

令和4年度

修士論文

動的スケジューリング問題のための
並列分散構造GPの開発とルール分析

先進理工系科学専攻 社会情報学研究室
M214041 前田 伊吹

指導教員

教授 西崎 一郎
准教授 林田 智弘
助教 関崎 真也

令和5年2月21日

広島大学大学院先進理工系科学研究科

あらかし

一般的に、不規則に到着する新しいジョブ群の処理順序を考える動的スケジューリング問題に対して、優先規則によってジョブ毎の優先度を計算し、それを基に順序を決定する方法が用いられる。優先規則としてディスパッチングルールを使用することが多いが、ジョブの特徴によって最適なルールが異なるため、全てのジョブ群に特定のルールを使用するのではなく、新たなジョブ群に適したルールを試行錯誤的に生成する必要がある。従来研究では、ニューラルネットワーク (NN:Neural Networks) や遺伝的アルゴリズム (GA:Genetic Algorithms) を用いてスケジューリングを行っていたが、事前学習に必要となるデータや、新たなジョブ群の到着の度に再学習をするための計算コストが多くかかるため、ディスパッチングルールの獲得を目的とした遺伝的プログラミング (GP:Genetic Programming) を用いた最適化手法が提案されている。本論文では GP を用いた手法の効率化を目的として、ディスパッチングルールを保有する複数の個体からなる個体群を複数用意し、到着したジョブ群の特徴に応じて遺伝的操作の対象となる個体群を選択する手法をいくつか構築し、比較する。さらに、GP によって生成したディスパッチングルールの特徴や、一般的に使用されているディスパッチングルールとの相違についての分析を行う。

目次

1.	はじめに	1
2.	従来研究	3
2.1.	動的スケジューリング問題	3
2.2.	ディスパッチングルール	4
2.3.	進化計算手法	5
2.3.1.	GA, GP	5
2.3.2.	遺伝的操作	6
2.3.3.	bloat 現象	8
2.3.4.	GP を用いたスケジューリング	8
2.3.5.	エリート保存則	9
2.3.6.	島モデル	10
3.	提案手法	12
3.1.	個体群選択型 GP	12
3.1.1.	概要	12
3.1.2.	個体群選択型 GP の適用方法	13
3.2.	個体群選択型 IGP	14
3.3.	個体群増減型二重並列構造 GP	14
3.3.1.	概要	14
3.3.2.	個体群増減型二重並列構造 GP の手順	15
4.	数値実験	19
4.1.	実験概要	19
4.2.	結果と考察	21
5.	ルール分析	28
5.1.	分析方法	28
5.2.	分析結果	29
5.2.1.	生成した DR と既存 DR の類似度	29
5.2.2.	既存 DR にはない特徴を持つ DR の分析	32
5.2.3.	考察	35
6.	おわりに	36

謝辭	37
参考文献	38

第 1 章

はじめに

複数のジョブが不規則に到着する動的スケジューリング問題を解くために、一般的にはディスパッチングルール (DR:Dispatching Rule) などの優先規則を用いてスケジューリングする手法が実用的な手法として用いられている [1]. 動的スケジューリング問題に対して優先規則を使用するメリットとして、低い計算コストで再スケジューリングができることが挙げられる. しかし、到着するジョブ群の特徴によって最適な優先規則が異なることが多くの研究によって明らかになっており、様々な特徴を持つジョブ群が到着する動的スケジューリング問題に特定の優先規則だけでは必ずしも優れたスケジュールを組むことができるとは限らない. そのため、機械学習や進化計算手法を用いて最適な優先規則を試行錯誤的に生成する手法が提案されている [2][3].

江口ら [2] は、ニューラルネットワーク (NN:Neural Networks) を使用して、到着するジョブ群に適した DR を学習する手法を提案している. しかし、NN は事前学習に膨大な学習データが必要であり、学習データの選定が重要となる. また、林ら [3] は、定められた期間に繰り返してスケジューリング問題を解くローリングスケジューリング法を適用しており、そのスケジューリングには進化計算手法の遺伝的アルゴリズム (GA:Genetic Algorithms) を用いて各時点において未処理の複数のジョブの順序を組み替えている. しかし、様々なジョブ群が到着する動的スケジューリング問題では、その度に個体群を初期化し、その個体に対して遺伝的操作を行う必要があるため、高い計算コストがかかってしまう.

また、GA の拡張モデルである遺伝的プログラミング (GP:Genetic Programming)[4] を用いて DR を試行錯誤的に生成する手法が多く提案されている [5]. その理由として、NN は解探索にジョブ群の特徴に適した学習データが必要であるため、動的スケジューリング問題で発生するジョブ群の特徴の種類の分だけのデータが必要となり、多量のデータを用意することになるが、GP はランダムに生成された初期解に遺伝的操作を行うことで試行錯誤的に DR を生成するため、動的スケジューリング問題を解くためのデータを必要としないこと、また、生成した DR は再利用が可能であるため GA のように再スケジューリングを行う際に個体の初期化をする必要がないことが挙げられる. さらに解探索において必要な集中的探索と多様性のバランスをとるため、使用する個体を複数の島と呼ばれる個体群に分割した GA 手法の一種である島モデル GA[6] と同様の並列分散型の GP である島モデル GP を開発し、一般的な GP よりも精度の高い解探索が可能となった [5].

最適な DR はジョブ群の特徴によって変化することに注目すると、ジョブ群の特徴が似ていると最適な DR も似ていると考えられる。すなわち、過去に似た特徴のジョブ群に対して最適化された個体を用いて遺伝的操作を適用することで、計算コストを削減することができ、島モデル GP よりも効率的な解探索を行えると考えられる。本論文では、このような効率化されたルール探索の枠組みを実現するために3つの手法を提案する。1つ目は、ある特徴に特化した個体群を複数用意し、到着した新たなジョブ群の特徴に適した個体群を選択し、その個体群の個体のみで遺伝的操作を適用する手法を構築する。2つ目は、1つ目の改良として、島内での解の多様化を進めるために島を二重構造にした手法を構築する。3つ目は、2つ目の改良として、より効率的な解探索を行うため、個体群の統合や新たな個体群を作成する手法を構築する。これらの手法と従来研究を用いて、周期性のある動的スケジューリング問題に対する数値実験を行い、提案手法の有用性を検証する。

本論文の構成は以下の通り。第2章では動的スケジューリング問題と進化計算手法に関する関連研究を簡単に紹介し、第3章では提案手法のアルゴリズムや特徴について説明する。第4章では従来研究と提案手法の比較実験によって有用性を検証し、第5章で生成したルールについての分析を行う。最後に第6章では本論文の総括と今後の課題について述べる。

第 2 章

従来研究

本章では、動的スケジューリング問題、ディスパッチングルール、本論文と関わりのあるいくつかの進化計算手法について紹介する。

2.1. 動的スケジューリング問題

スケジューリング問題とは複数のジョブを並び替えることで、定められた目的を最適化する問題のことをいい、動的スケジューリング問題とは、既に与えられているジョブ群の全ての処理が完了する前に新しいジョブ群が断続的に到着し、再スケジューリングを行う必要のあるスケジューリング問題のことをいう。現実の生産現場では、その多くが動的スケジューリング問題として定義でき、最適なスケジューリングを実用的な時間内で効率的に策定することが必要となる。

スケジューリング問題に関する変数を表 2.1 のように定義する。

表 2.1: 変数の定義

J_i	i 番目のジョブ ($i = 1, 2, \dots, L$) n_i 個のオペレーション o_{ij} ($j = 1, 2, \dots, n_i$) を有する
p_{ij}	オペレーション o_{ij} の処理時間
\bar{p}	1 ジョブあたりの平均処理時間 $\left(\bar{p} = \frac{1}{L} \sum_{i=1}^L \sum_{j=1}^{n_i} p_{ij} \right)$
d_i	ジョブ J_i の納期
st_i	ジョブ J_i の到着時刻
rpt_i	ジョブ J_i の残り処理時間
rpn_i	ジョブ J_i の残りオペレーション数
C_i	ジョブ J_i の処理完了時刻

L 個のジョブが存在し、ジョブ J_i ($i = 1, 2, \dots, L$) は n_i 個のオペレーション o_{ij} ($j = 1, 2, \dots, n_i$) から構成される。オペレーション o_{ij} の処理時間を p_{ij} として各オペレーションは 1 台のマシンで処理される。各マシンは複数のオペレーションを同時に処理することはできず、処理している途中のオペレーションを中断することもできない。ジョブ J_i が到着時刻 st_i 以降にオペレーションの処理を開始し、納期 d_i までに

完了するようなスケジュールを考えたい。各ジョブのオペレーションには処理を行う順番があるが、異なるジョブのオペレーションには優先順位は決められていない。

動的スケジューリング問題では断続的に新たなジョブ群が到着するため、その時々で存在する未処理のジョブ J_i は異なる。ジョブ群の到着時に再度スケジューリングを行うが、処理が完了しているオペレーションと現時点で処理を行っているオペレーションを除外し、処理が行われていないオペレーションと到着したジョブ群がスケジューリングの対象となる。

あるジョブ群の到着時に、ジョブ J_i に属している、スケジューリングの対象である未処理のオペレーションの処理時間を合計したものを rpt_i 、未処理のオペレーションの数を rpn_i とする。そして、ジョブ J_i に属する最後のオペレーション o_{in_i} の処理が完了した時刻を C_i とする。

2.2. ディスパッチングルール

スケジューリング問題を解くには、未処理のオペレーションに優先度をつける作業を行い、処理可能の状態になっている機械に優先度を基にしてオペレーションを並び替える必要がある。その優先度を与えるために用いられる規則のことをディスパッチングルール (DR:Dispatching Rule) という。代表的な DR を以下に示す。

- **EDD**: 納期が最も近いジョブに含まれているオペレーションを選択するルール

$$PI_{ij} = \frac{1}{d_i} \quad (2.1)$$

- **FIFO**: 到着時刻が最も早いジョブに含まれているオペレーションを選択するルール

$$PI_{ij} = \frac{1}{st_i} \quad (2.2)$$

- **SLACK**: 時間的余裕を表すスラック時間が最も短いジョブに含まれているオペレーションを選択するルール

$$PI_{ij} = \frac{1}{d_i - st_i - rpt_i} \quad (2.3)$$

- **ATC**: パラメータ b, κ に応じてジョブの待ち時間の長さや納期までの余裕時間を考慮するルール

$$PI_{ij} = \frac{1}{p_{ij}} \exp \left\{ -\frac{\max[d_i - rpt_i - b(rpt_i - p_{ij}), 0]}{\kappa \bar{p}} \right\} \quad (2.4)$$

- **SPT** : 処理時間の最も短いオペレーションを選択するルール

$$PI_{ij} = \frac{1}{p_{ij}} \quad (2.5)$$

- **SRPT** : 残り処理時間が最も少ないジョブに含まれているオペレーションを選択するルール

$$PI_{ij} = \frac{1}{rpt_i} \quad (2.6)$$

- **CR+SPT** : ジョブの残り処理時間と納期を考慮するルール

$$PI_{ij} = \frac{1}{p_{ij}} \left(\max \left[\frac{d_i}{rpt_i}, 1 \right] \right)^{-1} \quad (2.7)$$

- **(SL/RPN)+SPT** : ジョブの残りオペレーション数と余裕時間を考慮するルール

$$PI_{ij} = \frac{1}{p_{ij}} \left(\max \left[\frac{d_i - rpt_i}{rpn_i}, 0 \right] + 1 \right)^{-1} \quad (2.8)$$

DR の特性は、未知のジョブ群に対しても優先規則にパラメータを代入するだけで優先度を計算することができ、低い計算コストでのスケジューリングを行うことができるが、ジョブ群の特徴によって結果が左右してしまう。新たなジョブ群が到着する動的スケジューリング問題に対しては、随時最適な DR を決定することが必要となるため、機械学習や進化計算手法を用いて優先規則を生成する手法が数多く提案されている。

2.3. 進化計算手法

2.3.1. GA, GP

生物の進化過程は、その生物が生きる環境への適合度の高さがカギとなり、高い個体であれば生き残る確率が高くなるのに対し、低い個体は淘汰されていく。生き残った適合度の高い個体が子孫を増やすことにより、その環境に順応できる遺伝子を持つ個体がさらに増えていく。さらに突然変異により遺伝子の一部が変化することによって、新たな環境に対応できる個体が生成される。このような遺伝的操作が個体群に対して繰り返し行われることにより、その環境下で子孫を繁栄させることのできる個体が生成される。

このような進化過程を模倣するために、遺伝子の集まりである染色体を 1 次元配列として定義したものが、遺伝的アルゴリズム (GA: Genetic Algorithms) である。GA をスケジューリング問題に使用する場合は、配列の各要素がオペレーションを表すようになり、その配列を組み替えることでスケジューリングを行っていく。ただ、新しいジョブ群が到着する動的スケジューリング問題では、到着ごとに各要素に入っ

ているオペレーションの初期化をしなければならないため、計算コストがかかってしまう。

そのような GA の拡張として構築された手法として、遺伝的プログラミング (GP: Genetic Programming) がある [5]。GP の個体は木構造を持ち、数式などといった構造的なデータを保持できるようにした。木構造の終端ノードにはターミナルセット、非終端ノードはファンクションセットから選択された要素で構成される。ターミナルセットにはプログラムの外部入力や引数なしの関数、定数があり、ファンクションセットには四則演算子や引数ありの関数がある。GP はスケジューリング問題に対して、GA のようにオペレーションを表すのではなく、優先規則を表しているため、新たなジョブ群が到着したとしても初期化の必要はなく再利用が可能である。そのため、GP は動的スケジューリング問題に対して有効な手法と言える。

2.3.2. 遺伝的操作

GA, GP は個体群に対して、交叉、突然変異、再生といった遺伝的操作を繰り返し、最適解を探索する。ここでは、GP における遺伝的操作の要約を示す。

- 初期個体の生成

遺伝的操作に使用する初期個体群はランダムに生成される。生成方法として、全ての個体の木構造の枝を定められた深さまでランダムに成長させて初期個体にする Full メソッド法、定められた深さの範囲内で木構造の枝をランダムに成長させることで深さの異なる初期個数を生成する Grow メソッド法、個体群を半数ずつ Full メソッド法と Grow メソッド法で生成する Ramped half-and-half 法がある。本論文では、初期個体群にも解の多様性を持たせたいため、様々なサイズや形状の個体を生成できる Ramped half-and-half 法を用いる。

- 交叉

全個体からランダムに選択された 2 つの親個体との間で染色体の組み換えを行い、新たな染色体を持つ子個体を生成する。GP では木構造の一部の部分木を交換することで交叉を行う部分木交叉法がよく適用され、本論文でも使用する。部分木交叉法は、2 つの親個体の木構造からランダムにノードを選択し、それを交叉点とする。一方の親個体の交叉点以下の部分木を、もう片方の親個体の交叉点以下に置き換えることにより子個体を生成する。2 つの親個体の部分木どうしを交換する方法もあるが、それによって保護すべき木構造を壊してしまう可能性がより高くなってしまう為、一般的には使用されない。部分木交叉法 の概念を図 2.1 に示す。

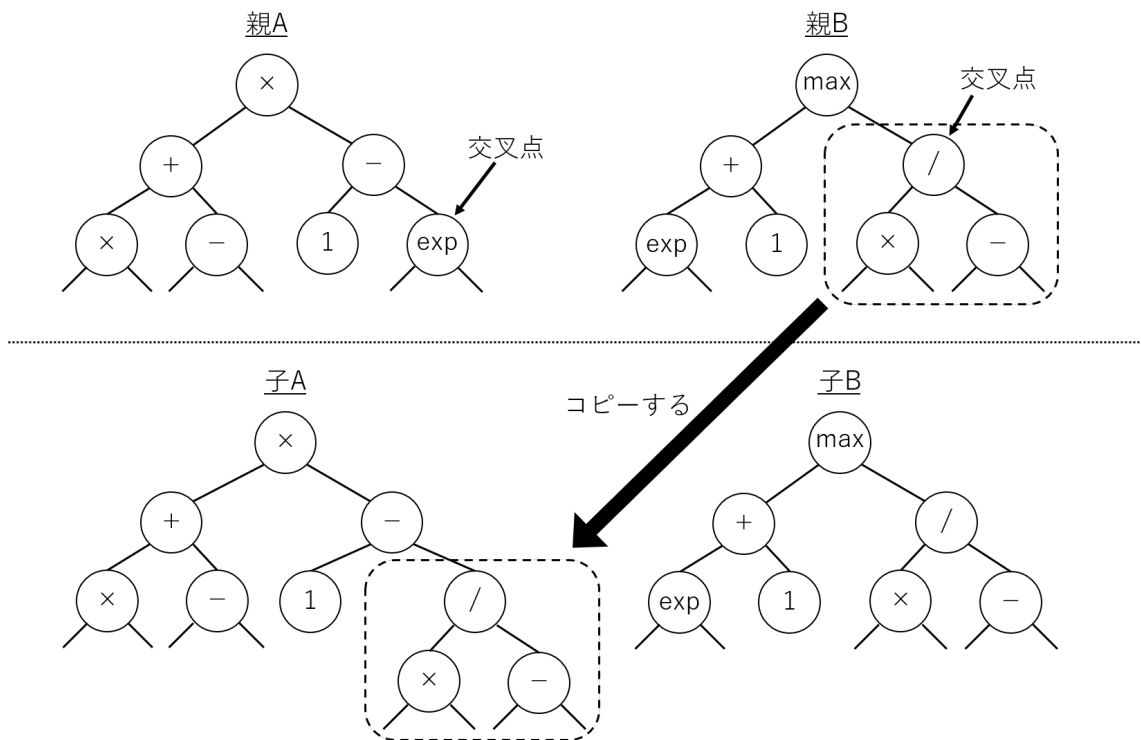


図 2.1: 部分木交叉法の概念図

- 突然変異

個体が持つ染色体の一部が何らかの影響によって変化し、新たな染色体を持つ個体が生まれることにより、個体群の進化の停滞を防ぐ。GP に対して多く使われる方法は、部分木突然変異法と点突然変異法がある。部分木突然変異法は、木構造の中からランダムにノードを選択し、それを突然変異点とし、その点以下の部分木を新たに生成した部分木に置き換える方法である。点突然変異法はランダムに選択されたノード自体が別のノードに変化する方法である。本実験では部分木突然変異法を用いる。部分木突然変異法の概念を図 2.2 に示す。

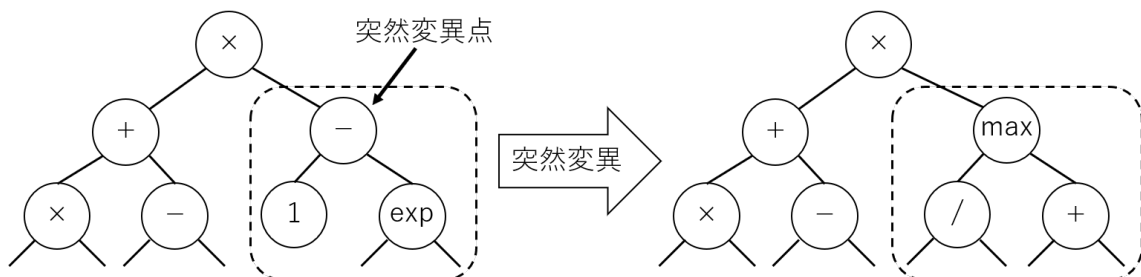


図 2.2: 部分木突然変異法の概念図

- 再生

自然界において、ある環境に対する個体の適合度が高ければ生き残り、低ければ淘汰されていく。本論文では、個体の選択方法としてトーナメント選択を用いる。トーナメント選択とは、全個体から一定数の個体を選出し、その中から適合度が最も高い個体だけを次世代に残していく選択方法である。トーナメント選択の特長として、優先度の差を考慮せずに個体に順位をつけるだけなので、容易に実装でき、集団の選択圧の影響を保つことができる。トーナメント選択の概念を図 2.3 に示す。

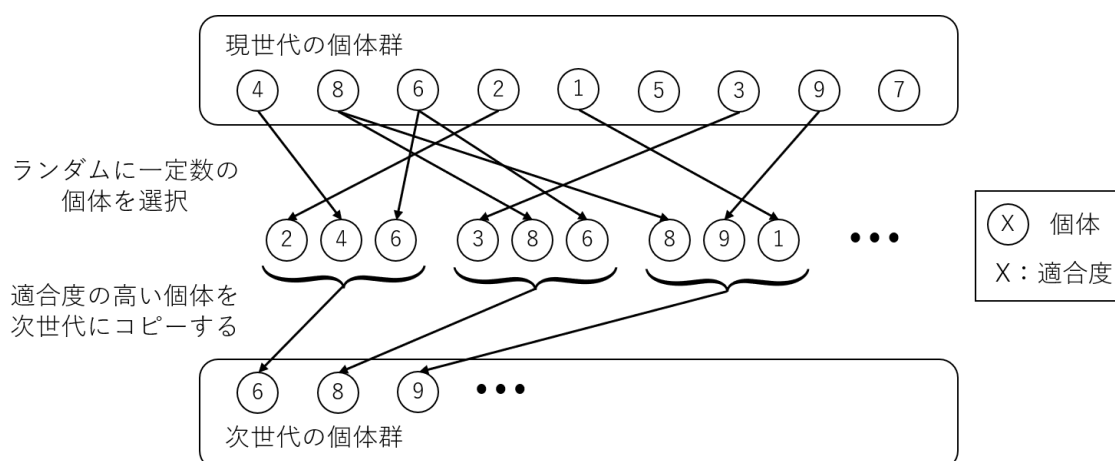


図 2.3: トーナメント選択の概念図

2.3.3. bloat 現象

bloat 現象とは、進化過程において GP の個体の木構造が大きくなる傾向のことをいう。GP の探索の特長として様々なサイズの個体が存在することが挙げられるが、大きくなりすぎると優先度の決定には必要のない部分木を有する個体が増える可能性がある。bloat 現象を抑えるために、木構造の深さに制限を持たせる方法 [7] や、深さに基づいたペナルティを与える方法 [5] がある。

2.3.4. GP を用いたスケジューリング

GP は木構造を持ち、個体そのものが DR を表している。例として既存 DR の ATC を GP の個体で表すと、図 2.4 のような木構造となる。

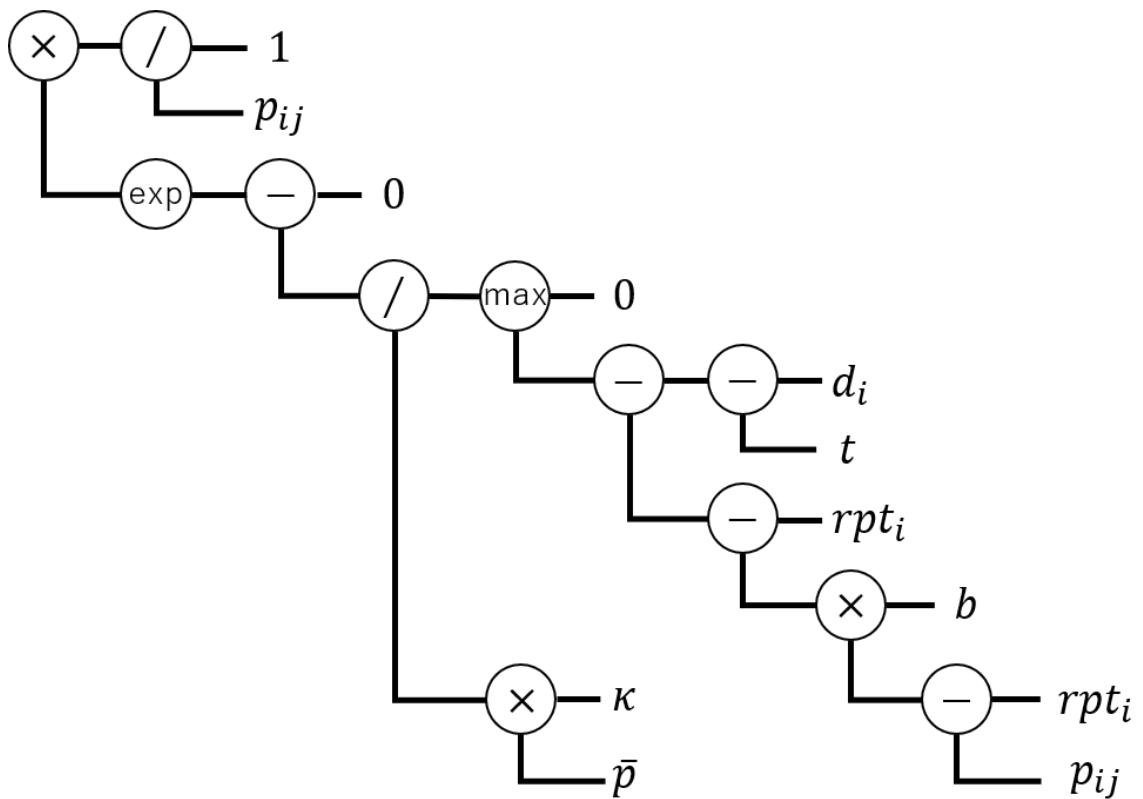


図 2.4: ATC ルール

このような木構造を持つ個体に対して遺伝的操作を行うことで、与えられたスケジューリング問題に適した DR を生成する。DR によってオペレーションの優先度を計算し、優先度の高い順番にオペレーションを並び替えてスケジュールを作成し、待機中のマシンでスケジュールにそった処理を行う。個体の DR によって作成されたスケジュールから適合度を求め、個体の評価を行う。適合度とは、与えられたスケジューリング問題に対する各個体の評価尺度を適合度 F とする。

2.3.5. エリート保存則

各世代において、与えられているスケジューリング問題に対する適合度 F が最も高い個体をエリート個体という。そしてエリート保存則とは、エリート個体を遺伝的操作の対象とせずに次世代に受け継ぐことをいう。これにより、環境に適している個体を壊してしまう確率が減少し、良好な個体が失われることを防ぐことができる。

2.3.6. 島モデル

2.3.6.1. 島モデル GA, 島モデル GP

一般的な GA は、用意された全ての個体に対して遺伝的操作を行うことで解探索をするが、局所解に収束してしまい、十分な解探索が行えない可能性がある。十分な解探索には集中的探索と多様性のバランスをとる必要があり、これを解決するための手法として島モデル GA がある。

島モデル GA(IGA) とは、与えられている個体を複数の島と呼ばれる個体群に分割し、各島で独立して遺伝的操作を行う並列 GA の一種である。島での独立した遺伝的操作による集中的探索と、島ごとに別々の進化を遂げることで解の多様性が保たれ、バランスの取れた解探索をすることができる。さらに IGA には、一定期間ごとに島間で複数の個体を入れ替える移住をさせることで島間での情報共有を行っており、より広範囲な解探索が可能となる。動的スケジューリング問題に対して GA より GP を用いたほうが効率的であるため、IGA の構造を GP に置き換えた島モデル GP(IGP)[5] が考えられた。IGA, IGP の総称を「島モデル」と呼び、図 2.5 に島モデルの概念図を示す。概念図では移住経路が島を 1 回ずつ通るような円状に配置しているが、それ以外にも様々な移住経路を持つ島モデルがある。

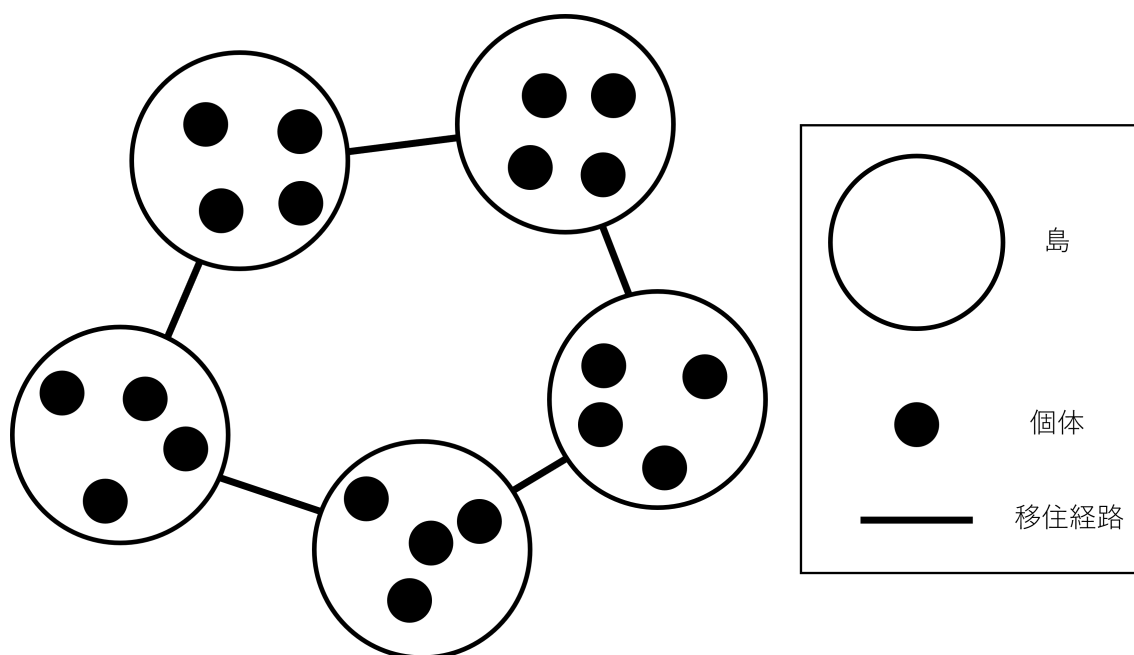


図 2.5: 島モデルの概念図

島モデルを使用するにあたり、移住間隔、移住率、島の数を決めなければならない。移住間隔とは個体を移住させる一定の世代周期のことをいい、移住率とは島

に含まれている個体数に対する移住する個体数の割合である。移住を行う際は、各島から移住率を基にした数の個体がランダムに選出される。

2.3.6.2. 移住トポロジー

移住トポロジーとは島モデルにおける島間での移住経路を決定するグラフモデルのことをいい、使用するグラフモデルの特徴や、個体の移住方向によって解探索の能力に影響が生じる [8]。

移住トポロジーに使用されている代表的なグラフモデルとして、Chain モデル、Ladder モデル、Ring モデル、Random Ring モデル、WS モデルなどが提案されている [9][10]。Chain モデルは、全ての島を一列に並べ、隣り合う島同士で移住経路を作成する。Ladder モデルは、全ての島から2つの島を選択してペアを作り、そのペアだけで個体の移住を行う。このモデルは、全ての島がペアになるよう、偶数個の島でなくてはならない。Ring モデルは、図 2.6 のように、全ての島を同心円上に等間隔で並べ、ある島から最短距離にある2つの島に移住経路を作成する。このモデルの発展型として Ring+1+2 モデルがあり、Ring モデルに追加して、2つ隣の島までを短絡的に繋ぐ移住経路を作成する。Random Ring モデルは、移住間隔ごとに Ring モデルの島の位置をランダムに変更させて、新たな島の並びを作成する。

最後に、WS モデルは本論文で使用するグラフモデルで、Watts and Strogatz[10] によって提案された。WS モデルは、図 2.6 のように、Ring モデルを基盤にしたモデルで、割合 p で Ring モデルのある島の移住経路を繋がっていない島へ付け替えることで、短絡的な移住経路を作成する。これにより、Ring モデルでは遠い島への移住に複数の島を経由していたが、WS モデルを用いることで直接的に移住ができる確率が高くなる。そのため、より広範囲な解探索が短時間で行うことが可能になる。

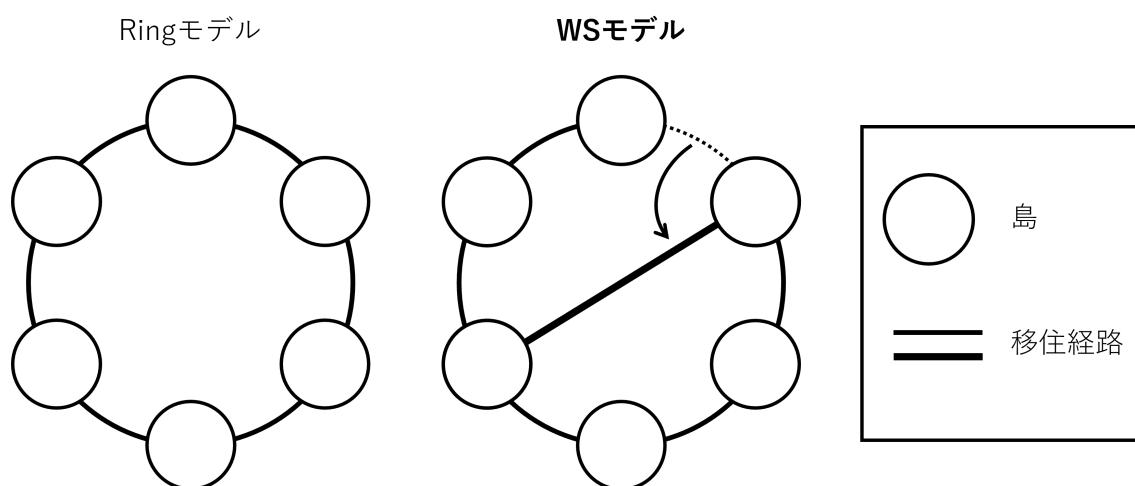


図 2.6: Ring モデルと WS モデルの概念図

第3章

提案手法

本章では提案手法である、個体群選択型 GP[11] と個体群増減型二重並列構造 GP[12] について紹介する。

3.1. 個体群選択型 GP

3.1.1. 概要

GP を用いた解探索は、個体の初期化の必要がなく再利用が可能なため、動的スケジューリング問題に適した手法である。しかし、ジョブ群の特徴によって最適な DR が異なることが多くの研究で明らかにされている。ジョブ群の特徴が似ている場合は最適な DR の構造も似ていると考えられ、以前に似た特徴のジョブ群を処理した個体を再度使用することで効率的な解探索ができると考えられる。ただ、進化計算手法は直近のデータの影響が大きく反映される特長があるため、直近で対応してこなかったような特徴を持つジョブ群が到着した場合、そのジョブ群に最適な DR に進化するまでに計算コストがかかってしまう。これを解決するために、過去のデータをいくつか分類して保存しておき、必要な時に適切なデータを再利用できるような枠組みが必要であると考えられる。

そのような枠組みを実現するために、島モデルのように複数の個体群をそれぞれ独立して、特徴の異なるジョブ群ごとに学習させる手法を考える。動的スケジューリング問題に対しては、到着したジョブ群の特徴を基に、複数の個体群の中から適合する個体群を選択することができれば、それを初期個体群として遺伝的操作を行うことで、解探索のための個体の変更箇所が少なくなると考えられ、計算コストの削減が期待できる。ジョブ群に対して個体群がどの程度適合しているかを図るために、各個体群が直前の遺伝的操作で保存したエリート個体を用いる。すなわち到着したジョブ群に対してエリート個体を用いて適合度を計算し、適合度が最大となったエリート個体を持つ個体群をそのジョブ群の特徴に似た他のジョブ群に対して学習済みであると解釈する。IGP では島間で移住を行うが、本手法は個体群をある特徴に特化させたルール探索が目的であるため、移住は行わない。島と個体群は、複数の個体の集合体という点で同じものであるが、本論文では、集合体の間で相互的な移住を行うものを島と呼び、相互的な移住を行わないものを個体群と呼ぶこととする。

以上のように、IGPのような並列分散の構造を持つGPを用いて獲得した個体群から、ジョブ群の特徴に最も適合すると判断された1つの個体群を選択し、特化された個体群で解探索を行う手法のことを個体群選択型GPとする。図3.1に選択型GPの概念図を示す。

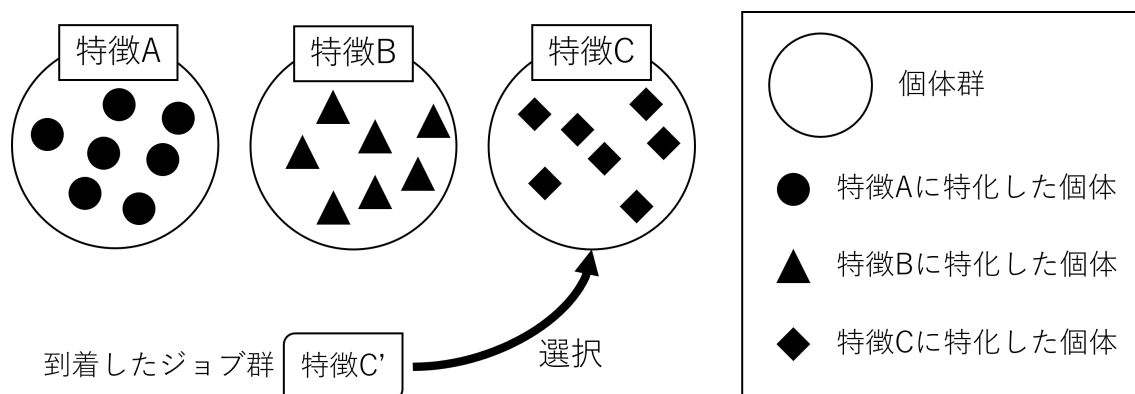


図 3.1: 個体群選択型 GP の概念図 [11]

3.1.2. 個体群選択型 GP の適用方法

個体群選択型 GP を動的スケジューリング問題に適用する場合の適用方法について、以下に述べる。

- **STEP1:初期個体群の獲得**

個体群の最大数を N として、 d 番目に到着するジョブ群 JS_d ($d = 1, 2, \dots, N$) に対して個体群 d を用いて遺伝的操作を行い、解探索を行う。このとき、個体群 d の初期個体群はランダムに初期化されるものとする。

ジョブ群 JS_d に対して、最も高い適合度となる DR を獲得した個体の1つを個体群 d のエリート個体として保存する。以上の操作を N 回を行い、STEP2へ進む。ただし、到着するジョブ群の数が N 未満である場合、最後のジョブ群に対する遺伝的操作が完了したら、終了する。

- **STEP2:新しいジョブ群の到着**

新しいジョブ群が到着した場合はSTEP3へ進み、なければ操作を終了する。

- **STEP3:個体群の選択**

各個体群に保存されているエリート個体を用いて、新たに到着したジョブ群に対する目的関数値 E_i ($i = 1, 2, \dots, N$) を計算する。個体群 s のエリート個体による目的関数値 E_s が最小である場合、個体群 s を用いて解探索を行う。目的関数

値が最小となるエリート個体をもつ個体群が複数ある場合は、その中から1つの個体群をランダムに選出して使用する。

解探索終了後に、その個体群のエリート個体を更新し、STEP2に戻る。

3.2. 個体群選択型 IGP

個体群選択型 IGP は、個体群選択型 GP における各個体群が、GP ではなく IGP のような並列分散型の構造を持つ二重並列進化の構造を持つ。個体群選択型 GP は個体群が独立した並列構造を持つ GP で、特徴ごとに特化させた個体群を用いて処理を行うが、解探索自体は一般的な GP と同様であるため広範囲な解探索は難しい。それを解消するために、各個体群を島モデルとすることで、個体群全体を個体群と島モデルの二重並列構造として、解の広範囲化を目的とする。ただし、各個体群の間では個体の移住は行われませんが、島間での移住を行う。個体群選択型 IGP の適用方法は個体群選択型 GP と同様である。

3.3. 個体群増減型二重並列構造 GP

3.3.1. 概要

個体群増減型二重並列構造 GP は、個体群選択型 IGP の改良で、Base 個体群の追加と、選択される個体群の増減という大きく2つの特徴を新たに持つ。図 3.2 に個体群増減型二重並列構造 GP の概念図を示し、それぞれの特徴については以下で述べる。

- Base 個体群

選択される個体群はジョブ群の特徴に適した DR を生成することを目的としているが、Base 個体群は全てのジョブ群に共通するような DR を生成することを目的とする。そのためスケジューリングを行う場合には、ジョブ群の特徴を基に選択された個体群と Base 個体群の2つの個体群を使用する。与えられたジョブ群に最適な DR を生成した個体が Base 個体群にある場合のみ、Base 個体群から選択された個体群に向けて、個体の移住を行う。適合度の高い個体を選択した個体群に移住させることで、選択された個体群のさらなる進化が期待される。

- 個体群の増減

さまざまな特徴を持つジョブ群が不規則に到着する動的スケジューリング問題において、到着したジョブ群の特徴に既存の個体群が適さない可能性がある。この場合、新たに個体群を作成して遺伝的操作を行うことで、他の個体群に影響

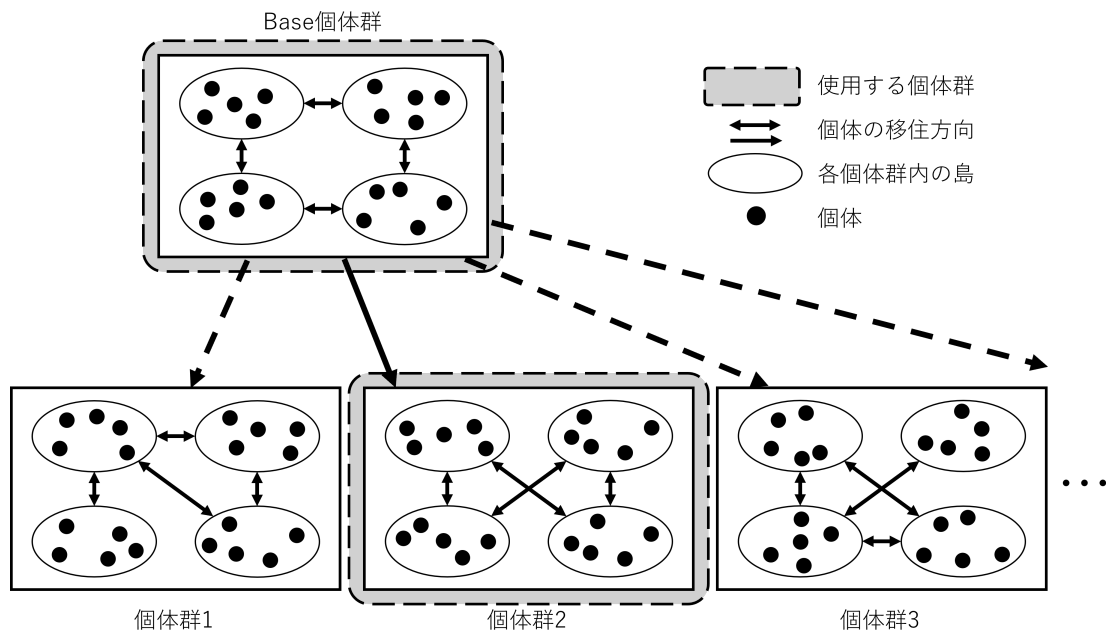


図 3.2: 個体群増減型二重並列構造 GP の概念図 [12]

響を下さずに効率的な解探索ができると考えられる。また、動的スケジューリング問題に対する学習を繰り返した結果として複数の個体群が同じような特徴のジョブ群に適合することで、似た特徴を持つ個体群が獲得される可能性がある。この場合、そのような複数の個体群を1つの個体群に統合することで効率的な解探索ができると考えられる。

3.3.2. 個体群増減型二重並列構造 GP の手順

個体群増減型二重並列構造 GP を動的スケジューリング問題に適用する場合の手順は以下のとおりである。ここで、 d はジョブ群が到着した回数、 N は Base 個体群を除く個体群の数である。また、個体群の選択と増減判断のフローチャートを図 3.3 に示す。

- **STEP1:** 初期個体群の獲得

動的スケジューリング問題は、前節の個体群選択型 GP と同様に初期の個体群の数を N_0 として、 d 番目に到着するジョブ群 JS_d ($d = 1, 2, \dots, N_0$) に対して、初期個体群をランダムに生成した個体群 d と Base 個体群を用いて遺伝的操作を行い、解探索を行う。ジョブ群 JS_d に対して最も適合度の高くなるような DR を持つ個体を個体群 d のエリート個体として保存する。以上の操作を N_0 回繰り返し、STEP2 へ進む。ただし、到着するジョブ群の数が N_0 未満である場合、最後の

ジョブ群に対する遺伝的操作が完了したら、終了する。

● **STEP2:**新しいジョブ群の到着

新しいジョブ群が到着した場合は STEP3 に進み、なければ操作を終了する。

● **STEP3:**到着したジョブ群の特徴に適合する個体群の有無を判定

到着したジョブ群に対して個体群 i に保存されているエリート個体を用いて、目的関数値 E_i を計算する。条件 I ($E_i \leq W \times \alpha_d$) を満たす個体群は、ジョブ群に適合すると判定する。ここで、式 (3.1) で与えられる個体群 d における α_d は過去の最良目的関数値の重み付き総和、 W は個体群の増減に関するパラメータである。 D はジョブ群到着回数の最大値、 E_{d-1} は $(d-1)$ 回目に着したジョブ群に対するエリート個体の目的関数値、 δ は α_d の更新に使用するパラメータ、 α_1 は α_d の初期値である。

$$\alpha_d = \frac{1}{\delta} \times \frac{d}{D} \alpha_{d-1} + \left(1 - \frac{1}{\delta} \times \frac{d}{D}\right) E_{d-1} \quad (3.1)$$

α_d を計算するにあたり、動的スケジューリング問題の初期は数少ない過去の最良目的関数値から大まかな基準値を作成するために E_{d-1} に対する重みを大きくし、新たなジョブ群が到着し続けるにつれて、今までに得られた最良目的関数値の平均をとるために、それぞれにかかる重みの大きさを近づかせるような重み付き和とした。

条件 I を満たす個体群の数により、以下の様に 3 つの手順に分岐される。

– STEP3-1:条件 I を満たす個体群が存在しない

新たに生成した個体群 ($N+1$) と Base 個体群において遺伝的操作を行う。新たに到着したジョブ群に適したエリート個体を個体群 ($N+1$) に保存して、STEP2 へ戻る。ただし、新たな個体群はランダムに生成する。

– STEP3-2:条件 I を満たす個体群が 1 つだけ存在する

条件 I を満たす個体群と Base 個体群において遺伝的操作を行う。新たに到着したジョブ群に適したエリート個体を条件 I を満たす個体群に保存して、STEP2 へ戻る。

– STEP3-3:条件 I を満たす個体群が複数個存在する

条件 I を満たす目的関数値が最も小さい 2 つの個体群 x, y ($E_x \leq E_y$) が、条件 II ($M_x \geq \beta$ and $\frac{s_{xy}}{M_x} \geq \gamma$) を満たせば、2 つの個体群を統合する。条件 II における M_x は個体群 x がこれまでに解探索に使用された回数、 s_{xy} は条件 I を満たす上位 2 位が個体群 x, y であった回数である。条件 II では、個体群 x を使用し

た回数が β 回以上であり、かつ、条件 I の上位 2 位が個体群 x, y の組み合わせになった回数と個体群 x を使用した回数の比率が γ 以上である場合、個体群 x, y からトーナメント選択を用いて個体を選出し、個体群 x に統合させる。これにより、新しく到着したジョブ群の特徴に適合すると推定される個体を個体群 x に集めることができ、効率的な解探索ができると考えられる。条件 II に含まれる使用回数に関する制約は、新しく到着したジョブ群の特徴に対して最適と判定された個体群が十分な成長ができているかを判断する。比率に関する制約は、ある 2 つの個体群の組み合わせで様々なジョブ群に対して最も適合していると判定される回数が多く、それが最適な個体群の使用回数に近いほど、その 2 つの個体群には似た特徴を持つ DR を獲得していると判断できる。条件 II を満たさない場合、条件 I を満たす目的関数値が最も小さい個体群 x と Base 個体群において遺伝的操作を行う。新たに到着したジョブ群に適したエリート個体を個体群 x に保存して、STEP2 へ戻る。

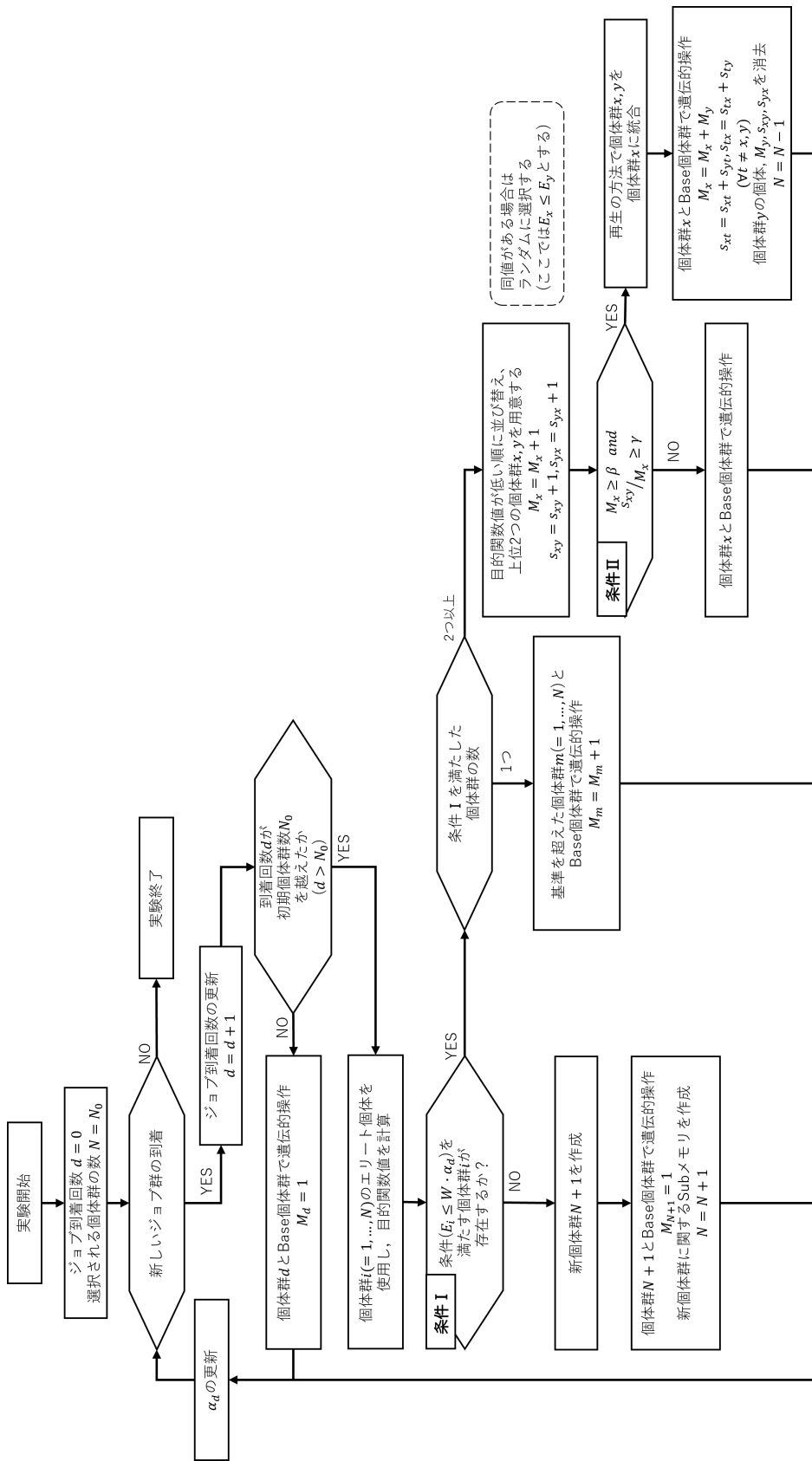


図 3.3: 個体群の選択と増減判断のフローチャート

第4章

数値実験

本章では、動的スケジューリング問題に対して従来研究の IGP[5]、提案手法の個体群選択型 GP(選択型 GP)[11]、個体群選択型 IGP(選択型 IGP)、個体群増減型二重並列構造 GP(増減型 IGP)[12] の4手法を用いて数値実験を行い、有用性を検証する。使用するシナリオは特徴の変わるジョブ群が断続的に到着するようにし、現実的なジョブ群の到着を模倣させる。

4.1. 実験概要

実験に使用するシナリオは、初期のジョブ数を 10,20,30 個とし、ジョブが 10,20,30 個のオペレーションを有するものとする。そのため、シナリオの種類は9種類である。シナリオサイズは $J \times O$ と表され、 J はジョブ数、 O はオペレーション数を意味する。ジョブ群の到着間隔 RT は、1つ前に到着したジョブ群をスケジュールした際のメイクスパン $C_{\max} = \max\{C_i | i = 1, 2, \dots, L\}$ に重み付けし、次式で与えられる。

$$RT = 0.8 \times C_{\max} \quad (4.1)$$

新たに到着するジョブ数は、1つ前の到着から今までに全てのオペレーションの処理が完了したジョブ数となる。ジョブ数は到着ごとに変化するが、オペレーション数は全てのジョブで一定とする。新しいジョブ群の到着回数は100回とする。使用するシナリオは、現実的なジョブ群の到着を模倣させるため、処理時間の最大値 s を式(4.2)のように周期4、納期のきつさを調節する変数 t を式(4.3)のように周期3で変化させることで、全体で周期12で特徴が周期的に変化するシナリオを作成した。

$$s : 5 \rightarrow 4 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow \dots \quad (4.2)$$

$$t : 5.0 \rightarrow 2.4 \rightarrow 7.6 \rightarrow 5.0 \rightarrow 2.4 \rightarrow \dots \quad (4.3)$$

それぞれのジョブの処理時間 p_{ij} と到着時刻 st_i 、納期 d_i は以下のように与えられる。ここで a_1, a_2, a_3 は $[0, 1]$ の一様乱数である。

$$p_{ij} = (s - 1)a_1 + 1 \quad (4.4)$$

$$st_i = a_2 \bar{p} \quad (4.5)$$

$$d_i = st_i + \bar{p}(1 + a_3t) \quad (4.6)$$

世代ごとのエリート個体を用いてスケジューリングを行い，目的関数値 E が 30 世代にわたって変化がない場合，このエリート個体が与えられたジョブ群に対して最適な DR であると考え，探索を終了する．ここで，最良値の変動が 1.0×10^{-5} 以下であれば変動していないものとする．

数値実験では，すべてのジョブが処理を完了するメイクスパン $C_{\max} = \max\{C_i | i = 1, 2, \dots, L\}$ と納期遅れ $T_i = \max\{0, C_i - d_i\}$ の重み付和の最小化を目的とし，目的関数 E は式 (4.7) のように定式化される．ここで WT は納期遅れに対する重みであり，本論文では 100 とする．

$$\text{minimize } E = WT \times \sum_{i=1}^L T_i + C_{\max} \quad (4.7)$$

個体の評価尺度である適合度 F は，式 (4.7) を用いて式 (4.8) のように定式化される．

$$F = \frac{1}{E} \quad (4.8)$$

GP を使用するためのパラメータを表 4.1 に示す．実験の試行回数を 100 回とし，各手法の実験結果には，各動的変化におけるエリート個体から算出された適合度 F ，メイクスパン C_{\max} ，納期遅れ $\sum T_i$ のそれぞれの平均値と，全てのジョブ群を処理し終えるのにかかる計算時間をデータとして用いる．

表 4.1: GP のパラメータ

パラメータ	IGP[5]	選択型 GP[11]	選択型 IGP	増減型 IGP[12]
初期個体群数	-	5	5	6 (Base 個体群を含む)
最大個体群数	-	(変化なし)	(変化なし)	11 (Base 個体群を含む)
島数	5	-	5	5
初期の全個体数	1000	1000	5000	3000
1 回の解探索に使用する個体数	1000	200	1000	1000
交叉確率	0.8			
突然変異確率	0.1			
島間の移住確率	0.1	(移住なし)	0.1	0.1
選択方法	トーナメント選択			
初期個体生成	Ramped half-and-half 法			
木の深さの最大値	8			
終了条件	最良解が 30 世代変動なし			
ターミナルセット	8 種類 (p_{ij} , \bar{p} , d_i , st_i , rpt_i , rpn_i , 1, 0)			
ファンクションセット	6 種類 (+, -, ×, /, max, exp)			
α_d の初期値	(使用しない)			0.0
規定回数 β				5
規定比率 γ				0.7
α_d の更新に使用するパラメータ δ				2.0
個体群の増減に関するパラメータ W				20

4.2. 結果と考察

先行研究と提案手法の比較を行うため、各項目の平均値と、試行回数 100 回の中で適合度 F が最良値となった場合のデータ、最悪値となった場合のデータを示す。シナリオごとの実験結果を表 4.2-4.10 に、100 試行分の適合度 F を図 4.1-4.9 に示す。実験結果では、各項目の最良値を太文字にしている。表 4.11 には、各シナリオにおける適合度 F の平均値が最良となった手法を纏めている。

・ 10 × 10

表 4.2: 実験結果

	$F(\times 10^{-6})$	C_{\max}	$\sum T_i$	time[s]
IGP[5]				
平均値	2059.92	42.99	4.44	197.06
最良値	2252.20	43.01	4.01	197.83
最悪値	1815.08	42.94	5.08	197.28
選択型 GP[11]				
平均値	2122.85	42.99	4.29	52.15
最良値	2288.80	42.91	3.94	54.62
最悪値	1865.74	42.98	4.93	50.35
選択型 IGP				
平均値	1999.54	42.99	4.59	188.47
最良値	2267.63	42.99	3.98	186.11
最悪値	1694.74	43.06	5.47	183.84
増減型 IGP[12]				
平均値	2090.87	42.99	4.37	187.45
最良値	2336.56	42.98	3.85	199.45
最悪値	1785.94	42.93	5.17	181.60

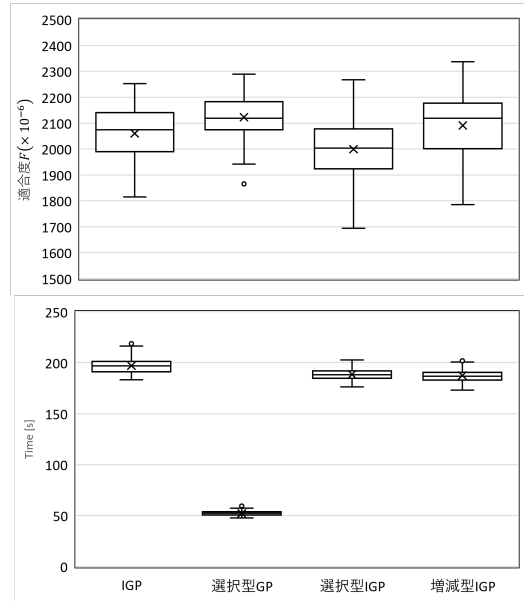


図 4.1: 適合度と計算時間

・ 10 × 20

表 4.3: 実験結果

	$F(\times 10^{-6})$	C_{\max}	$\sum T_i$	time[s]
IGP[5]				
平均値	1112.15	72.62	8.27	339.15
最良値	1160.67	72.57	7.89	344.88
最悪値	1038.83	72.62	8.90	335.93
選択型 GP[11]				
平均値	1046.59	72.62	8.83	74.34
最良値	1106.66	72.62	8.31	78.42
最悪値	1011.48	72.65	9.16	77.11
選択型 IGP				
平均値	1085.02	72.65	8.51	303.86
最良値	1159.15	72.7	7.90	298.43
最悪値	946.43	72.60	9.84	307.30
増減型 IGP[12]				
平均値	1103.04	72.63	8.34	305.75
最良値	1152.53	72.66	7.95	307.02
最悪値	1012.59	72.57	9.15	306.26

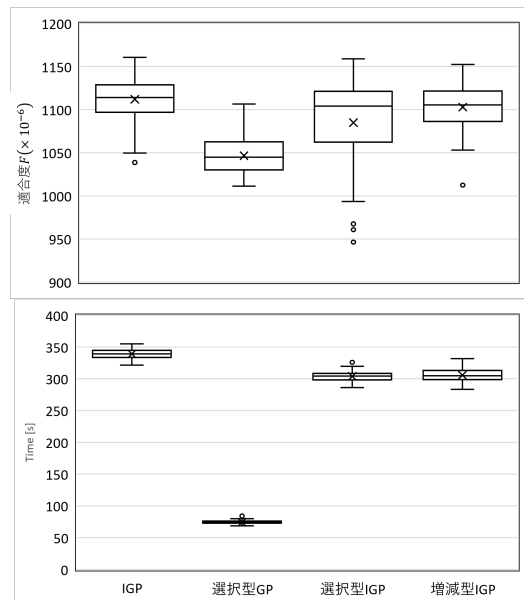


図 4.2: 適合度と計算時間

・ 10 × 30

表 4.4: 実験結果

	$F(\times 10^{-6})$	C_{\max}	$\sum T_i$	time[s]
	IGP[5]			
平均値	538.20	102.46	17.56	449.81
最良値	559.45	102.46	16.85	442.76
最悪値	516.40	102.49	18.34	447.69
選択型 GP[11]				
平均値	543.64	102.34	17.37	102.21
最良値	553.32	102.27	17.05	100.47
最悪値	534.68	102.29	17.68	98.59
選択型 IGP				
平均値	542.68	102.42	17.41	427.36
最良値	557.27	102.45	16.92	424.28
最悪値	522.89	102.45	18.10	417.48
増減型 IGP[12]				
平均値	543.50	102.43	17.38	436.85
最良値	563.88	102.43	16.71	439.97
最悪値	524.53	102.48	18.04	426.45

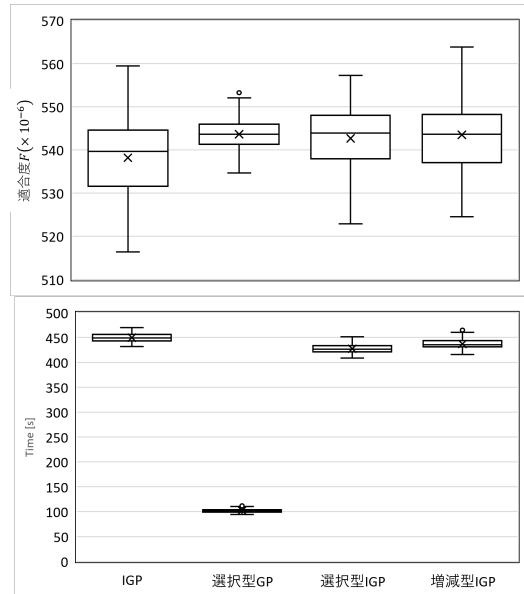


図 4.3: 適合度と計算時間

・ 20 × 10

表 4.5: 実験結果

	$F(\times 10^{-6})$	C_{\max}	$\sum T_i$	time[s]
	IGP[5]			
平均値	585.74	67.20	16.43	550.06
最良値	644.26	67.18	14.85	671.42
最悪値	543.72	67.18	17.72	566.67
選択型 GP[11]				
平均値	555.88	67.20	17.36	131.56
最良値	628.45	67.22	15.24	145.57
最悪値	485.39	67.21	19.93	125.20
選択型 IGP				
平均値	578.63	67.22	16.64	513.65
最良値	630.44	67.20	15.19	535.68
最悪値	530.16	67.22	18.19	517.33
増減型 IGP[12]				
平均値	568.33	67.21	16.94	517.77
最良値	619.50	67.21	15.47	526.88
最悪値	527.35	67.28	18.29	544.86

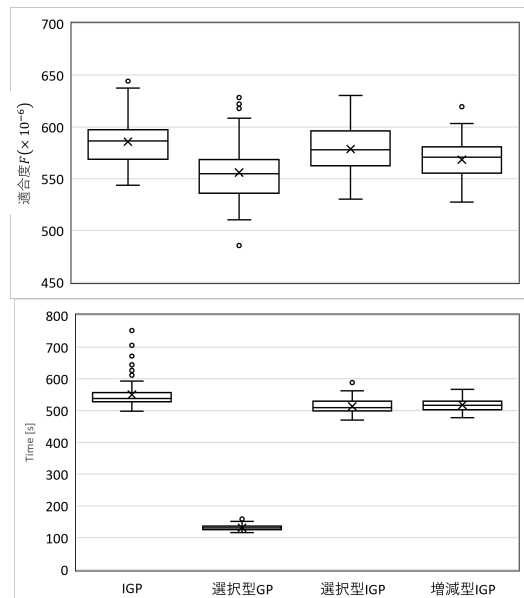


図 4.4: 適合度と計算時間

・ 20 × 20

表 4.6: 実験結果

	$F(\times 10^{-6})$	C_{\max}	$\sum T_i$	time[s]
	IGP[5]			
平均値	532.22	95.05	17.91	819.35
最良値	608.24	95.09	15.49	824.72
最悪値	446.80	95.13	21.43	811.54
選択型 GP[11]				
平均値	531.43	95.05	17.99	197.29
最良値	618.80	95.04	15.21	205.38
最悪値	421.22	95.05	22.79	174.88
選択型 IGP				
平均値	525.15	95.04	18.20	795.26
最良値	616.87	95.09	15.26	845.85
最悪値	400.30	95.11	24.03	765.58
増減型 IGP[12]				
平均値	548.04	95.04	17.37	792.92
最良値	621.85	95.10	15.13	758.59
最悪値	466.18	95.08	20.50	801.81

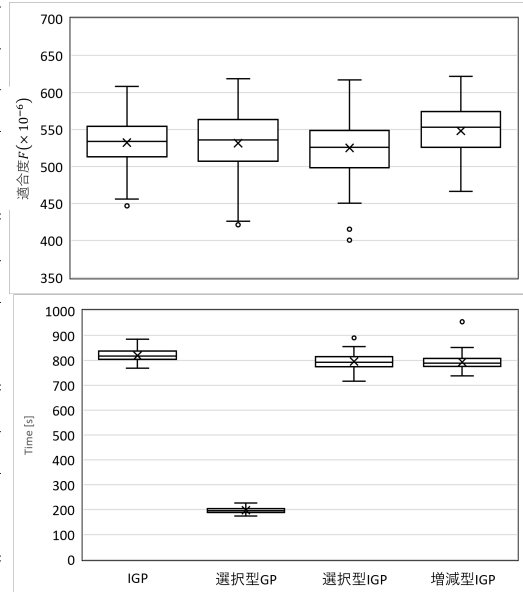


図 4.5: 適合度と計算時間

・ 20 × 30

表 4.7: 実験結果

	$F(\times 10^{-6})$	C_{\max}	$\sum T_i$	time[s]
	IGP[5]			
平均値	341.31	125.94	28.05	1072.82
最良値	357.53	125.97	26.71	1084.31
最悪値	323.11	125.95	29.69	1107.06
選択型 GP[11]				
平均値	333.03	125.95	28.78	259.20
最良値	349.90	125.97	27.32	275.13
最悪値	317.97	125.98	30.19	269.03
選択型 IGP				
平均値	334.30	125.95	28.67	1041.41
最良値	353.23	125.99	27.05	1102.89
最悪値	312.41	125.94	30.75	1037.23
増減型 IGP[12]				
平均値	334.93	125.95	28.61	1062.28
最良値	348.18	126.05	27.46	1103.58
最悪値	318.99	125.93	30.09	1037.42

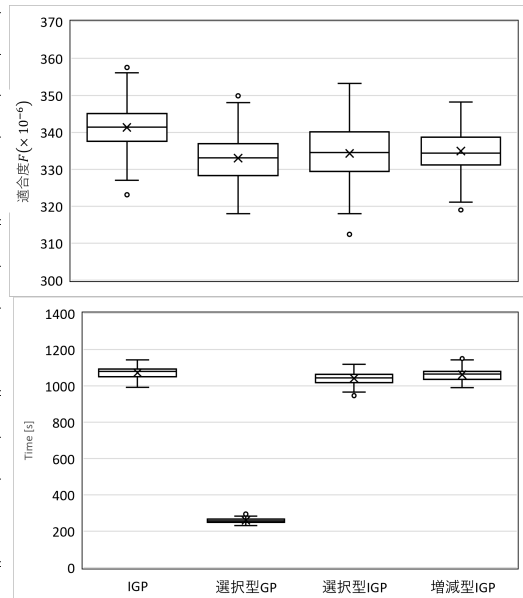


図 4.6: 適合度と計算時間

・ 30 × 10

表 4.8: 実験結果

	$F(\times 10^{-6})$	C_{\max}	$\sum T_i$	time[s]
IGP[5]				
平均値	89.31	84.73	111.21	1036.28
最良値	93.93	84.77	105.62	1077.83
最悪値	80.72	84.71	123.04	1071.94
選択型 GP[11]				
平均値	88.40	84.72	112.35	271.73
最良値	93.43	84.73	106.18	287.71
最悪値	82.76	84.70	119.98	275.65
選択型 IGP				
平均値	88.20	84.73	112.65	981.81
最良値	93.30	84.72	106.33	965.52
最悪値	77.28	84.71	128.56	954.27
増減型 IGP[12]				
平均値	88.96	84.72	111.63	930.57
最良値	93.02	84.62	106.66	971.88
最悪値	81.53	84.73	121.81	868.15

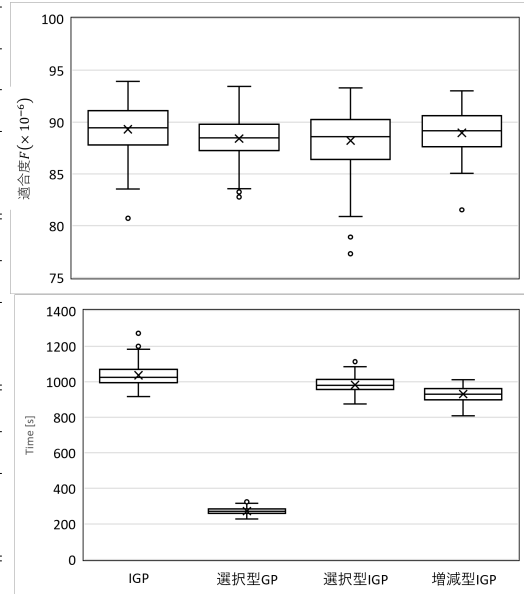


図 4.7: 適合度と計算時間

・ 30 × 20

表 4.9: 実験結果

	$F(\times 10^{-6})$	C_{\max}	$\sum T_i$	time[s]
IGP[5]				
平均値	213.93	120.12	45.60	1430.62
最良値	230.46	120.08	42.19	1423.61
最悪値	193.83	120.10	50.39	1418.59
選択型 GP[11]				
平均値	216.62	120.13	45.04	362.23
最良値	242.95	120.15	39.96	362.23
最悪値	193.68	120.08	50.43	378.73
選択型 IGP				
平均値	214.66	120.14	45.45	1382.59
最良値	238.72	120.07	40.69	1339.48
最悪値	194.40	120.11	50.24	1354.24
増減型 IGP[12]				
平均値	209.26	120.13	46.66	1350.79
最良値	229.25	120.08	42.42	1337.61
最悪値	191.79	120.08	50.94	1374.30

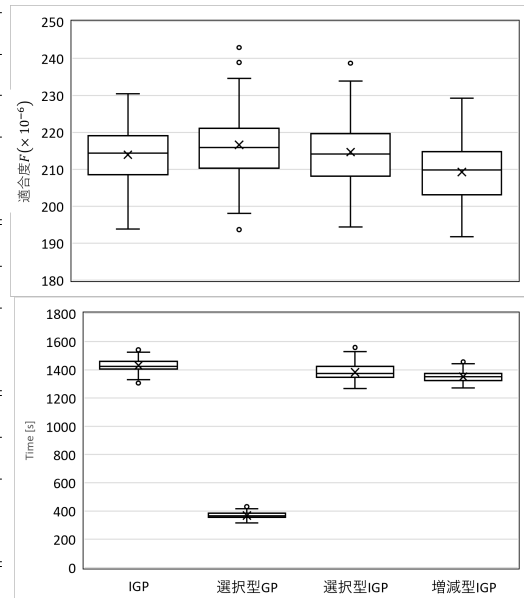


図 4.8: 適合度と計算時間

表 4.10: 実験結果

	$F(\times 10^{-6})$	C_{\max}	$\sum T_i$	time[s]
IGP[5]				
平均値	195.48	149.67	49.69	1905.01
最良値	210.99	149.59	45.90	1801.56
最悪値	184.00	149.73	52.85	1833.31
選択型 GP[11]				
平均値	197.17	149.66	49.27	496.51
最良値	214.79	149.70	45.06	530.51
最悪値	182.03	149.64	53.44	487.91
選択型 IGP				
平均値	193.65	149.66	50.18	1757.53
最良値	204.64	149.59	47.37	1677.78
最悪値	177.31	149.72	54.90	1803.69
増減型 IGP[12]				
平均値	191.22	149.66	50.73	1809.12
最良値	204.22	149.68	47.47	1802.03
最悪値	180.16	149.71	54.01	1753.72

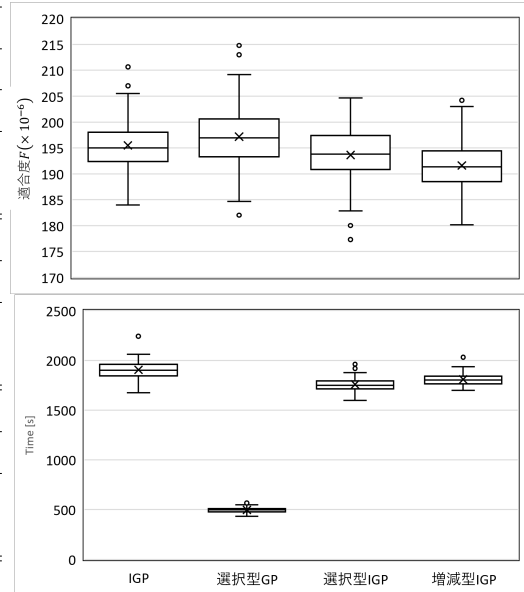


図 4.9: 適合度と計算時間

表 4.11: 最適な手法

	job=10	job=20	job=30
operation=10	選択型 GP	IGP	IGP
operation=20	IGP	増減型 IGP	選択型 GP
operation=30	選択型 GP	IGP	選択型 GP

表 4.11 より、各シナリオで適合度 F の平均値が最良となる手法が異なることがわかる。つまり、シナリオのサイズを基にして適した手法が判断できるような規則性はなく、表 4.2-4.10 を見ても、表 4.3 以外では、4つの手法で求められたメイクスパンや納期遅れに大差がほとんどないことがわかる。しかし、計算時間に関しては、全てのシナリオにおいて個体群選択型 GP が他の 3 手法より圧倒的に早いことがわかる。これは、1 回の解探索に使用する個体数が他よりも少ないからである。つまり、ほとんどのシナリオにおいて、個体群選択型 GP は少ない個体数であっても他の 3 手法と同等の良好な解探索を行うことができ、計算時間の短縮が可能であることがわかる。

個体群選択型 IGP と個体群増減型二重並列構造 GP は 1 回の解探索に IGP と同数の個体を使用しているが、図 4.1-4.9 より若干の計算時間の短縮が可能であることがわかった。これは、それぞれの特徴に適合できる複数の個体群を保存しており、到着するジョブ群の特徴にあわせて個体群を選択することで、最適な解を獲得するための個体に対する遺伝的操作を減らすことができ、計算コストを抑えることができたと考えられる。

また、IGP, 個体群選択型 IGP, 個体群増減型二重並列構造 GP の適合度 F の分布範囲に大きな変化はないように見えるが、個体群選択型 GP に関しては、他の 3 手法に比べて変化が大きいように見える。特に図 4.2 では、他の 3 手法より著しく適合度 F が減少していることがわかる。これは、使用している個体数が少ないことで、広範囲な解探索が難しく、あまり良好ではないスケジュールを作成する DR からの脱却ができなかった可能性がある。

第5章

ルール分析

本章では、動的スケジューリング問題に対して GP によって生成した DR が一般的に使用されている既存 DR と似ているのか、既存 DR にはない特徴を持つ DR についての分析を行う。

5.1. 分析方法

GP では試行錯誤的にターミナルセットとファンクションセットを組み合わせることにより、木構造を作成し DR を生成する。そのため、様々な特徴を持つ DR の生成が可能である。その中には既存 DR に似たものや、全く異なるものが生成されていると考えられるため、GP で生成された DR がどのような特徴を持っているのかを既存 DR との比較によって分析する。新しいジョブ群の到着ごとに生成した DR と既存 DR が似ているか判断するために、以下のような手順で比較を行う。

- DR を用いて優先度を計算
GP で生成した DR と第2章で紹介した8つの既存 DR (EDD, FIFO, SLACK, ATC, SPT, SRPT, CR+SPT, (SL/RPN)+SPT) を用いて、ある動的変化時の処理対象である全てのオペレーションの優先度を計算する。
- 優先度を基に相関係数を計算
それぞれの既存 DR によって計算された各オペレーションの優先度と、GP によって生成した DR を用いた優先度を基に、それぞれの既存 DR と生成した DR のペアの相関係数を求める。
- 生成した DR と既存 DR が似ているかを判断
類似度の判断には、求めた8つの相関係数と閾値 (0.0, 0.6, 0.9) を用いる。ある既存 DR との相関係数が 0.0 以下である場合、各オペレーションに対する GP による優先度と既存 DR による優先度が全体的に大きく異なっていることがわかるため、その既存 DR と生成した DR は似ていないと考えられる。相関係数が 0.9 以上である場合、それぞれから求められた優先度の分布が酷似していることがわかるため、その既存 DR と生成した DR の特徴が似ていると考えられる。閾値に 0.6 を用意した理由は、0.9 ほどの類似性はないが、動的スケジューリング問題に対して一般的に使用されている既存 DR のような特徴を持つ DR

を GP によって生成できているのかを確認するためである。閾値を越えるような相関係数が複数存在する場合は、その中でも最大となる相関係数となった既存 DR が生成した DR に対して一番似ていると考えることとする。

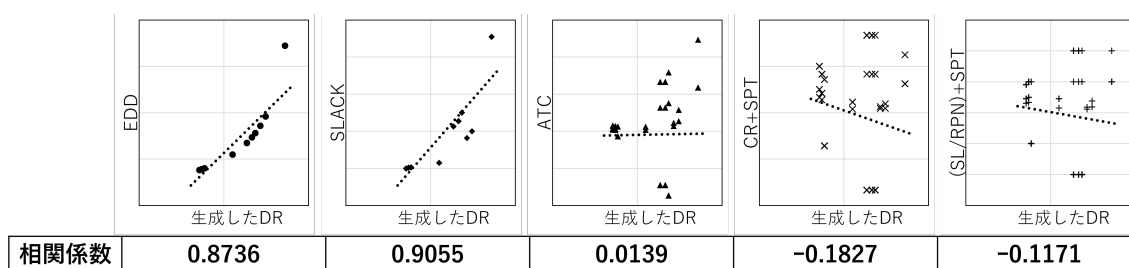


図 5.1: 相関係数の例

例として、図 5.1 を用いて分析方法を紹介する。グラフの縦軸は各既存 DR による優先度を、横軸は生成した DR による優先度を表している。それらの優先度を基に線形近似を行うことで、グラフ下の相関係数を求めることができる。相関係数の中で最大となるのは SLACK であることがわかるが、その値が与えられた閾値よりも大きい場合のみ、生成した DR は SLACK に似ていると判断する。

本論文では、前章の実験結果を基に、適合度 F の平均値が最良となった解探索手法で 100 試行実施し、それにより生成された 10000 個の DR を用いてルール分析を行う。そのため、シナリオのサイズによって分析に使用する DR を生成するための手法が異なるが、GP によって生成された DR の分析を目的としているため、手法の相違は気にしない。

5.2. 分析結果

5.2.1. 生成した DR と既存 DR の類似度

各既存 DR ごとに似ていると判断された回数を記録し、それぞれの割合を求める。全ての相関係数が閾値を越えていない場合は、生成した DR は 8 つの既存 DR にはない新しい特徴を持つ DR であると判断し、“NEW” とする。また、GP による DR の生成は試行錯誤的である為、exp を二重で使用しているなどで、各オペレーションに与えた優先度の最大値と最小値の幅が大きくなりすぎてしまった場合や、例として $\frac{1}{p}$ というような、生成された DR が DR としての性能を持たない場合などで、相関係数を計算できなかった DR は“UNDECIDABLE”として処理する。

閾値ごとの既存 DR との類似割合を図 5.2-5.4 に示す。

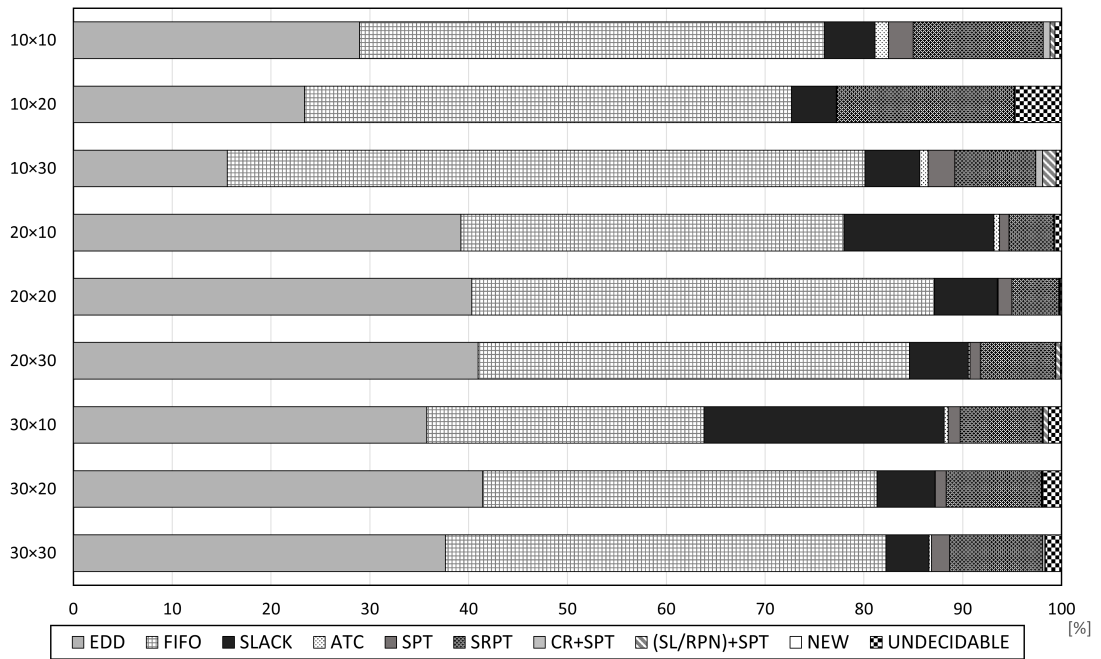


図 5.2: 閾値 0.0 における既存 DR との類似割合

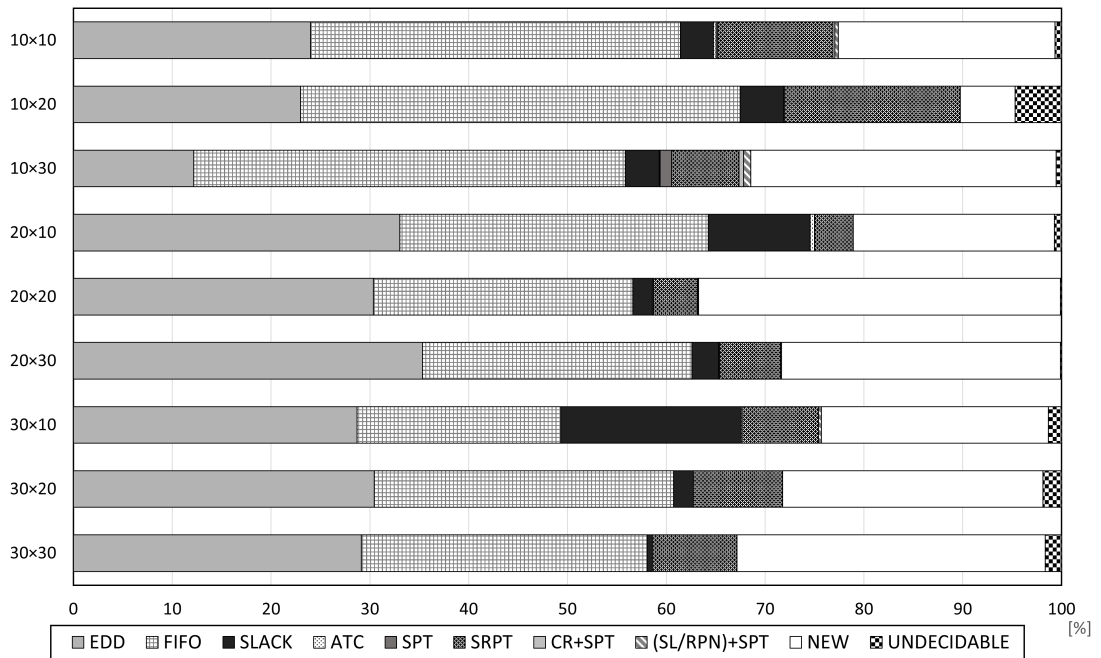


図 5.3: 閾値 0.6 における既存 DR との類似割合

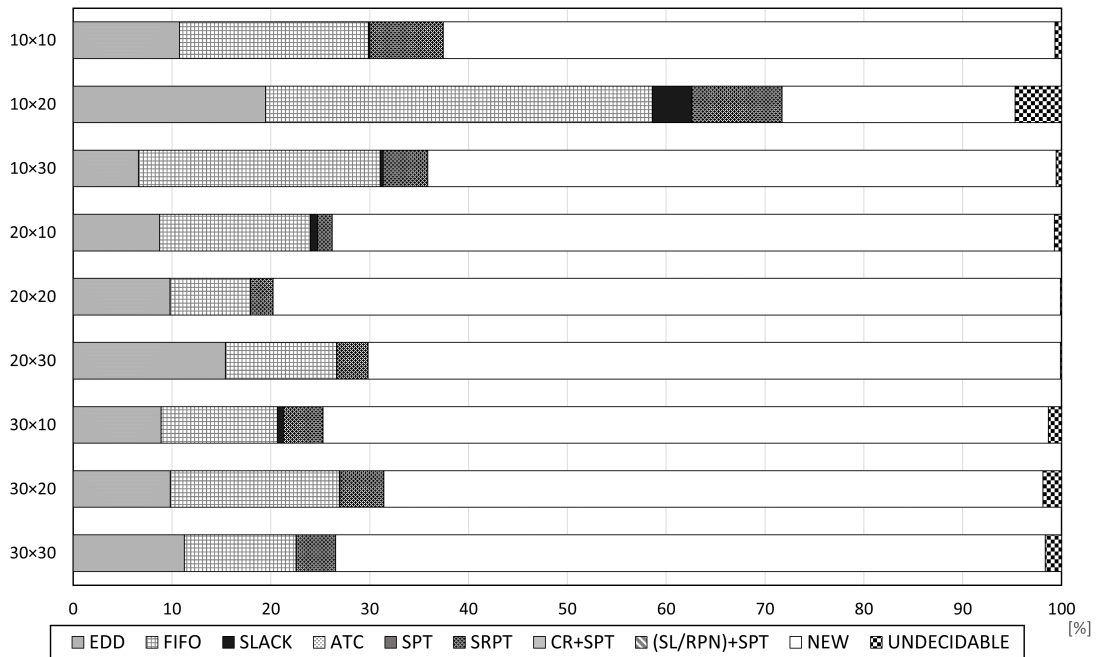


図 5.4: 閾値 0.9 における既存 DR との類似割合

図 5.2 より，“NEW”となる DR の割合がほとんどないため，GP で生成した DR による各オペレーションへの優先度決定と一般的な既存 DR による優先度決定が大きく異なることはなく，特異な DR の生成は少なかったと考えられる．図 5.3 より，全体の約 70% で既存 DR に似た特徴を持つ DR が生成されていることがわかる．これより，一般的に使用されている既存 DR は動的スケジューリング問題に対して良好なスケジューリングを作成できることが裏付けされ，また，GP を用いて既存 DR に似た DR が正しく生成されており，ジョブ群の特徴に適するように様々な DR が使用されていることがわかる．図 5.4 より，既存 DR に酷似した生成 DR は全体の約 30% を占めている．さらにその多くが EDD と FIFO の特徴に似ていることがわかる．この 2 つの DR はとても単純な構造を持っており，動的スケジューリング問題には，単純な DR が効果的であると考えられる．10×20 のシナリオだけが全体の約 70% の生成 DR が既存 DR と酷似している結果となったが，これは複数の既存 DR を適宜に使い分けることで効率的にスケジューリングできるようなシナリオが偶然にも作成され，それによって生成した DR も既存 DR に酷似したものが他のシナリオよりも増えたと考えられる．

5.2.2. 既存 DR にはない特徴を持つ DR の分析

本論文では、全ての相関係数が与えられた閾値を越えていない場合、8つの既存 DR にはない新しい特徴を持つ DR としている。その中でも全ての相関係数が0.0以下である、かつ、最大相関係数が最も小さくなるような DR は、8つの既存 DR から特徴が最もかけ離れていると考えられる。そのような DR を抽出し、構造や特徴について分析する。さらに、新しい特徴を持つ DR と8つの既存 DR を用いて、それぞれで計算した目的関数値 E を表 5.1-5.6 に示す。目的関数値 E が最小となる項目を太文字にしている。

- 10 × 10

$$PI_{ij} = \exp \{ rpt_i - st_i - rpn_i \} \quad (5.1)$$

表 5.1: 使用 DR ごとの目的関数値 E (10 × 10)

使用 DR	GP	EDD	FIFO	SLACK	ATC	SPT	SRPT	CR+SPT	(SL/RPN)+SPT
目的関数値 E	1349	1361	2449	2259	33199	21887	3856	26691	24889

一般的に SRPT のように、ジョブ J_i の残り処理時間 rpt_i を用いる場合、その値が小さいものから処理するようなスケジューリングを作成することが好まれるが、表 5.1 のように生成した DR が最適なスケジューリングができていることがわかる。これは、他に使用している到着時刻 st_i や残りオペレーション数 rpn_i が負の相関をとることで、バランスを保っていると考えられる。

- 10 × 20

$$PI_{ij} = \exp \{ rpt_i \} \quad (5.2)$$

表 5.2: 使用 DR ごとの目的関数値 E (10 × 20)

使用 DR	GP	EDD	FIFO	SLACK	ATC	SPT	SRPT	CR+SPT	(SL/RPN)+SPT
目的関数値 E	57	66	149	66	34531	37132	664	33531	33130

10 × 10 のシナリオと同様に rpt_i が一般的な利用とは異なる。表 5.2 を見ると、GP, EDD, SLACK は目的関数値 E が二桁であるため、納期遅れが発生していないことがわかる。そのため、納期に余裕がある環境では、生成した DR でのスケジューリングが効果を発揮したと考えられる。

- 10 × 30

$$PI_{ij} = \exp \{ rpt_i - 2st_i + 1 \} \quad (5.3)$$

表 5.3: 使用 DR ごとの目的関数値 E (10 × 30)

使用 DR	GP	EDD	FIFO	SLACK	ATC	SPT	SRPT	CR+SPT	(SL/RPN)+SPT
目的関数値 E	1176	1193	1974	1193	28041	58181	2578	22836	21936

これも一般的ではない rpt_i の使い方だが、 st_i に係数 2 がつくことで、10 × 10 の DR よりも st_i の重みが増していると考えられる。10 × 10 と 10 × 30 のシナリオは、正の相関の rpt_i と負の相関の st_i を使用しており、EDD より目的関数値 E を減少させることができているため、EDD よりメイクスパンを減少させることに効果があるのではないかと考えられる。

- 20 × 10

$$PI_{ij} = \max [\exp \{ \bar{p} \}, \exp \{ rpt_i - d_i \}] - d_i \quad (5.4)$$

表 5.4: 使用 DR ごとの目的関数値 E (20 × 10)

使用 DR	GP	EDD	FIFO	SLACK	ATC	SPT	SRPT	CR+SPT	(SL/RPN)+SPT
目的関数値 E	780	2685	7764	4185	259614	266707	19178	226297	237504

新しく到着したジョブ群は確実に $rpt_i \leq d_i$ であるため、 $PI_{ij} = \exp \{ \bar{p} \} - d_i$ で優先度が求められ、これは EDD のように納期 d_i が近いジョブのオペレーションが優先される。ただ、新たなジョブ群の到着時に、既に大幅に納期から遅れているジョブに属するオペレーションには \max の左成分を使用することで高い優先度を与えることができ、納期遅れの削減に効果を発揮していると考えられる。

- 20 × 20

最大の相関係数が 0.0 以下となる DR は存在しなかった。0.0 に最も近づいた値は 0.023 で、生成された DR は以下のような構造である。

$$PI_{ij} = \exp \left\{ d_i \left(\frac{\bar{p}}{rpt_i} - 1 \right) \right\} \quad (5.5)$$

処理途中のジョブと新しく到着したジョブで exp 内の正負が変わる可能性が高いことがわかる。

- 20 × 30

最大の相関係数が 0.0 以下となる DR は存在しなかった。0.0 に最も近づいた値は 0.021 で、生成された DR は以下のような構造である。

$$PI_{ij} = \frac{\bar{p}}{rpt_i^2 (d_i - \bar{p})} \quad (5.6)$$

納期が短くて括弧内が負になるようなジョブが後回しになる可能性が高いため、納期に余裕のあるジョブ群に対して効果的であると考えられる。

- 30 × 10

$$PI_{ij} = \frac{\bar{p}}{st_i + d_i - rpn_i} \quad (5.7)$$

表 5.5: 使用 DR ごとの目的関数値 E (30 × 10)

使用 DR	GP	EDD	FIFO	SLACK	ATC	SPT	SRPT	CR+SPT	(SL/RPN)+SPT
目的関数値 E	16667	20970	37269	21969	417307	498233	45268	416806	307173

DR の分母に使用されている st_i と d_i は FIFO と EDD と同じように使用されており、組み合わせることにより目的関数値 E を減少させることができたと考えられる。 st_i と d_i は時間的要素で、 rpn_i は残りオペレーション数と要素の種類が異なり、多くの場合で $st_i + d_i$ のほうが rpn_i よりも大きな数値になることは明らかである。ただ、 $st_i + d_i$ が同じ値となるような複数のジョブが存在する際には、それぞれの残りオペレーション数によって順序が決定される。

- 30 × 20

$$PI_{ij} = \frac{rpt_i}{\max[\exp\{d_i - 2rpt_i\}, st_i^2]} \quad (5.8)$$

表 5.6: 使用 DR ごとの目的関数値 E (30 × 20)

使用 DR	GP	EDD	FIFO	SLACK	ATC	SPT	SRPT	CR+SPT	(SL/RPN)+SPT
目的関数値 E	5236	13852	40923	32657	1552761	1646289	75349	1309580	913851

処理途中のジョブに用いると st_i は 0 となるため、 $PI_{ij} = \frac{rpt_i}{\exp\{d_i - 2rpt_i\}}$ で優先度を計算することになり、 rpt_i が優先度の計算に大きな影響を与えることがわかる。新しく到着したジョブ群に対しては、EDD や FIFO のどちらかの特徴を用いて優先度を与えることができると考えられる。

- 30 × 30

最大の相関係数が 0.0 以下となる DR は存在しなかった。0.0 に最も近づいた値は 0.017 で、生成された DR は以下のような構造である。

$$PI_{ij} = \frac{\exp\{\bar{p} - rpn_i\}}{d_i + rpn_i - \max[\bar{p}, st_i]} - st_i \quad (5.9)$$

処理途中のジョブに用いると st_i は 0 となるため、 $PI_{ij} = \frac{\exp\{\bar{p} - rpn_i\}}{d_i - \bar{p} + rpn_i}$ で優先度を決定することになる。さらに納期が短いと分母が負の値となるため、納期に余裕のあるジョブ群に対して効果的であると考えられる。

5.2.3. 考察

周期性のある動的スケジューリング問題を用いて、GP によって生成された DR について分析を行った。全てのシナリオにおいて、特異な DR を生成した回数は少なく、既存 DR に似た特徴を持つ DR が動的スケジューリング問題に対して有効的に使用されていることがわかった。また、GP によって生成した DR と既存 DR との相関係数を用いて、8 つの既存 DR の特徴と異なる、新たな特徴を持つ DR についての分析を行った。一般的には考えられないような変数の使用を行う DR もあり、試行錯誤的に解探索を行う GP の有用性が示された。ただ、生成された DR と既存 DR の構造が似ている点があるにも関わらず、相関係数が大きくならずに似ていないと判断されている場合がある。原因として、相関係数を求める際に線形近似を使用しているため、 \exp が DR に使用されていると、相関係数からの判断が難しくなっていると考えられる。

第 6 章

おわりに

本論文では、動的スケジューリング問題に対してジョブ群の特徴と DR の関係に注目して 3 種類の提案手法を構築した。従来研究の IGP と、ジョブ群の特徴によって遺伝的操作を行う個体群を選択する個体群選択型 GP、より広範囲な解探索を目的として個体群内を島モデル化した個体群選択型 IGP、さらなる解探索の効率化を目的として個体群を増減させる個体群増減型二重並列構造 GP に対して数値実験を行い、各シナリオに対する適合度と計算時間の平均値を基に比較を行うことで、動的スケジューリング問題に対してどの手法が効果的であるかを検証した。さらに、GP によって生成された DR と既存 DR を比較し、生成した DR がどの既存 DR にどのくらい似ているのか、動的スケジューリング問題に対してどのような割合で既存 DR に似た DR が生成されているのか、全ての既存 DR に似ていないと判断された DR にはどのような特性があるのかを分析した。

今後の課題として、1 回の解探索に使用する個体数の統一、個体群の増減のバランスをとるパラメータ W の改訂、ルール分析手法の改訂、個体群統合に適した個体選択法への改訂、現実問題への適用が考えられる。

謝辞

本論文作成の全過程を通じて終始理解ある御教授，御指導，御鞭撻を賜りました，西崎一郎教授，林田智弘准教授，関崎真也助教に厚く御礼申し上げます。また，本研究は県立広島大学地域創生学部の広谷大助准教授との共同研究であり，本論文の作成にあたり，御指導，御鞭撻を賜りましたことを，厚く御礼申し上げます。さらに，客観的な視点から研究に対するご助言をいただきました，副指導教員の森川克己准教授にも厚く御礼申し上げます。

2年間を通じて，親切な御助力を頂きました社会情報学研究室の皆様に深く御礼申し上げます。

参考文献

- [1] P. Kaweegitbundit and T. Eguchi, “Flexible job shop scheduling using genetic algorithm and heuristic rules”, *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, 10, pp. 1–18, 2016.
- [2] 江口 透, 前田 直希, 大場 史憲, 村山 長, “ニューラルネットワークを用いた動的ジョブショップスケジューリングのための優先規則作成法”, *日本機械学会論文誌 (C 編)*, 72, pp. 267–274, 2006.
- [3] 林 貴宏, 木村 春彦, 白山 政敏, “遺伝的アルゴリズムによる動的環境下での最適化”, *情報処理学会論文誌:数理モデル化と応用*, 43, pp. 58–69, 2002.
- [4] J. R. Koza and J. P. Rice, *Genetic Programming*, The Movie, Cambridge, MA : MIT Press, 1992.
- [5] 持田 大岳, 林田 智弘, 西崎 一郎, 関崎 真也, 広谷 大助, “遺伝的プログラミングを用いた動的スケジューリング問題のための優先規則生成”, *電気学会研究会資料: システム・制御合同研究会 「制御工学と機械学習の最新動向」*, CT-19-128, ST-19-050, pp. 27–30, 2019.
- [6] S. Limmer and D. Fey, “Comparison of common parallel architectures for the execution of the island model and the global parallelization of evolutionary algorithms”, *Concurrency and Computation*, 29, pp. 1–3, 2017.
- [7] 伊庭 齊志, *遺伝的プログラミング入門*, 東京大学出版会, 2001.
- [8] J. Luo, S. Fujimura, D. Baz and B. Plazolles, “GPU based parallel genetic algorithm for solving an energy efficient dynamic flexible flow shop scheduling problem”, *Journal of Parallel and Distributed Computing*, 133, pp. 244–257, 2018.
- [9] M. Rucinski, D. Izzo and F. Biscani, “On the impact of the migration topology on the Island Model”, *Parallel Computing*, 36, pp. 555–571, 2010.
- [10] D. Watts and S. Strogatz, “Collective dynamics of small-world networks”, *Nature*, 393, pp. 440–442, 1998.

- [11] 前田 伊吹, 林田 智弘, 広谷 大助, 西崎 一郎, 関崎 真也, “周期性のある動的スケジューリング問題のためのエリート個体を用いた並列遺伝的プログラミングの提案,” 2021 IEEE SMC Hiroshima Chapter 若手研究会講演論文集, pp. 2-5, (2021).
- [12] 前田 伊吹, 林田 智弘, 広谷 大助, 西崎 一郎, 関崎 真也, “進化計算手法により得られた周期性のある動的スケジューリング問題のためのルールに対する考察,” 2022 IEEE SMC Hiroshima Chapter 若手研究会講演論文集, pp. 74-80, (2022).