ＢＤＤを用いた仮説推論の並列化の一方法

1Ｄ－３

大野 典　原 耕治　加藤 昇平　世木 博久　伊藤 英則
名古屋工業大学

1 はじめに

仮説推論は仮説という不完全な知識を扱う高次推論の一形式であり，これまでさまざまな研究結果が報告されている [1]．しかし，その計算量が極めて大きいことから，いかにして効率良く高速に推論を行なうかが重要な課題となっている．

そこで，我々は論理関数の表現，論理演算を効率良く行なえる二分決定グラフ [2](Binary DecisionDiagram,BDD) を利用することにより，仮説推論の高速化を検討している [5]．

本研究では，BDD を用いた仮説推論の並列化の一方法を提案し実験を行なったので，その実験結果について報告する．

2 コスト付き仮説推論

知識ベース中の各仮説それぞれにコストを与え，観測 $\mathcal{O}$の説明の中で，説明に含まれる仮説のコスト総和が最小のもの (最適な説明) を求める仮説推論を考える．

定義 2.1 $\mathcal{F}$を命題ホーン節集合 (事実)，$\mathcal{O}$を基底アトムの連言 (観測)，$\mathcal{H}$を基底単位節の集合 (候補仮説の集合) としたとき，以下の条件を満たす $\mathcal{H}$の部分集合 h を $\mathcal{O}$の $\mathcal{F}\cup\mathcal{O}$による説明と呼ぶ．

$\mathcal{F}\cup h \vdash \mathcal{O}$　($\mathcal{F}\cup h$ から $\mathcal{O}$が証明可能)

$\mathcal{F}\cup h \nvdash false$　($\mathcal{F}\cup h$ が無矛盾)

$cost(h) \rightarrow min$　(h の仮説コスト総和が最小)　□

ここで，$\mathcal{F}$は常に成り立つ知識として扱われる．一方で，$\mathcal{H}$の部分集合は $\mathcal{F}$と矛盾する可能性がある．従って $\mathcal{H}$に対する無矛盾性制約として，例えば $false \leftarrow A_1, \cdots, A_k$(アトム $A_1, \cdots, A_k$が同時に真となるとき矛盾) の形をした節集合を用いて無矛盾性の検査を行う．

3 二分決定グラフ

BDD は，論理関数のすべての変数について，再帰的に Shannon 展開を適用した結果を，二分決定木で表現したものである．この二分決定木は閉路を含まない．また，BDD は論理関数を一意に表すことができる．BDD の一つの節点は，節点に与えられた変数名のラベルと "0"，"1" のラベルを持つ二本の枝により構成されている．"0" ラベルの枝を 0 枝，"1" ラベルの枝を 1 枝とする．Shannon 展開においてある変数が"0","1"のどちらに展開されたかについては，その変数に対応する BDD の節点において 0 枝，1 枝のどちらを通るかで表現される．

4 BDD を用いた仮説推論

BDD を用いたコスト付き仮説推論は以下の手続きで実現される．

知識ベース $\mathcal{F}\cup\mathcal{H}$ および観測 $\mathcal{O}$ が与えられたとき，

i)$\mathcal{F}\cup\mathcal{H}$に $\mathcal{O}$を頭部に持つようなルール ($\mathcal{O} \leftarrow B_1, B2, \ldots, B_n$) があれば，$\mathcal{O}$を $B_1, B_2, \ldots, B_n$に置換する．これを繰り返し，観測に関連する仮説のみから構成される論理関数 $H(h_1, \ldots, h_n)$ を生成する．

ii)$H(h_1, \ldots, h_n)$ の BDD を構築する．このとき，BDD の各ノードに「自ノードから 1 節点に至る経路の最小コスト」を与える最小コスト経路探索準備処理を行なう．

iii) 構築された BDD に対して 1 節点に至る経路の無矛盾性検査を行なう．

iv) 1 節点に至る最小コストの経路において 1 枝を選択したノード（仮説）の集合を最適解とする．

これら手続きのうち，ii),iii) の計算量が極めて大きいため，それらの並列化の一方法を提案する．

5 並列化アルゴリズム

本研究で提案する並列アルゴリズムは，前節における手続き i) の結果を Shannon の展開定理に基づいて分割し，各プロセッサ (PE) に割り付け並列に推論を行なうものである．各 PE では割り当てられた部分問題に対して手続き ii)～iv) を並列に実行することにより解候補を求める．全 PE において得られた解候補の中で最小のコストを持つものを最適解とする．

本アルゴリズムではさらに手続き ii) で実行される apply 演算 [2] において閾値 (定義 5.1 参照) を用いて探索空間を縮小している．

5.1 論理式の展開方法

Shannon の展開定理によると論理式 $f(x_1,x_2,\ldots,x_n)$ は $\overline{x_1}f(0,x_2,\ldots,x_n) + x_1f(1,x_2,\ldots,x_n)$ で表現できる．本方式は Shannon の展開定理を再帰的に用いることにより，i) の結果を 2^m(> PE 台数) 個の論理式に分割し，各 PE に割り当てている．このように論理式に現れる仮説の真偽値を 0 又は 1 に割り当てることにより PE に与える問題を分割しているが，1 に割り当てられた仮説のコストが現時点で得られている解候補のコストの最小値よりも大きくなる場合には，その問題を削除する．

処理の終了した PE には別の問題を割り付ける．これを，割り付ける問題がなくなるまで繰り返し，得られた解候補のうち最小のコストのものを最適解とする．

A parallel cost-based abductive reasoning method using Binary Decision Diagrams

Satoru Oono, Koji Hara, Shohei Kato, Hirohisa Seki and Hidenori Itoh.
Nagoya Institute of Technology.
Gokiso-cho, Showa-ku, Nagoya 466, Japan

定義 5.1 $A_i(0 \leq i \leq n)$ を仮説 $(h_0, h_1, \ldots)$ のみから構成される論理式とする．また，BDD_{A_i}を A_i を表現する BDD，BDD_{B_i}を $BDD_{B_{i-1}}$ と BDD_{A_i}との *Apply*演算により構築した BDD，$cost_{A_i}$,$cost_{B_i}$を BDD_{A_i},BDD_{B_i}における根ノードから 1 節点に至る最小コスト，MAXを仮説に与えられたコストの最大値，χ をある正の値とする．以下のアルゴリズムによって閾値を決定する．

if 解候補が得られていない **then**
　すべての A_iについて
　　閾値を ∞ として BDD_{A_i}を構築する．
　$BDD_{B_0} := BDD_{A_0}$
　$\chi := 0$
　while$(1 \leq i \leq n)$
　　閾値$_i := cost_{B_{i-1}} + cost_{A_i} + \chi$
　　閾値$_i$を用いて $BDD_{B_{i-1}}$と BDD_{A_i}との *Apply* 演算により BDD_{B_i}を構築する．
　　if BDD_{B_i} において根ノードから 1 節点に至るパスが得られない **then**
　　　$\chi :=$ 閾値$_i + MAX * 2$
　　　$i := 1$
　　else
　　　$i := i + 1$
else
　閾値$_i :=$ 得られている解候補の最小コスト　□

5.2 並列計算機 AP1000 への実装

本研究では，分散メモリ MIMD 型並列計算機（富士通 AP1000）を使用し，プログラムの作成には C 言語を使用した．プログラムは，各 PE のセルプログラムと，ホストプログラムからなる．ホストプログラムは，各 PE からの解候補の受信，それらから最適解の選択，Shannon 展開に関する情報と現時点の最小コストのセルへの送信を行なう．セルプログラムは，ホストプログラムからの送信により得られた Shannon 展開に関する情報を基にＢＤＤを構築し，根ノードから 1 節点へ至る最小コスト経路において，1 枝を選択したノード（仮説）の集合を部分解とし，(Shannon 展開において 1 を割り当てた仮説の集合)∪ 部分解 を解候補として，ホストプログラムに送信する．

6 実験結果

本研究で提案した並列アルゴリズムの有効性を確認するために実験を行った．例題として n bit 全加算器故障診断問題を用いた．回路を構成する論理素子の故障状態を仮説で表現し，そのコストは素子の耐故障信頼性を数値化し与えた．対象とする全加算器の bit 数を変化させることにより問題規模を変化させ，推論時間 (実時間) および並列効果について評価を行った．
実験結果を図 1，図 2 に示す．図 1 では，対象とする全加算器の bit 数を 1bit ～15bit まで変化させて実行時間を測定した結果を示す．本方式では，問題規模の比較的大きなものに対して有効であることが分かる．

図 2 では，15bit の故障診断問題における並列効果を示す．実線は本研究で提案した方法を示している．

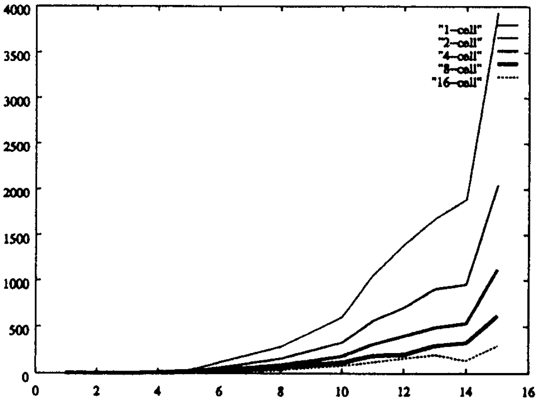

図 1 問題規模による実行時間

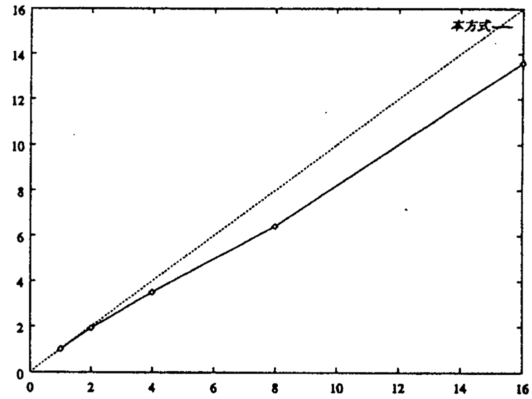

図 2 並列効果

7 おわりに

コスト付き仮説推論問題において BDD を用いた並列化を提案した．さらに実験 (15bit 全加算器故障診断問題) において 16 プロセッサで 13 倍程度の並列効果 (図 2 に示した) を得た．これにより，本アルゴリズムの有効性を確認した．今後は，より大規模な問題に対応できるようにすることが課題である．

参考文献

[1] 井上克巳, アブダクションの原理, 人工知能学会誌, Vol.7,No.1, pp.48-59, January, 1992.

[2] 湊真一, 計算機上での BDD の処理技法, 情報処理, Vol.34, No.5,pp.593-599, May, 1993

[3] 木村晋二, BDD の並列処理技術, 情報処理, Vol.34, No.5, pp.624-630, May, 1993

[4] 木村晋二・松本高明・羽根田博正，非共有記憶型並列計算機での二分決定グラフの並列アルゴリズムについて，DA シンポジウム'93, pp.129-132, 1993

[5] 原耕治・加藤昇平・世木博久・伊藤英則，二分決定グラフを用いた仮説推論処理の一手法，情報処理学会 第 52 回全国大会