Kautz 96] で提案されたプランニングシステムには名前が付けられていなかったが，後にSATPLANという名前で参照されるようになった．

*2 Blackboxが出力するプランは半順序プラン（2章参照）であるため，複数のアクションが一つのステップに現れることがある．

2. プランニンググラフ

以下ではロケット問題 [Blum 97] を例にとって説明する．これは，場所 l にある荷物 a, b をロケット r を使って，場所 p に運ぶ問題である．初期状態を，$at(a, l) \wedge at(b, l) \wedge at(r, l) \wedge has\text{-}fuel(r)$ とし，目標条件を $at(a, p) \wedge at(b, p)$ とする．アクションは，荷物を積み込む（$load$），荷物を降ろす（$unload$），移動する（$move$）の3種類が存在する：

load (R, C, P)
前提条件：$rocket(R) \wedge cargo(C) \wedge place(P) \wedge at(R, P) \wedge at(C, P)$
効果　　：$in(C, R) \wedge \neg at(C, P)$

unload (R, C, P)
前提条件：$rocket(R) \wedge cargo(C) \wedge place(P) \wedge at(R, P) \wedge in(C, P)$
効果　　：$at(C, P) \wedge \neg in(C, R)$

move (R, F, T)
前提条件：$rocket(R) \wedge place(F) \wedge place(T) \wedge (F \neq T) \wedge at(R, F) \wedge has\text{-}fuel(R)$
効果　　：$at(R, T) \wedge \neg at(R, F) \wedge \neg has\text{-}fuel(R)$

否定の効果は，STRIPS における削除リスト中の効果を表す．変数を含むアクションは，**スキーマ**（schema）として扱われる．スキーマは，変数の異なる具体化それぞれに対応するアクションの集合を表している．例えば，$move(R, F, T)$ は，$move(r, l, p)$，$move(r, p, l)$ と具体化される．このとき，アクションの前提条件と効果も具体化されるが（例えば $rocket(r)$ や $at(r, l)$ など），ここではそれらを命題と呼ぶ．

Graphplan は，このような STRIPS 形式のプランニング問題を受け取り，半順序プランを出力するプランナである．**半順序プラン**（partially ordered plan）とは，一部のアクションについて実行順序が定められていないプランをいう．順序付けされていないアクションは，どのような順番で実行しても目標を達成することができる．例えば，半順序プラン a_1, a_2 (a_{31}; a_{32}; a_{33}), a_4 は，アクション a_{31}, a_{32}, a_{33} をどのような順番で実行しても目標を達成できる．

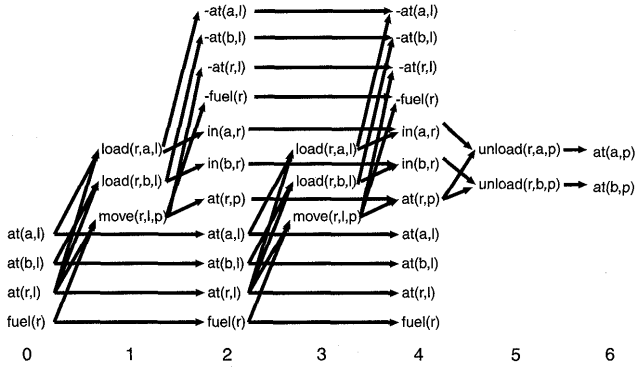

図1　ロケット問題のプランニンググラフ [Blum 97]

Graphplan は，入力としてプランニング問題を受け取ると，それをプランニンググラフに変換する．**プランニンググラフ**（planning graph）とは，レベル付き有向グラフである（図1）．偶数レベルの頂点は命題に対応し，奇数レベルの頂点はアクションに対応する．レベル0は初期状態に対応し，最終レベルは目標状態に対応している．あるレベル i の命題 f は，f を前提条件に含むレベル $i+1$ のアクションと辺で結ばれる．あるレベル j のアクション a は，a の効果に含まれるレベル $j+1$ の命題と辺で結ばれる．図中の命題どうしを結ぶ水平線は，レベル k の命題 f を次のレベル $k+2$ に伝播する何もしないアクション "$no\text{-}op(f)$" を表している．レベル l のノードの集合を L_l で表す．

プランニンググラフは，あるアクションレベルにおいて複数のアクションを含んでいるが，これらのアクションは並行して実行可能な（または，どのような順序で実行してもよい）アクションを表している．ただし，すべてのアクションが並行して実行できるとは限らない．例えば，二つのアクションのうち，片方のアクションの効果が，他方のアクションの前提条件を否定する場合，それらのアクションの実行順序は制限を受ける．このことを厳密に定義するのが，次に示す相互排他関係である．

相互排他関係（mutual exclusion relation; mutex）を次のように再帰的に定義する：

- **レベル i の二つのアクションが相互排他関係にあるのは**，以下のいずれかを満たす場合である．
  - 片方の効果が他方の効果を否定する．
  - 片方の効果が他方の前提条件を否定する．
  - 片方の前提条件のある命題と，他方の前提条件のある命題とが，レベル $i-1$ で相互排他関係にある．
- **レベル i の二つの命題が相互排他関係にあるのは**，以下のいずれかを満たす場合である．
  - 片方が他方の否定である．
  - 片方を導いたレベル $i-1$ のそれぞれのアクションが，他方を導いたレベル $i-1$ のそれぞれのアクションと相互排他関係にある．

あるレベルにおける任意の二つのアクション（命題）が相互排他関係にあるということは，そのレベルにおいてその二つのアクション（命題）が同時に成立することはあり得ない，ということを意味する．

2・1　プランニンググラフの作成

プランニンググラフ作成の手順は次のようになる．初期レベル L_0 は，初期状態において真である命題からなる．奇数レベル i の頂点集合 L_i は，次の条件を満たすすべてのアクションから構成される：

L_{i-1} で前提条件を満たすアクションであり，かつ，前提条件に含まれる任意の二つの命題間に L_{i-1} における相互排他関係がない．

さらに，各命題$f \in L_{i-1}$に対し，fを伝播する*no-op*(f)をL_iに追加する．偶数レベルの頂点集合L_{i+1}は，L_iに含まれるすべてのアクションの効果から構成される．

Graphplanは，プランニンググラフの最大レベルの頂点集合に，目標条件に含まれるすべての命題が含まれ，かつ，それら任意の二つの命題間に相互排他関係がないようなレベルになるまで，グラフを展開する．その後，次節で述べるプラン抽出アルゴリズムを適用する．もしプラン抽出の結果，プランが発見できない場合は，グラフをさらに2レベル展開し，再びプラン抽出を試みる動作を繰り返す．

先のロケット問題ではレベル6まで展開される(図1)．ただし図中において実際には，レベル3と5にはもっと多くのアクションが存在する(例えば$move(r,p,l)$など)．同様にレベル4と6にも，もっと多くの命題が存在するが，紙面の制約上必要な部分のみを記述している．

2·2 プラン抽出

Graphplanのプラン抽出アルゴリズムは，体系的な後向き探索アルゴリズムである．このアルゴリズムは，もし現在のプランニンググラフにプランが存在するならば，それを発見する．プランが存在しない場合，プラン抽出アルゴリズムは終了し，Graphplanは再びプランニンググラフの展開を行う．

プラン抽出アルゴリズムは次のようになる．ここで，グラフの最大レベルをmとし，目標条件を$(g_1 \wedge \cdots \wedge g_n)$とする．最初$i=m$である．

レベルiにおいて満たすべき副目標をG^iとする（レベルmでは，$G^m=(g_1 \wedge \cdots \wedge g_n)$である）．アルゴリズムは，$G^i$を導くようなレベル$i-1$におけるアクションの集合$A_1^{i-1}$を求める．すなわち，$A_1^{i-1}$に含まれるすべてのアクションを実行すると，副目標$G^i$が達成できるようなアクションの集合を求める．このアクションの集合は*no-op*を含んでよい．ただし，A_1^{i-1}に含まれる任意の二つのアクションの間に相互排他関係があってはならない．

このようなアクションの集合は複数存在する．それらを$\{A_1^{i-1}, \cdots, A_\kappa^{i-1}\}$とする．これらから任意に一つを選択する（バックトラックポイント）．ここでは添字順に選択するものとし，A_1^{i-1}を選ぶ．レベル$i-2$における副目標G^{i-2}は，A_1^{i-1}に含まれるすべてのアクションの前提条件の和となる．

プラン抽出アルゴリズムはこのようにして，(1) 副目標を達成するアクションの集合の選択，(2) アクションの集合から次の副目標の作成，を繰り返しながら動作する．そして，レベル0における副目標G^0が初期状態を満たせば，アルゴリズムは停止し，そこに至るまでのグラフ上の経路をプランとして出力する．もし，レベル0における副目標が初期状態を満たさなかった場合，または，それ以前に副目標を満たすようなアクションの集合を作れなかった場合，アルゴリズムはバックトラックし，再度プランの発見を試みる．

2·3 Graphplanの効率の良さ

プランニンググラフは，プラン探索空間の非常にコンパクトな表現である．[Blum 97]で示されているように，グラフのサイズとその作成にかかる時間は，命題数，アクション数に関する多項式オーダで抑えられる．一方，状態空間探索では状態空間自体がデータ構造となるため，探索と同じで木のサイズは指数オーダとなる．

Graphplanでは，プランニンググラフの作成よりも，むしろプラン抽出ステップのほうが非常に多くの時間を消費する．プラン抽出ステップは，基本的に後向きの状態空間探索に等しく，目標条件に到達する可能性のある部分状態（副目標）を再帰的に求めていくが，これは後向きに探索するにつれ指数関数的な組合せ爆発を生む．各命題レベルでk個の副目標が得られたとすると，レベルlのグラフを探索するのに$O(k^{l/2})$の選択肢が存在する．

ここで大きな効果を発揮するのがグラフ作成時に計算しておいた相互排他関係である．ある奇数レベル$2k+1$において二つのアクションa, bが相互排他関係にあるということは，初期状態から長さkのどのようなアクション列を実行したとしても，aとbを同時に実行できるような状態には到達しない（すなわち，aとbの前提条件を同時に満たすことはない）ということを意味する（もしくは，単にaとbが互いに矛盾する効果をもつアクションであるか，片方の効果が他方の前提条件を否定するアクションである）．したがって，プラン抽出時に，レベル$2k+1$においてaとbを要素にもつようなアクションの組合せは考慮する必要がない．この相互排他関係が，プラン抽出時の副目標の発生を抑え，多くのむだな探索の枝を刈り取る．例えば，タイヤの修理問題 [Blum 97] では，12ステップ，19個のアクションからなるプランを発見するまでに副目標を195個生成するが，その一方で相互排他関係により除去される副目標は1 236個にもなる．

文献 [Kambhampati 97] では，Graphplanのアルゴリズムについて，前向き状態空間探索と比較した解析を行っているのでそちらも参照されたい．

3. SATプランニング

本章では，Blackboxを生むきっかけにもなったプランニング問題をSAT問題として定式化するSATプランニング（SAT Planning）アプローチを紹介する．このアプローチは，KautzとSelmanにより最初に示された [Kautz 92]．このときの結果は，Graphplanと比較してそれほど良い結果ではなかったが，その後，より効率良くプランを求めるために改良した手法が提案され [Kautz 98a, Kautz 98b]，Graphplanを超える能力をもつことを

示した．ここでは，プランニング問題からSAT問題へのさまざまな符号化を評価した文献 [Ernst 97] から，最も良い性能を示した符号化の一つを紹介する*3．

まず，各命題と各アクションについて，時刻を表す非負整数を付加する．Graphplanと同様に，命題には偶数が，アクションには奇数が割り当てられる．例えば $move(r,l,p,i)$ は，時刻 i においてロケット r を l から p へ移動させることを意味する．また，各命題 f, $\neg f$ について，時刻 $i-1$ における命題の真偽値を時刻 $i+1$ へ伝播するアクション $no\text{-}op(f,i)$, $no\text{-}op(\neg f,i)$ を追加する．

SATプランニング手続きは，ある偶数時刻 $m(>0)$ におけるSAT問題を作成した後，それをSATソルバにより解く．もし充足不可能であれば（すなわち，長さ $m/2$ のプランが存在しなければ），時刻を進めて $m+2$ におけるSAT問題を作成し，再びSATソルバにより解くという動作を繰り返す．

ある偶数時刻 m におけるSAT問題は，次の四つの規則に従って作成される．

(1) 初期状態は時刻0において真であり，目標条件は時刻 m において真である．ロケット問題の場合，

$$at(a,l,0) \wedge at(b,l,0) \wedge at(r,l,0) \wedge has\text{-}fuel(r,0) \wedge at(a,p,m) \wedge at(b,p,m)$$

(2) 各奇数時刻 i において，すべてのアクションは，その前提条件と効果を含意する．

$$load(r,a,l,i) \supset at(r,l,i-1) \wedge at(a,l,i-1) \wedge in(a,r,i+1) \wedge \neg at(a,l,i+1)$$

(3) 各奇数時刻 i と各命題に対し，説明的フレーム公理（explanatory frame axioms）[Haas 87] を追加する．これは，命題の真偽値の変化が，あるアクションを実行することにより発生したことを含意する．ロケット問題では，次式などがある．

$$\neg in(a,r,i-1) \wedge in(a,r,i+1) \wedge \supset load(r,a,l,i) \vee load(r,a,p,i)$$

もし命題の真偽値が変化しないのであれば，そのときは *no-op* アクションが含意される．

(4) 各奇数時刻 i と矛盾するアクションの組 α, β に対し，$\neg\alpha \vee \neg\beta$ という形式の節を追加する．ここで，二つのアクションが矛盾するとは，片方の前提条件と他方の効果とが矛盾する場合をいう．

変換規則から得られたSAT問題を，任意の命題論理のSATソルバで解く．もし充足可能であるならば，そのモデルは妥当なプランを含む．なぜなら，命題の真偽値の変化は，規則 (3) よりそれを引き起こす可能性のあるアクションを真とし，真となったアクションは，規則 (2) より，その前提条件と効果が必ず成立する．このとき，並行して実行できないアクションが規則 (4) により取り除かれる．真偽値が変化しない命題は *no-op* アクションが伝播する．結果として，この変換規則は正しい状態遷移を生み，モデルには初期状態から目標状態への正しい状態遷移が含まれる．したがって，モデルから真であるアクションを取り出し，目標達成のために関係のない冗長なアクションを削除し，レベルの順に並べれば，それが求めるプランになる．

SATプランニングアプローチでは，プラン探索のための手続きの流れにとらわれることなく，各種制約を自由に伝播させ，またSATの研究分野で培われてきたさまざまなヒューリスティクスを適用することで効率良くプランを求めることができる．

4. プランニンググラフ上のSATプランニング

Graphplanの弱点は，大規模な問題において，体系的探索アルゴリズムであるプラン抽出アルゴリズムが非常に多くの時間を消費することである．これは，プランニンググラフのサイズが大きくなるにつれ，相互排他関係を活用したとしても，副目標の数が飛躍的に増大することが原因である．もし，このプラン抽出にSATソルバを利用できたならば，先に述べたように，プラン探索のための手続きにとらわれることなく，効率良く推論を進めることができる．

一方，SATプランニングにおいては，SAT変換規則 (2)，(3)，(4) が，初期状態や目標状態にかかわらず，すべての奇数時刻について等しい節集合を生成する．したがって，プランニンググラフと比較すると探索空間はSATプランニングのほうが大きい．それに対し，プランニンググラフは，初期状態から始まり，相互排他関係を活用することで明らかにむだな空間を除きながら，プランが存在する空間を符号化している．もし，プランニンググラフがSAT問題に変換できたならば，むだな探索空間を除外することができる．

KautzとSelmanは，プランニンググラフがSAT問題に自動変換できることを示し [Kautz 96]，これら二つのアプローチを融合させた新しいプランナBlackboxを提案した [Kautz 98a]．現在のところ，Blackboxは最速のプラン生成器の一つである．

プランニンググラフは，以下の規則に従ってSAT問題に変換される．ここでグラフの最大レベルを m とする．

(1) 初期レベル $L_0=\{f_1,\cdots,f_s\}$ はレベル0において真であり，目標 $\{g_1,\cdots,g_t\}$ はレベル m において真である：$f_1^0 \wedge\cdots\wedge f_s^0 \wedge g_1^m \wedge\cdots\wedge g_t^m$．

(2) レベル i において相互排他関係にある二つのアクション a,b は同時に成立しない：$\neg a^i \vee \neg b^i$．

(3) アクションは，その前提条件を含意する．すなわ

*3 ここで示す符号化は，文献 [Ernst 97] において，Regular Explanatory Encodingと呼ばれるものである．

ち，レベルiのアクションaと辺で結ばれているレベル$i-1$の命題$p_1, \cdots, p_n$に対し，以下の式を追加する：$a^i \supset p_1^{i-1} \wedge \cdots \wedge p_n^{i-1}$.

(4) 偶数レベルiに含まれる命題は，それらを効果としてもつレベル$i-1$のアクションの選言を含意する．すなわち，レベルiの命題fと辺で結ばれているレベル$i-1$のアクション$a_1, \cdots, a_n$に対し，以下の式を追加する：$f^i \supset a_1^{i-1} \vee \cdots \vee a_n^{i-1}$.

Blackboxは，Graphplanの弱点であった体系的プラン抽出アルゴリズムを，既存の高速なSATソルバによるプラン抽出アルゴリズムに置き換えることで，それまでのプランニングシステムでは解くことが困難であった大規模な問題からもプランを発見することができる．BlackboxとGraphplan，SATPLANとの性能比較については，文献[Kautz 99]を参照されたい．

Blackboxでは，体系的SATソルバと確率的SATソルバの両方を利用できる*4．例えば，体系的SATソルバとしてSatz [Li 97]や確率的SATソルバとしてWalksat [Selman 94, Selman 96]を含んでいる．確率的手法は，体系的手法よりも，多くの困難な問題において，より速く解を見つけることができる．しかし，解が最短のプランかどうかは保証されない．そこでKautzらは，確率的手法によりプランの存在をチェックし，プランが存在すれば，それが最短かどうかを体系的手法によりチェックすることを提案している[Kautz 96]．確率的手法は，プランが存在しないとき，すなわち充足不可能な場合であっても，規定の回数動作したのち終了する．これに対し体系的手法では，充足不可能なときは全解探索を試みるので，非常に多くの時間を消費する．この点からも，確率的手法と体系的手法の使い分けは有効であるといえる．

5. プランニング効率の改善

プランニング効率の改善については，最近のプランニング技術に関して要約した文献[Weld 99]によくまとめられている．ここでは，この中からいくつかの手法を紹介する．

型解析（type analysis）は，真偽値の変化しない命題をあらかじめ取り除いておくことで，グラフのサイズを抑える技術である．まず，初期状態に含まれる命題のうち，アクションの効果に現れない命題を求める．ロケット問題のアクション$move(r, l, p)$では，$rocket(r)$, $place(l)$, $place(p)$が該当する．これらの命題の真偽値は変化することがないので，アクションの前提条件から取り除くことができる．文献[Ernst 97]では，このような型解析の効果が非常に大きいことを述べている．

Graphplanで用いられているプランニンググラフの展開アルゴリズムは，初期状態からグラフを順次展開していくが，特に目標条件を指向しているわけではない．なぜなら各偶数レベルにおいて実行可能なすべてのアクションを次の奇数レベルに加えているからである．そこで，初期状態から目標条件に向かうグラフ展開に加え，目標条件から初期状態へ向かうグラフ展開を行うことで，グラフのサイズを抑えることが試みられている．Kambhampatiらは，目標達成のために使われる可能性のあるアクションと命題を単純に調べあげ，それに含まれる要素のみを使って前向きのグラフ展開を行う手法を提案している[Kambhampati 97]．筆者らは，目標条件から後向きにプランニンググラフを作成し，よりゴール指向にする手法を提案している[鍋島 00b]．

また，SATプランニングに特化したSATソルバの開発に関する研究も行われている．Giunchigliaらは，状態を構成する命題の値はアクションによって決定されるという観点に立ち，最も基本的な体系的アルゴリズムであるDavis-Putnum手続き[Davis 62]に対し，真偽値を仮定する命題変数の選択を，アクションを表す命題変数のみに限定することで，問題によっては4桁もの速度向上が見られることを示した[Giunchiglia 98]．

6. アクション記述形式の拡張

STRIPS形式のアクション記述の表現力には，その簡単な構文から，限界があることがわかる．ここでは，より表現力のある言語を扱うための研究について簡単に紹介する．

文献[Weld 99]では，**閉世界仮説**（初期状態で真偽値が明らかでない命題を偽と仮定する）をGraphplanに組み込むための効率の良い手法や，アクションの前提条件が選言を含む場合の取扱いについて述べている．また，アクションの効果に限量子を記述するためにこれまでに提案されてきた手法についても紹介している．

非決定性効果をもつアクションや，同時発生アクション，アクションの間接的な効果を表すことができる制約などをSAT表現に定式化する試みとしては[Giunchiglia 00]がある．筆者らもプランニンググラフとSATプランニングに基づく処理系AMPを開発しているが[Nabeshima 00a]，これは，プランニングだけでなく，事象の観測結果から初期状態を推定することができる．

7. おわりに

本稿では，近年大きな進展を見せたプランニングアルゴリズムについて，その発端となったGraphplanと

*4 BlackboxにはGraphplanと同じ体系的プラン抽出アルゴリズムも実装されている．どの手法を用いるかはコマンドライン引数から指定する．さらに，グラフの展開に応じて使用するプラン抽出アルゴリズムをスケジューリングすることもできる．

SATプランニングのアプローチと，それぞれの長所を融合させたプランナBlackboxのアルゴリズムを紹介した．Blackboxは，プランニンググラフによる探索空間の効率的な表現と，それをSAT問題に変換し，高速なSATソルバによりプランを求めるSATプランニングのアプローチを統合することで，従来のプランナでは扱えなかった問題を桁数の違う速さで解くことができる．

筆者らもこのアプローチに基づく処理系を開発しているが，現在ボトルネックになっていることは，充足不可能であることの証明に多大な時間を費やしてしまうことである．したがって，先に述べたように後向きグラフ展開を導入することでグラフのサイズを抑えることや，相互排他関係以外の情報の活用を模索すること[Minh 00]，また領域特有の知識をSAT問題に追加することでソルバの高速化を図ること [Kautz 98b] などが，今後の重要な課題である．

謝　辞

神戸大学の羽根田博正教授，山梨大学の岩沼宏治助教授の両氏には，有用な助言をいただきました．ここに，感謝の意を表します．

◇ 参 考 文 献 ◇

[Blum 97] Blum, A. L. and Furst, M. L.: Fast Planning through Planning Graph Analysis, *Artificial Intelligence,* Vol. 90, No. 1-2, pp. 279-298 (1997)

[Davis 62] Davis, M., Logemann, G. and Loveland, D.: A Machine Program for Theorem Proving, *Communications of the ACM,* Vol. 5, No. 7, pp. 394-397 (1962)

[Ernst 97] Ernst, M. D., Millstein, T. D. and Weld, D. S.: Automatic SAT-Compilation of Planning Problems, in *Proceedings of IJCAI-97,* pp. 1169-1177 (1997)

[Giunchiglia 98] Giunchiglia, E., Massarotto, A. and Sebastiani, R.: Act, and the Rest Will Follow: Exploiting Determinism in Planning as Satisfiability, in *Proceedings of AAAI-98,* pp. 948-953 (1998)

[Giunchiglia 00] Giunchiglia, E.: Planning as Satisfiability with Expressive Action Languages: Concurrency,Constraints, and Nondeterminism, in *Proceedings of KR-2000,* pp. 657-666 (2000)

[Green 69] Green, C. C.: Applications of theorem proving to problem solving, in *Proceedings of IJCAI-69,* pp. 219-239 (1969)

[Haas 87] Haas, A. R.: The Case for Domain-Specific Frame Axioms, in The Frame Problem in Artificial Intelligence, in *Proceedings of the 1987 Workshop,* pp. 343-348 (1987)

[Kambhampati 97] Kambhampati, S., Parker, E. and Lambrecht, E.: Understanding and Extending Graphplan,in *Proceedings of ECP-97: Recent Advances in AI Planning,* pp. 260-272 (1997)

[Kautz 92] Kautz, H. and Selman, B.: Planning as Satisfiability, in *Proceedings of the Tenth European Conference on Artificial Intelligence,* pp. 359-379 (1992)

[Kautz 96] Kautz, H. and Selman, B.: Pushing the Envelope: Planning, Propositional Logic and Stochastic Search, in *Proceedings of AAAI-96,* pp. 1194-1201 (1996)

[Kautz 98a] Kautz, H. and Selman, B.: BLACKBOX: A New Approach to the Application of Theorem Proving to Problem Solving, in *AIPS98 Workshop on Planning as Combinatorial Search,* pp. 58-60 (1998)

[Kautz 98b] Kautz, H. and Selman, B.: The Role of Domain-Specific Knowledge in the Planning as Satisfiability Framework, in *Proceedints of AIPS-98,* pp. 181-189 (1998)

[Kautz 99] Kautz, H. and Selman, B.: Unifying SAT-based and Graph-based Planning, in *Proceedings of IJCAI-99,* pp. 318-325 (1999)

[Li 97] Li, C. M. and Anbulagan: Heuristics Based on Unit Propagation for Satisfiability Problems, in *Proceedings of IJCAI-97,* pp. 366-371 (1997)

[Minh 00] Minh, B. D., Srivastava, B. and Kambhampati, S.: Investigating the effect of relevance and reachability constraints in SA Encoding of Planning, in *Proceedings of AIPS-2000,* pp. 308-314 (2000)

[Nabeshima 00a] Nabeshima, H., Inoue, K. and Haneda, H.: Implementing an Action Language Using a SA Solver, in *Proceedings of 12th IEEE International Conference on Tools with Artificial Intelligence,* pp. 96-103 (2000)

[鍋島 00b] 鍋島, 坂田, 井上, 羽根田: SAソルバと後向き推論によるアクション言語Aの実装, 第14回人工知能学会全国大会論文集, pp. 104-107 (2000)

[Selman 94] Selman, B., Kautz, H. and Cohen, B.: Noise Strategies for Improving Local Search, in *Proceedings of AAAI-94,* pp. 337-343 (1994)

[Selman 96] 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 (1996)

[Weld 99] Weld, D. S.: Recent Advances in AI Planning, *AI Magazine,* Vol. 20, No. 2, pp. 93-123 (1999)

2001年7月10日　受理

著 者 紹 介

鍋島　英知（正会員）

1998年豊橋技術科学大学大学院工学研究科情報工学専攻修士課程修了．2001年神戸大学大学院自然科学研究科情報メディア科学専攻博士課程修了．同年山梨大学工学部コンピュータ・メディア工学科助手．神戸大学博士（工学）．アクション理論などの人工知能，プランニング，オートマトンの研究に従事．情報処理学会会員．

井上　克巳（正会員）

1982年京都大学工学部数理工学科卒業．1984年同大学院工学研究科数理工学専攻修士課程修了．松下電器産業（株），（財）新世代コンピュータ技術開発機構，豊橋技術科学大学工学部情報工学系を経て，1997年より神戸大学工学部電気電子工学科助教授．京都大学博士（工学）．人工知能，論理プログラミング，計算機科学に関する教育研究に従事．