

FLEXSCHE Optimizer 入門ガイド



FLEXSCHE Optimizer とは

数理最適化手法を用いて計画するためのオプション製品

生産スケジューリングに関わる組み合わせ最適化問題を
数理最適化手法を用いて解決します。

ルールベースでは難しい複雑な制約や組合せ爆発に対応でき、
最適解または最良解を保証することができます。

GP(FLEXSCHE本体)が持つ制約やデータ構造をOptimizerが
自動で数理モデル化してくれるため、専門家でなくとも最適化技術
を利用することができます。



納期遅れ最小化再割付けメソッドとは 1

納期遅れがもっとも小さくなる割付けを自動的に探し出すメソッドです。

正しい使い方をすれば、人が考えたルールで計画するよりも最適な計画を作れる可能性があります。

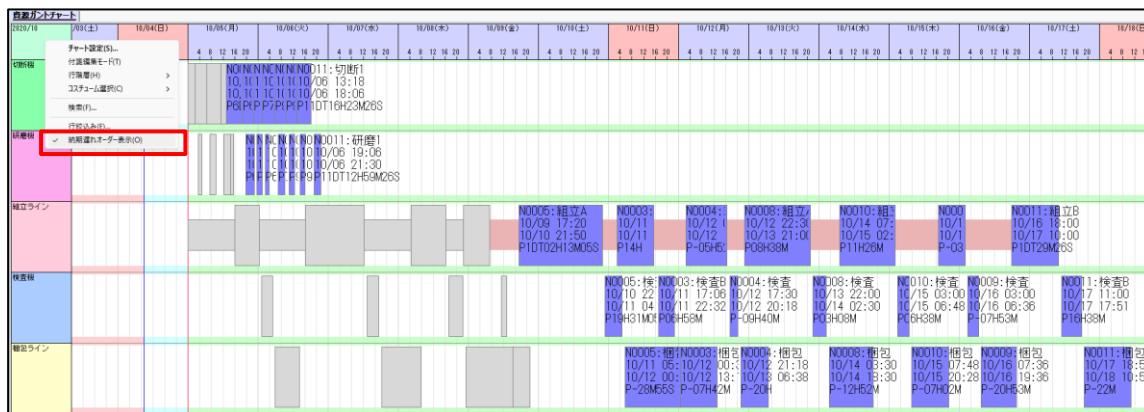
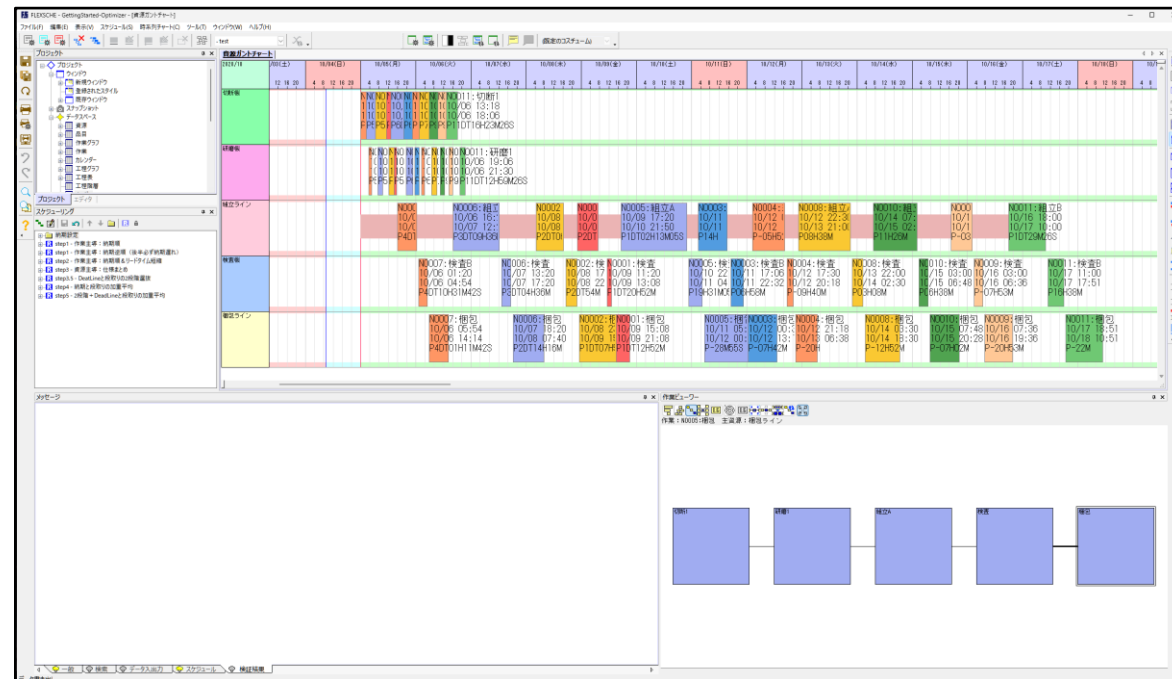
まずは実際に動かしてみましょう。

プロジェクトを確認しよう

サンプル集より
「FLEXSCHE Optimizer 入門ガイド」
を開いてください。

- サンプルData5をベースにより簡略化されたモデル
- 組立ラインに動的段取りがあり、作業効率と納期のバランスが難しい

このようなデータに対し、納期遅れの解消を試みます。

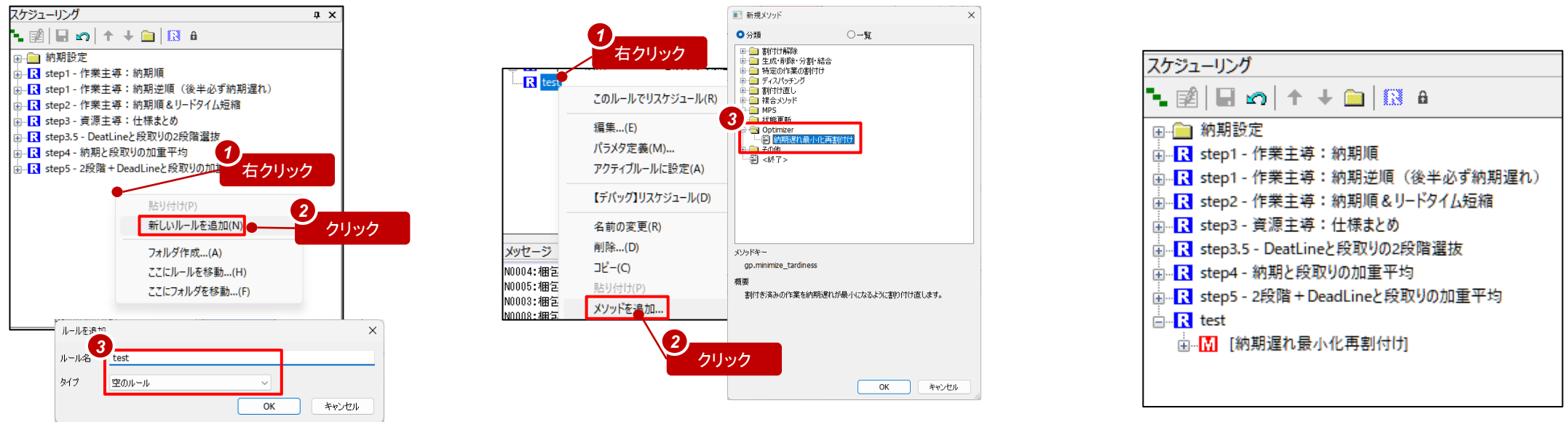


チャートコーナー部を右クリックし「納期遅れオーダー表示」とすると納期に遅れているものが着色されます。
(step1,2,3,,とルールがありますが、step5以外はすべて納期遅れとなります)

step1や2は納期の早い順にディスパッチした結果です。
納期順に並べることは、必ずしも納期遅れを解決しません。

納期遅れ最小化再割付けメソッドを動かしてみよう 1

納期遅れ最小化再割付けメソッドを使って納期遅れを解消してみましょう。



1

ルールを追加

スケジューリングパネルを右クリックし「新しいルールを追加」します。
今回は「test」という名前で作成します。

2

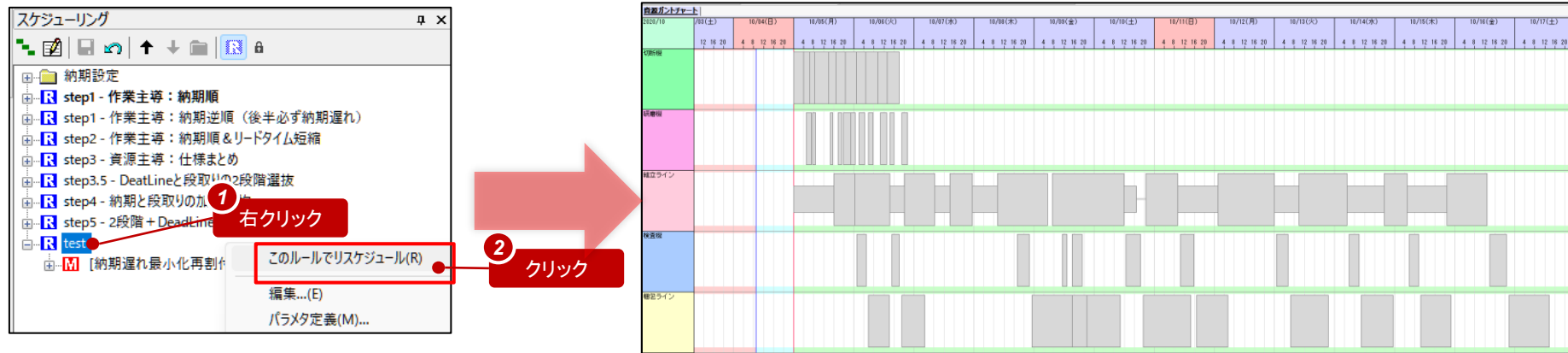
メソッドを追加

さらに「test」を右クリックし、納期遅れ最小化再割付けメソッドを追加します。

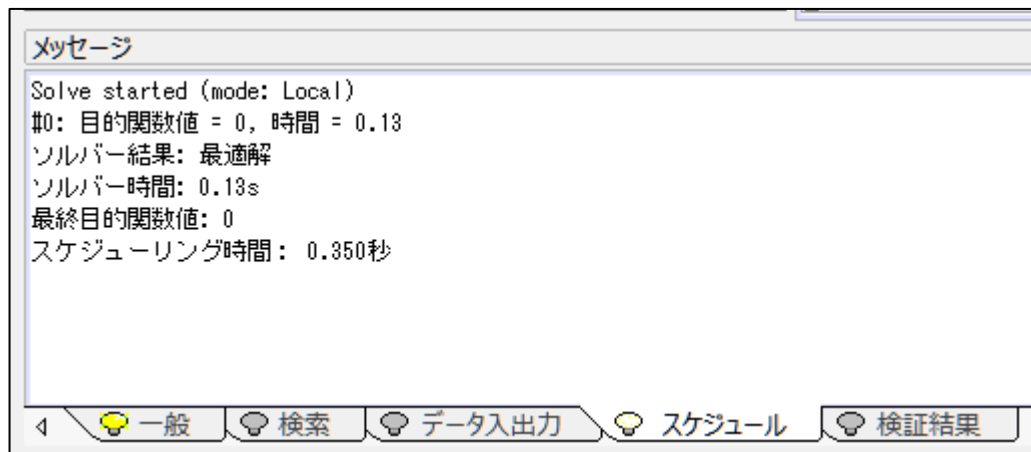
これでルールを作成できました。

納期遅れ最小化再割付けメソッドを動かしてみよう 1

スケジューリングルール「test」を実行してみましょう。



納期遅れが解消されました。



メッセージパネルを見ると

ソルバー結果: 最適解

最終目的関数値: 0

とあります。

納期遅れの無い計画が見つかり、
探索を終了した状態です。

Note – 目的関数値

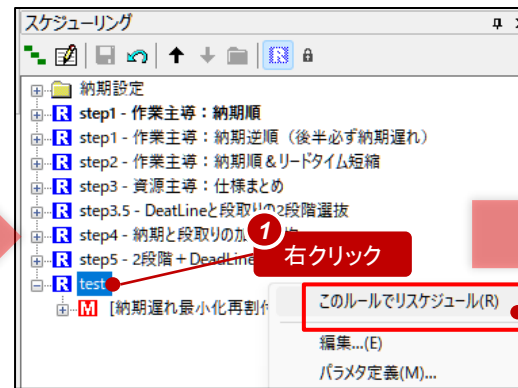
標準では納期遅れ時間の合計です。
0 = 納期遅れがないことを意味します。

納期遅れ最小化再割付けメソッドを動かしてみよう 1

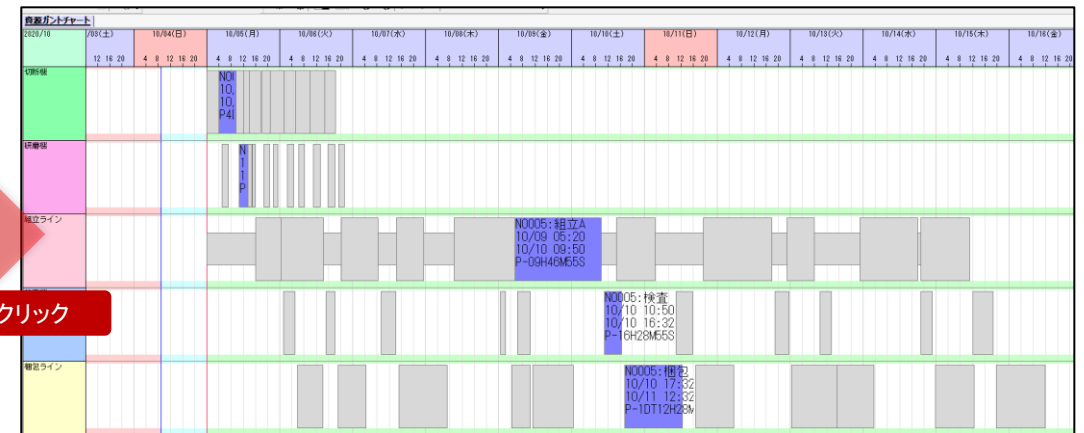
どんな前提でも納期遅れを解消できるわけではありません。
納期遅れが残ってしまうケースを確認してみましょう。



「無理な納期」(*)でリスケジュール



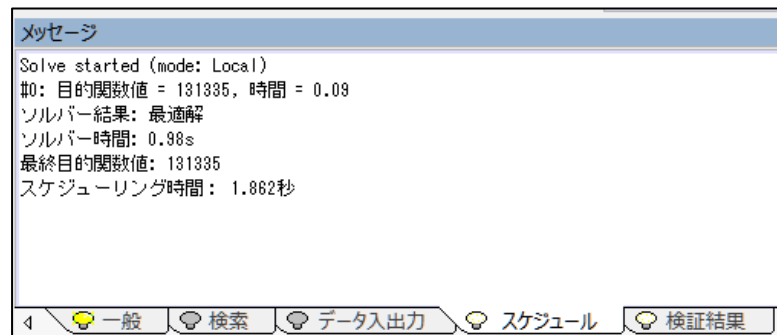
「test」でリスケジュール



納期遅れが発生します。

Note – ルール「無理な納期」

オーダーの納期を書き換えます。



メッセージパネルを見ると

ソルバー結果: 最適解

最終目的関数値: 131335

となりました。

納期遅れは解消できませんでしたが、「最適解」です。
これは、これ以上納期遅れ時間を短くできないことが
保証されていることを意味します。

ソルバー結果

ソルバー結果には以下のような種類があります。

ソルバー結果	意味	主な対応
最適解	実現可能な最適解が見つかりました。	最も良い結果であることが保証されています。これ以上Optimizerを実行する必要はないでしょう。
実行可能解	実現可能な解決策は見つかったものの、最適かどうかはわかりません。	さらに時間をかけてOptimizerを実行すれば、よりよい結果が得られる可能性があります。
実行不可能	この問題は実現不可能であることが証明されました。	Optimizerを実行する上での制約を満たしていません。データを調整する必要があります。(*)
初期解が見つかりませんでした	実行可能か実行不可能か不明です。	実行時間が不足している可能性があります。より長い時間実行するか、データをコンパクトにすれば実行可能となる可能性があります。(*)

(*) 原因不明の場合はお気軽にお問合せ下さい。

納期遅れ最小化再割付けメソッドとは 2

納期遅れ最小化再割付けメソッドは以下のような特徴があります。

- 数理最適化手法を用いたスケジューリングメソッド
- 割付け済み作業が処理対象
- 制約(*)を遵守するすべての割付けの中から、目的関数値がより小さい割付けを自動的に探索して再割付け
- 目的関数(*)のデフォルトは納期遅れ時間(の総和)



Note – 制約

資源制約、工程間制約、作業構造による制約etc.

Note – 目的関数

納期遅れ時間、段取り時間、作業開始/終了日時etc.
作業に特定の数値仕様を設定することで目的関数を変更可能です。
詳しくはオンラインマニュアルを別途ご覧ください。

Note – 納期遅れ時間

≠ 納期遅れ件数 であることに注意してください。

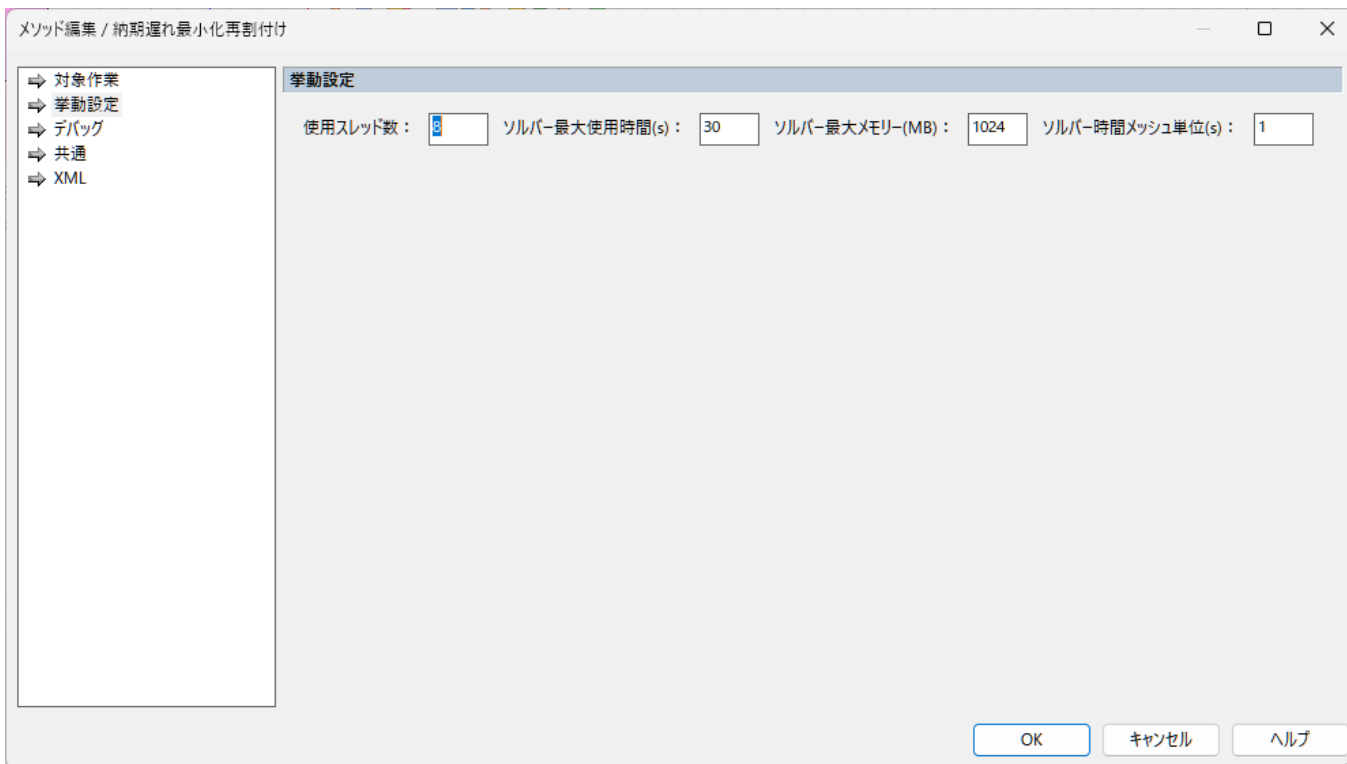
納期遅れ時間: P10H
納期遅れ件数: 5件

>

納期遅れ時間: P20H
納期遅れ件数: 2件

納期遅れ最小化再割付けメソッドとは 2

納期遅れ最小化再割付けメソッドの主な設定項目を以下にご紹介します。



◆使用スレッド数

並行探索するためのスレッド数を設定します。
推奨値は使用するパソコンのCPUの物理コア数です。
※ コア数8以上のPCを使用するのがおすすめです。

◆ソルバー最大使用時間

ソルバー実行の最大時間を秒単位で設定します。
実行結果を見ながら適宜調整するのがよいでしょう。

◆ソルバー最大メモリ

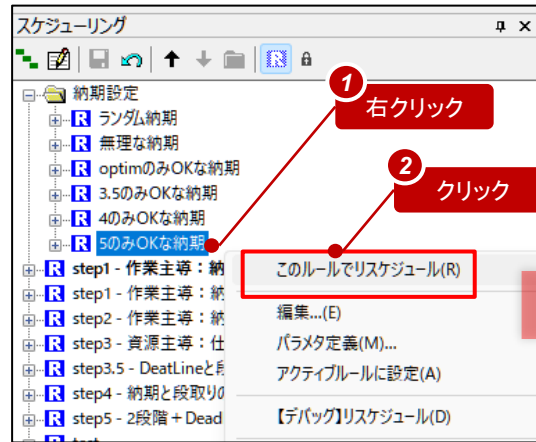
ソルバー実行の最大メモリをMB単位で設定します。

◆ソルバー時間メッシュ単位

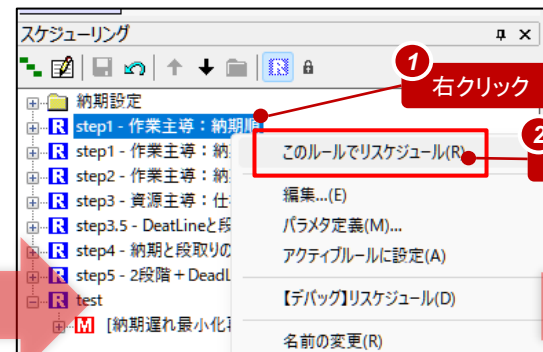
ソルバーの計画時間軸の単位(秒)を設定します。
資源の時間メッシュと合わせてください。

納期遅れ最小化再割付けメソッドを動かしてみよう 2

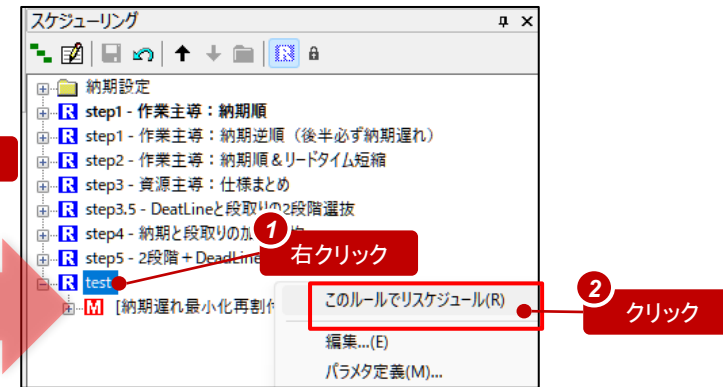
このメソッドは、直前の作業の割付け状況 = 「初期解」が処理時間や結果に影響を及ぼします。
初期解の違いによる影響を確認してみましょう。



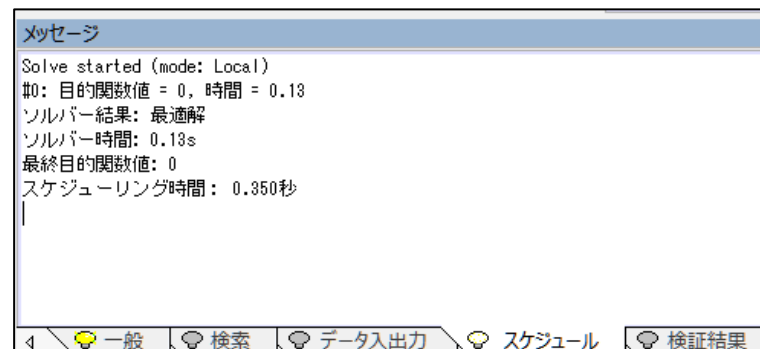
「5のみOKな納期」を実行
(納期設定を初期状態に戻します)



「step1」を実行



「test」を実行



メッセージパネルを見ると納期遅れは解消していますが、
ここで処理時間に注目してください。

ソルバー時間: 0.13s

となりました。(PCスペックによって変わります)

今回は「step1」を初期解としたときの処理時間です。

納期遅れ最小化再割付けメソッドを動かしてみよう 2

step3を初期解としたときの時間と比較してみます。

「step3」を実行

「test」を実行

メッセージ

Solve started (mode: Local)
#0: 目的関数値 = 0, 時間 = 0.13
ソルバー結果: 最適解
ソルバー時間: 0.13s
最終目的関数値: 0
スケジューリング時間: 0.350秒

メッセージ

Solve started (mode: Local)
#0: 目的関数値 = 0, 時間 = 0.31
ソルバー結果: 最適解
ソルバー時間: 0.31s
最終目的関数値: 0
スケジューリング時間: 1.154秒

ソルバー時間: 0.31s

となりました。(PCスペックによって変わります)
step1からのときよりも遅くなりました。
step1の方がstep3よりも良い初期解であると言えそうです。

Note – 制限事項

「資源上に一つでも動的段取りの定義が入っているタスクがある場合、その資源上の全タスクの資源選択は固定される」という制限があります。
モデル・初期解によっては 目的関数値 = 0 にたどり着かない可能性もあります。

Note – 処理時間の揺らぎ

入門ガイド用データでは作業数が少なすぎるため、初期解による違いが分かりにくいです。
場合によっては処理時間が逆転することもあるかもしれません。

納期遅れ最小化再割付けメソッド まとめ

- 数理最適化手法を用いたスケジューリングメソッド
- 本メソッドを利用するためには、初期解が必要
(割付け済み作業だけが対象)
- 初期解を与えると、目的関数値(デフォルトでは納期遅れ時間合計)を少なくするよう、効率的な結果となるよう割付け直す

ルールベースとOptimizer まとめ

最後に、従来のルールベース（作業主導/資源主導ディスパッチングetc.）の計画とOptimizerの違いを以下にまとめました。

	ルールベース	一般的な 数理最適化	Optimizer
スケジューリング手法	ヒューリスティック	数理最適化	数理最適化
計算時間	短い	長い	長い
ロジック実装	ルール作り込み	高度なモデル構築	ルール作りこみ不要 高度なモデル構築不要
最適解	未保証	保証	保証

数理最適化を利用するには本来複雑なモデル構築が必要ですが、OptimizerがGPとソルバーの間で決定変数、制約、目的関数といったデータを橋渡しすることで、誰でも簡単に利用することができます。

Note – 制限事項

OptimizerはGP上に表現された制約すべてに対応していません。
今後の開発によって制限を順次撤廃予定です。
詳しくはオンラインマニュアルの
FLEXSCHE Optimizer > 納期遅れ最小化再割り付けメソッド > 現状対応状況早見表
をご覧ください。

FLEXSCHE Optimizer その他の機能

本入門ガイドでは納期遅れ最小化再割付けメソッドを体験いただきましたが、実はOptimizerの機能は大きく以下の2つに大別されます。

汎用性が高い

メソッドとして使えるOptimizer機能

- **納期遅れ最小化再割付けメソッド**

特定用途向け

takt関数として使えるOptimizer機能

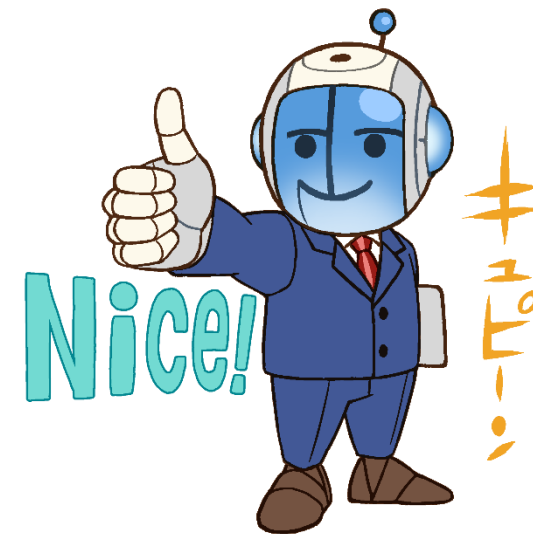
- ループ塗装ラインの周回計画
Optimizer.PaintingLinePlanning
- 板取り計画
Optimizer.RectangularPacking
- スリッター計画
Optimizer.SlitterPlanning

takt関数としてのOptimizer機能については、それぞれGPサンプル集にサンプルデータとして収録されています。興味がありましたら是非ご覧ください。

FLEXSCHE Optimizerを体験してみました。いかがでしたか？
感想や質問等をinfo@flexsche.com までお寄せいただければ幸いです。

なお、本書で紹介した内容は、FLEXSCHE のほんの一部の機能に過ぎません。
FLEXSCHE はこれまで、お客様からの要望に応じて実稼動に必要な様々な機能を追加してきました。
その結果、非常に奥の深いソフトウェアになっています。
また、FLEXSCHEを有効にご活用いただくために、フォロー・サポート体制(*)もごございます。

FLEXSCHE の理解をさらに深めるために、
「FLEXSCHE GP オンラインマニュアル」を活用してください。
オンラインマニュアルのサンプル集には色々なサンプルデータがあります。
これからも拡充していきますので、ご参照下さい。



Note

メーリングリスト、OpenDay、動画トレーニングサービス(有償)など。
詳しくは以下Webページをご覧ください。

<https://www.flexsche.com/support/>

FLEXSCHE Optimizer 入門ガイド Version 25.0

2026年 5月発行
株式会社フレクシェ
〒140-0001 東京都品川区北品川1-19-5 コーストライン品川ビル2F
TEL: 03-6712-9549
FAX: 03-6712-9539
E-Mail: info@flexsche.com
URL: <https://www.flexsche.com/>

本マニュアルの著作権は、株式会社フレクシェにあります。株式会社フレクシェの文書による承諾を得ずに、電子的、機械的、光学的またはその他のいかなる形や手段によっても、本書の一部または全部を無断で複製、翻訳、伝送、写本することはできません。

本書の内容は、予告なく変更されることがあります。