

プログラムテストに用いるパスジェネレータへの 一考察

柳沢 隆夫
芝浦工大

本論文は、プログラムの自動的なテストパス発生に於いて生じる、種々の問題
のためのアルゴリズムを考慮している。
これらの問題は、有向グラフの最も多くの指定された辺を含むパス集合を決定
することである。

A study of the Path Generator for Program Testing

Takao Yanagisawa
Shibaura Institute of Technology
Omiya-shi, Saitama-ken, Japan

In this paper we consider algorithm for problems that arise
in automatic test path generation for program: the problem
of determining a path which contain the most specified edges
of a directed graph.

1. はじめに

本研究は、プログラムテストに用いるパスジェネレータをグラフ理論を応用して解いたときに生じた種々の問題について検討している。

これらの問題は、プログラムグラフの指定された辺を最多に含む、スタート節からエンド節までのパスの中で、最短なものを見つけることである。

プログラムテスト法として、まず、プログラムグラフのテストパスを決定し次に、これを通る入力データを算出して行うものがある。^{[1][2]} 図-1

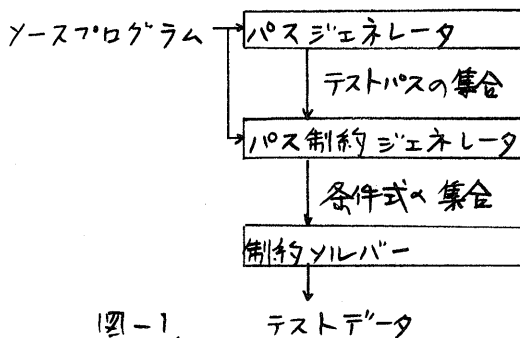


図-1. テストデータ

しかし、この手法は、テストパスが不実行パスであったときは破局となる。この問題に対し、Kundeは^[3] プログラムの全ての分枝をテストする問題を例にとり、未だテストされていない入力変数の領域より、次のテストデータを決定する手法を提案したが、この手法も実行可能なパスが多く含まれているプログラムの場合は容易には目的の分枝テストを満足させることは出来ない。

本研究はテスト法として、まず、スタートと各ディビジョンの直ぐ後にカウンターを挿入したプログラムに、いくつかの入力データをランダムに入力してテストを行い、次に残りのカウンター値が0を示している未だ未テストの部分にテストを振り向けるテスト法の問題を扱う。^[4]

この問題は、プログラムグラフの指定された辺を通るスタートパスを求めるのと等価となる。

本研究は、最短な未テスト辺を最多に含むテストパスをグラフ理論を応用して導出するときに生じた種々の問題を検討している。

未テスト辺の最多なパスを次々と求めてテストを行う手法は、テストの達成目標が100%未満で与えられている場合は有効である。

2. 未テスト辺が最多で最短なスタートパスの導出法について(1)

2.1 有向サイクルを含まないプログラムグラフの場合

この手法は、未テスト辺に大きな値、例えば100を割り当て、その他の辺に1を割り当て、SからTへの最長パスを求めることにより導出する。これにより求められる理由は、SからTへのパスの中で、最長のものの中に最多な未テスト辺が含まれていること、未テスト辺の最多のパスの中で、-1の辺を1番少くもつもの、即ち最短なパスは最長のパスとなるからである。

2.2 有向サイクルを含むのを許したプログラムグラフの場合

最長パスを求めることにより求める手法は、サイクルを廻る度に長さが増大するためここでは適用出来ない。

次に、いくつかの導出法を述べるが、それぞれの方法の効率性と適用条件についても述べて行く。まず、オ1案として既存のアルゴリズムをフルに利用して行うものが考えられる。図-2 この手法は、ハミルトンパスの最短なものを見つけるアルゴリズムを含んでおり、強連結成分内の未テスト辺の数が多きとき(例えば10以上)は余り効率の良い手法は見い出されていない。

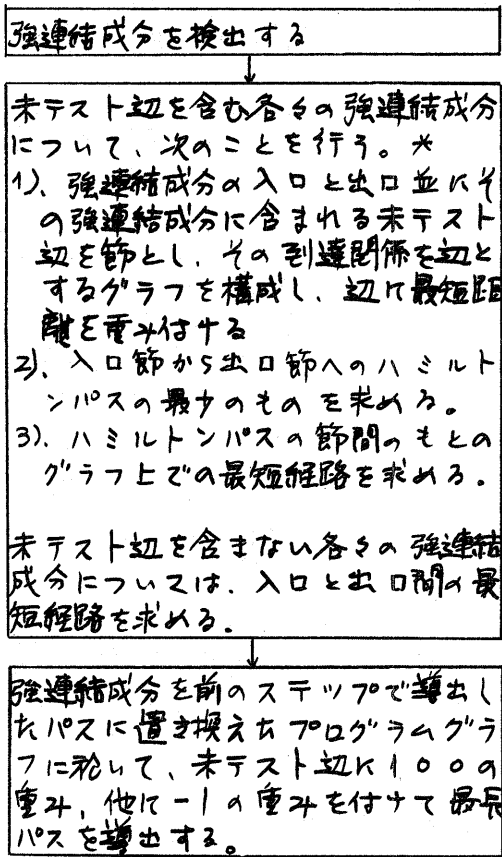


図-2

又、プログラムが構造化されていないとき、入口と出口の組合せの対により未テスト辺を含む最短なパスの長さが異なるため、(図-3) 入口と出口の組合せに対してその手続きを行う必要がある。しかし、この手法は、未テスト辺は本来数が少ないことを考慮すると、実用上は有用な手法と思われ。

ところで、もし、強連結成分内に未テスト辺が直列に連続的に生じている場合は、図-2の手法の*の部分を図-4に変えることにより、ハミルトンパスを求めずに簡単に求める手法が考えられる。

強連結成分内の未テスト辺の連続性のチェックは、強連結成分の検出の際に

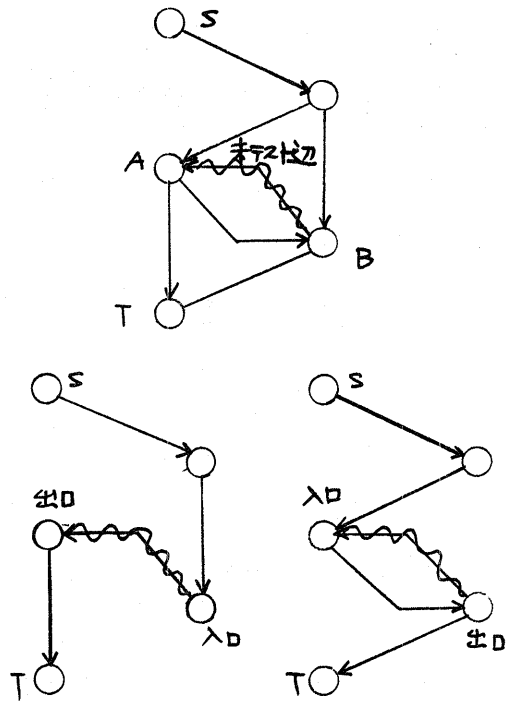


図-3

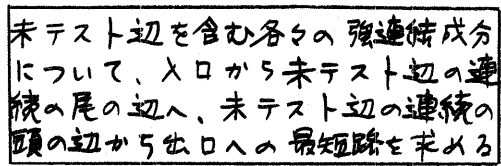


図-4

未テスト辺を記憶し、グラフの隣接行列を用いて調べることにより、チェックが可能となる。

このため、このテスト法の適用法は、いくつものテストの過程で、上記の条件が満足されることにより適用すればよいことになる。

3. 未テスト辺が最多で最短なS-Tパスの導出法について(2)

3.1 有向サイクルを含まないプログラムグラフの場合

次に、必ずしも最短ではないが、未テスト辺が最多なテストパスの効率的な導出法について述べる。

この方法は、図-5となる。

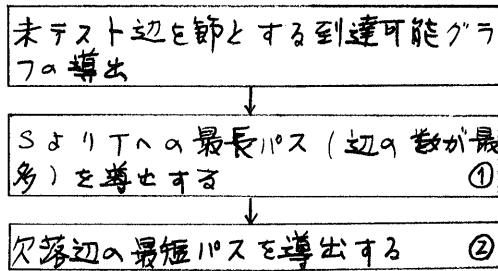


図-5

これにより求められる理由は、未テスト辺からなる到達可能グラフのパスはもとのグラフの未テスト辺を含むパスと、未テスト辺が同様な1対1対応が存在しているからである。

ここで、①は、ハットロッドによる、指定された節を通るS-Tパスの導出法に於ける節間の経路の導出法を分析して応用したものである。

②はラベリング法の他に、図-6のように未テスト辺に下限1を、全ての辺にコスト1を割り当てたグラフに最少コストフローを導出し、最短パスの全ての欠落辺を同時に求めることを考えられる。

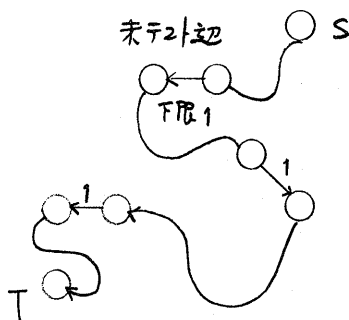


図-6

3.2 有向サイクルを含むものを許したプログラムグラフの場合

この導出法は、図-7となる。

この方法は、グラフの未テスト辺間の到達関係を保存しながら有向サイクル

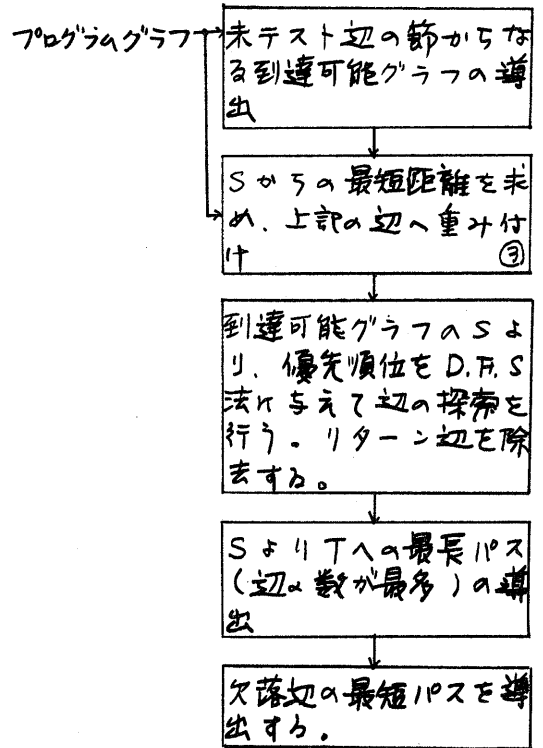


図-7

を削除し、最長パス導出法を導入して未テスト辺が最多で最短なパスを求めようものである。

③のステップは、例えば、図-8のようなグラフの全てが未テスト辺と仮定

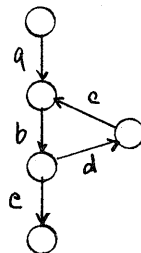


図-8

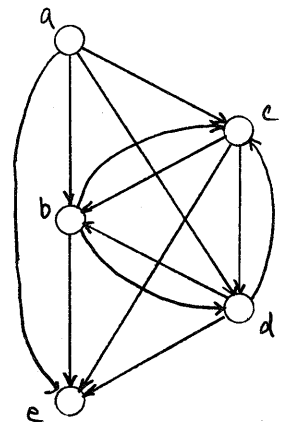


図-9

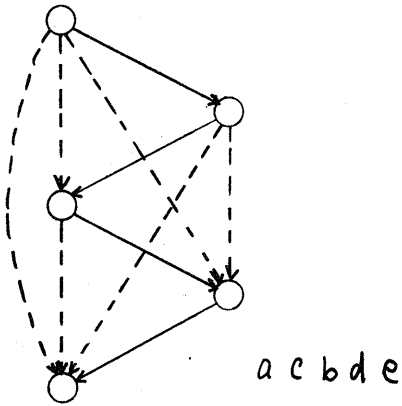


図-10

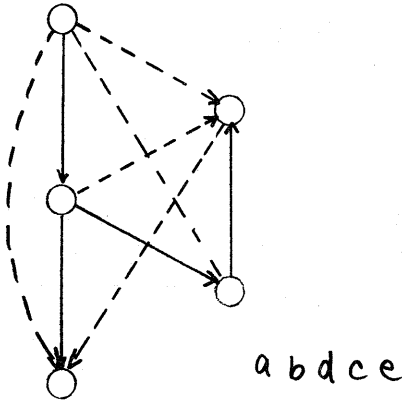


図-11

したとき、到達可能グラフ(図-9)にD.F.S法を適用して、図-10とする。最長パスはacbdeを通るパスとなって、最多なパスとはなるものの、そこで最短なパスを導出しない。このため、D.F.Sの辺の探索順位をSからの距離の小さいものとするものである。

優先順位に従うと、例えば、図-11となって、未テスト辺の最多な最短なパスの導出がされる。

この方法は、強連結成分内の未テスト辺が連続的な場合、例えば、図-12は適用可能であるが、それ以外の場合、

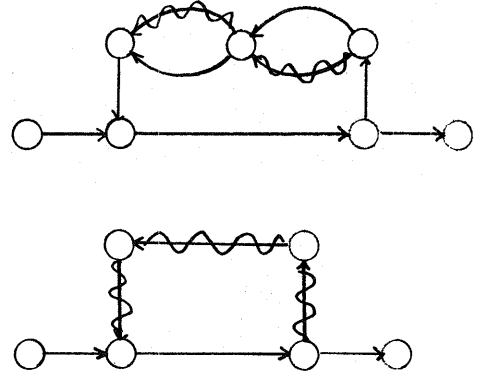


図-12

例えば図-13となる場合があり、

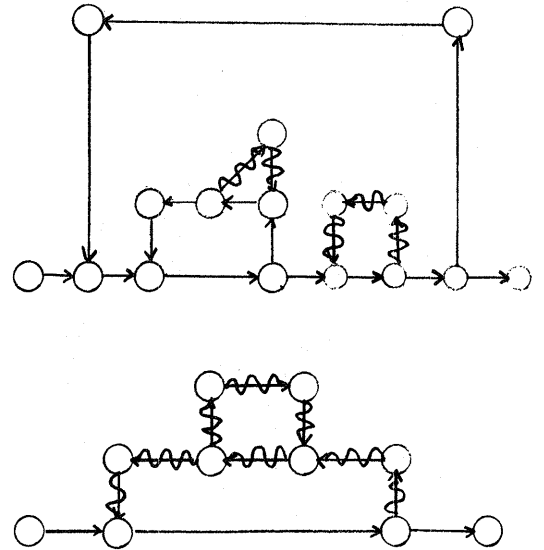


図-13

最短なパスの導出は保証されない。

4. 例題

図-15は、ランダムテストの結果、カウンタ①, ②, ③, ④の値が0のとき最短な未テスト辺を最多に含むテストパスは点線で表示されることを示している。^[6]

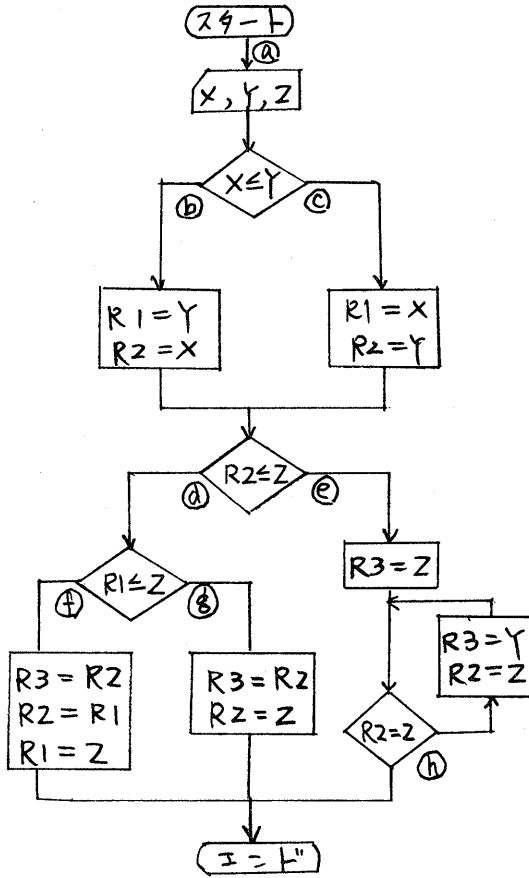


図-14

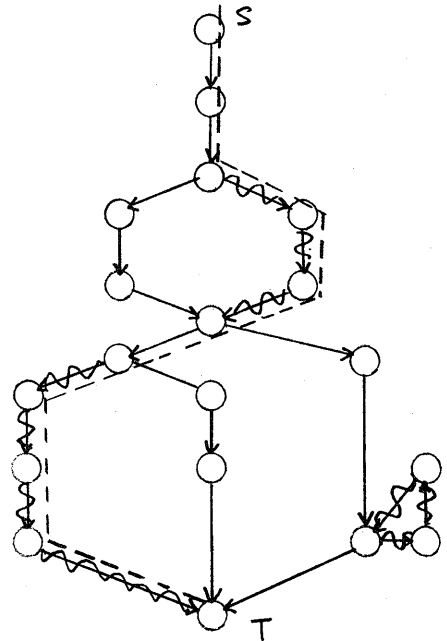


図-15

図-16は、グラフに有向サイクルが含まれないときの、未テスト辺が最多で最短なテストパスを示している。

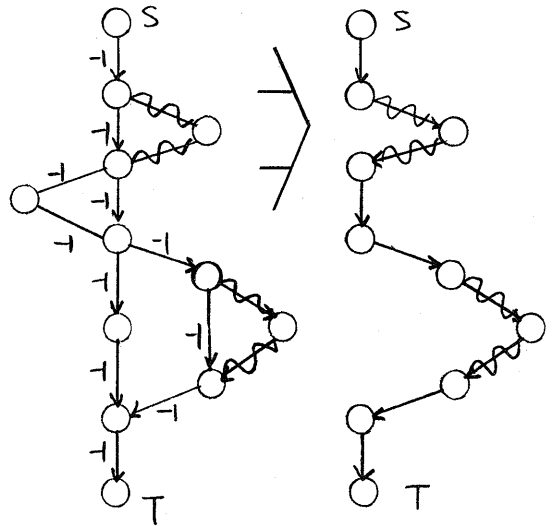


図-16

5. 参考文献

- [1] 柳沢隆夫, プログラムテストに用いるパスジェネレータへのグラフ理論の応用(オ1報), 茨浦工大研究報告理工系編, 30巻, オ1号, 1986.
- [2] 柳沢隆夫, パスカバー法によるプログラムテストデータの自動生成法に関して, 茨浦工大研究報告理工系編, 30巻, オ2号, 1986.
- [3] R. E. Prater, Theory of Program Testing - An Overview, THE BELL SYSTEM TECHNICAL JOURNAL Vol. 62, No. 10, Part 2, pp. 3073-3105, 1986.
- [4] 柳沢隆夫, プログラムテストに用いるパスジェネレータへの考察, 情報処理学会オ34回全国大会, 48-7
- [5] Harold N Gabow, On Two Problems in the Generation of Program Test Paths, IEEE TRANSACTION ON SOFTWARE ENGINEERING VOL SE-2, NO 2, 1976
- [6] 柳沢隆夫, プログラムテストに用いるパスジェネレータに関する研究, オ26回計測自動制御学会学術講演会(SICE'87)へ発表予定