

# EBL 効用問題の実用的解決とそのオンライン評価

## A Practical Solution of the EBL Utility Problem and Its On-Line Evaluation

山田 誠二\*  
Seiji Yamada

\* 大阪大学産業科学研究所  
ISIR, Osaka University, Ibaraki 567, Japan.

1993年3月29日 受理

**Keywords:** utility problem, speed-up learning, explanation-based learning.

### Summary

The utility problem has appeared through the intensive research on EBL (Explanation-Based Learning) in recent years. The problem means a trade-off between the benefit of learned knowledge and the cost to locate and apply the knowledge. One of the basic solutions is a search for a benefit and harmless subset of learned knowledge, however most of the searches are very expensive and thus need the off-line evaluation to ignore the learning cost. This paper describes an inexpensive method to search a locally optimal macro-ordering in EBL, macro-operator learning. The complexity of the search is restricted within linear order of the amount of learned macros, and the necessary information including statistics on given problems is obtained through the execution of a simple EBL procedure through partial problems. For verifying the effectiveness of our approach, we implemented an IS-EBL using the inexpensive estimation of the utility and made on-line evaluation with a non-learning system and a non-selective EBL in a map-coloring problem. As a result, we had good experimental results and clarified a class in which the IS-EBL works well.

### 1. はじめに

学習によって問題解決の効率が低下してしまう効用問題 (utility problem) は, EBL (Explanation-Based Learning) [DeJong 86, Mitchell 86] やマクロオペレータ学習などの効率化学習 (speed-up learning) において重要な課題となっている [Minton 88]. 学習による効率低下の主な原因は, 無制限に学習された大量の知識の適用条件の判定にコストがかかることである. 一般に, 学習された知識の一部分でも十分に効率化が可能なことから, 効用問題の解決法として, 有益で少数の知識を選択的に学習することが考えられる. この考え方にに基づき, 学習された知識のうち有効な部分を探索する研究 [Gratch 92, Greiner 92], ヒューリスティクスによりマクロを選択する研究 [Yamada 89] などが行われてきたが, それぞれ問題点がある. 有効な部分的知識の探索では, その探索空間が広いことに加え, 部分知識の評価が, 厳密には問題を解き直さないと求めることができないため, 探索コストが非常に大

きい [Gratch 92, Greiner 92]. また, ヒューリスティクスによるマクロオペレータの選択 [Yamada 89] は, その有効性がオペレータの記述に強く依存する.

本論文では, EBL 効用問題の実用的解決法として, 探索空間を制限して有益なマクロを求める IS-EBL を提案し, その有効性を制約充足問題におけるオンライン評価で検証する [山田 93]. まず, マクロを無制限に学習する単純 EBL を定義し, 次に IS-EBL のアルゴリズムを示す. IS-EBL では, 有益なマクロの候補が全マクロ数の線形オーダに制限され, さらに訓練フェーズで得られる情報から, マクロの効用を見積もれるという意味で実用的である.

### 2. 単純 EBL

単純 EBL の入出力を, 以下に示す.

〈入力〉

- 目標概念
- 領域理論
- 訓練例系列  $ES = [E_1, \dots, E_n]$

```

A simple EBL
ES=[E1, ..., En]
MO=[]
begin
  i ← 0;
  while i ≠ n do
    begin
      i ← i+1;
      if solving Ei with MO fails then
        begin
          solving Ei with domain theory;
          generalization and making a new macro;
          adding the new macro into the tail of MO
        end
      end
    end
  end
end

```

図1 単純EBL

・操作性基準：説明構造の葉を切って一般化する。

〈出力〉

・操作可能なマクロ系列  $MO$

単純EBLのアルゴリズムを図1に示す。入出力はホーン節で記述され、説明の生成とその一般化は、Prolog-EBG[Kedar-Cabelli 87]を用いている。また、新しく学習されたマクロは、マクロ系列の最後に追加される。よって、学習後の問題解決では、古いマクロから順に適用が調べられることになるが、この方法は逆に新しいマクロを先頭に追加するよりもパフォーマンスが良いことが報告されている[Mooney 89]。また、説明木の葉を切るという操作性基準は、広く使われているものであり[Kedar-Cabelli 87]、これによって、学習されたマクロの連鎖的適用が禁止され、後の効用の解析が容易になる。

### 3. IS-EBL

IS-EBLは、まず単純EBLで問題系列の初めの部分系列を解いて、得られたマクロ系列から局所最適な部分系列を選択し、それを用いて残りの問題系列を解く。本研究では、個々の問題がEBLにおける訓練例に対応し、問題は有限な系列から連続して与えられる。

#### 3.1 アルゴリズム

IS-EBLの入力は、単純EBLの入力と部分系列のサイズ(標本サイズ)  $t$  である。部分系列  $[E_1, \dots, E_t]$  ( $1 \leq t < n$ ) は、全体の問題系列を母集団としたランダムサンプルとする。よって、部分系列におけるマクロの肯定例の割合から、全体の問題系列において、ある例がそのマクロの肯定例である確率を統計的に推定可能である。また、この統計的推測の精度は  $t$  に依存するが、現時点では最適な  $t$  の決定法はなく、経験的な

値を用いている。

IS-EBLのアルゴリズムを以下に示す。

〈IS-EBL〉

**Step 1:** 部分問題系列  $[E_1, \dots, E_t]$  に対し単純EBLを走らせ、マクロ系列  $M = [M_1, \dots, M_m]$  を学習する。

**Step 2:** 全体のマクロ系列  $M$  から、有益なマクロ系列の候補となる部分系列  $[ ], [M_1], [M_1, M_2], \dots, [M_1, \dots, M_m]$  を生成する。これらの部分系列は、 $M$  の最後のマクロを一つずつ取り除くことにより得られる。

**Step 3:** それぞれの候補についてその効用を推定し、最大の効用を持つ、つまり最適な部分マクロ系列を選択する。

**Step 4:** 残りの問題系列  $[E_{t+1}, \dots, E_n]$  を、選択されたマクロ系列を使って、学習なしの単純EBLで解く。

Step 2 から明らかのように、マクロ系列の候補数は  $m+1$  であり、全マクロ数  $m$  の  $O(m)$  に抑えられている。また、Step 4 の学習なし単純EBLでは、まずマクロを使って問題解決を試みて、失敗すると領域理論で解き、マクロの学習は行わない。

#### 3.2 効用関数

本研究では、Step 4 の問題解決において一つの例を解く場合に、学習なしシステムよりも短縮されるCPU時間である短縮時間 (saving time) の期待値をマクロ系列  $MO$  の効用関数  $U(MO)$  として用いる。このとき、効用関数は下の式(1)で表される。式中で、 $NLC$  は学習なしシステムが一つの問題を解くために要するコスト (CPU 時間) の期待値であり、 $LC(MO)$  は IS-EBL システムが学習後に選択されたマクロ系列  $MO$  を用いて、一つの問題を解くために要するコストの期待値である。

$$U(MO) = NLC - LC(MO) \quad (1)$$

$U(MO)$  が正の数を返すとき、 $MO$  の学習は問題解決を効率化し、負の数の場合にはかえって効率を低下させてしまう。0 のときは、変化なしである。

#### 3.3 効用の低コストな評価

Step 2 で生成されたマクロの候補の効用を推定するために、以下のパラメータを設定する。また、以降の各平均コストは、問題系列全体の平均である。

・ $EBL(M_i, MO)$ : 学習後の EBL システムが、マクロ  $M_i$  の可解例をマクロ系列  $MO$  で解くために要する平均コスト。

- *No-learning* ( $M_i$ ): 学習なしシステムが、マクロ  $M_i$  の可解例を領域理論で解くために要する平均コスト。
- $Pr(M_i)$ : 全体の問題系列中のある問題が、マクロ  $M_i$  の可解例である確率。

ここで、マクロ  $M_i$  の可解例とは、単純 EBL において  $M_i$  で解かれる問題、つまり  $M_1 \sim M_{i-1}$  では解けない  $M_i$  の肯定例を意味する。

前述のように  $NLC$ ,  $LC(MO)$  は、学習システムと学習なしシステムが、一つの問題を解くのに要するコストの期待値であるから、それぞれの平均コスト  $EBL(M_i, MO)$ ,  $No-learning(M_i)$  とその確率  $Pr(M_i)$  の積の総和で表せる。よって、以下の式が得られる。

$$LC(MO) = \sum_{M_i \in M} Pr(M_i) \cdot EBL(M_i, MO) \quad (2)$$

$$NLC = \sum_{M_i \in M} Pr(M_i) \cdot No-learning(M_i) \quad (3)$$

Step 1 で、 $[E_1, \dots, E_i]$  におけるマクロ  $M_i$  の可解例の数  $N_i$  が数えられ、 $Pr(M_i)$  は統計的推測における比率の点推定を用いて、最尤推定量  $N_i/t$  で推定できる。

$EBL(M_i, MO)$  と  $No-learning(M_i)$  を厳密に求めるには、学習後に同じ問題を再度解く必要があり、これはコストが高すぎる。ところが好都合なことに、 $EBL(M_i, MO)$  と  $No-learning(M_i)$  は、Step 1 の結果だけから推定できる。以下で、このことを説明していく。

まず、Step 2 で得られたマクロ系列を  $MO = (R, D)$  と表す。ここで、 $D = \{M_{s+1}, \dots, M_m\}$  と  $R = [M_1, \dots, M_s]$  ( $1 \leq s \leq m$ ) は、削除されたマクロ集合と残りのマクロ系列を表す。式(2)の  $EBL(M_i, MO)$  を  $R$  と  $D$  に分割して、次式が得られる。

$$LC(MO) = \sum_{M_i \in R} Pr(M_i) \cdot EBL(M_i, MO) + \sum_{M_i \in D} Pr(M_i) \cdot EBL(M_i, MO) \quad (4)$$

$PS(M_i, M_j)$  ( $i \geq j$ ) をマクロ  $M_i$  の可解例をマクロ  $M_j$  だけで解く問題解決とし、 $NLPS(M)$  をマクロ  $M$  の可解例を領域理論で解く問題解決とする。そして、 $\overline{PS}(M_i, M_j)$  と  $\overline{NLPS}(M)$  をそれぞれ  $PS(M_i, M_j)$  と  $NLPS(M)$  の平均コストとする。このとき、 $PS(M_i, M_j)$  ( $i > j$ ) は、マクロ  $M_i$  の可解例の定義より必ず失敗し、 $PS(M_i, M_i)$  は肯定例を解くことなので必ず成功する。例えば、マクロ  $M_i$  の最初の可解例を解くときの学習時の問題解決は、図 2 のようになる。

前述のように操作性基準により、マクロの連鎖が禁止されるので、マクロ  $M_i$  が  $MO$  の  $R$  に含まれている場合の  $EBL(M_i, MO)$  は、 $\overline{PS}(M_i, M_1) \sim \overline{PS}(M_i, M_i)$

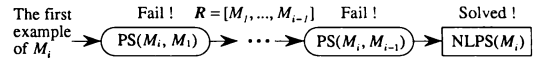


図 2 学習時の問題解決

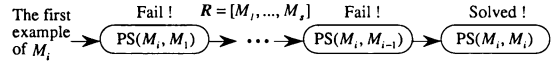


図 3 学習されたマクロの肯定例

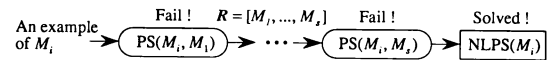


図 4 削除されたマクロの肯定例

の総和となる。図 3 に、 $R = [M_1, \dots, M_s]$  で  $M_i$  ( $1 \leq i \leq s$ ) の可解例を解くようすを示す。 $PS(M_i, M_1) \sim PS(M_i, M_{i-1})$  がすべて失敗し、最後に  $PS(M_i, M_i)$  が成功する。よって、 $EBL(M_i, MO) = \sum_{j=1}^i \overline{PS}(M_i, M_j)$  が得られ、さらに式(4)から下の式(5)が導かれる。

$$\sum_{M_i \in R} Pr(M_i) \cdot EBL(M_i, MO) = \sum_{M_i \in R} Pr(M_i) \sum_{j=1}^i \overline{PS}(M_i, M_j) \quad (5)$$

次に、削除されたマクロ  $M_i$  についての  $EBL(M_i, MO)$  が、 $\overline{PS}(M_i, M_1) \sim \overline{PS}(M_i, M_s)$  の総和と  $\overline{NLPS}(M_i)$  との和として計算される。ここでは、 $PS(M_i, M_1) \sim PS(M_i, M_s)$  ( $s < i$ ) がすべて失敗する。図 4 にそのようすを示す。よって、 $EBL(M_i, MO) = \sum_{j=1}^s \overline{PS}(M_i, M_j) + \overline{NLPS}(M_i)$  が得られ、さらに式(4)から式(6)が導かれる。

$$\sum_{M_i \in D} Pr(M_i) \cdot EBL(M_i, MO) = \sum_{M_i \in D} Pr(M_i) \left\{ \sum_{j=1}^s \overline{PS}(M_i, M_j) + \overline{NLPS}(M_i) \right\} \quad (6)$$

式(5)、式(6)そして式(4)から、 $LC(MO)$  は式(7)のようになる。

$$LC(MO) = \sum_{M_i \in R} Pr(M_i) \sum_{j=1}^i \overline{PS}(M_i, M_j) + \sum_{M_i \in D} Pr(M_i) \left\{ \sum_{j=1}^s \overline{PS}(M_i, M_j) + \overline{NLPS}(M_i) \right\} \quad (7)$$

一方、 $No-learning(M_i)$  は  $\overline{NLPS}(M_i)$  に等しいので、式(3)から下の式(8)が得られる。

$$NLC = \sum_{M_i \in M} Pr(M_i) \cdot \overline{NLPS}(M_i) \quad (8)$$

そして、最後に式(7)、式(8)、式(1)から  $U(MO)$  が得られる。

ここで、 $U(MO)$  を計算するためには、 $\overline{PS}(M_i, M_j)$  と  $\overline{NLPS}(M_i)$  が必要であるが、 $\overline{PS}(M_i, M_j)$  については式(7)からも明らかのように、 $1 \leq j \leq i \leq m$  の範囲で得られればよい。言い換えると、必要な  $\overline{PS}(M_i, M_j)$

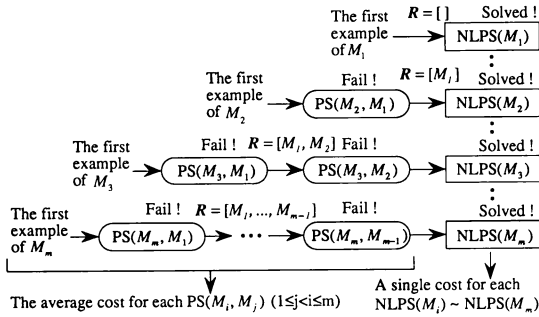


図5 Step 1での単純EBLの問題解決

は、あるマクロの可解例をそれよりも古いマクロで解く場合の平均コストである。ところが、このような  $\overline{PS}(M_i, M_j)$  は、実は Step 1 の単純 EBL の実行から推定可能である。Step 1 では、マクロ  $M_j$  が自分よりも後に学習されるマクロ  $M_i$  ( $j < i$ ) の可解例を少なくとも 1 回は解こうとするので、その実行コストから  $\overline{PS}(M_i, M_j)$  を推定できる。新しいマクロを系列の最後に追加すること、最後から一つずつ削除して部分マクロ系列を生成することが、この好ましい特性の十分条件である。このように  $\overline{PS}(M_i, M_j)$  ( $1 \leq j \leq i \leq m$ ) が、Step 1 の単純 EBL から得られるようすを図 5 に示す。ただし、 $M_i$  の可解例が一つしかない場合は、 $\overline{PS}(M_i, M_i)$  が直接には得られないが、その場合は領域理論による問題解決過程から解析的に求める [山田 92]。次に、Step 1 から得られる、 $M_i$  の最初の可解例を解く  $NLPS(M_i)$  のコストから、 $\overline{NLPS}(M_i)$  を推定する。

以上で述べた効用の推定に必要なすべての情報は Step 1 で得られ、付加的な問題解決は必要ないため、IS-EBL は単純 EBL を用いた場合のコスト最小の選択的 EBL の一つになっている。さらに、Step 2 と Step 3 からわかるように、部分マクロの探索空間のサイズは、全マクロ数の線形オーダーに抑えられ、その範囲内で (局所的に) 最適なマクロ系列が選択的に学習される。よって、IS-EBL は、EBL 効用問題の実用的解決といえる。また、マクロの候補に空リストが含まれるので、最悪の場合でも学習なしシステムと等価になるが、次章のオンライン評価では、学習のコスト分だけ、学習なしシステムよりも損失がある。

また、誤差であるが、統計的推測の理論により、推定される  $Pr(M_i)$  は、経験的値がわからないとき、確率 95% で  $1/\sqrt{t}$  以内の誤差を含む。さらに、 $EBL(M_i, MO)$  と  $No-learning(M_i)$  の精度も、 $t$  とマクロ総数に依存する。アルゴリズム全体での誤差解析および最適な  $t$  の決定方法は、今後の課題である。

## 4. オンライン評価

IS-EBL を評価するために、オンライン評価を行う。ここでのオンライン評価とは、学習コスト込みの CPU 時間を、学習システムと学習なしシステムで競うことを意味する。実際には、比較すべきシステムに同じ問題系列を与えて同時に問題解決を開始させ、すべての問題を解き終わるまでの時間を競う。1 章で述べた学習効率の悪い方法では、オンライン評価に耐えられないと思われる。本実験の目的は、IS-EBL が有効である問題のクラスを探ることである。

### 4.1 実験

対象領域は 2 次元地図の領域を、隣接する箇所は異なる色でぬり分けていく、地図の色ぬり問題 (map-coloring problem) である。この問題は、典型的な制約充足問題であり、一般性のある複雑な問題である。ここでは、領域の数、色数をそれぞれ 9 と 3 に固定し、さらに領域間の隣接関係も固定した図 6 のような地図を用いた。この問題では、 $3^9$  通りの色のぬり方のうち六つの解がある。

地図の色ぬり問題は、変数 (= 地図の領域) 順序、それぞれの変数に代入可能な値 (= 色) 順序、そして隣接する領域は異なった色でぬるという制約により表現される。そして、変数順序と値順序に従って、変数に値を代入していき、制約をチェックして、違反していればバックトラックを行うという単純バックトラック [Kumar 92] を解法として用いた。

EBL の領域理論は、値順序が学習されるように与えられ、マクロはある変数に対しその値順序の何番目の

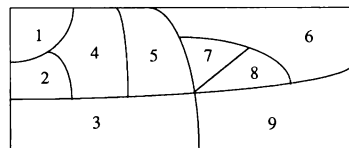


図6 地図

```

?- map_coloring([var1,var2,var3,var4,var5,var6,var7,var8,var9],Ans).
  Ans = [[var1, blue], [var2, green], [var3, blue], [var4, red], [var5, green],
         [var6, blue], [var7, red], [var8, green], [var9, red]]
% A learned macro
map_coloring([_R,_P,_N,_L,_J,_H,_F,_D,_B],
  [[_R,_S],[_P,_Q],[_N,_O],[_L,_M],[_J,_K],
  [_H,_I],[_F,_G],[_D,_E],[_B,_C]]):-
  value_ordering(_R,[_S_K1]),not_same([_R,_S]),
  value_ordering(_P,[_I1,_Q_J1]),not_same([_P,_Q],[_R,_S]),...
% Domain theory
value_ordering(var1,[blue,green,red]).
value_ordering(var2,[green,blue,red]).
value_ordering(var3,[blue,green,red]).
.....
    
```

図7 領域理論、実行例、マクロ

値が解であったかをそれぞれの変数について記憶するので、肯定例をバックトラックなしで解ける。図7に、実験に使われた領域理論の一部、実行例、そしてマクロの例を示す。またここでは、同じマクロの可解例である問題を同じ種類の問題とするため、マクロ数と問題の種類数は一致する。

すべての実験において、IS-EBL、単純EBL、そして学習なしシステムに同じ1000問からなる問題系列が与えられ、同時に問題解決を始めて全部解き終わるまでのCPU時間を測定した。今回の実験では、IS-EBLがStep1で解く問題数は100問に設定され、 $Pr(M_i)$ の推定の誤差はないとした。また、すべての実験で、同じ分布で5回の実験をしてその平均をとった。すべての実験において、三つのシステムはPrologで実装され、コンパイルされた。ただし、生成されたマクロはコンパイルされていない。

以上の条件で、以下の三つの実験を行った。

〈実験1〉 値順序をランダムに設定して、種類の違う問題を生成し、問題の種類を10から90まで10ごとに増やした。この場合、バックトラックが頻繁に起こり、得られた個々のマクロは有益なものになるが、その数の増加とともに効率低下が起こることが予想される。また、 $Pr(M_i)$ の分布は、一様分布を用いた。

〈実験2〉 実験1の $Pr(M_i)$ の分布だけを変えた実験を行った。1種類の問題が、全体の問題系列の $x\%$ を占め、残りはランダムに値順序を変えて問題を生成した。問題の種類に偏りを与えることにより、学習による効率向上が期待できる。

〈実験3〉 バックトラック回数が多い問題と少ない問題を作り、問題の難易度を変えてみた。その結果、難しい問題からは、有効なマクロが得られる。問題の種類(=全マクロ数)を10に固定し、そのなかの有益なマクロ数を0から8まで変えた。難問は、実験1と同様に値順序をランダムに変えて作り、簡単な問題は解から少しだけ値順序をずらして作った。また、問題の種類分布は一様にした。

4.2 実験結果

実験1~3の結果のグラフを図8~図10に示す。左の縦軸は正規化された効用で、右の縦軸は学習されたマクロ数を表す。正規化された効用とは、全問題を解いた際のEBLシステムの短縮時間を、学習なしシステムが要したCPU時間で割ったものである。この値が正のとき効率化がなされ、負のとき効率が低下する。0.5のときが、2倍の高速化に相当する。

図8では、40種類ですでに効用が負になり、学習な

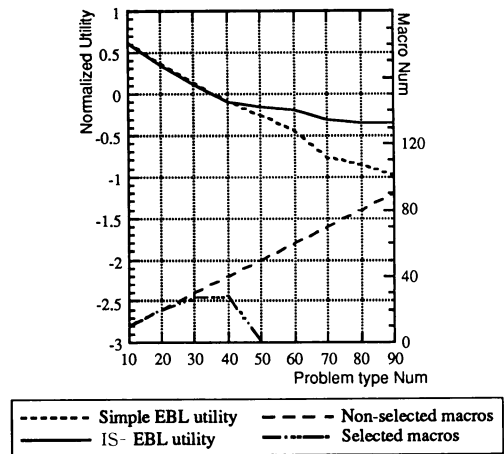


図8 実験1の結果

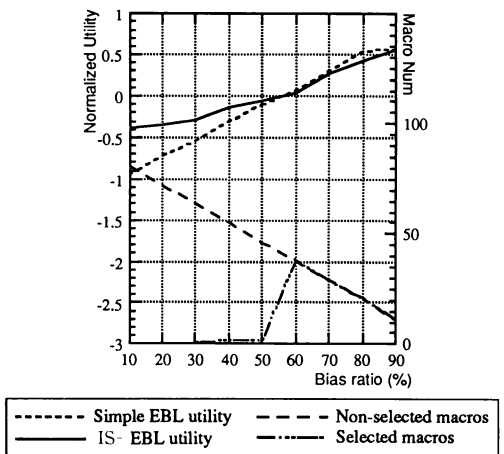


図9 実験2の結果

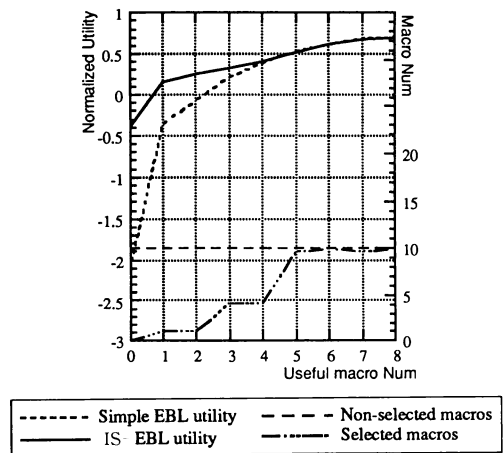


図10 実験3の結果

しシステムが最も良い効率を示している。マクロ数を見るとよくわかるが、10~20種題ではIS-EBLはすべてのマクロを選択しており、そのパフォーマンスは単

純 EBL と同じになっている。もちろん、IS-EBL では、マクロ部分系列の効用評価にコストがかかるが、マクロ数が多くないので、そのコストは小さい。また、50 種類以降では、マクロをまったく学習しないことを選択して、学習なしシステムと同様のパフォーマンスを示す。IS-EBL が他の二つのシステムよりうまく動作する場合は、部分マクロ系列が最適として選択されるときと考えられるが、そのようなケースが 30 と 40 種類に現れている。ただし、この実験では、IS-EBL はマクロをすべて学習するか、まったくしないかの切り換えを行っている傾向が強い。また、90 種類において単純 EBL と IS-EBL の効用は大きく差がついている。

次に図 9 では、図 8 とは対照的に全マクロ数が 40 (横軸 60% のところ) になっても、二つの学習システムはともに学習なしシステムより効率が良い。このことは、マクロの可解例の偏りが EBL にとっても良い効果があることを示している。また、部分マクロが最適のケースは、40% と 50% に現れており、それ以外では図 8 と同様にすべてか無かの切り換えがなされている。

図 10 では、部分マクロが最適である範囲が、10~40% という比較的広い範囲を占め、そこでは IS-EBL が最も良い効率を示している。特に 10% と 20% では、IS-EBL だけが学習なしシステムよりも効率が良い。このことから、問題の難易度(≒マクロの効用)が不均一な場合に、IS-EBL が効果的であることがわかる。

最後に、図 11 は実験 3 の有益なマクロが 30% のときの三つのシステムのオンラインパフォーマンスである。横軸が解いた問題数で、縦軸が延べ CPU 時間である。最初の 100 問まで、IS-EBL は単純 EBL と同じよ

うに問題を解き、100 問目で最適なマクロ系列を評価選択して、それ以降はそのマクロ系列を用いて問題解決を行う。その様子が、IS-EBL の延べ CPU 時間が、100 問以降に単純 EBL のそれから離れていくところに現れている。そして、ついには学習なしシステムよりも良い効率が達成される。

以上の実験結果から得られた知見を、以下にまとめる。

- オンライン評価でも、IS-EBL は単純 EBL より効率が良い場合が多い。
- IS-EBL は、学習の是非を低コストで決定可能であり、効用問題を回避できる。
- IS-EBL は、可解例の偏りや個々のマクロの効用の不均一性を含む問題に対し、特に有効である。

## 5. 関連研究

Minton の PRODIGY [Minton 88] は、ヒューリスティックにより制御ルールを選択的に学習する。制御ルールの短縮時間は一つの例から見積もられ、その後そのルールが適用されるたびにその値が割り当てられる。PRODIGY はマクロの学習はしないので、実験的には比較できないが、本手法との重要な違いは、PRODIGY は学習なしシステムよりも効率を低下させるような有害な知識を学習する場合があることである。我々の IS-EBL は、そのような知識は学習しない。

Gratch の COMPOSER [Gratch 92] と Greiner の PALO [Greiner 92] は、ともに山登り法で局所的かつ確率的に最適な制御ルールを探索する。この二つの手法の主な違いは、得られたルールの最適性の信頼度である。これらもまた、IS-EBL と直接比較できないので、ここでは探索コストを考える。彼らの手法では、その探索空間のサイズが線形オーダで抑えられていないし、特に PALO は非常に探索のコストがかかることが報告されている [Gratch 92]。高い学習効率はオンライン評価では必須となり、彼らの手法ではオンライン評価で学習なしシステムをしのぐことは不可能であると考えられる。

Laird も、並び換え (reordering) と展開 (unfolding) により、局所最適な変換を山登り法で探索する方法を示し、そして統計的情報を獲得する手法として DTAG を提案した [Laird 92]。しかし、我々の手法での主要な変換である展開後の並び換えを使っていない点が異なるし、学習の停止が線形オーダではない。

また、筆者もすでに EBL 効用の計算法を提案している [山田 92]。しかし、その方法は解析的で、ホーン

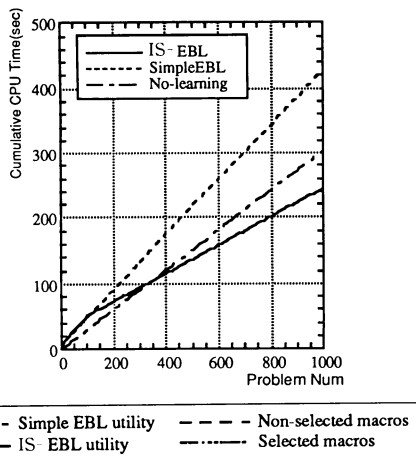


図 11 オンライン評価の典型例

節で表現された領域理論とSLD導出による問題解決に依存しており一般性に乏しい。また、「冗長な訓練例はない」などの厳しい仮定が必要である。それに対し、IS-EBLは経験的であるためコストがかかるが、表現に依存せず、また厳しい仮定も必要としない。

## 6. ま と め

局所的に最適なマクロ系列を選択し、EBL効用問題を解決する手法を提案した。本手法は、全体のマクロ数の線形オーダでその探索空間のサイズが抑えられ、個々の候補の効用評価に必要な情報は、単純EBLの実行結果からすべて得られるという意味で実用的である。この効用評価法を用いたIS-EBLを実装し、制約充足問題においてオンライン評価によりその有効性を

確認した。最後に、今後の展望を示す。

- 動的確率分布に対応可能にして、連続的に再組織化するシステムの実現を目指す。
- IS-EBLを一般化し、事例に基づく学習における事例の選択問題および事例かルールかの選択問題への適用を目指す。

## 謝 辞

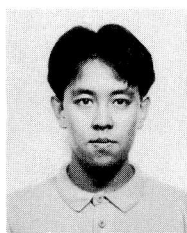
研究をサポートしていただいている豊田順一教授に感謝いたします。また、有益なコメントをいただいた石田 亨氏(NTT, 現在 京大)、山口智浩氏(阪大)、そしてお世話になっている豊田研FAIグループに感謝いたします。最後に、的確なご指摘をいただいた査読者の方に感謝する次第です。

## ◇ 参 考 文 献 ◇

- [DeJong 86] DeJong, G. and Mooney, R.: Explanation-Based Learning: An Alternative View, *Machine Learning*, Vol. 1, No. 2, pp. 145-176 (1986).
- [Gratch 92] Gratch, J. and DeJong, G.: COMPOSER: A Probabilistic Solution to the Utility Problem in Speed-up Learning, *AAAI-92*, pp. 235-240 (1992).
- [Greiner 92] Greiner, R. and Jurisica, I.: A Statistical Approach to Solving the EBL Utility Problem, *AAAI-92*, pp. 241-248 (1992).
- [Kedar-Cabelli 87] Kedar-Cabelli, S. T. and McCarty, L. T.: Explanation-based Generalization as resolution theorem proving, *4th Int. Machine Learning Workshop*, pp. 383-389 (1987).
- [Kumar 92] Kumar, V.: Algorithms for Constraint-Satisfaction Problems, *AI Magazine*, Vol. 13, No. 1, pp. 32-44 (1992).
- [Laird 92] Laird, P.: Dynamic Optimization, *9th Int. Machine Learning Conf.*, pp. 263-272 (1992).
- [Minton 88] Minton, S.: *Learning Search Control Knowledge*, Kluwer Academic Publishers, Boston (1988).
- [Mitchell 86] Mitchell, T. M., et al.: Explanation-Based Generalization: A Unifying View, *Machine Learning*, Vol. 1, No. 1, pp. 47-80 (1986).
- [Mooney 89] Mooney, R. J.: The Effect of Rule Use on the Utility of Explanation-Based Learning, *IJCAI-89*, pp. 725-730 (1989).
- [Yamada 89] Yamada, S. and Tsuji, S.: Selective Learning of Macro-operators with Perfect Causality, *IJCAI-89*, pp. 603-608 (1989).
- [山田 92] 山田誠二: 論理プログラミング環境におけるEBLの有効性計算, *人工知能学会誌*, Vol. 7, No. 2, pp. 309-319 (1992).
- [山田 93] 山田誠二: EBL効用問題の低コスト解決法とそのオンライン評価, *情報処理学会人工知能研*, 93-AI-87-10, pp. 85-94 (1993).

[担当編集委員: 有川節夫, 査読者: 廣渡栄寿, 有川節夫]

## 著 者 紹 介



山田 誠二(正会員)

1984年大阪大学基礎工学部卒業。1989年同大学院博士課程修了。同年、基礎工学部システム工学科助手。1991年より大阪大学産業科学研究所講師、現在に至る。工学博士。人工知能、特に、効率化学習、即応プランニング、ロボットナビゲーション、マルチエージェント系に興味を持つ。情報処理学会、日本認知科学会、

日本ソフトウェア科学会、AAAI、IEEE各会員。