

分岐・共通部優先ヒューリスティック探索による サイフォン・トラップ構造検出の効率化

Efficiency Improvement of Siphon/Trap Structure Detection using Branch/Common-Part-First Heuristic Search

南 史弥† 張江 洋次郎†† 和崎 克己††
Fumiya Minami Yojiro Harie Katsumi Wasaki

1 はじめに

ペトリネットは、事象発生の並列性、非同期性、非決定性を有する離散事象システムの振る舞いを表す数学モデルであり、グラフィックツール、シミュレーションツール、および数学的方法論としての側面を持つ [1][2]。ペトリネットの構造的性質の一つにサイフォン・トラップ構造がある。これらは、ペトリネット内のトークンの増減に影響し、重要な性質である。筆者らが開発している HiPS[3] にはサイフォン構造判定器が実装されているが、現在実装されているアルゴリズムは大規模ネットワークを対象とした場合探索コストが大幅にかかる。そこで本研究では、ヒューリスティック探索手法を用いて従来法と比べ探索コストを抑えてサイフォン・トラップ構造を検出することで、特に極小構造から優先的に検出する工夫を提案する。

2 サイフォントラップ構造

2.1 正規ペトリネット

ペトリネットは、有向グラフ (directed graph) の一つであり、初期マーキング (initial marking) M_0 と呼ばれる初期状態をもつ。ペトリネットを作る基本的なグラフ N は、重みつき有向 2 部グラフ (weighted directed bipartite graph) であって、プレースおよびトランジションと呼ばれる 2 種類のノードからなる。ここで、アークはプレースからトランジションに接続するもの、あるいはトランジションからプレースへのもののいずれかである。グラフ的な表現では、プレースは丸、トランジションは棒または箱で描かれる。アークには重み (正整数) が付記され、重みが k であるアークは、 k 重アーク (多重アーク) と呼ばれ、 k 本の平行なアークの集合と解釈できる。

2.2 サイフォン・トラップ

正規ネットワーク N において、空でないプレースの部分集合 S は、 $\bullet S \subseteq S^\bullet$ であるならば、すなわち S 内に出カプレースを持つすべてのトランジションが入カプレースを S 内にもつならば、サイフォン (siphon) と呼ばれる。空でないプレースの部分集合 Q は、 $Q^\bullet \subseteq \bullet Q$ であるならば、すなわち Q 内に入カプレースをもつすべてのトランジションが出カプレースを Q 内にもつならば、トラップ (trap) と呼ばれる。サイフォン (トラップ) は、他のどのサイフォン (トラップ) もその部分集合となっ

ていなければ、極小であると呼ばれる。

2.3 従来法

現在、HiPS ツールの、FC/AC Liveness Safeness Check 検証器に Siphon 検出ツールが存在する。Algorithm1 を以下に示す。

Algorithm 1 Detect-Siphon (Conventional)

```

1: for place in page.Places do
2:   pset.Add(place)
3:   SiphonGenerate.Add(pset)
4:   pset.clear
5: end for
6: for pset in SiphonGenerate do
7:   if pset is Siphon then
8:     addSiphon.Add(pset)
9:   else if pset.PlaceToGenerateSiphon.Count == 0 then
10:    CannotGenerated.Add(pset)
11:   else
12:     for Placeset in pset.PlaceToGenerateSiphon do
13:       pset.Addrange(Placeset)
14:       newpss.Add(pset)
15:     end for
16:   end if
17: end for
18: SiphonGenerate ← newpss

```

任意のプレース集合に対して、サイフォン構造を成すように順次プレースを追加していくプログラムであるが、これはプレースの組み合わせ毎にサイフォン判定を行っている為、計算量は $O(2^n)$ であり非常に効率が悪い。

3 分岐・共通部に着目した優先探索

3.1 ヒューリスティック手法の概要

本研究では、ヒューリスティックアルゴリズムを用いて、ネットワーク構造の特徴点を検知し、全ての極小構造を列挙する手法を提案する。具体的には、複数の構造に共通する路から優先的に探索し、部分サーキットを取得する。解析の流れとしては、1) 強連結成分分解、2) 共通路、分岐点検出、3) 構造判定の順で処理を行う。

- 1) 強連結成分分解を行い、各成分に対して互いに到達可能であることを示す (頂点を V , エッジを E とすれば計算量は $O(V + E)$)。)
- 2) 強連結各成分に対して、共通路、分岐点の発見ないし部分サーキットを発見する (合流トランジションの数を T とすると、計算量は $O(TE)$)。)
- 3) 全ての強連結成分のサーキットを発見すれば、それが構造を成しているかどうかを判定する (計算量は $O(E)$)。)

† 信州大学大学院総合理工学研究科, Graduate School of Science and Technology, Shinshu University

†† 信州大学大学院総合工学系研究科, Interdisciplinary Graduate School of Science and Technology, Shinshu University

3.2 強連結成分分解

有効グラフにおいて、互いに行き来することができるようにそれぞれ頂点グループを分解することを強連結成分分解という。アルゴリズムには、正向きネット、アークの向きを逆にした逆向きネットの順に深さ優先探索を用いる。正向き深さ優先探索では、最深部まで到達すれば、到達した順に小さい数字からラベル付けを行う。逆向き深さ優先探索では、ラベルの数字が大きいものから探索をスタートし、辿ったことのある頂点に到達すれば強連結成分として保存する。極小サイフォン構造は、シンクプレース、ソースプレースを除いて強連結上で構成されることが多いため、強連結成分に注目しながら探索を行うことは有意である。

3.3 部分強連結

強連結成分内において、その部分集合で強連結を形成できる場合、それらは部分強連結と呼ばれる。強連結成分分解の逆向き深さ優先探索 (DFS) の際、ある程度の部分強連結を発見することで、共通路分岐点による探索数を減らす。

3.4 共通路, 分岐点

強連結成分ネット分解が完了した後、共通路 (点), 分岐点を探す。方法としては、流入アークが複数あるトランジション群を探し、見つければ共通路の合流点を始点として流出アークが複数ある頂点 (分岐点) まで前向き探索を行う。この合流点から分岐点までの間を共通路と呼ぶ。ネット内のサイフォン・トラップ構造の数は、トランジションの入出力アークの数に依存する。サイフォン構造では、任意のプレース集合において、それらプレースの入力アークにつながるトランジション集合を出力アークにつながっているトランジションで包含すれば成立する。つまりトランジションにつながる流入アークが多いほど、サイフォン構造が流入アークの数だけ増え、一つの共通路探索で多くのサイフォン構造候補を検出できる。

3.5 サイフォン・トラップ候補リスト

共通路, 分岐点を見つけ次第, 分岐点から深さ優先探索を行い, 共通路入口まで到達したら候補リストとして保存する。以上, 3 ステップにわたる提案手法を Algorithm2 に示す。

Algorithm 2 Detect-Siphon (Proposed)

```

1: ScListElement ← ScC(Graph)
2: for ScList.Trans in ScListElement do
3:   for Trans in ScList.Trans do
4:     if Trans.SourceArcs.Count ≥ 2 then
5:       for place in Trans.SourceArcs do
6:         EntrancePlc.Add(place)
7:         SiphonList ← CommonPlaceList
8:       end for
9:       EntranceTrans ← Trans
10:      SiphonList ← DFS(Branch, visited)
11:    end if
12:  end for
13: end for

```

4 優先探索の例

優先探索の処理ステップを説明する。図 1 中において, (1) の点枠が強連結成分, (2) の点線部分は共通路, 分岐点を示す。図の強連結成分では共通路が二つ存在するため, 共通路・分岐点処理は 2 回行う。分岐点 (共通路から出ていく部分) から, 同じ共通路に入る部分までを最深部とした DFS を実行し, それまで探索してきたプレースを部分サーキットとして保存する。分岐点からすべてのアークを探索終了し次第, 次の共通路へ移り探索を行う。強連結成分内の共通路をすべて探索終了すれば, 次の強連結成分へ移行する。強連結成分を一通り探索後, 保存した部分サーキットに対し構造判定を行ってサイフォン構造かどうかの判断を行う。

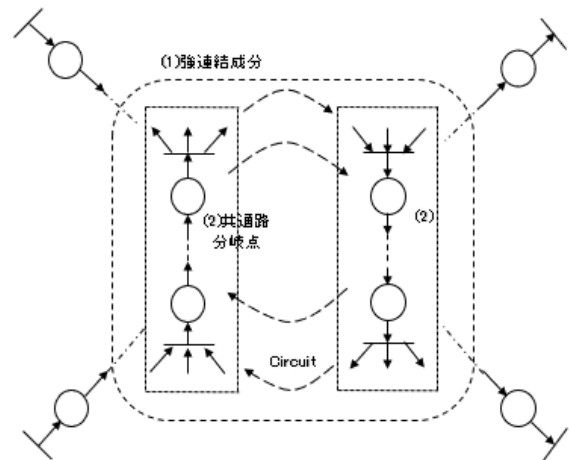


図 1 共通路・分岐点の存在するネット

5 まとめと今後の課題

極小サイフォンをヒューリスティックを用いて, 高速に取り出す手法を提案した。しかし, 今回のアルゴリズムは共通路の数が少なく, さらに路の長さが長いようなネットに対しては有効だが, MCC2015 の SquareGrid[4] のような, ネット内すべてのトランジションが路の長さ 1 の共通点である場合は探索効率が悪くなる。これらのネットに対して高速に取り出す方法として, 部分強連結の発見である。3.3 で述べたように, 強連結成分分解の段階でこれらを見出すことで, 共通路分岐点による探索数を減らす。これらの課題を修正し次第, 従来法との性能比較を行う。比較対象としては, (1) 検出時間 (2) 従来法で取得したサイフォン構造数に対して, 現手法での構造の取得率の検証を行う。

参考文献

- [1] T. Murata : "Petri Nets: Properties, Analysis and Applications", Proc. of the IEEE, 77(4), 1989.
- [2] J.L. ピーターソン: "ベトリネット入門 情報システムのモデル化", 共立出版, 1984.
- [3] HiPS : Hierarchical Petri net Simulator, <https://sourceforge.net/projects/hips-tools/>
- [4] MCC2015 SquareGrid : <https://mcc.lip6.fr/pdf/SquareGrid-form.pdf>