不具合修正タスク量の最適化を目的とした バグトリアージ手法の提案

柏 祐太郎[1,a)]　大平 雅雄[1,b)]

概要：本研究の目的は個々の不具合修正タスクに最適な開発者を割当てることができるバグトリアージ手法の構築である．既存研究の問題点は不具合に対する開発者の適性のみを考慮しているため，特定の一部の開発者に修正タスクが集中する傾向にあることである．そこで本研究では一部の開発者にタスク集中させずに適任の開発者を推薦する手法を提案する．

1. はじめに

近年の大規模システム開発では，試験工程のみならず運用工程においても多数の不具合が検出されるため，不具合管理システムを用いて，検出された不具合を詳細に記録し，漏れのないように不具合を管理することが求められる．報告された不具合一つ一つに対して，重要度や優先度を設定し，適任の担当者に修正タスクを割当てることをバグトリアージと呼ぶ．

大量に不具合が報告される現状では，個々の不具合に対して適切にバグトリアージすることは容易ではない．大規模オープンソース開発プロジェクトのEclipse[*1]やMozilla[*2]では，約4割の不具合に対して，担当者の再割当てが行われており [1]，人手によるバグトリアージには限界があることが知られている．再割当てとは不具合の担当者が変更される事を表し，1度の再割当てで平均50日修正が遅れると報告されている [1]．人的リソースの浪費だけではなく，不具合が修正されるまでに多くの時間を要するため，出来る限り再割当てが生じないようにすることが望ましい．そのため，バグトリアージを支援する研究が現在盛んに行われている [1], [2], [3].

既存手法の多くは不具合に対する開発者の適性のみを考慮しているため，極めて有能な一部の開発者に集中して修正タスクを割当てる可能性が高く，再割当てを誘発する恐れがある [3]．そのため，開発者の適性を考慮した上でタスク量を最適化できる割当手法が求められる．

[1] 和歌山大学
Wakayama University
[a)] s141015@sys.wakayama-u.ac.jp
[b)] masao@sys.wakayama-u.ac.jp
[*1] http://www.eclipse.org
[*2] http://www.mozilla.org

2. 整数計画法によるタスク割当て

不具合修正タスクに対する開発者の適性を考慮しつつ，一部の開発者へタスク集中し過ぎないようにするためには，どの開発者がどの不具合を一定期間中にいくつ担当すれば最も効率的に不具合修正が行われるかについての最適解を求める必要がある．本研究では，不具合修正タスクの割当て問題を，開発者とタスクの組合せ問題と捉え，組合せ問題において最適解を得ることができる整数計画法を用いる．

2.1 整数計画法

整数計画法は，与えられた条件の下で目的を達成するためにより良い解を求める方法である．整数計画法に代表される数理計画法は，近年の計算機の発達により改めて注目されている最適化手法である．生産問題やスケジューリング問題といったオペレーションズリサーチ分野をはじめとして，ソフトウェア工学の分野でも応用され始めている [4]．整数計画法は以下のように定義される [5].

$$Max: \quad c^{\mathsf{T}}x \tag{1}$$

$$Subject: \quad Ax \leq b \tag{2}$$

$$x \geq 0 \tag{3}$$

$$x \text{ は整数} \tag{4}$$

ここで，x はn次元整数ベクトル，c はn次元ベクトル，b はm次元ベクトル，A はm行n列の行列である．x を問題の解で目的変数と呼ぶ．(1) を目的関数と呼び，(2), (3), (4) の制約の下，目的関数を最大とする目的変数 x の組合せを求める．

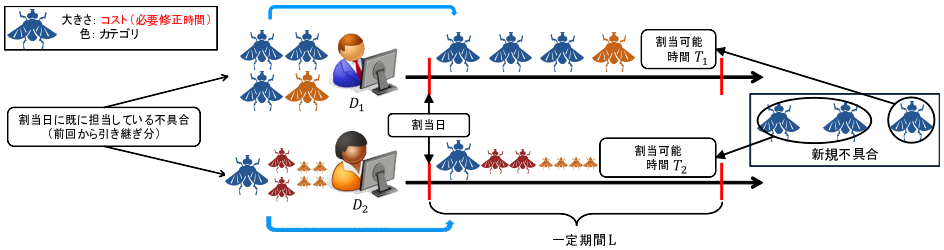

図 1　修正可能なタスク量の求め方

Fig. 1　A way to calculate the amount of tasks for each developer

2.2　定式化

本研究では，開発者（$D_i, i = 1, 2, ..., m$）と優先度とコンポーネントで分類されたカテゴリー j（$j = 1, 2, ..., n$）の不具合票が存在すると想定し，目的変数，目的関数，制約条件を次のように定義する．

目的変数：

$$x_{ij} \geq 0 \quad (x_{ij}\text{は整数に限る}) \tag{5}$$

x_{ij} とは開発者 D_i がカテゴリー j のタスクをいくつ担当させるかを表す．

目的関数：

$$Max : \sum_{i=1}^{m}\sum_{j=1}^{n} P_{ij}x_{ij} \tag{6}$$

各開発者と各カテゴリのプリファレンスとその目的変数の積の総和を最大化する．**プリファレンス** P_{ij} とは開発者 D_i のカテゴリー j の不具合に対する適性のことで，開発者 D_i がカテゴリー j の不具合を修正した数がカテゴリー j の修正数に占める割合である．この関数を最大化する解を求める事で，個々の開発者の適性に合うタスクがプロジェクト全体として最大となるような組合せを求める．

制約条件：

$$\sum_{j=1}^{n} C_{ij}x_{ij} \leq T_i \ (i = 1, 2, ..., m) \tag{7}$$

開発者が一定期間内に修正できるタスク量には限りがあると考えるのが自然である．本研究では，開発者 D_i が**一定期間** L に修正可能なタスク量を考慮した不具合の割当てを行う．修正可能なタスク量は**割当可能時間の上限** T_i を基に算出する．また，上限 T_i は，あらかじめ設定する L と，新規の修正タスク割当時点ですでに開発者 D_i が担当している不具合の**コスト** C_{ij} の総和から引いた値である．C_{ij} は，過去に開発者 D_i がカテゴリ j の不具合修正タスク B_j を完了するのに要した時間の平均（中央値）とした．**新規に割当てる修正タスクのコストの合計が** T_i **を超えないようにすることで，特定の開発者へ修正タスクが極端に集中するのを防ぐ効果を期待できる．**

2.3　修正可能なタスク量の求め方

修正可能なタスク量の求め方について，図 1 を用いて説明する．図 1 では，コストを不具合の大きさ，カテゴリを色で表している．開発者 D_1 と D_2 の修正可能なタスク量を求めるには，まず，D_1 と D_2 が既に担当している不具合のコストの総和をそれぞれ求める．次に一定期間 L からコストの総和を引き，各開発者の割当可能時間 T_1 と T_2 を算出する．カテゴリ j の新規不具合 B_j のコスト C_{1j} と C_{2j} を割当可能時間 T_1，T_2 内で割当てる事ができる不具合数が修正可能なタスク量となる．担当している不具合数では一見，D_1 の方が少ないが，不具合修正のコストを考慮すると D_2 の方が担当可能時間に余裕がある．このように本手法では開発者によって異なるタスクの重みを考慮している．

3.　おわりに

本研究では，既存手法が特定の一部の開発者に修正タスクが集中する傾向にあることに着目し，整数計画法に基づいてタスク割当てを最適化する手法を提案した．本研究の今後の課題は，複数のプロジェクトで実験を行い提案手法の一般性を確かなものにすること，機械学習による方法 [2] を用いてプリファレンスの精度を高めること，などが挙げられる．

謝辞　本研究の一部は，文部科学省科学研究補助金（基盤 (B): 23300009）および（基盤 (C): 24500041）による助成を受けた．

参考文献

[1] Jeong, G., Kim, S. and Zimmermann, T.: Improving bug triage with bug tossing graphs, *Proceedings of ESEC/FSE'09*, pp. 111–120 (2009).

[2] Anvik, J., Hiew, L. and Murphy, G. C.: Who should fix this bug?, *Proceedings of ICSE'06*, pp. 361–370 (2006).

[3] Guo, P. J., Zimmermann, T., Nagappan, N. and Murphy, B.: "Not my bug!" and other reasons for software bug report reassignments, *Proceedings of CSCW'11*, pp. 395–404 (2011).

[4] 阿萬裕久：論理的制約条件付き 0-1 計画モデルを用いた重点レビュー計画法，コンピュータソフトウェア（日本ソフトウェア科学会誌），Vol. 29,No.2, pp. 612–621 (2012).

[5] 福島雅夫：数理計画法入門，朝倉書店，東京 (1996).