

Title	不確実的事象下における動的プロジェクトスケジューリングに関する研究
Author(s)	藤原, 稔久
Citation	
Issue Date	
Text Version	ETD
URL	https://doi.org/10.18910/34559
DOI	10.18910/34559
rights	
Note	

Osaka University Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/repo/ouka/all/>

不確実的事象下における動的プロジェクト
スケジュールリングに関する研究

大阪大学大学院情報科学研究科

2015年 1月 7日

藤原 稔久

研究業績

学術論文

Journal of papers

1. T. Fujiwara, H. Suwa and H. Morita, Dynamic project scheduling with reduction of activity durations, *International Journal Project Organization and Management*, (掲載決定) .
2. T. Fujiwara, H. Suwa and H. Morita, A dynamic scheduling with using propriety rules strategy and reduction processing time, *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, (投稿中) .

査読付き国際会議

1. H. SUWA and T. FUJIWARA, A new hybrid rescheduling policy based on cumulative delay *Proceedings of the 19th International Conference on Production Research (CD-ROM)*, 2007.
2. T. FUJIWARA, H. SUWA and H. MORITA, Capability of CPM based dynamic project scheduling, *2009 INFORMS International Conference on Service Science*, Hong Kong, China, 2009.
3. T. FUJIWARA, H. SUWA and H. MORITA A dynamic scheduling with using rescheduling strategy and reduction proceeding time, *International Symposium on Scheduling 2013*, Tokyo, Japan, 2013.

口頭発表

1. 藤原, 諏訪, 森田, 資源制約付きプロジェクト・スケジューリング問題に対するリアクティブ戦略に関する基礎的研究, 日本機械学会生産システム部門講演会 2008, 33-34, 2008.
2. 藤原, 諏訪, 森田, 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, RIMS 研究集会, 21 世紀の数理計画: 最適化モデルとアルゴリズム, 2008.
3. 藤原, 梅谷, 森田, パッキング問題の解法を利用した資源制約付きプロジェクトスケジューリング問題の初期スケジュール生成法, 日本 OR 学会研究部会『若手による OR 横断研究』「若手研究交流会」, ポスターセッション, 2008.
4. 藤原, 梅谷, 森田, Critical Path Method を用いたリアクティブ・プロジェクト・スケジューリングに関する基礎的研究, 大阪大学大学院情報科学研究科, 情報数理学専攻公開シンポジウム-IPS2009-ポスターセッション, 2009.
5. 藤原, 諏訪, 森田, Critical Path Method に基づくリアクティブ・プロジェクト・スケジューリングに関する基礎的研究, 第 53 回システム制御情報学会研究発表講演会, 2009.
6. 藤原, 諏訪, 森田, 作業分割および短縮費用に基づく動的プロジェクトスケジューリングに関する基礎的研究, 日本 OR 学会「計算と最適化の新展開」研究部会 SCOPE @ つくば- 未来を担う若手研究者の集い 2009 -, 2009.
7. 藤原, 諏訪, 森田, 作業短縮費用に基づく動的プロジェクトスケジューリングに関する基礎的検討, 日本 OR 学会研究部会『若手による OR 横断研究』「若手研究交流会」, 2009.
8. 藤原, 諏訪, 森田, 作業短縮費用に基づく動的プロジェクトスケジューリングに関する基礎的検討, 第 54 回システム制御情報学会研究発表講演会, 2010.
9. 藤原, 諏訪, 森田, プロジェクト管理における作業時間短縮を用いたリアクティブ戦略に関する基礎的研究, 第 55 回システム制御情報学会研究発表講演会, 2011.

10. 藤原, 作業時間の縮約を考慮した動的プロジェクトスケジューリングに関する研究, JFE スチール第 15 回 OR 技術交流会 研究発表会.
11. 藤原, 諏訪, 森田, 作業時間の縮約を考慮した動的プロジェクトスケジューリングに関する研究, スケジューリングシンポジウム 2011 講演論文集, 61-66, 2011.

目次

第 1 章 緒論	1
第 2 章 プロジェクト管理におけるスケジューリング問題	5
2.1 背景	5
2.2 プロジェクト計画におけるスケジューリング問題	7
2.3 RCPSP の表現とクラス分類	8
2.3.1 RCPSP の表現	8
2.3.2 RCPSP のクラス分類	9
2.4 プロジェクトのモデル	11
2.4.1 プロジェクトの記述	11
2.4.2 資源制約	12
2.4.3 基準スケジュール	12
2.5 動的スケジューリング	13
2.5.1 リアクティブ・スケジューリング	14
2.6 プロジェクト管理の技法	15
2.6.1 PERT/CPM	15
2.6.2 生産業界における CPM の役割	19
2.7 結言	19
第 3 章 Critical Path Method に基づく動的プロジェクト・スケジューリング	21
3.1 背景	21
3.2 CPM に基づく動的スケジューリング	22
3.2.1 提案手法	22
3.3 数値実験	27

3.3.1	実験条件	27
3.3.2	スケジュールの修正のタイミング	29
3.3.3	比較実験	31
3.3.4	考察	35
3.4	結言	35
第 4 章	優先ルールと処理時間の縮約に基づく動的スケジューリング	37
4.1	背景	37
4.2	優先ルールに基づくリスト生成	38
4.2.1	優先ルール	38
4.2.2	リスト生成の対象となる作業の抽出	40
4.3	スケジュール修正と処理時間の縮約の手順	41
4.3.1	スケジュール修正	41
4.3.2	コストを伴う処理時間の縮約	43
4.3.3	提案手法の手順	45
4.4	数値実験	46
4.4.1	問題設定	46
4.4.2	比較実験	48
4.5	結言	50
第 5 章	結論	53
	謝辞	55
	参考文献	57

目 次

2.1	スケジューリング問題における RCPSP の位置付け	8
2.2	ネットワークを用いたプロジェクトの表現	9
2.3	タイムチャートを用いたスケジュールの一例	9
2.4	スケジューリング問題における意思決定のタイミング	14
2.5	プロジェクトに存在するコスト	18
2.6	CPM の役割	20
3.1	作業 2 に発生した遅延後のスケジュール	24
3.2	部分ネットワークとカット集合	27
3.3	提案手法適用後のスケジュール	28
3.4	点検間隔の違いによるコスト増加 (j3016_1)	31
4.1	基準スケジュール	42
4.2	遅延発生後のスケジュールと抽出対象作業	43
4.3	優先ルールを用いたスケジュール修正	44
4.4	処理時間縮約後のスケジュール	46

表 目 次

3.1	各点検区間の結果	30
3.2	実験結果 ($N = 30, R = 4$)	32
3.3	実験結果 ($N = 60, R = 4$)	33
4.1	優先ルールの一例	40
4.2	実験結果	51

第1章 緒論

産業の発展および生産技術の発達に伴い、生産システムの規模が肥大化し生産の工程も多岐にわたるようになった。これに伴いスケジューリング理論に対する要請も複雑化し、限りある労働力、設備、時間のなかで効率的に生産活動を実施することが非常に重要な課題となっている。このようにスケジューリング問題は、産業への応用において重要であり、その理論も複雑であることから従来より非常に多くの研究がなされている。

スケジューリングは、多数の作業あるいは活動が有限な資源を競合的に使用しなければならないときに行われる計画行為の一つである。これは資源容量や先行関係などの制約を満たしながら、作業を時間軸上で資源に割り当てること指す。資源の例として機械を考えると、スケジュールにはそこで処理される多数の作業が存在し、それらを同時に処理できないとき作業の処理順序を決定する順序付けが必要となる。また、スケジューリングの目的は、与えられた制約条件を満たす実行可能なスケジュールを見つけ出すこと、あるいは、もし複数個の実行可能なスケジュールが存在するならば、それらの中から与えられた評価基準を最適にするスケジュールを見つけ出すことである。

先に述べたようにスケジューリング問題に関しては、これまでさまざまな研究がなされているが、一口にスケジューリング問題と言っても、生産スケジューリング問題、配送計画問題、時間割作成問題などさまざまな問題が考えられる。それに加え、研究（特に問題のモデル）が現実問題に適用されにくい形であり、実際の産業に応用するとなると個々の状況に応じた特殊な制約なども考慮しなければならない。そのため、スケジューリング問題に対する学術的研究の産業への応用が非常に困難であった。これにより近年では、様々な種類のスケジューリング問題を扱うことができる、資源制約付きプロジェクト・スケジューリング問題（Resource-Constrained Project Scheduling Problem; RCPSP）の研究がなされるようになってきている。

RCPSP では、従来のスケジューリング問題では考慮していなかった資源制約を追加する

ことによって、広範囲の問題を扱うことができる。RCPSP は、先行制約および資源制約下において作業の開始時刻を決定する問題として定義され、その目的関数や制約条件も多岐にわたる。しかしながら、事前に生成されたスケジュールのみに頼っていると、不確実性の影響で予定通りのスケジュールに従った遂行が困難となる。例えば、配送計画では訪問者の不在、構造物建築では資材の到着遅れなどの状況である。このような状況を回避するには、スケジュールを生成する段階で予めトラブルを予測し、頑健なスケジュールを生成し対処する方策。または、トラブルが発生してから適応的にスケジュールを立て直す方策などがある。前者を静的スケジューリング、後者を動的スケジューリングと呼ぶ。

静的スケジューリングは不確実的事象に関する事前情報をいかに効果的に利用するかという点を含め、理論面における膨大な関連研究が報告されている。動的スケジューリングに関しては、配送計画問題や生産スケジューリング問題など現実問題に適用されつつある。しかしながら、RCPSP においては静的スケジューリングの範疇に収まった理論面に関する論文が多く、動的スケジューリングを扱った報告は事例研究などが多くを占めている。適応的にスケジュールを立て直すという立場からは依然として報告が少ないのが現状である。このような背景を踏まえ、本論文では、効果的かつ一定の有用性を有する動的スケジューリングの手法を、

▷ スケジュールの生成方法：

- 1) Critical Path Method を用いた動的スケジューリング
- 2) 優先ルールを用いたスケジュール修正と処理時間の縮約に基づく動的スケジューリング

▷ タイミング方策：

- 1) 動的スケジューリングにおけるスケジュール修正実施の判断のための枠組みおよび方策の提案

の二つの視点から確立することを目指す。

まず、第2章にプロジェクト管理からみたスケジューリング問題を定義する。また、対象とする RCPSP を紹介し問題を定式化する。次に、動的スケジューリングが対象とする問題を明確にするとともに、従来研究の整理および紹介を行う。RCPSP の解の生成の側面から、スケジュール生成のタイミングを整理し、動的スケジューリングを定義するとともに従来研

究として整理する。また、プロジェクトにおける管理技法および生産業界での役割を紹介する。

第3章では、動的スケジューリングに Critical Path Method を取り入れたスケジュール修正方策を提案しスケジュール修正に有効なタイミング方策を明らかにする。不確実性に対する影響を受けた作業に対し Critical Path Method の考えを取り入れプロジェクト期日の遵守を目指す。その際に、発生するコストを低減させるために新たな評価指標を導入する。また、タイミング方策では、既存のタイミング方策に加え、各作業がもつ余裕時間に着目した方策の導入を行う。

次に、第4章では、Critical Path Method を取り入れたスケジュール修正方策の拡張として、優先ルールを導入する。スケジューリング問題の解法の一つとしてヒューリスティクス解法がある。この解法は、構成型解法を用い初期解を生成し、改善型解法を用い初期解を改善していく解法である。提案手法では、構成型解法に優先ルールを用い、改善型解法では Critical Path Method を取り入れ解を改善していく。数値実験により、プロジェクト管理に有効な優先ルールを明らかにする。第5章は、本論文のまとめであり、本研究で得られた知見を整理するとともに、今後の課題について述べる。

第2章 プロジェクト管理におけるスケジューリング問題

2.1 背景

プロジェクトを成功に導くには、プロジェクトを適切に管理することが重要となる。プロジェクト管理は、PMBOK と呼ばれる国際基準の知識体系によって標準化されている。PMBOK は、PMI (Project Management Institute) によって提供され、幅広いプロジェクトに対して適用できるプロジェクト管理の基盤を与える。PMBOK では、プロジェクト管理を以下の九つのグループに分類する。

(1) プロジェクト統合マネジメント

プロジェクト管理の特定、定義、結合、統一、調整などを行うためのプロセスからなる。すなわち、プロジェクト管理における各プロセスを相互に調整した上でプロジェクトの達成を図る役割を持つ。

(2) プロジェクト・スコープ・マネジメント

プロジェクトを成功のうちに完了するために必要なすべての作業を洗い出すために必要なプロセスからなる。具体的には、スコープ計画、定義、検証などがある。

(3) プロジェクト・タイム・マネジメント

プロジェクトを所定の時期に完了させるためのプロセスからなる。プロジェクト・タイム・マネジメントでは、作業の資源および所要時間の見積もりを通して、最終的にプロジェクト・スケジュールが立案される。

(4) プロジェクト・コスト・マネジメント

プロジェクト・コスト・マネジメントは、プロジェクトの各作業に必要な費用を明らかにし、その予算どおりにプロジェクトをコントロールすることを目的とする。プロジェ

クトを予算内で完了するために必要なコストの見積もり、予算化、コントロールなどのプロセスからなる。

(5) プロジェクト品質マネジメント

プロジェクトのニーズを確実に満足させるためのプロセスである。品質方針、目標および責任を定め、それら達成するために、品質計画、品質保証、品質管理および品質改善を実施していく。

(6) プロジェクト人的資源マネジメント

プロジェクトのチームメンバーのみならず、ステークホルダー全体をプロジェクトの目的達成のために効果的に活用するためのマネジメントである。プロジェクト人的資源マネジメントでは、組織計画や要員調達に加えて、チーム育成も含まれる。

(7) プロジェクト・コミュニケーション・マネジメント

プロジェクトの実行に関する情報を、プロジェクトメンバーを含めたステークホルダーに正確に届けるためのプロセスからなる。プロジェクトを成功に導くために、ステークホルダー間で各個人の考えや共有すべきプロジェクトマネジメント情報の流通を密接にすることを目的とする。

(8) プロジェクト・リスク・マネジメント

プロジェクトに関するリスクの管理の計画、特定、分析などの実施にかかるプロセスからなる。プロジェクト・リスク・マネジメントの目標は、プロジェクトに対してプラスとなる事象の発生確率と影響度を増加させ、マイナスとなる事象の発生確率と影響度を減少させることである。

(9) プロジェクト調達マネジメント

作業の実行に必要な製品、サービスをプロジェクトチームの外部から購入・取得するプロセスからなる。製品を調達するための計画の策定や外部との契約の管理などが含まれる。

プロジェクト管理はプロジェクトの計画と進捗管理の二つに大別される。プロジェクトの計画では、各作業の使用する資源の種類や処理時間の決定などが行われ、最終的な成果物と

してプロジェクト・スケジュールが作成される。プロジェクトの進捗管理では、スケジュール通りにプロジェクトを進めていくために進捗状況が監視される。また、プロジェクトの遅延が明らかとなった際には、必要に応じてスケジュールが変更される。本論文では、プロジェクト計画を研究の対象とし、プロジェクト・スケジュールを生成するための手法を提案する。また、進捗管理の数値実験を通じて提案手法の妥当性および有用性を検討する。

2.2 プロジェクト計画におけるスケジューリング問題

プロジェクト計画におけるスケジューリング問題（以下、スケジューリング問題）とは、多くの場合「諸活動を限られた資源の下で処理する際に、各作業をいつ、どの資源で処理するかを決定する問題」と定義される [1, 2]。スケジューリングを広義として捉えると、プロジェクトの遂行手順の決定、配送ルート決定、工場内での生産順序決定など様々な活動に応用されていることが分かる。

スケジューリング問題は、以下のように分類が可能である。

- 1) 各作業を完了させる場合において、順序制約が存在する場合。
- 2) 各作業を完了させる場合において、順序制約が存在しない場合。

1) に含まれるスケジューリング問題は、巡回型スケジューリング問題と呼ばれる。この問題は、巡回すべき需要地集合に対して、どのような順序でも構わず一度だけ通過すればよしとされる。次に、2) に含まれる問題は、資源配置型スケジューリング問題と呼ばれ、特に、処理に数種の資源を必要とする作業に対し資源配分および各作業の開始時刻を決定するスケジューリング問題を、資源制約付きプロジェクトスケジューリング問題 (Resource-constrained Project Scheduling Problem; RCPSP) と呼ぶ。例えば、いくつかの製品を製造する工場で、使用する機械や人など資源に限りがあるとき、与えられた制約を考慮し、効率的な仕事順序を求める問題がそれである。作業の処理方法について複数の選択肢が与えられている RCPSP をマルチモード型資源制約付きプロジェクトスケジューリング問題 (Multi-mode RCPSP; MRCPSPP) と呼ぶ。MRCPSPP は RCPSP の一般形であり、MRCPSPP において、仕事の処理方法が固定された特殊形が RCPSP である。さらに、工場などの生産現場で最も多いスケジューリング問題は、複数の仕事を対象としその加工順序と加工開始時刻を決めるジョブ・ショップ型である。この問題は、使用する資源数が1単位であり、RCPSP の特殊

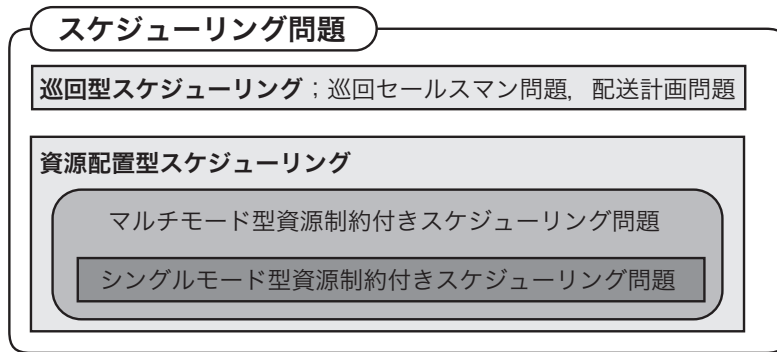


Fig. 2.1: スケジューリング問題における RCPSP の位置付け

形と捉えることができる。Fig. 2.1 にスケジューリング問題における RCPSP の位置付けを示す。

2.3 RCPSP の表現とクラス分類

RCPSP は、製品開発、構造物建築など対象は多岐にわたる。RCPSP は作業間の先行関係と資源容量の不足、競合を考慮しつつ目的関数の最適化を目指し、各作業の開始時刻を決定する問題である。

2.3.1 RCPSP の表現

プロジェクトを構成する作業の先行関係は、ネットワークを用いて表現することができる [3]。例として、六つの作業で構成されるプロジェクトを考えると、そのアローダイアグラムは、Fig. 2.2(a) のようになる。アローダイアグラムのノードは作業に対応し、矢印（アーク）は作業間の先行関係を意味する。このようなアローダイアグラムは Activity-on-Node (AoN) ネットワークと呼ばれる。AoN ネットワークによる表現方法の他に、作業をアークによって表現する Activity-on-Arc (AoA) ネットワークがある。Fig. 2.2(b) に AoA ネットワークを示す。本論文ではプロジェクトの表現に AoA ネットワークを用い RCPSP を表現する。また、各作業の処理の時間的流れは、資源量を縦軸に、時間経過を横軸にとったタイムチャートで表現する (Fig. 2.3)。各矩形が作業を表し、内部に示された番号が作業番号となる。また矩形の幅が処理時間、高さが作業の処理に必要な資源量を表している。

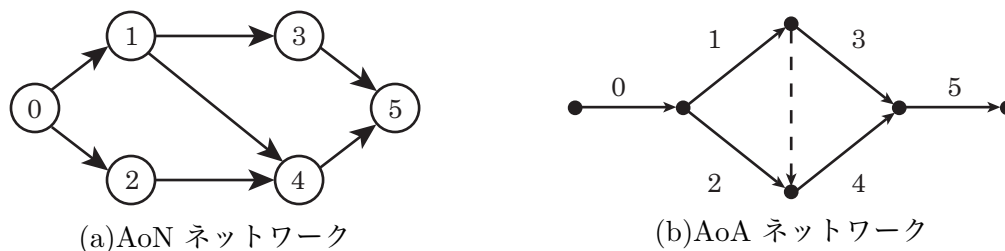


Fig. 2.2: ネットワークを用いたプロジェクトの表現

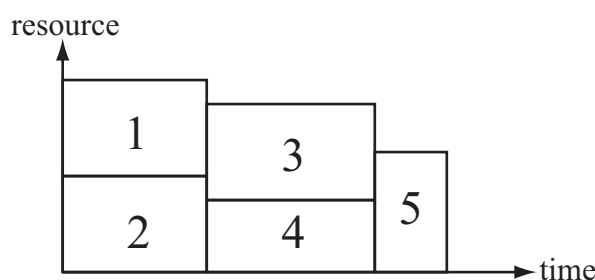


Fig. 2.3: タイムチャートを用いたスケジュールの一例

2.3.2 RCPSP のクラス分類

より現実に近い状況を扱うために、先に述べた RCPSP よりも一般的なプロジェクト・スケジューリング問題が、多くの研究者によって検討されている。ここでは、スケジューリング問題における RCPSP の位置付けをより明確にするため、資源環境 (α)、作業特性 (β) および目的関数 (γ) の三つの領域に区分し RCPSP のクラス分類を行う [4].

α : 資源環境領域

特定の機械スケジューリング問題と計画スケジューリング問題を区別するために α -領域では、*PS*(Project Scheduling) や *MPS*(Multi-mode Project Scheduling) といった表記を用いる。 *PS* という表記は、Blazewicz ら [5] の表記法を導入することにより、資源制約付きプロジェクトスケジューリングにおいて $PS_{m, \sigma, \rho}$ といった表記への拡張が可能となる。

- PS* : 計画スケジューリング
- MPS* : マルチモード型計画スケジューリング
- $PS_{m, \sigma, \rho}$: 資源数 m 個, 各資源の利用可能量 σ , 各資源が必要とする資源量は最大で ρ

もし、 m, σ, ρ が * に置き換えられていた場合、そのパラメータ値は入力値に依存する。また、 $PSm, *, *$ や $PSm, \sigma, *$ と表記されている場合は、省略してそれぞれを PSm , PSm, σ と表記する。全ての値 m, σ, ρ が入力値に依存するのなら、 $*, *, *$ と表記する代わりに * のみで表記する。以下に、表記の一例を示す。

- $PSm, 1, 1$: 資源数 m 個、各資源の使用可能量は 1,
各仕事が使用する資源量は最大で 1
 PSm, ∞ : 資源数 m 個、使用可能資源量に制約なし (資源制約が存在しない)
 $PS1$: 資源数は一つ

β : 作業特性領域

β -領域では作業の特性について表記を行う。

- $p_i = 1$: 全作業の処理時間は 1 である
 $p_i = sto$: 作業の処理時間は確率分布に従う
 d : 計画期間の締め切り
 $prec$: 作業間に先行制約がある
 $chains,intree,$: 作業間の先行関係は、鎖、内木、外木、木などにより特定される
 $outtree,tree\dots$
 $temp$: 作業間における開始時刻の差の最小値または最大値により得られる一時的な制約

γ : 目的関数領域

γ -領域では、目的関数についての表記を行う。機械スケジューリング問題と同様に、計画スケジューリング問題においても種々の目的関数が与えられる。頻繁に用いられる C_{max} (完了時刻最小化), L_{max} (最大納期ずれ), $\sum w_i C_i$ (重み付き完了時刻最小化) などに加えプロジェクトスケジューリング問題では、以下のような目的関数も用いられる。

- $\sum c_i^F \omega^{C_i}$: 正味現在価値 (c^F : キャッシュフロー, ω : 割引係数, i : 作業)
 $\sum c_l f(r_l(S, t))$: 資源水準測量 (c_l : 資源 l を使用する際の単位あたりのコスト,
 $r_l(S, t)$: 与えられたスケジュール S において時刻 t に資源 l を利用する場合の利用方法)
 $\sum c_l \max r_l(S, t)$: 資源の投資

以上の表記を用いて様々なモデルを表現することが可能となる。次にクラス分けの一例を示す。

$PS | prec | C_{max}$: このモデルは、資源制約付きスケジューリング問題の中で最も基礎となるモデルである。先行制約と資源制約を考慮しつつ、全ての仕事の完了時刻を最小化するモ

デルである.

$MPS | prec | C_{max}$: 仕事の処理方法 (または, 処理時間) は複数のモードから選択でき, 先行制約と資源制約を考慮しつつ, 完了時刻を最小化する問題である. この問題では, モードの違いによる資源の増減分, 処理に要する時間と使用資源量の間が発生するトレードオフを考慮している.

$PS | p_i = sto | C_{max}$: 各作業の処理時間がある確率分布に従う場合において, 完了時刻を最小化するモデルである.

本論文では, RCPSP の中で最も基礎となるモデル $PS | prec | C_{max}$ を採用し, 初期解を生成する. モデルの詳細は, 次にて述べる.

2.4 プロジェクトのモデル

2.4.1 プロジェクトの記述

プロジェクトの期日 (あるいは納期) を DL , プロジェクト実行期間を $[0, T]$ とする. プロジェクトは N 個の作業で構成され, 作業集合を $\mathcal{A} = \{1, 2, \dots, N\}$ とし作業 $i (i = 1, 2, \dots, N)$ の処理時間を $p_i (p_i \geq 1)$ とする. また, 作業間には先行関係が存在し記号を次のように定義する.

\mathcal{P}'_i : 作業 i の開始時刻には完了していなければならない作業全ての集合

\mathcal{P}_i : \mathcal{P}'_i 中の作業 i の直前に処理されなければならない先行作業の集合

$$\mathcal{P}_i = \{j \mid j \neq i, \forall k \in \mathcal{P}'_k\}$$

\mathcal{Q}_i : 作業 i 完了後ただちに開始することが可能な作業 (後続作業) の集合

$$\mathcal{S}_i = \{j \mid i \in \mathcal{P}_j\}$$

プロジェクトを表すアローダイアグラムとして本論文では, 作業がアークに対応する AoA (Activity on Arc) 表現を用いるためプロジェクトの開始および完了を表す架空の作業 0 ($\mathcal{P}_0 = \phi, \mathcal{Q}_0 = \{i \mid \mathcal{P}_i = \phi\}$), および作業 $N+1$ ($\mathcal{P}_{N+1} = \{i \mid \mathcal{Q}_i = \phi\}$) を新たに導入する. ただし, $p_0 = 0, p_{N+1} = 0$ である. 架空の作業の導入にともなって整合性を保つ目的で, $\mathcal{Q}_i = \phi$ なる作業 $i \in \mathcal{A}$ について $\mathcal{P}_i = \{0\}$, $\mathcal{Q}_i = \phi$ なる作業 $i \in \mathcal{A}$ について $\mathcal{Q}_i = \{N+1\}$ と更新する.

2.4.2 資源制約

資源制約は、資源の機能によって再生可能資源制約と再生不可能資源制約の二種類に分けられる。再生可能資源制約とは、単位時間が経過すると再び使用可能になるような資源のことである。例として機械や人員などが挙げられる。次に、再生不可能資源制約とは、一度消費された分については再度使用ができない資源のことである。例として資金や資材などが挙げられる。

一般にプロジェクトの実行過程において、作業のいくつかはその開始時刻が当初の予定よりずれる場合が多い。資源の利用に時間制約が課せられている状況や、資源量自体が時間的に変化する場合は、開始時刻のずれは資源の競合や不足を生じさせる [19]。これを解消する一つの方策として、CC/BM (Critical Chain Buffer Manegement) に見られるように、計画段階において時間的ゆとりの挿入を考慮したうえで資源制約が付与される必要がある [20]。一方、必要な資源の見積もりと手配、さらにはプロジェクトの遅延による資源の競合や不足といった問題は、動的に対応せねばならない。本論文では、資源の使用に対する時間的制約や資源量の時間的変化は対象外とし、各時刻において使用可能量の上限を持つ資源 $l (l = 1, 2, \dots, R)$ の供給量 \tilde{r}_l は、プロジェクト実行期間 $[0, T]$ のどの時刻においても一定である再生可能資源とする。すなわち、作業 i の処理時間において各単位時刻に必要な資源 l の量を r_{il} と表すと、各時刻 $t (t = 0, 1, \dots, T)$ での資源 l に関する制約は、

$$\sum_{i \in \mathcal{A}_t} r_{il} \leq \tilde{r}_l (t = 0, 1, \dots, T, l = 1, 2, \dots, R) \quad (2.1)$$

で表される。ただし、 \mathcal{A}_t は時刻 t で処理中の作業集合を表す。

2.4.3 基準スケジュール

所与のプロジェクトに対して計画されたスケジュールを基準スケジュールと呼ぶ。この基準スケジュールは、RCPSP に対する可能解として生成されるものとする [21]。

基準スケジュール π のメイクスパンを $MS(\pi)$ とし、 π 上での各作業の開始時刻を $s_i(\pi)$ で表す。ただし、基準スケジュール上では $s_i(\pi)$ は作業 i の最早開始時刻に一致する。また、架空の作業 $s_0(\pi)$ および $s_{N+1}(\pi)$ の開始時刻を $s_0(\pi) = 0$ および $s_{N+1}(\pi) = MS(\pi)$ とする。

本論文では、プロジェクトの実行過程で管理する基準スケジュール π 上のクリティカル・

チェーンが, 次の式 (2.2), (2.3) を満足する $N_c(\pi)$ 個のクリティカル作業 $\mathcal{C}^\pi\{a(1), a(2), \dots, a(N_c(\pi))\}$ ($\mathcal{C}^\pi \subseteq \mathcal{A}, N_c(\pi) \leq N$) で構成されるとする.

$$s_{a(j)}(\pi) + p_{a(j)} = s_{a(j+1)}(\pi), \quad j = 1, 2, \dots, N_c(\pi) - 1, \quad (2.2)$$

$$\sum_{j=1}^{N_c(\pi)} p_{a(j)} = MS(\pi). \quad (2.3)$$

また, クリティカル・チェーン上にない $N - N_c$ 個の作業 (集合 $\mathcal{A} \setminus \mathcal{C}$) を通常作業と呼ぶこととする. 通常作業 i' ($i' \in \mathcal{A} \setminus \mathcal{C}$) の開始時刻 $s_{i'}(\pi)$ は,

$$s_{i'}(\pi) = \max_{k \in \mathcal{P}_{i'} \cup \tilde{\mathcal{P}}_{i'}} (s_k(\pi) + p_k) \quad (2.4)$$

である. ただし, $\tilde{\mathcal{P}}_{i'} (\neq \mathcal{P}_{i'})$ は, 基準スケジュール生成過程の生成過程で作業 i' と資源競合の関係にある作業のうち競合解消の結果, 作業 i' に先行して処理されることとなった集合をあらわす.

2.5 動的スケジューリング

プロジェクトの実行中に生起する不確実的事象への対応を考慮したスケジューリング手法に動的スケジューリングがある. 動的スケジューリングは, プロジェクト実施中に新たなスケジュールへ更新, 既存のスケジュールの修正を行う手法である [6, 7]. スケジュール立案時期, スケジューリングにおける意思決定のタイミングに注目した場合, スケジューリングは, 図 2.4 に示すように以下の三つで実施することが可能である.

(1) プロアクティブ・スケジューリング

計画立時にスケジュールを生成するアプローチ

(2) リアクティブ・スケジューリング

スケジュール実施中に事前 (計画立案時) に生成したスケジュールでは対応が困難となる場合や偶発的にスケジュールが遅れた場合などに対応するアプローチ

(3) リアルタイム・スケジューリング

事前にスケジュールを作成することなく, 着手できる作業を逐次実施するアプローチ

(1) のプロアクティブ・スケジューリングは, 計画立案時においてスケジュールを生成するが, その時に不確実的事象を考慮した立場をとる. そのため, スケジュール実施中は可能

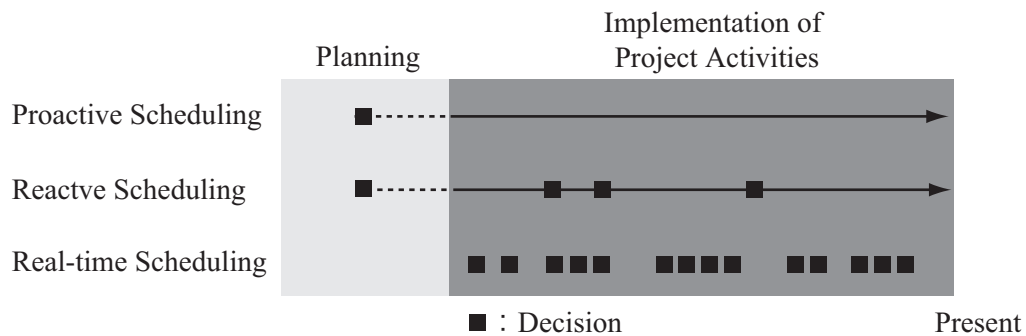


Fig. 2.4: スケジューリング問題における意思決定のタイミング

な限りスケジュールの修正を実施しない。これは静的スケジューリングとよばれ動的スケジューリングとは対極的な技法である。プロアクティブ・スケジューリングに関する研究の多くは、不確実的事象に耐えうる頑健性をスケジュールに与えることに主眼が置かれている。プロアクティブ・スケジューリングに関する研究動向は、Van de Vonder ら [8], Bruni ら [9], Deblaere ら [10] の文献が詳しい。

次に、(2) のリアクティブ・スケジューリングは、計画立案時で生成したスケジュールに対し不確実的事象の影響に応じてスケジュールの更新・修正を実施する。これは、スケジュールの反復的な改善操作に着目することになり、ヒューリスティクスや知識ベースに基づくことが多い。改善操作そのものは、計算量 [7] や手続きの制約から、ディスパッチングルールといった比較的シンプルな操作に限定される。

最後に (3) のリアルタイム・スケジューリングは、着手できる作業を逐次実施するためプロジェクトの存在する制約条件の影響を考慮しつつスケジュールを生成せねばならない。このため、着手可能作業に対してシンプルな形式で記述されたルール [11, 12] を用いスケジュールを生成することが多い。

2.5.1 リアクティブ・スケジューリング

2.5 において、スケジューリングにおける意思決定のタイミングについて説明した。リアクティブ・スケジューリングのスケジュール生成手法に着目すると、意思決定のタイミングと併せ以下の二つのアプローチを取ることができる。

(1) スケジュール生成：スケジュールを更新する際に用いる手法の決定

(2) 意思決定のタイミング：スケジューリングを実施するタイミングの決定

スケジュール生成に関しては、

- ▷ ヒューリスティクス，メタ・ヒューリスティクス [13, 14, 15]
- ▷ プライオリティルール，ディスパッチングルール [16, 17]
- ▷ 知識ベース [18]

などの手法が挙げられる。スケジュール生成の意味からすれば、事前にスケジュールを生成することとスケジュールを更新・修正することは同等であるため、プロアクティブ・スケジューリングの考え方を取り入れることも可能である。

次に、意思決定のタイミングでは、

- ▷ 定期型：定期的にスケジュールを確認 [22, 46]
- ▷ 事象駆動型：不確実的事象の発生を契機にスケジュールを確認 [46, 24]
- ▷ マイルストーン型：重点管理作業など事前に決定しておいた事項が完了した場合にスケジュールを確認 [25]

が挙げられる。

2.6 プロジェクト管理の技法

2.6.1 PERT/CPM

PERT (Program Evaluation and Review Technique) [26] と CPM (Critical Path Method) [27] はいずれも大規模なプロジェクトを計画・管理するための技法として開発された。ともにネットワークを中心とした手法である。ただし、PERT は最適化を直接目的とせず、CPM はそれを直接目的とする。また、PERT における作業の処理時間は、不確定的に三点（悲観値、最可能値、楽観値）で見積もられる確率モデルであるのに対し、CPM は一点見積もりの確定モデルである。

PERT/CPM の計算式

はじめに、PERT/CPM で用いられる計算手法を紹介する。

▷ 最早開始・完了時刻

作業が最も早く開始・完了出来る時刻をさす。プロジェクトの開始を表す作業を 0 とすると、作業 i の最早開始時刻 ES_i および最早完了時刻 EC_i は次式で算出される。

$$ES_0 = 0, \quad (2.5)$$

$$ES_i = \max\{ES_j + p_j \mid j \in \mathcal{P}_i\}, \quad i \in \mathcal{A} \setminus \{0\}, \quad (2.6)$$

$$EC_i = ES_i + p_i, \quad i \in \mathcal{A}. \quad (2.7)$$

ただし、 \mathcal{P}_i は、作業 i の先行作業集合である。

▷ 最遅開始・完了時刻

作業がプロジェクトのメイクスパンを守る範囲で最も遅く開始・完了できる時刻を示す。作業 i の最遅開始時刻 LS_i および最遅完了時刻 LC_i は次式で算出される。このとき、プロジェクトの完了を表す作業を $N + 1$ とする。

$$LC_{N+1} = EC_{N+1}, \quad (2.8)$$

$$LC_i = \min\{LC_j - p_j \mid j \in \mathcal{Q}_i\}, \quad i \in \mathcal{A} \setminus \{N + 1\}, \quad (2.9)$$

$$LS_i = LC_i - p_i, \quad i \in \mathcal{A}. \quad (2.10)$$

ただし、 \mathcal{Q}_i は、作業 i の後続作業集合である。

▷ 全余裕

プロジェクトのメイクスパンを逸脱しない範囲で作業 i が持つことが出来る余裕時間を全余裕と呼び、 TF_i で表す。

$$TF_i = LS_i - ES_i = LC_i - EC_i, \quad i \in \mathcal{A}. \quad (2.11)$$

▷ 自由余裕

後続作業の開始時刻に影響を与えない範囲で持つことができる余裕時間を自由余裕と呼ぶ。作業 i の自由余裕 FF_i は、最早完了時刻と直後の後続作業の最早開始時刻から

算出できる.

$$FF_i = \min\{ES_j - EC_i \mid j \in Q_i\}, \quad i \in A. \quad (2.12)$$

クリティカル・チェーン

PERT/CPMの計算を行うことにより、クリティカル・チェーンが明らかとなる。クリティカル・チェーンとは、AoAネットワークの作業0から作業 $N-1$ へ到達する経路の中で全余裕が0である作業からなる一連の経路である。クリティカル・チェーン上にある作業は、プロジェクトのスケジュール上で全く余裕がなく、もしそのうちどれか一つでも遅延すれば遅延分だけ工期が遅れるような作業である。反対に、もし工期を短縮したい要求が起これば、これらの作業の処理時間を短縮せねばならない。クリティカル・チェーン以外の作業の処理時間を短縮しても、工期の短縮には直結しない。したがって、クリティカル・チェーンに含まれるクリティカル作業は重点的に管理せねばならない作業といえる。一般的に、プロジェクトに含まれるクリティカル作業は割合は低く10~30%の割合（プロジェクトの規模に依存）であることから、管理上非常に有意義であるといえる。

プロジェクトのコスト曲線

ある一定期間でプロジェクトを完了させるためには、クリティカル作業に対し縮約を実施しなければならないことは明らかである。しかしながら、プロジェクトの完了時刻を短縮するにはコストが必要となる。ここでコストは、直接的なコスト（直接コスト）として、人件費、資材または機械などが挙げられ、間接的なコスト（間接コスト）としてプロジェクト管理費、配賦費などがこれに相当する。一般的に直接コスト、間接コストはプロジェクトの実施期間が延びるとともに増加していく。

一方で、プロジェクト期日を早く仕上げれば奨励金が貰える場合も考えられる。また、納期違反により発生するペナルティをコストと捉えることもできる。これらを、機会損失コストとする。プロジェクトの総コストは、直接コスト、間接コスト、機会損失コストの総和となりプロジェクト実行期間の関数となる。Fig. 2.5より、総コスト曲線における最適点においてプロジェクトを実行するのが最も有効となる。

CPMでは、日程短縮と費用増加の関係を明確にすることにより、最小の費用増加で日程短縮を図る。日程短縮の際には、各作業の標準所要時間と、特急所要時間に加えて、作業を

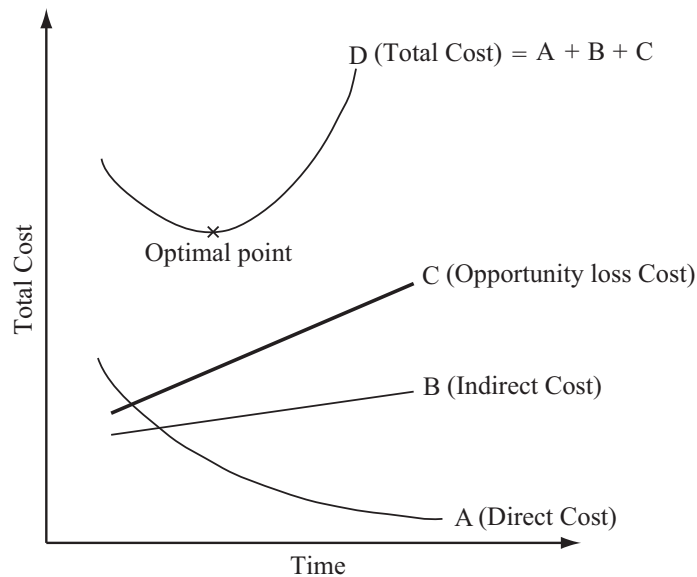


Fig. 2.5: プロジェクトに存在するコスト

一単位時間縮める事により発生する費用である費用勾配も与えられる。プロジェクトの長さが短縮できる作業の組み合わせをカットと呼ぶ。また、カットの中で費用の一番低いカットを最小カットと呼ぶ。まず、日程短縮の手順を以下に示す。

Step 1: クリティカル・チェーンを特定する。

Step 2: クリティカル・チェーン上で単位時間当たりの費用勾配が最小であるカットを算出する。

Step 3: 最小カットの作業または作業の組の所要時間を、その費用勾配で縮められる限界まで縮めた日程を作成する。

Step 4: Step 3 の日程の短縮による費用の増加分を計算する。

Step 5: 新たに得られた日程においてカットが存在しなければ、これ以上の短縮はできない。そうでなければ、Step 1 に戻る。

以上の手順より得られる時間短縮とそれにともなう費用の増加を図示することにより、プロジェクト費用曲線を作成することができる。

2.6.2 生産業界における CPM の役割

産業界における CPM の役割は、経営者を含む上位レベルと管理者を含む下位レベルではそれが異なる。ここでは、Galloway[28] の文献をもとに各レベルにおける CPM の役割を紹介する。

上位レベル

上位レベルにおける CPM は、プロジェクトの全体像を把握するために用いられる。一例として、必要予算および品質方針の決定、成果物を得るまでの必要な作業の洗い出しなどが挙げられる。また、プロジェクトの可視化や受託業者との進捗状況の把握にも用いられる。このように、上位レベルにおける CPM はプロジェクトの理解を容易にするために利用されている。

下位レベル

下位レベルにおける CPM は、主に計画および進捗の管理に利用される。また、上位レベルに対して問題作業の特定、または資源の過剰供給または不足などを的確に報告するために用いられる。CPM を用いる利点を Fig.2.6 に示す。Fig.2.6 より計画スケジュールの改善、スケジュールリングへの利用、プロジェクトの理解を向上させることなどが利点として挙げられている。また、リスク増加のや遅延の削減といった不確実的事象の削減にも用いられている。

CPM は標準的なプロジェクト管理ツールとなっており、上位・下位レベルにおいて有益な情報を与えていることが分かる。また、受託業者と意思決定を円滑な実施にも利用可能であり、プロジェクト管理において重要な役割を果たす。

2.7 結言

本章前半部では、プロジェクト管理におけるスケジュールリング問題として、まずプロジェクト・マネジメントとしてのプロジェクト管理の基盤を紹介した。次に、スケジュールリング問題を定義し現実問題により近い状況を扱うための問題の拡張を行い、多くの状況を取り扱うことを可能とした。後半部ではプロジェクトの実施中に発生する問題、すなわち不確実的事象への対処として、スケジュールリングの立場からの解決手法を紹介しそれに伴う意思決定の

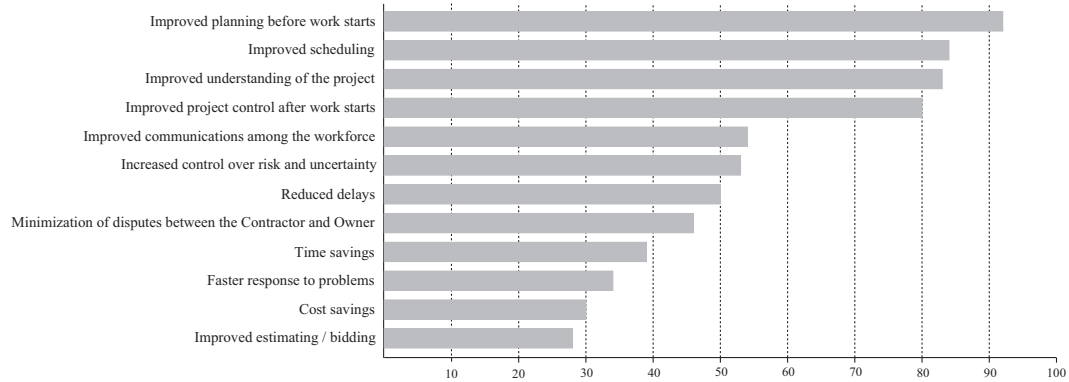


Fig. 2.6: CPM の役割

タイミングについて述べた。最後に、プロジェクト管理における一般的な手法 PERT/CPM を紹介し、生産業界における CPM の役割を紹介した。

第3章 Critical Path Method に基づく動的プロジェクト・スケジューリング

本章では、動的スケジューリングにおけるスケジュール修正に対し、各作業の処理時間の縮約を取り入れた手法を提案する。この手法は、プロジェクトの期日を逸脱した場合において、いくつかの作業の処理時間を縮約することにより期日を遵守する。縮約に必要な追加コストを最小化し期日の遵守および作業間の先行関係を絶対制約として問題を定式化し縮約の対象となる作業を抽出する。その後、抽出された作業に対して縮約を実施する手法である。数値実験では提案手法を実施するにあたり、有効なスケジュール修正実施時点を調べることを目的とした予備実験を、次にスケジュール事象駆動型、定期型および余裕時間型の3種類を用い比較実験を行い提案手法の特性を述べる。

3.1 背景

プロジェクトの目標達成には、計画、実行、確認、処置といった一連の作業からなるプロジェクト管理が重要となる [35]。計画段階において基準スケジュールを立案し、それに基づきプロジェクトを実行していく。しかしながら多くの場合、実施過程で生じる不確実的事象への対処が課題となる [36]。不確実的事象の影響により、基準スケジュール通りに実行し完了させるのが困難となる。原因として考えられるのは、作業の遅延、必要資源量の変動、作業間制約の変更などが挙げられる。このとき、基準スケジュールと不確実的事象後のスケジュールとの差異を確認し必要であれば処置を講じる必要がある。このため、管理者はプロジェクトの実施中においてスケジュールの変更、新たに必要となる資源の追加投入、一部の作業に対するスケジューリング手法の適用など意思決定が必要とされる。

3.2 CPM に基づく動的スケジューリング

本章では、いくつかの作業の処理時間を縮約することによりプロジェクト期日を遵守するアプローチを採用する。大幅な縮約は資源の追加にかかるコストの増加を招き、このコストを重視しすぎると処理時間の縮約の効果が薄れ、結局期日に間に合わない恐れがある [29, 30]。このような処理時間の縮約とそれの実施に必要な追加コストのバランスを考慮するために、CPM に基づいた動的スケジューリング手法を提案する。

3.2.1 提案手法

提案手法では、不確実的事象の影響によりプロジェクト期日を逸脱した場合に限り、処理時間の縮約を可能とする。時刻 t' において作業 $i' (i' \in \mathcal{A}_{t'})$ に不確実的事象が発生したとする。作業 i' の完了時刻は、 $f_{i'}(\pi) = s_{i'}(\pi) + p_{i'}$ から $f_{i'}(\tilde{\pi}) = f_{i'}(\pi) + \delta$ と変化する。このとき、スケジュールの実行可能性の維持を目的として、右シフト操作 [21] を実施する。右シフト操作実施後に得られた新たなメイクスパン（以下、実際のメイクスパン）を $MS(\tilde{\pi})$ とする。 $\tilde{\pi}$ への更新は、 $t > t'$ 以降に発生した不確実的事象に対して全て適用される。ただし、作業 i の完了時刻が、最遅完了時刻を超えた場合に限り、

$$MS(\tilde{\pi}) = MS(\pi) + (f_i(\tilde{\pi}) - f_i^l(\tilde{\pi})) \quad (3.1)$$

とする。この時、 $f_i^l(\tilde{\pi})$ は作業 i の最遅完了時刻を表す。

プロジェクト期日を遵守するには、クリティカル・チェーンに対し対策を講じなければならない [31, 32]。しかしながら、基準スケジュール生成時に決定したクリティカル・チェーンは、不確実的事象の影響で変化する可能性がある。その結果、通常作業がクリティカル作業になり、またその逆も起こりうる。プロジェクト期日を短縮するためには、クリティカル作業に対して処理時間の縮約を実施せねばならない。したがって縮約対象となる作業の抽出の前処理として PERT 計算を実施しクリティカル作業およびクリティカル・チェーンを再度決定する。抽出された作業に対し、追加コストを付与することにより処理時間の縮約をはかる。また、総追加コストの最小化を目的として定式化する。

縮約対象作業の抽出

時刻 t' において作業 i' に不確実的事象が発生したとする。このとき、時刻 $t' (< t)$ において実際のメイクスパン $MS(\bar{\pi})$ がプロジェクト期日 DL を超えたとする ($MS(\bar{\pi}) > DL$)。逸脱した場合に限り、作業集合 \mathcal{A}'_t に対して作業の抽出を実施する。抽出された作業集合を $\mathcal{F}_t^{(i')} (\mathcal{F}_t^{(i')} \subseteq \mathcal{A})$ と定義する。

縮約対象作業の抽出の例として、Fig.3.1(a) を用いる。基準スケジュールのメイクスパン $MS(\pi) = 12$ 、資源の種類は一種類 ($R = 1$) である。いま、時刻 $t' (0 < t' < 5)$ において、作業 2 に遅延が発生したとする。このとき、時刻 5 において右シフト操作を適用すると、実際のメイクスパン $MS(\bar{\pi}) = 15$ となる (Fig.3.1)。基準スケジュールにおいて、作業 2 ($i = 2$) の完了時刻は $t = 3 (f_2(\pi) = f_2^l(\pi) = 3)$ であるが、不確実的事象の影響で時刻 $t = 6$ となる。この結果、後続作業 4,5 ($\mathcal{Q}_2 = \{4, 5\}$) は当初の予定の開始時刻に処理を開始することができない。 \mathcal{Q}_2 に続く作業 7,8 においても同様である。スケジューリング時刻 $t = 5$ において作業集合 \mathcal{A}_5 には作業 2 および 3 が含まれており、この時刻以降の未処理の作業集合 $\mathcal{F}_5^{(2)} (i = 2, t = 5)$ には、作業 4, 5, 7, 8 が属することになる。

作業時間の縮約と追加コスト

3.2.1 でも述べたが、提案手法では、不確実的事象の影響でプロジェクト期日を逸脱した場合 ($MS(\bar{\pi}) > DL$) に限り処理時間の縮約の実施が可能となる。処理時間を一単位時間ずつ縮約することにより期日を遵守させる方策である。各作業が持つ処理時間を p_i とする。各作業には処理時間の下限値 \bar{p}_i が与えられその範囲内において縮約が可能となる。次に一単位時間縮約した処理時間を \tilde{p}_i とする。 \tilde{p}_i は、縮約されるたびに更新される。ただし、処理時間 $p_i = 1$ の作業は縮約に対象とはならない。

次に、各作業を一単位時間縮約するために必要な追加コストを定義する。プロジェクトを構成する作業と必要な資源は直接的な関係があるとして、一単位時間の縮約に必要な追加コストは、

$$AC_i^l = \left\lceil \frac{r_{il}}{p_i} \right\rceil, \quad (l = 1, \dots, R, i \in \mathcal{F}_t^{(i)}). \quad (3.2)$$

で与える。したがって、作業 i における処理時間 p_i から縮約後の処理時間 \tilde{p}_i で必要な追加

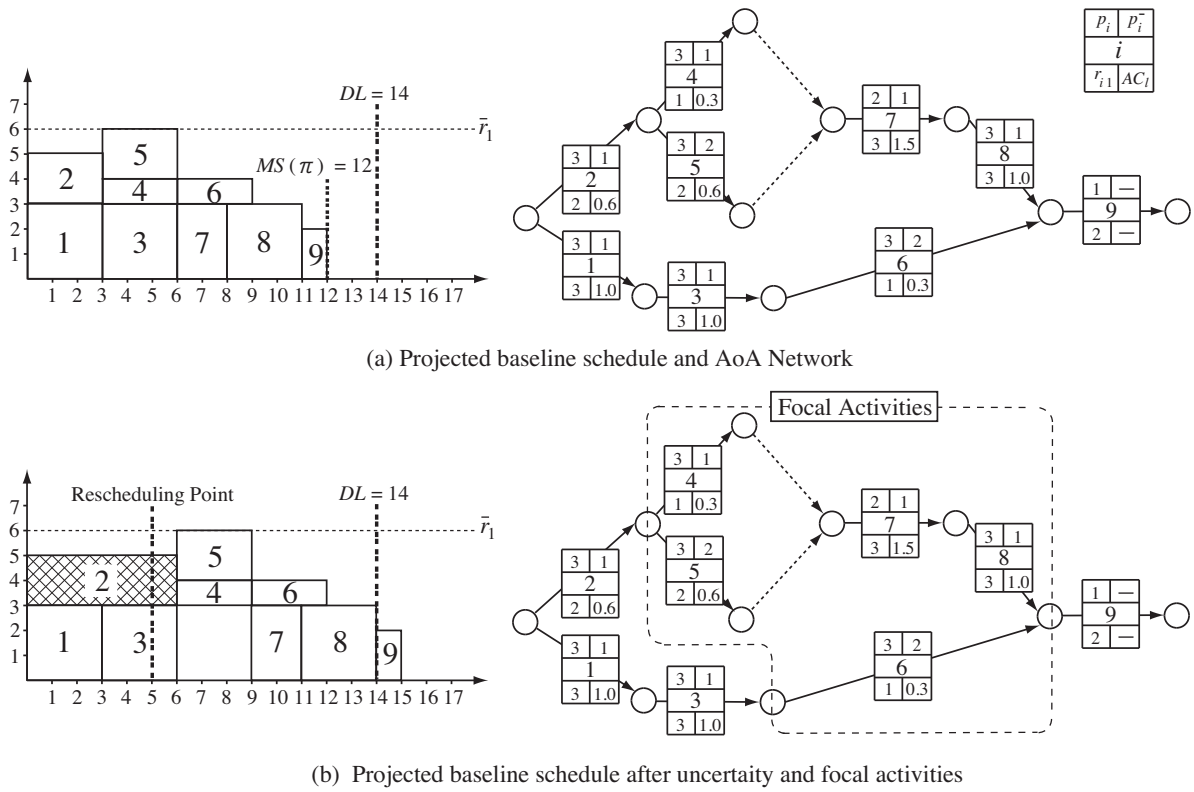


Fig. 3.1: 作業2に発生した遅延後のスケジュール

コストは,

$$AC_i = (p_i - \tilde{p}_i) \sum_{l=1}^R AC_i^l \tag{3.3}$$

で与えられる。Fig.3.1において作業5の処理時間は $p_5 = 3$ である。作業5の資源要求量は、資源の種類は1、要求量は単位時間あたり2である。このとき、追加コストは $AC_5^1 = \lceil r_{5,1}/p_5 \rceil = \lceil 2/3 \rceil = 1$ となる。したがって、作業5の縮約後の処理時間 $\tilde{p}_5 = 1$ とするならば、追加コストは $AC_5 = (p_5 - \tilde{p}_5)AC_5^1 = 2$ と算出される。

定式化

本章では、絶対制約を満たす（実行可能）範囲で、目的関数の満足度を最大化する。プロジェクト期日の遵守と先行制約を必ず満たさなければならない絶対制約とし、資源制約を満たすことが望ましい考慮制約とする。この制約下において、処理時間の縮約に対して発生する各資源の追加コスト $AC_l (l = 1, 2, \dots, R)$ の総和を TC とし、縮約後に得られたメイクス

パン $MS(\pi_{\tilde{p}})$ の二つを用い解（実際のスケジュール）を評価する。追加コスト和を、

$$TC = \sum_{i \in \mathcal{F}_t^{(i)}} AC_i \quad (3.4)$$

で表し、縮約後のメイクスパンを

$$MS(\pi_{\tilde{p}}) = \sum_{j=1}^{N_c(\pi_{\tilde{p}})} \min(p_{a(j)}, \tilde{p}_{a(j)}) \quad (3.5)$$

とする。式 (4.8) における $N_c(\pi_{\tilde{p}})$ はスケジュール \tilde{p} におけるクリティカル作業を表す。スケジュール \tilde{p} のクリティカル・チェーンは、 $C^{\pi_{\tilde{p}}} = \{a(1), a(2), \dots, a(N_c(\pi_{\tilde{p}}))\}$ となる。

以上より、時刻 t において \tilde{p} を決定するモデルは、

$$\text{minimize} \quad TC \quad (3.6)$$

$$\text{subject to} \quad MS(\pi_{\tilde{p}}) \leq DL, \quad (3.7)$$

$$s_i + \min(p_i, \tilde{p}_i) \leq s_j, \quad (i \in \mathcal{F}_t^{(i)}, i \neq j, j \in \mathcal{Q}_i) \quad (3.8)$$

$$\sum_{i \in \mathcal{A}_{t'}} r_{il} \leq \tilde{r}_l, \quad (l = 1, \dots, R, t' = t + 1, \dots, DT) \quad (3.9)$$

$$1 \leq p_i^- \leq \tilde{p}_i < p_i, \quad (i \in \mathcal{F}_t^{(i)}) \quad (3.10)$$

と記述できる。式 (3.7) は、実際のメイクスパンがプロジェクト期日を超えてはならないことを示し、式 (3.8) は、作業間の先行制約を表す。次に式 (3.9) は、資源制約を表し、式 (3.10) は各作業の縮約可能範囲を表している。

制約を考慮した縮約の手順

プロジェクト期日を遵守するには、クリティカル・チェーンを構成するクリティカル作業に対して縮約を実施しなければならない。しかしながら、縮約を実施した結果クリティカル作業が通常作業へと変化する可能性がある。このため、縮約の対象となる作業を明確にするため縮約実施後に PERT 計算を用い、新たにクリティカル・チェーンを決定する必要がある。このとき算出されるメイクスパンを、 $MS(\pi)$ とする。

ここで、縮約の対象となる作業集合を $\mathcal{F}_t^{(i)}$ とする。これをカットと呼ぶ。 $\mathcal{F}_t^{(i)}$ に含まれる作業は、 $\mathcal{C} = \{C_1, C_2, \dots, C_{n'}\} (n' = |\mathcal{C}|)$ とする。 $\mathcal{F}_t^{(i)}$ に対し、以下の手順で縮約を実施する。

1° $\mathcal{F}_t^{(i)}$ に含まれる作業を用い、部分ネットワークを生成する。

2° 部分ネットワーク $\mathcal{F}_t^{(i)}$ において列挙できるカット \mathcal{C} を生成する。

i) カット \mathcal{C} に含まれる $\mathcal{C}_j (C_j \subseteq \mathcal{C})$ 対し、必要となる追加コスト、

$$AC(j) = \sum_{i \in \mathcal{C}_j} AC_i \quad (j = 1, \dots, n') \quad (3.11)$$

を算出する。

ii) 最小となるカット \mathcal{C}_{j^*} を選択する。ただし、 $AC(j^*) = \min\{AC(1), \dots, AC(n')\}$ である。

3° \mathcal{C}_{j^*} に含まれる作業 i' の処理時間 $p_{i'}$ を一単位時間縮約する。この時、 $p_{i'} \leftarrow p_{i'} - 1 (i' \in \mathcal{C}_{j^*})$ および $\tilde{p}_{i'} \leftarrow p_{i'}$ と更新する。

4° 新たなメイクスパン $MS(\bar{\pi})$ および、総追加コスト (TC) を算出する。

5° $MS(\bar{\pi}) \geq DL$ なら 6° へ。それ以外は、2° へ。

6° 総追加コスト TC および新たなメイクスパン $MS(\bar{\pi})$ を出力後、終了する。

Fig.3.2(a) に $\mathcal{F}_t^{(i)}$ に含まれる作業の部分ネットワークを、Fig.3.2(b) に各カットを示す。このときのカットの組み合わせは、

$$\mathcal{C} = \{(4, 5, 6), (4, 5), (6, 7), (7), (6, 8)\}$$

となる ($n' = 5$)。Fig.3.2(a) において部分ネットワーク $\mathcal{F}_5^{(2)}$ となり、Fig.3.2(b) において各カットは、 $C_1 = (4, 5, 6)$, $C_2 = (4, 5)$, $C_3 = (6, 7)$, $C_4 = (7)$, $C_5 = (6, 8)$ となる。このとき、追加コスト AC は、式 (3.11) より、

$$AC(1) = 2, AC(2) = 1, AC(3) = 2, AC(4) = 2, AC(5) = 2$$

となる。

このとき、最小追加コストのカットは $C_2 (j^* = 4, 5)$ であり、 C_2 に含まれる作業を一単位時間ずつ縮約することにより、 $MS(\bar{\pi}) = 14$, $TC = 1$ を得る。Fig.3.3 にその結果を示す。

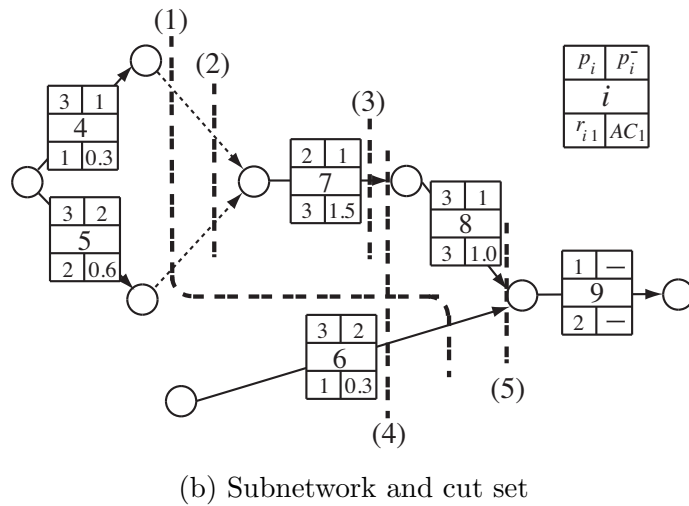
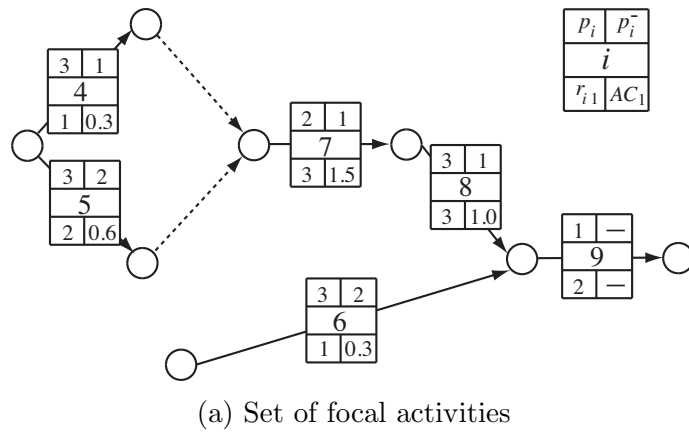


Fig. 3.2: 部分ネットワークとカット集合

3.3 数値実験

3.3.1 実験条件

問題例

PSPLIB の問題例から作業数 30 である j3014.1 ($N = 30, R = 4$), j3015.1 ($N = 30, R = 4$), j3022.1 ($N = 30, R = 4$), j3023.1 ($N = 30, R = 4$) と, 作業数が 60 である j6014.1 ($N = 60, R = 4$), j6015.1 ($N = 60, R = 4$), j6022.1 ($N = 60, R = 4$), j6023.1 ($N = 60, R = 4$) の八種類を取り上げる. これらの問題例に対して, 基準スケジュールを生成する. 得られたメイクスパン $MS(\pi)$ に対して期日を設定する. 期日 DL は $DL = 1.2MS(\pi)$ とした.

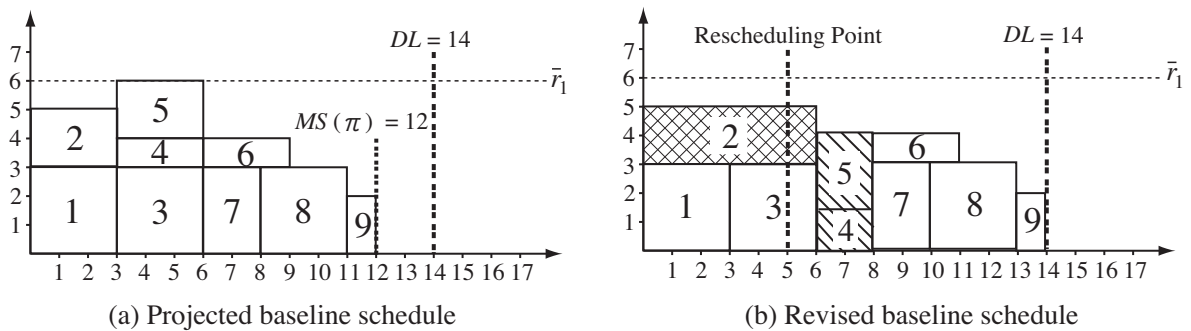


Fig. 3.3: 提案手法適用後のスケジュール

基準スケジュールの生成

作業数が30である j3014.1, j3015.1, j3022.1, j3023.1 に対しては, 分枝限定法 [33] を適用し最適な基準スケジュールを生成した. 分枝限定法による基準スケジュールのメイクスパン $MS(\pi)$ は以下の通りである.

$$\begin{aligned} j3014.1 : MS(\pi) &= 50 \\ j3015.1 : MS(\pi) &= 46 \\ j3022.1 : MS(\pi) &= 42 \\ j3023.1 : MS(\pi) &= 63 \end{aligned}$$

次に, 作業数が60である j6014.1, j6015.1, j6022.1, j6023.1 に対しては, タブー探索 [34] により基準スケジュールを生成した. 探索前の初期スケジュールはランダムに生成し, 資源制約と作業間の先行関係を満たす範囲で作業を交換することにより近傍スケジュールを生成し, スケジュールを更新していく. タブーリストは直近10解の入れ替え履歴とし, 探索回数の上限を1000回とした. なお, 初期スケジュールは任意に10個生成し, それぞれについてタブー探索を適用し最良のメイクスパンを持つスケジュールを基準スケジュールとし採用する. タブー探索による基準スケジュールのメイクスパン $MS(\pi)$ は以下の通りである.

$$\begin{aligned} j6014.1 : MS(\pi) &= 61 \quad (\text{Best Solution Value: } 61, \text{ Lower Bound: } 61) \\ j6015.1 : MS(\pi) &= 84 \quad (\text{Best Solution Value: } 84, \text{ Lower Bound: } 84) \\ j6022.1 : MS(\pi) &= 64 \quad (\text{Best Solution Value: } 64, \text{ Lower Bound: } 64) \\ j6023.1 : MS(\pi) &= 75 \quad (\text{Best Solution Value: } 75, \text{ Lower Bound: } 75) \end{aligned}$$

プロジェクトの遅延モデル

不確実的事象として, 作業の処理時間がランダムに延長することにより, プロジェクトに遅延が生じる状況(メイクスパンが直接的に増加し, クリティカルチェーンが動的に変化

する)を考える。通常作業について見ると、その時間延長は必ずしもメイクスパンの増加を招くとは限らない。ところが、遅延量の大きさによってクリティカル・チェーンが変化することがある。この場合、リアクティブな意思決定戦略を含めた複雑な状況を考えなければならぬ。

プロジェクトを構成する作業の処理時間の延長については、以下に述べるポアソン-指数モデルにしたがって実現することとする。ポアソン-指数モデルでは、プロジェクトの実行を阻害する不確実的事象(遅延事象)は、単位時間あたり平均 λ のポアソン分布にしたがって発生することとし、生じた遅延事象の処理に要する時間は平均 μ の指数分布にしたがうと仮定する。本章では、プロジェクトの遅延のシナリオを (λ, μ) で表すこととし、

$$(\lambda, \mu) = (0.3, 2), (0.3, 3), (0.4, 2), (0.4, 3)$$

の四種類のシナリオを取り上げる。

3.3.2 スケジュールの修正のタイミング

処理時間の縮約とそれの実施に必要な追加コストのバランスを考慮するために、以下の3種類の手法を用いてスケジュール修正を実施する。

▷ 事象駆動型 (EDS; Event-Driven Scheduling)

外乱が生じた作業が完了した時刻において $MS(\pi) \leq DL$ を確認する。 $MS(\pi) \geq DL$ ならば、3.2.1 で記述した作業集合に対して提案手法を実施する。事象駆動型の方策では、過度にスケジュール修正を実施してしまう状況に陥り易いことに注意する。

▷ 定期点検型 (RS; Rolling Scheduling)

基準スケジュール生成後、監視時刻を決定する。監視時刻においてスケジュール修正が必要であれば事象駆動型と同様に対象となる作業を抽出し提案手法を実施する。ただし、プロジェクト実施中に遅延事象の影響で期日を逸脱しても監視時刻に到達しない限りスケジュール修正は実施しない。

▷ 余裕時間型 (IFDS; Interfering Float-Driven Scheduling)

基準スケジュール生成後、通常作業の余裕時間を算出する。外乱が生じた作業が完了した時刻において通常作業の余裕時間を算出する。このとき、通常作業の一つでも

Table 3.1: 各点検区間の結果

	$MS(\bar{\pi})$	Freq.	TC
$PI = 3$	56	5	19.3
$PI = 6$	57	5	19.1
$PI = 12$	63	4	13.7

余裕時間消費している場合に限り提案手法を実施する。ただし、 $MS(\pi) \leq DL$ の時はその限りではない。

予備実験

ここで、定期点検型の点検間隔を決定する目的として予備実験を実施する。ここでは、 $PI = 3, 6, 12$ の3種類の点検間隔を考え、実験によりどの点検間隔が適切かを調べる。そして、プロジェクト実施状況を把握する上で有用となりうるそれを決定する。評価指標は、プロジェクト終了時のメイクスパン ($MS(\bar{\pi})$)、スケジュール修正回数 (Freq.) および追加コスト和 (TC) とする。対象としたプロジェクトは、PSPLIB のベンチマーク問題例から j3016_1 ($N = 30, R = 4$) を取り上げ分枝限定法を用い基準スケジュールを生成した。このときメイクスパンは、 $MS(\pi) = 51$ となる。プロジェクト期日は、 $DL = 57$ とした。作業遅延のシナリオは、 $(\lambda, \mu) = (0.3, 2)$ を用いた。この状況を簡単に表すと、次のようになる：メイクスパン $MS(\pi) = 51$ [日] の基準スケジュールに対して、プロジェクト期日 $DL = 57$ [日] のとき、あるプロジェクトの実施過程において、約3日間に一回のペースで半日程度の遅延が予想される状況。

Table 3.1 に予備実験の結果を、点検間隔の違いによるコスト増加の振る舞いを Fig. 3.3.2 に示す。グラフ中の実線は、プロジェクト期日を示す。プロジェクト期日の遵守に着目すると、 $PI = 3, 6$ のときは、期日を遵守できるが、 $PI = 12$ は遵守できない結果となった。次に、スケジュール修正回数の違いは、 $PI = 3, 6$ の場合は5回、 $PI = 12$ では4回となった。 $PI = 3$ と $PI = 6$ の追加コスト和を比較すると、 $PI = 6$ の追加コスト和が $PI = 3$ のそれと比べ減少していることが分かる。過度のスケジュール修正を防ぐことを目的として、比較実験では $PI = 6$ を用いる。

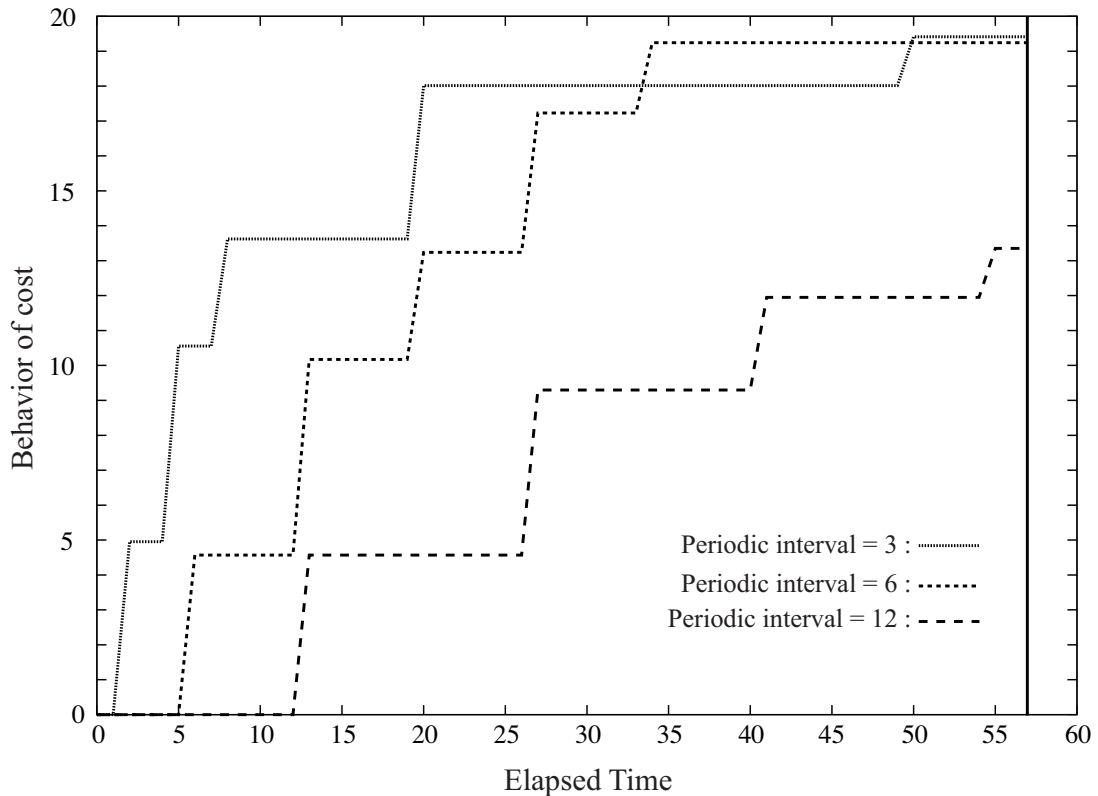


Fig. 3.4: 点検間隔の違いによるコスト増加 (j3016_1)

3.3.3 比較実験

提案手法の特性を検証することを目的として、3種類のスケジュール修正のタイミングを用い比較実験を行う。各シナリオに対して、プロジェクト遅延のシミュレーションを50回づつ実行した。スケジュール修正時刻において、提案手法実施後のメイクスパン $MS(\bar{\pi})$ 、スケジュール修正回数 (Freq.) および追加コスト和 (TC) の平均を Table3.2(a), (b), (c), (d) に作業数30の問題例である j3014_1, j3015_1, j3022_1, j3023_1 を、Table3.3(a), (b), (c), (d) に作業数60の問題例である j6014_1, j6014_1, j6022_1, j6023_1 をまとめておく。なお、各シナリオでの最良値を太字で表し、各セルの括弧内の値はプロジェクト期日 (式 (3.7)) を満足することができずシミュレーションが終了したときの $MS(\bar{\pi})$ を表す。

評価指標

まず、遅延事象の発生頻度が低い状況に着目する。作業数に関わらず、スケジュール修正回数では定期点検型が、期日の遵守および追加コスト和に関しては事象駆動型が良好な結果

Table 3.2: 実験結果 ($N = 30, R = 4$)(a) j3014_1, $MS(\pi) = 50, DL=55$

(λ, μ)	Event-Driven			Rolling			Interfering Float-Driven		
	$MS(\bar{\pi})$	Freq.	TC	$MS(\bar{\pi})$	Freq.	TC	$MS(\bar{\pi})$	Freq.	TC
(0.3,2)	55	13	36.5	55	2	38.8	55	3	37.4
(0.3,3)	55	0	0	55	0	0	55	0	0
(0.4,2)	(56)	27	72.6	55	3	93.5	55	13	86.2
(0.4,3)	(56)	3	12.1	55	1	29.2	55	1	29.3

(b) j3015_1, $MS(\pi) = 46, DL=50$

(λ, μ)	Event-Driven			Rolling			Interfering Float-Driven		
	$MS(\bar{\pi})$	Freq.	TC	$MS(\bar{\pi})$	Freq.	TC	$MS(\bar{\pi})$	Freq.	TC
(0.3,2)	50	22	65.6	50	4	101.5	50	3	119.7
(0.3,3)	50	1	3.6	49	1	15.1	50	1	11.7
(0.4,2)	(52)	34	83.8	(51)	9	98.8	50	11	94.4
(0.4,3)	(53)	13	14.5	(52)	7	40.3	(52)	6	37.2

(c) j3022_1, $MS(\pi) = 42, DL=50$

(λ, μ)	Event-Driven			Rolling			Interfering Float-Driven		
	$MS(\bar{\pi})$	Freq.	TC	$MS(\bar{\pi})$	Freq.	TC	$MS(\bar{\pi})$	Freq.	TC
(0.3,2)	48	9	11.1	48	4	16.9	48	9	11.4
(0.3,3)	(52)	14	19.8	(54)	2	25.5	59	13	27.7
(0.4,2)	48	37	58.6	48	7	64.2	49	24	64.4
(0.4,3)	47	15	13.9	46	2	29.2	48	8	21.7

(d) j3023_2, $MS(\pi) = 63, DL=75$

(λ, μ)	Event-Driven			Rolling			Interfering Float-Driven		
	$MS(\bar{\pi})$	Freq.	TC	$MS(\bar{\pi})$	Freq.	TC	$MS(\bar{\pi})$	Freq.	TC
(0.3,2)	75	14	28	71	4	11.4	73	13	25.2
(0.3,3)	70	2	0.8	67	1	7.7	67	2	2.7
(0.4,2)	71	18	11.5	69	2	28.0	71	14	22.9
(0.4,3)	68	2	0.5	68	1	4.9	69	1	2.0

Table 3.3: 実験結果 ($N = 60, R = 4$)(a) j6014_1, $MS(\pi) = 61, DL=67$

(λ, μ)	Event-Driven			Rolling			Interfering Float-Driven		
	$MS(\bar{\pi})$	Freq.	TC	$MS(\bar{\pi})$	Freq.	TC	$MS(\bar{\pi})$	Freq.	TC
(0.3,2)	(73)	101	184.0	(69)	14	255	(70)	12	233.1
(0.3,3)	68	29	39.3	67	3	75.6	67	3	82.5
(0.4,2)	(71)	149	322.1	(71)	13	365.4	(71)	11	362.3
(0.4,3)	(68)	38	97.4	(67)	4	123.6	(67)	17	119.6

(b) j6023_1, $MS(\pi) = 84, DL=92$

(λ, μ)	Event-Driven			Rolling			Interfering Float-Driven		
	$MS(\bar{\pi})$	Freq.	TC	$MS(\bar{\pi})$	Freq.	TC	$MS(\bar{\pi})$	Freq.	TC
(0.3,2)	(93)	77	138.1	92	11	155.1	92	14	154.2
(0.3,3)	92	27	52.3	90	3	80.6	92	9	62.6
(0.4,2)	(93)	68	119.5	91	4	149.4	92	44	135.1
(0.4,3)	92	14	14.9	90	1	31.4	92	2	28.4

(c) j6022_1, $MS(\pi) = 64, DL=76$

(λ, μ)	Event-Driven			Rolling			Interfering Float-Driven		
	$MS(\bar{\pi})$	Freq.	TC	$MS(\bar{\pi})$	Freq.	TC	$MS(\bar{\pi})$	Freq.	TC
(0.3,2)	72	33	22.7	72	8	39.5	72	35	30.1
(0.3,3)	71	5	4.8	69	2	12.3	68	2	11.6
(0.4,2)	76	79	67.9	74	10	92.9	73	62	62.3
(0.4,3)	71	37	31.8	79	6	66.1	71	21	49.2

(d) j6023_1, $MS(\pi) = 75, DL=84$

(λ, μ)	Event-Driven			Rolling			Interfering Float-Driven		
	$MS(\bar{\pi})$	Freq.	TC	$MS(\bar{\pi})$	Freq.	TC	$MS(\bar{\pi})$	Freq.	TC
(0.3,2)	84	22	49.8	84	4	32.6	84	24	51.8
(0.3,3)	82	19	13.5	82	3	27.2	82	17	14.4
(0.4,2)	(88)	66	74.2	(86)	8	81.3	84	41	80.7
(0.4,3)	84	27	18.2	84	5	47.2	84	18	39.5

を得た。遅延事象の処理に多くの時間を必要としない場合においては、全ての評価指標においてRSが有効であることが分かる。また、過度のスケジュール修正を必要とせずプロジェクト期日を遵守できることも確認した。しかしながら、遅延事象の処理に多くの時間を要する場合においては、事象駆動型、定期点検型ともに期日を遵守できない状況が起こる。

次に、遅延事象の発生頻度が高い状況について考察する。余裕時間型は、追加コスト和を抑えつつ期日の遵守が可能であることが分かる。スケジュール修正回数に着目すると、定期点検型と比べスケジュール修正回数は多い。スケジュール修正を過度に実施することは、後続作業の開始時刻がを頻繁に変更する可能性があり、現場に混乱をもたらすことも考慮せねばならない。定期点検型においては、遅延事象の影響を放置している状況が多いためメイクスパン低減の効果は低いことを確認した。また、一回にかかるスケジュール修正の縮約に必要な追加コストも事象駆動型および余裕時間型に比べ高いことも分かる。メイクスパンとスケジュール修正回数の関係を見ると余裕のないスケジュールを生成している可能性も否定できず、新規作業の割り込みなどの問題に対応できない状況が発生することも考えられる。

作業数

作業数の違いによる提案手法の有用性を検討する。比較対象として問題例 j3023_1 と j6023_1 を取り上げる。

まず、遅延事象の発生頻度が低い場合 (Table3.2(0.3,2)) は、定期点検型を用いたスケジュール修正の実施が有効であることがわかる。定期点検型は、事象駆動型でスケジュール修正を実施した場合より、全ての評価指標において良好な結果となった。作業数に関係なくスケジュール修正時刻以降の作業は、遅延事象の影響を排除することができプロジェクトは円滑に実施されていることがわかる。遅延事象の処理に時間を要さない場合 (Table3.2(0.3,3)) については、追加コスト和は事象駆動型が、スケジュール修正回数は定期点検型がよい結果となった。定期点検型は、遅延事象の影響を放置している状況が存在するため、一度に多くのカットに対して縮約を実施しその結果追加コストの不必要な増加を招いたことが原因と考えられる。

次に、遅延事象の発生頻度が高い場合 (Table3.3(0.4,2)) は、ともに追加コスト和は余裕時間型、スケジュール修正回数は定期点検型のほうが良い結果となった。事象駆動型における追加コスト和とスケジュール修正回数の関係に着目すると、追加コスト和の値は低いのに

対し、スケジュール修正回数の頻度はかなり高い。これは、常にスケジュール修正を実施している状況に陥っているためと考えられる。

3.3.4 考察

提案手法の特性を考察する。作業数に関係なく遅延事象の発生頻度低い場合は、定期点検型を用いた処理時間の縮約を実施が有効である。次に、遅延事象の発生頻度は高いが、処理に要する時間が短い場合に限り事象駆動型のほう定期点検型、余裕時間型と比べ有効であるとわかる。ただし、品質を重視したスケジュールの生成は、スケジュール修正回数の増加を招くことに注意する。しかしながら、遅延事象の発生がプロジェクト後期段階に発生する場合は、その影響の大きさを考慮しなければならない。遅延事象の影響が少ない場合の対処は可能であるが、遅延事象の影響が大きい場合は、その対処が困難となる。比較実験により、実行不可能なスケジュール (Table3.3 (0.4,2)) を生成することもわかった。このような状況においては、作業時間を縮約するよりも、作業の分割を実施したり作業の処理を並列に実施する方策が有効となる場合がある。

3.4 結言

本章では、プロジェクト管理におけるリアクティブ・スケジューリングを用いた意思決定法の確立の一環として、Critical Path Method (CPM) を拡張したスケジュール修正手法を提案した。数値実験をとおり、遅延事象の発生頻度が少ない場合において提案手法を定期点検型で実施することによりプロジェクト期日を遵守できることがわかった。この状況においては、品質のよいスケジュールを生成することも確認した。遅延事象の発生頻度が高い場合は、スケジュール修正回数は増加するが事象駆動型を用いることによりプロジェクト期日の遵守できる。作業数が異なる状況においても、提案手法は作業数の違いに依存しないこともわかった。

第4章 優先ルールと処理時間の縮約に基づく動的スケジューリング

前章では、動的スケジューリングにおけるスケジュール修正として、CPMに基づく処理時間の縮約を導入した方策を提案しその有用性を示した。本章では、処理時間の縮約を実施する前処理として、優先ルールを用いたスケジュール修正を取り入れた方策を提案する。この方策は、まずスケジューリング対象作業に対し優先ルールを適用した作業リストを生成し、スケジュール修正を行う。次に、期日の違反を確認した場合において作業リストの順に処理時間の縮約を実施しプロジェクト期日を遵守する。優先ルールに基づくスケジュール修正を導入することにより、縮約にかかるコストの低減が期待できる。

提案手法のスケジューリング回数への依存を確認するため、スケジューリングにおけるタイミング方策は、定期型および事象駆動型の2種類を用い予備実験を行う。次に、優先ルールと縮約効果を特性を調べるために各優先ルールを用いた比較実験をおこない提案手法の有用性を検討する。

4.1 背景

不確実的事象の影響を回避し、プロジェクトを期日通りに完了させるには、作業をできるだけ早く連続して処理しなければならない。しかしながら、不確実的事象の影響により計画スケジュール通りに作業を処理することは困難となる。この状況の解決方策は、2.5に述べた通りであるが再スケジューリング方策またはスケジュール修正方策の適用が挙げられる。それらの方策を適用することにより、プロジェクト期日の遵守の保証が確認できた場合はスケジュールを更新しそれに従い処理を実施する。プロジェクト期日の遵守を保証できない場合は、資源の追加投入を含めた新たな対策が必要となる。

本章では、スケジュール修正方策と作業時間の縮約の両者を組み合わせた、プロジェクト期日の遵守を考える。提案手法では、スケジューリング対象作業に対し優先規則に従ったス

スケジュールの一部変更を実施する。また、プロジェクト期日を遵守していなければ、対象となる作業に対し処理時間の縮約を実施しプロジェクト期日を遵守する。

4.2 優先ルールに基づくリスト生成

RCPSPはNP困難な問題に属することから、非常に複雑な問題である。RCPSPにおける計算の複雑性を考慮すると、ヒューリスティックな解法を用いることが高速かつ効果的であると考えられ、これまで様々なアルゴリズムが提案されている [35, 36, 37, 38]。

ヒューリスティックな解法は大きく、構成型解法と改善型解法に分類することができる。構成型解法は、スケジュール生成時点において決められた順序に従い作業をスケジュールに加えることにより、実行可能解を得る方法である。RCPSPにおける構成型解法も同様の手続きで実行可能解を得るが、先行制約も考慮しなければならない。対象となる作業が全くスケジューリングされていない状態から作業の割当を開始する。その後、各ステップごとスケジューリング可能な作業の部分集合を生成し、これらの部分集合から順次作業を選択し開始時刻を与える。スケジューリング対象となる作業全ての開始時刻が決定するまでこのステップを繰り返す。各ステップにおいて作業に開始時刻を割り当てる際には、優先ルールに基づいて各作業に順序付けを行い、この優先順位に従って開始時間を割り当てる。

次に、改善型解法は構成型解法により生成されたスケジュールを改善することにより解を得る方法である。各ステップごとに、現在得られている最良解を改善し、最終的に最適解または近似解を取得することが目的とされる。最良解を改善する際には、しばしばタブーサーチ法 [39, 40]、シミュレイトッドアニーリング法 [41, 42]、遺伝的アルゴリズム [43] などのメタヒューリスティック解法が用いられる。また、シミュレイトッドアニーリング法と遺伝的アルゴリズムを組み合わせた解法も提案されている [44]。

提案手法では、構成型解法は既存の優先ルールを適用しスケジュールを生成する。改善型解法にて作業時間の縮約を実施することによりプロジェクト・スケジュールを更新する。

4.2.1 優先ルール

改善型解法において改善操作を実施する際、リスト上のどの作業に操作を行うかを決定することが重要となる。この決定基準としてしばしば用いられる基準が優先ルール [45] である。優先ルールを用いると、各ルールの判断基準に基づいてリスト上の各作業 $i \in \mathcal{A}$ の優

先値を算出し、優先値が最大または最小となる作業を選択する。もし、優先値が同値の作業が複数存在する場合は、その中から更に一つの作業を選択する為に新たな優先ルールを適用する。これまで、RCPSP に対する種々の優先ルールが提案されている。

優先ルールは生成されるタイミングによって分類される。静的な優先ルールにおいて優先値は、スケジューリングを開始する前に一度だけ計算されるだけだが、動的な優先ルールでは優先値は作業がスケジューリングされるたびに計算し直される。また、優先ルールは使用される情報によって大きく4つに分類される。

作業時間に基づく優先ルール

一番目のグループは、処理時間に基づく優先ルールを示す。各作業が持つ処理時間のみの情報を利用する。

ネットワークに基づく優先ルール

2番目のグループはネットワークに基づく優先ルールを示す。プロジェクト・ネットワーク (AoA ネットワーク, AoN ネットワークを指す。) に関する情報のみを利用して生成される。例えば、仕事の実行期間や後続作業数などが情報として用いられる。その際、資源に関する情報は一切考慮しない。

クリティカル・チェーンに基づく優先ルール

クリティカル・チェーンに基づく優先ルールは、各作業の最早開始時刻や最遅開始時刻により生成される。3番目のグループは、静的な優先ルールを示す。動的に扱う場合は、不確的事象後の実際のスケジュールに基づき作成する。

資源に基づく優先ルール

資源に基づく優先ルールは各作業の使用資源量に基づいて作成される。例えば、各作業が使用する全種類の資源の平均使用量を基に優先ルールが作成される。4番目のグループは、資源に基づく優先ルールを示す。

Table 4.1: 優先ルールの一例

作業時間に基づく優先ルール		
ルール	判断基準	優先値
<i>SPT</i> (smallest processing time)	min	p_i : 処理時間の昇順
<i>LPT</i> (largest processing time)	max	p_i : 処理時間の降順
ネットワークに基づく優先ルール		
ルール	判断基準	優先値
<i>LIS</i> (least immediate successor)	min	Q_i : 後続作業数の昇順
<i>MIS</i> (most immediate successor)	max	Q_i : 後続作業数の降順
<i>LTS</i> (least total successor)	min	$\sum_{i \in \mathcal{A}} Q_i$: 総後続作業数の昇順
<i>MTS</i> (most total successor)	max	$\sum_{i \in \mathcal{A}} Q_i$: 総後続作業数の降順
<i>GRPW</i> (greatest rank positional weight)	max	$p_h + \sum_{i \in \mathcal{Q}_i} p_i$: 後続作業の総処理時間の降順
資源に基づく優先ルール		
ルール	判断基準	優先値
<i>EST</i> (earliest starting time)	min	ES_i : 最早開始時刻の昇順
<i>ECT</i> (earliest completion time)	min	LS_i : 最早完了時刻の昇順
<i>LST</i> (latest starting time)	min	ES_i : 最遅開始時刻の降順
<i>LCT</i> (latest completion time)	min	LS_i : 最遅完了時刻の降順
<i>MSLK</i> (minimum slack)	min	$LS_i - ES_i$: 最遅開始時刻および 最早開始時刻の差の昇順
<i>GRR</i> (greatest resource requirements)	max	$p_i \sum_{l \in \mathcal{A}} r_{il} (l = 1, 2, \dots, R)$: 資源量と処理時間の積の降順

4.2.2 リスト生成の対象となる作業の抽出

時刻 t' において作業 i' に不確実的事象が発生したとする。基準スケジュールにおいて、作業 i' の完了時刻は、 $f_i(\pi) = s_i(\pi) + p_i$ から $f_i(\tilde{\pi}) = f_i(\pi) + \delta$ と変化する。右シフト操作実施後、時刻 $t' (< t)$ において実際のメイクスパン $MS(\tilde{\pi})$ がプロジェクト期日 DL を超えた ($MS(\tilde{\pi}) > DL$) 場合に限り作業集合 \mathcal{A}'_t に対して作業の抽出を実施する。抽出された作業集合を $\mathcal{L}'_t (\mathcal{L}'_t \subseteq \mathcal{A}'_t)$ と定義する。

縮約対象作業の抽出の例として、Fig. 4.1 を用いる。基準スケジュールのメイクスパン $MS(\pi) = 21$ 、資源の種類は一種類 ($R = 1$) である。いま、時刻 $t' (0 < t' < 3)$ において、作業 1, 2 に遅延が発生したとする。このとき、時刻 5 において右シフト操作を適用すると、実際のメイクスパン $MS(\tilde{\pi}) = 25$ となる (Fig. 4.2)。作業 2 ($i = 1$) の完了時刻は

$t = 2(f_2(\pi) = f_2^l(\pi) = 2)$ であるが、不確実的事象の影響で時刻 $t = 7$ となる。この結果、後続作業 3 ($Q_2 = \{3\}$) は当初の予定の開始時刻に処理を開始することができない。 Q_2 に続く作業 4,5,6 においても同様である。スケジューリング時刻 $t = 5$ 以降の未処理の作業集合 $\mathcal{L}_5^{(2)}(i = 2, t = 5)$ には、作業 3,4,5,6,7,9,10 が属することになる。

次に、未処理の作業集合 $\mathcal{L}_5^{(2)} = \{3, 4, 5, 6, 7, 9, 10\}$ に対して優先ルールを適用しリストを生成する。例として *GRR* ルールを適用する。*GRR* ルールは、

$$p_i \sum_{i \in \mathcal{A}} r_{il}, \quad (l = 1, 2, \dots, R). \quad (4.1)$$

であった。しかしながら、優先ルールを適用するのは未処理の作業集合のみであるため、上式は、

$$p_i \sum_{i \in \mathcal{L}_t^{(i)}} r_{il}, \quad (l = 1, 2, \dots, R). \quad (4.2)$$

となる。未処理の作業集合 $\mathcal{L}_5^{(2)} = \{3, 4, 5, 6, 7, 9, 10\}$ の優先値は、それぞれ作業 3=12, 作業 4=3, 作業 5=2, 作業 6=12, 作業 7=10, 作業 9=6, 作業 10=2 となり、*GRR* ルール適用後のリストは、 $GRR = \{3, 6, 7, 9, 4, 5, 10\}$ となる。

4.3 スケジュール修正と処理時間の縮約の手順

抽出した作業を基にリスト生成後、スケジュール修正を実施する。リストは優先ルールに従い生成されるが、本節では、4.2.2において生成したリストを用いスケジュール修正の手順を説明する。また、生成後の実際のスケジュールがプロジェクト期日を違反した場合においては、処理時間の縮約を適用することによりプロジェクト期日を遵守する。縮約の手順は、前章と同様となるが、スケジュール修正の手順を加えた手順の説明を加える。

4.3.1 スケジュール修正

スケジュール修正は、不確実的事象の影響でプロジェクト期日を逸脱した場合 ($MS(\bar{\pi}) > DL$) に限り実施が可能となる。4.2.2において、優先ルールを使用したリストの生成をもとに、スケジュール修正を実施する。このとき、作業間における先行制約を、

$$s_i + p_i \leq s_j, \quad i \in \mathcal{L}_t^{(i)}, i \neq j, j \in Q_i, \quad (4.3)$$

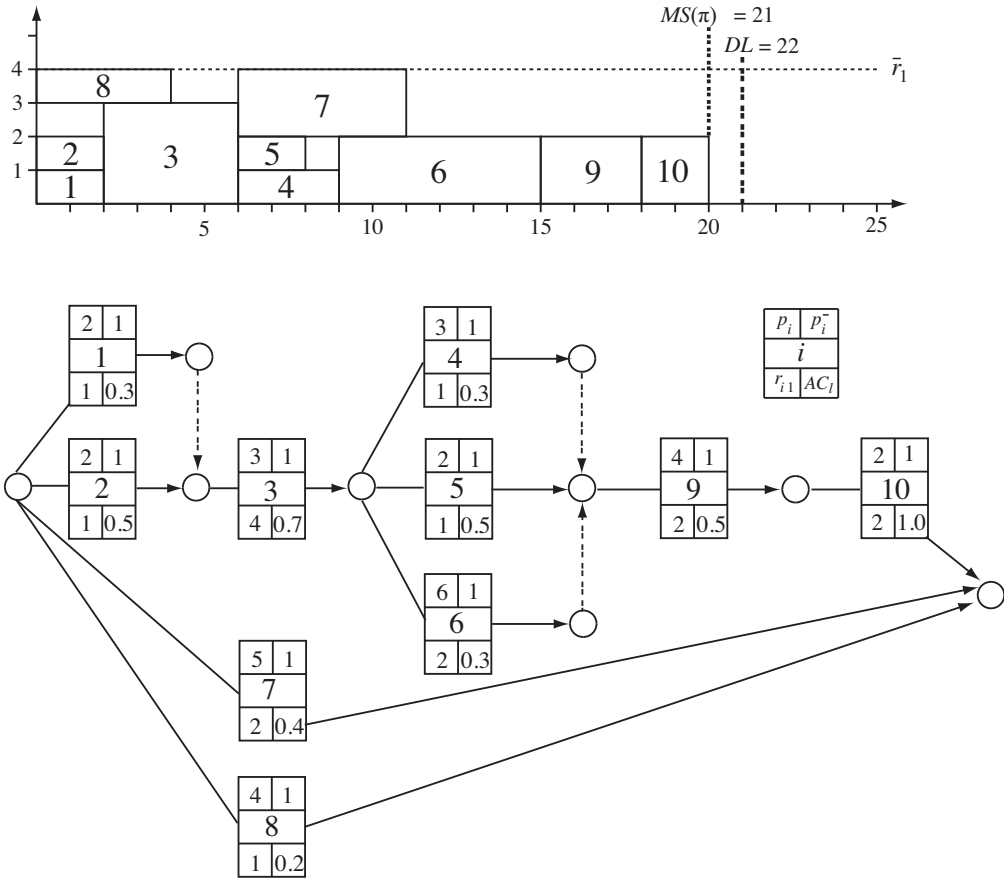


Fig. 4.1: 基準スケジュール

とし、資源制約を

$$\sum_{i \in \mathcal{L}_t^{(i)}} r_{il} \leq \tilde{r}_{il}, \quad l = 1, \dots, R, t' = t + 1, \dots, DT, \tag{4.4}$$

とする。スケジューリング時点において、リスト $GRR = \{3, 6, 7, 9, 4, 5, 10\}$ の作業に対しスケジューリングを実施する。遅延作業 i' の完了時刻は $f_{i'}(\pi) + \delta$ より、作業1の完了時刻は、 $t = 7$ となる。よって、作業3の開始時刻を s_3 における先行制約は、

$$f_i(\pi) + \delta \leq s_3, \tag{4.5}$$

になり、リストの先頭である作業3の先行制約を満たす開始時刻は $t = 7$ と決定される。次に、資源制約の違反を確認する。時刻 $t = 7$ の時点で、利用可能な資源量は $\tilde{r}_{i1} = 4$ である。作業3の必要資源量 $\tilde{r}_{31} = 1$ より資源制約の違反は発生しない。よって、作業3の開始時刻は、 $t = 7$ と決定される。以降のリスト内の作業も同様に先行制約および資源制約を確認し

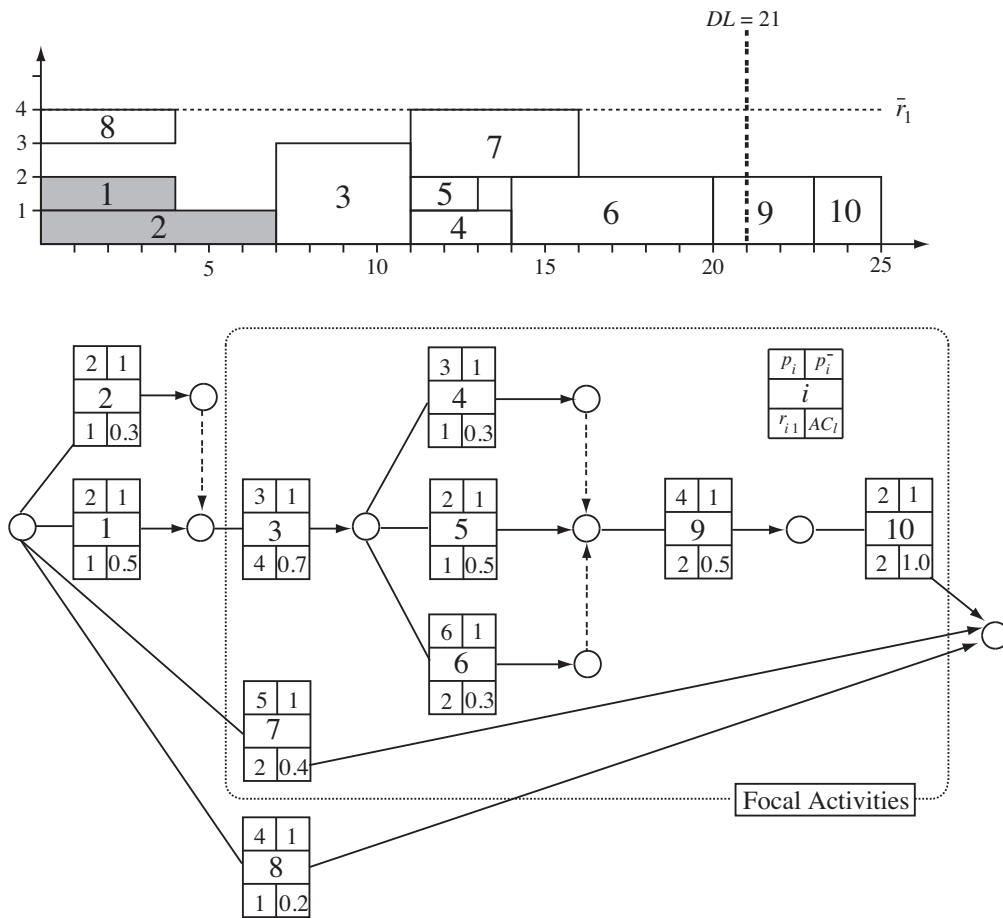


Fig. 4.2: 遅延発生後のスケジュールと抽出対象作業

スケジュールを逐次生成していく。ただし、スケジューリング対象作業が、先行制約を違反した場合に限り、直後の作業と交換することとする。また、このとき得られるメイクスパンを $MS(\pi_L)$ とする。生成後のスケジュールを Fig. 4.3 ($MS(\pi_L) = 21$) に示す。

4.3.2 コストを伴う処理時間の縮約

4.3.1 で得られたスケジュールは、 $MS(\pi_L) \geq DL$ より、プロジェクト期日を逸脱している。この場合に限り、処理時間を縮約が可能となる。3.2.1 と同様に、各作業には処理時間の下限値 \bar{p}_i が与えられその範囲内において縮約が可能となる。次に 1 単位時間縮約した処理時間を \tilde{p}_i とする。 \tilde{p}_i は、縮約されるたびに更新される。ただし、処理時間 $p_i = 1$ の作業は縮約に対象とはならない。

次に、各作業を 1 単位時間縮約するために必要な追加コストを定義する。1 単位時間の縮

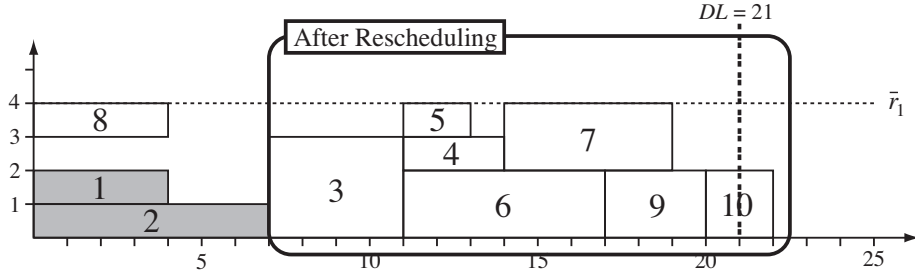


Fig. 4.3: 優先ルールを用いたスケジュール修正

約に必要な追加コストは,

$$RC_i^l = \frac{r_{il}}{p_i}, \quad l = 1, \dots, R, i \in \mathcal{L}_t^{(i)}, \quad (4.6)$$

で与える. したがって, 作業 i における処理時間 p_i から縮約後の処理時間 \tilde{p}_i で必要な総追加コスト TRC は,

$$TRC = \sum_{i \in \mathcal{L}_t^{(i)}} \sum_{l=1}^R (p_i - \tilde{p}_i) RC_i^l, \quad (4.7)$$

で与えられる. 縮約後のメイクスパンを

$$MS(\pi_{\tilde{L}}) = \sum_{j=1}^{N_c(\pi_{\tilde{p}})} \min(p_{a(j)}, \tilde{p}_{a(j)}), \quad (4.8)$$

とする. 4.8式における $\tilde{p}_{a(j)}$ は, 実際のスケジュールにおけるクリティカル作業 $a(j)$ の処理時間を表す. また, $N_c(\pi_{\tilde{L}})$ はスケジュール \tilde{L} におけるクリティカル作業を表す. スケジュール \tilde{L} のクリティカル・チェーンは, $C^{\pi_{\tilde{L}}} = \{a(1), a(2), \dots, a(N_c(\pi_{\tilde{L}}))\}$ となる. 以上より, スケジューリング時点 t において \tilde{L} を決定するため, 目的関数を総追加コストの最小化とし定式化する.

$$\text{minimize} \quad \sum_{i \in \mathcal{L}_t^{(i)}} \sum_{l=1}^R (p_i - \tilde{p}_i) RC_i^l \quad (4.9)$$

$$\text{subject to} \quad \sum_{j=1}^{N_c(\pi_{\tilde{L}})} \min\{p_{a(j)}, \tilde{p}_{a(j)}\} \leq DL, \quad (4.10)$$

$$s_i + \min\{p_i, \tilde{p}_i\} \leq s_j, \quad i \in \mathcal{L}_t^{(i)}, i \neq j, j \in Q_i, \quad (4.11)$$

$$\sum_{i \in \mathcal{A}_{t'}} r_{il} \leq \tilde{r}_l, \quad l = 1, \dots, R, t' = t+1, \dots, DT, \quad (4.12)$$

$$1 \leq p_i^- \leq \tilde{p}_i < p_i, \quad i \in \mathcal{L}_t^{(i)}. \quad (4.13)$$

式 (4.10) は、実際のメイクスパンがプロジェクト期日を超えてはならないことを示し、式 (4.11) は、作業間の先行制約を表す。次に式 (4.12) は、資源制約を表し、式 (4.13) は各作業の縮約可能範囲を表している。

4.3.3 提案手法の手順

プロジェクト・スケジュールには、複数のクリティカル・チェーンが存在している。プロジェクト管理者は、事前にクリティカル・チェーンを決定しクリティカル作業を重点的に管理している。しかしながら、不確実的事象の影響で計画段階に採用したクリティカル・チェーンが変化してしまう。このため、縮約の対象となる作業を明確にするため縮約実施後において PERT 計算を用い、新たにクリティカルチェーンを決定する必要がある。

縮約の対象となる作業集合は、 $\mathcal{L}_t^{(i)}$ であった。 $\mathcal{L}_t^{(i)}$ に含まれる作業は、 $\mathcal{C}^l = \{C_1^l, C_2^l, \dots, C_{n'}^l\}$ ($n' = |\mathcal{C}^l|$) とする。 $\mathcal{L}_t^{(i)}$ に対し、以下の手順で提案手法を実施する。

- 1° スケジューリング時点において $\mathcal{L}_t^{(i)}$ に含まれる作業を用い、部分ネットワークを生成する。
- 2° $\mathcal{L}_t^{(i)}$ に対し、優先ルールを選択しリストを生成する。
- 3° リスト内の優先値に従い、式 (4.11) および式 (4.12) を満たす範囲でスケジュールを逐次生成する。
- 4° 生成後のスケジュールが式 (4.10) を満たすならば、新たなメイクスパン $MS(\pi)$ を出力し終了する。
式 (4.10) を違反するならば、5° へ。
- 5° 部分ネットワーク $\mathcal{L}_t^{(i)}$ において列挙できるカット \mathcal{C}^l を生成する。

- i) カット \mathcal{C}^l に含まれる $C_j^l (C_j^l \subseteq \mathcal{C})$ に対し、必要となる追加コスト、

$$RC(j) = \sum_{i \in C_j^l} RC_i, \quad j = 1, \dots, n', \quad (4.14)$$

を算出する。

- ii) 最小となるカット $\mathcal{C}_{j^*}^l$ を選択する。ただし、 $RC(j^*) = \min\{RC(1), \dots, RC(n')\}$ である。

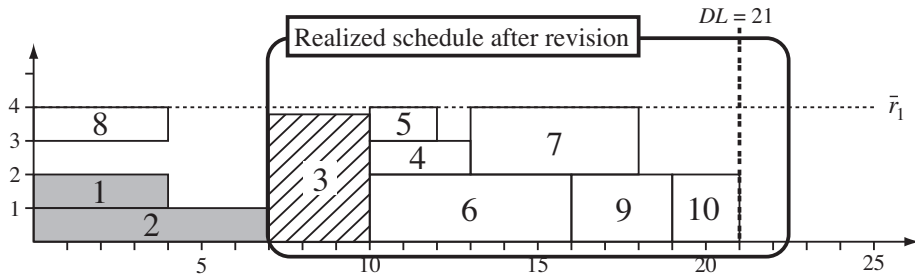


Fig. 4.4: 処理時間縮約後のスケジュール

6° $C_{j^*}^l$ に含まれる作業 i' の処理時間 $p_{i'}$ を一単位時間縮約する. この時, $p_{i'} \leftarrow p_{i'} - 1 (i' \in C_{j^*}^l)$ および $\bar{p}_{i'} \leftarrow p_{i'}$ と更新する.

7° 新たなメイクスパン $MS(\bar{\pi})$ および, 総追加コスト (TRC) を算出する.

8° $MS(\bar{\pi}) < DL$ なら 6° へ. それ以外は, 2° へ.

9° 総追加コスト TRC および新たなメイクスパン $MS(\bar{\pi})$ を出力後, 終了する.

Fig. 4.2 に対し提案手法を適用結果を Fig. 4.4 に示す. このとき, 最小追加コストのカットは $C_1^l (j^* = 3)$ であり, C_2^l に含まれる作業を一単位時間縮約することにより, $MS(\bar{\pi}) = 21$, $TRC = 0.7$ を得る.

4.4 数値実験

4.4.1 問題設定

対象問題

PSPLIB の問題例から作業数 30 である j3014.1 ($N = 30, R = 4$), j3015.1 ($N = 30, R = 4$), j3022.1 ($N = 30, R = 4$), j3023.1 ($N = 30, R = 4$), の 4 種類をとりあげ, 基準スケジュールを生成する. 得られたメイクスパン $MS(\pi)$ に対して期日を設定は, 期日 DL は $DL = 1.2MS(\pi)$ とした.

基準スケジュールの生成

すべての問題例に対し, 分枝限定法 [33] を適用し最適な基準スケジュールを生成した. 分枝限定法は, 問題をより小規模な子問題に分割する分枝操作と既知の最良の解 (暫定解) よ

りも良い解を与えない子問題を削除する限定操作等からなる最適化手法である。分枝限定法による基準スケジュールのメイクスパン $MS(\pi)$ は以下の通りである。

$$\begin{aligned} \text{j3014.1} : \quad & MS(\pi)=50 \\ \text{j3015.1} : \quad & MS(\pi)=46 \\ \text{j3022.1} : \quad & MS(\pi)=42 \\ \text{j3023.1} : \quad & MS(\pi)=63 \end{aligned}$$

プロジェクトの遅延モデル

プロジェクト実行段階において不確実的事象の影響を放置するとスケジュールの品質が著しく低下するような状況を想定する。本章での不確実的事象は、作業遅延とする。作業遅延は、ポアソン-指数モデルを用い再現する。プロジェクト期日 DL までに生起する作業遅延を計数過程とすれば、ポアソン分布を用い表現できるため作業遅延が単位時間あたり平均 λ のポアソン分布にしたがって発生すると仮定する。また、発生した遅延量は平均 μ の指数分布にしたがうとする。プロジェクトの遅延シナリオを (λ, μ) で表すこととし、

$$(\lambda, \mu) = (0.3, 2), (0.3, 3), (0.4, 2), (0.4, 3)$$

の4種類のシナリオを用いる。

優先ルール

優先ルールは使用される情報に基づき、処理時間、ネットワーク、クリティカル・チェーンおよび資源の4種類に分類できる。数値実験に用いる優先ルールは、

- ▷ ネットワーク：*LIS* ルール, *MIS* ルール
- ▷ クリティカル・チェーン：*EST* ルール, *ECT* ルール
- ▷ 資源：*GRR* ルール

の5種類を使用する。処理時間に基づくルールは、各作業の資源と処理時間に直接的な関係があると仮定し本実験では除外する。

修正のタイミング

タイミング方策を講じる場合、スケジュールの評価尺度のみならずスケジューリング回数も同様に扱わなければならない。特に、過度のスケジューリングは対象プロジェクトが常に

スケジューリングの状況下に置かれるため、現場の混乱を招く恐れがある [46]。すなわち、評価尺度が改善されようともプロジェクト管理としてのスケジュールが機能していなければスケジューリング自体が効果をなさない。

本章では、修正のタイミング方策として、定期型と事象駆動型の2種類を用いる。定期型のスケジュールの確認および監視時刻は、3.3.2より $PI = 6$ とする。

4.4.2 比較実験

提案手法の特性を検証することを目的として、2種類のスケジュール修正のタイミングを用い比較実験を行う。各シナリオに対して、プロジェクト遅延のシミュレーションを50回ずつ実行した。スケジュール修正時刻において、提案手法実施後のメイクスパン $MS(\pi_{\bar{L}})$ 、スケジュール修正回数 (Freq.) および総追加コスト (TRC) の平均を Table 4.2(a), (b), (c), (d) に作業数30の問題例である j3014_1, j3015_1, j3022_1, j3023_1 をまとめる。なお、各シナリオでの最良値を太字で表し、各セルの括弧内の値はプロジェクト期日を満足することができずシミュレーションが終了したときの $MS(\pi_{\bar{L}})$ を表す。

タイミング方策

全ての問題例において、ED 方策より RS 方策のほうがスケジュール修正回数は低い結果となった。また、遅延事象の発生頻度が低く、その処理に多くの時間を必要としない場合においては、両方策ともプロジェクト期日の遵守を違反することはない。しかしながら、プロジェクト工期段階に発生する遅延事象に対処できず、プロジェクト期日を逸脱した結果も確認した (Table 4.2a)(0.3,3)。このことから、不確実的事象の発生頻度が高い場合は、ED 方策および RS 方策の両者を組み合わせた新たなタイミング方策が必要である。まず、遅延事象の発生頻度が低い状況に着目する (Table 4.2(0.3,2)(0.3,3))。作業数に関わらず、スケジュール修正回数では RS 方策が、期日の遵守および総追加コストに関しては ED 方策が良好な結果を得た。しかしながら、このような状況では RS 方策を用いプロジェクト管理を実施した方がスケジューリング回数を抑制することができ、スケジュール変更に伴う、現場の混乱を防止することが可能となる。

次に、遅延事象の発生頻度が高く処理に時間を要す場合は (Table 4.2(0.4,2)) は、ともに総追加コストは ED 方策が、処理に時間を要さない場合は、スケジュール修正回数は RS

方策が良い結果となった。遅延事象の発生頻度が高く処理に時間を要す場合における RS 方策は、遅延事象の影響を無視している状況があると考えられる。このことから、時刻での点検に加え、作業の進捗をみる新たな指針が必要である。

優先ルール

全ての問題例において、GRR ルールを除く優先ルールで良好な解を得る結果となった。まず、LIS および MIS ルールについて考察する。両ルールとも優先値は後続作業数である。後続作業数の降順で優先値が決定される MIS ルールは、縮約を実施した作業の後続作業の開始時刻は早めることができるため縮約の効果が現れた結果となった。LIS ルールにおいては、その効果が期待できず縮約を実施する作業の増加を招く要因になる恐れがある。

次に、EST および ECT ルールは、両ルールとも良好な解を得ることができた。この結果は、初期スケジュールの立案の影響が考えられる。数値実験において初期スケジュールは、分枝限定法を用い最適スケジュールを採用した。初期スケジュールにおける各作業には、十分な余裕時間が与えられておらず EST および ECT ルールの優先値を有効に活用できなかったために類似した解を得たと考えられる。EST および ECT ルールの評価は、規模の大きい問題を用い再度確認する必要がある。

最後に GRR ルールは、その優先値の特徴から多くの追加コストを必要とする結果となった。

考察

提案手法の特性を考察する。作業数に関係なく遅延事象の発生頻度低い場合は、RS 方策を用いた処理時間の縮約を実施が有効である。次に、遅延事象の発生頻度は高いが、処理に要する時間が短い場合に限り EDS 方策のほうが RS 方策と比べ有効であるとわかる。

優先ルールについては、優先値に後続作業数を用いた LIS および MIS ルールを用い処理時間を縮約することで、作業の開始時刻の前倒しが可能となる。本提案手法をまとめると、遅延事象の発生頻度が低い場合においては、RS 方策と MIS ルールを用いたスケジューリング方策が有効であるといえる。ただし、遅延事象の発生頻度が高い場合においては、RS 方策および ED 方策のハイブリッド型を新たに導入する必要がある。

4.5 結言

本章では、まずスケジューリング問題における代表的な解法の一つであるヒューリスティックな解法を、構成型解法と改善型解法の二つに分け紹介した。構成型解法に、優先ルールを用いたスケジュール修正を説明した。優先ルールを適用する際に、スケジューリング対象となる作業の抽出方法を提案し、リストを生成する。リスト生成には、優先ルールがもつ優先値を利用するため、各ルールがもつ情報において分類整理を行った。

次に、プロジェクト期日を違反した場合におけるリスト内作業の処理時間縮約方法を提案した。優先値を用いスケジューリングされた実際のメイクスパンが期日を違反した場合に限り処理時間の縮約を実施することでプロジェクト期日の遵守を可能とし、既存のタイミング方策との関係性を明らかにすることを目的として数値実験を実施した。また、各優先ルールの特徴を調べるためにネットワークに基づくルール、クリティカル・チェーンに基づくルール、資源に基づくルールの3種類から、代表的なルールを選び、提案手法におけるそれらの特性も明らかにした。

提案手法は、遅延事象の発生頻度が少ない場合において良好なスケジュールを生成することを確認するとともに、遅延事象の処理に要する時間が長いときにおいても同様の解を得ることも明らかとなった。また、遅延事象の発生頻度、それに要する処理時間が長い場合においては新たなタイミング方策としてハイブリッド型が有用であることも確認した。提案手法とハイブリッド型を用いたタイミング方策の有用性の確認は、今後の検討課題とする。

Table 4.2: 実験結果

a) j3014.1($MS(\pi) = 50, DL = 60, R = 4$)

		(0.3,2)		(0.3,3)		(0.4,2)		(0.4,3)	
		ED	RS	ED	RS	ED	RS	ED	RS
<i>LIS</i>	$MS(\pi_{\bar{L}})$	60	60	60	60	60	60	60	60
	<i>TRC</i>	26.9	12.7	14.3	17.2	28.1	21.6	15.5	19.3
	<i>Freq.</i>	13	4	16	2	27	4	16	3
<i>MIS</i>	$MS(\pi_{\bar{L}})$	60	60	69	59	60	60	59	60
	<i>TRC</i>	12.1	13.6	16.5	18.7	22.7	29.3	16.2	17.6
	<i>Freq.</i>	9	4	14	3	22	3	13	3
<i>EST</i>	$MS(\pi_{\bar{L}})$	60	60	(61)	60	60	60	59	60
	<i>TRC</i>	12.9	17.7	11.4	9.6	22.5	25.8	13.7	14.2
	<i>Freq.</i>	11	3	14	3	24	6	15	4
<i>ECT</i>	$MS(\pi_{\bar{L}})$	60	60	(63)	60	60	60	60	60
	<i>TRC</i>	12.4	14.4	11.7	12.6	26.8	32.6	13.2	16.5
	<i>Freq.</i>	11	6	15	5	25	6	14	4
<i>GRR</i>	$MS(\pi_{\bar{L}})$	(61)	60	(63)	60	60	60	60	60
	<i>TRC</i>	26.9	27.1	14.7	16.2	30.2	31.9	15.7	17.1
	<i>Freq.</i>	11	8	10	3	24	5	10	4

b) j3015.1($MS(\pi) = 46, DL = 55, R = 4$)

		(0.3,2)		(0.3,3)		(0.4,2)		(0.4,3)	
		ED	RS	ED	RS	ED	RS	ED	RS
<i>LIS</i>	$MS(\pi_{\bar{L}})$	55	55	55	(57)	54	53	53	54
	<i>TRC</i>	14.2	15.7	14.1	17.4	24.3	25.6	16.2	28.6
	<i>Freq.</i>	9	3	14	3	27	4	16	3
<i>MIS</i>	$MS(\pi_{\bar{L}})$	55	54	55	(56)	55	55	52	54
	<i>TRC</i>	12.1	13.6	14.7	18.1	26.2	28.5	16.9	29.3
	<i>Freq.</i>	9	4	14	3	27	4	17	3
<i>EST</i>	$MS(\pi_{\bar{L}})$	54	55	(56)	55	55	55	54	55
	<i>TRC</i>	14.9	17.7	13.4	12.1	24.3	26.4	15.7	28.4
	<i>Freq.</i>	11	3	12	5	27	5	15	4
<i>ECT</i>	$MS(\pi_{\bar{L}})$	53	55	55	55	55	55	55	55
	<i>TRC</i>	12.4	14.1	13.7	14.6	24.9	23.7	14.1	24.2
	<i>Freq.</i>	10	3	12	5	27	5	15	4
<i>GRR</i>	$MS(\pi_{\bar{L}})$	55	55	(58)	(60)	55	55	(56)	55
	<i>TRC</i>	12.8	16.1	11.5	9.2	23.6	28.1	16.5	24.2
	<i>Freq.</i>	10	4	10	3	26	4	12	4

c) j3022.1($MS(\pi) = 42, DL = 50, R = 4$)

		(0.3,2)		(0.3,3)		(0.4,2)		(0.4,3)	
		ED	RS	ED	RS	ED	RS	ED	RS
<i>LIS</i>	$MS(\pi_{\bar{L}})$	49	50	50	50	50	49	48	49
	<i>TRC</i>	13.2	12.7	15.3	16.1	23.8	24.8	14.9	29.8
	<i>Freq.</i>	10	4	14	2	28	5	16	3
<i>MIS</i>	$MS(\pi_{\bar{L}})$	49	49	50	50	50	50	47	49
	<i>TRC</i>	12.1	13.6	15.1	17.2	24.6	23.5	15.4	27.1
	<i>Freq.</i>	9	4	14	3	28	4	17	3
<i>EST</i>	$MS(\pi_{\bar{L}})$	48	50	50	50	50	50	49	50
	<i>TRC</i>	14.9	17.7	13.8	12.4	23.5	25.8	12.7	13.7
	<i>Freq.</i>	11	3	12	3	27	6	15	5
<i>ECT</i>	$MS(\pi_{\bar{L}})$	48	50	50	(51)	50	50	50	50
	<i>TRC</i>	12.4	14.1	11.7	9.5	24.9	23.6	12.3	15.1
	<i>Freq.</i>	11	6	11	5	28	6	15	5
<i>GRR</i>	$MS(\pi_{\bar{L}})$	50	50	50	50	50	50	(51)	50
	<i>TRC</i>	13.5	17.4	13.5	11.2	24.2	22.1	9.3	10.6
	<i>Freq.</i>	11	8	11	4	27	5	10	5

d) j3023.1($MS(\pi) = 63, DL = 75, R = 4$)

		(0.3,2)		(0.3,3)		(0.4,2)		(0.4,3)	
		ED	RS	ED	RS	ED	RS	ED	RS
<i>LIS</i>	$MS(\pi_{\bar{L}})$	75	75	74	75	75	75	75	75
	<i>TRC</i>	22.1	24.7	1.1	0.5	10.2	9.3	0.3	0.8
	<i>Freq.</i>	10	5	2	2	14	3	1	1
<i>MIS</i>	$MS(\pi_{\bar{L}})$	75	75	73	75	75	75	73	75
	<i>TRC</i>	21.8	28.6	0.5	1.3	9.5	8.5	0.7	0.2
	<i>Freq.</i>	12	4	2	1	15	3	1	1
<i>EST</i>	$MS(\pi_{\bar{L}})$	75	75	74	75	75	75	74	74
	<i>TRC</i>	20.3	21.6	0.8	1.2	8.1	10.7	0.8	0.8
	<i>Freq.</i>	13	4	2	1	14	7	1	1
<i>ECT</i>	$MS(\pi_{\bar{L}})$	75	75	(76)	75	75	75	75	75
	<i>TRC</i>	22.3	22.8	0	0.7	8.6	9.2	0.5	0.5
	<i>Freq.</i>	12	5	0	1	14	6	1	1
<i>GRR</i>	$MS(\pi_{\bar{L}})$	75	75	74	75	75	75	75	(76)
	<i>TRC</i>	22.7	24.4	0.7	0.9	10.2	12.4	1.2	0
	<i>Freq.</i>	15	5	1	2	16	7	1	2

第5章 結論

本研究では、効果的かつ一定の有用性を有する動的スケジューリング方式の実現を目指し、

▷ スケジュールの生成方法：

- 1) Critical Path Method を用いた動的スケジューリング
- 2) 優先ルールとを用いたスケジュール修正と処理時間の縮約に基づく動的スケジューリング

▷ タイミング方策：

- 1) 動的スケジューリングにおけるスケジュール修正実施の判断のための枠組みおよび方策の提案

について述べた。

第1章では、本研究の背景を整理するとともに、本研究の目的を述べた。従来研究と本研究の違いを述べ、従来研究が現実的な問題への応用の報告が少ないことに触れ、動的スケジューリングの重要性を述べた。

第2章にプロジェクト管理からみたスケジューリング問題を定義し、対象とする RCPSP を紹介し問題を定式化を行った。次に、動的スケジューリングが対象とする問題を明確にするとともに、従来研究の整理および紹介を行った。RCPSP の解の生成の側面から、スケジュール生成のタイミングを整理し、動的スケジューリングを定義するとともに従来研究とともに整理しプロジェクトにおける管理技法を紹介した。

第3章では、動的スケジューリングに Critical Path Method を取り入れたスケジュール修正方策を提案しスケジュール修正に有効なタイミング方策を明らかにした。また、プロジェクト管理におけるリアクティブ・スケジューリングを用いた意思決定法の確立の一環として、Critical Path Method (CPM) を拡張したスケジュール修正手法を提案した。数値実験をとおし、遅延事象がの発生頻度が少ない場合において提案手法を RS で実施することによりプロジェクト期日を遵守できることがわかった。この状況においては、品質のよいスケジュー

ルを生成することも確認した。遅延事象の発生頻度が高い場合は、スケジュール修正回数は増加するがEDSを用いることによりプロジェクト期日の遵守できる。作業数が異なる状況においても、提案手法は作業数の違いに依存しないこともわかった。

第4章は、まずスケジューリング問題における代表的な解法の一つであるヒューリスティックな解法を、構成型解法と改善型解法の二つに分け紹介した。構成型解法に、優先ルールを用いたスケジュール修正を説明した。優先ルールを適用する際に、スケジューリング対象となる作業の抽出方法を提案し、リストを生成する。リスト生成には、優先ルールがもつ優先値を利用するため、各ルールがもつ情報において分類整理を行った。

次に、プロジェクト期日を違反した場合におけるリスト内作業の処理時間縮約方法を提案した。優先値を用いスケジューリングされた実際のメイクスパンが期日を違反した場合に限り処理時間の縮約を実施することでプロジェクト期日の遵守を可能とし、既存のタイミング方策との関係性を明らかにすることを目的として数値実験を実施した。また、各優先ルールの特徴を調べるためにネットワークに基づくルール、クリティカル・チェーンに基づくルール、資源に基づくルールの3種類から、代表的なルールを選び、提案手法におけるそれらの特性も明らかにした。

提案手法は、遅延事象の発生頻度が少ない場合において良好なスケジュールを生成することを確認するとともに、遅延事象の処理に要する時間が長いときにおいても同様の解を得ることも明らかとなった。また、遅延事象の発生頻度、それに要する処理時間が長い場合においては新たなタイミング方策としてハイブリッド型が有用であることも確認した。

以上、本論文は、第2章においてプロジェクト管理からみたスケジューリング問題を定義し、動的スケジューリングを軸に分類整理を行った。第2、第3章においてプロジェクト期日を遵守する動的スケジューリング手法の提案したが、限られた（不確実性の大きさ）環境では有用であった。しかしながら、不確実性の種類、規模など依然と確認せねばならない状況がある。数値実験を通し、提案手法の振る舞いを再度確認し、効果的な動的スケジューリング方策の再構築が今後の課題となる。

謝辞

本研究を進めるにあたり、終始適切なお指導およびご助言を賜りました大阪大学大学院情報科学研究科 森田 浩教授に厚く御礼申し上げます。

論文審査の副査をお引き受け戴き、懇切なるご指導およびご指摘を戴きました、大阪大学大学院情報科学研究科 谷田 純教授、沼尾 正行教授、梅谷 俊治准教授に謹んで感謝を申し上げます。

摂南大学理工学部 諏訪 晴彦教授には、本研究を進めるきっかけを与えてくださるとともに、ご指導ご鞭撻を賜りました。また、公私にわたるご助言に大変感謝するとともに、ここに厚く御礼申し上げます。

大阪大学大学院情報科学研究科 蓮池 隆助教には研究途上において懇切なお指導を賜りました。謹んで感謝申し上げます。

研究全般にわたり多くの有益なお助言を頂きました摂南大学理工学部 川野 常夫教授、福井 裕技師、森田 大輔博士には心から感謝の意を表します。日頃の様々な事柄でお世話になりました、大阪工業大学情報科学部情報ネットワーク学科 杉川 智特任助教には貴重なご助言を頂きました。深く感謝いたします。

森田研究室のみなさまには謹んで感謝申し上げます。研究を進める上で多大な協力を頂き感謝します。またこの論文を直接または間接的に支援していただいた皆様に厚く御礼申し上げます。

最後に、私の我が儘を叶えてくれた両親に心から感謝いたします。ありがとう。

参考文献

- [1] 鍋島一郎, “スケジューリング理論”, 森北出版, 1974.
- [2] 関口 恭毅, 木瀬 洋, “スケジューリング理論の基礎と応用-I: 順序付けの基礎数理”; システム制御情報学会論文誌, Vol. 44, No. 8, pp. 468–475, 2000.
- [3] 関根 智明, “PERT・CPM”; 日科技連, 1994.
- [4] P. Brucker, A. Drexl, R. Möhring, K. Neumann and E. Pesch, “Resource-constrained project scheduling: Notation, classification, models, and methods”; *European Journal of Operational Research*, Vol. 112, pp. 3–41, 1999.
- [5] J. Blazewicz, J. Lenstra and A. Rinnooy Kan, “Scheduling subject to resource constraints: Classification and complexity”; *Discrete Applied Mathematics*, Vol. 5, No. 1, pp. 11–24, 1983.
- [6] J. Blazewicz, K. Ecker, E. Pesch, G. Schmidt and J. Weglarz, “*Scheduling Computer and Manufacturing Process*”; Springer, 2001.
- [7] P. Brucker, “*Scheduling Algorithms*”; Springer, 2001.
- [8] S. Van de Vonder, E. Demeulemeester and W. Herroelen, “Proactive heuristic procedures for robust project scheduling: An experimental analysis”; *European Journal of Operational Research*, Vol. 189, No. 3, pp. 723–733, 2008.
- [9] M. Bruni, P. Beraldi, F. Guerriero and E. Pinto “A heuristic approach for resource constrained project scheduling with uncertain activity durations”; *Computers & Operations Research*, Vol. 38, No. 9, pp. 1305–1318, 2011.

- [10] F. Deblaere, E. Demeulemeester and W. Herroelen “Proactive policies for the stochastic resource-constrained project scheduling problem ” ; *European Journal of Operational Research*, Vol. 214, No. 2, pp. 308–316, 2011.
- [11] J. Kelley, “ The Critical-Path Method: Resource Planning and Scheduling ” ; in J. F. Muth and G.L. Thompson(Eds.), *Industrial Scheduling*, Prentice-Hall, Englewood Cliffs, pp. 347–365, 1963.
- [12] G. Brooks and C. White, “ An algorithm for finding optimal or near optimal solutions to the production scheduling problem ” ; *Journal of Industrial Engineering*, January - February Issue, pp. 34–40, 1965.
- [13] D. Golenko-Ginzburg and A. Gonik. “ A heuristic for network project scheduling with random activity durations depending on the resource allocation ” ; *international Journal on Production Economics*, Vol. 55, pp. 149–162, 1998.
- [14] C. Briand, E. Coffman and F. Roubellat, “ Scheduling with time lags and preference: A heuristic ” ; Proceedings of *the 8th Workshop on Project Management and Scheduling*, Valencia, pp. 3–5, 2002.
- [15] P. Czyzak and A. Jaskiewicz, “ Metaheuristic technique for solving multi objective investment planning problem ” ; *Control and Cybernetics*, Vol. 25, pp. 177–187, 1996.
- [16] R. Kolish, “ Efficient priority rules for the resource-constrained project scheduling problem ” ; *Journal of Operations Management*, Vol. 14, No. 3, pp. 179–192, 1996.
- [17] E.O.P. Akpan, “ Priority rules in project scheduling: a case for random activity selection ” ; *Production Planning and Control*, Vol. 11, No. 2, pp. 165–170, 2000.
- [18] E. Szelke and R. Kerr, “ Knowledge-based reactive scheduling ” ; *Production Planning and Control*, Vol. 5, No. 2, pp. 124–145, 1994.
- [19] O. Lambrechts, E. Demeulemeester and W. Herroelen “ Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities ” ; KU Leuven Working Paper No. KBI 0606, 2006.

- [20] 諏訪 晴彦, 森田 大輔, 三道 弘明, “プロジェクト管理におけるダミーに基づくクリティカル・チェーン・スケジューリング”; システム制御情報学会論文誌, Vol. 22, No. 12, pp. 407–415, 2009.
- [21] E. Demeulemeester and W. Herroelen, “*Project Scheduling(A Research Handbook)*”; Kluwer Academic Publishers, 2002.
- [22] P. Galloway, “Survey of the construction industry relative to the use of CPM scheduling for construction projects”; *Journal of Construction Engineering and Management*, Vol. 132, No. 7, pp. 697–711, 2006.
- [23] L. Church and R. Uzsoy, “Analysis of periodic and event-driven rescheduling policies in dynamic shop”; *International Journal of Computer Integrated Manufacturing*, Vol. 5, pp. 153–163, 1992.
- [24] H.P. Kao, B. Wang, J. Dong and K.C. Ku, “An event-driven approach with makespan/cost tradeoff analysis for project portfolio scheduling”; *Computer in Industry*, Vol. 57, pp. 379–397, 2006.
- [25] Z. He, R. Liu and T. Jia “Metaheuristics for multi-mode capital-constrained project payment scheduling”; *European Journal of Operational Research*, Vol. 223, pp. 605–613, 2006.
- [26] D. Malcolm, J. Roseboom and C. Clark, “Application of a technique for research and development program evaluation”; *Operational Research* Vol. 7, No. 5, pp. 646–669, 1959.
- [27] J. Kelley and M. Walker, “Critical-Path Planning and Scheduling”; *Proceedings of the Eastern Joint Computer Conference*, 1959.
- [28] P. Galloway. “Survey of the construction industry relative to the use of CPM scheduling for construction projects” *Journal of Construction Engineering and Management*, Vol. 132, No. 7, pp. 697–711, 2006.

- [29] C. Akkan, A. Drexl and A. Kimms, “Network decomposition-based benchmark results for the discrete time-cost tradeoff problem”; *European Journal of Operational Research*, Vol. 165, pp. 339–358, 2005.
- [30] S.S. Leu and C. Yang, “GA-based multi-criteria optimal model for construction scheduling”; *Journal of Construction Engineering and Management*, Vol. 125, No. 6, pp. 420–427, 1999.
- [31] E.M. Goldratt, “*Critical Chain*” The North River Press, MA, USA, 1997.
- [32] S. Van de Vonder, E. Demeulemeester, W. Herroelen and R. Leus, “The use buffers in project management: Trade-off between stability and makespan”; *International Journal of Production Research*, Vol. 44, No. 2, pp. 215–236, 2005.
- [33] B. De Reyck and W. Herroelen, “A branch-and-bound procedure for the resource-constrained project scheduling problem with generalized precedence relations”; *European Journal of Operational Research*, Vol. 111, No. 1, pp. 152–174, 1998.
- [34] K. Nonobe and T. Ibaraki, “A tabu search approach to the constraint satisfaction problem as a general problem solver”; *European Journal of Operational Research*, Vol. 106, No. 2–3, pp. 599–623, 1998.
- [35] L. Özdamar and G. Ulusoy, “A survey on the resource-constrained project scheduling problem”; *IIE Transactions*, Vol. 27, pp. 574–586, 1995.
- [36] S. Hartmann and R. Kolisch, “Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem”; *European Journal of Operational Research*, Vol. 127, pp. 394–407, 2000.
- [37] R. Kolisch and R. Padman, “An integer survey of deterministic project scheduling”; *Omega*, Vol. 29, pp. 249–272, 2001.
- [38] R. Kolisch and S. Hartmann, “Experimental investigation of heuristics for resource-constrained project scheduling problem: An update”; *European Journal of Operational Research*, Vol. 174, pp. 23–37, 2006.

- [39] N.H. Pana, P.W. Hsaio and K.Y. Chena, “A study of project scheduling optimization using tabu search algorithm”; *Engineering Applications of Artificial Intelligence*, Vol. 21, No. 7, pp. 1101-1112, 2008.
- [40] M. Verhoeven, “Tabu search for resource-constrained scheduling”; *European Journal of Operational Research*, Vol. 106, No. 23, pp. 266-276, 1998.
- [41] D. Jeffcoat and R. Bulfin “Simulated annealing for resource-constrained scheduling”; *European Journal of Operational Research*, Vol. 70, No. 1, pp. 43-51, 1993.
- [42] J.H. Cho and Y.D. Kim, “A simulated annealing algorithm for resource-constrained scheduling problem”; *journal of the Operational Research Society*, Vol. 48, pp. 736–744, 1997.
- [43] S. Hartmann, “A competitive genetic algorithm for resource-constrained project scheduling”; *Naval Research Logistics*, Vol. 45, pp. 733–750, 1998.
- [44] M. Mika, G. Waligóra and J. Weglarz “Simulated annealing and tabu search for multi-mode resource-constrained project scheduling with positive discounted cash flows and different payment models”; *European Journal of Operational Research*, Vol. 164, No. 3, pp. 639–668, 2005.
- [45] R. Kolisch, “Efficient priority rules for the resource-constrained project scheduling problem”; *Journal of Operations Management*, Vol. 14, No. 3, pp. 179–192, 1996.
- [46] L. Church and R. Uzsoy, “Analysis of periodic and event-driven rescheduling policy in dynamic shops”; *International Journal of Computer Integrated Manufacturing*, Vol. 5, No. 3, pp. 153–163, 1992.