特集論文

Technical Papers

情報保存的な問題領域におけるトレースからの問題分割戦略の獲得

A Method for Acquiring Problem Decomposition Strategy from Traces

小林　聡*1　堀　浩一*2　大須賀　節雄*2
Satoshi Kobayashi　Kohichi Hori　Setsuo Ohsuga

*1 東京大学工学部
Faculty of Engineering, The University of Tokyo, Tokyo 153, Japan.
*2 東京大学先端科学技術研究センター
Research Center for Advanced Science and Technology, The University of Tokyo, Tokyo 153, Japan.

1992年3月30日　受理

Keywords: macro operator, macro table, problem decomposition, information preserving, explanation based learning.

Summary

Search is one of the most important and basical technique underlying most computer problem solving methods. Exhaustive search method explore all possible paths to a goal state. However, this method requires exponentially large time and space for computation as state spaces become large. Problem decomposition is one of the effective methods to reduce search space.

General Problem Solver implements means-ends analysis. The necessary condition for its applicability is the existence of a set of subgoals and an ordering among them, such that once satisfying a subgoal, it must not be violated in order to satisfy the remaining subgoals. But problems, like Rubik's Cube, whose subgoals have strong interaction among one another, do not satisfy this condition.

Korf has developed Macro Problem Solver and used macro operators to overcome this difficulty. His system solves a problem based on a macro table without no search. A macro table is a table of macro operators whose column headings are state components (differences in GPS) and whose row headings are the values of state components. Korf proposes a method to learn macro tables. His idea is basically a backward search from a goal by applying the inverse of the primitive operators. But in this paper, we consider the problem of acquiring a macro table from traces.

We first define the extended definition of Korf's Macro Table, called Problem Decomposition Strategy. Because the acquisition of it seems to be intractable in general, we limit the attention to the domains which are information preserving and whose goal state space is composed of only one state. After investigating the characteristics of such domains, we present the efficient method for acquiring Problem Decomposition Strategy from traces. Our method does not need the information of subgoal ordering, which is necessary in advance for applying Korf's method. This is one of the most important contributions of this research.

1. はじめに

問題解決システムが解の探索において全解探索を行うのは通常困難であり，探索空間を削減するための理論技術の開発が重要であると認識されている[1][2]．説明に基づく学習[3]は，例題の説明の一般化を利用してシステムの効率化を行うための一般的な枠組みであり，多くの研究が行われている．本論文は，複数例題を利用して説明に基づく学習により有用なマクロを得る問題に対し，問題分割戦略という視点に絞って，一つの理論的な解析を行ったものである．

本稿では，ある定義に基づく状態空間において，ゴール状態に至るまでのオペレータ列を求める問題を考

える.Korf[2]はマクロオペレータを導入することにより，サブゴール間の干渉が非常に強い問題に対しても適用可能な，問題分割手法を提案している．Korfの手法では，解決するサブゴールの順序を与えなければならない．これに対し，本稿では，ある未知の問題分割戦略によって解かれた解の軌跡(トレース)の情報から，その問題分割戦略を同定する問題を考える．したがって，サブゴールの解決順序を明示的に与える必要はない．

同様の研究は，Tadepalli[4][5]，山村[6]らが行っているが，これらの研究との比較は，問題設定を行った後，**2・3節**で述べる．

まず，**2章**で，問題分割戦略を定義する．そして，正例のみから問題分割戦略を獲得することを考える．そして，**3章**では，情報保存的な問題領域に対する多項式時間の問題分割戦略獲得アルゴリズムを示す．

2．問題分割戦略学習

2・1 問題分割戦略

【定義1】 問題とは，初期状態空間I，ゴール状態空間G，オペレータ集合$O=\{o_1, o_2, \cdots, o_k\}$の3つ組$<I, G, O>$である．問題例とは，初期状態空間の中の一つの状態$s$であり，その解ans$(s)$とは，$o_n \cdot o_{n-1} \cdots o_1(s) \in G$なるオペレータ列$o_n \cdot o_{n-1} \cdots o_1$である．また，$s \xrightarrow{o_1} o_1(s) \xrightarrow{o_2} o_2 \cdot o_1(s) \xrightarrow{o_3} \cdots \xrightarrow{o_n} o_n \cdot o_{n-1} \cdots o_1(s)$を解ans$(s)$のトレースという．このとき，解ans$(s)$をこのトレースのオペレータ適用列と呼ぶことにする．

ここで，二つ以上のオペレータを組み合わせてできるオペレータを，マクロオペレータ（あるいは，単にマクロ）と呼ぶ．また，オペレータを全く適用しないことをϵで表す．

【定義2】 問題$P=<I, G, O>$の問題分割戦略(Problem Decomposition Strategy)とは，次の条件を満足する部分問題$P_i=<I_i, G_i, M_i>$の列，$\overline{P_i}=P_1, \cdots, P_k$である．

(1) $I_1=I, G_k=G$

(2) $I_{i+1}=G_i$ for $i=1, \cdots, k-1$

(3) $I_i \not\subseteq G_i$ for $i=1, \cdots, k$

(4) 各M_iはオペレータ，マクロオペレータまたはϵを要素とする集合である．

【定義3】 問題$P=<I, G, O>$，その問題分割戦略$\overline{P_i}=<\overline{I_i, G_i, M_i}>(1 \leq i \leq k)$，解ans$(s)$のトレース$t=s \xrightarrow{o_1} o_1(s) \xrightarrow{o_2} o_2 \cdot o_1(s) \xrightarrow{o_3} \cdots \xrightarrow{o_n} o_n \cdot o_{n-1} \cdots o_1(s)$が与えられたとき，次の条件を満たすマクロオペレータの列$m_k, m_{k-1}, \cdots, m_1$を，トレース$t$の問題分割戦略$\overline{P_i}$による生成という．

(1) $m_i \in M_i$ for $i=1, \cdots, k$

(2) $m_1=o_{d_1} \cdots \cdot o_1$, $m_2=o_{d_2} \cdots \cdot o_{d_1+1}, \cdots$, $m_k=o_{d_k} \cdots \cdot o_{d_{k-1}+1}$
となるd_i $(i=1, \cdots, k)$が存在する．ただし，ここで$d_k=n$

(3) $s \in I, m_i \cdots \cdot m_1(s) \in G_i$ for $i=1, \cdots, k$

(4) すべてのjについて，$o_p \cdots \cdot o_1(s) \in G_j$なる$d_j$より小さい$p$が存在しない

$o_p \cdots \cdot o_1(s) \in G_j$なる$d_j$より小さい$p$が存在するとき，部分問題$P_j$でオペレータのむだな適用があるという．

【定義4】 問題分割戦略$\overline{P_i}=<\overline{I_i, G_i, M_i}>$ $(i=1, \cdots, n)$の各M_iのj番目の要素を$m_{i,j}$，M_iの要素の個数をk_iと表すと，正則表現

$$(m_{n,1}+m_{n,2}+\cdots+m_{n,k_n}) \cdot (m_{n-1,1}+m_{n-1,2}+\cdots+m_{n-1,k_{n-1}}) \cdots (m_{1,1}+m_{1,2}+\cdots+m_{1,k_1})$$

が表すオペレータの適用列の集合を$\overline{P_i}$の解集合と呼ぶ．また，$\overline{P_i}$が生成可能なトレースのオペレータ適用列すべてからなる集合と，$\overline{P_i}$の解集合が等しいとき，$\overline{P_i}$は完全であるという．

解集合は，各部分問題から任意にマクロを取り出して作ることのできるすべてのオペレータ列の集合である．

【例1】 Fig. 1(a)のような問題を考える．これは，二つの部屋rm 1, rm 2があり，そこに置かれている箱bをロボットrがgの位置まで運ぶ問題である．オペレータは，七つからなる．このとき，問題例{in(r, rm 2), in(b, rm 2), open(d)}に対する解は，7・4・6・3であり，そのトレースはFig. 1(d)に示してある．また，Fig. 1(b)(c)には，問題分割戦略PDS 1，PDS 2が示されている(ただし初期状態空間は省略)が，(d)のトレースのPDS 2による生成は，74，63となる．

PDS 1の解集合は74・(6321+6763+63251)となる．PDS 1において，部分問題subP 1で6763を選択し，部分問題subP 2で74を選択してできるトレースは，Fig. 1(e)(ただし状態表示省略)となるが，図に太線で示す部分にオペレータのむだな適用があるので，生成可能ではない．したがって，PDS 1は完全な問題分割戦略ではない．一方，PDS 2は完全な問題分割戦略である．このように，できるだけ短いオペレータ適用列を探すような問題領域では，定義3の(4)の条件は自然である．

本稿では完全な問題分割戦略のみを対象とする．したがって，以後，完全な問題分割戦略を単に問題分割戦略と呼ぶことにする．

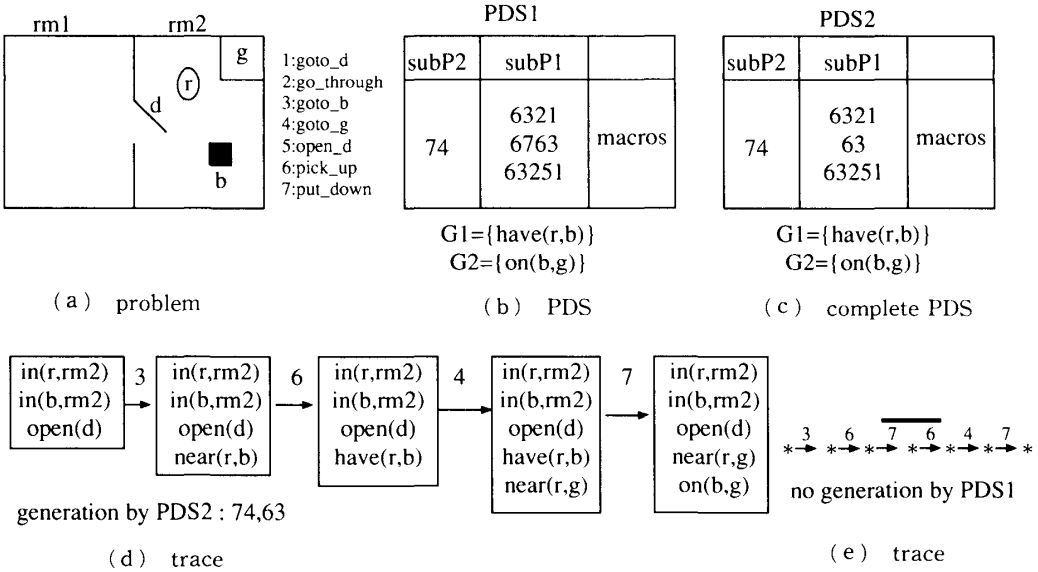


Fig.1 Problem and problem decomposition strategy.

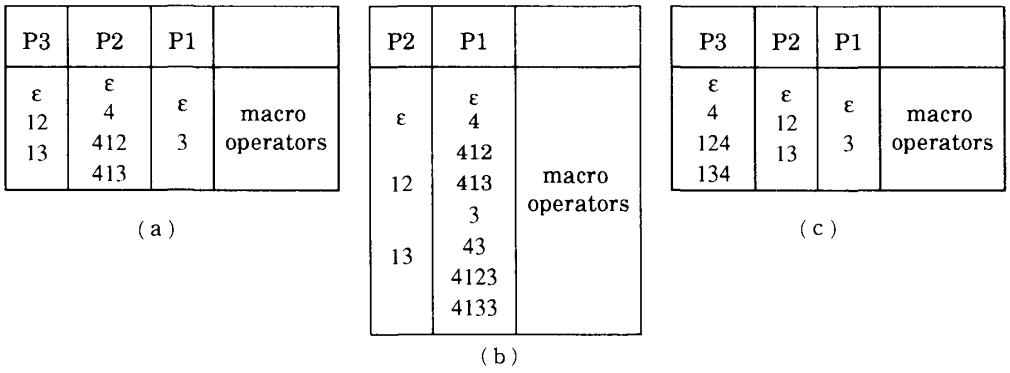


Fig.2 Problem decomposition strategy and its unfolding.

2・2 極小問題分割戦略

問題分割戦略 $\overline{P_i}=\langle\overline{I_i, G_i, M_i}\rangle$ $(i=1,\cdots, n)$ で連続する部分問題 P_j, P_{j+1} を一つの部分問題に合成することを考える．M_j に含まれるマクロと，M_{j+1} に含まれるマクロのすべてを組み合わせることによってできるマクロの集合を，M_j' とすると，新しくできる問題分割戦略 $\overline{P_i'}$ $(i=1,\cdots, n-1)$ は，次のようになる．

$P_i'=P_i$ for $i=1,\cdots, j-1$

$P_i'=P_{i+1}$ for $i=j+1,\cdots, n-1$

$P_j'=\langle I_j, G_{j+1}, M_j'\rangle$

$\overline{P_i'}$ と $\overline{P_i}$ が，生成することのできるトレース集合が等しいとき，$\overline{P_i'}$ は $\overline{P_i}$ から展開 (unfolding) 可能であるという．これを，$\overline{P_i}\leq\overline{P_i'}$ と表す．

【例2】 Fig.2(a)の部分問題1と部分問題2を展開すると，Fig.2(b)のような問題分割戦略になる．ただし，ここでは，各部分問題の初期状態空間，ゴール状態空間の記述を省略してある．また，この図でのオペレータ番号は，Fig.1のものとは無関係である．

問題分割戦略のサイズを考えれば，$\leq$ の意味で極小な戦略が好ましい．ある問題分割戦略 $\overline{P_i}$ の $\leq$ の意味で極小な問題分割戦略を極小問題分割戦略と呼ぶ．

一般に，この極小問題分割戦略を正例から効率良く獲得できるかどうかは未解決である．したがって次章以降では，対象とする問題領域を限定することにより，効率の良い手法を与える．

2・3 他研究との比較

Tadepalli[4][5]は，トレースにおける状態を見ながら，Korf[2]のマクロ表を獲得する，Serial Parsing, Batch Parsingという手法を開発している．しかし，そこでは問題 $P=\langle I, G, O\rangle$ の未知の問題分割戦略 $\overline{P_i}=\langle\overline{I_i, G_i, M_i}\rangle$ $(i=1,\cdots, n)$ において，各 G_i が G のサブゴールの連言からなることが仮定されている．これは，ロボット計画立案のような問題を考えるとかなり強い制約である．本研究ではこのような制約はない．一方，Tadepalliの手法では，問題分割戦略に完全という制約がないので，その点では本手法のほうが劣っている．

山村らの研究[6]では，EBL[3]の枠組みを複数例題下に拡張した拡張EBL[7]のもとで，マクロ表を所属性質問[8]を用いて同定する手法を提案している．しかし，彼らの問題分割戦略では各部分問題において，$G_{i+1}\subseteq G_i$が成り立つこと，および，所属性質問を利用するために各M_iに空オペレータが含まれていることが仮定されている．これに対し本稿で提案する手法ではそのような制約はない．

3. 情報保存的な問題領域に対する問題分割アルゴリズム

3・1 対象とする問題領域

本稿で対象とする問題領域を次のような条件が成り立つ領域に限定する．

条件(a) 問題領域が情報保存的(information preserving)[2]である．

条件(b) ゴール状態空間がただ一つの状態からなる．

ただしここで，情報保存的とは次のように定義される．

【定義5】[2] 問題$P=<I, G, O>$が与えられたとき，すべてのオペレータo_i，およびすべての状態のペアs1 s2に対して次が成り立つとき，問題Pは情報保存的であるという．

s1 s2にo_iが適用可能であり，$o_i\ (s1)=o_i\ (s2)$ならば

$$s1=s2$$

3・2 情報保存的な問題領域のトレース集合の特徴

〈補題1〉[9] 条件(a)(b)を満たす問題Pとその問題分割戦略$\overline{P_i}$が与えられたとき，$\overline{P_i}$が生成可能なトレースの数は有限であり，その数は，$\overline{P_i}$の解集合の要素数に等しい．

《証明》 ゴール状態空間がただ一つからなり，情報保存的であるので，同じオペレータ適用列を持つ異なるトレースは存在しない．したがって，題意は満たされる． □

〈補題2〉 条件(a)(b)を満たす問題P，問題分割戦略$\overline{P_i}=\langle\overline{I_i, G_i, M_i}\rangle\ (i=1,\cdots, k)$，およびその生成可能なトレースすべての集合$T$が与えられたとき，そのトレースの生成の仕方はただひと通りに定まる．

《証明》 一つのトレースを2通りに生成できたとして矛盾を導く．二つの生成ではどこかの部分問題でマクロの適用が異なる．1番目の部分問題から順に見ていって，最初にマクロの適用が異なる部分問題をP_jとして，どちらか一方の生成が，P_jにおいてむだなオペレータの適用を含むことを利用する．詳細は文献(9)を参照． □

あるオペレータ適用列$m=o_n\cdot o_{n-1}\cdot\cdots\cdot o_1$が与えられたとき，$p\cdot q=m$となるオペレータ適用列$q$が存在するようなオペレータ適用列$p$を$m$のPrefixと呼ぶ．

【定義6】 条件(a)(b)を満たす問題$P=<I, G, O>$，その問題分割戦略$\overline{P_i}$，およびその生成可能なトレースすべての集合Tが与えられたとき，次のように構成される根付き木$\mathrm{PTree}=(V\ E)$をTのPrefix Treeと呼ぶ．

(1) $V=\{s|s$はあるトレース$t\in T$のオペレータ適用列のPrefixである$\}$

(2) $E=\{(i, j)|j=i\cdot o_p, o_p\in O, j$はあるトレース$t\in T$のオペレータ適用列のPrefixである$\}$

(3) 上記で，各頂点$o_k\cdot o_{k-1}\cdots o_i$には$g$をゴール状態として，状態$o_1^{-1}\cdot\cdots\cdot o_{k-1}^{-1}\cdot o_k^{-1}(g)$をラベルづける．これを頂点$o_k\cdot o_{k-1}\cdots o_i$の状態と呼び，$st(o_k\cdot o_{k-1}\cdots o_i)$と表す．

(4) 辺$(o_k\cdot o_{k-1}\cdots o_p, o_k\cdot o_{k-1}\cdots o_p\cdot o_{p-1})$には，オペレータ$o_{p-1}^{-1}$をラベルづける．これを，辺の適用オペレータと呼ぶ．

さらに，トレースの初期状態に相当する頂点にFig. 3(a)に示すδコンポーネントをつけるものとする．

【例3】 Fig. 2(a)の戦略のトレースすべてが与えられたとき，そのPrefix TreeはFig. 3(b)のようになる．ただし，辺のラベルはオペレータの番号で表し，-1も省略してある．

【定義7】 条件(a)および(b)を満たす問題P，その極小問題分割戦略$\overline{P_i}=\langle\overline{I_i, G_i, M_i}\rangle\ (i=1,\cdots, n)$，その生成可能なトレースすべての集合$T$，およびPrefix Treeが与えられたとき，

$$\{m_n\, m_{n-1}\cdots m_i|m_j\in M_j\quad(\text{for } j=i,\cdots, n)\}$$

で表されるPrefix Treeの頂点の集合を，第iレベルの分割頂点(Division Node)の集合と呼ぶ．

【例4】 Fig. 3(b)に，Fig. 2(a)の極小問題分割戦略における第2レベルの分割頂点を□で，第3レベルの分割頂点を○で示す．ただし，第3レベルの分割頂点はすべて第2レベルの分割頂点にもなっている．

Prefix Tree上ですべてのレベルの分割頂点が求められれば，それから極小問題分割戦略を求めることは容易である．したがって，分割頂点がPrefix Tree上で示す特徴を分析する．

〈補題3〉 条件(a)(b)を満たす問題P，そのトレース有限個の集合T，およびそのPrefix Treeが与えられたとき，Prefix Treeの頂点集合$\{D_j\}$が，Tを生

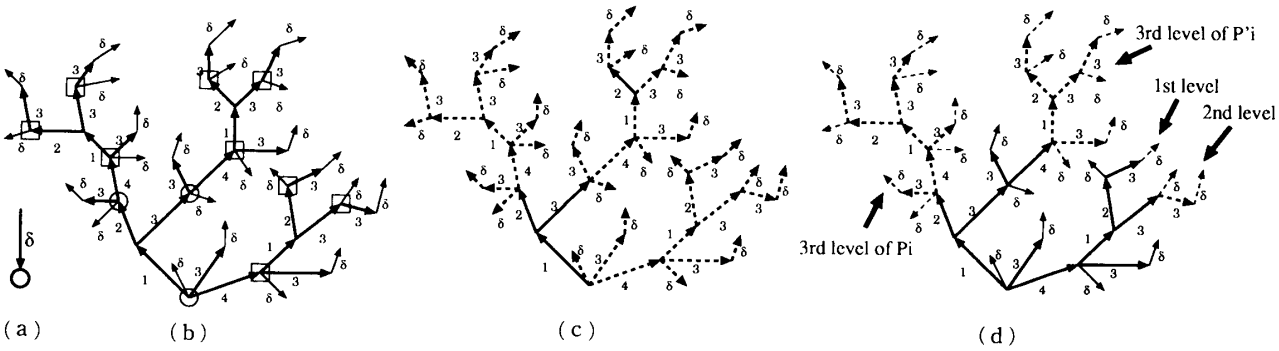


Fig. 3 Prefix tree and its common tree.

成可能なトレース集合とするある極小問題分割戦略 $\overline{P_i}=\langle\overline{I_i, G_i, M_i}\rangle$ $(i=1,\cdots, n)$ において，同じレベルの分割頂点であるための必要十分条件は，各頂点 D_j を根とする Prefix Tree の部分木の集合 $\{Td_j\}$ で次の条件を満たすものが存在することである．

(1) 各 Td_j は，互いに，頂点の状態の対応を無視すれば同型である．

(2) 各 Td_j は，もとの木 Prefix Tree において互いに頂点を共有しない．

(3) Prefix Tree のすべての葉が Td_j のどれかに含まれている．

(4) Td_i の根の状態をすべて集めた集合を Rstate とすると，各 Td_j の根以外の状態は，Rstate に含まれない．

《証明の概略》

必要性の証明　厳密には，分割頂点のレベルに関する帰納法だが，概略を示す．$\overline{P_i}$ が与えられたとき，$\overline{P_i}$ は完全であるからレベル i の分割頂点からは，部分問題 P_1 から P_{i-1} までを解いたトレースが逆向きに出ているはずである．各分割頂点に対してこれらのトレースで構成される木を $\{Td_j\}$ とすれば，これらの木が(1)(3)の条件を満たすことは明らかである．次にこの $\{Td_j\}$ が(2)を満たすことを示す．二つの木，$Td_1\,Td_2$ が頂点を共有すると仮定する．Prefix Tree は木であるから，どちらか一方の根がもう一方の根の先祖となっている．Td_1 の根 (D_1) が Td_2 の根 (D_2) の先祖となっているものとする．このとき Td_1 には

$$D_1 \overset{o_1^{-1}}{\rightarrow}\cdots\rightarrow D_2 \overset{o_{p+1}^{-1}}{\rightarrow}\cdots\rightarrow Z$$

なる葉 Z に至るパスが存在する．D_1 と D_2 はレベル i の分割頂点であるから，$s1, s2$ を頂点 D_1, D_2 が表す状態とすると，部分問題 P_{i-1} において $G_{i-1}\supseteq\{s1\}\cup\{s2\}$ が成り立つ．ここで，D_1, D_2 を通るトレース

$$\cdots\overset{o_t}{\leftarrow}st(D_1)\overset{o_1}{\leftarrow}\cdots\overset{o_p}{\leftarrow}st(D_2)\overset{o_{p+1}}{\leftarrow}\cdots\overset{o_q}{\leftarrow}st(Z)$$

を考えると，補題 2 により，この生成の仕方は一通りであるから，

$$st(D_1)\overset{o_1}{\leftarrow}\cdots\overset{o_p}{\leftarrow}st(D_2)\overset{o_{p+1}}{\leftarrow}\cdots\overset{o_q}{\leftarrow}st(Z)$$

に至るまでが，部分問題 P_1 から部分問題 P_{i-1} まで解いたトレースに相当する．しかし，このトレースの途中に $st(D_2)\in G_{i-1}$ なる D_2 が存在する．これは，T にむだな適用をしたトレースがないことに矛盾する．したがって，(2)も成り立つ．

(4)も(2)の場合と同様にして証明できる．

十分性の証明　交献(3)を参照．　□

【定義 8】 この補題 3 の(1)(2)(3)の条件を満たす木集合の個々の要素を共有木 (Common Tree) と呼ぶことにする．また，その共有木の根の集合が，問題分割戦略 $\overline{P_i}$ の第 i レベルの分割頂点であるとき，その共有木のレベルを i と定める．

【例 5】 Fig. 3(b)の共有木の例を Fig. 3(c)に示す．ここで，破線で示してある部分木が Fig. 2(a)の戦略の第 3 レベルの共有木である．

この補題 3 は本稿の主要な成果であり，同じレベルの分割頂点の集合が，共有木を持つという性質を利用して，各レベルの分割頂点を求める手法が **3・4 節**で示される．

本稿では，正例のみからの戦略の獲得を考えているので，少なくとも目的の問題分割戦略の解集合に相当するトレースは必要になる．ところで，補題 1 により，条件(a)，条件(b)を満たす問題領域ではすべてのトレースが必要になる．したがって次節以降では，次のような問題を考える．

問題 1　条件(a)(b)を満たす問題領域の有限個のトレースの集合 T が与えられたとき，その T を生成可能なトレースの集合とするような問題分割戦略のうち，極小なものを求める．

3・3 共有木の性質

ある有限個のトレースが与えられたとき，それを生

成可能なトレースの集合とするような極小問題分割戦略は複数存在する．例えば，Fig. 2の(a)と(c)の戦略は解集合が等しい．それぞれの戦略を，$\overline{P_i}$, $\overline{Q_i}$ と表すと，Fig. 3(d)には，$\overline{P_iQ_i}$ の第1，第2，第3レベルの共有木が示されている．ただし，ともに第1，第2レベルの共有木は同じである．

ここで，一般の根付き木の間の関係 $\leq_t$ と，共有木間の関係 $\leq_{ct}$ を導入する．

【定義9】 二つの根付き木（共有木） T_1, T_2 において，T_1 が T_2 を部分木として含むとき（ただし，状態の対応は考えない），$T_2 \leq_t T_1$ ($T_2 \leq_{ct} T_1$) と表す．真に含むときは，$T_2 < T_1$ ($T_2 <_{ct} T_1$) を表す．

【定義10】 ある有限個のトレース集合 T とその Prefix Tree PT，および同じレベルの分割頂点の集合 $\{D_i\}$ が与えられたとき，次のグラフ $G=(V, E)$ を，$\{D_i\}$ の分割頂点グラフ (Division Node Graph) と呼ぶ．

$V=\{D_i\}$

$E=\{(D_i, D_j) | PT$ 上で D_i から D_j へのパスが存在し，その途中に V の要素が存在しない$\}$

【例6】 Fig. 3(b)の Prefix Treeにおいて，Fig. 2(a)の第2レベルの分割頂点の集合の分割頂点グラフを Fig. 4(a)に太線で示す．

以下に，共有木の性質を表す二つの補題を示す．

〈**補題4**〉 共有木間の関係 $\leq_{ct}$ は全順序関係である．

《**証明**》 文献(9)を参照． □

［**系1**］ 有限個の共有木の集合が与えられるとその $\leq_{ct}$ の意味で最大の共有木が存在する．

〈**補題5**〉 同じ極小問題分割戦略の共有木について，レベル i の任意の一つの共有木 CT と，レベル $i-1$ の共有木で CT に部分木として含まれる共有木の集合 $\{DT_i\}$ を考えると，DT_i は CT を Prefix Tree としたときの共有木になり得る．

《**証明**》 補題3の必要性の証明における共有木の構成の仕方から明らか． □

3・4 問題分割アルゴリズム

まず，異なる根付き木の間の同型な部分木の一つとして，葉同型な部分木を定義する．

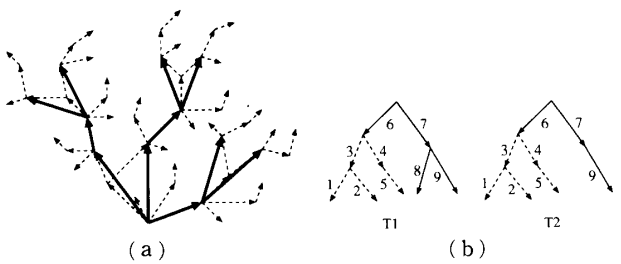


Fig. 4 Division node graph and leaf-isomorphic trees.

【定義11】 二つの異なる(辺にラベルを持つ)根付き木 T_1, T_2 が与えられたとき，次の条件を満たすそれぞれの部分木 ST_1, ST_2 を，葉同型な (leaf-isomorphic) 部分木という．

(1) ST_1, ST_2 はともに同型である．

(2) T_i において，ST_i の頂点から出ていて ST_i に含まれない葉が存在しない．

最大の葉同型な部分木は，木の部分同型性判定アルゴリズム[10]を利用して $O(mn)$ で求められる．ここで，m, n は二つの木のサイズである．

【例7】 二つの部分木の葉同型な部分木の例を Fig. 4(b)に示す．破線で示した部分木が互いに葉同型な部分木である．この場合，最大な葉同型部分木になる．

次に，木とその部分木からなる演算－を次のように定義する．

【定義12】 根付き木 T およびその部分木 T' が与えられたとき，T' に含まれる葉の集合を $\{L_i\}$, $\{L_i\}$ 以外の T の葉の集合を $\{L_i'\}$ とする．このとき，T において T の根から各 L_i' に至るまでのパスは残しながら，T の根から L_i に至るパスを除いた木を，$T-T'$ で表す．

【例8】 演算－の例を Fig. 5(a)に示す．破線で示した部分木 T' を T から取り除くと，右側の $T-T'$ が得られる．

最後に，Div という演算を定義する．ここで，出次数が2以上の頂点を枝分かれ点と呼ぶ．

【定義13】 根付き木 T とその頂点 v が与えられたとき，頂点 v から枝分かれしている各枝より下の木(v を含まない)をそれぞれ DT_i とすると，$\{DT_i\}$ を v を分離点とする分割という．また，この分割を Div(T, v) で表す．

【例9】 Fig. 5(b)に v を分離点とする分割を示す．

これらの演算を用いて，Fig. 6 に最大の共有木を求めるアルゴリズム Algorithm **AL1** を示す．

【例10】 以下に AlgorithmAL 1 の適用例を示す．

Fig. 2(a)の戦略の第3レベルの共有木を求める．まず，Prefix Tree を作成する．これは，Fig. 3(b)に示されている．この木に AlgorithmAL 1 を適用する．

まず，(1)において Fig. 7 の(a)(b)(c)(d)の四つの分割が得られる．このうち，(b)(c)(d)は(a)に

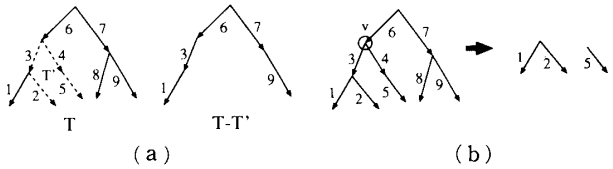


Fig. 5 $T-T'$ and Div (T, v).

真の部分木として含まれるので(2)では(a)が選択される．

checkCommonTree の while ループを 1 回適用した後の RTree は Fig. 7(e)に示す．2 回目のループでは，if の条件に成功せず NO を返す．次に，(3)で Fig. 7(a)の v を分離点とする分割を行う．その結果，Fig. 8(a 1)(a 2)のような分割が得られる．

(4)では，(a 1)(a 2)どちらも選択され得るが，仮に(a 2)が C として選ばれたものとする．

再び，checkCommonTree が呼び出されるが，while ループを 1 回適用した後 RTree は Fig. 8 の(b)となる．その後 while ループを 2 回適用して，Fig. 8(b)の破線部が順次取り出された後，C を返す．この C は第 3 レベルの共有木となっている．

```
入力 ： トレース集合 T の Prefix Tree PT
出力 ： PTの最大の共有木
Select({DTi}) i = 1,...,k
begin
    ST = DT 1
    for i=2 to k do
        if ST が部分木として DTi に含まれる then
            ST = DTi
        endif
    end
    return ST
end
checkCommonTree(C,PT)
begin
    RTree = PT
    while(RTree が空木でない)do
        if RTree が C と 同型な部分木 C' を持つ then
            RTree = RTree - C'
        else
            return NO
        endif
    end
    return C
end
begin
    {DTi} = Div(PT,PTの根) .....(1)
    C = Select({DTi}) .....(2)
    while(Cが空木でない)do
        if checkCommonTree(C,PT) != NO then
            output C
        endif
        {DTi} = Div(C,Cの根) .....(3)
        C = Select({DTi}) .....(4)
    end
    fail
end
```

Fig. 6 Algorithm AL 1.

3・5 Algorithm AL 1 の正しさ

まず，関数 Select に関する補題を示す．

〈**補題 6**〉 Select は入力として，Prefix Tree PT のある頂点における分割 $\{DT_i\}$ で空木以外の木を含むものを受け取ると，$\{DT_i\}$ の中から，最大の共有木を含む木の一つを選ぶ．

《**証明**》 空木でない DT_i は必ず δ コンポーネントを含む．δ コンポーネントは共有木の一つであり，補題 4 の系 1 より，空木でない各 DT_i が含む共有木で最大の共有木が存在し，それをそれぞれ $\mathrm{Max}CT_i$ とおく．

$\mathrm{Max}CT_i$ の最大な DT_i が選択されることを示す．方針としては，$\mathrm{Max}CT_1 <_{ct} \mathrm{Max}CT_2$ ならば $DT_1 <_t DT_2$ であることを示せばよい．詳細は文献(9)参照．□

次に，checkCommonTree に関する補題を示す．

〈**補題 7**〉 checkCommonTree に C として PT の共有木を渡すと，C を返し，C が共有木でない場合は NO を返す．

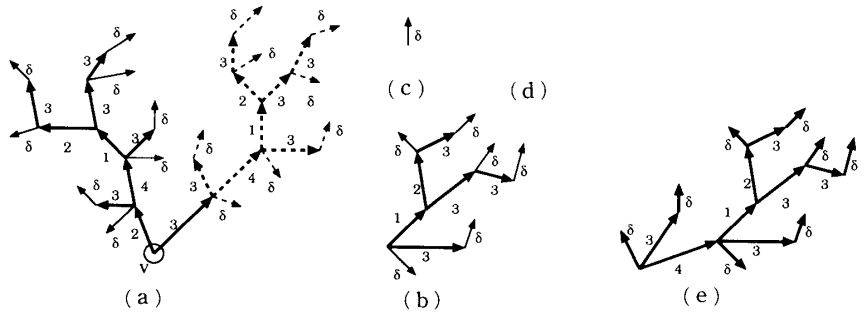


Fig. 7 Application of the algorithm(1).

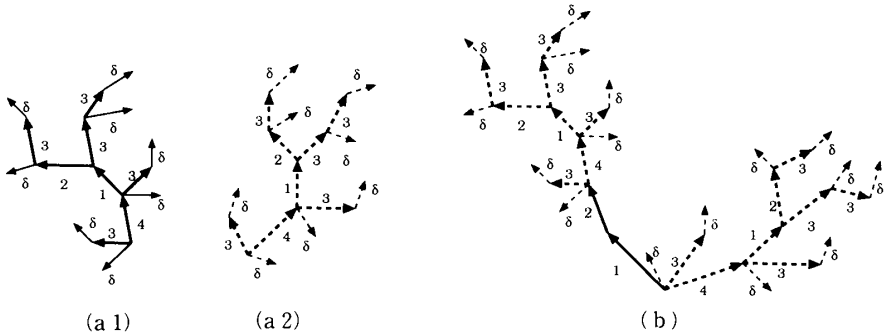


Fig. 8 Application of the algorithm(2).

《証明》 木 C が出力されるなら C が共有木であることを示す．まず，出力される C が補題3の(1)と(3)の条件を満たすことは，if の部分でマッチングをとっていること，RTree が空になって C が出力されることから，明らかである．また，C とマッチングした木 C_1, C_2 が互いに頂点を共有するとすると，C_1 の頂点から C_2 の葉が出ていることになる．これは，(1)で葉同型な部分木のマッチングをとっていることに反する．したがって，補題3の(2)の条件も満たされる．よって，C は共有木である．

次に C が共有木なら C を出力することを示す．まず，入力 C の根と同じレベルの分割頂点の集合を $\{D_i\}$ とし，その分割頂点グラフを G とする．すると，G において葉に相当する分割頂点の共有木が，順次 C と葉同型な木として取り除かれていき，最終的に RTree は空木となる．よって，C が出力される． □

次に，Algorithm **AL 1** の正しさを示す．

〈**定理 1**〉 Algorithm **AL 1** は，Prefix Tree PT の最大な共有木を出力する．

《**証明**》 PT は δ コンポーネントのみからならないので，明らかに，(1)で $\{DT_i\}$ の中に少なくとも δ コンポーネントを含む木が存在する．補題6より，(2)では $\{DT_i\}$ の中から最大の共有木を含む木が選択される．

C が最大の共有木に等しければ，補題7により，C が返される．

C が最大の共有木より真に大きければ，補題7により，NO が返され，(3)において C が真に小さい木の集合 $\{Td_i\}$ に分割される．$\{Td_i\}$ には，必ず最大の共有木を含む木が存在するので，while ループを繰り返すうちに C は最大の共有木に等しくなる．よって，題意は示された． □

Algorithm **AL 1** は最大の共有木を求めるためのアルゴリズムであるが，これを極小問題分割戦略を求めるアルゴリズムに簡単に変換できる．すなわち，与えられた入力に対して最大共有木を求め，それが補題3の条件(4)を満たすかどうかを確かめればよい．もし満足しなければ，次に小さい共有木を求めるために，Algorithm **AL 1** の(3)から再び実行すればよい．また，補題3の条件(4)を満たす最大の共有木が求められたら，その共有木を Prefix Tree として再び Algorithm **AL 1** を適用すれば，補題5より次のレベルの共有木が求められることがわかる．

3・6 Algorithm AL 1 の計算量

トレースの入力個数を N，トレースの長さの最大値を L，極小問題分割戦略の部分問題の個数を K，各部分問題のマクロの個数の最大値を M とする．このとき，一つの最大共有木を求めるのに要する時間を，N, L, M, K で表す．まず，Select には，二つの木の部分同型性判定に $O(N^2 \cdot L^2)$ かかり，それをたかだか M 回行う．よって，$O(N^2 \cdot L^2 \cdot M)$ となる．

checkCommonTree では，葉同型性の判定がたかだか N 回しか繰り返されないから $O(N^3 \cdot L^2)$ である．

したがって，メインのアルゴリズムで while ループが繰り返す回数はたかだか L 回であるから，計算量は $O(N^3 \cdot L^3 + N^2 \cdot L^3 \cdot M) = O(N^3 \cdot L^3)$ となる．Algorithm **AL 1** を補題3の(4)の条件を満たす共有木を求めるように変更してもこのオーダは変わらない．

また，**AL 1** を用いて極小問題分割戦略をすべて探索することも可能である．その際には，極小問題分割戦略の個数（最悪の場合には，トレースの長さに関して指数関数個存在）に関して多項式の時間で抑えることができる．

4. 適　用　例

適用例として，Fig. 1(a)の問題において，ゴール状態を，{in(r, rm 2), in(b, rm 2), open(d), near(r, g), on(b, g)} とした場合を考えた．この場合条件(a)(b)を満たす．問題例としては，ロボットの存在する部屋，箱の存在する部屋，戸の開閉の状態を組み合わせてできる8通りの場合を考え，その解のトレースに本手法を適用すると，Fig. 9のような問題分割戦略が得られる．ただし，各部分問題のゴール状態空間は各分割頂点の状態の最小汎化により求め，オペレーター一つからなる連続する部分問題は展開して表示してある．

また，5-Puzzele，ハノイの塔のパズルの戦略を同定することも成功している．

5. お　わ　り　に

本稿では，情報保存的でゴール状態空間が一つの状態からなる問題に対して，トレースからその極小問題分割戦略を多項式時間で獲得する手法を示した．この

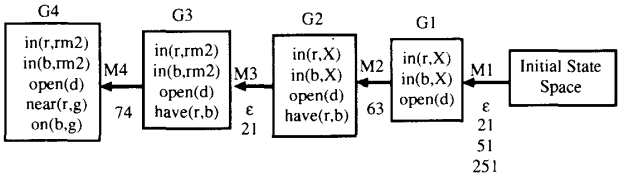


Fig. 9 Problem decomposition strategy for robot planning example.

手法はサブゴール間の干渉が強い問題に対しても有効である．

問題領域に対する制限は，本手法においてそれほど本質的ではなく，より広いクラスの問題分割戦略が，同じアルゴリズムにより，求められることが証明できている．

本手法は，問題分割戦略のサイズから考えると，指数関数個のトレースが必要である．また，すべての例を必要としているので，一般化を行っているとは言い難い．しかし，問題例が等確率で現れるような確率的な学習モデルにおいては，少し制限を加えれば，多項式個のトレースで確率的に正確に同定できることもわかっている．

また，問題分割戦略が完全であるということ，各部分問題で1回のマクロオペレータの適用しかできないということは大きな制約である．これらは，今後の課題である．

謝　　辞

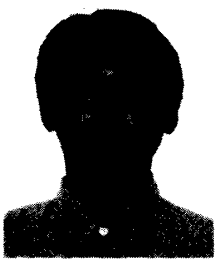
本研究の特にアルゴリズムの部分に関して，工業技術院機械技術研究所の阿久津達也氏には，貴重なコメントをいただきました．ここに，感謝致します．

◇参　考　文　献◇

(1) Güvenir, H. A. and Ernst, G. W.: Learning Problem Solving Strategies Using Refinement and Macro Generation, *Artifi. Intell.*, Vol. 44, No. 1-2 pp 209-243 (1990).
(2) Korf, R. E.: Macro-Operators: A Weak Method for Learning, *Artif. Intell.*, Vol. 26, No. 1, pp 35-77 (1985).
(3) Mitchell, T. M., Keller, R. M. and KedarCabelli, S. T.: Explanation-Based Generalization: A Unifying View, *Machine Learng*, Vol. 1, pp. 47-80 (1986).
(4) Tadepalli, P.: A Formalization of Explanation-Based Macro-Operator Learning, *Proc. of IJCAI'91*, pp. 616-622 (1991).
(5) Tadepalli, P.: Learning with Inscrutable Theories, *Proc. of Int. Machine Learning Workshop'91*, pp. 544-548 (1991).
(6) 山村雅幸，小林重信：拡張EBLに基づく問題解決マクロテーブルの獲得，人工知能学会誌，Vol. 6, No. 1, pp. 72-83 (1991).
(7) 山村雅幸，小林重信：EBLの複数例題下への拡張，人工知能学会誌，Vol. 4, No. 4, pp. 389-397 (1989).
(8) Angluin, D.: Queries and Concept Learning, *Machine Learning*, Vol. 2, pp. 319-342 (1988).
(9) 小林　聡，堀　浩一，大須賀節雄：情報保存的な問題領域におけるトレースからの問題分割戦略の獲得，人工知能学会研資，SIG-FAI-9201-5 (6/12), pp. 45 54 (1992).
(10) Reyner, S. W.: An Analysis of a Good Algorithm for Subtree Problem, *SIAM J. of Comp.*, Vol. 6, No. 4, pp. 730-732 (1977).

〔担当編集委員・査読者：小林重信，滝　寛和〕

著　者　紹　介



小林　聡（学生会員）

1988年東京大学工学部航空学科卒業，1990年同大学院工学系研究科修士課程修了．現在，同大学院博士課程在学中．人工知能，学習理論の研究に従事．特に，説明に基づく学習，帰納推論等に興味を持つ．情報処理学会会員．

堀　浩一（正会員）

1979年東京大学工学部電子工学科卒業．1984年同大学院博士課程修了．同年，国文学研究資料館助手．同助教授を経て，1988年東京大学先端科学技術研究センター助教授，現在に至る．人工知能，特に，発想支援システムの研究に従事．情報処理学会，日本認知科学会，日本ソフトウェア科学会各会員．

大須賀　節雄（名誉会員）

1957年東京大学工学部航空学科卒業．同年，富士精密工業(現：日産自動車)(株)入社．1961年東京大学航空研究所助手，1967年同研究所助教授，1981年東京大学工学部境界領域研究施設教授，1988年同大学先端科学技術研究センター教授，1991年同センター長．工学博士．人工知能，マンマシンインタフェース，データベース，CAD等の研究に従事．情報処理学会等各会員．