

プロジェクト・スケジュールの遅延リスク耐性評価方法の提案

富田 旋^{a)}・高塚 佳代子^{b)}・山場 久昭^{c)}・油田 健太郎^{d)}・岡崎 直宣^{e)}

Proposal of Delay Risk Tolerance Evaluation Method of Project Schedule

Meguru TOMITA, Kayoko TAKATUKA, Hisaaki YAMABA, Kentaro ABURADA, Naonobu OKAZAKI

Abstract

We propose methods related to schedule management even in project management. In schedule management of large-scale and complicated projects, due to irregular delays, rescheduling problems such as "when", "what kind of work", "how long" should be shortened frequently occur. In response to this problem, the existing method focused only on the work on the critical path (the longest work path), and solved by shortening arbitrary work on that path. However, regarding the question "Which work on the critical path should be abbreviated", the current state of the existing method has relied on the experiences and intuition of experts. Therefore, in this proposed method, focusing on all tasks including work on critical path, we propose an index that can evaluate the point that "Which task is shortened more effectively to shorten the overall construction period?" This index can be quantified by utilizing "Float" which can be afforded on the project schedule which has not been utilized in existing methods. In this paper, we call this index "Delay Risk Tolerance" and "the possibility that when the arbitrary work item gives an unexpected delay, the overall required time of the project does not exceed the contract due date"

Keywords: Project management, Critical path, Float, PERT, EVM

1. はじめに

本稿では、プロジェクトマネジメントの中でもスケジュール管理に関する手法の提案を行う。大規模かつ複雑なプロジェクトのスケジュール管理では、不規則に発生する遅延により、「どのタイミングで」「どの作業を」「どのくらい短縮すべきか」などの再スケジュールリングの問題が頻繁に発生する。この問題に対し、既存手法では、クリティカルパス(最長作業経路)上の作業のみに着目し、そのパス上の任意の作業を短縮することで解決していた。しかし、「クリティカルパス上のどの作業を短縮すべきなのか」という点に関しては、熟練者の経験や勘に頼ってきたのが既存手法の現状である。そこで、本提案手法では、クリティカルパス上の作業を含む全ての作業に着目し、「どの作業を短縮するのが(先々を考えたときに)より全体工期短縮に効果的なのか」という点を評価することができる指標を提案する。この指標は、既存手法では活用されてこなかった、プロジェクト・スケジュール上の余裕日数「フロート」を活用することによって定量化することができる。本稿では、この指標を「遅延リスク耐性」と呼び、「任意の作業項目が予期せぬ遅延をきたした際に、プロジェクトの全体所要時間が契約納期を超過せずに済む可能性の大きさ」と定

義する。

1.1 研究背景

近年、大規模、複雑化、多様化するプロジェクトの様態は大きな不確定変動を持つ方向に変貌している。作業遅延が不規則に発生し、現行スケジュールのままではプロジェクトに課された時間制約が満たせなくなるため、再スケジュールリングが頻繁に必要となる。

ある Activity の所要時間をプロジェクトの全体所要時間に影響を及ぼさない範囲で遅らせることができるスケジュールの余裕日数をトータル・フロートと言う。トータル・フロートはスケジュールリング計算の結果として生じる余裕日数であるが、プロジェクトの進捗段階では不確定変動による Activity 遅延の影響を受け時々刻々と変化していく。したがって、不確定変動がスケジュールに及ぼす影響は、トータル・フロート(以降「フロート」或いは「スケジュール・フロート」と記す)に現れる。このことから、不確定変動の大きなプロジェクトでは、フロートに基づくスケジュール管理が有効である。

しかしながら、Activity の所要時間を確定的に見積もる従来の PERT/CPM 手法¹⁾²⁾では、フロート(余裕日数)は管理される必要がないものとされ、プロジェクトの全体所要時間に影響を及ぼすクリティカル・パス上の Activity のみ重点的に管理すれば十分とされてきた。しかし、不確定変動の大きいプロジェクトの場合、ある時点においてプロジェクト全体所要時間に影響しない非クリティカル・パス上の Activity (“Non-critical な Activity” と記す)の余裕であっても、フロート日数を消費してしまうと、それがクリティカルパス上の Activity

^{a)}工学専攻機械・情報系コース大学院生

^{b)}教育研究支援技術センター技術専門職員

^{c)}情報システム工学科助教

^{d)}情報システム工学科准教授

^{e)}情報システム工学科教授

(“Critical な Activity” と記す) に変わった途端にプロジェクトの早期完了の可能性を下げてしまう。このため、不確定変動の大きなプロジェクトでは Non-critical な Activity のフロートであっても管理は必要と考えられる。

実際、近年のプロジェクト管理の現場では、スケジュール管理にフロートを活用する動きがある。Critical な Activity のみならず、Non-critical な Activity であっても、それらのフロート間の関係や時間変動を考慮することが合理的な意思決定の助けとなることを経験的に知っているためである。しかしながら、このような現場の状況と認識に学術研究の方は追いついておらず、フロートは管理される必要がないものという従来の認識のままである。或いは、フロートの管理の必要性はわかっているが、従来の PERT/CPM 法での対応は実用レベルでは難しいという認識である。

学術研究においては、フロートを直接管理する以外の不確定変動の問題に対処する手法研究が行われている。具体的に、大きく分けて次の 2 種類が報告されている。一つは、従来のクリティカルパスに基づく方法の確率論的な拡張であり、もう一つは事例や学習用データから経験則を導き出す手法である。

従来の PERT 法の確率論的な拡張として、Goto and Masuda (2007, 2014) は、管理上の指針であるクリティカルパスが移り変わらないことを前提とした静的分析法の提案をしている。また、V.G.Kulkarni, et.al. (1986) や Derek O' Connor (2007) は、クリティカルパスの移り変わりに着目し、クリティカルパスになりやすい作業を特徴づけることを目的とした分析・解析手法を提案している。前者は適用範囲が限定的であり、後者は管理目的としては使用し難いといった問題がある。一方、経験則を導き出す手法としては、リアクティブ・スケジューリングのタイミングに関する意思決定手法として、遅延タスクの個数に基づくスケジュール修正時期の数理モデルが提案されている (Suwa et al., 2003; Fujiwara et al., 2009)。また、バッチプロセスを対象とした研究では、不確定変動への対応策として、バッチ間のバッファ時間のシステムティックな決定法が提案されている (Kawano et al., 2007, 2008)。いずれも実践的だが、経験則を獲得する手法であることから、新規のプロジェクトや学習データが得られにくい対象は手法の適用範囲外である。

以上のように、フロートを活用することで汎用的かつ実践的なスケジュール管理ができる可能性は現場のスケジュール管理者は経験的に知っているが、そのための方法論研究は未だ行われていない。このため、現場のスケジュール管理では、膨大な数の Activity の個々のフロート日数といった部分的かつ限定的な情報から、Activity 間のフロートの相互関係を掌握し、納期超過やそれに伴うコスト超過を最大限回避するようなプロジェクトの再スケジューリングを実現するための意思決定が、全て人の経験と勘で行われているという状況 (※) である。

しかし、従来のリアクティブ・スケジューリングを用いた手法では、いつ再スケジューリングすべきか、どの作業項目をどれだけ短縮すべきか、等を合理的に判断するための方法論は確立されていない。したがって、実際に応用するときには、対象依存の経験則や熟練者の経験や勘にゆだねられているのである³⁾⁴⁾。

以上を踏まえ、本研究室では、スケジュール全体の余裕を統合的かつ合理的に管理するためのモデルや手法の開発を行ってきた。そして、リアクティブ・スケジューリングにおいて熟練者の経験や勘に依存しない合理的な意思決定を支援する方法の提案を行ってきた。

具体的には、まず、スケジュール・フロートのベクトル空間表現法を考案し、この表現方法に基づくスケジュール管理のためのモデルを提案した。本方法及びモデルは、グラフ理論ベースの手法である従来法の PERT に比べ、代数計算の簡便さゆえの高い分析力を潜在的に持つ。更に、本モデルはスケジュール・ネットワークの構造の直感的分かりやすさも PERT と同程度に兼ね備えている。すなわち、本モデルは、不等式制約の集まりとして表されるモデルを成す制約式の一つ一つがスケジュール・ネットワークを成す作業パスに対応付けられ、各制約式の変数の前後関係が作業項目間の順序関係に対応付けられるためである。その上、PERT ではできなかったモデルの自動生成も可能である。従来のこの種のモデルは、いずれも、分析能力は高いが直感的理解が難しい (プロセス代数、MaxPlus 代数、時相論理 など) か、或いはその逆に直感的理解はし易いが分析能力が低い (ペトリネット、ステートチャート、Function Block など) という問題を持っていたが、本提案モデルは分析力と表現力の双方をある程度兼ね備えている点で有用である。

次に、以上のような提案手法やモデル用い、納期制約問題をフロートパラメータベクトルの存在可能領域 (“フロートベクトルの解空間”、或いは“フロートの解空間”と呼ぶ) を求める問題に帰着させる方法を提案した。更に、既存の管理指標 (クリティカルパス、フロート残量 (安全余裕残量)、DRAG 等) を本モデルベースで簡便に計算できる方法を明らかにし、この方法を使って、プロジェクト実行中時々刻々と変化していく重点管理項目を計算し可視化するシステムを開発した⁵⁾⁶⁾。なお、開発したシステムに盛り込まれた重点管理項目可視化のためのノウハウは、汎用性があり、様々な業種のプロジェクトマネジメントで活用可能な知見と考えられる。こういったプロジェクトマネジメントに関するノウハウや手法を体系立ててまとめたガイドブック PMBOK (Project Management Body of Knowledge)⁷⁾ がある。PMBOK は、アメリカに本部のある PMI (Project Management Institute) が中心になって刊行しているが、プロジェクトマネジメントの世界標準といわれている。本研究室で提案されたノウハウは、PMBOK で未だ取り上げられていない、本研究室で新規に見出された実務上役立つ汎用的知見である。

しかしながら、以上の提案、すなわち直感的分かりやすさと分析能力の備わったモデルの構築、及び重点管理項目可視化ノウハウの提案は、有意義ではあるが、冒頭で目標として述べた「スケジュール全体の余裕を統合的に扱う方法」としては不十分である。個々の作業項目の余裕 (フロート) という断片的な情報からスケジュール全体余裕の大きさ、ばらつきを推し量る手助けはするが、それらの大きさ、ばらつきを直接顕在化させるには至っていない。そのため、PERT でもできる分析をより簡潔に行えるようなモデル・手法の提案レベルに留まっている。

では、何が出来れば全体余裕を直接扱える手法/モデルと

言えるのか。これを遅延リスク解析の場合即ち、個々の作業項目の遅延が全体余裕へ及ぼす影響の度合い（“遅延リスク耐性”と呼ぶ）を解析する場合で説明する。遅延リスク解析で、もしフロートの大きさのみならずフロート間の独立・従属関係によって生じる影響も結果に反映されるような解析ができるのであれば、全体余裕を直接扱える手法／モデルといえる。実際、フロート値が同じ作業項目同士であっても、遅延リスクの大きさは同じではない。全体的により長い作業パスによりたくさん乗っている作業項目ほど遅延リスク耐性へ及ぼす影響は大きい(5.4節)⁸⁾。こういった遅延リスク耐性の見極めは、大規模プロジェクトを完遂させる上で非常に重要である。全体所要時間を同じ日数分短縮させる場合でも、遅延リスク耐性への影響力がより大きい作業項目を短縮したほうが、全体余裕を大きく維持できるためである。しかしPERTではこの種の分析をカバーできていない。それゆえプロジェクトの現場では、個々のフロートの断片的情報のみから遅延リスク耐性を見極めを経験や直感を頼りに行っているのである（(※)参照）。一方、本研究室で提案されたフロート解空間をうまく活用すると、フロート間の独立・従属性による影響が反映された遅延リスク解析が可能となる。具体的には、一作業項目の遅延による一フロートの減少が他フロート全体へ及ぼす影響を、解空間の縮退率及び形状の変化として定量的かつ直接的に捉える可能性があるということである。これは、フロー解空間が座標系で表現される際、各座標軸が互いに独立なフロート（の集まり）に対応付けられるためである。

以上のことから、本研究では、まず、プロジェクト・スケジュールの全体余裕そのものの大きさやばらつきを直に知る方法として、フロートの解空間、すなわち納期制約問題で求められるフロートパラメータベクトルの存在可能領域自体の大きさと形状を求める方法について検討した。しかし、フロートの解空間の体積を求める方法の一般化は原理的には困難とわかった。本問題は、 N 次元凸多面体 ($1 \leq N$) の体積（高次元の場合、“超体積”と呼ぶ）を求める問題に帰着されるが、超体積計算の一般化・自動化は原理的には困難ということである。そこで、体積計算に代わる方法として、 N 次元凸多面体に N 次元格子を被せオーバーラップする部分の格子点数をカウントし、大きさを見積もることとし、この方法を自動化した。ところが、この方法は計算量の問題が大きい。フロート解空間の次元増加に対し、指数オーダーで増加する。具体的に、例えば、15次元のフロート解空間の大きさを格子幅1の格子点で数え上げた場合、最大で20時間程度かかる。現在、Activity数が1000程度の実務レベルのプロジェクトへの適用を目指しているため、この方法は現実的ではない。

そこで本研究では、フロート解空間の大きさそのものを算出させるといった厳密性を多少犠牲にする代わり、実用的な計算時間で求められる評価指標を定義・導入することとした。具体的には、少なくとも上述の格子点数の大小関係は保持するような指標を見出すための検討を行った。遅延リスク耐性を(※)で述べた現状問題に寄与させるためには、フロート解空間そのものの大きさでなくとも、解空間同士の大小関係が分れば良いためである。

指標候補として、フロート解空間を成す N 次元凸多面体の L_p ノルムや原点-重心間の距離など試した中、群を抜いてよい

結果を示したのが N 次元凸多面体に内接する最大の N 次元単体 (N -Simplex) の超体積である。原点周りの凸多面体に限定すれば、 N 次元単体の超体積は、原点から各座標軸上の点までの距離の直積 D に対し $D / N!$ として求められ、計算量は $O(1)$ となり数秒で計算可能となる。

1.2 本論文の構成

以下では、2節で研究目的と課題を述べ、3節でプロジェクトマネジメントの基本事項及び、プロジェクト進捗段階での予期せぬ遅延がスケジュールに及ぼす影響（リスク）に関して概説する。そして、4節で本研究室での提案事項であるスケジュール・フロートのベクトル空間表現方法や重点管理項目可視化手法について説明した後、5節で本稿での提案事項である遅延リスク耐性の評価方法について示すとともに、提案方法の妥当性を調べるために行った検証実験について示す（建築プロジェクト、発電設備の分解点検修理プロジェクト（オーバーホール）を適用例として用いる。）。また6節では提案手法のリスクマネジメントへの活用について示す。この6節の目的は、この種の管理手法としては比較的新しい、プロジェクト実施段階での時間とコストの進捗管理を一元化できる“アーンドバリュー法 (EVM: Earned Value Method)”⁹⁾ について示すとともに、EVMではできない遅延リスク管理手法のための手法 RDM (Resistance to schedule-delay Management) を提案をする。RDMでは遅延リスク耐性を活用するが、本稿ではこれをEVMと併用できる形で提案する。こうすることにより、プロジェクト実施段階での時間とコストの進捗と併せて（現況スケジュールの）遅延リスク耐性の評価が可能になる。更に7節では、建築工事プロジェクトを適用例として用いたケーススタディについて示し、最後に8節で結論を述べる。

2. 研究目的

2.1 研究目的

本研究目的は、任意の作業項目（以下“Activity”と同義）が予期せぬ遅延をきたした際に、プロジェクトの全体所要時間が契約納期を超過せずに済む可能性の大きさ（遅延リスク耐性）を定量的に評価できるようにすることである。言い変えると、個々の作業項目の遅延が全体余裕へ及ぼす影響の度合いを遅延リスク耐性の大きさとして定量的に評価できるようにすることである。

なお、本稿では、納期とは「発注者との間で契約された納期」の意味で用い、全体余裕とは「プロジェクト全体所要時間の（納期に対する）余裕」の意味で用いる。

2.2 研究課題

研究課題は以下の通りである。

- 「遅延リスク耐性」を概念的定義し、定量化し、算出方法を明らかにする。
- 「遅延リスク耐性」算出の計算量問題に対処するための指標化について検討する。
- 指標化された「遅延リスク耐性」の妥当性の有無を検討するため、当該指標の精度評価実験を実施する。

- 提案する遅延リスク耐性評価指標を従来のプロジェクト管理手法である“アーンドバリュー法 (EVM: Earned Value Method)”と組み合わせて使用することの効果の有無を実験的に確かめる。

精度評価実験の適用例として、建築プロジェクト、発電設備の分解点検修理 (オーバーホール) プロジェクト、および JAPEX モデルを使用する。なお、各々の適用例の特徴は順に、Activity 数 30 程度の中規模プロジェクト、Activity 数が 100 以上の比較的大規模のプロジェクト、Activity 数は中程度だが並行作業パス同士の所要時間が拮抗しておりクリティカルパスの入れ替わりが頻繁に起きるプロジェクトとなっている。

JAPEX モデルとは、中東・北アフリカ地域において中規模ガス処理プラントを建設する EPC フルターンキー・プロジェクトを想定し、実情に基づき作成された研究用モデルのことである。

3. プロジェクト・マネジメントについて

3.1 プロジェクトについて

3.1.1 プロジェクトの特徴

プロジェクト¹⁰⁾とは、独自の製品やサービスを創造するためになされる特別な活動・取り組みのことである。また、ある取り組みがプロジェクトと呼ぶに足るものであるための最も重要な要件は以下の3つである。

1. 特別な目的がある
2. 期限が限られている
3. 資源が限られている

更にもう少し具体的に、プロジェクトと呼べる活動・取り組みの特徴を書くと、以下のようになる。

1. 独自の、或いは新たな目的を持つ活動・取り組みである
2. 目的達成の手段やプロセスが一通りではない
3. 達成期限やリソース制約がある
4. 簡単に達成できる取り組みではない
5. 特に近年のプロジェクトは大規模複雑化しているため目的達成が容易ではない
6. そのためチームプレイで組織的に取り組まねば達成が難しい

等

3.1.2 プロジェクト管理

3.1.1 節から、プロジェクトでは、(A) その都度の様々な条件や起こり得るリスクを考慮し、目的達成の確実性を評価しながらなるべく合理的な手法で実施計画を立案する必要がある。しかし、プロジェクトでは事前に分らないことが多々ある。このため、いくら科学的な手法で計画立案されとしても、推測や仮定に基づく部分を排除できない。このため、(B) プロジェクトでは頻繁に進捗状況を監視し、計画に基づき進捗を評価し、計画の軌道修正を行う必要がある。つまり、プ

ロジェクトのマネジメントは、(A) 計画立案段階、(B) 計画実施段階の双方で重要ということである。

3.2 プロジェクト実施段階におけるマネジメント

3.1 節で述べたように、プロジェクト・マネジメントは、(A) 計画立案段階と (B) 実施段階の2段階に大きく別けられるが、本研究は (B) 実施段階のマネジメントをサポートする手法の提案であるため、以下ではフェーズ (B) に焦点を当てた議論をしていく。

3.2.1 コントロールとマネジメントの違い

従来の管理はコントロール (Control) と呼ばれていた。コントロールは現場の統制に重点が置かれていた。具体的には、制御命令と観測結果を比較評価し、次の制御命令を決定するという形態であった。これに対し、コントロールの進化した形であるマネジメント (Management) は、計画と評価を含めたプロセスを重視した管理の形態をとる。具体的には、計画と実施結果を比較評価し、目的達成にとって効率良い方向で計画を修正する形態である。本稿の以下では、“管理”は“マネジメント”と同義とする。

3.2.2 プロジェクト実施段階での管理・評価

実施段階で管理・評価すべき対象は、主に工程、コスト、品質であるが、以下、工程とコストに限定し議論する。現在の管理手法では品質の評価は組み込まれていない。このため工程に関する評価とは「(実施計画に対して) どれだけ計画通りにできているか」の評価であり、コストに関しては「(計画予算に対して) どれだけ予算内で実施されているか」の評価である。従来これら (工程評価とコスト評価) は別々のシステムで計算され、これらの結果に基づく総合評価・判断は人 (管理者) の判断 (直感・経験則) に頼っていた。

以下では、プロジェクトの現状が順調かどうかを評価するための二つの考え方を紹介する。

3.2.3 進捗度評価

進捗度評価というのは、文字通りプロジェクトの進み具合を評価するもので、従来、工程管理の一環として行われてきた。つまり、工程の計画をベースにして、どこまで完了したのかを調査し、それがプロジェクト全体からみてどの程度の割合となるのかを把握することが進捗度評価ということになる。

例えば、住宅建築のようにモノを作るプロジェクトでは、出来上がったモノの大きさや量で評価することができる。また、システム開発など形や量で表しにくいプロジェクトでは、完了した活動の価値を金額で換算することにより進捗度を評価することができる。このようにして、進捗度評価に用いる指標を出来高 (earned value) と呼ぶ。

3.2.4 達成度評価 (工程進捗とコストの総合評価)

出来高を指標として用いた進捗度評価では、コスト変動が大きいプロジェクトなどにおいて、現況を見誤るケースも起こりうる。事実、プロジェクトマネージャーは工程だけでなく原価や品質などを総合評価して、プロジェクトが順調に進んでいるのかどうか判断している。そこで、工程だけでなく原価要素も加味してプロジェクトの現況を評価しようとする考え方が 1970 年頃から米国で開発されてきた。これが達成度評価である。

達成度評価では、“どれだけできたか”という量的な判断だけではなく、それにかかったコストを併せて評価する。このため、“工程は順調だが原価はオーバーしている”とか“原価は計画通りだが工程が遅れている”などというより総合的な判断ができるようになってきている。そして、この達成度を求める手法が、6. 節で紹介する EVM (アーンドバリュー法) である。

4. 本研究室での先行研究

本研究室の先行研究では、まずプロジェクト・スケジュールの全体余裕を統一的かつ合理的に管理するための手法として、スケジュール・フロートのベクトル空間表現法が考案された。また、本手法に基づくスケジュール管理のためのモデルが提案された。また、既存の管理指標(クリティカルパス、フロート残量(安全余裕残量)、DRAG等)を本モデルベースで簡便に計算できる方法が明らかにされた。また、この方法を使って、プロジェクト実行中時々刻々と変化していく重点管理項目を計算し可視化するシステムが開発された⁵⁾⁶⁾。

4.1 スケジュール・フロートのベクトル空間表現

に基づく工程管理モデル

スケジュール・フロートのベクトル空間表現法及び、この表現方法に基づくスケジュール管理のためのモデルについて示す。本手法は、グラフ理論ベースの手法である従来法の PERT に比べ、代数計算の簡便さゆえの高い分析力を潜在的に持つ。更に、本モデルはスケジュール・ネットワークの構造の直感的分かりやすさも PERT と同程度に兼ね備えている。

4.1.1 フロートのベクトル空間表現

プロジェクトが Activity network で表されるとき、個々の Activity の時間的余裕をパラメータ表示し(“余裕パラメータ”と呼ぶ)、納期制約問題をパラメータベクトルの存在可能領域(“フロートの解空間”と呼ぶ)を求める問題に帰着させる。フロートの解空間は、個々の Activity の所要時間と余裕、Activity 間の先行関係、及び納期制約に基づき機械的な手順で求められ、余裕パラメータから成る係数 1 の不等式制約の集まりとして表される。各不等式制約はクリティカルパスに成り得る経路の経路長制約に対応付けられ、各式のパラメータの並び順は各経路の作業順序を表す。また、モデルの本質とは関係ないが、数式モデルにありがちな構造的理解のし難さを回避するための取り決めとして、各式のパラメータの並び順はスケジュール・ネットワーク上の作業項目の生起順序と定義した。また、作業項目とそれに付随する余裕パラメータとは、同じアルファベットの大きい文字と小さい文字で表記させることとした。このため、実数ベクトルで処理できる利点である高い分析力と計算力はもちろんのこと、ネットワークの構造的理解も従来 PERT と同程度には可能なモデルとなっている。

4.1.2 ベクトル空間表現モデル“フロートの解空間”の生成方法

フロートの解空間は、所与のスケジュール・ネットワークから、各作業項目の生起順序と作業時間を抽出し、各作業項目の時間的余裕を表す余裕パラメータを導入し、生起順序と生起時刻を同時に扱える max-plus 代数¹¹⁾ を使用し、各作業

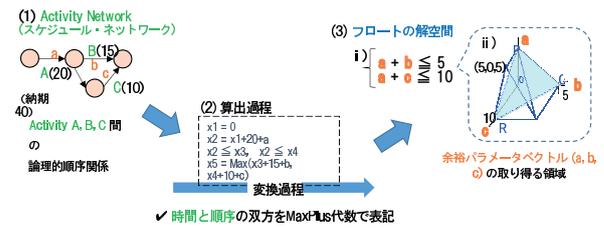


図 1: フロートの解空間生成の流れ

項目の終了までにかかる時間を順に求めていく。具体的には以下のように求められる。

$$\text{作業 } p \text{ 終了までに掛かる時間} = \text{Max}\{p_j \text{ の終了時刻} \mid p_j \text{ は } p \text{ の先行作業全て}\} + p \text{ の所要時間} + p \text{ の余裕パラメータ}$$

なお、最終作業項目の終了までに掛かる時間は、即ちプロジェクト全体所要時間を意味する。したがって、上記方法で得られる“最終作業終了までに掛かる時間”に対し、納期制約を与えて得られる制約条件式「最終作業終了までに掛かる時間 ≤ 契約納期」がフロート解空間の構成要素となる。ただし、最終作業項目の終了までに掛かる時間は、初期作業項目から最終項目へ至る経路の数だけ得られる。従って、所望のモデルは、上述の制約条件式全体として得られる。

4.1.3 フロートの解空間の生成

図 1-(1) のように、Activity A, B, C があり、所要時間がそれぞれ 20 日,15 日,10 日とする。A と B、A と C は直列に繋がりが、B と C は並行である。() 内は Activity の標準的所要時間を表す。x1 と x2 で Activity A の開始時刻と終了時刻、x3 と x5 で B、x4 と x5 で C の開始時刻と終了時刻を各々表し、小文字アルファベットで同大文字の Activity の余裕パラメータを表すと、所定の計算過程を経て(図 1-(2))、本例のフロートの解空間が(図 1-(3)) のように得られる。

あるシステム製品をおさめる仕事を表す簡単な例を考える(図 2 参照)。図 2 の Activity Network の () 内は各 Activity の標準的所要時間を表し、アルファベット大文字は作業項目名を表す ID、同小文字は当該作業項目の余裕パラメータを表す。つまり、例えば、「基本設計」を指す ID は A であり、標準的所要時間は 20、余裕パラメータは a である。契約納期を 80 とすると、本例のフロートの解空間は以下の連立不等式で表される。

$$\begin{cases} a + b + d + f \leq 5 \\ a + c + e + f \leq 15 \end{cases} \quad (1)$$

具体的には、まず、スケジュール・ネットワーク(Activity Network) から、各作業項目の生起順序と作業時間を抽出した工程表を作成し、各作業項目の時間的余裕を表す余裕パラメータを導入する。次に、作業項目の生起順序と生起時刻を同時に扱える max-plus 代数を使用し、各作業項目の終了までにかかる時間を順に求めていく。例えば、F の終了時刻は以下のように求められる。

あるシステム製品をおさめる仕事



図 2: フロート解空間生成例

F 終了までに掛かる時間 = Max{D の終了時刻、E の終了時刻} + F の所要時間 + f (F の余裕パラメータ)

ただし、余裕パラメータ f の取る値は、F 実施期間の前後の余裕の和である。具体的には、ここでは、実施前の余裕とは既に作業可能だが何らかの理由で開始時間を先延ばしにしている処理待ち時間とし、実施後の余裕とは当該処理の遅延時間としている。なお、最終作業項目 F の終了までに掛かる時間は、即ちプロジェクト全体所要時間を意味する。従って、プロジェクト全体所要時間は 75+a+b+d+f、65+a+c+e+f と表せ、所望のモデルは、納期制約 80 を与えた {75+a+b+d+f ≤ 80、65+a+c+e+f ≤ 80} となり、結果として上記の (4.1) が得られる。

4.1.4 フロート解空間に基づく管理指標の計算

まず、フロート (ここではトータルフロート) は、プロジェクト全体所要時間が契約納期を超過しない範囲でとれる作業項目の余裕日数の最大値であるが、これはフロートの解空間を表す不等式制約の集まりに対し、着目パラメータ以外の余裕パラメータの各々に、それが取り得る最小値を代入することで求められる。具体的には、実行済の作業項目の余裕パラメータには実データ (処理待ち時間+遅延時間) を与え、未実行項目の余裕パラメータには 0 を与えれば良い⁵⁾。

また DRAG¹²⁾¹³⁾¹⁴⁾ は、クリティカルな作業項目がプロジェクト全体の工期を短縮できる日数の最大値であるが、これはクリティカルな作業項目と並行する経路の有無やその並行経路のフロート値を基に算出されるのだが、これら DRAG 算出に必要なデータ取得のための計算プロセスは PERT では煩雑になり易い。しかし、提案手法のフロートの解空間を使うと簡潔かつシステマティックに算出できる⁶⁾。

4.2 重点管理項目の可視化手法

まず、本稿での前提 (スケジュール管理上の利便性を考慮したフロートの扱い) について述べる。

時々刻々と変化していく各作業項目のフロートに基づき、クリティカルパスの所在とスケジュール全体の安全余裕残量の変遷を可視化する手法について示す。安全余裕残量とは、プロジェクト・スケジュールの全体余裕を表し、クリティカルパス上の作業項目のフロート値がそれに相当する。なお、通常フロート計算では、クリティカルパス上の作業項目のフロートが 0 となるように、他の作業項目のフロートはそれを基準に数値化される。しかし、本稿では、スケジュール管理

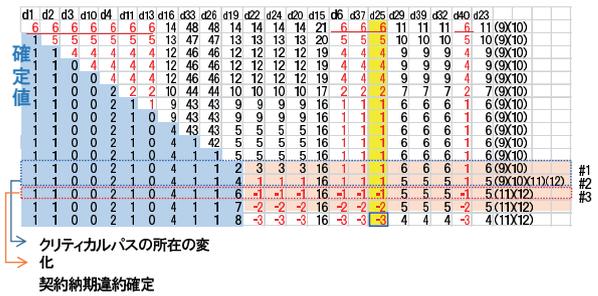


図 3: クリティカルパスと安全余裕残量の移り変わりの可視化のイメージ

上の利便性より、クリティカルパス上の各作業項目のフロートが、契約納期に対する余裕となるようにしている。こうすることにより、あとどれだけ (何日) 遅れたら契約納期違約に至るかがダイレクトにわかるためである。

4.2.1 可視化方法

所与のスケジュールが示す作業項目間の順序制約に従って進められるプロジェクトの実行中、不規則に発生する作業遅延により、各作業項目の余裕が不規則に変化していく。従来、このような変化は、個々のフロートの変化としてばらばらに捉えられていたが、作業項目同士の有機的繋がり (余裕パラメータベクトル) の変化として捉えることにより、クリティカルパスの所在と安全余裕残量の変遷が可視化が可能となる。具体的には、各時点での余裕パラメータベクトルの成分 (各作業項目のフロート) を横に並べて行ベクトル表示し、各作業項目の余裕パラメータの時系列を各々縦に並べて列ベクトル表示させる。更に、各行での成分の最小値の全てをハイライト表示させる。こうすることにより、各時点でのクリティカルパスの所在は、各行でハイライト表示された作業項目の成分の並びとして顕在化する。また、各行のハイライト表示された数値として安全余裕残量の変遷が顕在化する。

図 3 は、作業項目 23 の建築プロジェクトの実施段階で本手法を適用した結果得られた余裕パラメータベクトルの時系列データである。各行の成分の最小値は赤字で表示されているが、これは各時点でのクリティカルパス上の作業項目のフロートを表している。時系列データの #1 行から #2 行では、この赤字の並び位置が変化しているが、これはクリティカルパスの所在変化が起きていることを表している。具体的には、赤字の並び位置が、d6,d37,d25,d40 から d22,d24,d20, d6,d37,d25,d40 へ変化している。このため、これら余裕パラメータ並びに対応付けられる作業パス D6,D37,D25,D40 から D22,D24,D20, D6,D37,D25,D40 へクリティカルパスが移り変わったことを表している。また、各行の最小値を縦方向に辿ると安全余裕残量の変遷がわかる。例えば図の黄色部分は安全余裕残量の変遷にあたる。なお、負値の出現する #3 行では、このタイミングで契約納期違約確定したことを意味する⁵⁾⁶⁾。

以上、本節で示したような重点管理項目を可視化させるためのノウハウは、汎用性があり、様々な業種のプロジェクトマネジメントで活用可能な知見と考えられる。こういったプロジェクトマネジメントに関するノウハウや手法を体系立ててまとめたガイドブック PMBOK (Project Management Body

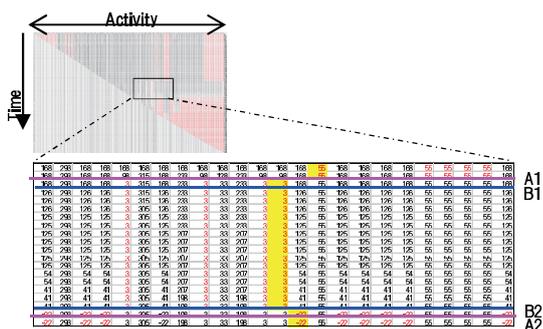


図 4: 重点管理項目を視覚化するプロトタイプシステムの実行結果

of Knowledge) ⁷⁾がある。PMBOK は、アメリカに本部のある PMI (Project Management Institute) が中心になって刊行しているが、プロジェクトマネジメントの世界標準といわれている。本節で提案した可視化手法は、PIMBOK で未だ取り上げられていない、新規に見出された実務上役立つ汎用的知見と考えられる。

4.2.2 重点管理項目可視化実験

以上のような可視化の仕組みを自動化し、管理項目可視化のためのプロトタイプシステムを実装し、住宅建築 ¹⁵⁾ や電力設備分解点検修理 (オーバーホール) といった現実的なプロジェクト事例への適用実験を通し、本方法の妥当性を確認した。本システム実行結果の一例として図 4 を示す。図 4 は作業項目 130 程度のオーバーホールプロジェクトのネットワーク・ダイアグラムに本手法を適用した結果得られた余裕パラメータベクトルの時系列データである。各行の成分の最小値、つまりクリティカルパス上の作業項目のフロートは赤字で表示されている。この赤字の並びが変化すると A1 行から B1 行、及び B2 から A2 では、クリティカルパスの所在変化が起きていると分かる。また、各行ベクトルの最小値を縦方向に辿ると安全余裕残量の変遷がわかり、負値の出現する A2 行では契約納期違約確定したことが分かる。

5. 提案手法

良いプロジェクト・スケジュールを立案するためには、ネットワーク・スケジュールの評価と改善ができなくてはならない。1. 節で述べたように、従来の管理手法では、評価の尺度として、「全体工期の長さ」、つまり、クリティカルパスの長さという一指標しかなかった。しかし、最近のプロジェクトでは、作業遅延が不規則に発生し、クリティカルパスの移り変わりが多いため、従来どおりの全体工期の長さのみを指標とする手法では不十分である。不確定変動の大きいプロジェクトの場合、ある時点において、プロジェクト全体所要時間が契約納期を超過することにはならない非クリティカルパス上の Activity (Non-critical な Activity) の余裕であっても、フロート日数を消費してしまうと、それがクリティカルパス上の Activity (critical な Activity) に変わった途端にプロジェクトの早期完了の可能性を下げってしまうためである。

そこで、本研究では、個々の作業項目の遅延がプロジェクト・スケジュールの全体余裕へ及ぼす影響の度合いを“遅延リ

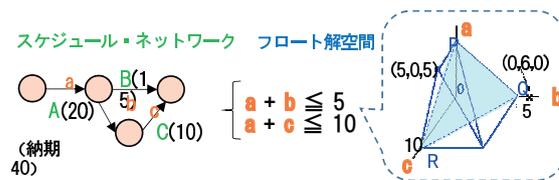


図 5: フロートの解空間の例

スク耐性”と呼び、遅延リスク耐性の大きさをフロートの解空間の大きさに基づき評価する方法を考案した。遅延リスク耐性の大きさは不確定性の大きいプロジェクト・スケジュール管理上重要な一尺度となり得る。

5.1 評価指標「遅延リスク耐性」の提案

5.1.1 遅延リスク耐性の概念的定義

5. 節冒頭で述べたように、任意の Activity が予期せぬ遅延をきたした際にプロジェクト全体所要時間が契約納期を超過せずに済む可能性の大きさを「プロジェクト全体の遅延リスクへの耐性」と定義し、“遅延リスク耐性”と略記する。

5.1.2 遅延リスク耐性の定量化

フロートの解空間上の各点は、「プロジェクトの全体所要時間を変えない Activity レベルでの遅延パターン」(“ADP”と記す)に相当する。このため、遅延リスク耐性の大きさをフロートの解空間の大きさで評価することは妥当と考え、遅延リスク耐性の大きさを ADP で評価することとした。

4.1.3 節で示したフロート解空間の例を図 5 に再掲する。本解空間を示す凸多面体の境界及び内部の点は全て ADP に対応付けられる。例えば、境界上の点 (5,0,5) は ADP に対応付けられる。実際、Activity A と C が共に 5 日遅延してもプロジェクト全体所要時間は納期 40 日を超過しない。一方、本解空間からはみ出ている点 (0,6,0) は ADP に対応付けられない。実際、Activity B が 6 日遅延すると、たとえ A,C の遅延が 0 であったとしても、全体所要時間は納期 40 日を超過する。

5.2 遅延リスク耐性の算出と問題点

5.2.1 格子点数による評価

フロートの解空間は、係数が +1 の連立一次不等式で表されることから第 1 象限上の原点周りの N 次元凸多面体 (1 ≤ N) を成す。従って、遅延リスク耐性の評価のためには N 次元凸多面体の超体積を求めればよい。しかし、N 次元凸多面体の体積計算は一般化できない。そこで一般化可能な方法として N 次元格子の格子点数として評価することとした。即ち、ある格子幅の N 次元格子をフロート解空間に被せ、当該解空間に含まれる格子点の数をカウントし評価することとした (図 6) ¹⁶⁾¹⁷⁾。

ただし、フロート値は一経路 (分岐・合流点から次の分岐・合流点までのパス) の中では一定になる性質がある。したがって、フロート解空間を成す余裕パラメータは互いに独立ではない。そのため、上述のような格子点数上げをするためには、まず余裕パラメータ同士の独立・従属関係を調べ、当該解空間のベクトル空間上での次元を明らかにする必要がある。なお、パラメータ同士の独立・従属関係は、シンプレックス法

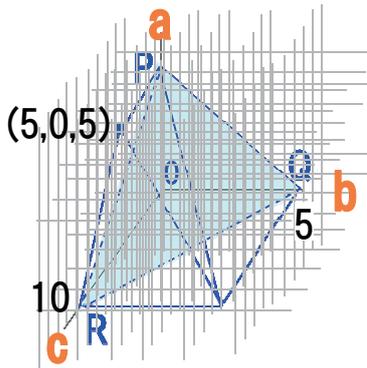


図 6: フロートの解空間に整数格子を被せたイメージ

で凸多面体の端点を求めるための計算過程から機械的な手順で検出可能であることから、本研究では、シンプレックス法¹⁸⁾を用いて本手順を自動化した。

5.2.2 フロートの解空間の大きさを求める手順

1. 4.1.2 節の方法でフロート解空間を表す不等式制約の集まりを求める。
2. 1. の不等式制約を成す余裕パラメータ全体で、フロートが一致する余裕パラメータの集まりを、互いに従属関係にあるパラメータの集まりとして検出し(シンプレックス法使用)、パラメータの統合と式の整理を行い、互いに独立なパラメータから成る不等式制約の集まりとしてのフロート解空間の表現形式(※ 5-1)を得る。
3. フロート解空間に含まれる格子点の数を求める。
 - 対象とするフロート解空間 (N 凸多面体) の端点を求め、これら端点を手がかりに、解空間を含む出来るだけ小さな領域を切り出す。
 - 切り出された領域内に含まれる格子点各々に関し、フロート解空間に含まれるか否かを 1 点ごとにチェックし含まれる点をカウントしていく。1 点ごとのチェックは、具体的には、不等式制約に当該点の座標を代入し、これが制約条件式を満たすかどうかでチェックしていく。制約条件式を満たせば当該点はフロート解空間に含まれる点であり、満たさなければ含まれない点である。

なお、以下では、(※ 5-1) のようなパラメータの統合・整理されていない不等式制約を統合・整理前のフロート解空間、統合・整理後の不等式制約を単にフロート解空間(工程管理モデル)と呼ぶ。

5.2.3 遅延リスク耐性(フロート解空間の大きさ)を求める例-1

まず簡単な例を示す(図 7 参照)。A, B, C の 3 つの activity があり、所要時間がそれぞれ 5 日、3 日、2 日とする。A-B-C が直列につながる場合(図 7-(1))、余裕パラメータ a,b,c は互いに従属関係にあり、フロート解空間は 1 次元となるため、格

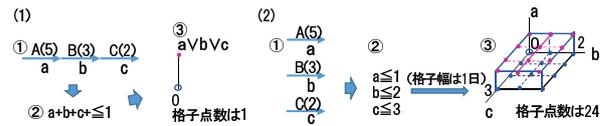


図 7: 遅延リスク耐性(フロート解空間の大きさ)の算出(簡単な例)

子幅 1 でカウントすると、そのフロート解空間の大きさは 1 となる。一方、A, B, C のすべてが並行の場合(図 7-(2))、余裕パラメータ a,b,c は互いに一次独立関係となり、フロート解空間は 3 次元の凸多面体となるため、格子幅 1 でカウントすると、フロート解空間の大きさは 24 となる。

5.2.4 遅延リスク耐性(フロート解空間の大きさ)を求める例-2

4.1.3 節の適用事例(図 2 参照)に対し、上述の“フロートの解空間の大きさを求める手順”で、その大きさを求めると次のようになる(図 8 参照)。

1. 4.1.3 節の(4.1)より、統合・整理前のフロート解空間は(図 8-(0))である。
2. (図 8-(0))に含まれる余裕パラメータ a,b,c,d,e,f,g のうち、互いに従属関係にある余裕パラメータの組は、(a,d)、(b,c)、(e,f)であり、これらを順に a,d = A、b,c = B、e,f = C のようにパラメータ統合し、式を整理すると、統合・整理後のフロート解空間は(図 8-(1))のような 4 次元領域となる。この領域は、g の値を任意の定数で固定すると(図 8-(1'))のように図示できる。
3. 格子幅 1 の整数格子を解空間(図 8-(1)、(1'))に被せ、解空間に含まれる格子点の数をカウントする。具体的には以下のように行う。
 - まず、当該フロート解空間の端点を求めると、(5,0,0)、(0,5,0)、(0,0,15)、(5,0,10)、(0,5,15)となる。したがって、当該解空間を成す点は、A={0,1,2,3,4,5}、B={0,1,2,3,4,5}、C={0,1,2,...,15}、の全ての組み合わせ{(0,0,0)、...、(5,0,0)、...、(0,5,0)、...、(0,0,15)、...、(5,0,10)、...、(0,5,15)}(図的には(図 8-(1'))の破線の直方体)の中に含まれると判る。
 - これらの点が当該解空間の制約条件式を満たすか否かを個別にチェックし、制約条件式を満たす点を当該解空間に含まれる点としてカウントしていくと、当該フロート解空間に含まれる格子点数は 301 とわかる。

5.2.5 格子点数による評価の問題点

本方法には計算量の問題がある。実際、N 次元凸多面体の格子点数上げに掛かる計算量は次元の指数オーダーとなる。具体的に、15 次元のフロート解空間の分析において、15 次元凸多面体の格子点を格子刻み幅 1 日で数え上げた場合、最大で 20 時間程度かかった(5.4.3 節)。現在、Activity 数が 1000 程

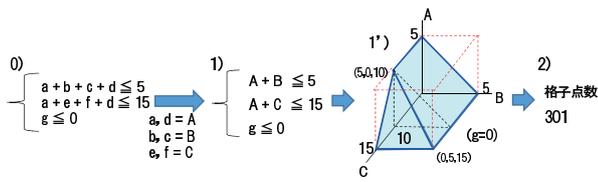


図 8: 遅延リスク耐性 (フロート解空間の大きさ) の算出 (4次元の例)

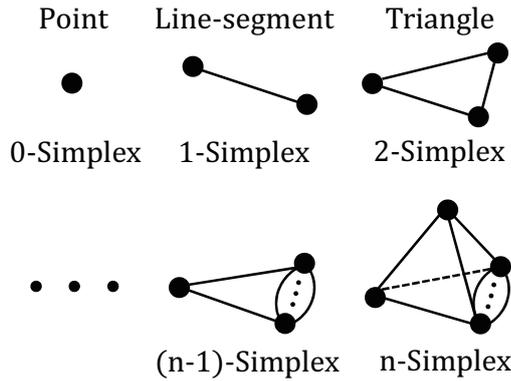


図 9: 0次元単体 (一点) から n次元単体までの概形図

度の実務レベルのプロジェクトへの適用を目指しているため、この方法は現実的ではない。

5.3 遅延リスク耐性の指標化について

計算量の問題を回避するため、本研究では、フロート解空間の大きさそのものを算出させるといった厳密性を多少犠牲にする代わり、実用的な計算時間で求められる評価指標を定義・導入した。具体的には、個々の Activity の遅延が全体所要時間へ及ぼす影響の度合いを知るためには、フロート解空間を成す N次元凸多面体の超体積そのものでなくとも、N次元凸多面体同士の大小関係が分れば良いという考えである。そこで、少なくとも格子点数の大小関係は保存するような指標を見出すための検討を行った。

5.3.1 N次元単体の超体積に基づく評価

指標候補として L_p ノルムや原点-重心間の距離など試した中、群を抜いてよい結果を示したのが「N次元凸多面体に内接する最大の N次元単体 (N-Simplex) の超体積」(「NdSV」と記す) である¹⁹⁾。

n次元単体 (n-Simplex) とは、(n+1) 個のアフィン独立な点によって作られる図形のことである。例えば、0次元単体は一点、1次元単体は線分、2次元単体は三角形、3次元単体は四面体、n次元単体は n+1 胞体・・・のようになる (図 9)。

単体は、高次元でも体積計算の一般化が可能で、計算量の問題がない。一般に N次元単体の超体積 NdSV は次の式で求められる。

$$NdSV = \frac{\sqrt{\det[L^T L]}}{N!}$$

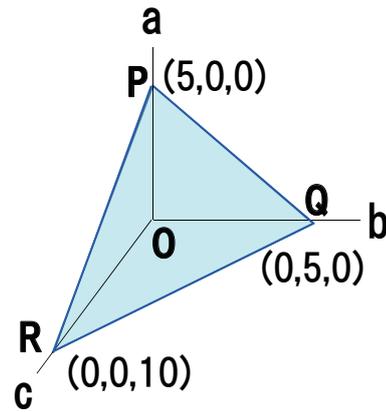


図 10: 3次元単体の一例

L: N次元単体のある頂点から出る互いに線形独立な n本の有向辺の方向ベクトル (l_1, \dots, l_n) を各列成分に持つ $m \times n$ 行列

5.3.2 N次元単体の超体積の算出例

図 10 のような 3次元単体の頂点の一つである原点 O から出る互いに独立な方向ベクトルは

$l_1 = P - O = (5, 0, 0)$ 、 $l_2 = Q - O = (0, 5, 0)$ 、 $l_3 = R - O = (0, 0, 10)$ となるため、

$$L = (l_1, l_2, l_3) = \begin{pmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 10 \end{pmatrix} \text{ となり、} L^T =$$

$$\begin{pmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 10 \end{pmatrix} \text{ となる。}$$

したがって、超体積は以下のように求められる。

$$\begin{aligned} NdSV &= \frac{\sqrt{\det[L^T L]}}{3!} \\ &= 250/3! \\ &= 41.666\dots \end{aligned}$$

本稿で扱う単体超体積は、上の例のような、原点を頂点の一つとする第 1 象限上の凸多面体である。従って、その超体積は常に、各座標軸の切片の大きさの直積 D を次元数 N の階乗で割った $D / N!$ として求められる。従って、もし NdSV が遅延リスク耐性の大きさを測る指標として妥当ならば、遅延リスク耐性評価は計算量 $O(1)$ となり数秒で可能となる。

5.4 遅延リスク耐性の精度評価実験

NdSV が遅延リスク耐性評価指標として利用可能かどうか調べるための評価実験を行った。具体的には、Activity network の個々の Activity の遅延がフロートの解空間に与えるインパクトの大きさを、フロートの解空間に含まれる格子点数の減少率と NdSV の縮小率の双方で評価した。そして、格子点数減少率の降順の Activity 列と、NdSV 縮小率の降順の Activity 列の順序関係が一致するかどうかを調べた。このような一致

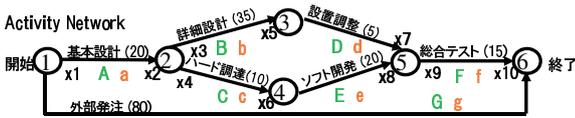


図 11: 「あるシステム製品をおさめる仕事」の Activity Network

遅延パターン	格子点数	単体体積
遅延無し	1692	1666
D_a が1遅延	1260	1152
D_b が1遅延	1344	1224
D_c が1遅延	1584	1568
g が1遅延	1128	833

解空間縮小率大
(遅延リスク耐性へのインパクト大)

 単体体積 : $g > D_a > D_b > D_c$
 格子点数 : $g > D_a > D_b > D_c$

図 12: 各 Activity ごとの 1 日遅延当たりの解空間の縮小率の比較 (例)

が任意の Activity network で確認されるならば、格子点数による評価の代わりに NdSV による評価が可能となり、5.2.5 節の計算量の問題は解消されることとなる。

5.4.1 あるシステム製品をおさめる仕事を表す簡単な例

4.1.3 節の図 2 で示した、「あるシステム製品をおさめる仕事を表す簡単な例」に Activity 「外部発注」を追加した Activity Network を図 11 に示す。契約納期を 80 とすると、本例のパラメータ統合・整理前のフロート解空間は 7 つの余裕パラメータから成る以下の連立不等式で表される。

$$\begin{cases} a + b + d + f \leq 5 \\ a + c + e + f \leq 15 \\ g \leq 0 \end{cases} \quad (2)$$

パラメータ同士の従属関係より、(a,f)、(b,d)、(c,e) の各々を順に、 D_a 、 D_b 、 D_c にパラメータの統合を行った結果、フロート解空間を表す連立不等式は以下ようになった。

$$\begin{cases} D_a + D_b \leq 5 \\ D_a + D_c \leq 15 \\ g \leq 0 \end{cases} \quad (3)$$

この連立不等式を用いて単体体積の妥当性評価実験を行った。まず、Activity の 1 日遅延あたりの解空間縮小率を単体体積指標で評価し結果を降順に並べた。次に、格子幅 1 日での格子点数で評価した結果も同様に並べた。図 12 がその結果である。

これらの結果を対比させたところ、単体体積による評価の順序関係と格子点数による評価の順序関係が完全に一致した。

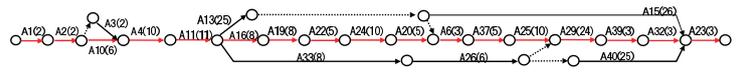


図 13: 建築プロジェクトの Activity Network

このことから、この例のように小規模のプロジェクトであれば、単体体積による遅延リスク耐性の評価は可能であるといえる。

5.4.2 建築プロジェクト

Activity 数 23、並行パス数 12 の建築プロジェクトの Activity Network を図 13 に示す。ここで、A1、A2、…、A23 は Activity 名を表し、() 内は各々の標準的所要時間を表す。また、図中に記していないが、各々のフロートパラメータは同名小文字の a_1 、 a_2 、…、 a_{23} とする。パラメータ同士の従属関係より、(a1,a2,a4,a11,a23)、(a3)、(a6,a37,a25)、(a10)、(a13)、(a15)、(a16,a19,a22,a24,a20)、(a29,a39,a32)、(a33,a26)、(a40) の各々を順に、 D_1 、 D_3 、 D_6 、 D_{10} 、 D_{13} 、 D_{15} 、 D_{16} 、 D_{29} 、 D_{33} 、 D_{40} にパラメータの統合を行い、納期は 120 日とした結果、フロート解空間を表す連立不等式は以下ようになった。

$$\begin{cases} D_1 + D_{10} + D_{16} + D_6 + D_{29} < 2 \\ D_1 + D_3 + D_{16} + D_6 + D_{29} < 6 \\ D_1 + D_{10} + D_{16} + D_6 + D_{40} < 7 \\ D_1 + D_3 + D_{16} + D_6 + D_{40} < 11 \\ D_1 + D_{10} + D_{13} + D_6 + D_{29} < 13 \\ D_1 + D_3 + D_{13} + D_6 + D_{29} < 17 \\ D_1 + D_{10} + D_{13} + D_6 + D_{40} < 18 \\ D_1 + D_3 + D_{13} + D_6 + D_{40} < 22 \\ D_1 + D_{10} + D_{13} + D_{15} < 35 \\ D_1 + D_3 + D_{13} + D_{15} < 39 \\ D_1 + D_{10} + D_{33} + D_{29} < 42 \\ D_1 + D_3 + D_{33} + D_{29} < 46 \end{cases}$$

5.4.1 節の例題と同様に、Activity の 1 日遅延あたりの解空間縮小率を単体体積指標で評価し結果を降順に並べ、次に、格子幅 1 日での格子点数で評価した結果も同様に並べた。図 14 がその結果である。なお、図中で赤く表示されている文字は、クリティカルパスの部分に該当する。

これらの評価結果を対比させたところ、まず、赤字の並びからクリティカルパス上の Activity の遅延が与えるインパクトが総じて大きいということが分る。次に並び順の対比に着目すると、一部一致しない箇所は見受けられたが、概ね一致するという結果になった。具体的には、10 のパラメータのうち 8 つは大小関係が一致し、残る 2 つ (d_{29} と d_{10}) の大小関係のみ入れ替わるという結果になった。

5.4.3 JAPEX モデル

JAPEX モデルとは、中東・北アフリカ地域における中規模ガス処理プラントを建設する EPC フルターンキー・プロジェ

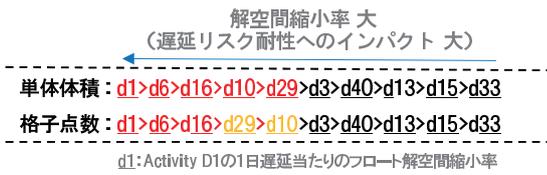


図 14: 各 Activity ごとの 1 日遅延当たりの解空間の縮小率の比較 (建築)

クト(※5-2)²⁰⁾を模したプロジェクトモデルである。この種のプロジェクトを熟知した現場経験者らにより作成された研究用モデルであり、この種のプロジェクトの扱いにくい性質(並行作業パスが多く、クリティカルパスが頻繁に入れ替わる等)をふんだんに取り入れる形で作成されたものである。ただし、本評価実験では、計算量の都合により、実際に提供されている JAPEX モデルをやや縮小したものを使用した。その縮小版の JAPEX モデルの Activity Network は図 15 である。

縮小版 JAPEX モデルプロジェクトの Activity 数は 15、並行パス数 10 である。ここで、D1、D2、・・・、D23 は Activity 名を表し、() 内は数字は各々の標準的所要時間を表す。各 Activity のフロートパラメータは同名小文字の d1、d2、・・・、d23 で表している。また、このモデルプロジェクトは、パラメータ同士の従属関係に基づいて、すでにパラメータ統合の処理がされている。納期を 44 日とした結果、フロート解空間を表す連立不等式は以下ようになった。

※5-2 建物や工場、発電所などの建設に関する契約の一つが、「EPC 契約」である。設計(engineering)、調達(procurement)、建設(construction)を含む、建設プロジェクトの建設工事請負契約を指す。EPC を提供することを「EPC サービス」と呼ぶこともある。また、「ターンキー契約」もしくは「フルターンキー契約」という表現もほぼ同じ定義で使われる。工事を発注した人が、カギを回せばすぐにも稼働する状態まで請け負うので、こう呼ばれる。EPC 契約を採用すれば、発注者にとっては、施工ミスや資材の高騰など設計・調達・建設にかかわるリスクを低減できる。

$$\left\{ \begin{array}{l} d1 + d3 + d7 + d13 + d20 + d22 < 2 \\ d1 + d4 + d11 + d16 + d22 < 2 \\ d1 + d3 + d7 + d15 + d20 + d22 < 3 \\ d1 + d3 + d7 + d15 + d19 + d23 < 3 \\ d1 + d5 + d11 + d16 + d22 < 3 \\ d1 + d3 + d8 + d11 + d16 + d22 < 4 \\ d1 + d3 + d8 + d12 + d23 < 5 \\ d1 + d4 + d11 + d23 < 14 \\ d1 + d5 + d11 + d23 < 15 \\ d1 + d3 + d8 + d11 + d23 < 16 \end{array} \right.$$

建築プロジェクト同様に、Activity の 1 日遅延当たりの解空間縮小率を単体体積指標で評価し結果を降順に並べ、次に、格子幅 1 日での格子点数で評価した結果 (L_1) と格子幅 0.5 日

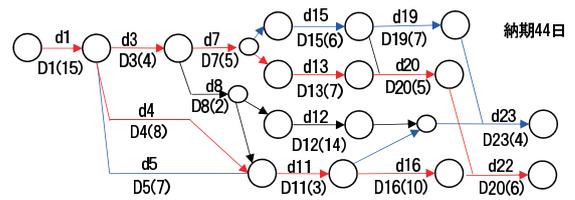


図 15: 縮小版 JAPEX モデルの Activity Network

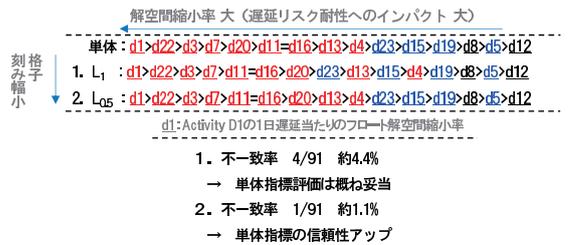


図 16: 各 Activity ごとの 1 日遅延当たりの解空間の縮小率の比較 (JAPEX)

での格子点数で評価した結果 ($L_{0.5}$) も同様に並べた (図 16)。表記方法は基本的に建築プロジェクトの例と同じであるが、図中赤く記したパスや文字は初期スケジュールでの CrP、青は準 CrP のうち CrP とのオーバーラップのない部分に該当する。

これらの結果を対比させたとこ、まず、赤・青の色の分布より、クリティカルパス上の Activity の遅延が与えるインパクトが総じて大きいということが分る。

また、Nd-SV の結果は格子幅の細かい $L_{0.5}$ の方の結果により近いということが分る。具体的には、 L_1 との比較で大小関係が一致しなかった組は (d11=d16,d20),(d23,d13),(d23,d4),(d15,d4) の 4 組だが、 $L_{0.5}$ との比較では (d11=d16,d20) の一組のみであった。

5.4.4 発電設備のオーバーホール

精度評価実験の最後に、Activity 数 130 程度、並行パス数 30 の発電設備のオーバーホールプロジェクト (図 17) を対象とし、単体体積 (NdSV) が遅延リスク耐性評価指標として利用可能かどうか調べるための評価実験を行った。パラメータ同士の従属関係より、都合 70 程度の Activity を 14(D1~D14) のパラメータに統合した (図 18)。

これまでの精度評価実験同様、格子点数と単体体積の減少率の順序関係を比較したいところではあるが、5.2.5 節で述べたように、格子点数の数え上げには計算量の問題があるため、今回は単体の体積の減少率の順序関係のみで妥当性を評価した。具体的には、減少率を降順に並べたときに、その順序関係が、よりプロジェクト全体工期に影響を及ぼしやすい作業順に並んでいるかどうか、つまり、クリティカルパスや準クリティカルパス上の作業の遅延による単体体積の減少率が高いかどうかを確かめた。

減少率を降順に並べた結果は以下の通りである。

$$D1 > D6 > D4 > D13 > D9 > D3 > D14 > D12 > D10 = D2 > D7 > D5 > D11 > D8$$

まず、クリティカルパス上の Activity(D1, D3, D4, D6, D9,

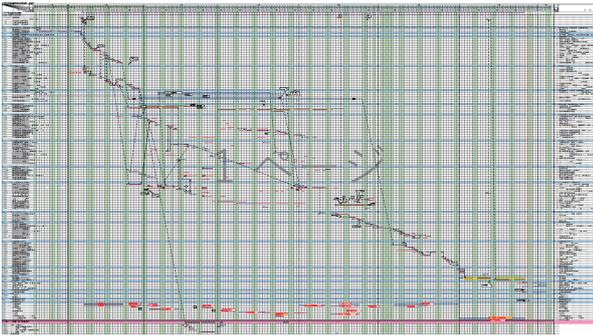


図 17: 発電設備のオーバーホールプロジェクトの Activity Network

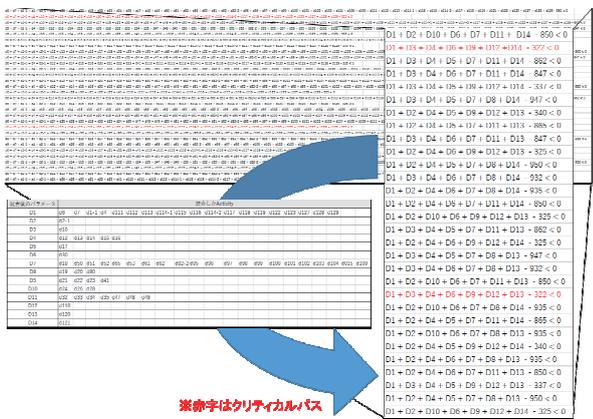


図 18: 発電設備 OH プロジェクトの解空間を表す連立不等式

D12, D13, D14)に着目すると、非クリティカルパス上のアクティビティに比べ、減少率が高いことが見て取れる。また、クリティカルパス上の Activity の中でも D1 や D6 のように、クリティカル性の高いパスにより多くの含まれる Activity がより減少率が高いことも確認できた。

5.5 遅延リスク耐性の精度評価実験の考察

以上、Activity の 1 日遅延がフロート解空間に与えるインパクトの大きさを、格子点数と Nd-SV の 2 通りで評価し、双方の結果を、インパクトの大きい方から降順で Activity 列を作成し比較した。

5.5.1 クリティカルパスの入れ替りが頻繁でないケース

今回試したケース (3~15 次元) では、いずれにおいても、格子点と Nd-SV による評価結果はほぼ一致した。

5.5.2 クリティカルパスの入れ替りが頻繁なケース

格子点と Nd-SV による評価結果は必ずしも一致しなかった。しかし、格子幅を細かくすると一致率が上がった。このことは、格子点評価の精度を上げるほどに、格子点と Nd-SV による評価結果は一致するのではないかと示唆する。なお、格子点評価の場合、刻み幅が大きいほど、尖った領域の評価が荒くなる傾向がある²¹⁾。ちなみに、格子点数に基づく体積評価がある程度厳密に行われるためには格子幅は次元 N に対し少なくとも $1 / (N-1)$ の細かさが必要と予想され

ている (Reeve, 1957)。しかし、格子幅を細かくする程計算時間の問題が大きくなることは先に述べたとおりである。

6. リスクマネジメントへの実践的活用

リスクマネジメントとは、事前に起こり得るリスク (予算、スケジュール、メンバー等に関するリスク) を考え、そのリスクを避けていくことによりプロジェクトを成功に導くことである。冒頭で述べた PMBOK (Project Management Body of Knowledge)⁵⁾でも、リスクマネジメントはナレッジ・エリアの 1 つとして定義されていることから、プロジェクトマネジメントの中でも重要視されるべき項目であるといえる。しかし、リスクマネジメントとは、まだ起きていないことを予測し、それに対する対策を講じようとする活動であるため、まだ起きていないどの事案にどれだけ投資するかの意思決定をし、その投資の妥当性・必要性を客観的に示さなくてはならない。ここにリスクマネジメントの難しさがある。

ここでは、プロジェクト進捗段階で起こり得るリスクのうちの一つである「遅延リスク」、すなわち、スケジュール遅延が起こり、契約納期違約を引き起こす可能性のあるリスクに着目し、この遅延リスクに関する一管理法を提案する。具体的には、まず、プロジェクトの成功を「契約納期までに完遂すること」と限定的に定義する。そして、プロジェクトの成功を妨げる可能性のある個々のスケジュール遅延がプロジェクトの全体工期に及ぼす影響の程度を、数値化、視覚化し、それをリスクマネジメントに活用する方法を提案する。ここで、「プロジェクトの全体工期に及ぼす影響程度」は、本稿で提案する遅延リスク耐性の大きさを求める方法で求める。こうすることにより、クリティカルパスの長さのみで影響評価していた従来法では出来なかったクリティカルパスの入れ替わりの影響も含めた総合的な評価が可能ということは先に述べてきたとおりである。

以降、ここでの提案手法、即ち、「遅延リスク耐性の大きさで測るプロジェクト全体工期に及ぼす影響程度」を RD (Resistance to schedule-delay) と呼び、RD に基づく分析方法全般を、RDM (Resistance to schedule-delay Management) と呼ぶこととする。

以下では、まず、既存の“アードバリュー法 (EVM)”について示す (6.1 節)。EVM は、プロジェクトの進捗段階での「スケジュールとコスト」を一元管理できる最近の手法である。EVM が開発される以前は、スケジュールとコストは独立に管理されてきたが、EVM では、これらをコストベースで一元化し、数値化、視覚化できる。次に、今回の提案である RDM について示すとともに、これを EVM と併用する効果について考察する (6.3 節)。

6.1 アードバリュー法 (EVM 法)

進捗状況を客観的に示せないプロジェクトの現場でありがちなのは、プロジェクトメンバーからリーダーへの客観的でない報告である。即ち、「来週までに終わるつもりです」、「少し遅れ気味になってきました」、「予定していた作業量が想定より多そうです」などといった主観的な報告である。このような主観的な報告だけでは合理的な進捗管理はできない。合理的な進捗管理のためには、進捗を測り、数値化、視覚化する

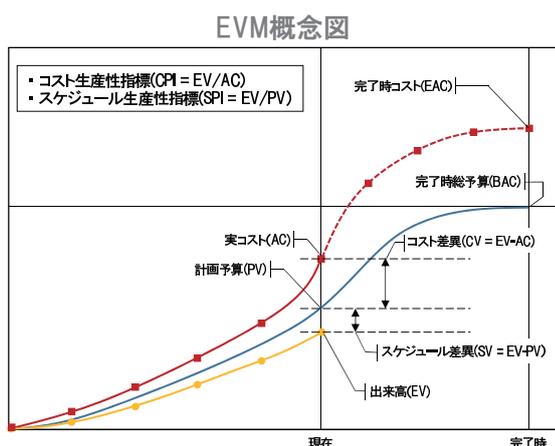


図 19: EVM の概念図

手立てが必要なのであるが、これを可能にするのが EVM である。

6.1.1 概説

コスト単位で進捗を測る EVM という手法がある。EVM は Earned Value Management の略であり、敢えて日本語に直訳すると「出来高管理」となる。これで何ができるのかというと、「スケジュールとコスト」を同時に管理でき、しかも視覚的に管理できる利点がある。スケジュール管理のみの技術 (PERT/CPM 等) は 1960 年代より前から発達していたが、「コスト」に関する手法の開発は後から (1960 年代～) であり、まず米国国防総省や NASA で使われはじめ、プロジェクトの現場で一般的になり始めたのは 1990 年代と言われており、歴史は浅い。なお、日本では更に遅く、その 10 年後の 2000 年代から利用され始めたが、現場でこれを使いこなせる人材は未だ少ない。日本のプロジェクトマネジメント技術力は世界水準でみるとかなり遅れているというのが現状である。

6.2 EVM 手法の意味

図 19 に EVM の概念図を示す²²⁾。EVM で重要なことは、計画予算 (PV: Planned Value)、実績コスト (AC: Actual Cost)、出来高 (EV: Earned Value) の 3 つの情報によって、プロジェクトの状況がどのようにになっているのかを見定め、今後の予測を行うことである。そのためには、プロジェクトにおけるパフォーマンスを確認することになるが、次の 4 つの指標が確認すべきパフォーマンス指標となる。

- コスト差異 (CV: Cost Variance) = $EV - AC$
- スケジュール差異 (SV: Schedule Variance) = $EV - PV$
- コスト生産性指標 (CPI: Cost Performance Index) = EV/AC
- スケジュール生産性指標 (SPI: Schedule Performance Index) = EV/PV

以下例を使って示す。

図 20 の左側は古典的コストマネジメントの考え方を示している。5 ヶ月で 1,200 万円の予算を持ったプロジェクトのコストを古典的な手法でマネジメントするとどうなるかとい

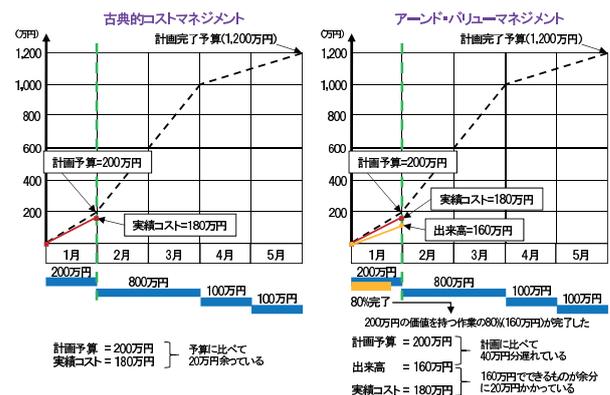


図 20: 古典的コストマネジメントと EVM の比較

う例である。まず 1 月が終わった段階での状況を見てみると、この時点では、計画 (PV) の消費予算 200 万円に対して実績 (AC) をみると 180 万円となっており、計画予算に対して実績は - 20 万だったことがわかる。つまり計画に比べ予算に 20 万円ほど余裕がありそうだという状況が、この 2 つの情報 (PV と AC) によって得られることになる。

一方、図 20 の右側を見てみると、古典的コストマネジメントに対して新たに出来高 (EV) という情報が加わっている。出来高とは、実際に出来たものの価値を示すもので、計画予算に対して獲得できた価値がどれだけあるのかを示すものである。出来高の算出方法は一通りではないが、この例の場合、200 万円の作業が 80% 完了したとして、200 万円の 80% に相当する 160 万円分の作業が完了したということで、出来高は 160 万円と算出されている。この出来高の情報が手に入ることによって、次のようなことが分かるようになる。まず計画予算と出来高を比べてみる。計画予算が 200 万円ということは、計画では 1 月末時点では 200 万円分の作業が完了する予定であったことを意味し、それに対して出来高が 160 万円であるということは、40 万円分の作業が未完了でプロジェクトが 40 万円分遅れているということになる。つまり、プロジェクトは予定より遅れており、このまま行けば納期が遅れる可能性がありそうだという状況が、この 2 つの情報 (PV と EV) によって得られることになる。次に、実績コストと出来高を比べてみると、出来高は 160 万円であったが、160 万円の出来高を獲得するために費やしたお金は 180 万円であったという結果になっている。つまり、160 万円では出来ずに、180 万円必要だったことが示されており、この調子でいくとプロジェクトの予算は不足、コスト超過の可能性があるので、この 2 つの情報 (AC と EV) によって得られることになる。

以上のように、当該事例に対し、古典的なコストマネジメントによる分析からは「予算が余りそう」という楽観的な分析結果が得られたのに対し、同じ事例にもかかわらず、EVM による分析からは「納期は遅れそうで、予算も超過しそうだ」という分析結果が得られている。つまり、計画予算 (PV) と実績コスト (AC) の 2 つの情報によってプロジェクトの状況がどのようにになっているのかを見定め今後の予測を行う古典的なコストマネジメントからは得られなかった正しい情報が、PV、AC に加え出来高 (EV) の情報を扱う EVM 手法からは得られたということである。マネジメントで重要なことは、いかに早い段階で正しい判断と対応が取れるかということであ

る。正しい情報が得られなければ正しい判断・対応は取れないためである。手遅れになる前に早く対応策を講じることでプロジェクト失敗の危険性を回避することができる。

6.3 RDM (Resistance to schedule-delay Management)

従来の PERT・CPM ではプロジェクト実施段階での所要時間の評価・管理が可能であり、上述の EVM では所要時間とコストの一元管理が可能である。本節で提案する RDM は、従来法でできなかった実施段階での遅延リスクを評価・管理が可能な手法である。以上より、本稿で提案する RDM を EVM と併用することにより、プロジェクトスケジュールの時間、コスト、遅延リスク管理し易さの 3 つの指標で評価できるようになる。

6.4 考え方

ネット上に路線情報検索のサイトがある²³⁾。出発地と到着地を示すと、行き方の候補を複数示すとともに、それぞれの方法を 3 種類の指標「所要時間」、「コスト」、「乗り換え回数」で評価するというものである。

例えば上記サイトで出発地を「宮崎空港」、到着地を「宮崎大学 (木花キャンパス)」とすして路線検索すると、6 件の行き方を評価と共に示す。図 21 は、6 件中 3 件の結果である。行き方や価格を具体的に示すとともに、「早い」、「安い」、「楽」という評価が与えられている。ここで、「早い」は所要時間に関する評価、「安い」はコストに関する評価である。なお、早いか安いかの判断基準はシステム内部で決まっており、その基準に従った結果「早い」(「安い」)となれば、「早」(「安」と表示される。また、「楽」に関しては、本システムの場合は乗り換え回数が少ないものを「楽」とみなし、乗り換え回数 1 回までなら「楽」と表示するようになっている。したがって、たとえ最寄り駅から到着地まで 30 分以上歩かなくてはならない場合でも「楽」と表示されてしまう(ケース 3 参照)。しかし、これは定義の仕方次第であるため、適宜「楽」の定義を設定しなおせばよいことであり、本質的な問題ではない。基本的に非常に便利な評価を示すツールである。

提案する RDM は、上記路線検索システムの評価でいうところの「早」と「楽」を、プロジェクトスケジュールに対し評価するための手法である。特に「楽」は従来手法 (PERT、EVM) では評価できない指標である。本手法を従来法と併用することで、「早」、「安」、「楽」の全てを評価できることとなる。

プロジェクトスケジュールに対する各評価について、「早」は当該スケジュールが遅延しなかった場合の全体工期の長さに対する評価となり、「安」は同スケジュール通りに実施された場合にかかるコストに対する評価となる。では、「楽」とは何に対する評価か。本稿では、これを、スケジュール管理上の「楽」とし、遅延リスク管理のし易さを評価する指標とみなす。具体的には、「楽」の評価が高いスケジュールとは、スケジュールが定める全体工期の長さが個々 Activity 遅延に影響されにくいスケジュールとする。つまり Activity 遅延がダイレクトに全体工期を引き延ばすことが少ないスケジュールほど「楽」の評価は高いということである。先に定義した「遅延リスク耐性」とは、まさにこの「楽」の程度を定量化したもの



図 21: 路線情報検索のサイト実行例

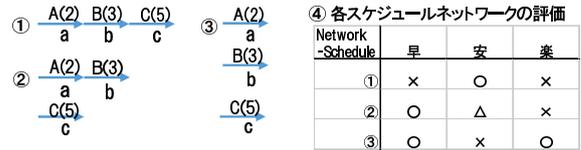


図 22: スケジュールの「早」、「安」、「楽」の評価を説明する簡単な例

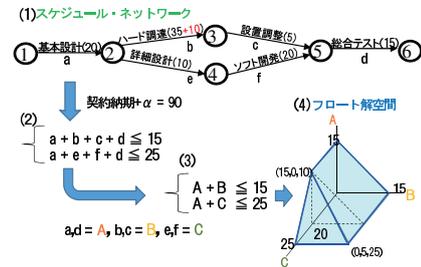


図 23: 簡単なプロジェクト例の Activity Network とフロート解空間

に相当する。

スケジュールの「早」、「安」、「楽」の評価について、簡単な例 (図 22) を用いて説明する。①、②、③は各々スケジュール・ネットワークを略記したものである。いずれも Activity A, B, C から構成されているが、ネットワーク構造が各々異なる。具体的に、①は A, B, C の順に直列であり、②は直列な A, B と C が並行しており、③は A, B, C の全てが並列である。これら 3 種類のスケジュール・ネットワークの評価が④である。まず「早」の評価であるが、①～③の所要時間は順に 10 日、5 日、5 日であることから、「早」の評価は表のとおりである。次に「安」であるが、バスごとに 1 名分の人件費がかかると仮定すると表の評価結果は明らかである。最後の「楽」であるが、厳密には遅延リスク耐性に基づき評価値を計算するようにするのだが、ここでは表のとおりの結果であることは直感的に理解できる。つまり、①、②が×というのは、①、②は構造上どのような Activity 遅延であっても (スケジュールが定める) 全体工期に直ちに影響を及ぼす (全体工期を引き延ばす) ためであり、それに対し③は Activity C 以外は少し遅れても全体工期に影響を及ぼさない (全体工期を引き延ばさない) 余裕を持つため、①、②に対し相対的に○ということである。

6.4.1 分析方法

プロジェクトスケジュールの遅延リスク耐性の静的な評価方法を 5. 節で示した。ここでは、プロジェクト実施期間通しての遅延リスク耐性の動的な変化を分析・評価する方法を示す。具体例で説明する。図 23-(1) は 4. 節で示した簡単なプロ

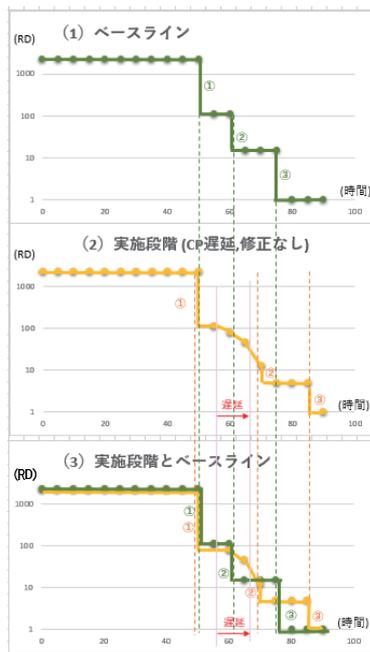


図 24: 「プロジェクト実施期間通しての遅延リスク耐性の時間変化

プロジェクト例のスケジュール・ネットワークであり、このフロート解空間の式表現と図的表現が(2)~(4)である(注 6.1)。そして、この図 23 の RD (Resistance to schedule-delay) の時間変化を表したグラフが図 24 である。図 24-(1) はスケジュール通りにプロジェクトが実施された場合の RD の時間変化を表す。これを RD のベースラインと呼ぶこととする。これに対し、図 24-(2) は実際の実施結果での RD の時間変化を表す。ここでは「ハード調達」が 10 遅延したとしている。なお、図 24-(3) は、実施結果 RD(図 24-(2)) とベースライン RD(図 24-(1)) との差異がわかるように(1)、(2)を重ねて描いたものである。

分析方法の前に、図 24 のグラフ形状の見方について説明する。まず、RD のベースラインのグラフ(図 24-(1))を見ると、RD が大きく減少するタイミングが 55 日目、60 日目、75 日目(プロジェクト終了時点)の 3 回あるが、これらはスケジュール遅延による減少ではない。RD の計測対象であるフロート解空間の次元の降下を表している。次元の降下は Activity 終了のタイミングで起きる。ただし Activity 終了タイミングで毎回次元が落ちるわけではない。例えば、基本設計が終了した段階では次元降下はない。基本設計の余裕パラメータは総合テストのそれと一次従属の関係にあるため、基本設計と総合テストの双方が終了しないと次元は落ちないということである。なお、基本設計と総合テストの終了による次元降下は、図 23 でいえば、座標軸 A が消失することに相当する。仮にスケジュール遅延がないならば、本例で最初に次元が落ちるタイミングは、ソフト開発終了のタイミングである。すなわち、詳細設計とソフト開発の終了時点ということであるが、このとき消失するのは座標軸 C である。

また、実施段階での RD のグラフ(図 24-(2))を見ると、ベースラインでの次元低下の 2 点(50 日目と 60 日目)に加え、もう 1 点、RD 減少のポイントがある。2 度目の次元低下ポイン

ト(60 日目)直後の 60~65 日目である。これはハード調達の Activity 遅延による RD 減少であり、次元低下のような急激な減少ではなくならかな減少である。

以上のグラフ形状の見方を踏まえ、次に分析上の着眼点について説明する。実施段階での RD をベースラインと比較するうえでの着眼点は次の 2 点である。一つは当該 Activity 遅延発生直後の差異・(1)であり、もう一つは契約納期上のプロジェクト終了時点付近の差異・(2)である。

上記(1)の大きさは、当該 Activity 遅延直後の全体工期に対する影響の大きさを表す。この大きさはクリティカルパス長へ及ぼされる顕在的な影響の大きさのみならず、その他のパス長へ及ぼされる潜在的なものも加味した影響大きさの総和となる。前者の顕在的な影響大きさは従来法でも遅延リスク管理対象であった。

一方、後者の潜在的な影響大きさはこれまで遅延リスク管理の対象外であり管理手法の検討すらされてこなかった。しかしこの潜在的な影響大きさも、顕在的影響大きさ(着目時点でのクリティカルパス長)と同様、今後はできるだけ減らさない方向で管理することが重要である。近年のプロジェクトのクリティカルパスの入れ替わりが頻繁かつ工期の長いという特徴ゆえ重要ということである(1. 節、5. 節)。

ただし、当然ながら、潜在的な遅延リスクの影響の大きさはクリティカルパスが入れ替わらない限り、プロジェクト終了時点が近づくにつれ減少していき、最終的には 0 になる。したがって、上記(2)の差異の大きさは、当該 Activity 遅延が(当該時点での)クリティカルパス上で発生したものならプラスの値を取り、非クリティカルパス上で発生したものなら 0 となる。このことは、クリティカルパスの入れ替わりがない限りは非クリティカルパス上の遅延の影響の大きさは顕在化しないという従来知見と同じことを意味する。

(注 6.1) 制約条件式の上限は、5. 節のような契約納期そのものではなく「契約納期+ α 」としてある。 $+\alpha$ の措置は、本方法で実施段階 RD の分析をする上での便宜上のものである。具体的には、本分析はフロート解空間が正の領域(第 1 象限)にないと分析できないため、スケジュール遅延により第 1 象限から逸脱することを防ぐため、今回はフロート解空間の分布領域を $+\alpha$ 分平行移動させることで対処した。

6.4.2 分析例

図 25、26 に分析結果(ケース 1~6)のグラフを図示する。緑がベースライン RD、黄色が実施段階 RD である。各ケースは各々以下のような特徴を持つ。

- ケース 1: クリティカルパス (CP) と準 CP に同時に乗る Activity の遅延のケース (スケジュール修正なし)
- ケース 2: CP のみに乗る Activity の遅延、スケジュール修正なしのケース
- ケース 3: ケース 1 でスケジュール修正ありのケース
- ケース 4: 非 CP(準 CP のみ)に乗る Activity の遅延、スケジュール修正なしのケース
- ケース 5: ケース 4 でスケジュール修正ありのケース

ケース6：非CP(準CPのみ)に乗るActivityの遅延が2回起きた結果、非CPがCPに変化したケース(スケジュール修正なし)

ケース1では、クリティカルパスに乗るActivity「基本設計」の遅延に伴い実施段階RDは減少しベースラインを大きく下回る。その後の3段階の次元降下とプロジェクト終了は、いずれも遅延した分だけベースラインのそれに遅れて起きる。

ケース2では、同じくクリティカルパスに乗る「ハード調達」の遅延に伴いRDは減少しベースラインを立たまわる。ここで、ケース1との違いとして着目すべき点は、遅延幅はケース1のときと同じであるにもかかわらず、ケース2でのRD減少幅の方がケース1のそれよりもずっと小さいということである。つまり、共にクリティカルパス上の遅延であるにも関わらず、全体工期へ及ぼす影響は、ケース1の遅延の方がケース2の遅延より大きいということを示している。この違いはどこからくるのか。それは、遅延Activityがクリティカルパスだけに乗るのかそれ以外のパスにも乗るのかの違いからきている。つまり、ケース2での遅延Activity「ハード調達」は、クリティカルパスのみに乗るのに対し、ケース1での遅延の「基本設計」は、準クリティカルパスにも乗っているということである。このため、厳密に言えば、基本設計の遅延の方が遅延リスクをより大きくするということである。このように、従来法の遅延リスク分析は、現況のクリティカルパスへ及ぼす影響しかみなかったのに対し、提案法ではクリティカルパスの入れ替わりの可能性まで含めて評価値に反映させる分析ということである。従って、遅延リスク分析に関しては、従来法より提案法の方がより精度が高いということである。なお、ケース2では、1回目の次元降下は遅延前であるため、1回目の次元降下はベースラインが示すタイミングで起き、それ以降の2回の次元降下、及びプロジェクト終了はケース1同様に遅延分遅れて起きている。

次に、ケース3はケース1と同様の遅延が起きるケースだが、プロジェクト最後のActivity「総合テスト」の所要時間を遅延時間10の半分の5だけ短縮するスケジュール修正を行っている。その結果、遅延後の3段階の次元降下はいずれも遅延した分だけベースラインのそれに遅れて起きるが、プロジェクト終了は修正幅分ベースラインの終了タイミングまで回復している。

以上、クリティカルパスに乗るActivityの遅延分析の結果である。次にクリティカルパスに乗らないActivityの遅延分析を行ったケース4～6について示す。

ケース4では、非クリティカルパスに乗る「詳細設計」遅延に伴いRDは減少し、その直後の1回目の次元降下はベースラインより遅れて起きる。しかし、クリティカルパス遅延と異なり、その次の2回目の次元降下のタイミングでの実施段階RDはベースラインRDのレベルまで復活し、その後の3回目の次元降下及びプロジェクト終了までベースラインと同じレベルで推移する。このケースで着目すべき点は、従来の遅延分析では及ぼす影響は0と見なされてきた非クリティカルパス上のActivity遅延の影響が数値化できている点である。これについても、従来法の遅延リスク分析が現況のクリティカルパスへ及ぼす影響しかみなかったのに対し、提案法

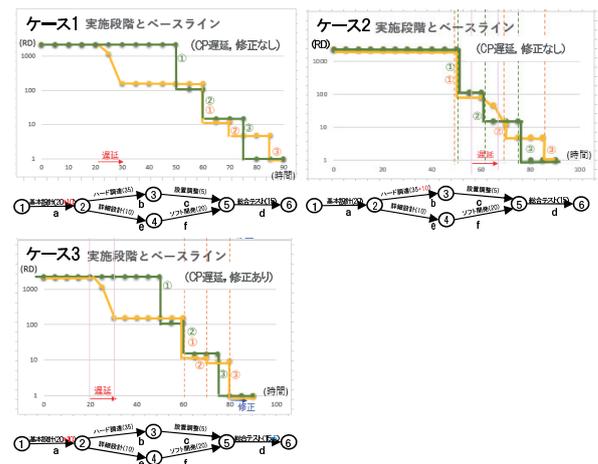


図 25: 「実施段階のRD分析結果(ケース1～3)

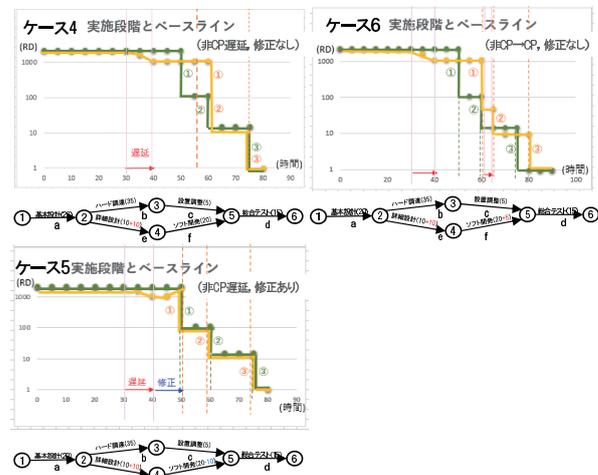


図 26: 「実施段階のRD分析結果(ケース4～6)

ではクリティカルパスの入れ替わりの可能性まで含めて評価値に反映させる分析になっているが故の精度の高い分析結果である。

ケース5は、ケース4と同じく「詳細設計」が10遅延するケースだが、そのすぐ後に同非クリティカルパス上の「ソフト開発」の短縮でスケジュール修正を行っている例である。

遅延も修正も非クリティカルパス上で起きているため、従来法の遅延分析では数値化されない状態変化であるが、本方法ではRDの(遅延による)減少と(修正による)増加が遅延、修正に伴って起きていることがグラフ上明らかにされている。

以上、ケース4、5では、非クリティカルパス上での遅延や修正すら評価値に表れる手法であることを示した。次に、このような非クリティカルパス上の遅延や修正をわざわざ数値化することの意義をケース6で説明する。

ケース6は非クリティカルパス上での遅延が2度発生し、2度目の遅延で非クリティカルパスからクリティカルパスへ変化するケースである。実施段階RDを見ると、1回目、2回目の遅延に伴いRDが順次減少している様子が確認できる。具体的には、まず、遅延の30～40でベースラインを下回り、その後ベースラインより遅延分遅れて1回目の次元降下が起こり、60～65で2回目の遅延と直後に2度目の次元降下が起き、そのタイミングでクリティカルパスの入れ替わりが起きている。

る。そのため、最後までクリティカルパスの入れ替わりの起きないケース 4、5 と異なり、最後までベースラインレベルまで回復することなく、3 度目の次元降下とプロジェクト終了はベースラインのそれより遅れて生じる。以上のケース 6 で着目すべき点は、まだ非クリティカルパス上の遅延に過ぎない 1 回目の遅延の (全体工期に対する) 影響がグラフ上確認できているということである。従来法では非クリティカルパス上の遅延の影響は、のちにそれがクリティカルパスにとって代わるとしても、そうなるまでは確認する術が全くない (数値化されない) ということがあった。したがって、クリティカルパスになったのちに過去の遅延理由を慌てて分析し、「もっと早く (クリティカルパスに変化する前に) 気づいていれば手を打てたのに」といったことが実際現場で起きてきた。それゆえ、本方法を用いることで、どのパスに乗っているとしても、乗っているパスの数や各パスのクリティカル性の高さの程度を加味した (全体工期への) 影響程度を数値化できるといことは、近年の大規模複雑かつ予期せぬ遅延の頻発するプロジェクト管理上は非常に有用と言えるのである。

7. ケーススタディ

建築プロジェクトの例 (5.4.2 節参照) を用い、リスクマネジメントのために提案手法を活用するケーススタディを実施した。建築プロジェクトの Activity Network は図 13 の通りであるが、契約納期は 122 日とした。図 27、図 28 は、プロジェクトのシミュレーション結果の一つを EVM で分析したグラフと RDM で分析したグラフである。グラフからもわかる通り、このシミュレーションの実施段階では、計画段階に比べ、トータルで 8 日の遅延が発生しており、1000 万円程のコスト超過が起こっている。なお、このシミュレーションの実施段階では、再スケジュールリングによる Activity の所要時間の短縮は行っていない。

以上のことを踏まえ、まず、EVM での分析結果のグラフ (図 27) を見ると、青い線で示された「計画予算コスト」と黄色い線で示された「出来高」を比較することで、全体工期に少しずつ遅れが生じていることがわかる。また、「計画予算コスト」と赤い線で示された「実績コスト」を比較することで、早い段階からコストの超過が起きていることが見て取れる。このように、EVM では「時間とコスト」についてのプロジェクトの現状を知ることができる。

一方で、RDM での分析結果からは、ベースラインの RD の変化と実施段階の DR の変化を比較することで、「時間と余裕」についてのプロジェクトの現状を知ることができる (6.4.1 節)。以下、具体的に述べると、まず、時間については、ベースラインの RD の次元降下による減少のタイミングと実施段階の RD の次元降下による減少のタイミングのずれをみると、全体工期の遅れを知ることができる。また、余裕については、図 27 のグラフ上の遅延が発生した箇所に着目すると、実施段階の RD が緩やかに減少していることから、各 Activity の遅延が全体工期に及ぼす影響の大きさを評価することができる。更に詳しく見ると、図 27 の緑色で囲まれた箇所

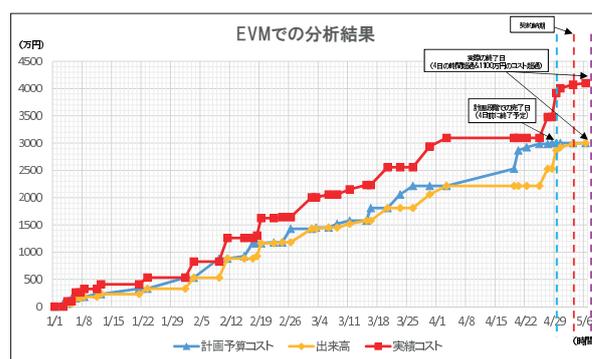


図 27: 建築プロジェクトの EVM での分析結果

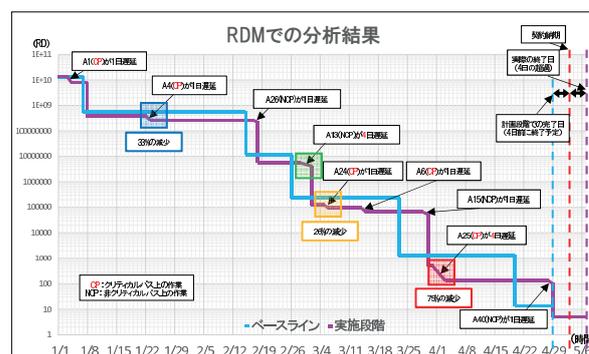


図 28: 建築プロジェクトの RDM での分析結果

所 (A13 の遅延) では、非クリティカルパス上の作業の遅延により RD が緩やかに減少している。つまり、従来手法では表現できなかった「非クリティカルパス上の作業の遅延が全体工期に及ぼす影響」が数値として表現できるということである。また、青色で囲まれた箇所 (A4 の遅延) と黄色で囲まれた箇所 (A24 の遅延) に着目すると、同じクリティカルパス上の作業の 1 日遅延でも、減少率が異なるということがわかる。これは、同じクリティカルパス上の作業でも、「どのパスに含まれているのか」、「何本のパスに含まれているのか」などの条件で、全体工期に及ぼす影響が異なるからである。

以上のように、RDM での分析を EVM での分析と併用することにより、プロジェクト実施段階での時間とコストの進捗と併せて (現況スケジュールの) 遅延リスク耐性の評価が可能となる。

8. まとめ

本稿では、近年の大規模かつ複雑化、多様化したプロジェクトにおける既存のスケジュール管理手法の問題を取り上げ、既存手法では管理の対象とされなかったプロジェクト・スケジュール上の余裕日数「フロート」に着目し、プロジェクト全体工期が契約納期を超過せずに済む可能性の大きさ「遅延リスク耐性」の評価ができる手法の提案を行った。具体的には、任意の作業項目が予期せぬ遅延をきたした際に、プロジェクトの全体所要時間が契約納期を超過せずに済む可能性の大きさ「遅延リスク耐性」を定量的に評価できるようにした。また、「遅延リスク耐性」算出の計算量問題に対処するための指標化について検討し、更に、精度評価実験により指標化された「遅延リ

スク耐性」の妥当性の有無を検討した。最後に、進捗管理手法であるアーンドバリュー法 (EVM) ではできない遅延リスク管理手法のための手法 RDM (Resistance to schedule-delay Management) を提案をした。RDM は遅延リスク耐性を活用する遅延リスク管理手法であるが、本稿ではこれを EVM と併用できる形で提案した。こうすることにより、プロジェクト実施段階での時間とコストの進捗と併せて (現況スケジュールの) 遅延リスク耐性の評価が可能になった。以下に本稿で行ったことを列挙する。

- 「遅延リスク耐性」の概念的定義と定量化、算出方法の明確化
- 「遅延リスク耐性」算出の計算量問題に対処するための指標化について検討
- 指標化された「遅延リスク耐性」の妥当性の有無を検討するための精度評価実験の実施
- 時間・コストの進捗評価手法 EVM と併用できる形での時間・余裕の進捗評価手法 RDM の提案

以上が本稿のまとめである。今後の課題としては、更に大規模なプロジェクトに、提案手法を適応して、指標化された「遅延リスク耐性」の妥当性の有無を検討する必要があると考える。

参考文献

- 1) Program Evaluation and Review Technique (PERT) - Wikipedia
- 2) クリティカルパス法 - Wikipedia
- 3) 佐藤知一：プロジェクトにおけるスケジュールと費用のトレードオフを考える, スケジューリング学会シンポジウム講演会論文集, pp.135-137(2015).
- 4) 藤原他：資源制約付きプロジェクト・スケジューリング問題に関する基礎的研究, 数理解析研究所講究録, pp.125-130 (2009).
- 5) 韓鳳天：水力発電設備の定期修理管理プロジェクトへの適用による工程管理システムの評価, 宮崎大学修士学位論文 (2016).
- 6) 富田旋：時間とコストの一元管理のためのプロジェクト管理システムの大規模プロジェクトへの適用に関する検討, 宮崎大学修士学位論文 (2017).
- 7) Project Management Institute：プロジェクトマネジメント知識体系ガイド (PMBOK ガイド) 第6版 (2017).
- 8) 門川真樹：プロジェクトのスケジュール・ネットワークの遅延リスク耐性の評価について, 宮崎大学修士学位論文 (2018).
- 9) アーンド・バリュー・マネジメント (EVM) - Wikipedia
- 10) 池田将明：建設エンジニアのための PMS によるプロジェクト計画入門 (2005).
- 11) 潮俊光：マックス代数によるシステム理論の基礎, 数理解析研究所講究録 1020 巻, pp.165-179 (1997).
- 12) 佐藤知一：スケジュールの DRAG とはどんな尺度か、タイム・コンサルタントの誌から available from(<http://brevis.exblog.jp/20000432/>), (2017/02 アクセス).
- 13) Stephen A.Devaux：The Drag Efficient, Defense AT&L, January-February (2012).
- 14) Drag Cost - 納期を加味した真のコストを考える available from(<http://brevis.exblog.jp/20044525/>), (2017/02 アクセス).
- 15) 新築注文住宅の建築費とコスト構成比の関係 available from(<http://www.house-support.net/tisiki/mitumori.htm>), (2017/02 アクセス).
- 16) ピックの定理 - Wikipedia
- 17) 佐藤郁郎：ピックの公式の拡張 (その 1、その 2) available from(http://www.geocities.jp/ikuro_kotaro/koramu/531_p1.htm), (http://www.geocities.jp/ikuro_kotaro/koramu/532_p2.htm), (2018/01 アクセス).
- 18) シンプレックス法 - Wikipedia
- 19) neetubot 「m 次元ユークリッド空間内での n 次元単体の五心などの導出」 [1.0] available from(<http://www7.atwiki.jp/neetubot/pub/neetbot-1.0.pdf>), (2018/01 アクセス).
- 20) EPC 契約とは available from(<https://tech.nikkeibp.co.jp/dm/article/WORD/20131015/309062/?ST=msb>), (2018/09 アクセス).
- 21) Ian Stewart (原著), 山崎秀記 (翻訳)：スチュアート教授のおもしろ数学入門 (1993/12).
- 22) Innovation Management 第 17 回 EVM (Earned Value Management) available from(<https://www.innovationmanagement.co.jp/column/no17/>), (2019/01 アクセス).
- 23) Yahoo! JAPAN 路線情報 available from(<https://transit.yahoo.co.jp/search/>), (2019/01 アクセス).