

数理計画モデルを利用したテストケース選択手法

Test Case Selection Method Using Mathematical Programming Model

佐々木 愛美

小笠原 秀人

津久井 秀樹

■ SASAKI Manami

■ OGASAWARA Hideto

■ TSUKUI Hideki

ソフトウェア開発の最終工程に位置するテスト工程は、ソフトウェアの品質を確保するための重要な工程である。近年、ソフトウェアの大規模・複雑化に伴いテストケース^(注1)の規模も数千~数万件に増大しているが、その一方で、開発期間は短縮される傾向にある。このような状況下でも品質を確保するためには、潜在する不具合をなるべく早期に検出し、その修正を確実に実施することが重要である。

東芝は、国立大学法人 愛媛大学（以下、愛媛大学と略記）と共同で、この課題を解決するための一つの方法として、数理計画モデルを利用したテストケースの選択手法を開発した。これは、投入可能な工数の範囲内で実施可能であり、かつ、膨大なテストケースの中から不具合検出の期待値が最大になるようにテストケースのセットを選択する手法である。更に、この手法を医用機器の開発に適用し、その有効性を確認した。

Software testing performed in the final stage of development of software systems is essential to ensure software quality. While the number of test cases for verification and validation has sharply increased with the dissemination of larger scale and more complex software systems in recent years, software development periods have been becoming shorter year by year. Under these circumstances, there is a need for highly sophisticated and rapid testing that can detect defects earlier, more effectively, and with greater certainty.

As a solution to this issue, Toshiba, in cooperation with Ehime University, has developed a test case selection method that offers a set of the most effective test cases by calculating the highest expected values of failure detection based on a 0-1 programming model from among large numbers of test cases within an environment of constrained resources. We have confirmed the effectiveness of this method by applying it to the development of medical equipment.

1 まえがき

ソフトウェアの品質を確保するためには、不具合やその修正による悪影響で新たな不具合が発生するデグレードの見逃しは許されない。そのため、ソフトウェア開発におけるテスト工程では、テストの実施と不具合の修正を繰り返し行い、テストで検出された不具合がデグレードを起こすことなく適切に修正されていることを確認する必要がある。

不具合やデグレードの見逃しをより確実に防ぐためには、ソフトウェアが更新されるたびに、全ての機能に対して、これまでの開発で蓄積した全てのテストケースを再実施することが理想である。しかし、近年のソフトウェア開発では、ソフトウェアの大規模・複雑化に伴いテストケースの規模も増大しているが、その一方で、より短期間で開発が求められる傾向にある。このため、ソフトウェアが更新されるたびに、全てのテストケースを再実施するのは時間的に困難である。

このような状況に対応するため、ソフトウェアのテスト技術者

は、投入可能な工数内で実施でき、かつ、不具合やデグレードの見逃しがないように、過去のテストの実績や不具合検出の傾向などを総合的に考慮して、再実施するテストケースのセットを選択している。この作業は、ソフトウェアの品質の作込みと確認において、非常に重要な役割を果たしている。

しかし、膨大なテストケースの1件1件を吟味し、投入可能な工数内で効果をもっとも高くなるようにテストケースを取捨選択することは極めて難しい。また、テスト技術者のノウハウやスキルに依存する部分が多く、効率的なテストケースのセットが選択できていないとは必ずしも言えない。そのため、テスト技術者のノウハウやスキルに左右されず、投入可能な工数内で効果をもっとも高くなるようにテストケースのセットを選択する手法を確立することが重要になる。

東芝は、愛媛大学と共同で、前述した課題を解決するための一つの方法として、数理計画モデルを利用したテストケースの選択手法を開発した⁽¹⁾。ここでは、テストケース選択手法の概要と、東芝グループ内で製品開発へ適用した事例について述べる。

(注1) ソフトウェアのテストにおいて、個々のテストの適合要件を定めたもの。一般に、実行前条件、入力値、期待結果、及び実行後条件の組合せとして定義される。

2 数理計画モデルを利用したテストケース選択手法の概要

数理計画モデルを利用したテストケース選択手法とは、投入可能なテスト工数の上限とテストケース間の依存関係を満たしたうえで、各テストケースの再実施における不具合やデグレード検出の期待値が最大になるように、最適なテストケースのセットを選択する手法である。

テストケースのセットを選択する手順を図1に示す。選択の手順は、次の三つのステップで構成される。

ステップ1 テストケースごとに不具合検出の期待値を算出

ステップ2 テストケース間の依存関係を抽出

ステップ3 テスト工数の上限とテストケース間の依存関係を満たす最適なテストケースのセットを選択

各ステップの詳細について以下に述べる。

2.1 テストケースごとの不具合検出の期待値算出

どのテストケースを再実施すべきかを判断するため、再実施によりどの程度不具合やデグレードの検出が期待できるかを各テストケースごとに数値化する。

一般に、不具合の80%は全体の20%の箇所に集中するという経験則が知られており、過去に不具合が頻発した箇所は、今後も不具合を引き起こす可能性が高い。そこで、過去の不具合検出の傾向から、何度も不合格となっているテストケースは不具合検出の期待値が高いと考え、“不合格の割合”を期待値として利用する。不合格の割合とは、そのテストケースが

テストケース名	バージョン1	バージョン2	バージョン3	バージョン4	バージョン5	不合格の割合：1/2 = 0.5 未実施期間の長さ：3
テストケース1	×	○	-	-	-	不合格の割合：1/3 = 0.3 未実施期間の長さ：0
テストケース2	-	○	-	×	○	
テストケース3	-	×	×	○	-	不合格の割合：2/3 = 0.7 未実施期間の長さ：1
テストケース4	○	-	-	-	-	
						不合格の割合：0/1 = 0 未実施期間の長さ：4

図2. 不具合検出の期待値の算出例 — テストの実施履歴から、テストケースごとに不具合検出の期待値を算出する。期待値には、“不合格の割合”と“未実施期間の長さ”の2種類がある。テストケースのセットを選択する際には、どちらか一方の期待値を選び利用する。

Example of calculation of expected values of failure detection

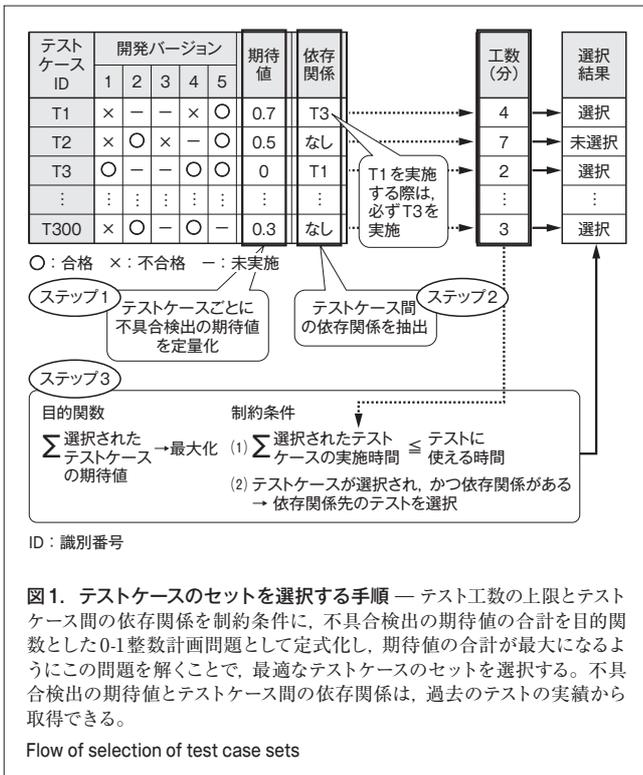
過去に実施された事例のうち、結果が不合格となったものの割合であり、それが大きいほど不具合検出の期待値が高いとみなす(図2)。図2のテストケース1を例に不合格の割合を算出すると、図中の○が合格、×が不合格、-が未実施を表しており、実施した2回のテストのうち1回が不合格であったことから、テストケース1の不合格の割合は、1/2=0.5となる。

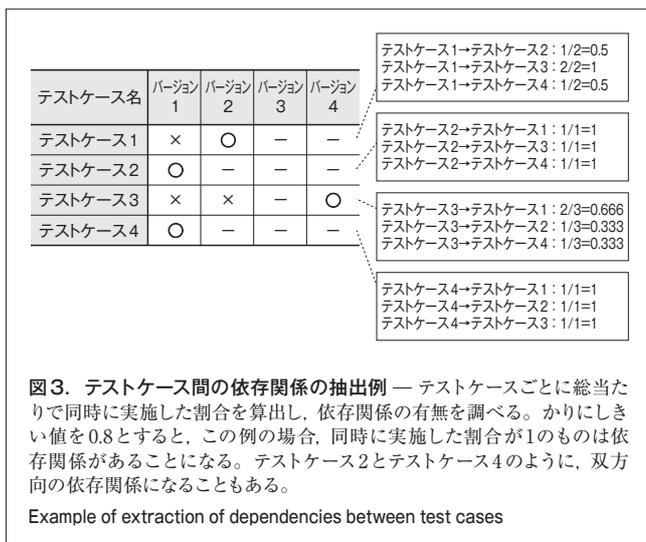
いったん合格となったテストケースに対しては、ソースコードの修正による影響を受けている可能性が強くないかぎり、再実施されることは少ない。しかし、実際にテストケースを実施し合格であることを確認していない期間が長いと、デグレードの発生に気づかないリスクが高くなる。そこで、しばらくの間実施されていないテストケースはデグレードを検出する期待が高いと考え、“未実施期間の長さ”を期待値として利用する。未実施期間の長さとは、そのテストケースが最後に実施されたバージョンから現在のバージョンまでの間で、テストケースが実施されなかったバージョンの数を表し、その数が多いほど不具合検出の期待値が高いとみなす。図2のテストケース1を例に未実施期間の長さを算出すると、テストをバージョン2で実施した後、バージョン3、バージョン4、及びバージョン5では実施していないことから、テストケース1の未実施期間の長さは3となる。

2.2 テストケース間の依存関係の抽出

不具合修正の確認のためテストを実施するときには、その不具合が原因で不合格となっていたテストケースだけでなく、修正箇所の影響を受けるテストケースもいっしょに実施することが多い。このように、あるテストケースを実施するときには、同時に他のテストケースも実施したほうがよいという関係を、ここではテストケース間の依存関係と呼ぶ。

最適なテストケースのセットを選択する際には、この依存関係にも注意を払って選択する必要がある。通常は、修正確認を行うときに、修正箇所の影響範囲を洗い出し、該当箇所を対象としたテストケースを選択することで依存関係を抽出する。しかし、膨大なテストケースの全てに対して依存関係を確





認することは難しい。

そこで、あるテストケースが実施されたときに同時に実施しているテストケースは依存関係にある可能性が高いと考え、テストの実施履歴から自動で依存関係を抽出する方法を開発した。この方法では、あるテストケースに対して、そのほかの各テストケースが同時に実施された割合を算出し、その値がしきい値以上になったテストケースは依存関係があるとする。

図3のテストケース1を例に依存関係を算出してみる。テストケース1はバージョン1とバージョン2で実施されており、テストケース2はバージョン1だけで実施されている。このとき、テストケース1に対して、テストケース2が同時に実施された割合は1/2=0.5となる。同様に、テストケース3が同時に実施された割合は1、テストケース4が同時に実施された割合は0.5である。かりに、しきい値を0.8とすると、テストケース1に対して依存関係にあるのはテストケース3となる。

この依存関係は、ステップ3で最適なテストケースのセットを選択する際に、再実施においてテストケース1を選択した場合には、テストケース3も必ず選択するというように利用する。

2.3 テスト工数の上限とテストケース間の依存関係を満たす最適なテストケースのセットの選択

投入可能な工数の上限とテストケース間の依存関係を満たしたうえで、不具合検出の期待値が最大になるようにテストケースのセットを選択する。これを実現するために、選択されたテストケースによる不具合検出の期待値の合計を目的関数、投入可能な上限工数を制約条件1、及びテストケース間の依存関係を制約条件2とする、0-1整数計画問題⁽²⁾として定式化する。

N 件のテストケースが存在するとき、 i 番目のテストケースの不具合検出の期待値を F_i とすると、目的関数は式(1)で表される。

$$\sum_{i=1}^N F_i \cdot x_i \quad (1)$$

x_i は、 i 番目のテストケースを再実施するとして選択するかを表し、選択する場合には1、選択しない場合には0となる。

次に、二つの制約条件について考える。 i 番目のテストケースの実施工数を C_i 、テストに投入可能な上限工数を L とすると、制約条件1は式(2)で表される。

$$\sum_{i=1}^N C_i \cdot x_i \leq L \quad (2)$$

また、 i 番目のテストケースを実施するときには j 番目のテストケースも実施したほうがよいという依存関係は、制約条件2として式(3)で表される。

$$x_i \leq x_j \quad (3)$$

i 番目のテストケースが選択される($x_i=1$)と、 j 番目のテストケースも選択されなければ($x_j=1$)、式(3)を満たすことはできない。この制約条件を、存在する依存関係の数だけ作成することで、テストケース間の全ての依存関係を表現できる。

このように、定式化した0-1整数計画問題を解くことで、期待値が最大になるテストケースのセットを選択できる。

3 製品開発への適用事例

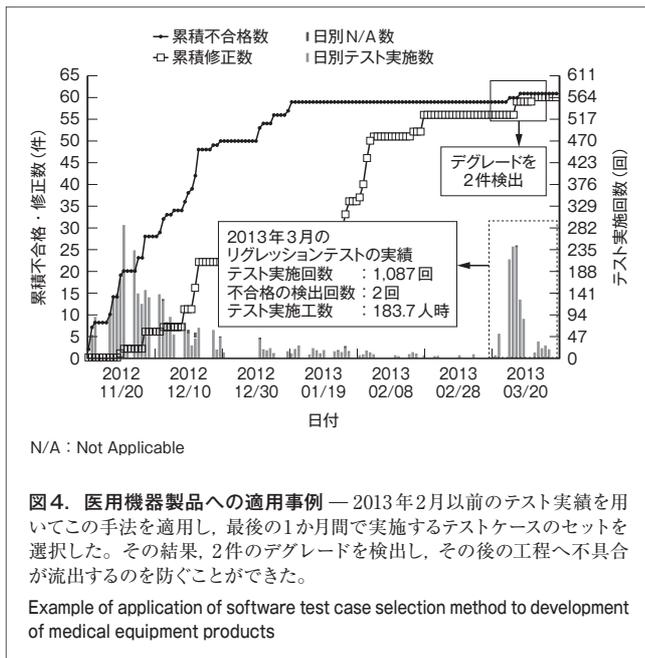
ここでは、数理計画モデルを利用したテストケースの選択手法を、製品開発へ適用した事例について述べる。

適用対象製品は、東芝メディカルシステムズ(株)と東芝医用システムエンジニアリング(株)が開発している全身用MRI(磁気共鳴イメージング)装置で、およそ半年から1年の間隔でシリーズ製品を商品化している。品質保証部門による製品試験の前段階で、開発部門によりソフトウェア品質の最終確認が行われるが、そのためのシステム機能試験工程でこの手法を適用した。システム機能試験工程は、2012年11月20日から2013年3月末までのおよそ4か月半で実施された。

この手法を適用する目的は、最後の1か月間に実施するリグレッションテスト(回帰テスト)^(注2)のテストケースを選択することである。システム機能試験工程で実施するテストケースは全部で2,997件あるが、最後の1か月間でテストに投入できる工数は約170人時であった。そこで、主要な機能群を対象に、テストに投入可能な工数を170人時とし、この手法を利用してテストケースの選択を実施した。その結果、全テストケースの約35%にあたる1,062件のテストケースが選択された。

システム機能試験工程で実施したテストの累積不合格数を図4に示す。この図の横軸は日付を表しており、2013年3月に

(注2) 変更によりソフトウェアの未変更部分に新たに欠陥が入り込んだり、発現しないことを確認するため、変更実施後に、既にテスト済みのプログラムに対して実行するテスト。ソフトウェアや実行環境が変わるたびに行う。



実施されたリグレッションテストの実績がグラフの右端に表示されている。この手法を使って選択されたテストケースのセットを実施した結果、2件のデグレードを検出できた。テスト技術者がノウハウだけに頼ってテストケースのセットを選択した場合には、この2件のうち1件のデグレードを検出したテストケースは選択されなかったことを確認した。更に、その後の工程への不具合流出はゼロであったことから、この手法を適用することで、時間制約がある状況下でも、潜在する不具合やデグレードをもれなく検出できたと考えられる。

実際の開発では、限られた工数の中で全てのテストケースを繰り返し実施することは困難である。開発の経験やノウハウに頼るだけでなく、不合格の割合や未実施期間の長さといった明確な数値を使って、テストケースの再実施の可否を判断することが重要であり、かつ効果的であることを確認した。

4 数理計画モデルを利用したテストケース選択手法のツール化

この手法を製品開発で適用する際の導入コストを低減するため、当社で普及が進んでいるテスト管理システムTETRAPLUS™⁽³⁾にテスト選択機能として実装した。これにより、テストケースごとの期待値算出からテストケースのセット選択までの一連のステップを自動化でき、テスト担当者に掛かる負荷を大きく低減できた。

TETRAPLUS™は、テストケースとその合否の結果を一元管理するシステムである。合否の結果には、そのテストケースの実施に要した工数や、テスト対象となったソフトウェアのバージョンが属性情報として登録されており、この手法を使ったテ

ストケースのセットの選択に必須のデータがそろっている。TETRAPLUS™を使えば、この手法を利用するための環境を簡単に構築できる。また、経験の浅いテスト技術者でも、機能の使い方と留意点を理解すれば利用可能であり、再実施するテストケースをベテランの技術者と同じ観点で選択できる。

5 あとがき

当社は、限られた工数の範囲内で効率的にソフトウェアの不具合を検出するため、投入可能なテスト工数の上限とテストケース間の依存関係を満たしたうえで、不具合検出の期待値が最大になるように、数理計画モデルを利用して最適なテストケースのセットを選択する手法を開発した。

今後、テスト管理システムTETRAPLUS™の社内への普及を加速するなかで、この手法の適用を拡大していく予定である。また、テスト実績だけでなく、不具合情報やソースコードメトリクスなどのデータも活用した、テストケースの不具合検出の新たな期待値についても検討を進めていく。

文 献

- (1) Aman, H. et al. "Application of the 0-1 Programming Model for Cost-Effective Regression Test". Proc. 37th Annual International Computer Software & Applications Conference (COMPSAC2013). Kyoto, Japan, 2013-07, IEEE Computer Society. IEEE, p.720 - 721.
- (2) 福島雅夫. 数理計画入門. 東京, 朝倉書店. 2011, 216p.
- (3) 河村 透 他. ソフトウェアのテスト管理システム. 東芝レビュー. 66, 1, 2011, p.32 - 36.



佐々木 愛美 SASAKI Manami

ソフトウェア技術センター プロセス・品質技術開発担当。
ソフトウェアテスト技術の開発に従事。情報処理学会会員。
Corporate Software Engineering Center



小笠原 秀人 OGASAWARA Hideto, D.Eng.

ソフトウェア技術センター プロセス・品質技術開発担当主査, 博士(工学)。ソフトウェアテスト技術の開発に従事。IEEE, 情報処理学会, 電子情報通信学会, プロジェクトマネジメント学会会員。
Corporate Software Engineering Center



津久井 秀樹 TSUKUI Hideki

東芝医用システムエンジニアリング(株) 技術管理部 技術推進担当参事。ソフトウェア開発プロセス技術及び開発支援ツール活用技術の開発に従事。
Toshiba Medical Systems Engineering Co., Ltd.