論　文

Technical Papers

実時間探索に副目標生成機構を組み込んだ実時間プラニング

Real-Time Planning by Interleaving Real-Time Search with Flexible Subgoaling

松原　繁夫*1　石田　亨*2
Shigeo Matsubara　Toru Ishida

*1 NTTコミュニケーション科学研究所
NTT Communication Science Laboratories, Kyoto 619-02, Japan.
*2 京都大学大学院工学研究科情報工学専攻
Dept. of Information Science, Kyoto University, Kyoto 606-01, Japan.

1995年9月20日　受理

Keywords: planning, real-time problem solving, real-time search, subgoaling, robot task planning.

Summary

We propose a real-time planning algorithm called RTSS (Real-Time Search with Subgoaling), which incorporates the STRIPS subgoaling function into a real-time search algorithm called RTA*. This algorithm interleaves subgoaling and real-time search processes in the way that it continues to make subgoals until the subgoal becomes simple enough for an RTA*'s heuristic function, then applies RTA* to the subproblem. RTSS can overcome the drawbacks of RTA* and STRIPS in real-time problem solving, i. e., the algorithm does not lead to a blind search in contrast to the other two. RTSS includes the function of flexibly changing subgoal sequences in order to utilize information obtained during problem solving in dynamic environments. By an analysis using a simple model, we show that the search cost can be significantly reduced by switching between subgoaling and real-time search. Furthermore, experiments on robot task planning problems show that RTSS can attain the goal without performing many superfluous actions, while RTA* and STRIPS often tend to perform a blind search that fails to attain the goal.

1. まえがき

近年さまざまな分野で実時間での問題解決に対する関心が高まっている．プラニング分野においても，実世界の問題への技術の適用を考えるうえで，実時間性は重要な概念の一つである．ロボットやプラントの制御問題では，問題解決中に世界が動的に変化する．このため，計画実行前にあらかじめオフラインで計画を立案しておいても，その計画の実行途中で失敗に終わる場合も生じ，十分な問題解決ができない．動的環境を扱う一つの方法として，計画立案と実行をインタリーブする方法がある．本論文で，我々は副目標を生成しながら実時間探索を行うプラニングアルゴリズムReal-Time Search with Subgoaling (RTSS)を提案する[松原 93, Matsubara 94a, 松原 94b]．

これまでに，実時間問題解決を目指した研究の一つとして実時間探索アルゴリズムRTA*[Korf 90]がある．これは，現状態から遷移可能な状態を求め，目標状態までの距離の推定関数を用いて，最良優先探索を行う．このアルゴリズムは計画立案と実行をインタリーブし，定数時間で次に実行する動作の決定が可能という特徴を持つ．ところが各遷移可能状態から目標状態までの距離の違いを十分判別できる推定関数が得られないとき，効率的な問題解決が困難となる．問題が大規模複雑化すると，問題解決アルゴリズムの選択だけではなく，精度の高い推定関数の獲得自体が困難となる．

この推定関数の問題の解決策として副目標を与える方法がある．例えば，目標 $A \wedge B$ が与えられたとき，

A, Bの達成を同時に考えるよりも，まず，Aの達成を考え，次にBの達成を考えるほうが問題解決が容易になる場合がある．容易になる場合の必要条件として線形仮説の成立があげられる．線形仮説とは，全体の目標を構成する副目標が独立で，副目標を任意の順序で達成すれば全体の目標が達成できるとする仮定をいう．この仮定を採用した計画立案システムとしてSTRIPS[Fikes 71]がある．STRIPSは目標の一部を副目標とし，それを達成する動作を求める．その動作実行の前提条件が現状態で未充足の場合にはその前提条件を新たな目標とする．これを目標達成まで繰り返す．この線形仮説が成立しない場合への対処法として非線形プラニング[Sacerdoti 75]がある．ところが，線形仮説が成立する場合でも，副目標生成だけでは効率的な問題解決ができないことが少なくない．すなわち，副目標を生成しても問題の複雑さが減少しない場合，盲目的な振舞いをしたり，後向きの探索を行うことと同じになって，動的環境に対応できなくなる現象がしばしば起こる．

そこで，本論文では，実時間探索のなかに副目標生成を組み込み，各手法が有効なときにだけ使用することを提案する．この切換えを行うため，目標分解基準を定義し，しきい値との大小関係により，実時間探索を行うか副目標生成を行うかを決定する．また，実時間性を向上させるため，副目標選択基準を定義して用いる．これにより，計画立案時間と計画実行時間双方を大きく削減できることを示す．

計画立案と実行をインタリーブする利点は環境の変化に柔軟に対応できることにある．単純に副目標生成と実時間探索を切り換えるだけであれば，副目標系列が固定されるため環境の変化に十分対応した計画が立案できないことが生じる．そこで，生成済みの副目標系列を環境の変化に応じて，柔軟に変更する方法も合わせて示す．

実時間問題解決を行う別の方法として，世界の状態に対して動作を規定しておく即応的なプラニングがある(例えば，[Schoppers 87])．しかし，現実世界の多様な状態に対して効果的な対応表の作成は容易ではない．世界の状態の多様性と環境の動的な性質を扱うためには，推論を行う余地のあることが望ましい．RTSSはこれを満たすものである．

以下本論文では，**2章**で実時間探索アルゴリズムRTA*と副目標生成による計画立案システムSTRIPSの概要と問題点を述べる．**3章**で実時間探索のなかに副目標生成を組み込んだアルゴリズムRTSSを提案する．**4章**では簡単なモデルを用いて，RTSSの有効性を評価する．**5章**ではRTSSで用いる二つの指標，目標分解基準と副目標選択基準を目標の抽象度に基づいて定義する方法を述べる．**6章**では移動ロボットの例題を用いて，RTSSの有効性を検証し，**7章**をむすびとする．

2. 実時間探索と副目標生成

本章では，我々が提案するアルゴリズムRTSSの基礎となる実時間探索アルゴリズムRTA*と副目標生成を行う計画立案システムSTRIPSの概要を述べる．

2・1 実時間探索：RTA*

〔1〕 概 要

RTA*[Korf 90]は，従来の探索アルゴリズムが目標を達成する完全な計画を立案するのとは異なり，計画立案と物理世界での計画実行を交互に繰り返す．RTA*は目標状態に到達するまで，以下の一連の操作を繰り返す．

1. 現状態xから遷移可能なすべての状態$\{x'\}$を求める．
2. 遷移可能なすべての状態$\{x'\}$に対し目標状態までの推定距離$f(x')$を計算する．
3. 現状態の推定距離$f(x)$を2番目に小さい$(f(x')+d)$に更新する．ここで，dは現状態xからx'までの距離である．この更新方法は，同じ状態を繰り返し通過することを防ぐ役割を持つ．
4. 最小の推定距離を持つ状態($\min(f(x')+d)$となるx')に実際に遷移する．

RTA*は定数時間で次に行う動作を決定でき，必要記憶量も計画の長さに対し線形となる．また，ある条件のもとで完全性，すなわち，解があれば必ず見つけるという性質を満たす．ただし，適格な推定関数を持つA*アルゴリズムのような解の最適性は保証されない．

〔2〕 RTA*の問題点

RTA*は推定関数の精度が高いとき有効である．ここで，推定関数の精度が高いとは，遷移可能な状態のなかから実際の距離が最小の状態を選ぶ確率が高いこととする．しかし，推定関数の精度が高くない場合には，どの状態に移動すべきかの判断を誤る．

例えば，格子空間内に二つの箱A，Bがあり，1台のロボットが目標位置まで運ぶという課題を考えよう．初期状態では箱Aのほうがロボットに近い位置

にあるとする．

推定関数を，“箱A，Bおのおのの現在位置と目標位置間のManhattan距離の和”とする．このとき，初期状態でロボットが箱を動かせる位置にいなければ，ロボットが最初どの方向に動いても推定関数値は変化しない．よって，直線的に箱の位置に向かわず，目標達成に無関係な動作が生じる．

そこで，推定関数を改良して，先の推定値に“ロボットの現在位置と最も近くにある目標位置にない箱の位置間の距離”を加えて考える．すると，箱Aを比較的むだな動作をすることなく目標の位置に運び得る．しかし，箱Aを離れて箱Bの場所へ向かうとき，追加部分の推定値の値が大きくなる．これは，目標から遠ざかることを意味し，ロボットは箱Bの現在位置に簡単に行けない．

この例では，推定関数の修正(状態ごとに細かく推定関数を規定するなど)により，精度の高い推定関数を得ることが可能である．しかし，一般には多くの要素が絡み合い，問題が複雑になると，性能向上のためには，繰り返し推定関数を修正し改良することが必要となる．また，満足のいく推定関数の獲得が困難になる場合も生じる．

2・2 副目標生成：STRIPS

〔1〕 概　要

世界の状態は述語とその否定，すなわちリテラルを用いて表現する．ここでは限量子を含まないとする．

STRIPSでは基本となる動作をオペレータを用いて表現する．オペレータは前提条件式(precondition)，削除リスト(delete-list)，追加リスト(add-list)の三つ組で定義される．それぞれオペレータ実行の条件，実行後に真でなくなるリテラル，実行後に真となるリテラルを表す．削除リスト，追加リストに記述がないリテラルは動作実行前の状態が保存されるとする．すなわち，オペレータ実行後の状態記述を求めるには，まず現在の状態記述から削除リストにあるリテラルを削除し，その後追加リストにあるリテラルを加える．

STRIPSは上述のように線形仮説をとる．この仮説に従い，目標を単純な副目標に分割し，手段目標解析により目標と現状態の差の解消に有効なオペレータを選択する．そのオペレータが実行可能な場合はそのオペレータを適用する．実行可能でなければ，オペレータの前提条件を新たな副目標として，目標状態の達成を目指す．副目標の設定により探索の方向づけが行われ，当面関連する部分のみを考えることにより，探索の深さと分岐数が削減され，問題解決効率の向上が期待できる[Korf 90]．

〔2〕 STRIPSの問題点

STRIPSは副目標間の干渉が少なく，目標状態と現状態との差を減少させる適切なオペレータが選択できる場合は有効である．しかし，適切なオペレータが選択できないとき，盲目的な後向き探索と同様の振舞いをする．

例えば，格子空間上のX地点からY地点に行くという目標を与える．目標状態と初期状態との差は“Y地点に位置する”というリテラルである．このとき前後左右どちらかの隣接地点に移動するオペレータしか与えられていなければ，そのオペレータの前提条件式から“Yの隣接地点に位置する”という副目標しか生成できない．Y地点がX地点から遠く離れていれば，この副目標が生成されても問題解決の難しさはほとんど変化しない．つまり，適用候補となるオペレータ数(状態空間探索の分岐数)を減らすことができない．Yの隣接地点が複数ある場合，本来X地点から各隣接地点に移動するのに必要なコストに差があるときでも，オペレータを見ただけではその差を評価できないこともある．これは副目標は状態の一部を表すだけであり，副目標状態と現状態との距離を推定することが難しいことによる．よって，ランダムな選択が行われ，むだな動作を行うことになる．この場合盲目的な後向き探索に等しい動作となる．後向きの探索と同様となれば，計画全体が求まるまで何ら動作を実行できず，実時間での問題解決は不可能になる．

3. 実時間プラニングアルゴリズムRTSS

RTA*とSTRIPSの欠点を解消して，実時間で効率良く問題解決を行うため，RTA*にSTRIPS的な副目標生成機構を組み入れたReal-Time Search with Subgoaling (RTSS)を提案する．このアルゴリズムは，実時間探索による目標状態の達成が容易であるかどうか，すなわち目標に対する推定関数の精度を評価する．容易と判断すればRTA*を適用する．容易でないと判断すれば，目標の分割が不十分であるとして，STRIPS的な副目標生成を行う．RTA*から見れば，目標がより単純な副目標に分解されるため，その副目標に対する推定関数の精度の向上が期待できる．STRIPSから見れば，問題分割の効果の減少後の副目標生成が有効に働かない領域で，後向きの探索が長く続くことを回避できる．

実時間探索部分は即応的システムとみなすことがで

きる．この場合，RTSSは計画立案システム(STRIPS)が即応的システム(RTA*)の振舞いを改善しているといえる．これに関連した考え方は[McDermott 92]にある．

3・1 目標分解基準と副目標選択基準

RTSSでは以下の二つの指標を考える．

目標分解基準：実時間探索から見れば，推定関数が有効に働くか働かないか，つまり，目標が推定関数に対して複雑か単純かその程度を示す．副目標生成から見れば，探索の分岐数が減少するか変化しないかを表す．現状態と目標状態が与えられたときに，副目標生成を行うか，あるいは，実時間探索を行うかの判断基準となる．

副目標選択基準：オペレータがどの程度早く実行可能となるかを表す．複数のオペレータ候補があるとき，長く副目標生成が続くオペレータ選択を避けることは実時間での問題解決効率を高める．実時間性への寄与を表す基準である．

具体的な計算方法については**5章**で述べる．

3・2 アルゴリズム

処理の概略を図1に示す．以下にRTSSのアルゴリズムの詳細を示す．本アルゴリズムは，副目標スタック，オペレータスタック，オペレータ次候補スタックを用いる．

[**初期化**]

1.目標スタックに目標状態を入れる．
2.オペレータスタック，オペレータ次候補スタックを空にする．
3.**切換え判断**へ行く．

[**切換え判断**]

1.現状態をC，目標スタック中の一番上の要素をSG，その差を$D(=SG-C)$とする．ここで，差DとはCで未充足のSGの要素をいう．
2.$D=\phi$のとき終了(**動作実行**部が目標スタック中の現状態で充足される副目標を上から順に削除するので，最終目標達成前に終了することはない)．
3.$D\neq\phi$のとき
(a) (C, SG)の目標分解基準値＞しきい値tのとき，**副目標生成**(D)へ行く．
(b) (C, SG)の目標分解基準値≦しきい値tのとき，**実時間探索**(C, GS)へ行く．ここで，副目標生成から切り換わるときは各状態の目標への推定値をクリアしてから**実時間探索**(C, SG)へ行く．

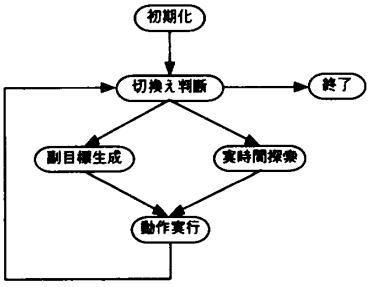

図1 アルゴリズムRTSSの概略図

[**副目標生成**(D)]

1.差Dに含まれるリテラルを追加するオペレータを求める．
2.Dに含まれるリテラルを追加するオペレータがないとき，目標スタック，オペレータスタック，オペレータ次候補スタックを操作し，後戻り処理を行う．
3.最大の副目標選択基準値を持つオペレータをoとし，オペレータスタックに加える．
4.残りのオペレータのリストをオペレータ次候補スタックに加える．
5.oの前提条件を目標スタックに加える．
6.**動作実行**へ行く．

[**実時間探索**(C, SG)]

1.現状態Cで適用可能なオペレータを調べ，適用後の状態を計算し，推定関数を用いて目標状態SGまでの推定値を求める．
2.小さいほうからの2番目の推定値に移動コストを加えた値を現状態の新たな推定値とする．
3.最小の推定値を持つ状態に遷移するオペレータoをオペレータスタックに加え，**動作実行**へ行く．

[**動作実行**]

1.オペータスタックの上から順に可能なだけのオペレータを実行する．
2.目標スタック，オペレータスタック，オペレータ次候補スタックから対応部分を削除する．
3.**切換え判断**へ行く．

上記アルゴリズムの[**切換え判断**]中のしきい値tは副目標生成を行う程度を表す．

ここでは記述を簡潔にするため動作実行部分を切り出したが，**2・1節**で述べたように，本来の実時間探索には動作実行も含める．

ここでアルゴリズムRTSSの実時間性について考える．実時間探索部は定数時間で次の動作を決定できる．副目標生成がいつ終了するかは一般にはいえない．しかし，副目標生成により推定関数の精度が単調に増加すると仮定できるなら，副目標生成部に最終期

限を設けて実時間探索部に制御を移せば，定数時間で次の動作を決定でき，しかも実時間探索だけを行うよりも性能向上が期待できる．

3・3 副目標系列の変更による RTSS の動的環境への対応

計画立案と実行を交互に行う利点は次の二つである．

(1) 実際に計画の実行を試みる段階で，その時点までに立案した計画が環境変化のために実行不可能となることを回避する．

(2) 問題解決途中で得た環境の観測結果をその後の計画立案に反映できる．

後者に関して，上記の RTSS アルゴリズムは弱さを持つ．この改善を本節で検討する．

RTSS の実時間探索部では細かく計画立案と実行が繰り返されるが，副目標生成部で生成した副目標系列は変更されない．つまり，副目標のレベルでは，環境の変化により，より短い動作系列で最終目標の達成が可能となった場合でも，古い環境認識のもとで生成した副目標系列を順に達成しようとする．これは目標達成に必要な動作数の増加を意味する．そこで，この副目標系列固定の問題を解消するため，RTSS に副目標系列変更機能を付加する．

副目標系列の変更は，**3•2 節**に示したアルゴリズムのわずかな修正で可能である．[**動作実行**]部の 1 での各動作実行の後に以下のサブルーチンを呼ぶ．

[**副目標変更**]

1. 現状態から副目標レベルで遷移可能な状態を深さ限界 l まで計算する．
2. 1 で求めた状態遷移系列のそれぞれについて，既存の副目標系列の前半部分(目標スタックの上位部分)と置換可能であるか調べる．ここでは，置換可能である条件を，既存の副目標系列中の一つの副目標が 1 で求めた遷移状態に包含されることとする．
3. 置換可能である場合，置換後と置換前の副目標系列の長さを比べ，置換後のほうが短くなれば実際に置換して復帰．複数ある場合は最短の系列になるものを選択する．
4. 3 に該当する状態遷移系列がなければ，既存の副目標系列を変更せず，復帰．

図 2 に副目標系列変更の様子を簡単に示す．この図では，既存の副目標系列 Q と副目標レベルで求めた状態遷移系列 P から，新たな副目標系列 R を生成している．**5 章**では，目標の抽象度(重要度)に基づく目

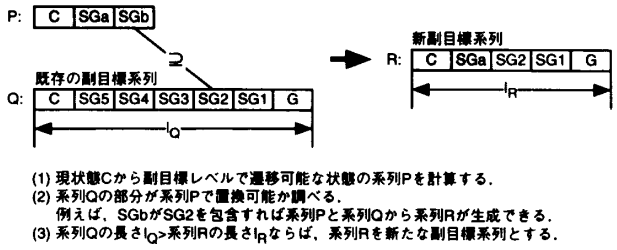

図 2 副目標系列の変更

標分解基準の計算方法を提案する．その場合，上記アルゴリズムでの副目標レベルで遷移可能な状態を展開するとは，オペレータの前提条件式中の重要度の低いリテラルを隠して遷移状態を求めることに対応する．オペレータ自身を新たに定義し直す必要はない．また，深さ限界の設定は，新たな副目標系列の発見が困難であれば，もとの副目標系列に従うことを意味する．深さ限界 l は副目標系列変更による利益とその探索のオーバヘッド間のトレードオフを制御する変数である．現状態からの副目標レベルでの探索を問題解決の最初から行うと，目標が絞れず盲目的な振舞いが現れる場合がある．また，副目標レベルでの推定関数も必要となる．このため，提案手法が有効と考えられる．

一方，実時間探索部も別の問題を有する．RTA* は隣接状態に移動するとき，現状態の推定値を，隣接状態の推定値と移動コストの和のなかで 2 番目に良い値に更新する．先に述べたように，これは同じ状態を繰り返し通過することを防ぐためである．このため，迷路のような問題では，一度探索して行止りと認識すると，その後通行可能になってもそこに行くことなく，通行可能となったことに気づかないことが生じる．しかし，RTSS では，①実時間探索部に与えられる問題は比較的小規模のものであり，②実時間探索部に制御が移るとき，以前の推定値を消去して探索が始まるため，この影響は少ないといえる．

4. 簡単なモデルを用いた RTSS の性能解析

本章では RTSS の性能を簡単なモデルを用いて評価する．切換えしきい値の増減により，計画に要するコストが大きく変化すること，すなわち，適切なしきい値の設定により，計画立案実行コストを大きく削減できることを示す．

まず，RTA* によって得られる計画の長さを推定する．状態空間は木により表現され，深さ d に唯一の目標節点があるとする．さらに，任意の節点から目標節点へ向かう方向へ正しく移動する確率を p で表す．この値は推定関数の精度を表す．このとき，目標

節点への到達に要する平均ステップ数は確率論のランダムウォークに関する平均到達時間の議論から以下で表せる*1.

$$m(0, d|p)=\frac{d}{2p-1}$$

次に，副目標生成の効果を推定する．副目標生成の正の効果は実時間探索と組み合わせたときに確率 p の増加として現れる．負の効果は，適切なオペレータ選択ができないときに計画の長さが増加することである．これは，常に最短経路上に副目標を設定できないことによる．オペレータ選択は，選択候補が多い場合，問題に依存したヒューリスティクスを用いなければランダムな選択に近くなる．目標達成までの計画長さが長い場合には，副目標生成による探索の分岐数の削減などの効果の比重は小さくなり，計画の長さの増加の項が，STRIPS 全体の性能を支配するようになる．ここでは，適切なオペレータが選択ができない場合を考え，STRIPS を単純な後向き探索として評価する．すなわち，STRIPS を確率 q で目標状態の方向へ正しく移動する探索として扱う．

次に，RTSS により得られる計画の長さを推定する．問題解決過程で k 回副目標が生成されるようにしきい値が設定されるとする．このとき，元問題が副問題に等しく分割されるとすれば，深さ $(d-k)/(k+1)$ に目標節点を持つ副問題を解くことになる．また，副目標生成により推定関数の精度が向上する．RTA* と区別するため，目標節点の方向へ正しく移動する確率を p' と表記する．一方，副目標生成の負の効果のため，より多くの副問題が生成される．このため，RTSS は $k+1$ 個ではなく，$m(0, k|q)+1$ 個の副問題を解くことになる．

上の議論から，RTSS，RTA*，STRIPS の三つのアルゴリズムは以下の計画立案コストを持つ．ここで，u_R, u_S はおのおの RTA*，STRIPS により長さ 1 の計画を立案するのに必要なコストとする．

$$Cost_{RTA*}=m(0, d|p)u_R$$

$$Cost_{STRIPS}=m(0, d|q)u_S$$

$$Cost_{RTSS}=(m(0, k|q)+1)m\left(0, \frac{d-k}{k+1}\middle|p'\right)u_R + m(0, k|p)u_S$$

*1 非負の整数全体上で，時刻 n に状態 i にいて，時刻 $n+1$ に確率 p で状態 $i+1$，確率 $1-p$ で状態 $i-1$ に移動する推移確率を持つ 1 次元のランダムウォークを考える．状態 0 では，確率 $1-p$ で状態 0 に移動する．このとき，状態 i から状態 0 への平均到達時間は $m(i, 0|p)=d/(2p-1)$ で表される．非負の整数上で負の方向への移動を目標節点の方向への移動と考えると上式が導ける．

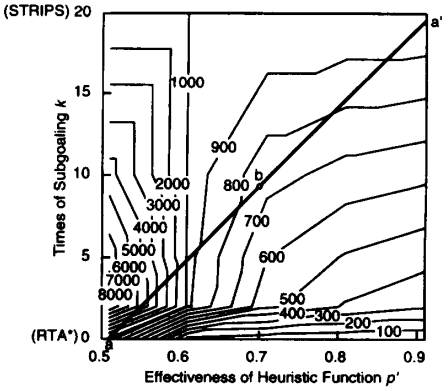

図 3 簡単なモデルの解析により得られた計画立案コストの等高線図 ($d=20$, $q=0.55$, $u_R=1$, $u_S=5$)

RTSS のコスト式は右辺第 1 項の実時間探索のコストと，第 2 項の副目標生成のコストの和で表される．

u_R, u_S を長さ 1 の計画実行に要するコストとすれば，各式は計画実行コストを表す．

図 3 は p', k が与えられたときの，RTSS の計画立案コストを示す．ここでは，$d=20$ としてある．計画立案コストは $k=0$ のとき，RTA* のコストに等しくなり，$k=20$ のとき，STRIPS のコストに等しくなる．p' と k の値は相関があり，その変化の仕方は問題領域により異なる．一般には，副目標生成 k が増えれば，副問題がより単純となり推定関数の精度 p' が大きくなるので，例えば，p', k が $a-a'$ に沿って変化するとしよう．このとき，k の増加につれ，計画立案コストは最初増加し，次に減少し，また増加に転じる．コストは点 b で最小値をとる．この結果から，RTSS はしきい値の適切な調整により，RTA*，STRIPS よりも，少ないコストで目標達成が可能であることがわかる．

5. 目標の抽象度に基づく目標分解基準と副目標選択基準

RTSS を実際の問題に適用するには，目標分解基準と副目標選択基準の計算が必要である．本論文では目標分解基準と副目標選択基準を，ABSTRIPS [Sacerdoti 73]で提案された目標の抽象度に基づいて以下のように定義する．ABSTRIPS は副目標に重要度(抽象度)を設定し，副目標間に階層関係を導入する．そして，重要度の高い部分から計画を立案して先に大枠を決め，重要度の低い部分，つまり詳細部分を後回しにする．これにより，むだな計画立案を削減するものである．

5・1 目標分解基準

当面の目標状態と現状態の差として得られるリテラル(副目標)のなかに抽象度の高い副目標が含まれない場合を考えよう．この場合，すでに抽象度の高い副目標は達成されており，未達成部分は単純な問題で，推定関数による状態の選別がうまく機能すると考えられる．すなわち，未達成の副目標は実時間探索を用いて比較的容易に達成されると考えられる．

そこで，当面の目標 SG に対する目標分解基準値を次式のように，未達成目標の抽象度の最大値で表す．

$$\text{目標分解基準値}(C, SG)=\max_{d_i} f(d_i)$$

ここで，$D=\{d_i\}$ は $SG-C$(当面の目標状態 SG と現状態 C の差)で，$f(x)$ はリテラル x の抽象度を表す．$f(x)$ は抽象的事柄を表すリテラルには大きな値を返し，詳細な事柄を表すリテラルには小さな値を返す．

目標分解基準は推定関数が精度の良い値を返すかどうかの判断基準であるため，この基準に推定関数自体は使えない．

5・2 副目標選択基準

STRIPS の副目標選択基準として，いくつか考えられる．その一つは以下で表せる．これは，オペレータにより達成される副目標中のリテラル数と削除される目標中のリテラル数を重みづけした差である．

$$\text{副目標選択基準値}(Operator)=w\sum_{y_i}1-(1-w)\sum_{z_j}1$$

ここで，D は $SG-C$(当面の目標状態 SG と現状態 C の差)，$y_i\in\{D\cap\text{add-list}(Operator)\}$，$z_j\in\{G\cap\text{delete-list}(Operator)\}$，$G$ は最終の目標，w $(0\leq w\leq 1)$ は重み係数である．

上式では抽象度によるリテラルの区別がない．ここでは，抽象度の高い副目標の達成を重視する以下の式を考える．

$$\text{副目標選択基準値}(Operator)=w\sum_{y_i}f(y_i)-(1-w)\sum_{z_j}f(z_j)$$

ここで，$f(x)$ はリテラル x の抽象度を表す．

この式を用いるねらいは，右辺第1項で，抽象度の高い副目標を追加するオペレータを優先して選ぶことで，探索空間の絞り込みを可能とし，実時間探索に制御を移せる副目標を早く生成することである．また，右辺第2項で，目標中の抽象度の高いリテラルを削除するオペレータ選択を避けることで，副目標生成回数を減少させている．

6. 実験による RTSS の評価

ロボットの作業計画を例題として，まず，RTSS，RTA*，STRIPS の性能比較を行い，次に，副目標系列の変更機能と動的環境との関係を調べる．最後に，**4章**で得たモデルによる理論値と実験値との比較を行う．

6・1 評 価 問 題

・問題設定

課題は1台のロボットが図4に示す空間内で2個の箱をそれぞれの目標位置に運ぶことである．空間全体は3×3の9部屋からなり，各部屋は5×5の25の格子からなる．部屋は開閉できる扉でつながれている．ロボットは，何も持たないか，あるいは箱を一つだけ持って，前後左右の隣接位置に動ける．また，扉の開閉ができる．ロボットは作業空間の構造と箱の正確な位置を知っているとする．

・世界の状態表現

世界の状態を以下の六つの述語を用いて表す．

(**box-in** *box room*), (**robot-in** *room*), (**box-at** *box location*), (**robot-at** *location*), (**open** *door*), (**close** *door*).

これらを抽象度の観点から二つに分ける．

抽象度が高い box-in, robot-in

抽象度が低い box-at, robot-at, open, close

おのおの抽象度は2，1とする．box-in の抽象度が高いのは，その1状態に対して25状態の box-at が含まれるからである．このとき，**3・2節**で示した RTSS アルゴリズムにおいて，しきい値 t を1未満とすれば，RTSS は STRIPS と同じ動作を，しきい値 t を2以上とすれば，RTSS は

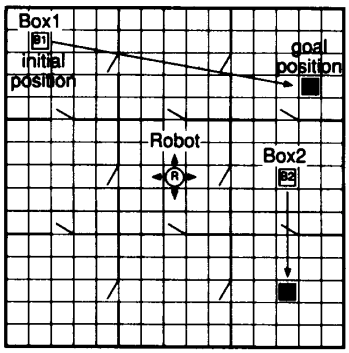

図 4 ロボット作業空間

RTA* と同じ動作をする*2．本問題において，副目標生成と実時間探索が組み合わせて実行されるのは，しきい値 $t=1$ の場合だけである．しきい値として複数の値がとれる場合，どの値に設定するか，つまり，どの程度副目標生成を行うかは，試行錯誤的に決定される．

・オペレータ

本実験では，goto-room-location, push-box, go-through-door, push-through-door, open-door, close-door の六つのオペレータを用いる．

・推定関数

実時間探索で用いる目標までの距離の推定関数は以下の二つの Manhattan 距離の和である．

1. おのおのの箱の現在位置と目標位置間の Manhattan 距離の和
2. ロボットの現在位置とロボットに対して最も近くにある目標位置にない箱の位置間の Manhattan 距離

・評価方法

-RTSS, RTA*, STRIPS の比較

ここでは，副目標生成と実時間探索の組合せの効果を見るため，RTSS とその両極にある STRIPS，RTA* を比較する．3 アルゴリズムの性能を計画立案時間と実行時間の両面から評価する．計画立案時間は解を得るのに要した CPU 時間，実行時間は目標達成に要した動作系列の長さとする．1 回の試行につき CPU 時間が 1 200 [s] を超えても目標状態に到達できない場合は，解が求まらないとする．各アルゴリズムについて 30 問ずつ解く．ロボットと箱の初期位置と目標位置は乱数を用いて設定する．扉はすべて初期状態では閉じているとする．また，環境は静的とする．

-副目標系列の変更機能と動的環境の関係

環境については静的と動的，副目標系列については固定と可変，という組合せで合計 4 通りの場合について問題を与え，実験を行う．評価は目標達成までの動作系列の長さで行う．各場合について，10 問ずつ与え，各問題を 30 回解いた結果の平均値で評価する．ここで，動的環境について説明する．各部屋は扉のついた通路により結ばれている．この通路が時間の経過に従い，通行不能になったり，通行可能になる．時間はロボットの一つの動作実行を 1 単位時間と計る．通路の状態変化は乱数により決定される．ただし，ある地点から任意の地点に到達不可能になる変化は起こらない．各通路の変化の程度は，平均して 30 単位時間通行可能の後，16 単位時間通行不能という繰返しであった．ロボットは自分のいる部屋の内部(通路を含めて)の環境観測は正確にできるとする．

6・2 RTSS, RTA*, STRIPS の比較

図 5 に RTA*, STRIPS と本論文で提案した RTSS(しきい値 $t=1$)の比較を示す．図の横軸は問題の番号を示し，縦軸は計画立案時間と実行時間を示す．各点が一つの問題を解いた結果を示す．

この結果から以下のことがいえる．

・RTA* と STRIPS は計画の品質が大きくばらつくのに対し，RTSS は安定して高速に目標達成が行える．RTA* は多くの場合，時間切れで目標達成に失敗している．

6・1 節で述べた推定関数を用いて目標全体を一度に考えると，隣接状態に対し，目標までの距離の真値の大小関係と推定値の大小関係が逆転する場合が多く見られる．また，この推定値は減少するが真値は増加する隣接状態が複数存在するため，その推定値の修正に時間を要する．このため，箱 B1 と箱 B2 の間も何度も往復する現象が見られ，RTA* は多くの問題で目標達成を失敗する．

RTSS は RTA* と STRIPS を有効な領域でのみ使用するため，安定した問題解決が可能となる．

・副目標生成は実時間探索より計画立案コストが大きい．

STRIPS の結果を RTSS の結果と比較すると，実行時間の比(平均 2.6 倍)に対し，計画立案時間

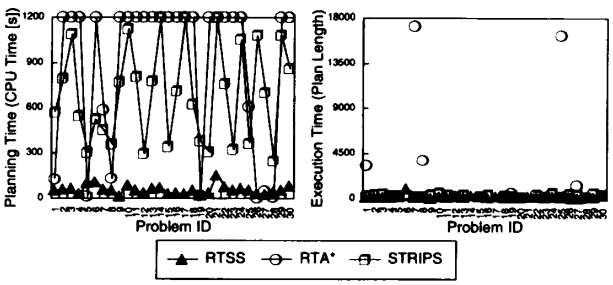

図 5 RTA*, STRIPS, RTSS の性能比較

*2 RTSS アルゴリズムを見るとわかるように，しきい値 $t<1$ としても，副目標選択部でのオペレータ選択時に副目標選択基準が用いられる．これは，抽象度の高いリテラルを優先的に副目標として選択し，計画を立案することを意味する．そのため，本実験でいう STRIPS は Fikes らの提案した標準の STRIPS とは異なり，ABSTRIPS の階層的問題解決の要素を取り入れたものとなっている．

の比(平均15.7倍)の大きさがが目立つ．オペレータ具体化時に，実時間探索では前提条件が現状態と厳密に一致するという条件でオペレータを具体化する．一方，副目標生成では，目標状態と現状態の差を追加するという条件で具体的する．このため，未束縛変数への多くの具体化が考えられ，選択候補が多くなり，その選択操作などに時間を要する．よって，副目標生成のみで問題を解くSTRIPSは計画立案コストが大きくなる．

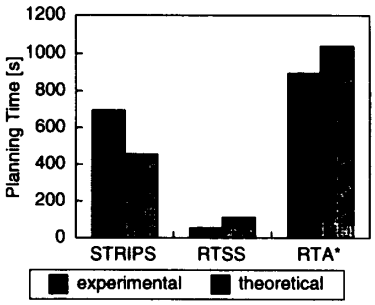

図 7 計画立案コストの理論値と実験値の比較

6・3 副目標系列の変更機能と動的環境との関係

副目標変更時の現状態から遷移可能な状態を計算するときの深さ限界を$l=2$と設定した．実験結果を図6に示す．横軸は問題の番号を，縦軸は目標達成に要した動作系列の長さを表す．この結果から以下のことがいえる．

- 動的環境では静的環境より計画長さが長くなるため，相対的に副目標変更による計画長さ短縮効果が大きくなる．
  副目標可変の場合，固定の場合に比べて2割程度(静的環境では平均して16%，動的環境では20%)計画長さが短縮される．動的環境で副目標変更がより効果的となるのは，動的環境での計画長さが静的環境の2〜3倍になるためである．ただし，問題により効果にばらつきがある．この点は今後の検討事項である．

計画立案時間に関して，副目標可変の場合，固定の場合に比べ，静的環境では微増(+7%)し，動的環境では減少(−26%)した．静的環境では副目標変更に伴う探索のオーバヘッドの影響が現れている．一方，動的環境の結果は，そのオーバヘッドが計画長さの短縮による計画立案時間の減少に相殺されたことを示している．

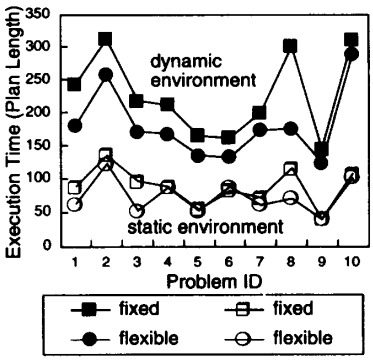

図 6 静的，動的環境，副目標固定，可変の各場合におけるRTSSの性能比較

6・4 モデルによる理論値と実験値の比較

図7は**4章**で求めた，三つのアルゴリズムに対するモデルを用いた計画立案コストの理論値と実験値との比較図である．モデルに現れるパラメータ値は**6・2節**の実験結果から推定した．制限時間超過で目標達成に失敗した場合(RTA*では，30回中21回)を，この推定では，制限時間経過後すぐ目標へ至る計画が求まるとして扱った．

- STRIPSでは実験値が理論値より大きい．
  ①目標へ正しく移動する確率qは，問題解決過程で変化する(本例題では，高抽象レベルのq＞低抽象レベルのq)．②理論値$Cost_{STRIPS}$はqに対して非線形である．③パラメータ推定では高低両抽象レベルを平均化した値を推定した．これらの理由から，低抽象レベルでのコストが過小に評価されたと考えられる．
- RTSSでは実験値が理論値よりも小さい．
  RTSSは高抽象レベルでのみ副目標生成するため，STRIPSの結果から推定したq値は実際値よりも小さい．その結果，副目標生成部分のコストが過大評価されたと考えられる．
- RTA*では実験値が理論値よりも小さい．
  モデルでは，状態空間を木表現し，深さdに唯一の目標節点があると仮定した．しかし，実際には，目標節点は複数存在する．このため，理論値が実験値より大きく評価されたと考えられる．

7. むすび

本論文では副目標生成機構を実時間探索に組み入れた実時間プラニングアルゴリズムReal-Time Search with Subgoaling(RTSS)を提案した．RTSSは実時間問題解決におけるRTA*とSTRIPSの盲目的な探索に陥るという問題点の解消に成功した．RTSSでは二つの基準，目標分解基準と副目標選択基準を導入した．目標分解基準は目標の難易度，すなわち，

RTA*における状態の推定関数が有効に働くかどうかを表す．これを副目標生成から実時間探索への切換え時の判断に用いた．また，副目標選択基準はより早く実時間探索に切り換わる副目標を選択する基準で，その使用は実時間性の向上に寄与する．さらに，動的な環境での性能向上を目的として，生成済みの副目標系列の柔軟な変更方法を提案した．副目標生成から実時間探索へうまく切り換えることにより，計画立案コストを大きく削減できることを簡単なモデルを用いて示した．また，ロボットの作業計画を例題としてシミュレーションを行い，RTSSがRTA*やSTRIPSよりも安定して高速に問題を解決できることを示した．

今後の課題は副目標生成から実時間探索への切換えしきい値の設定基準を与えることである．現在は，性能が向上するよう試行錯誤的にしきい値を与えている．これを自動化できれば，RTSSはさらに有用なものとなる．

謝　辞

本研究を支援いただいたNTTコミュニケーション科学研究所松田晃一所長，大里延康主幹研究員，討論いただいた横尾　真主任研究員に感謝します．

◇参考文献◇

[Fikes 71] Fikes, R. E. and Nilsson, N. J.: STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving, *Artif. Intell.*, Vol. 2, No. 3-4, pp. 189-208 (1971).

[Korf 87] Korf, R. E.; Planning as Search: A Quantitative Approach, *Artif. Intell.*, Vol. 33, No. 1, pp. 65-88 (1987).

[Korf 90] Korf, R. E.: Real Time Heuristic Search, *Artif. Intell.*, Vol. 42, No. 2-3, pp. 189-211 (1990).

[McDermott 92] McDermott, D.: Transformational Planning of Reactive Behavior, Yale University, Department of Computer Science, Technical Report YALEU/CSD/RR#941 (1992).

[松原 93] 松原繁夫，石田　亨：副目標生成機能を備えた実時間プランニング，人工知能学会全国大会(第7回)論文集，pp. 343-346(1993).

[Matsubara 94] Matsubara, S. and Ishida, T.: Real Time Planning by Interleaving Real-Time Search with Subgoaling, *2nd Int. Conf. on AI Planning Systems (AIPS-94)*, pp. 122-127 (1994).

[松原 94] 松原繁夫：柔軟な副目標変更機能を備えた実時間プランニング，人工知能学会全国大会(第8回)論文集，pp. 313-316 (1994).

[Sacerdoti 73] Sacerdoti, E. D.: Planning in a Hierarchy of Abstraction Space, *IJCAI-73*, pp. 412-422 (1973).

[Sacerdoti 75] Sacerdoti, E. D.: The Nonlinear Nature of Plans, *IJCAI-75*, pp. 206-214 (1975).

[Schoppers 87] Schoppers, M. J.: Universal Plans for Reactive Robots in Unpredictable Environments, *IJCAI-87*, pp. 1039-1046 (1987).

〔相当編集委員・査読者：石塚　満〕

著者紹介

松原　繁夫(正会員)

1990年京都大学工学部精密工学科卒業．1992年同大学院工学研究科修士課程修了．同年，NTTに入社．現在，NTTコミュニケーション科学研究所勤務．人工知能の研究に従事．

石田　亨(正会員)

1976年京都大学工学部情報工学科卒業．1978年同大学院修士課程修了．同年，日本電信電話公社電気通信研究所入所．横須賀研究所においてソフトウェア工学，知識処理などの研究開発に従事．1983年から84年にかけて，米国コロンビア大学計算機科学客員研究員．現在，京都大学大学院工学研究科情報工学専攻教授．工学博士．問題解決，分散人工知能，コミュニケーションに興味を持つ．1992年度人工知能学会論文賞，人工知能学会設立10周年記念優秀論文賞受賞．