

G-5

事例ベース推論を用いた負荷分散方式 Load Balancing by Case-Based Reasoning

井上 裕之[†]
INOUE Hiroshi

六沢 一昭[‡]
ROKUSAWA Kazuaki

1. はじめに

本稿では、事例ベース推論を基に探索木の状況に応じて負荷分散を行う方式について述べる。

事例ベース推論とは、「過去の事例から現状に最も類似するものを検出し、それを基に問題解決を導く」というものである。

本方式では、探索問題を解くたびに新たな事例が生成される。このため、繰り返し探索を行うことで推論の材料となる事例が増え、より適した負荷分散が行われることが期待できる。

2. 事例ベース推論

事例ベース推論の具体的な手順を以下に示す。

1. 過去の問題解決例を抽象化せずに事例として蓄えておく。
2. 新しい問題 (これを試験例と呼ぶ) が与えられる。
3. 試験例と最も類似する事例を検出し、その事例を基に試験例の問題解決を導く。

試験例と事例との類似について、図1の例を使って説明する。既に事例 ($I_1 \sim I_3$) が蓄えられている所に、新たに試験例 (I_4) が与えられたとする。属性値 ($E1, E2$) が構成する空間における I_4 と $I_1 \sim I_3$ との距離を計算する。 I_4 との距離が最も短い I_1 が、 I_4 と最も類似している事例であるとする。

本方式では、試験例と最も類似する事例を検出し、その事例にて観測された事象が試験例に対しても起こると推論する。

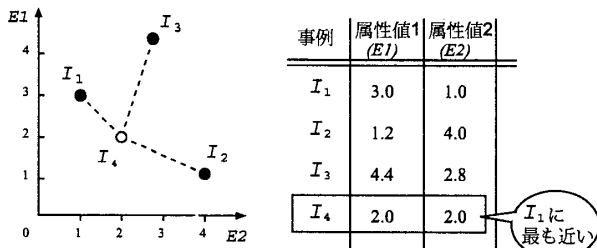


図 1: 試験例と最も類似する事例

3. 本方式の概要

本方式は、探索木の状況に応じた負荷分散を、事例ベース推論を用いて行う。ここでは、そのための事例の定義と具体的な手順を述べる。

3.1 事例の設定

一般に、事例を構成する属性の種類を増やすと、推論の精度は高くなるが、推論の時間も増加する。一方、負荷分散を決定する処理は、余分な処理である。以上のことから、事例ベース推論はできるだけ短い時間で終わることが望ましい。そこで本方式で用いる事例は、事例の検索が軽い処理となるべく次のように設定した。

事例を、過去に行われた探索にて生成された探索木における一つのノードに対する以下の三つの要素によって定義する。

- ・ 属性値
- ・ 評価値
- ・ 分散判定

属性値と評価値に関しては以下で詳しく述べる。分散判定とは、そのノードが探索の過程において他の計算機へ投げられたかどうかである。

3.2 事例の属性値

事例の属性値として、対象のノードにおける以下の三つを用いる。

階層 ルートからの距離 (深さ)。

兄弟番号 親ノードから見て何番目の子供であるか。

引越し回数 ルートからこれまでに経由した計算機の延べ台数。

3.3 事例の評価

本方式では評価値として、以下の二つをそれぞれ正規化して用いる。

探索時間 探索にかかった時間。

負荷の均一度 探索処理に参加した全ての計算機における負荷の均一度。

この評価値は、その事例において分散判定が適切に行われたかどうかを表している。評価値が高い事例は分散判定が適切であり、逆に評価値が低い事例は適切ではない。

[†]千葉工業大学 大学院 工学研究科 情報工学専攻

[‡]千葉工業大学 情報科学部 情報工学科

ここで、事例を検索する際に、評価の低い事例が検出されることを考える。このような場合、基となる事例自体の評価が低いため、試験例が高評価を得ることは期待できない。そこでこのような事態を回避するために、事例を検索する際には、検出された事例をそのまま採用するのではなく、しきい値を用いてその判定を行うようにする。これにより、適切に分散判定が行われなかった事例が検出されても、試験例がその悪しき前例に倣うことを防ぐことができる。

本方式ではこのしきい値を、全事例の平均評価値とする。これにより、試験例に適用される事例は適切に分散判定が行われたものとなり、その評価も期待できる。そして、探索処理を繰り返すことで適切に分散判定が行われた事例が増えていき、それによって平均評価値が上り、しきい値も良くなっていく。このことから、本方式を用いることにより、探索を繰り返すことでより適切な分散判定が期待される。

3.4 事例ベース推論の手順

(1) 事例の検索

しきい値として、全事例の平均評価値を用いる。

探索過程にて新たなノードが生成されるたびに、そのノード以降の探索を他の計算機に投げ出すかどうかを過去の事例から判断する。そのために、試験例(新たに生成されたノード)と最も類似する事例を検出する。

同距離の事例が複数検出された場合には、評価値の良い事例を選ぶ。

(2) 推論

(1) で求めた事例が本当に試験例に対して有益なものかどうかを、その事例の評価値としきい値とを比較して判定する。

評価値がしきい値以上ならば、その事例の分散判定の通りに試験例を処理する。つまり、今新たに生成されたノード(試験例)と良く似たノード(検出された事例)に対して過去に行った分散判定が、新たに生成されたノードに対しても有効であると推論する。

評価値がしきい値未満ならば、その事例の分散判定は試験例に対して有効ではないと判断し、それとは逆の処置を試験例に対して施す。

(3) 事例の生成

探索終了後(解の発見後)に、自分が生成した各試験例に評価値を加え、新たな事例とする。それを事例ベースに蓄え、以降の探索に活用する。

3.5 事例生成の例

図2の結果、各計算機に表1のような事例が生成される。事例A1、A2、A3、A4はノードB、C、D、Eから、

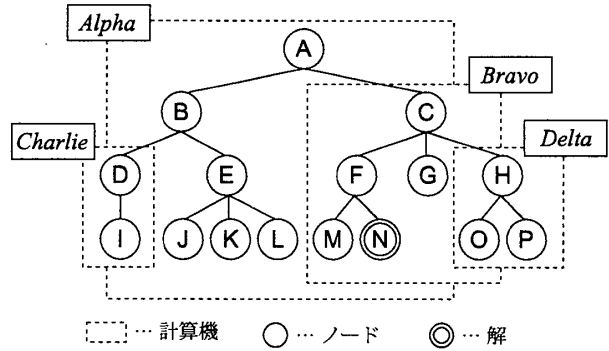


図2: 事例生成の例

表1: 生成される事例

計算機名	事例名	属性値	評価値	分散判定
		(i, j, k)	(l, m)	
Alpha	A1	(1, 1, 0)	(t, b)	N
	A2	(1, 2, 1)	(t, b)	Y
	A3	(2, 1, 1)	(t, b)	Y
	A4	(2, 2, 0)	(t, b)	N
Bravo	B1	(2, 1, 1)	(t, b)	N
	B2	(2, 3, 2)	(t, b)	Y

i: 階層, j: 兄弟番号, k: 引越回数
l: 探索時間, m: 負荷の均一度

事例B1、B2はノードF、Hから、それぞれ生成される。但し、探索結果の評価は、探索時間がt、負荷の均一度がbであったとする。また分散判定は、そのノードを他の計算機に投げ出していればY(Yes)、投げ出していなければN(No)が書き込まれる。

3.6 事例がまだない場合

本方式では、事例がないことには負荷分散が行えない。そこで、探索を行う計算機が事例を一つももっていない場合は、新たに生成されたノードを他の計算機に投げ出すかどうかの分散判定を、乱数を用いてランダムに行うことにする。

しかし、ランダムで行われた分散判定が、そのノードに適切であるとは限らない。分散判定が適切に行われなかった結果、評価の悪い事例が生成されてしまい、次回以降の探索に悪影響をおよぼしかねない。この問題に対しても、事例検出時にしきい値を用いることで対応することができる。

4. まとめ

本稿では、「過去の負荷分散経験を活かして現状に最適な負荷分散方法を導き出す」という人工知能的な解法を負荷分散へ適用することを試みた。本方式の特徴は、探索木における負荷分散という問題に対して、事例ベース推論という人工知能的な解法を適用したところにある。このことから、探索を繰り返し行うことでより適切な負荷分散が期待される。