

混合整数計画法を用いた属性付き有向グラフの差分評価方式 Difference Extracting for Directed Graphs with Attributes Based on Mixed Integer Programming

笠原 孝保[†] 阿部 育世[‡] 小林 克年[¶] 中島裕信[¶]

Takayasu Kasahara Ikuyo Abe Katsutoshi Kobayashi Hironobu Nakajima[‡]

1. はじめに

エンジニアリング図面などの変更管理は、もとの図面と変更後の図面との差分を把握することが基本となる。UML^[1]等、属性付き有効グラフとしてモデル化できる図書やエンジニアリング図面は多い。差分抽出にあたっては、どこまでを差分とみなすかの定義をドキュメントの種類や差分抽出の目的に合わせて柔軟に定義できることが必要となる。

本研究は、属性付き有効グラフを対象として、混合整数計画法に基づいたモデル化により、評価関数の重み付けの設定などにより、目的にあった柔軟な差分抽出が可能であることを示した。

2. 混合整数計画法によるモデル化

2.1 属性付き有向グラフの構成

本研究の対象となる属性付き有向グラフとは、ノードと、ノード相互をつなぐリンクからなり、ノードおよびリンクには属性が付与されるものである。表1に、属性付き有向グラフの典型的なノード属性とリンク属性を記載した。ノードにはノードの種類とノード固有のIDを属性としてもち、リンク属性は、リンクの種類、始点ノードのID、終点ノードのID、始点ポートの種類、終点ポートの種類をもつ。属性付き有向グラフの例として、UMLの状態遷移図、およびロジック図について、どのようにノードおよびリンクの属性を対応づけられるかをTab. 1に示した。例えば、ノードの種類は、ロジック図では、ロジック記号の種類に対応し、信号の種類は、アナログ/デジタル信号などの信号の種類と対応づけられる。また、UML状態遷移図では、ノードの種類は、状態名に、リンクの種類は、遷移の条件と対応づけることができる。このように、属性付き有効グラフが、UMLの状態遷移図や、ロジック図など広範囲のエンジニアリング図面に対応することができる。

Tab. 1

	属性付き有向グラフモデル	UML状態遷移図	ロジック図
ノード属性	ノードの種類	状態名	記号種別(AND,OR,FF,...)
	ノードID	ノードID	ノードID
リンク属性	リンクの種類	遷移の条件	信号の種類(A/D)
	始点ノードID	始点ノードID	始点ノードID
	終点ノードID	終点ノードID	終点ノードID
	始点ポート種別		始点ポート種別
	終点ポート種別		終点ポート種別

2.2 問題の定義

ここで、属性付き有向グラフにモデル化できる、ロジック図などのエンジニアリング図面の差分を評価する問題を考える。簡単のため、参照図面と、差分評価図面の2枚の図面の差分を評価する。図面は以下の構成要素からなる。

- (a) 記号要素：グラフのノードにモデル化できるもので、少なくとも以下の2つの属性をもつ。以下では単に、要素とも呼ぶ。
 - ・記号種別：ロジック図では AND や OR などの記号の種類にあたる。
 - ・エレメントID：1つの図面の中で、1つの記号に固有の値をもつID
- (b) 接続線：一組の入出力端をもつ線で、有向グラフのリンクにあたる。以下の属性をもつ
 - ・入力記号エレメントID：接続線の始点となる記号のエレメントID。
 - ・入力ポート：接続線の始点となるポートの種別名。ロジック記号の AND や OR などでは同じポートに多くの入力があることになる。
 - ・出力記号エレメントID：接続線の終点となる記号のエレメントID。
 - ・出力ポート名：接続線の始点となるポートの種別名。

このように定義されたエンジニアリング図面の参照図と差分評価図をそれぞれの各記号要素が、どんな種類の記号とどのように接続線でつながっているかに関わる類似性に基づいて、記号要素のペアを設定し、あふれた記号要素を差分として出力する問題として、差分評価問題を定義する。

2.3 定式化

参照図面と差分評価図面中の記号を対応づけるため混合整数計画法に基づき、以下のようにモデル化した。

(1) 対応変数の定義

参照図面の記号種別 C_i ($i = 1, \dots, Cnum$; $Cnum$ は、参照図面、または、差分評価図面全体での記号種別数) の、 n 番目の要素 ($n = 1, \dots, Rnum(i)$; $Rnum(i)$ は参照図面における記号種別 i の個数) が、差分図面の記号種別 C_i の m 番目の要素 ($m = 1, \dots, Dnum(i)$; $Dnum(i)$ は差分図面における記号種別の個数) と対応付くか否かをバイナリ変数 $I(C_i, n, m)$ で表す。値として1をとる場合が対応づく事を、0をとる場合が対応づかない事を表す。

(2) 評価関数の定義

バイナリ変数 $I(C_i, n, m)$ に対して、参照図面中の要素 (C_i, n) と差分評価図面中の要素 (C_i, m) との類似度の評価関数を式(1)のように定義した。ここで、 $E_1(C_i, n, m)$ 、 $E_2(C_i, n, m)$ は、記号の接続関係の類似度に関する評価関数であり、 $E_c(C_i, n, m)$ は、対応関係の整合性に関する評価関数であり、 $E_i(C_i, n, m)$ は記号の属性情報に関する類似度の評価関数であり、 $E_p(C_i, n, m)$ は、位置の類似度に関する評価関数である。また、 k_1 、 k_2 、 k_c 、 k_i 、 k_p は、それぞれの評価関数に対する重みづけ係数であり、対応付けのための

調整パラメータとなる。同様にCは、調整用の定数値である。以下、各評価関数について説明する。なお、以下の説明では、評価関数が小さいほど差異が少ない(類似度が高い)ように評価関数を定義している。

$$E(C_i, n, m) = k_1 \times E_1(C_i, n, m) + k_2 \times E_2(C_i, n, m) + k_c \times E_c(C_i, n, m) + k_i \times E_i(C_i, n, m) + k_p \times E_p(C_i, n, m) + C \times I(C_i, n, m) \dots (1)$$

(a) $E_1(C_i, n, m)$: 直接つながっている記号種別の類似性
これは、記号要素間の類似度を記号種別ごとに直接つながっている個数によって評価する評価関数である。

記号要素は入力側、出力側に複数のポートをもっており、記号種別によってはポートによって処理が異なるため、どのポートとつながっているかを区別する必要があるものがある一方、AND や OR の出力ポートのように、どれも同じ機能に対応し、区別する必要がないものもある。

そこで、区別すべきポートの種別をあらかじめ登録しておき、Fig.1 に示すように、区別したポートごとに、どの記号種別が、何個つながっているか集計しておく。図の例では、参照図面側の記号 n に対して、記号種別 k1 が 3 つ、記号種別 k2 が 3 つ、記号種別 k3 が 2 つつながっている。また、差分評価図面の記号 m に対して、記号種別 k1 が 3 つ、記号種別 k2 が 3 つ、記号種別 k3 が 2 つつながっている。

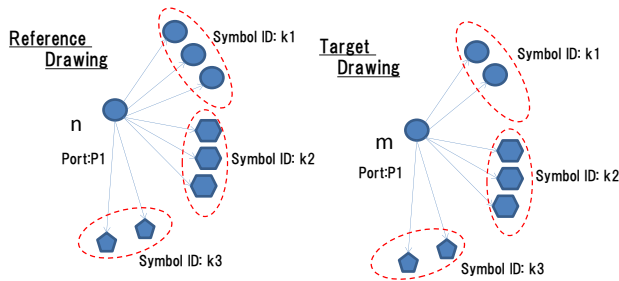


Fig.1 Primary Connectivity Evaluation

このとき、差分評価図の記号要素 m に接続している記号種別 k1 の 2 つ分は、参照図の n と一致していて、1 つ分は不一致と考えることができる。評価関数を、不一致の接続の数から一致した接続の数を引いた値として定義し、記号種別ごとのリンク数が異なることを考慮してリンク数で標準化すると、あるポート p に対する記号種別 k に関する接続の類似度 $E_1(C_i, n, m, p, k)$ は、式(2)で表される。ただし、 $Cr(n, k, p)$ は、参照図面で記号 n がポート p で記号種別 k とつながっている数、 $Cd(m, k, p)$ は、差分評価図面で記号 m がポート p で記号種別 k とつながっている数である。ただし、ここでいうポートとは、記号種別ごとに決めた区別する必要のあるポートで、AND や OR の出力ポートのように、相互に区別する必要のないポートについては一つのポートと見なして集計す

る。また、 $Min(a, b)$ 、 $Max(a, b)$ は、a, b のそれぞれ最小値、および最大値を表す。

$$E_1(C_i, n, m, p, k) = \{ 2 \times Min(Cr(n, k, p), Cd(m, k, p)) - Max(Cr(n, k, p), Cd(m, k, p)) \} / Cr(n, k, p) \dots (2)$$

これを、記号種別 k、ポート p で集計すると、対応関係 $\langle n, m \rangle$ に対する $E_1(C_i, n, m)$ を得る。

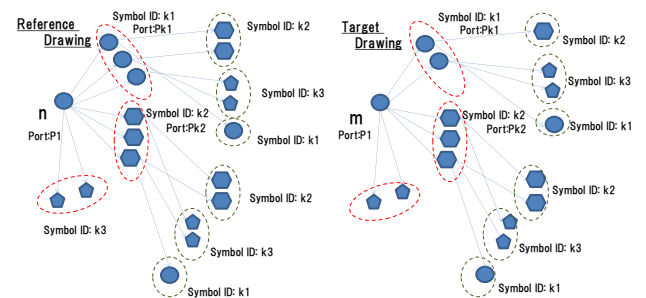
$$E_1(C_i, n, m) = (\sum_p \sum_k E_1(C_i, n, m, p, k)) \times I(C_i, n, m) \dots (3)$$

(b) $E_2(C_i, n, m)$: 2 次的につながっている記号種別の類似性

E_2 は、2 次的につながっている記号の類似性を評価する関数である。ただし、ロジックの構成という観点から、Fig.2 に示したように経由するポートの種別と、記号の種別は区別して集計する必要がある。図では、記号 n のポート P1 から、記号種別 k1 のポート種別 Pk1 を経由して接続している記号種別 k2 の記号数を $Cr(n, k1, p, k2, Pk1)$ とする。図の場合は 2 となる。また、差分評価図で、記号 m のポート P1 から記号種別 k1 のポート種別 Pk1 を経由して接続している記号種別 k2 の記号数を $Cd(m, k1, p, k2, Pk1)$ とする。図の場合は 1 となる。

Fig.2 Secondary Connectivity Evaluation

この集計に基づいて、ポート P1 から記号種別 k1 のポート種別 Pk1 を経由して接続している記号種別



k2 の記号数に基づく類似度 $E_2(C_i, n, m, p, k1, Pk1, k2)$ は、1 次接続の評価と同様、以下ようになる。

$$E_2(C_i, n, m, p, k1, Pk1, k2) = (2 \times Min(Cr(n, k1, p, k2, Pk1), Cd(m, k1, p, k2, Pk1)) - Max(Cr(n, k1, p, k2, Pk1), Cd(m, k1, p, k2, Pk1))) \dots (4)$$

これを、記号種別 k、ポート p、記号種別 k2、ポート Pk1 で集計すると E_2 は、以下ようになる。

$$E_2(C_i, n, m) = \left\{ \sum_p \sum_{k1} \sum_{Pk1} \sum_{k2} E_2(C_i, n, m, p, k1, Pk1, k2) \right\} \times I(C_i, n, m) \dots (5)$$

なお、Fig.2 では、出力側のポートに関する説明となっており、信号線の順方向に 2 次接続の記号まで

たどっているが、入力側のポートに対しては、信号線の出力側のポートを順次たどって、信号線と逆の方向に 2 次接続の記号数を評価する。

(c) $E_c(C_i, n, m)$: 対応関係の整合性

Fig. 3 のように、記号 n, m それぞれ同じポートに記号種別 k_3 の記号がひとつだけつながった記号 s と p とがあるとすると、このような関係にある $n, m, P1, s, k$ の組 $\langle n, m, P1, s, k \rangle$ の集合を **Cons** と定義する。このとき、記号 n と記号 m とが、対応関係にあるとすると、記号 s と記号 p が対応関係にあるのが整合的であり、また、逆に記号 n と記号 m とが対応関係にないとすると、記号 s と記号 p が対応関係にないのが

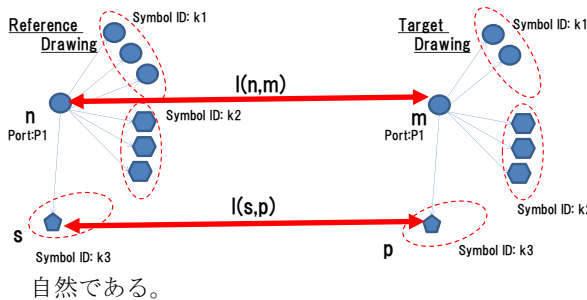


Fig.3 Primary Connectivity Evaluation

この問題に対応するため新たな変数 $C_f(n, m, P1, t1, t2)$ を導入する。 $C_f(i1, i2, P1, t1, t2)$ は、 $I(C_i, n, m)$ と $I(C_i, s, p)$ の差に対応するように、以下の式により定義する。

$$\begin{aligned} C_f(n, m, P1, t1, t2) &\geq I(C_i, m, n) - I(C_i, s, p) \\ C_f(n, m, P1, t1, t2) &\geq I(C_i, s, p) - I(C_i, m, n) \\ C_f(n, m, P1, t1, t2) &\geq 0 \end{aligned} \quad \dots (6)$$

$I(C_i, m, n)$ がともに、1 または 0 のとき、 $C_f(i1, i2, t1, t2)$ は 0 以上の任意の値をとる一方、どちらかが 1 でどちらかが 0 の時、 $C_f(i1, i2, t1, t2)$ は、1 以上の値をとらなければならない。

このとき、式(7)のように評価関数 $E_c(C_i, n, m)$ を定義すると、最適値を探す最小化の過程で、 $I(C_i, m, n)$ がともに、1 または 0 の場合が選ばれやすくなる。

$$\begin{aligned} E_c(C_i, n, m) &= \sum_{\langle n, m, P1, s, k \rangle \in \text{Cons}} C_f(n, m, P1, s, p) \end{aligned} \quad \dots (7)$$

Fig. 3 の記号種別 k_2 のように、同じ種類の記号種別が複数ある場合には、対応する記号の組合せが発生し、整合性の制約は複雑なものになる。このため、この評価関数は、それぞれの記号の特定のポートに、同じ記号種別が 1 つずつ、つながっている場合のみ追加することとした。

(d) $E_i(C_i, n, m)$: 記号属性の類似性

ロジック図 1 枚で比較する場合、外部への信号のとりあいを表す記号種別“CEDI”、“CEDO”は、外

部入出力参照信号 ID を属性としてもつ。したがって、これらの記号に対しては、同じ外部入出力参照信号 ID をもつものは対応している可能性が高いので、そのような関係のある組合せ $\langle n, m \rangle$ の集合を **Ca** とするとき、(3-2-6)のように、 $E_i(C_i, n, m)$ を与える。評価関数を最小にする最適化を行うため外部入出力参照信号 ID が同じ記号 n, m に対する変数 $I(C_i, n, m)$ の値が 1 になりやすくなる。

$$E_i(C_i, n, m) = \sum_{\langle n, m \rangle \in \text{Ca}} (-1) \times I(C_i, n, m) \quad \dots (8)$$

(e) $E_p(C_i, n, m)$: 図面中の位置記号種別の類似性

ロジック図の位置は、ロジックの表す処理内容に対しては、何の意味ももたないが、参照図からロジックを編集して差分評価図を作成することを考えると、同じ記号種別の記号に対して、位置がまったく同じなら(このような記号 n, m の組合せの集合を **Cp** とする)、対応する可能性は高くなる。式(9)のように $E_p(C_i, n, m)$ を定める。

$$E_p(C_i, n, m) = \sum_{\langle n, m \rangle \in \text{Cp}} (-1) \times I(C_i, n, m) \quad \dots (9)$$

(3) 制約条件

混合整数計画法のモデルとして、各変数の意味付けに対応した以下の制約条件を与えた。

- (a) 対応付け変数 $I(C_i, n, m)$ がバイナリ変数、つまり 0 または 1 の整数値をとること。
 (b) 参照図の記号 1 つに対応する差分評価図の記号の数は 1 以下であること。つまり、

$$\text{For } \forall n \quad \sum_m I(C_i, n, m) \leq 1 \quad \dots (10)$$

同様に差分評価図の記号 1 つに対応する参照図の記号の数は 1 以下であること。つまり、

$$\text{For } \forall m \quad \sum_n I(C_i, n, m) \leq 1 \quad \dots (11)$$

3. 処理結果と評価

3.1 評価図面と評価関数

前章による混合整数計画法によるモデル化にもとづき、記号対応付け機能を作成し、原因となるサンプル図面を参照図面として、これを変更した差分評価図面を作成し、手法を評価した。また、2.3 節で議論した各評価関数の重み係数を変更して、処理時間、処理結果を比較した。

(1) 差分評価図面

差分評価図としては、記号の種別の置き換え、記号の追加、及び記号の削除を行った以下の 3 種類のロジック図を作成して用いた。

(a) Variation-A: 記号種別の置き換え

Fig. 4. に示したように元の図面の記号識別番号 619、記号種別“AND”の記号を記号識別番号はそのままで、

記号種別を”OR”に変更した図面。

(b)Variation-B: 記号の置き換え

Fig. 5 に示したように元の図面の記号識別番号 617 と 619 の間に、記号種別”NOT”の記号を加えて信号線でつないだ図面。

(c)Variation-C: 記号の削除

Fig. 6 に示したように元の図面の記号識別番号 613 と 616 の間にあった、記号種別”OFD”の記号を削除して、記号種別 613 の記号と 616 の記号を直接でつないだ図面。

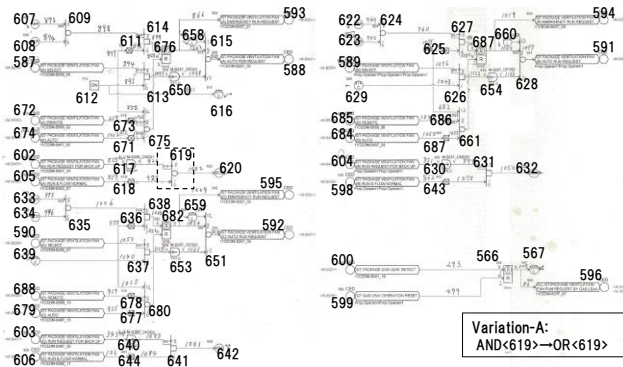


Fig. 4 Variation-A

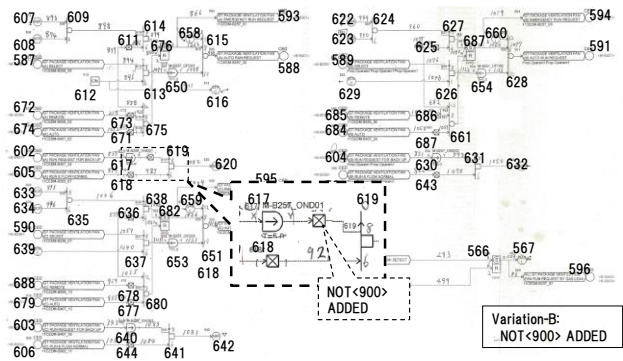


Fig. 5 Variation-B

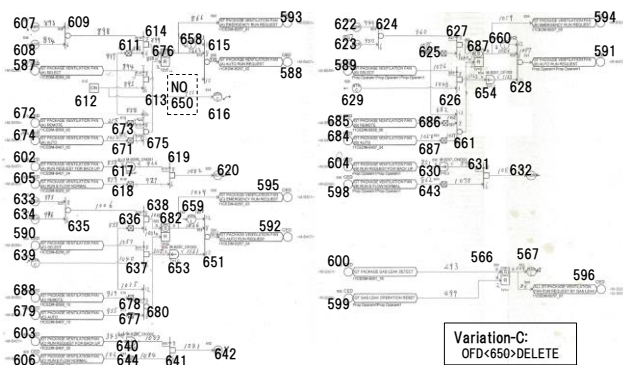


Fig. 6 Variation-C

(2) 評価パラメータの設定

式(1)に示した、評価関数の各パラメータを以下のように設定して比較した。なお、各評価関数に-1の定数を引い

ているのは、適度に多くの記号要素が照合するための設定値であり、事前の評価により定めた。

(a) Setting-1: 接続関係の類似度だけで評価

$$E(C_i, n, m) = 1.0 \times E_1(C_i, n, m) + 0.1 \times E_2(C_i, n, m) + 1.0 \times E_i(C_i, n, m) - 1.0 \quad \dots (12)$$

これは、個々の記号の接続関係だけで、どの程度の精度ができるかの確認のための設定である。2次接続の係数を、1次接続よりかなり小さくしたのは、直接つながっている記号種別の類似度を、より高く評価するという自然な考え方による。

(b) Setting-2: Setting-1 に位値の一致を追加。

$$E(C_i, n, m) = 1.0 \times E_1(C_i, n, m) + 0.1 \times E_2(C_i, n, m) + 1.0 \times E_i(C_i, n, m) + 0.1 \times E_p(C_i, n, m) - 1.0 \quad \dots (13)$$

位置が一致した場合の評価関数の係数を 0.01 と小さくしたのは、接続関係の評価を優先し、その評価が同じだった場合にのみ、位置の一致が効くようにするためである。

(c) Setting-3: Setting-2 に整合性の制約を追加

$$E(C_i, n, m) = 1.0 \times E_1(C_i, n, m) + 0.1 \times E_2(C_i, n, m) + 1.0 \times E_i(C_i, n, m) + 0.1 \times E_p(C_i, n, m) + 0.1 \times E_c(C_i, n, m) - 1.0 \quad \dots (14)$$

これは、外部入出力取り合い記号 (CEDI/CEDO) に対して、外部入出力参照信号 ID が一致しているものは一致するという、現実的な評価を加える事により、位置の評価関数のような、不確実な評価を入れなくても、整合性の制約を加えれば、一致する事を期待しての設定である。

3.2 処理結果

計算は、まず、Variation-A に対して、Setting-A, B, C の3通りで評価した。その結果を Tab. 2 および Fig. 7 に示す。

Tab. 2 に、3種類の図面のバリエーションに対して、3種類の評価関数の設定をして処理した結果を示す。用いた計算機の仕様は以下である。

- ・OS : Windows 7 64bit
- ・CPU スペック : Intel® Core™ i5-4670 CPU @ 3.6GHz
- ・メモリ容量 : 32GB

混合整数計画法のソルバは切除平面法^[2]を利用した分枝切除法の自作のソルバを用いた。

表中、全9通りの結果に対して、定式化の結果としてのバイナリ変数と実変数、及び制約条件数を示し、下段に記号の対応付けの結果を[誤って対応付けされた記号要素数/正しく対応付けされた記号要素数]の形式、および[誤って対応付けされずに残った記号要素数/正しく対応づけされずに残った記号の正解数付けされた記号要素数]で示すとともに、処理時間を記載している。ここで、対応付かなかった記号のうち不正解のものとは、本来、参照図のどれかの記号に対応づくと思われる記号をさす。なお、対応付けの正解、不正解は報告者が見て判断した。

また、もっとも正確な処理ができた、Setting-3 に対する、Variation-A,B,C の処理結果を図面上で、Fig.7, Fig.8, Fig.9 に示す。各図で、右側が参照図面、左側が差分評価図面、上半分が処理対象の図面をあらためて説明したもので、下半分が処理の結果対応付けができなかった記号(差分と判断された記号)を点線で囲って示したものである。以下に結果を要約する。

(a) 対応付けの正確性

Tab.2 を見ると、Setting-1 では、生成される記号のペアの正確性がかなり低く、Setting-2 でも、若干の誤りがあるのに対して、Setting-3 では、すべて正解のペアを抽出するとともに、差分の記号もすべて正解を出力している。

一般には、この対応付けの処理結果をもとに、記号の属性や信号線の差分をさらに抽出する処理があるため、システム全体の信頼性と効率のために、記号の対応付けの処理には高い正確性が要求される。Tab.2 の結果から、このためには、整合性の制約と図面の入出力取り合い記号に対する外部参照信号 ID の一致に関する評価関数を組み込む必要があることがわかる。その一方で記号の位置に関する評価は必ずしも必要でないことがわかる。

(b) 処理時間

Tab.2 の結果から、Setting-1,2 に対して、Setting-3 では、100 倍以上の処理時間がかかる場合があることがわかる。これは、整合性の制約条件を導入したため、変数と制約式が増えて、1 回の線形計画法の処理では、バイナリ変数の値が 0/1 値にならず、分岐切除法による探索が行われるためである。一方、Setting-1,2 の定式化では、1 回の線形計画法の処理で、バイナリ変数の値が必ず 0/1 値になるため、非常に短い処理時間で処理できる。

Tab.2 Results of Evaluation

評価ケース	Evaluation Parameters		
	Setting-1	Setting-2	Setting-3
	$K1=1.0, k2=0.1, kc=0.0$ $Kp=0.0, ki=1.0, C=-1.0$	$K1=1.0, k2=0.1, kc=0.0$ $Kp=0.1, ki=1.0, C=-1.0$	$K1=1.0, k2=0.0, kc=0.1$ $Kp=0.1, ki=1.0, C=-1.0$
Variation-A "AND(619)" を "OR(619)" に置き換え	Binary Variables:597 Real Variables:0 Constraints:114	Binary Variables:597 Real Variables:0 Constraints:114	Binary Variables:597 Real Variables:329 Constraints:772
不正解要素ペア/正解要素ペア 差分要素数(不正解/正解)* 処理時間(秒)	Pair:23/56 Non Pairs: 2/1 TIME:0.4	Pair:20/59 Non Pairs: 2/1 TIME:0.6	Pair:0/79 Non Pairs: 1/1 TIME:196.2
Variation-B "NOT(900)" 追加	Binary Variables:601 Real Variables:0 Constraints:115	Binary Variables:601 Real Variables:0 Constraints:115	Binary Variables:601 Real Variables:347 Constraints:809
不正解要素ペア/正解要素ペア 差分要素数(不正解/正解)* 処理時間(秒)	Pair:28/53 Non Pairs: 0/1 TIME:0.4	Pair:28/53 Non Pairs: 2/1 TIME:0.3	Pair:0/81 Non Pairs:0/1 TIME:244.5
Variation-C "OFD(650)" 削除	Binary Variables:594 Real Variables:0 Constraints:113	Binary Variables:594 Real Variables:0 Constraints:113	Binary Variables:594 Real Variables:335 Constraints:783
不正解要素ペア/正解要素ペア 差分要素数(不正解/正解)* 処理時間(秒)	Pair:35/45 Non Pairs: 0/1 TIME:0.6	Pair:32/48 Non Pairs: 0/1 TIME:0.3	Pair:0/80 Non Pairs:0/1 TIME:200.8

*差分図に対して評価

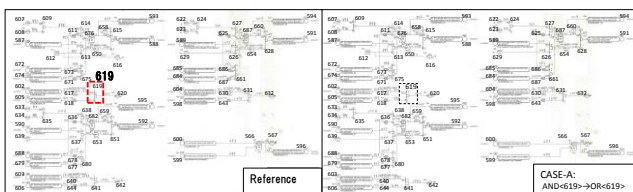


Fig. 7 Result (Variation-A, Setting-3)



Fig. 8 Result (Variation-B, Setting-3)

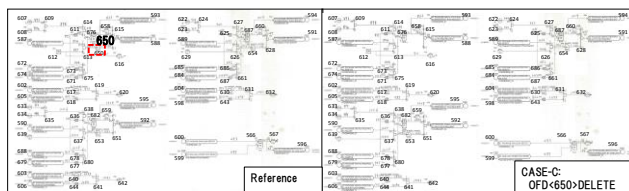


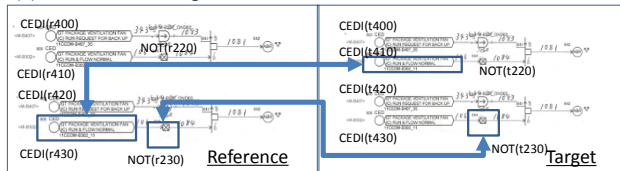
Fig. 9 Result (Variation-C, Setting-3)

4. 考察

4.1 整合性の評価関数 Ec の必要性

手法は、個々の記号に接続している記号の種類に着目して対応関係を評価している、しかし、Fig.10のように、同じパターンが2つある場合、(a)は非整合で、(b)は間違いかもかもしれないが、整合性はあることがわかる。これは、CEID(r440)と NOT(r230)とが信号線でつながっているにもかかわらず、(a)の場合は、信号線でつながっていない CEDI(t410)と NOT(t230)とにそれぞれ対応づけられているからである。個別の記号の接続状態をいくら精度よく評価しても、このような事がおきるのは容易にわかる(この例でも、NOT(220)と NOT(230)のつながり方は全く同じである)。Ec は、整合性を満たす対応づけが、より選ばれやすくするための評価関数である。

(a) Inconsistent Pairing



(b) Consistent Pairing

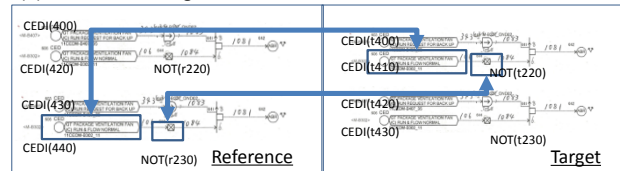


Fig.10 Consistency

4.2 差分の定義/正解の曖昧性

自分自身が同じものでも、まわりの接続関係が変わった場合に、参照図面のもとの記号と対応づけるべきかどうかは、哲学的な議論になる。しかし、ここでは、工学的に、用途によって差分の定義は異なると割り切り、評価関数の係数を目的によって調整する枠組みを与えることとした。

特に、ロジック図の差分評価の目的には、Tab.2 に示すように、整合性の評価関数と、取り合い参照信号の同一性を反映した評価関数を用いることにより、正確な差分評価ができると見込まれる。

4.3 手法の拡張性従来研究との関係

本研究に近い研究としては、グラフのパターン照合がある。大規模なグラフのパターン照合は、近年、盛んになったビッグデータ解析の主要な技術的チャレンジのひとつであり、数多くの、提案、研究が行われている^{[3][4][5]}。本研究は、そのひとつとして位置づけられるが、設計図面を対象としているため、違いを確実に抽出することと、用途によって差分の定義を変更できる柔軟性が求められる。大規模なデータの高速照合という観点からは、近似的な方法^[6]など、より優れたアプローチもあるが、このような柔軟性と正確性を意図したところに本研究の特徴がある。

5. まとめ

混合整数計画法に基づいた属性つき属性グラフの差分評価手法を提案した。3種類のロジック図面を例に評価し、正確な差分評価のためには、要素間のリンク関係にもとづく整合性に関する変数と制約条件の導入が必要であり、その場合、処理時間は、この制約条件の導入をいれない場合の100倍程度かかることを示した。

参考文献

- [1] ジョゼフ・シエムラ著, 長瀬嘉秀監訳, "独習 UML", 翔泳社, 2000年.
- [2] 久保他監修: "応用数理計画ハンドブック", 朝倉書店, 2002年.
- [3] 松田ほか, "一般グラフ構造データに対する Graph-Based Induction とその応用", 人工知能学論文誌 16 巻, 第 4 号 A, pp262-374, 2001年.
- [4] 和佐ほか, "半構造データに対する効率良い近接パターン照合アルゴリズム", DEIM Forum, 2012, B6-6, 2012.
- [5] 塩川ほか, "構造的類似度に基づくグラフクラスタリングの高速化", DEIM Forum 2014
- [6] 大原剛三, 鷺尾隆, "大規模グラフにおける近似部分グラフ探索手法", The 21st Annual Conference of the Japanese Society for Artificial Intelligence, 2007, 3F7-1, 2007. 人工知能学論文誌 16 巻, 第 4 号 A, pp262-374, 2001年.

† (株)日立製作所 Hitachi Ltd.

‡ (株)三菱日立パワーシステムズ MHPS

¶ (株)日立パワーソリューションズ Hitachi Power Solutions Co. Ltd.