

形式仕様に基づく ATM システムにおける テストケース自動生成への GA の適用

杉原 拓[†] 劉 少英^{††} 佐藤 裕二[‡]

法政大学情報科学部^{††‡}

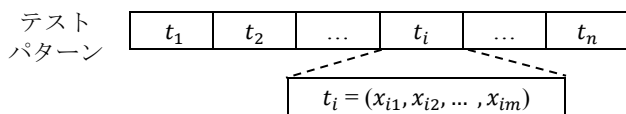
1. まえがき

形式仕様に基づいたソフトウェア開発において、テストケースの自動生成に関しては既にいくつかの手法が提案されている。しかし、これまでに提案された手法は仕様が正しく実現できていることを確認するためのものがほとんどであり、予想外のパスを含むプログラムの全てのパスを通過するテストケース自動生成の問題に関しては未解決である。全てのパスを通過するテストケース自動生成の研究として確率的探索手法(“Vibration method”)により、従来のテストケース生成手法 Pairwise 法よりもパス網羅率の精度を向上させることに成功した例が報告されている[1]。GA を用いたテストケース生成手法では、特定のパスを通過するテストケース生成するために標準的 GA を用いた例が報告されている[2]。本稿では、全てのパスを通過するテストケース自動生成手法として、新たな GA の適用方法を提案することで、従来手法に比べてプログラムのパスの網羅率を大きく向上できる可能性を示す。

2. テストケース自動生成のための GA の提案

2.1. 染色体の定義

テストケースを要素とした一次元染色体を定義する。染色体の構成を図 1 に示す。テストケースは全ての入力変数を含み、染色体の定義長はプログラムのパスの総数に一致させる。パスの総数が不明の場合は、冗長性を持たせて少し長めの定義長とする。また、テストケースの一部が判明している場合、遺伝子の一部に判明しているテストケースを前提知識として与えることが可能である。



t : テストケース, n : パス総数
 x : 入力変数, m : 入力変数の数

図 1 染色体の構成

2.2. 評価値

最適解は全てのパスを網羅するテストパターンとなる。テストケースを入力することでプログラムのパスを通過させる。テストケースがプログラムのパスを通過したとき、そのテストケースがそれ以前に通過しているパスの

集合 $P = \{p_1, p_2, \dots, p_k\}$ の各要素との比較を行う。その結果、新たなパスと判明した場合そのパスを p_{k+1} とし、通過したパスの集合に加える。プログラムのパスの総数に対する通過したパスの数の割合をテストパターンの評価値として与え、評価値が 1 となったときに全てのパスを網羅するテストパターンを生成したものとし、処理を終了する。パス総数が不明の場合は、通過したパスの数を評価値として与え、最大評価値が一定の世代数を経ても更新されない場合にその評価値をパス総数と判断し、処理を終了する。

2.3. 遺伝的操作

2.3.1. 交叉

テストケース生成に適した交叉手法として、二点交叉、一様交叉の 2 つの手法について予備実験を行った。その結果、一様交叉を用いた手法において、安定した結果を得ることができたため、一様交叉を用いる。

2.3.2. 突然変異

突然変異は、テストケース $t_i(x_{i1}, x_{i2}, \dots, x_{im})$ を n 次元ベクトルと捉え、変異ベクトルを付加することで、新たなベクトルを生成して突然変異後のテストケースとする。入力変数が 2 つのテストケース $t_i(x_i, y_i)$ の場合の突然変異例を図 2 に示す。

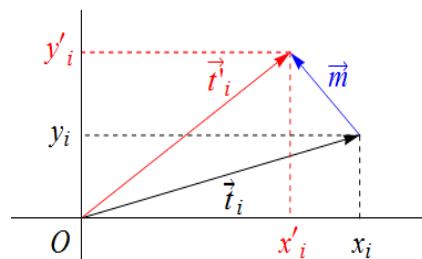


図 2 テストケース $t_i(x_i, y_i)$ の場合の突然変異例

3. 評価実験

提案する手法によるテストケース生成の評価実験として、人工的なプログラムを用いた予備実験を行い本手法によるテストケース生成の妥当性を示した[3][4]。一方、前述した関連研究である Vibration Method と Pairwise 法では、SOFL に基づいて設計された ATM システムが実験に用いられているが、プログラムが公開されていない。そこで、本研究においても同様に SOFL に基づき ATM システムを設計し、類似したシステムを用いて関連研究との比較実験を行う。

3.1. ATM システム概要

ATM システムの主な操作(service)は、口座開設(open account), 引出し(withdraw), 振替(transfer), 預入(deposit), 残高照会(show balance), 暗証番号変更(change password) とする。システムは入力された操作を受け取ると、入力

Applying GA for Automatically Generating Test Cases on ATM System Based on Formal Specification

[†] Taku Sugihara Hosei University

^{††} Shaoying Liu Hosei University

[‡] Yuji Sato Hosei University

に応じた出力をする。顧客番号と暗証番号が適合しない場合など、各操作において適さない入力が行われた場合にはその旨のエラーメッセージを出力する。

3.2. 実験方法

仕様通りの動作をするパス総数が既知の場合を想定した実験 1, 仕様外の動作や仕様と異なる動作を行うバグを挿入した, パス総数が未知の場合を想定した実験 2 の 2 種類について実験を行う。実験 2 に挿入したバグは, ATM システムに用いているグローバル変数がある値をとった際に仕様外の動作をするものである。また, 実験 2 における染色体の遺伝子長は, パスの総数が不明であるため, バグを挿入する前のパス総数の 2 倍とした。実験のプログラムのパラメータを表 1 に示す。

表 1 実験 1, 実験 2 プログラムのパラメータ

	実験 1	実験 2
個体数	250	500
遺伝子長	32	64
交叉率	0.9	0.8
突然変異率	0.016	0.011
テストケース	$t_i(\text{service}, \text{service})$ に適した入力変数)	
選択方式	トーナメント選択	
交叉方式	一様交叉	
パスの総数	32	不明

3.3. 実験結果

実験 1, 実験 2 の結果を表 2, 表 3 に示す。また, 評価値を縦軸に, 世代数を横軸にとった GA の学習曲線のグラフを図 3, 図 4 に示す。パス総数が既知の場合を想定した実験 3 では網羅率 100% のテストパターンの生成を実現した。パス総数が未知の場合を想定した実験 4 では, 50 回の試行のうち評価値はすべて 40 となった。パス数を確認すると 40 のパスが存在していることがわかり, 実験 4 においても網羅率 100% のテストパターンの生成に成功した。

表 2 実験 1 実験結果

世代数	平均時間	最長時間	網羅率
124.56	0.98 秒	1.56 秒	100.0%

表 3 実験 2 実験結果

世代数	平均時間	最長時間	網羅率
156.9	4.18 秒	6.15 秒	100.0%

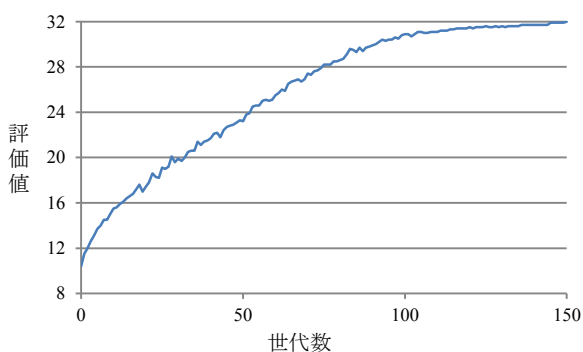


図 3 実験 1 学習曲線

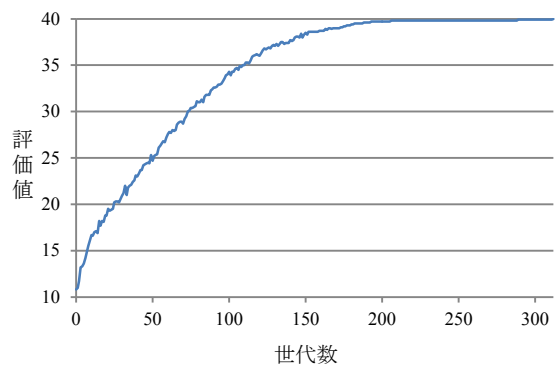


図 4 実験 2 学習曲線

4. 考察

パス総数が既知の場合を想定した実験 1 ではパスの網羅率 100% のテストパターンの生成に成功した。パス総数が未知の場合を想定した実験 2 でもパスの網羅率 100% のテストパターンを生成することに成功した。また, 関連研究として前述した Pairwise 法と Vibration Method との比較では, 同じテストデータではないが前者のパス網羅率は 53%, 後者のパス網羅率は 92% という結果が報告されている[1]。関連研究に用いられたシステムと類似したシステムを用いたことを考慮すると, 正確な比較ではないが, 本手法によるテストパターン生成は有効であると考えられる。

5. むすび

本稿では, 形式仕様に基づいたソフトウェア開発において, GA を用いてプログラムの全てのパスを通過するテストケースを自動生成する方法を提案した。実用的なプログラムを想定し, 形式仕様に基づき設計した ATM システムを用いてテストケース生成の実験を行った。その結果, パスの総数を既知とした実験では網羅率 100% の精度のテストパターン生成することを示し, パスの数を未知とした実験でも網羅率 100% の精度のテストパターンを生成することを示した。類似したシステムを用いて実験を行った従来手法と比べ, パス網羅率を大きく向上できる可能性を示した。

文 献

- [1] S. Liu, S. Nakajima: "A Vibration Method for Automatically Generating Test Case Based on Formal Specifications," Software Engineering Conference, 2011 18th Asia Pacific, pp. 73-80, Dec. 2011
- [2] Pargas, P. R. Harrold. R, R, Peck: "Test-Data Generation Using Genetic Algorithms," Journal of Software Testing, Verification and Reliability vol. 9, Issue 4, pp.263-282, 1999.
- [3] 杉原拓, 劉少英, 佐藤裕二, "形式仕様に基づくテストパターン生成への GA の適用," 研究報告数理モデル化と問題解決(MPS), 2014-MPS-100, no.15, pp.1-2, Sep.2014.
- [4] Y. Zhou, T. Sugihara, Y. Sato, "Applying GA with Tabu list for Automatically Generating Test Cases Based on Formal Specification," SOFL+MSVL 2014, Nov. 2014