

修士学位論文

題 名

プロジェクトマネジメントにおけるタ
スク割当の多目的最適化

頁 1 ～ 4 4

指導教員 近藤 伸彦准教授

2024年 1月10日提出

東京都立大学大学院
経営学研究科経営学専攻（博士前期課程）

学修番号 22836304

ふ り が な かとう たいち
氏 名 加藤 太一

目 次

第 1 章	序論	1
第 2 章	プロジェクトマネジメントとタスク割当の最適化	2
2.1	プロジェクトマネジメントの概念	2
2.2	タスク割当の重要性と課題	2
2.3	タスク割当と人員資源の最適化に関する先行研究	4
2.4	本研究の位置づけと研究の進展方向性	5
第 3 章	多目的最適化問題とその解法	6
3.1	多目的最適化手法	6
3.2	遺伝的アルゴリズムと NSGA-II	7
3.3	パラメータと重み設定	8
第 4 章	タスク割当の多目的最適化	10
4.1	遺伝的アルゴリズムへのコーディング	10
4.2	目的関数	11
4.3	制約条件	12
4.4	データセット	12
4.5	開発環境	13
第 5 章	実験設計	14
5.1	実験の目的	14
5.2	実験条件	14
5.3	遺伝的アルゴリズムの適合度評価	16
第 6 章	実験結果, 評価および考察	19
6.1	実験結果の概要	19
6.2	実験結果の詳細分析	24
6.3	目的関数間のトレードオフ	29
6.4	重みの影響	32
6.5	最適化結果のガントチャートによる妥当性の評価	38
6.6	今後の研究方向	41
第 7 章	結論	42
	参考文献	43
	謝辞	44

第1章 序論

プロジェクトマネジメントは、日々の業務から大規模事業に至るまで、組織の戦略的目標達成の中核を担うプロセスである。現代のプロジェクトマネジメントは、プロジェクトマネージャーが全体像を把握し、各タスクが全体にどのように寄与するかを理解し、効率的なリソース配分を行うことを要求する。この複雑なプロセスを適切に適用することで、プロジェクトは目的を効果的に達成し、組織のより広い戦略的目標に貢献する。重要な要素として、強固なリーダーシップ、明確なコミュニケーション、適切なプロジェクトマネジメントツールの選択が挙げられる。

日本経済において、労働力不足や生産性の問題は深刻であり、これに対するプロジェクトマネジメントの質の向上は組織の競争力を高める鍵となる。しかし、多くのプロジェクトで見られる属人的な管理は、効率的な実行を妨げる障壁である。

本研究は、プロジェクトマネジメントにおける効果的なタスク割当の最適化に焦点を当てる。具体的には、多目的最適化アルゴリズムを用いて、メンバーのスキル、コスト、予算、依存関係などの多様な要件を満たす最適なバランスを見出すことを目的とする。この課題は、適切なタスク割当、コストと時間の効率化、限られた予算内での成果最大化などを含む。これらの要素を適切にバランスさせることは、プロジェクトマネージャーにとっての日々の挑戦であり、本研究が提供する新しいアプローチは、実務における意思決定を支援し、日本のビジネス環境における生産性向上に貢献することが期待される。

第2章 プロジェクトマネジメントとタスク割当の最適化

本章では、プロジェクトマネジメントの基本的な概念とタスク割当の最適化の重要性について論じる。この章では、プロジェクトマネジメントがどのように発展してきたか、そしてタスク割当がプロジェクト成功にどう影響するかを明らかにし、最適化手法がこれらのプロセスにどのように貢献するかを考察する。

2.1 プロジェクトマネジメントの概念

プロジェクトマネジメントは、複雑化する現代のビジネス環境において、組織の戦略的な目標を達成するための中核的なプロセスである。Project Management Institute (PMI) によると、プロジェクト要求事項を満たすために、知識、スキル、ツールと技法をプロジェクト活動に適応することと定義されている[1]。プロジェクトマネジメントは、限られたリソースを最大限に活用し、多様なステークホルダーの要求を満たすために、タスクの効果的な割当と管理を行う技術と方法論の体系であり、意図した成果を上げるためにプロジェクトの作業を導くことを指す[1]。プロジェクトマネジメントの範囲は、単にタスク割当や管理に限らず、目標の設定、リスクの評価、品質の保証、コミュニケーションの促進など、プロジェクトの成功に必要な多岐にわたる要素を包含している。その重要性は、日々の業務から大規模な事業プロジェクトまで、あらゆるレベルでの成果に直結する。特に、複数のプロジェクトが同時に進行する状況では、効果的なタスク割当がプロジェクトの成功を決定づける。

2.2 タスク割当の重要性と課題

タスク割当は、プロジェクトマネジメントの中核を成す要素であり、プロジェクトの目標を達成するための主要な手段である。タスク割当の過程では、メンバーのスキル、経験、および適性を考慮し、それぞれのタスクに最も適した人材を配置することが求められる。このプロセスは、タスクの品質と効率を大幅に向上させることができ、結果としてプロジェクト全体の成功に大きく寄与する。

タスク割当の難しさは、プロジェクトに関わる変数の多さとそれらの変数間の相互作用

にあり、特にメンバーが複数のプロジェクトを並行して担当している場合、その複雑性はさらに増す。各メンバーが持つ独自のスキルセットと利用可能な時間、プロジェクト間の優先順位、資源の制約などが絡み合い、適切な割当を見つけることは容易ではない。プロジェクトマネージャーはこれらの要因を総合的に分析し、全体の目標達成に向けた最適なリソース配分を実行する必要がある。

さらに、タスク割当においては、スキル、コスト、予算、依存関係といった複数の基準をバランス良く考慮することが不可欠である。各メンバーのスキルレベルとタスクの要求とのマッチング、タスクに関連するコストの見積もり、全体の予算内での資源配分、タスク間の依存関係の管理など、これら多元的な基準を適切に評価し、調整することがプロジェクトの成果を最大化する鍵となる。

これらの基準間にはしばしばトレードオフが存在し、例えば、高いスキルを持つメンバーの割当は品質と効率の向上に寄与するが、それはコストの増加を意味することもある。プロジェクトマネージャーは、プロジェクトのスコープ、期間、予算、およびステークホルダーの期待など、多様な制約と目標の間で最適なバランスを見つけるための戦略を立てることが求められる。

タスク割当の過程を視覚化した例を図 1 に示す。ここでは、メンバーのスキル、経験、および適性を考慮して、各タスクに最も適した人材を配置している。図 1 では、プロジェクトマネージャーがプロジェクトの目標を達成するために、メンバーごとのスキルセットとコストを考慮しながらタスク割当を行い、ガントチャートを用いてプロジェクトのスケジュールを管理している様子がわかる。このように、タスク割当とプロジェクトスケジューリングはプロジェクトマネジメントの成功に不可欠であり、全体の目標達成に向けた戦略の一環として機能する。

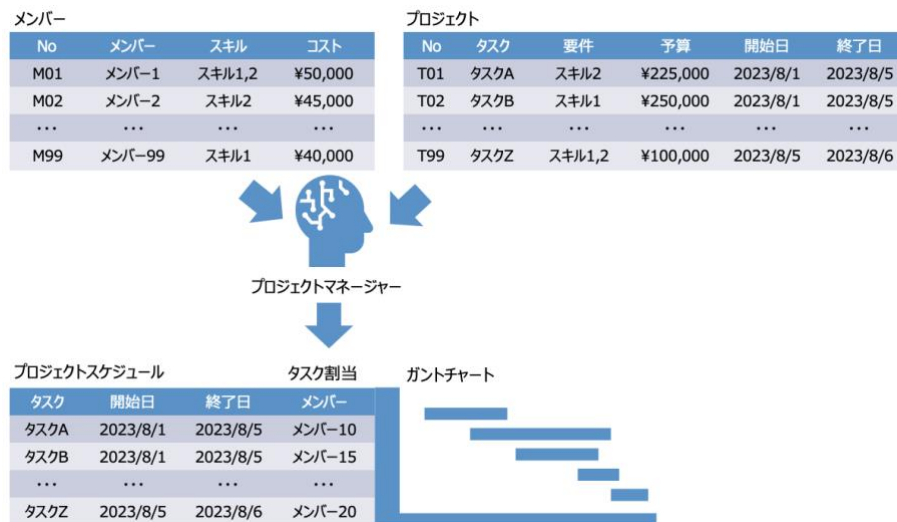


図 1 タスク割当のイメージ

2.3 タスク割当と人員資源の最適化に関する先行研究

高津ら[2]はプロジェクトマネジメントにおいて、メンバーの割当と人員資源の最適化を数理モデルによって行う方法を提案した。このモデルはメンバーの意図と経験を反映し、屋外工事業務の効率化を実現した。人間の意図を数値化し、割当品質の向上を図るとともに、プロジェクトマネージャーがより適切なスケジュールを選択するための基盤を築いた。具体的には、アンケートやヒアリングを通じて収集したデータをモデル化し、リソース配分プロセスを数理的に支援した。この研究は、意図の構造化と推測手法を統合し、プロジェクトの成功への貢献が期待されている。

高須賀[3]は、ソフトウェア開発における人的資源の効果的な割当モデルを提案し、メンバーの能力差を考慮した最適化を実現した。大規模なシステム開発、特に多機能を同時に進める EMS の開発を例に、メンバーの個々のスキルと経験を活かすリソース配分の重要性を強調している。この研究は、メンバーの得意・不得意を数理モデルで反映し、直感ではなく数理的に根拠づけられたタスク割当を実現することを目的としている。このモデルは非定型業務にも適用可能で、限られた人員で最大の生産性を引き出せるよう設計されている。実業務に適用し性能評価を行った結果、メンバーの能力差を考慮した効果的なリソース配分を実現した。

小林と森口[4]は、NSGA-II という多目的遺伝的アルゴリズムを IT プロジェクトスケジュー

ーリングに適用し、その複雑性を解決する手法を提案している。彼らの提案する手法は、納期短縮、要員の重複タスク日数の削減、要員数の最小化という 3 つの目的を同時に最小化することで、多様なプロジェクト要件に対応するスケジューリングソフトウェアの開発に貢献するものである。この研究により、プロジェクトマネージャーは複数の準最適解の中から、プロジェクトの状況に合ったスケジュールを選択できるようになり、環境の変化やプロジェクトの動的な要求に対する適応性が向上する。また、提案されたアルゴリズムは、パレート最適解の一部を効率的に抽出し、探索初期段階で導入することで探索効率を高めていることが示されている。これは、限られた計算リソースでも迅速かつ正確なスケジューリングが可能であることを意味しており、プロジェクトマネジメントにおけるスケジューリング手法の進化に寄与し、プロジェクトの成果を最大化する新たな方向性を示している。数値実験とインタビューを通じて、提案手法がプロジェクトマネージャーの負担を軽減し、優れたプロジェクト運営を実現することが確認されている。

2.4 本研究の位置づけと研究の進展方向性

本研究は、プロジェクトマネジメントにおけるタスク割当のアプローチを再考し、個々のメンバーにとって最も適合するタスク割当方法に着目している。これは、単にタスクを時間軸上に配置することを超え、メンバーの独自の能力、経験、利用可能な時間を考慮し、プロジェクトの複数の目標を同時に考慮することを目指すものである。

プロジェクト内でのタスク配分は、単一の解ではなく、多角的な視点からの複数の解を求める複雑な問題である。そのため、複数の目標を持つタスク割当問題に対して、実用的かつ効果的な戦略を提案することを目指している。これには、メンバーのスキルマッチング、プロジェクト要件への適合性、リソースの最適な利用など、複数の側面が考慮される。特にプロジェクト固有の制約、如何にして限られたリソースを効率的に割当るか、そしてそれらがプロジェクトの納期、品質、コストにどのように影響を及ぼすかという問題に対処する。

総じて、本研究はプロジェクトマネジメントの実践における改善策を提供し、より効果的なタスク割当が可能となるような方法論の開発を目指している。

第3章 多目的最適化問題とその解法

本章では、多目的最適化問題とその解法について述べる。複数の競合する目的関数を持つ最適化問題からパレート最適解を導出する方法に焦点を当て、多目的最適化の基本的なアプローチとパレート最適性の概念について紹介する。また、遺伝的アルゴリズムの一つである NSGA-II の原理と特徴についても詳しく解説し、その効率的な探索能力について検討する。

3.1 多目的最適化手法

多目的最適化問題では、相反する目的関数群に対してバランスの取れた解を探索する。パレート最適性は、一方の目的を達成するために他方を犠牲にするトレードオフが生じる状態を指し、最適解を求める上での基本原則である。たとえば、この問題は数式で次のように表現される。

$$\begin{array}{ll} \text{minimize/maximize} & f_1(x), f_2(x), \dots, f_m(x) \\ \text{subject to} & x \in X \end{array}$$

複数の目的関数 $f_1(x), f_2(x), \dots, f_m(x)$ を最小化または最大化する解 x を求める。ここで、解 x は意思決定空間 X に属している必要がある。

複数の目的関数が競合または相反する場合にどのようにパレート最適解を導出するかについて、多目的最適化の基本的なアプローチとその解法について述べる。たとえば、ある企業が新製品の開発においてコストを最小限に抑えつつ、品質を最大化し、かつ市場への投入時間を短縮したい場合、これらの目的は同時に達成することが難しい。なぜなら、通常、品質向上にはコストと時間がより多くかかるからである。このような状況において、多目的最適化は最良のトレードオフを見つけ出すための有効なアプローチを提供する。

多目的最適化においては、図 2 に示すように、単一の最適解という概念が存在せず、代わりにパレート最適性という概念が用いられる。パレート最適性とは、ある解が他のすべての解に対して少なくとも一つの目的で優れており、他の目的において劣っていない状態

を指す。この条件を満たす解の集合をパレートフロントと呼び、通常はこれらの解から状況に応じて最も適切な解を選択する。パレートフロントには第 1 フロント、第 2 フロントといった階層があり、第 1 フロントは他のどの解にも劣らない最上位の解の集合である。

このアプローチの利点は、異なる目的関数の間でのトレードオフを明確に理解できることにある。例えば、コストを若干増やすことで品質を大幅に向上させることができる解や、逆にコストを大幅に削減する代わりに品質を若干落とす解など、さまざまな選択肢を提供する。

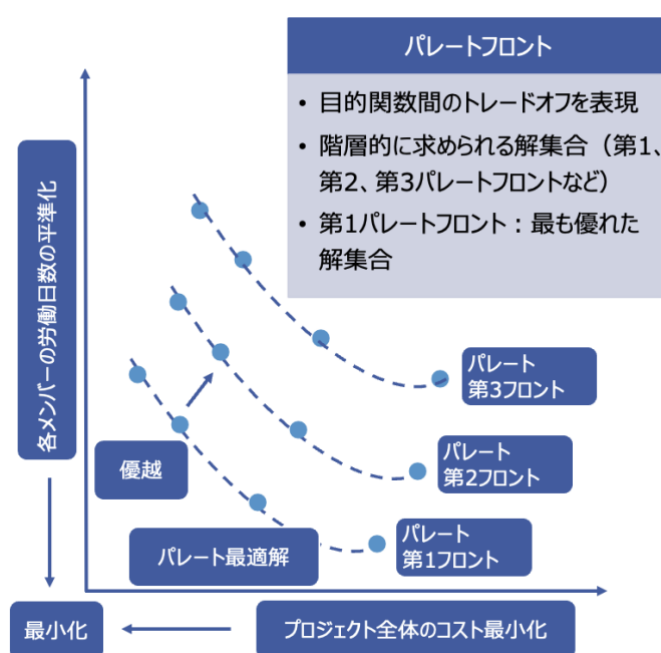


図 2 パレートフロント

3.2 遺伝的アルゴリズムと NSGA-II

遺伝的アルゴリズムは、生物の進化のメカニズムを模倣した確率的探索手法であり、複雑な多目的最適化問題を解く際に広く利用されている。この手法は、互いに競合する複数の目的を持つ問題において、解の多様性を維持しながら最適解を進化させる能力に優れているとされる。その基本的な原理は、初期の候補解の集団から始め、選択、交叉、突然変異といった操作を反復して、問題に対するより良い解を進化させるプロセスである。特に遺伝的アルゴリズムは、図 3 に示されたフローチャートに沿って、解の多様性を維持しながら

ら、最適解または準最適解へと探索を進める能力に優れている。

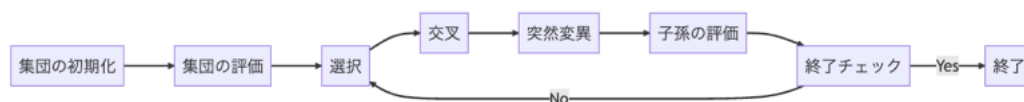


図 3 遺伝的アルゴリズムのフローチャート

Deb ら[5]が提案した **NSGA-II** は、この遺伝的アルゴリズムの枠組みを多目的最適化問題に特化して拡張したものである。複数の目的関数を持つ問題において、それぞれの目的関数が互いに競合する場合、**NSGA-II** は効率的にパレートフロントを形成する解の集合を探索する。このアルゴリズムは、特に計算コストが高い問題、エリート主義を採用しない問題、シェアリングパラメータを必要とする問題といった、多目的進化アルゴリズムが直面する一般的な課題に対処するために設計されている。

NSGA-II の核心は、非劣ソートと呼ばれる手法による解のランク付けと、混雑度計算に基づく多様性の維持にある。非劣ソートとは、各解がどの程度他の解に優れているか、また他の解に劣っていないかを評価するプロセスである。このプロセスを通じて、最適解に近い解に高いランクが付けられる。混雑度計算は、解の周囲の密集度を評価し、解が適応的な多様性を持っているかどうかを判断するための手法である。これにより、パレートフロント上の解が均等に分布するように導くことができる。

シミュレーション結果からは、**NSGA-II** が **PAES** や **SPEA** など他のエリート主義を採用する多目的進化アルゴリズムよりも、解の多様性と真のパレート最適フロントへの収束において、多くの問題で優れていることが明らかになっている。さらに、制約のある多目的最適化問題への適用可能性を高めるため、制約付き問題への効率的な適用を目指して支配関係の定義を変更する提案もしている。実施されたテスト問題における制約付き **NSGA-II** のシミュレーション結果は、他の制約付き多目的最適化手法と比較して優れた性能を示しており、**NSGA-II** が実世界のより複雑な多目的最適化問題への適用を促進する可能性を示唆している。

3.3 パラメータと重み設定

遺伝的アルゴリズムの適用に際して、パラメータと重み設定は解の探索効率と質に大き

な影響を与える。以下に、本研究で用いられた主要なパラメータと重みについて説明する。

交叉確率

新しい世代での個体が親の遺伝子を受け継ぐ割合を定める。この割合を適切に設定することで、良質な解の特性を保ちつつ多様性を確保することが可能となる。

突然変異の確率

突然変異の確率は、個体の遺伝子列にランダムな変更が加えられる可能性を示す。これには個々の遺伝子座における突然変異の確率と、個体全体における突然変異の確率の二つの側面がある。解の多様性を維持し、局所的な最適解からの脱却を促すため、これら二つの確率のバランスを取ることが重要である。

交叉パターン

子孫を生成する際の遺伝子の組合せ方を決める重要な要素である。本研究では、「一点交叉」、「二点交叉」、「一様交叉」という三種類を採用している。一点交叉は、親の遺伝子列を一定の点で断ち切り、新たな個体を作り出す基本的な手法である。二点交叉は、より複雑な遺伝的特徴を子孫に受け継がせる可能性があり、一様交叉は遺伝的変異の度合いを高める。

突然変異タイプ

突然変異タイプは、遺伝的アルゴリズムにおける解の探索過程において、新たな遺伝的変異を導入する方法を定める。具体的には、個体群内の各個体に対して、その遺伝子列にランダムな変化を加えることで、探索空間における多様性を増やし、アルゴリズムが局所最適解に陥るのを防ぐ役割を果たす。

スケーリング係数

異なる単位を持つ目的関数間の公平な評価を実現するために導入される。例えばコストと労働日数のように単位が異なる目的関数の結果を適切に比較することが可能になる。

制約条件の重み

各制約条件に与えられた重みは、その制約違反が解の適合度に与える影響の大きさを示す。これにより、解の選択過程において、どの制約をより重視するかを調整することができる。

第4章 タスク割当の多目的最適化

本章では、プロジェクトマネジメントにおけるタスク割当の問題を、多目的最適化の観点からアプローチする方法を提案する。タスク割当問題に対して遺伝的アルゴリズムを用いることにより、プロジェクトのコスト最小化とメンバーの労働日数の平準化という二つの目的を同時に達成する解を探索する。この章では、まず遺伝的アルゴリズムをタスク割当問題に適用するためのコーディング方法を紹介し、次に目的関数の設計と制約条件について詳細を述べる。さらに、遺伝的アルゴリズムのパラメータと重み設定とその調整方法についても検討する。最後に、データセットの構築と管理について説明し、開発環境としてのツール選定に関しても触れる。こうして、プロジェクトの具体的な状況に応じた効率的なタスク割当の最適化手法を、理論的および実践的な観点から展開する。

4.1 遺伝的アルゴリズムへのコーディング

本研究では、タスク割当問題を多目的最適化問題と捉え、遺伝的アルゴリズムを適用する。このアプローチは、問題の柔軟性と効率性を考慮した上で、最適なりソース配分を発見するための有効な手段を提供する。ここで、遺伝的アルゴリズムにおける個体は、タスクに割当られるメンバーのインデックスを表す整数変数で定義される。この表現法により、遺伝子はタスク割当の解決策を符号化するものとなり、各遺伝子座は特定のタスクへのメンバーの割当を明示する。

タスク t に割当られたメンバーのインデックスを示す整数変数を x_t と定義する。 x_t が m である場合、タスク t がメンバー m に割当られている状態を示す。 x_t はメンバーの集合の中から取りうる値の範囲である 0 から $M-1$ までの整数のいずれかを取ることになる。ここで M はメンバーの総数を示す。プロジェクトの各メンバーはユニークなインデックスを割当られており、これによってタスクとメンバーの割当が一意に定義される。数式で次のように表現される。

$$x_t \in \{0, 1, \dots, M - 1\}$$

図 4 に示すように、初期個体群の生成はランダムに行われ、各個体はプロジェクトのタスクに対するランダムな割当を表現する。

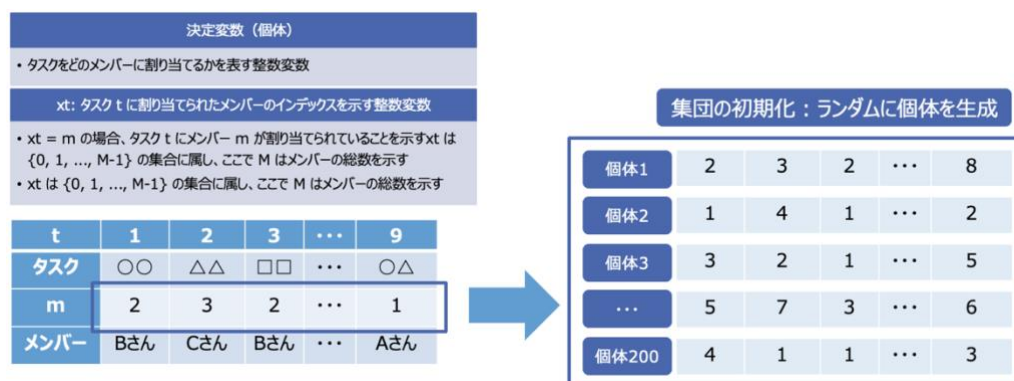


図 4 遺伝的アルゴリズムへのコーディング

4.2 目的関数

目的関数の設計において、設計者は様々な目的関数を自由に設計することが可能である。この柔軟性は、問題に対する多面的なアプローチを可能にし、特定の業界や状況に依存しない解決策の探索を促進する大きなメリットをもたらす。本研究では、実践的に重要と考えられる二つの目的関数を選定し、それらが普遍的な価値を持つことを重視しつつ、実験を通じてそれらの有効性を検証している。本研究における第一の目的関数は総コストの最小化である。この目的関数を採用した理由は、プロジェクトマネジメントにおいて総コストの削減は基本的な要求であり、多くの組織がコスト効率の向上を目指しているためである。総コストの最小化には、組織の財務的持続可能性と直接的な関連があり、プロジェクトの成果に対して投じた投資の対効果を最大化するための根本的な手法となる。

第二の目的関数は、メンバーの労働日数の標準偏差を最小化することによる作業負荷の平準化である。この目的関数の選択は、作業の過負荷または過少な負荷がプロジェクトの進行に与える悪影響を考慮したものである。負荷の均一化は、プロジェクトチーム内のバランスを取り、過度なストレスや燃え尽き症候群を防ぐことで、チーム全体の生産性と満足度を高める。また、平準化された作業負荷は、個々のメンバーが専門性を発揮しやすい環境を作り出すため、プロジェクトの質と効率の両方を向上させることが期待される。

4.3 制約条件

制約条件は、プロジェクトマネジメントの複雑さをモデル化し、現実的なプロジェクト運営をシミュレーションするために不可欠である。本研究で設定した3つの制約条件は、それぞれ特定のプロジェクトマネジメントの側面を反映している。

C1: メンバーの排他的タスク割当

現実的なプロジェクト実行におけるリソース配分を反映し、各メンバーが同時に異なるタスクに従事してはいけない制約を設けた。これにより、リソースの割当と時間管理の現実性が確保される。

C2: タスクの予算内での割当

プロジェクトの財務的側面を考慮し、各タスクが与えられた予算内で完了しなければならないという制約を設けることで、コスト意識的な意思決定を促進し、経済的制約の下でのリソース割当を最適化する。

C3: スキルマッチの必要性

メンバーがタスク要件にマッチしていることを確認するこの制約は、ただタスク要件に応じたスキルの有無だけでなく、メンバーのキャリア開発や成長にも焦点を当てている。メンバーが身につけたいスキルを事前にタスク割当に組み込むことで、個人の成長と組織のニーズの両方を満たすような割当が可能になる。

これらの制約条件は、プロジェクトの実行可能性と効率性を高めるために、バランスを取りながら選ばれた。制約条件の厳格さは、適応度関数によって評価され、最適な解の探索におけるペナルティとして機能する。この適応度関数と制約条件の詳細な扱いについては、後述の第5章「実験設計」にて詳しく述べられている。

4.4 データセット

本研究で使用されたデータセットは、プロジェクトマネジメントに関連する複数の要素から構成されている。具体的には、以下の情報を含んでいる。

タスク情報

各プロジェクトタスクに関連する詳細情報を含む。これにはプロジェクト名、タスクID、タスク名、開始日、終了日、予算、必要とされるスキルセット、前提となるタスク、所要日

数が含まれる．この情報によって，タスク間の依存関係や必要とされるリソースを把握することができる．

メンバー情報

プロジェクトに関わるメンバーのスキルとコストを集約した情報．メンバーID，名前，持っているスキルセット，およびそのコストが記載されている．この情報は，タスクへのメンバー割当と，スキルとコストのバランスを最適化するために使用される．

最適化結果

最適化アルゴリズムが出力した解の集合．これは通常，プロジェクトのタスクとリソース割当のスケジュール形式で表現され，プロジェクトの進捗状況とリソース割当を視覚的に確認するために使用される．

これらのデータセットは，最適化アルゴリズムの入力として，また得られた解の評価という二つの主要な目的に使用された．本研究で扱われるデータセットは，プロジェクトマネジメントの一般的な概念と実践に基づいており，特定のツールや方法論に依存しない．

4.5 開発環境

本研究の実装においては，Visual Studio Code (VS Code) を統合開発環境として使用し，プログラミング言語に Python を採用した．Python はその可読性の高さと，科学技術計算に適した豊富なライブラリを備えており，多目的最適化問題への対応に優れている．進化計算には，DEAP ライブラリを使用しており，特に多目的遺伝的アルゴリズム NSGA-II が組み込まれていることから，本研究の要件に合致した効率的な最適解探索を実現する．

第5章 実験設計

本章では、プロジェクトマネジメントにおけるタスク割当問題への多目的遺伝的アルゴリズムの応用可能性を検証するための実験設計を詳述する。アルゴリズムが実際のプロジェクトマネジメントの複雑な要求に対して、どのようにしてコストと労働負荷の最適化を同時に行うか、遺伝的アルゴリズムによる解の生成と評価のプロセス、実験条件の設定、評価指標の選定、および実験の流れについて詳細に説明する。

5.1 実験の目的

本研究における実験の目的は、プロジェクトマネジメントにおけるメンバー割当問題に対して、多目的遺伝的アルゴリズムを応用する手法の有効性を検証することである。具体的な焦点は以下の二点に置かれる。

コスト最小化と労働日数平準化のトレードオフ

プロジェクトの総コスト最小化と、メンバーの労働日数の平準化を同時に達成しようとする多目的遺伝的アルゴリズムの効率と効果を検証する。これは、コストとリソースの最適な配分のバランスを取ることが求められる現実のプロジェクトマネジメントの課題を反映している。

遺伝的アルゴリズムによる意思決定支援

遺伝的アルゴリズムが、プロジェクトマネージャーの意思決定をどの程度効率的かつ効果的に支援できるかを評価する。具体的には、複数の目的関数と制約条件を考慮した状況下でのタスク割当の最適化に注目し、遺伝的アルゴリズムが提供する解の品質を分析する。

これらの目的は、実験を通じて、遺伝的アルゴリズムの応用がプロジェクトマネジメントの複雑な問題解決において実用的かつ有益なアプローチであるかどうかを明らかにすることを目指している。

5.2 実験条件

本研究における実験条件の設定は、実際のプロジェクトマネジメント環境を反映した有効なアルゴリズムの評価を目的としている。以下にその詳細を示す。

データセットの選択

現実のプロジェクトデータの複雑さと多様性を反映するための実験用のデータセットが必要であるが、実際のプロジェクトデータは、機密性の高さや入手の困難さから、直接的な使用が難しい。そのため、本研究では 3 つの架空のプロジェクトにわたる 111 のタスクと、20 人のメンバーを含むデータセットを作成した。

パラメータと重みの設定

本研究で適用された遺伝的アルゴリズムは、複数のパラメータと重みに基づいて最適化を行う。これらのパラメータは、アルゴリズムがどのように解を探索し、どのように解を生成するかに大きな影響を与える。主要なパラメータと重みは以下の通りである。

交叉パターン

交叉の手法として、'cxUniform'（一様交叉）を採用した。これにより、親個体の遺伝子を均一に組合せ、多様な子個体を生成することが可能となる。一様交叉は、個々の遺伝子座が独立して親から遺伝子を受け継ぐため、新たな特性の組合せを探索するのに有効である。

突然変異タイプ

'mutShuffleIndexes'（インデックスシャッフル突然変異）を選択した。このタイプの突然変異は、遺伝子列内のインデックスをランダムに入れ替えることで新しい遺伝子配置を試み、解の多様性を増すとともに、局所最適解に陥るのを防ぐ助けとなる。

選択タイプ

選択メカニズムには、'selNSGA2'（非劣解選択法）を採用している。これにより、複数の目的関数を持つ最適化問題において、適合度の高い個体を選択し、良質な遺伝子を次世代に伝えることができる。

実験設定において、集団サイズを 100、交叉確率を 1.0、世代数を 300 とした。突然変異確率に関しては、個々の遺伝子座における突然変異の確率を 0.1、個体全体に対する突然変異発生確率を 1.0 として設定した。これは、各遺伝子座において 10%の確率で突然変異が発生し、各世代において少なくとも 1 つの個体が突然変異を経験することを意味する。スケール係数は、異なる目的関数が異なるスケールを有する場合、それらを同一の尺度で比較可能にするために利用される。目的関数 2 に 3000, 4000, 5000 という異なるスケール係数を適用することで、スケール係数の妥当性を確認し、総コストと労働日数の影響に濃淡をつけて検証する。これにより、多目的最適化問題において、適切な解の導出におけるバランスを取ることが可能である。この係数は、単位や重要度が異なる目的

関数の評価値を比較し、組合せる際に重要な機能を果たす。

制約条件には、C1: メンバーの排他的タスク割当, C2: タスクの予算内での割当, C3: スキルマッチの必要性という3つの異なる種類が設定される。各制約条件には個別に重みを設定でき、各制約違反に対するペナルティの度合いを変更し、プロジェクトの特定の要件に応じて解の選択過程で異なる制約に優先順位を付けることが可能となる。たとえば, C1, C2, C3 に対してそれぞれ異なる重みを 1 から 1000 の範囲で設定することにより、アルゴリズムが各制約をどの程度厳格に遵守するかを調整できる。このように、各制約条件に対する重み付けによって、最適化プロセス中の制約遵守の度合いを効果的に管理することが可能である。

このアプローチにより、異なるプロジェクトシナリオに対して柔軟に対応し、最適な解を導出する過程での制約条件の適用を効果的に管理することができる。制約条件の重み付けによって調整されるペナルティは、解の適応度評価に直接影響を及ぼし、制約違反がある場合、その重みに応じたペナルティが目的関数の適合度に加算される。このペナルティの加算により、制約違反を含む解の適合度が低下し、次の世代に残りにくくなる。このメカニズムにより、制約条件を満たす解が選択される確率が高まり、最終的な解がプロジェクトの要件により適合するものになる可能性が高くなる。重み付けはアルゴリズムの挙動を制御し、より望ましい解を導き出すための重要な手段となる。

5.3 遺伝的アルゴリズムの適合度評価

本研究では、遺伝的アルゴリズムを用いて生成される解の性能を評価するために適合度を指標として採用している。適合度は遺伝的アルゴリズムにおける個体の性能を測定する尺度であり、最適な解の選出に不可欠である。以下の指標を採用した。

目的関数 1: 総コストの最小化の適合度

第一の適合度は、プロジェクト全体の経済的効率性を表す総コストの最小化に焦点を当てている。この適合度は、タスクに割当られたメンバーのコスト合計を最小にすることを目標とする。数式で次のように表現できる。

$$f_1 = \sum_{i=1}^n \text{cost}(\text{member}(i)) \times \text{duration}(\text{task}(i))$$

f1: 目的関数 1 (総コストの最小化の適合度)

n: タスクの総数

cost(member(i)): i 番目のタスクに割り当てられたメンバーのコスト

duration(task(i)): i 番目のタスクの期間 (労働日数)

目的関数 2: 労働日数の標準偏差の最小化の適合度

第二の適合度は、メンバー間の労働負担のバランスを評価するため、労働日数の標準偏差の最小化を目指している。これにより、全メンバーに均等な労働機会の提供を目指す。数式で次のように表現できる。

$$f_2 = \text{std_dev}(\text{labor_days}(\text{member}(i)))$$

f2: 目的関数 2 (労働日数の標準偏差の最小化の適合度)

std_dev: 標準偏差を計算する関数

labor_days(member(i)): i 番目のメンバーの労働日数の配列

制約条件 (C1, C2, C3) 違反の適合度へのペナルティ加算

制約条件の遵守度も適合度の重要な要素であり、各制約違反に対するペナルティが適合度に加算される。このペナルティは、解の適合度評価において重要な役割を果たし、生成される解が要求をどれだけ反映しているかを示す。数式で次のように表現できる。

$$f'_1 = f_1 + \text{penalty}$$

$$f'_2 = f_2 \times \text{scaling_factor} + \text{penalty}$$

f1, f2: 制約違反ペナルティを含む目的関数 1 と目的関数 2 の値

f1, f2: 元の目的関数 1 と目的関数 2

penalty: C1, C2, C3 の各制約違反ペナルティの合計

scaling_factor: 目的関数 2 に適用されるスケーリング係数

第 6 章 実験結果，評価および考察

本章では，実施した実験に基づき，多目的遺伝的アルゴリズムがプロジェクトマネジメントにおけるタスク割当問題に与える影響を詳細に分析し，評価する．本章では，実験結果の概要から始め，各目的関数の詳細な分析，制約条件違反の分析，目的関数間のトレードオフ，そして重みの組合せが結果に及ぼす影響を考察する．さらに，得られた結果からプロジェクトマネジメントにおける意思決定支援としての遺伝的アルゴリズムの利点と制限を明らかにし，今後の研究方向について展開する．全体を通じて，アルゴリズムの実用性の検証と，プロジェクトマネジメントにおけるその応用可能性を深掘りすることを目的とする．

6.1 実験結果の概要

本研究では，メンバーのタスク割当問題に対する多目的遺伝的アルゴリズムの有効性を評価するため，72 の異なるパラメータ組合せで実験を行った．これらの実験は，コストの最小化と労働日数の均一化のバランスを取りつつ，制約条件を満たす解を見つける能力が評価の焦点であった．この実験では，遺伝的アルゴリズムに関連するパラメータを以下の表 1 の通り固定している．予備実験の結果，これらのパラメータの変更が結果に顕著な影響を与えないことが分かったためである．

集団サイズ	100
世代数	300
交叉確率	1.0
個体突然変異確率	1.0
遺伝子座突然変異確率	0.1
交叉パターン	一様交叉
突然変異タイプ	インデックスシャッフル突然変異

表 1 遺伝的アルゴリズムのパラメータ

一方，制約条件 C1, C2, C3 に対する重み付けと目的関数 2 のスケーリング係数を以下の

表 2 の通り設定した．これらを変化させることで，各制約条件の重要性和目的関数間のスケールを調整することが可能となり，アルゴリズムがこれらの変化にどう対応するかを評価することができる．また，異なるスケーリング係数によって各目的関数の相対的な重要性がどのように変わるかを観察し，最終的な解の品質にどのような影響を与えるかを検証した．この実験設計により，アルゴリズムが実際のプロジェクトマネジメントのシナリオにおいて柔軟に対応できるかどうかを理解するための基盤を提供することを意図した．実験における制約条件の重みと目的関数 2 のスケーリング係数の組合せは以下の通りである．

実験番号	C1 重み	C2 重み	C3 重み	スケーリング係数
1	1	10	100	3000
2	1	10	100	4000
3	1	10	100	5000
4	10	100	1000	3000
5	10	100	1000	4000
6	10	100	1000	5000
7	10	200	500	3000
8	10	200	500	4000
9	10	200	500	5000
10	10	20	30	3000
11	10	20	30	4000
12	10	20	30	5000
13	1	100	10	3000
14	1	100	10	4000
15	1	100	10	5000
16	10	1000	100	3000
17	10	1000	100	4000
18	10	1000	100	5000
19	10	500	200	3000
20	10	500	200	4000

21	10	500	200	5000
22	10	30	20	3000
23	10	30	20	4000
24	10	30	20	5000
25	10	1	100	3000
26	10	1	100	4000
27	10	1	100	5000
28	100	10	1000	3000
29	100	10	1000	4000
30	100	10	1000	5000
31	200	10	500	3000
32	200	10	500	4000
33	200	10	500	5000
34	20	10	30	3000
35	20	10	30	4000
36	20	10	30	5000
37	100	1	10	3000
38	100	1	10	4000
39	100	1	10	5000
40	1000	10	100	3000
41	1000	10	100	4000
42	1000	10	100	5000
43	500	10	200	3000
44	500	10	200	4000
45	500	10	200	5000
46	30	10	20	3000
47	30	10	20	4000
48	30	10	20	5000
49	10	100	1	3000

50	10	100	1	4000
51	10	100	1	5000
52	100	1000	10	3000
53	100	1000	10	4000
54	100	1000	10	5000
55	200	500	10	3000
56	200	500	10	4000
57	200	500	10	5000
58	20	30	10	3000
59	20	30	10	4000
60	20	30	10	5000
61	100	10	1	3000
62	100	10	1	4000
63	100	10	1	5000
64	1000	100	10	3000
65	1000	100	10	4000
66	1000	100	10	5000
67	500	200	10	3000
68	500	200	10	4000
69	500	200	10	5000
70	30	20	10	3000
71	30	20	10	4000
72	30	20	10	5000

表 2 制約条件 C1, C2, C3 の重みと目的関数 2 のスケーリング係数の組合せ

各実験結果の説明では、この表 2 の組合せ表に記載された実験番号に言及し、どの組合せに対応するかを示す。これにより、特定の実験条件下でのアルゴリズムの挙動と成果を具体的に理解することができる。

図 5 と図 6 は、実験ごとの目的関数の進化を集団サイズ 100 の平均値として示す折れ線

グラフである。各折れ線は特定の実験番号の重みの組合せに対応し、横軸の世代数に沿って目的関数の値がどのように変化するかを表している。

図 5 には、異なる重みの組合せの下で実施された実験ごと（実験番号：1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11）の目的関数 1（コストの最小化）の進化の平均が示されている。これは、アルゴリズムがどのように総コストを削減する方向に進化していったかを示しており、どの重みの組合せを用いても、目的関数 1 は平均的に小さくなる傾向にあったことを表している。

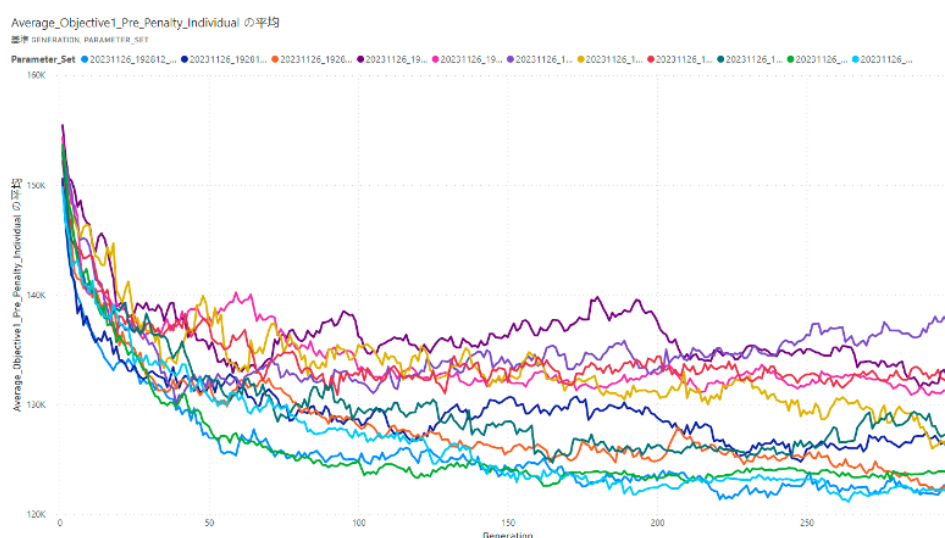


図 5 目的関数 1 の進化の世代ごとの平均を示すグラフ
(実験番号：1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11)

図 6 は、同様に実験ごと（実験番号：1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11）の目的関数 2（労働日数の標準偏差の最小化）の進化を示しており、アルゴリズムが労働日数のバランスをどのように取りながら適応していったかが視覚化されている。初見では目的関数 2 の進化が目立たないように見えるが、後述の実験結果の詳細分析では、制約条件の重みによって異なる傾向が確認されている。



図 6 目的関数 2 の進化の世代ごとの平均を示すグラフ
(実験番号 : 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11)

この結果から、アルゴリズムのパフォーマンスと解の進化に対する重みの組合せの影響を明確に理解するためには、複数の指標による総合的な分析が必要であることが示唆されている。それぞれの折れ線が特定の重みの組合せにおけるアルゴリズムの振る舞いを反映しており、総合的な評価にはこれらの結果を総合的に考慮することが重要である。

6.2 実験結果の詳細分析

目的関数 1 (コスト最小化) の詳細分析

実験結果における目的関数 1 (コスト最小化) の進化パターンと、制約条件 C1, C2, C3 への異なる重み付けがそのパフォーマンスに与える影響について詳細に分析する。

図 8 (実験番号 : 16, 17, 18) により制約条件 C2 (タスクの予算内での割当) に重きを置くと、目的関数 1 への効果が強化される傾向にあることが観察された。これは、予算制約を重視することでコスト削減を最優先の目標とするアルゴリズムの挙動が反映されている。一方、図 9 (実験番号 : 4, 5, 6) により制約条件 C3 (スキルマッチの必要性) に重きを置くと、目的関数 1 の値の振れ幅が大きくなる。これは、スキルの適合性を高く評価することで、コスト効率よりも適切なスキルセットを持つメンバーの割当が優先されているためである。また、図 7 (実験番号 : 64, 65, 66) により制約条件 C1 (メンバーの排他的タスク割

当)に重きを置いた場合は,C2 と C3 の中間の効果が見られ,コスト削減は一定の効果を保ちつつ,リソースの過剰割当を防ぐバランスが取られた.

データセットの特性によっても結果は影響を受けるが,全体としてはどの制約条件に重きを置いても目的関数 1 は有効に機能する.ただし,目的関数に対する効果の強弱には明らかな差が存在した.図 7, 図 8, および図 9 において,各色の折れ線は特定の実験番号に対応しており,アルゴリズムがそれぞれの制約にどのように反応し,適応していくかを視覚的に比較できるようになっている.ここでいう「強い重み」とは,その制約条件を満たさない際に適用されるペナルティの量が相対的に高い設定を指しており,制約違反が最終的な適応度に与える影響が増大することを意味する.具体的な重みの数値としては,他の制約条件の重みに比べて高い値を持つものがこれに該当する.

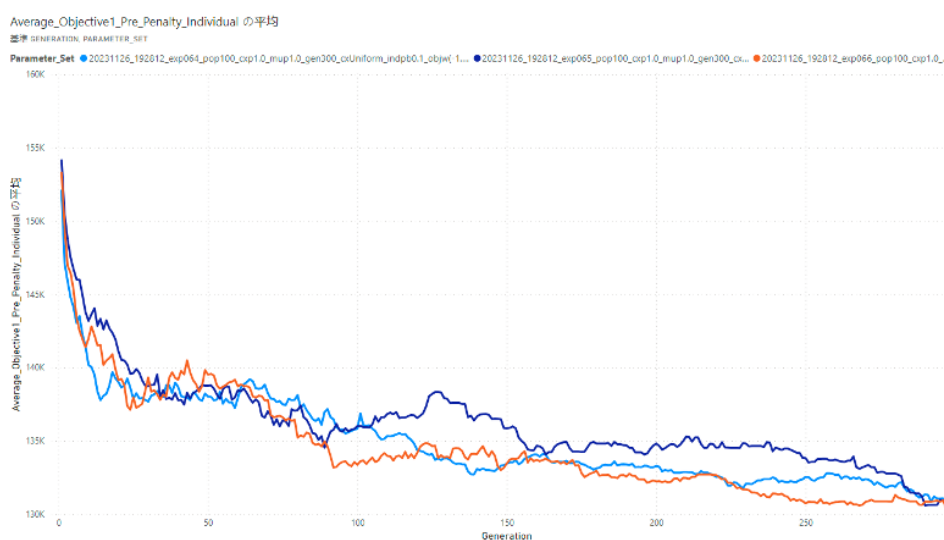


図 7 制約条件 C1 に強い重みを設定した際の目的関数 1 の進化の平均
(実験番号 : 64, 65, 66)

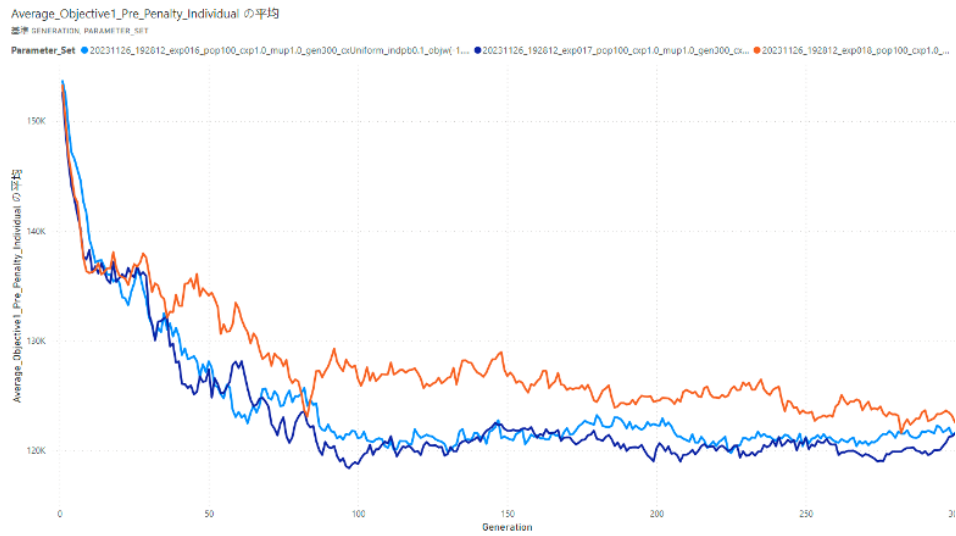


図 8 制約条件 C2 に強い重みを設定した際の目的関数 1 の進化の平均
(実験番号 : 16, 17, 18)

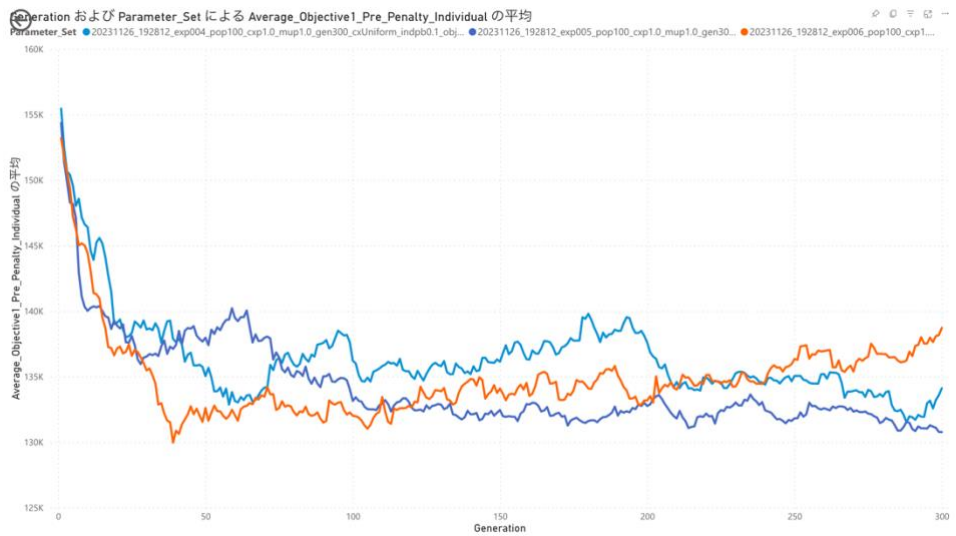


図 9 制約条件 C3 に強い重みを設定した際の目的関数 1 の進化の平均
(実験番号 : 4, 5, 6)

これらの観察から、プロジェクトマネジメント実践においてアルゴリズムのパフォーマンスを最適化するためには、プロジェクトの特定のニーズに合わせて制約条件の重み付けを適切に調整することが鍵であることがわかる。また、制約条件の強弱は、目的関数 1 の達成度に直接的な影響を与え、プロジェクトのコスト効率と質を同時に向上させるための

重要な調整ポイントであると言える。

目的関数 2（労働日数の標準偏差最小化）の詳細分析

目的関数 2 の進化を通じて、メンバー間の労働負担の均等化がどのように達成されているかを検証した。異なる制約条件の重み付けが目的関数 2 に与える影響を評価することで、メンバーの労働負担を適切に配分することを目指される。

図 10（実験番号：40, 41, 42）により制約条件 C1（メンバーの排他的タスク割当）に強い重みを置くことで、労働日数の標準偏差の縮小が促進される一方で、目的関数 1（コスト削減）にも適度に影響を及ぼす。図 11（実験番号：52, 53, 54）により C2（タスクの予算内での割当）に重きを置いた場合、目的関数 2 の改善は限定的であることが観察され、図 12（実験番号：28, 29, 30）により C3（スキルマッチの必要性）に重点を置いた場合には、目的関数 2 の振れ幅が大きくなることが確認された。

図 10, 図 11, および図 12 は、制約条件の重み付けが目的関数 2（労働日数の標準偏差の最小化）の進化に与える影響を示している。これらの図における各折れ線は、特定の制約条件に対して設定された重みの強さに基づいており、プロジェクトの具体的な要件に応じた制約条件の調整が目的関数 2 の最適化にどのように作用するかを視覚的に表している。

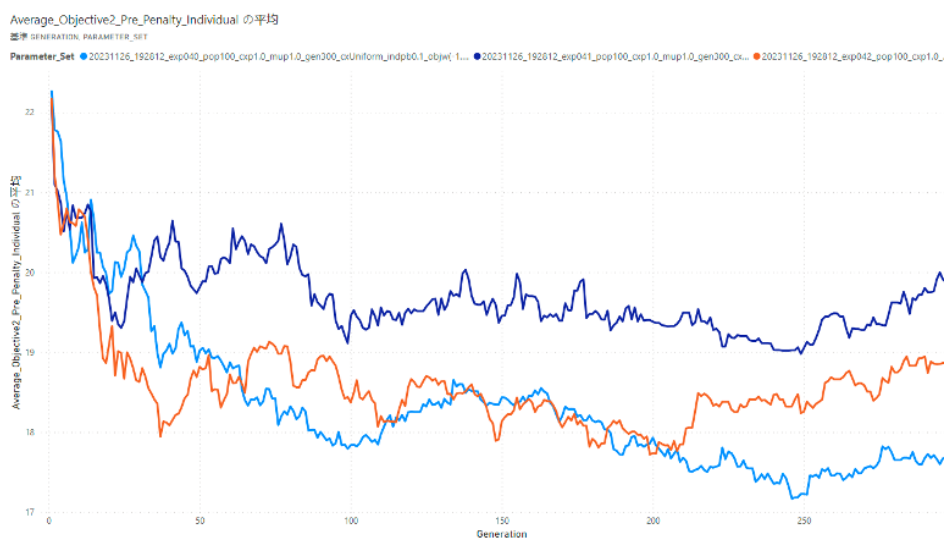


図 10 制約条件 C1 に強い重みを設定した際の目的関数 2 の進化の平均
(実験番号：40, 41, 42)

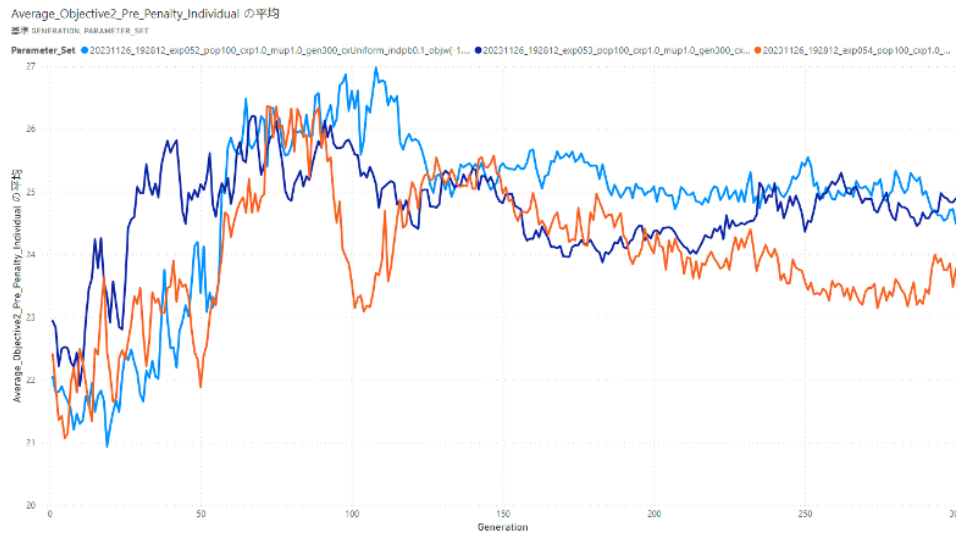


図 11 制約条件 C2 に強い重みを設定した際の目的関数 2 の進化の平均
(実験番号 : 52, 53, 54)

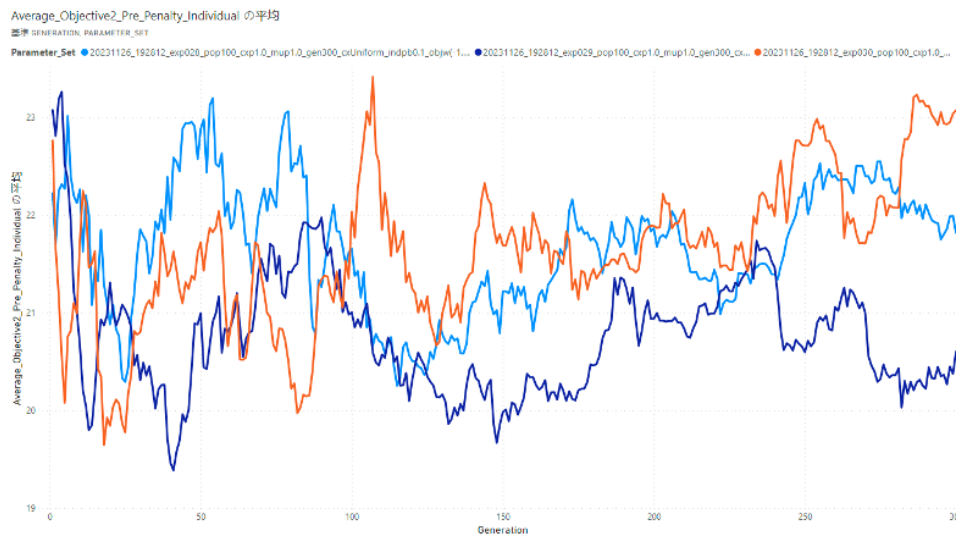


図 12 制約条件 C3 に強い重みを設定した際の目的関数 2 の進化の平均
(実験番号 : 28, 29, 30)

これらの結果は、目的関数 1 と 2 の間で求められるバランスの微妙さを示しており、制約条件の重み付けがタスク割当に影響を与えることを示している。目的関数 2 への影響を最適化しつつ、コスト効率も維持するための制約条件の調整がプロジェクトマネジメントにおける重要な課題であることが示されている。

6.3 目的関数間のトレードオフ

本節では、多目的最適化問題における二つの目的関数—コスト最小化を目指す目的関数 1 と労働日数の標準偏差最小化を目指す目的関数 2—の間に存在する相互作用とトレードオフについて考察する。目的関数 1 と目的関数 2 は、しばしば相反する要求を示す。コストを削減する努力は労働の偏差を増加させる可能性があり、逆もまた真である。この複雑な相互作用を理解することは、プロジェクトマネジメントにおける意思決定において不可欠である。

目的関数 1 の改善と目的関数 2 への影響

図 13（実験番号：1）により目的関数 1 のパフォーマンスが改善するケースでは、図 14（実験番号：1）により目的関数 2 の効果が薄れることが観察される。一方、図 15（実験番号：6）により目的関数 1 のパフォーマンスが低い状態では、図 16（実験番号：6）により目的関数 2 は有効に機能し、労働日数の偏差を最小限に抑えることができている。

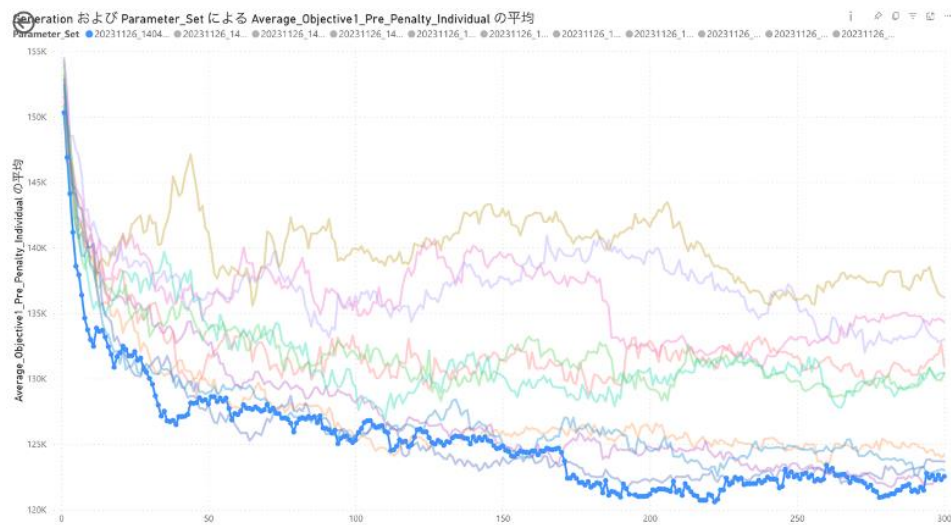


図 13 目的関数 1 の平均パフォーマンス
(実験番号：1)

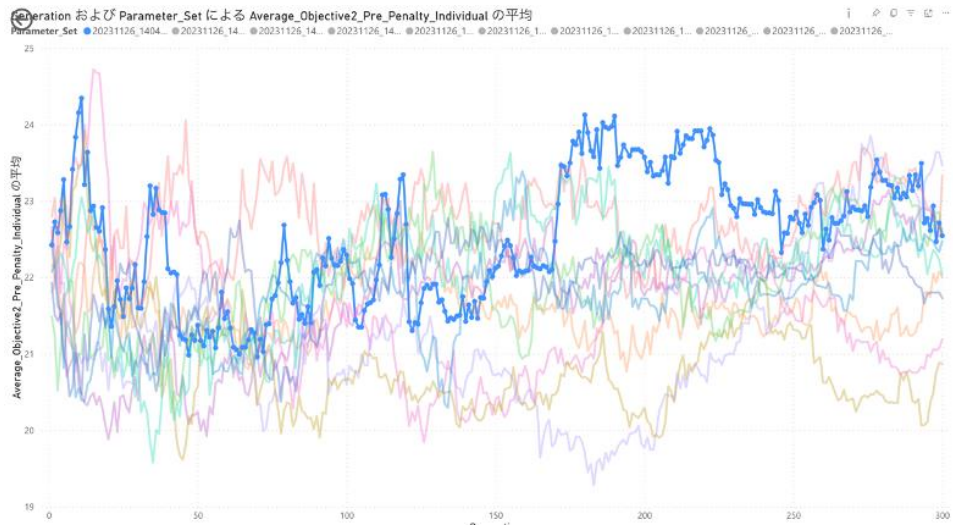


図 14 目的関数 2 の平均パフォーマンス

(実験番号 : 1)

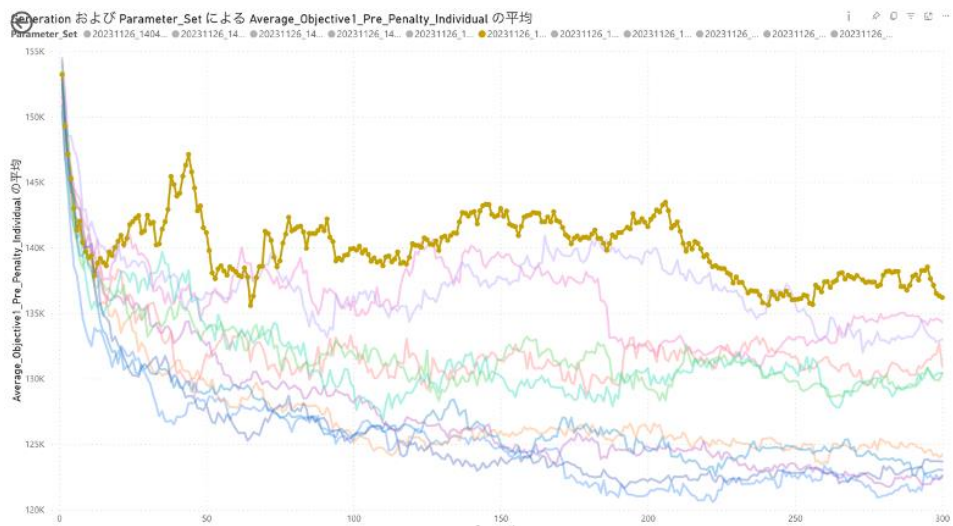


図 15 目的関数 1 の平均パフォーマンス

(実験番号 : 6)

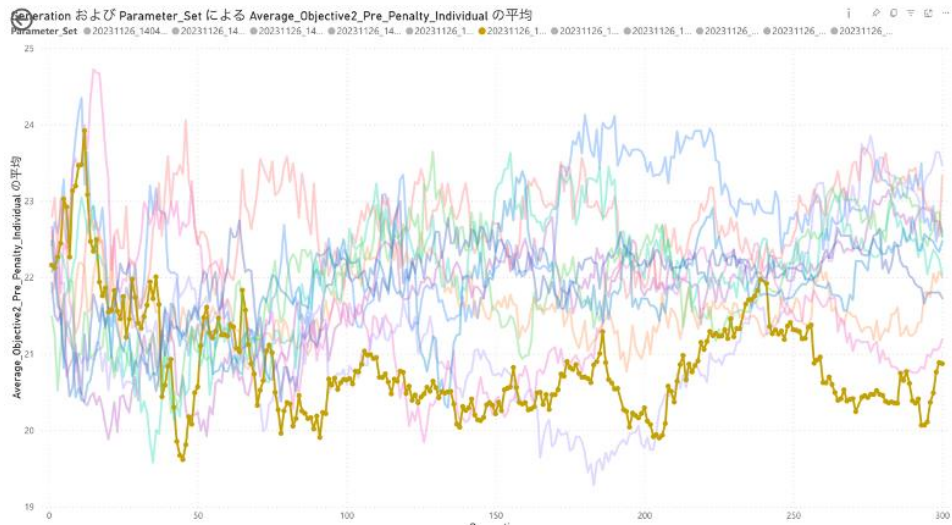


図 16 目的関数 2 の平均パフォーマンス

(実験番号 : 6)

目的関数間のトレードオフのバランス

最も望ましいのは、両目的関数のバランスが良い状態である。このような状況は図 17 (実験番号 : 4) および図 18 (実験番号 : 4) により、適切な重みの組合せによってのみ達成可能であり、プロジェクトの目標と実際の要件に合わせた微妙な調整が求められる。目的関数 1 と 2 が均等に最適化される状況は、プロジェクトのコスト効率とメンバーの労働バランスの両方を同時に向上させる。

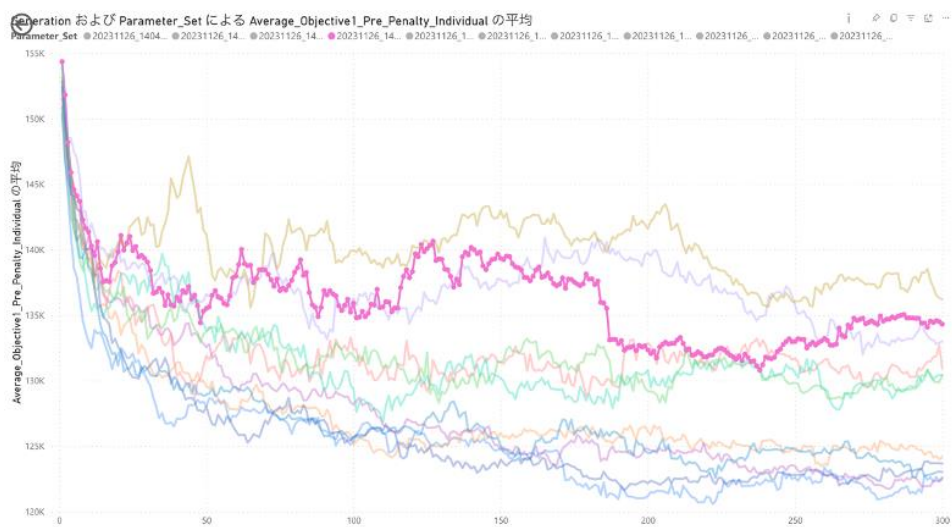


図 17 目的関 1 の平均パフォーマンス

(実験番号 : 4)

2 のスケーリング係数が大きくなるほど、図 19（実験番号：70）、図 20（実験番号：71）および図 21（実験番号：72）により目的関数 2 の探索範囲が広がる傾向が確認された。

図 19、図 20、図 21 では、制約条件の重みを $C1=30$, $C2=20$, $C3=10$ と固定し、スケーリング係数をそれぞれ 3000, 4000, 5000 に設定した状態でのパレートフロントが示されている。これらの図において、縦軸は目的関数 2 の値を、横軸は目的関数 1 の値を示しており、色の濃さは世代数を表している。

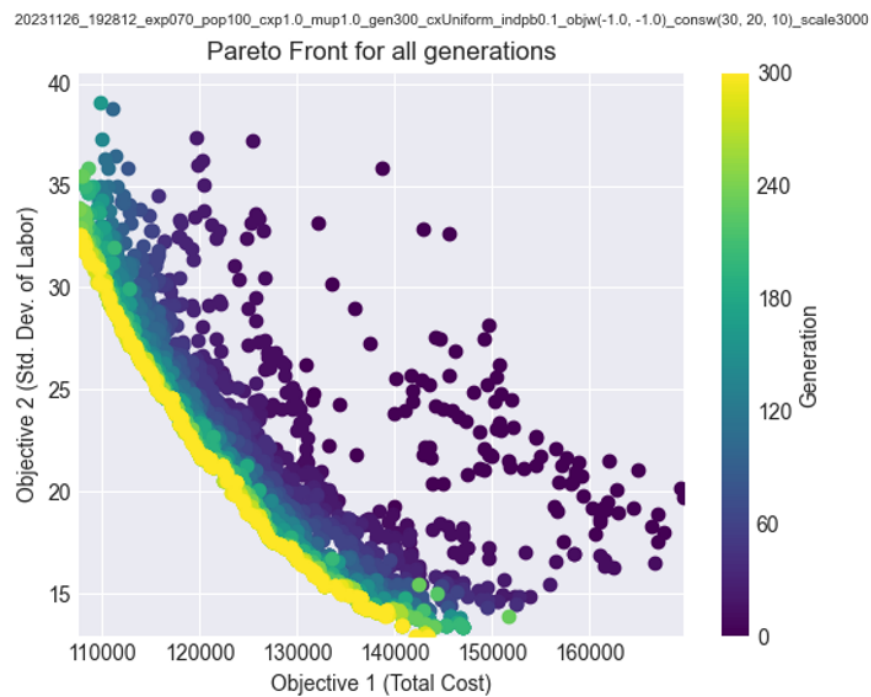


図 19 重みの組合せ： $C1=30$, $C2=20$, $C3=10$, スケーリング係数=3000
(実験番号：70)

20231126_192812_exp071_pop100_cxp1.0_mup1.0_gen300_cxUniform_indpb0.1_objw(-1.0, -1.0)_consw(30, 20, 10)_scale4000

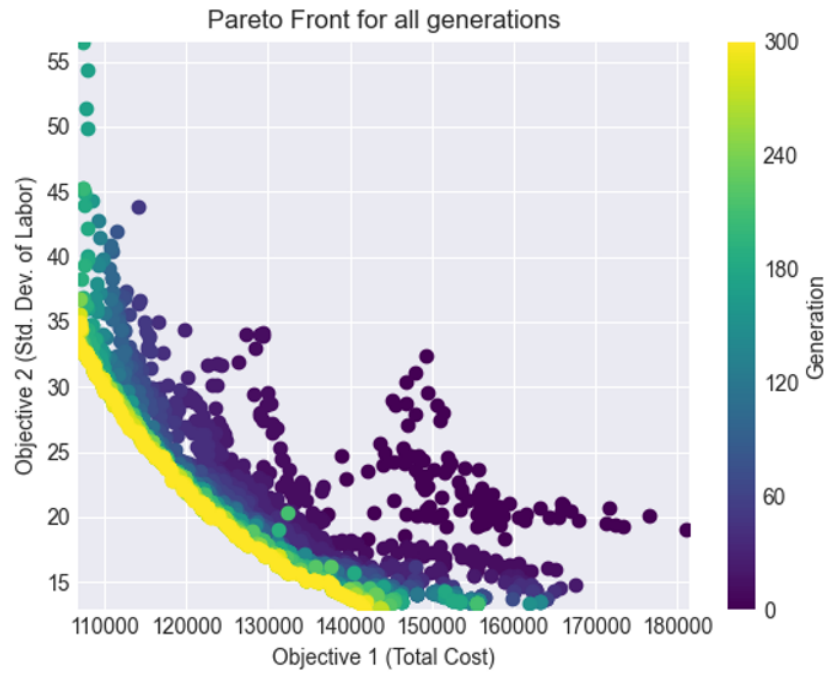


図 20 重みの組合せ : $C1=30, C2=20, C3=10$, スケーリング係数=4000
(実験番号 : 71)

20231126_192812_exp072_pop100_cxp1.0_mup1.0_gen300_cxUniform_indpb0.1_objw(-1.0, -1.0)_consw(30, 20, 10)_scale5000

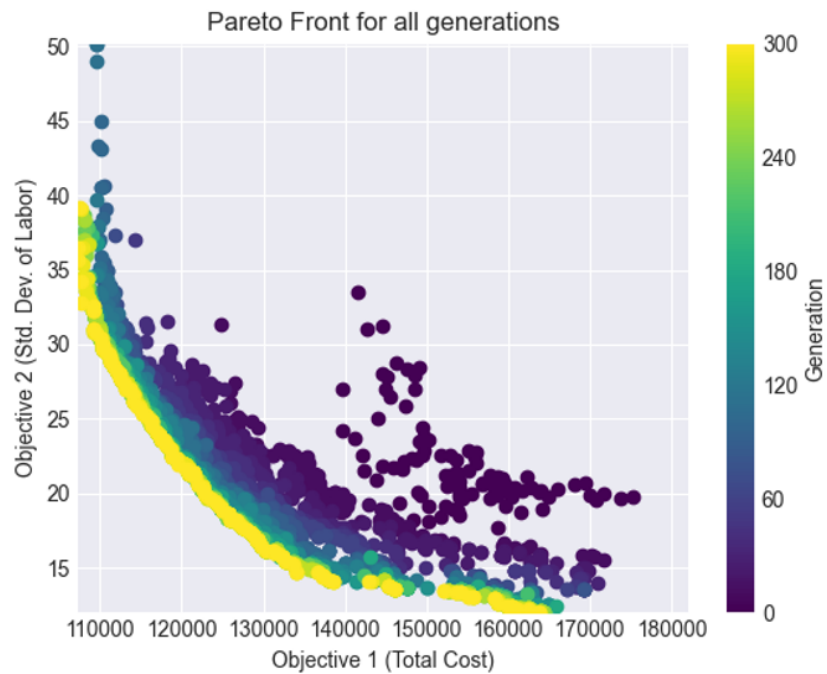


図 21 重みの組合せ : $C1=30, C2=20, C3=10$, スケーリング係数=5000
(実験番号 : 72)

図 19, 図 20, および図 21 のパレートフロントは, 目的関数間のスケールを変化させることによるトレードオフを示している. 目的関数間のスケールの調整は実験の重要な部分であり, データセットによってスケーリング係数を変えていく必要がある. 実際の運用では, 最適化アルゴリズムを実施する前に各目的関数のとり得る最大値を検討し, その上でスケーリング係数を動的に設定するのが望ましい.

各制約条件の重みの影響

多目的最適化問題において, 制約条件の重みは解探索の挙動において重要な役割を果たす. スケーリング係数を一律に 3000 と固定し, 制約条件 C1, C2, C3 の異なる重み付けが遺伝的アルゴリズムを通じて生成されるパレート最適解集合にどのような影響を与えるかを検証した. 図 22 (実験番号: 4), 図 23 (実験番号: 64), 図 24 (実験番号: 52) は, それぞれ異なる重み付けを施した C1, C2, C3 の制約条件がパレート最適解集合に及ぼす影響を示している.

これらの図により, 特定の制約に重きを置いた場合のアルゴリズムの探索挙動と解の質の変化を確認できる. これらの図は, アルゴリズムがどのように探索過程を進め, 解の質をどのように変えていくかを示している. 特定の制約条件への重み付けが最適化過程においてどのような効果を持つかが, これらの図によって明確になる.

図 22 において, $C1=10$, $C2=100$, $C3=1000$ という重み付けを行った結果, 目的関数 2 に関しては労働日数の標準偏差が小さい解が多く, 目的関数 1 においては総コストが低い解に収束していることが観察される. これは, $C3$ に大きな重みを置くことでスキルマッチングを優先し, 労働の均等性を高める傾向にあり, $C1$ および $C2$ の影響が相対的に抑えられていることが示されている.

図 23 では, $C1=1000$, $C2=100$, $C3=10$ という重みを設定した結果, 目的関数 1 (コストの最小化) に関して低コストの解が多く, 目的関数 2 (労働日数の標準偏差の最小化) では標準偏差が大きい解が出現している. これは, $C1$ の重みが大きい場合, コストの削減はより達成されやすいが, 労働日数の均等分配はあまり考慮されない結果になることを示している.

図 24 では, $C1=100$, $C2=1000$, $C3=10$ に設定されている. この条件下では, 目的関数 1 (コストの最小化) に関してはコスト削減が効果的に行われているが, 目的関数 2 (労働日数の標準偏差の最小化) の結果としては労働日数の偏差が大きい解が確認される. $C2$ に最も大きな重みを置いたことにより, コストに関する制約が解の選択に強い影響を与え, その結

果として労働日数のバランスが取れにくくなっていることが示されている。

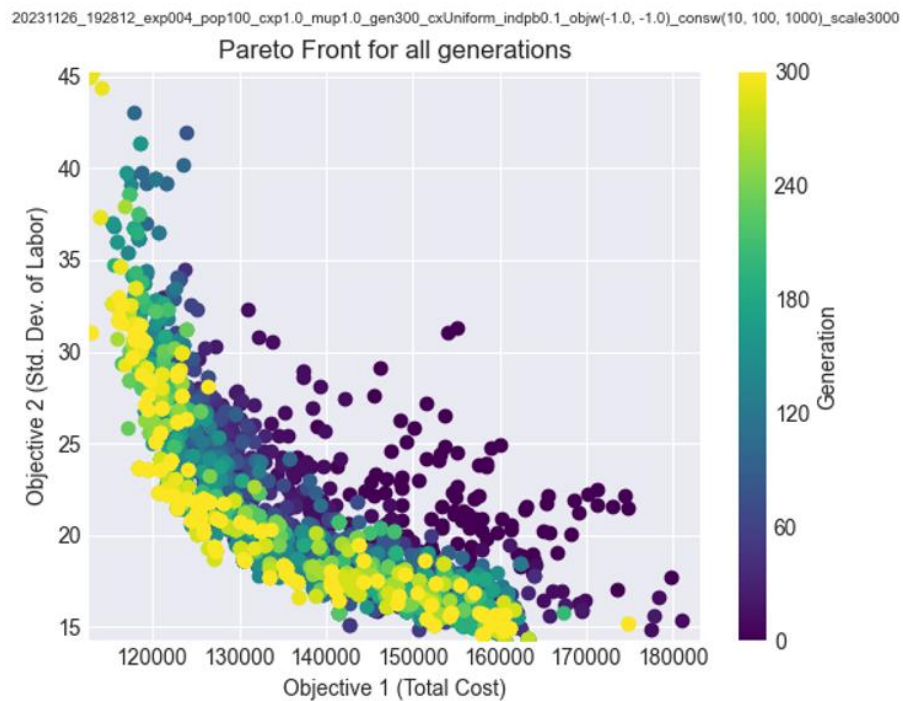


図 22 重みの組合せ：C1=10, C2=100, C3=1000, スケーリング係数=3000
(実験番号：4)

20231126_192812_exp064_pop100_cxp1.0_mup1.0_gen300_cxUniform_indpb0.1_objw(-1.0, -1.0)_consw(1000, 100, 10)_scale3000

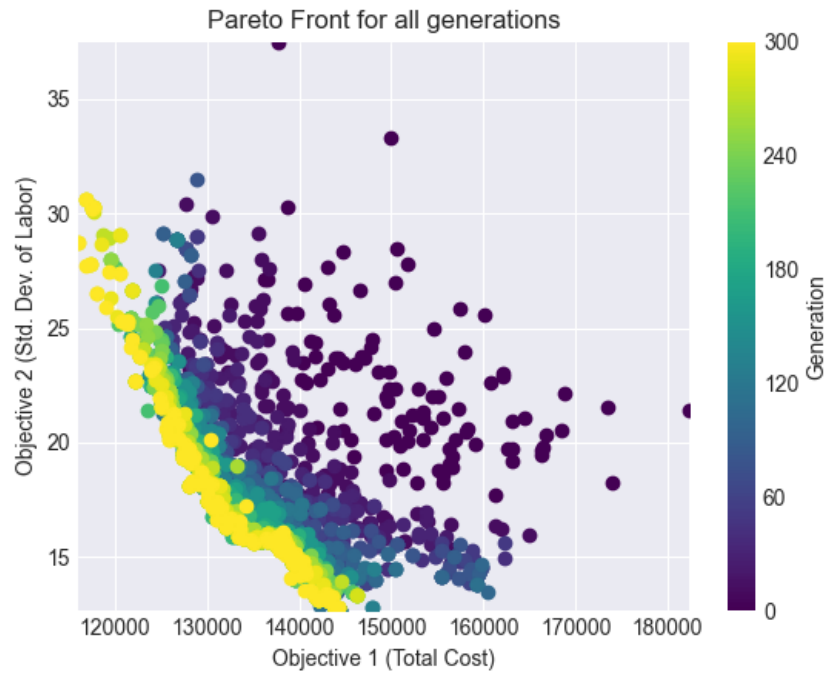


図 23 重みの組合せ : $C1=1000$, $C2=100$, $C3=10$, スケーリング係数=3000
(実験番号 : 64)

20231126_192812_exp052_pop100_cxp1.0_mup1.0_gen300_cxUniform_indpb0.1_objw(-1.0, -1.0)_consw(100, 1000, 10)_scale3000

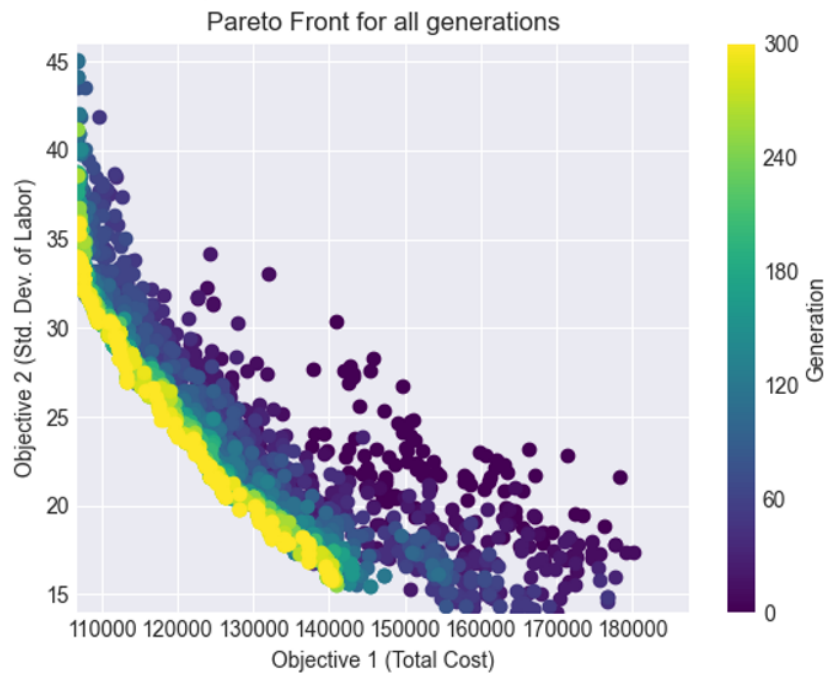


図 24 重みの組合せ : $C1=100$, $C2=1000$, $C3=10$, スケーリング係数=3000
(実験番号 : 52)

これらの図から得られる知見は、プロジェクトの要求に即した制約条件のバランスをとることの重要性を明確にし、それに応じたパラメータ設定の調整がプロジェクトの目標達成に必要であることを示している。適切な重み付けにより、目的関数 1 と目的関数 2 の望ましいトレードオフが実現され、プロジェクト全体の成果に貢献することが期待される。

制約条件の重み付けは目的関数 1 と目的関数 2 の結果に直接作用し、最適解の質に差を生じさせる。目的関数間スケールの調整が解の探索範囲と多様性に与える効果は大きく、望ましいトレードオフを達成するためには、これらのパラメータを慎重に調整する必要がある。

6.5 最適化結果のガントチャートによる妥当性の評価

本節では、遺伝的アルゴリズムによって導出された最適化結果の妥当性を、ガントチャートとして表現した形で評価する。目的関数 1（総コストの最小化）と目的関数 2（労働日数の標準偏差の最小化）の間で良好なバランスを示したパラメータセットを分析のための代表例として選択した。具体的には、パラメータセット $C1=10$, $C2=100$, $C3=1000$, スケーリング係数=5000 に基づくパレートフロントが図 25（実験番号：6）に示されており、目的関数間の効果的なトレードオフを視覚的に表している。

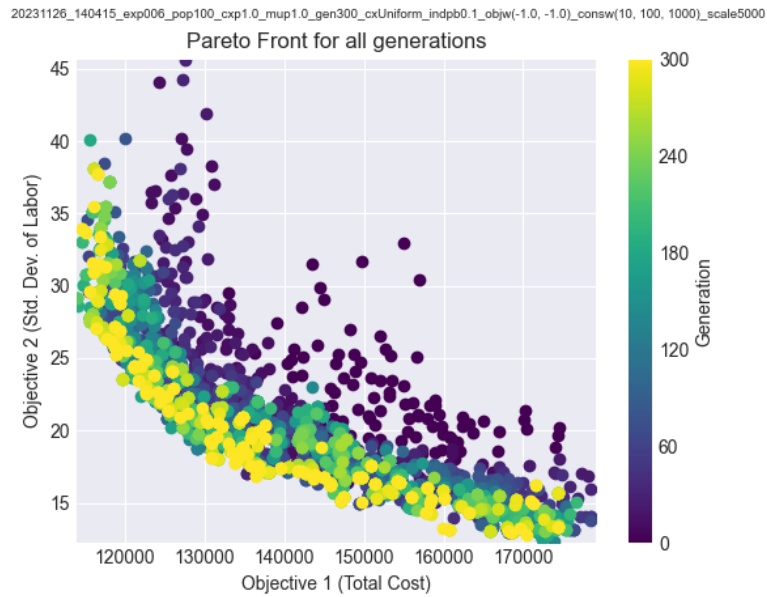


図 25 重みの組合せ：C1=10, C2=100, C3=1000, スケーリング係数=5000
(実験番号：6)

図 26 は、図 25 で示された最終世代のパレートフロントからの解抽出を、ガントチャートで具体的に表現した概念図である。この表現により、個々の解が示すタスクの時系列的流れと割当状況が視覚化され、プロジェクトマネジメントの意思決定に役立つ形に具体化される。これを通じて、遺伝的アルゴリズムによる理論的解が実際のプロジェクトスケジュールにどう反映するかが視覚的に評価できるようになる。

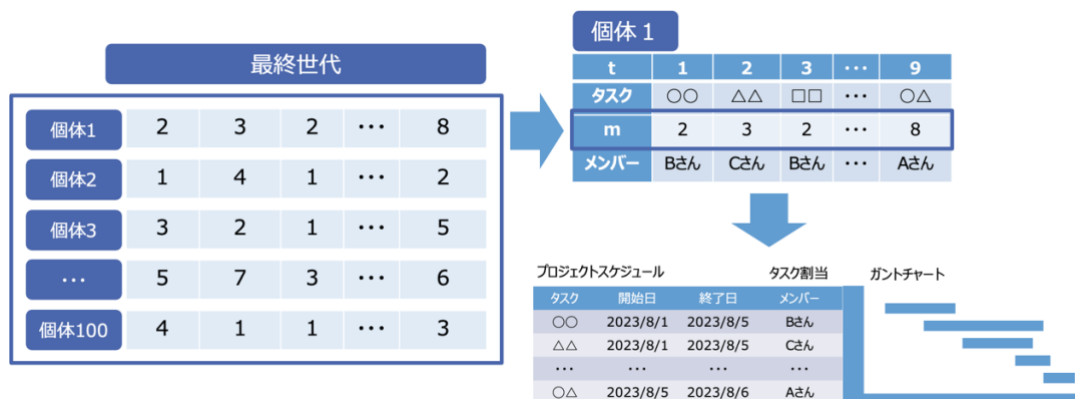


図 26 個体のガントチャートへの変換

図 27 は、図 25 で示された最終世代のパレートフロントから選ばれた 1 つの解、すなわ

6.6 今後の研究方向

本研究の適用によって、多目的遺伝的アルゴリズムがプロジェクトマネジメントの課題解決に寄与する可能性が示されたが、さらなる探求が求められる。今後の研究では、実用的なフレームワーク構築に向け、以下の方向性を追求する。

人材育成指標の統合

タスク割当だけではなく、人材の能力開発を促進する指標を最適化モデルに組み込む。このアプローチにより、組織全体の持続的な成長を目指す。

多目的最適化手法と大規模言語モデルの統合

大規模言語モデルを駆使した自然言語処理の技術の進化は、プロジェクトマネージャーが自然言語を用いて最適化の目的関数や制約条件を含めた複数のパラメータや重みの組合せを直感的に調整することを可能にする。この新たな枠組みでは、大規模言語モデルがプロジェクトマネージャーからの自然言語による指示を解析し、それに基づいて最適化の目的関数や制約条件を動的に調整する。プロジェクトマネージャーは、プロジェクトの現実的な要求や意思決定をシステムに直接伝えることができる。この対話主導型のプロセスは、プロジェクトの複雑さやダイナミックな変化に柔軟に適応し、最適化の精度と効果を高めることが期待される。

フィードバックによるアルゴリズムの改善

プロジェクト運営の実地から得られるフィードバックを大規模言語モデルに組み込み、システムの学習プロセスを通じてその適応性と提案品質を持続的に向上させる。このアプローチにより、現場の経験と知見をシステムに反映させ、プロジェクトマネジメントの具体的な実務において有益な改善を実現する。

これらの方向性を踏まえ、プロジェクトマネジメントの実践に即した最適化アプローチの開発を進める。意思決定の迅速化、リソース配分の効率化、人材育成の戦略化を実現し、プロジェクトの成果向上に貢献することを目的とする。これにより、プロジェクト管理の課題に対する具体的かつ実践的な解決策を提供する研究成果が期待される。

第 7 章 結論

本研究は、プロジェクトマネジメントの文脈におけるタスク割当問題に対する多目的最適化アプローチに注目し、その有効性と実践的な応用可能性を探究した。複数の目的関数を持つこの最適化問題には、コストの最小化と労働日数の標準偏差の最小化という二つの目的が含まれており、それぞれの目的間でのトレードオフが考慮された。

実験では、遺伝的アルゴリズムとその重みの組合せが、最適化プロセスに及ぼす影響を詳細に分析した。その結果、目的関数 1 と目的関数 2 の間に明確なトレードオフが存在すること、さらに、重みの組合せによっては、これらのトレードオフを適切にバランスさせることが可能であることが示された。また、様々な重みの組合せに基づくパレートフロントの分析を通じて、それぞれの重みが最適化結果に与える影響を視覚的に捉えることができた。

本研究のアプローチは、プロジェクトマネジメントにおけるタスク割当問題への新しい解法を提案することにより、その理論的な枠組みとしての価値を示している。実験に用いられたデータセットは、実際の業務データではなく、研究目的で作成されたものであるため、実務への直接的な適用性については確証がない。この点は本研究の一つの限界として認識しており、実際のプロジェクト環境における手法の有用性を検証するためには、実証実験を行う必要がある。

最終的に、本研究で開発された多目的最適化手法が、プロジェクトマネジメントの実務におけるタスク割当において、より効率的かつ適切な意思決定を支援する可能性については、今後の実証的な検証を通じて明らかにしていく必要がある。この研究の理論的な成果が、将来のプロジェクトマネジメントの効率化と質の向上に向けた一歩となることを願っている。

参考文献

- [1] 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 7 版＋プロジェクトマネジメント標準”, p.33, 2023.
- [2] 高津諭, 吉田寛, 坂本昌史, 柴田朋子, “人員資源割当におけるヒトの意図反映手法の確立に向けた数理モデル開発と検証”, FIT2020(第 19 回情報科学技術フォーラム), pp.63-66, 2020.
- [3] 高須賀将秀, 吉田寛, “個人の能力差を考慮した開発における人員資源割当問題に対する解法の提案”, 情報処理学会研究報告(数理モデル化と問題解決(MPS)), Vol.2017-MPS-112, No.26, pp.1-4, 2017.
- [4] 小林敬明, 森口聡子, “多目的遺伝的アルゴリズムによる IT プロジェクトスケジューリング”, 情報処理学会論文誌トランザクション 数理モデル化と応用(Web), vol. 11, no. 3, pp. 42-57, 2018.
- [5] Kalyanmoy Deb, Multi-Objective Optimization using Evolutionary Algorithms, Wiley, 2001.

謝辞

本研究を進めるにあたり、絶え間ないご指導と深い洞察を賜った近藤伸彦先生に心からの感謝を表す。近藤先生の豊かな知識と熱心な指導は、研究の全過程で不可欠な支えであった。副査を務めていただいた森口聡子先生にも感謝を申し上げる。

勉強と仕事の両立を支えてくれた家族にも深い感謝の意を表す。家族の絶え間ない支援と理解があったからこそ、研究を継続することができたのである。

職場での理解と協力にも感謝する。仕事と大学院の研究のバランスを取ることの難しさに対し、柔軟に対応してくれた会社と上司のおかげで、研究を続けることが可能であった。

最後に、私の研究活動を陰ながら支えてくださった全ての方々に感謝の意を表す。皆様のご協力があつてこそ、本研究を無事に終えることができたのである。