

摂動を用いた影響波及解析によるボトルネック工程自動検出 ～検出のためのアルゴリズムの検討～

Automatic Detection of Bottleneck Processes through Perturbation-based Repercussion Analysis :

Consideration of an Algorithm for Detecting them

町田多嘉之† Takayuki Machida 木下大輔‡ Daisuke Kinoshita 内川裕貴 Uchikawa Yuki
小坂祐也‡ Kosaka Yuya 古宮誠一‡ Seiichi Komiya

† 芝浦工業大学 工学部

‡ 芝浦工業大学大学院 大学院工学研究科

†College of Engineering, Shibaura Institute of Technology.

‡Graduate School of Engineering, Graduate School of Shibaura Institute of Technology.

・要旨

近年企業では短納期での開発をするために、複数のプロジェクトが同時期に並行で行われる。そのため、各作業で必要となるリソース(作業員、開発環境等)は複数のプロジェクトで共有され、ソフトウェア開発プロジェクトのある作業を遂行するのに必要なスキルを持つリソース(要員)が十分でない場合、工程遅延が起これるとその作業の日程調整もリソースの変更も行えない状況が起これる。そのような工程をボトルネック工程と呼ぶ。このボトルネック工程が問題となりプロジェクトが納期までに完了できなくなる危険性がある。そこで、本稿では摂動を用いた影響波及解析によるボトルネック工程を自動検出するアルゴリズムを検討する。

1. はじめに

近年企業では短納期での開発をするために、複数のプロジェクトが同時期に並行で行われる。そのため、各作業で必要となるリソース(作業員、開発環境等)は複数のプロジェクトで共有され、同時期に作業が割り当てられ競合することがある。この競合により一部の作業で、リソースの変更も日程調整もできないような状況が生じる。このような状況にある工程を我々はボトルネック工程と呼んでいる。遅延がこのような工程にまで波及した場合、対策が行えずプロジェクトが納期までに完了できないことがある。このため、工程遅延がボトルネック工程に波及しないように、ボトルネック工程よりも前に工程遅延対策を講じる必要がある。しかし、他プロジェクトの開発計画を随時把握し、潜在的なボトルネック工程を事前に検出することは難しい。

この問題を解決するため、本研究では仮の工程遅延を与えること(これを摂動という)により、後続工程への影響を他の開発計画をも考慮してシミュレート(影響波及解析)[1]することで、ボトルネック工程を自動検出する方法を提案する。

2. ボトルネック工程の定義とその事例

複数のプロジェクトで共有されているリソースがあったとする。このリソースを使用している工程に遅延が波及したときに、その工程の日程を調整することも、その工程で使用するリソースを取り替えることもできない場合がある。このような状況にある工程をボトルネック工程と呼ぶ。図1にその具体例を示す。

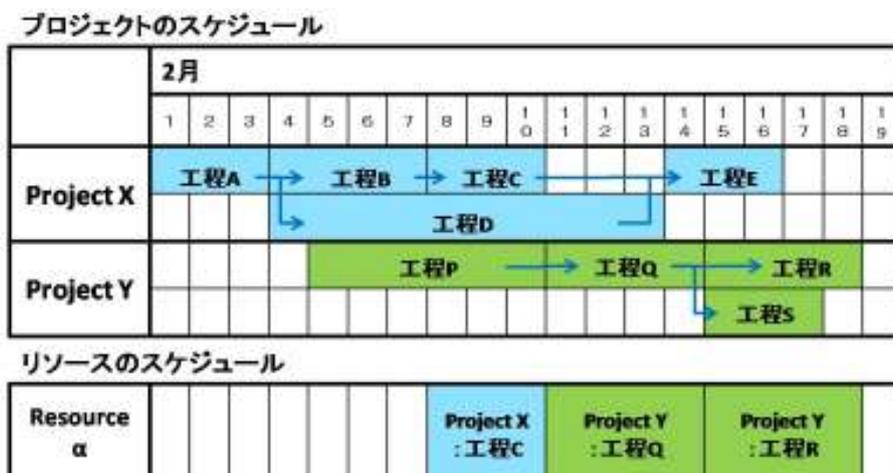


図1 ボトルネック工程の例

図1の上部は、プロジェクトのスケジュールを表し、矢印は「作業順序に関する制約」を示している。具体的には、図1の上部は「工程Aの作業が完了してから工程BとDの作業が開始される」という作業順序に関する制約を示している。図1の下部はリソースの使用スケジュールを表している。理解しやすくするために、ここでは工程Cに割り当て可能なリソースはαのみであると仮定する。図1はリソースαがプロジェクトXの工程Cと、プロジェクトYの工程Qと工程Rに割り当てられている状況を示している。この事例では、リソースαが割り当てられている、連続する3つの工程C、Q、Rの間には余裕日がまったくないことに注意して欲しい。

このような状況において、プロジェクトXの工程Bに遅延が発生した場合、「作業順序に関する制約」により、遅れた日数分だけ工程Cの開始日と終了日が予定より遅くなる。プロジェクトXのスケジュールだけを見ると、工程Cには3日間の余裕日があるため、3日間の遅れまでなら、遅れた日数分だけ工程Cの開始日と終了日を遅らせることで調整できると読み取れてしまう。しかし、そのようにすると「リソースの割り当て可能期間に関する制約」により、工程C、Q、Rに割り当てられているリソースαは、遅れた日数分だけ工程Qの作業に着手するのが遅くなり、工程Qのために3日間の作業日数を確保できなくなるので、工程Qにはリソースαを割り当てることができない。そのため、リソースαを工程Cに割り当てするには、リソースαの空きスケジュールが3日以上連続する19日まで待たなければならない。このため、要員の追加、プロジェクトY工程Qのリソース変更などを行わない限り、プロジェクトの終了日が大幅に延期され、プロジェクトは失敗してしまう。このため、工程Cはその作業日程を1日も動かすことができない。また、工程Cに割り当て可能なリソースはαだけなので、工程Cでは使用するリソースを取り替えることもできない。従って、図1の例では、工程Cがボトルネック工程である。プロジェクトを成功へ導くためには、例え1日の工程遅延でもその影響が、ボトルネック工程に波及しないようにプロジェクトを進めて行かねばならない。そのためには、ボトルネック工程（図1の例では工程C）の直前の工程（図1の例では工程B）の終了時まで、1日の遅延も生じていない状態にしなければならない。

3. 潜在的ボトルネック工程

潜在的ボトルネック工程の具体的な例を図1に示す。元来はボトルネック工程ではなかった工程が、工程遅延の影響を受け、ボトルネック工程に変質する場合がある。そのような工程を潜在的ボトルネック工程と呼ぶ。潜在的ボトルネックについて、図2を用いて説明する。

図2において、矢印は工程の作業順序を表している。

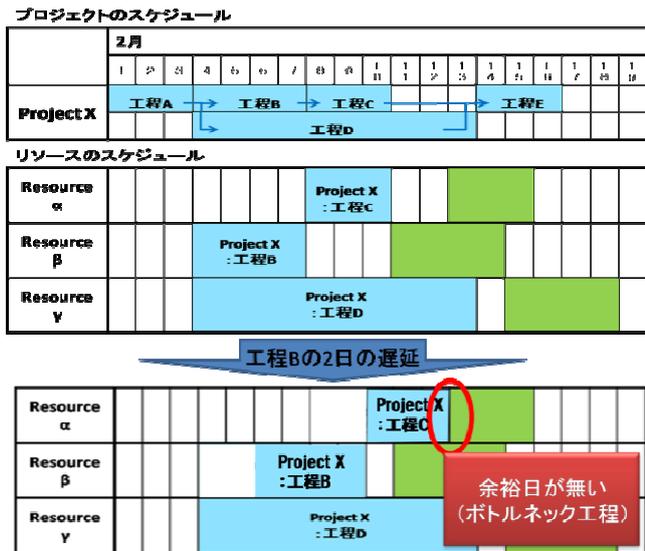


図2 潜在的ボトルネック工程の例

工程B、C、Dにリソースβ、α、γがそれぞれ割り当てられていてβ、α、γ以外に割当て可能なリソースは存在しないとする。リソースのスケジュールを見ると工程Bは3日、工程Cは2日、工程Dは1日、それぞれ遅延してしまうとボトルネック工程になってしまうことがわかる。しかし、実際には工程Bが2日遅延した段階(図1下部)で、その後続工程Cも遅延の影響を受けボトルネック工程になってしまう。

4. ソフトウェア開発計画問題が持つ制約

本研究では、ソフトウェア開発計画が満たさなければならない条件を制約と捉える[1、2]。その制約には、(1)作業順序に関する制約、(2)リソースの割り当て条件に関する制約、(3)リソース割り当て可能期間に関する制約、(4)リソースの能力的限界に関する制約の4種類がある。

5. 影響波及解析

工程遅延が発生したときに、プロジェクトへの影響の大きさを解析するためには、4章で列挙した制約を考慮しなければならない。特に、プロジェクトに従事する作業者の数が多く、しかも並列に実施される作業が多い場合には、発生した工程遅延が後続工程にどのような影響を与えるかをこれらの制約を考慮に入れてプロジェクト管理者が把握するのは非常に困難である。

影響波及解析の手順を図を用いて説明する。なお、下記の番号は図3の番号と対応している。

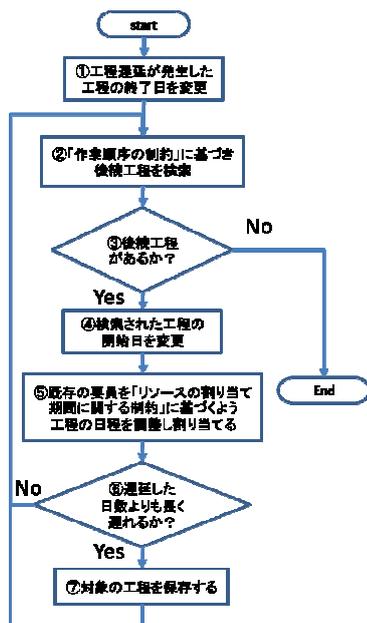


図3 影響波及解析の流れ

- ①工程遅延が発生した工程の終了日を変更。
- ②工程遅延が発生した工程の後続工程を「作業順序に関する制約」に基づいて検索する。
- ③検索の結果から後続工程の有無をチェックする。
- ④すべての後続工程で、開始日の変更が必要な工程の開始日をすべて変更する
- ⑤変更された開始日より「リソースの割り当て期間に関する制約」に基づき、要員のスケジュールを変更する。ただし、解析対象のプロジェクト以外の予定は変更しない。もし、その要員に他のプロジェクトの予定が入っている場合、その予定が終了した後工程に、作業を割り当てる。
- ⑥要員の再割り当てを行った結果、遅延した日数よりも遅れが大きいかを確認する。
- ⑦遅れが大きい工程の場合、調整が必要となる工程である可能性が高いため、その工程の情報を保存しておく。②から⑦を繰り返し、後続工程が無くなった(プロジェクトが終了した)時点で、影響波及解析を終了する。

