OptSeq Pythonパッケージ チートシート

OptSeq Pythonパッケージ チートシート

概要

OptSeqは、スケジューリング最適化問題を解くためのPythonパッケージです。特に一般化資源制約付きスケジューリングモデルを効率的に解決するよう設計されており、大規模な問題でも短時間で良好な解を得られるようメタヒューリスティクスを活用しています。

インストール

以下のコマンドでOptSeqをインストールできます。

```
pip install optseq
```

公式マニュアルや練習問題はサポートページで参照可能です。

主要クラスとメソッド

Modelクラス

スケジューリング問題全体を定義します。

主なメソッド

- **addActivity**: 作業を追加します。
- **addResource**: 資源を追加します。
- **addTemporal**: 時間制約を追加します。
- **addState**: 状態を追加します。
- **optimize**: 最適化を実行します。
- **write**: 最適化結果をGanttチャート形式で保存します。

使用例

```
from optseq import Model

model = Model(name='Sample Model')
activity = model.addActivity(name='A', duedate=100)
resource = model.addResource(name='Res', capacity=10)
model.addTemporal(pred=activity, succ='sink', delay=5)
model.optimize()
model.write('output.txt')
```

属性

- `pred`: 先行作業。
- `succ`: 後続作業。
- `delay`: 時間のずれ。

使用例

```
temporal = model.addTemporal(pred=activity, succ='sink', delay=10)
```

Parametersクラス

最適化の動作を制御するパラメータを定義します。

主なパラメータ

- `TimeLimit`: 計算時間の上限（秒）。
- `RandomSeed`: 乱数シード。
- `OutputFlag`: ログの詳細度。

使用例

```
model.Params.TimeLimit = 300
model.Params.OutputFlag = 1
```

使用例: 簡単なジョブスケジューリング

以下は、ジョブスケジューリングの簡単な例です。

```
from optseq import Model, Mode

# モデルを作成
model = Model(name='Job Scheduling')

# 作業とモードを追加
job1 = model.addActivity(name='Job1', duedate=10)
mode1 = Mode(name='Mode1', duration=5)
job1.addModes(mode1)

# 資源を追加
machine = model.addResource(name='Machine', capacity=3)
mode1.addResource(machine, requirement=1)

# 最適化を実行
model.optimize()
model.write('schedule.txt')
```

Activityクラス

作業を定義します。

属性

- `name`: 作業名。
- `duedate`: 納期。
- `weight`: 納期遅れペナルティ。
- `modes`: 作業の実行方法（モード）のリスト。

使用例

```
activity = model.addActivity(name='Task1', duedate=50, weight=10)
```

Modeクラス

作業の処理方法を定義します。

主なメソッド

- **addResource**: 必要な資源を指定。
- **addBreak**: 中断可能な設定。
- **addParallel**: 並列実行の設定。

使用例

```
mode = Mode(name='Mode1', duration=5)
mode.addResource(resource, requirement=2)
activity.addModes(mode)
```

Resourceクラス

資源を定義します。

属性

- `name`: 資源名。
- `capacity`: 資源の容量。
- `rhs`: 再生不能資源制約の右辺定数。

使用例

```
resource = model.addResource(name='Machine', capacity=10)
resource.addCapacity(start=0, finish=10, amount=5)
```

Temporalクラス

時間制約を定義します。

デバッグとヒント

1. `print`でモデルやクラスの詳細を確認可能。
2. 出力ファイルを確認してスケジュールや資源使用状況を検証。
3. 小規模な問題でテストしてから拡大する。

さらに詳細な情報や高度な使用例については、公式マニュアルを参照してください。