1G-6　発見誤りの重複度を用いたプログラムの残存誤り個数の推定法

若杉忠男

三菱電機株式会社　コンピュータシステム製作所　システム製造管理部

1　概要

ここで述べる残存誤り推定法は次のようなものである．テスト対象のプログラムに，何ケースかのテストデータを入力し，その間，プログラムの修正や変更をしないものとすれば同じ誤りを重複して発見することがある．残存誤りが少なくなれば，同じ誤りを重複して発見する割合は増加するであろう．この発見誤りの重複の程度と発見個数とから誤りの個数を推定する方法を述べる．

2　定義と計算式

N'を誤り総数，N1を重複を含む誤り発見回数，N2を重複を除いた誤りの発見個数とし，N'の推定値をn'，N1，N2の実測値をn1，n2などと表す．ここでN1はテストに投入した労力に比例し，N2はGoel－Okumotoの曲線

$$N2=N'\times(1-e^{-T}) \qquad (1)$$

で表されると仮定する．この関係を図示すると図1のようになる．

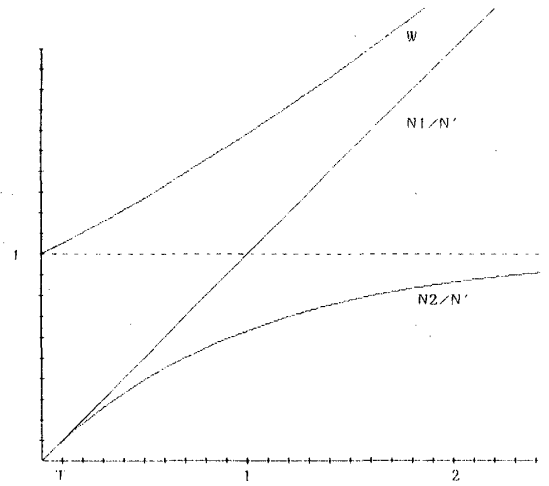

図1　T，N1／N'，N2／N'とW

ここで横軸はテストへの労力投入量　T　である．これはたとえば，テストケース数×各テストケースの誤り発見能力の平均　などで表される．

またWは発見誤りの重複度で次のように定義する．

$$W=N1/N2 \qquad (2)$$

図1からWはTが増えるにつれて単調に増加していることがわかる．したがってWからそれに対応するTが分かり，そのTとN2からN'が読み取れる．

数式で表すと，仮定から

$$N1=N'\times T \quad \text{かつ} \quad N2=N'\times(1-e^{-T})$$

であるから，

$$W=N1/N2=T/(1-e^{-T}) \qquad (3)$$

となる．（3）式を使って，WからTが求められ，このTとN2から次式によりN'が求められる．

$$N'/N2=1/(1-e^{-T}) \qquad (4)$$

図2は，Wとそれに対応する　N'／N2　のグラフである．

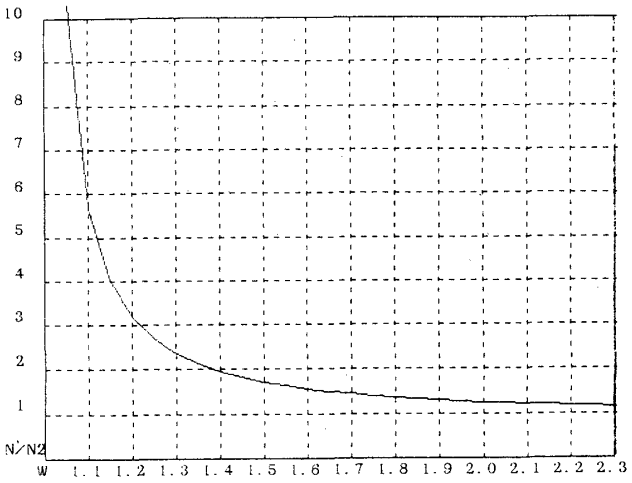

図2　　WとN'／N2

この関係を使って，N2の実測値n2から，次式によりN'の推定値n'を求める．

$$n'=n2\times N'/N2 \qquad (5)$$

3　シミュレーションの結果

この方法の妥当性を見るために，次のようなコンピュータによるシミュレーションを行った．１００個のエリアの内の５０個に１を埋め込み，一様乱数を発生させ，乱数の示すエリアに１があれば誤りが見つかったとみなした．乱数の発生一つがテストデータの１回投入に相当する．

A Method to estimate the Number of Errors left in Program

Tadao Wakasugi

Mitubishi Electric Corporation

実験の条件は次のとおりである．

（1） 乱数発生ごとにn1，n2，およびw＝n1／n2を求め，それによって推定したn'のカーブを描いた．ただし，n1－n2＞3かつn2＞4になってから推定計算を行った．

こうして求めたグラフの一例が図3である．これは図1に相当するグラフのシミュレーション結果に，n'／N'を添えたものである．言うまでもなく，n'／N'が1に近いほど推定精度が高い．

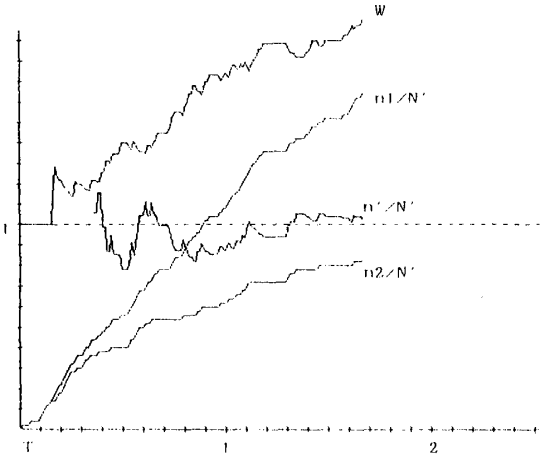

図3 シミュレーション結果（一例）

（2） 次に，このシミュレーションをwが1．8に達するまで行い，1．8に達した時点でn'を計算するということを，100回繰り返した．こうして得たn'／N'をヒストグラムで表したものが図4である．

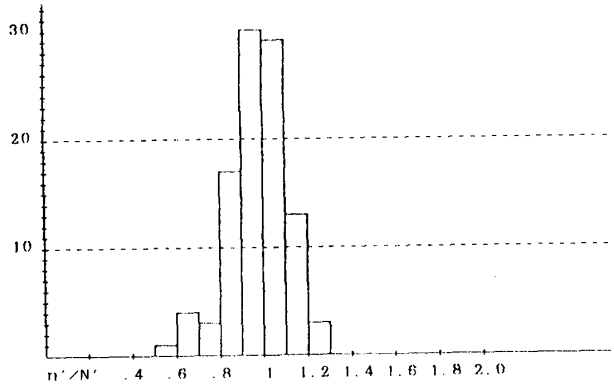

図4 n'／N'のヒストグラム

（3） 上記の（2）において，シミュレーションの打ち切り条件を，w＝1．2から2．0まで0．2きざみに変えて，n'／N'の平均と標準偏差を求めた．そのグラフが図5である．

4 考察

シミュレーションの結果を見ると，

（1） w＝1．8くらいになるまでテストケース数を増やすと，残存誤りの推定精度が高まり充分予測に使える．

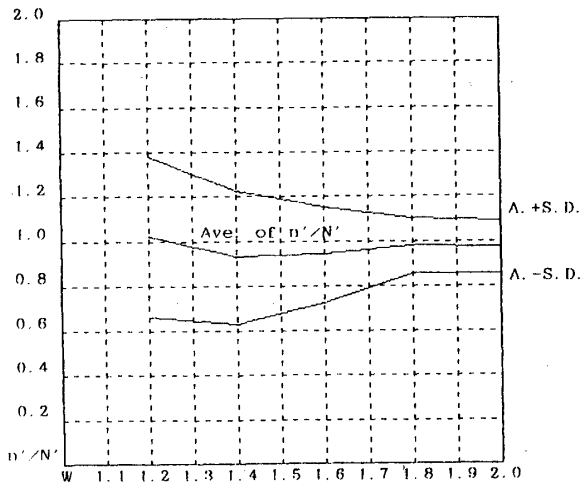

図5 wとn'／N'

（2） 1．8以上になったら，それ以上続けても推定精度はあまり向上しない．

（3） w＝1．8のとき，図2から発見した誤り個数を約1．4倍すれば誤り総数となることが分かる．

このシミュレーションでは，w＝1．8になるまでテストを行うには，残存誤りの1％を発見できるだけの質をもったテストデータを約140ケース投入する必要がある．

他の方法と比較するために，最小自乗法による曲線近似を試みたが，最小自乗法による推定値は不安定で比較できるような値が求められなかった．

したがって，ここで述べた方法は安定して信頼性の高い方法として役に立つと考えられる．

5 結論

（1） この手法は次のような場合に使えるであろう．

a プログラムの出荷検査のためのプログラム評価判定．

b プログラムの出荷後のクレームの分析評価．

（2） 実際にこの方法で残存誤り数を推定するには，次の手順によるとよいであろう．

a 被テストプログラムに偏りのないテストデータを何ケースか投入する．テスト中は被テストプログラムは変更しない．

b そのテスト結果から，n1とn2を求める．

c w＝n1／n2 を求め，図2によって N'／N2 を読みとり，それにn2をかけてN'の推定値とする．

（3） 今回行ったシミュレーションの範囲では，この方法は曲線近似法よりも推定の信頼度が高い．

6 参考文献

若杉忠男，“残存誤り数の一推定法”，情報処理学会研究報告，Vol.86,No.7,(1986.2)

以上