

# SBFL手法における疑惑値の分布と バグ限局精度の関係について

高橋 佑介<sup>1</sup> 阿萬 裕久<sup>1</sup> 川原 稔<sup>1</sup>

**概要:** 人間の手を介さずにプログラム中のバグを修正する自動プログラム修正において、テスト等の結果に基づいてバグの原因箇所を特定する“バグ限局”は重要な役割を担っている。有用なバグ限局手法であるSBFLではテストの実行トレースを用いて限局を行うが、条件分岐によって分割されるブロックが存在すると、ブロック中のコード行で差別化を図れないという課題が知られている。そこで本稿では、SBFLによる疑惑値の分布に注目しバグ限局精度との関係についてデータ分析を行う。

## On Relationship Between Suspicious Score Distribution and Bug Localization Accuracy

### 1. はじめに

ソフトウェアライフサイクルにおいて、バグの修正（デバッグ）を含む保守作業は最もコストのかかる作業となっている [1]。そこで近年、人間の手を介さずにプログラム中のバグを修正する自動プログラム修正 [2] が注目されている。一般に自動プログラム修正においては、テスト等の結果に基づいてバグの原因箇所を特定する“バグ限局”を行い、その上でさまざまなコード修正を試行するというアプローチがとられている。それゆえ、バグ限局を適切に行うことが自動プログラム修正の鍵となっている。これまでさまざまなバグ限局手法が提案されてきているが、その中でもテストにおける実行トレース（プログラムスペクトル）を活用したSBFL（Spectrum-Based Fault Localization）と呼ばれる手法が有用な手法として知られている [3]。

SBFL手法ではさまざまなテストケースを実行して経路情報を記録しておき、各コード行を実行する失敗テストケースや成功テストケースの個数並びに割合を調べることで当該コード行がバグの原因箇所となっている可能性を“疑惑値”として定量化する。しかしながら、そういったSBFL手法では実行経路情報に基づいている都合上、条件分岐によって分割されるブロックがバグ位置として推定可能な最小の単位になってしまう。つまり、そのようなブ

ロックが大きくなればなるほどバグ位置の絞り込みは難しくなってしまう。

本稿では、SBFLによる疑惑値のばらつきと限局精度との関係に注目し、データ分析の観点から考察する。

### 2. プログラムスペクトルに基づいたバグ限局

#### 2.1 疑惑値の算出式

SBFL手法での代表的な疑惑値算出式に Ochiai がある。Ochiai はSBFL技術の中で最も推定程度が高く有望な手法といわれている [4]。便宜上、疑惑値を算出したいプログラム行を“ $s$  行目”として、表 1 に示す記号で関係する値を表現した場合、疑惑値は以下の式で算出される。なお、疑惑値算出にあたっては、失敗テストケースが少なくとも 1 個は存在する ( $N_f > 0$ ) ことを前提とする。

$$susp(s) = \frac{n_f(s)}{\sqrt{N_f \cdot \{n_f(s) + n_p(s)\}}} \quad (1)$$

表 1 SBFL 手法での疑惑値算出式で使用する記号一覧

記号	意味
$N_f$	失敗テストケースの総数 (> 0)
$N_p$	成功テストケースの総数
$n_f(s)$	$s$ 行目を実行した失敗テストケースの数
$n_p(s)$	$s$ 行目を実行した成功テストケースの数
$susp(s)$	$s$ 行目の疑惑値

<sup>1</sup> 愛媛大学  
Ehime University

## 2.2 SBFL 手法の課題点

Ochiai での疑惑値の算出には  $n_f(s)$  並びに  $n_p(s)$  の値が主たる要素として使われていた。そのこと自体は決して否定されるものではなく、実際に行われたテストの結果を明確に反映した有用な手法であると考えられる。しかしながら、そのような“実行テストケースの個数”はプログラムの実行経路、即ち、制御フローに依存した値となるため、条件分岐が登場しない範囲内の実行可能行では全て同じ結果になってしまう。つまり、そのような範囲内では実行可能行の差別化ができず、バグの原因箇所を絞り込むことができなくなってしまうという問題がある。

## 2.3 コード修正履歴情報を用いた拡張手法とその結果

前述の問題に対応するため、我々は既存の SBFL 手法によって算出された疑惑値にコード修正履歴情報から得られた係数を重みとして乗じる拡張手法を提案している(図 1)。

この手法で予備実験を行った結果、全体で精度の向上が見られた。しかし、低下したものも少なからず散見されており、その中にはももとの手法による限局精度の方が高いものも含まれていた。よって、従来手法で高い精度を発揮できるものに関しては変更を加えず、反対に従来手法で限局精度の低いものには拡張手法を適用することができればさらなる精度の向上が見込めると考える。

## 3. 分析結果

本節では、どのような場合に SBFL 手法での限局精度が低いあるいは高いのかについて検討する。具体的には、SBFL 手法の限局精度を“EXAM スコア”[5]と呼ばれる評価指標を用いて数値化したものと疑惑値のばらつきについて分析し、結果を基に考察していく。

### 3.1 疑惑値の最大値と最小値の差を用いた分析

分析の結果、限局精度と疑惑値の最大値と最小値の差について相関があることがわかった。この結果を図 2、表 2 に示す。

図 2 の 3 つの箱ひげ図は、左から疑惑値の差が 0 以上 0.447 (第 1 四分位数) 未満、0.447 以上 0.707 (中央値)

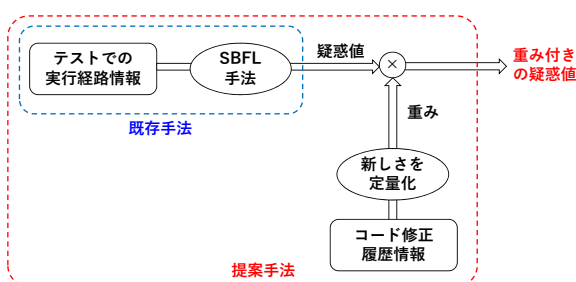


図 1 拡張手法の概念図

表 2 疑惑値の最大値と最小値の差に関する統計量

最小値	25%	中央値	平均値	75%	最大値
0.040	0.447	0.707	0.658	1.00	1.00

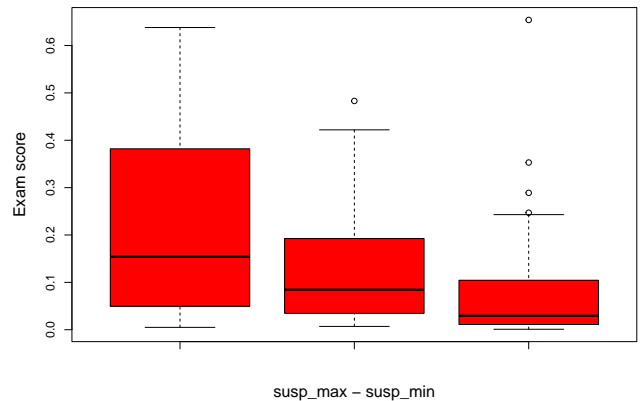


図 2 疑惑値の最大値と最小値の差に対する EXAM スコアの分布

未満、0.707 以上 1 以下となっている。なお、EXAM スコアは値が小さいほど限局精度が高いことを示している。

図 2 より、疑惑値の差が大きいほど限局精度が高いことが分かる。つまり、疑惑値の最大値と最小値の差が大きいものについては変更を加えず、反対に差が小さいものには適切な改良を加えることで限局精度の向上を見込めるといえる。

## 4. おわりに

本稿では、代表的なバグ限局手法である SBFL において限局精度と疑惑値のばらつきに係性についてデータ分析の観点から考察した。ワークショップでは、さらなる限局精度の向上に向けた取り組みについて議論したい。

## 参考文献

- [1] LaToza, T. D., Venolia, G. and DeLine, R.: Maintaining Mental Models: A Study of Developer Work Habits, *Proc. 28th Int'l Conf. Softw. Eng.*, pp. 492–501 (online), DOI: 10.1145/1134285.1134355 (2006).
- [2] Le Goues, C., Nguyen, T., Forrest, S. and Weimer, W.: GenProg: A Generic Method for Automatic Software Repair, *IEEE Trans. Softw. Eng.*, Vol. 38, No. 1, pp. 54–72 (online), DOI: 10.1109/TSE.2011.104 (2012).
- [3] Abreu, R., Zoetewij, P. and Van Gemund, A. J.: An Evaluation of Similarity Coefficients for Software Fault Localization, *Proc. 12th Pacific Rim Int'l Symp. Dependable Computing*, pp. 39–46 (online), DOI: 10.1109/PRDC.2006.18 (2006).
- [4] Abreu, R., Zoetewij, P., Golsteijn, R. and van Gemund, A. J.: A practical evaluation of spectrum-based fault localization, *Journal of Systems and Software*, Vol. 82, No. 11, pp. 1780–1792 (online), DOI: <https://doi.org/10.1016/j.jss.2009.06.035> (2009).
- [5] Wong, E., Wei, T., Qi, Y. and Zhao, L.: A Crosstab-based Statistical Method for Effective Fault Localization, *Proc. 1st Int'l Conf. Softw. Testing, Verification, & Validation*, pp. 42–51 (online), DOI: 10.1109/ICST.2008.65 (2008).