Structure

- 1. Data preparation
  - 1.1. Import
  - 1.2. Filter
  - 1.3. Enrich
  - 1.4. Random test and train split
  - 1.5. k-fold cross-validation
- 2. Data prediction
  - 2.1 Import Libraries and Data
  - 2.2 Regression models and hyperparameter grids
  - 2.3 Cross-validation with GridSearchCV
  - 2.4 Model performance evaluation
  - 2.5 Select best model
  - 2.6 Predict future values
- 3. Data prescription
  - 3.1 Baseline Scheduling
  - 3.2 Smart Stochastic scheduling (SSS)
    - 3.2.1 Loss function
    - 3.2.2 Stochastic model
- 4. Sensitivity analysis
- 5. GUI

1. Data preparation

1.1. Import

Load the needed data:

- job_logs.csv = Logs of historical Jobs with Timestamps
- weekly_releases.csv = List of Jobs that are executed every week

In [1]:
```
import pandas as pd

# Load historical data into a Pandas DataFrame
logs = pd.read_csv('data/job_logs.csv')

# Load scheduled job data into a Pandas DataFrame
releases = pd.read_csv('data/weekly_releases.csv')
```

1.2 Filter

Filter everything out that is not needed

In [2]:
```
# Keep only the rows in logs whose names appear in releases.
logs = logs[logs['ProcessName'].isin(releases['Name'])]
```

Remove outliners

In [3]:
```
# Calculate the interquartile range (IQR)
Q1 = logs['DateDiff'].quantile(0.25)
Q3 = logs['DateDiff'].quantile(0.75)
IQR = Q3 - Q1

# Remove outliners
filtered_logs = logs[~((logs['DateDiff'] < (Q1 - 1.5 * IQR)) | (logs['DateDiff'] > (Q3 + 1.5 * IQR)))]
```

1.3 Enrich

Adding min, max, mean and median execution time as well as the weekday and calenderweek to the data

In [4]:
```
# Format columns as datetime
filtered_logs.loc[:, 'date'] = pd.to_datetime(filtered_logs['StartTime']).dt.date.copy()
filtered_logs.loc[:, 'time'] = pd.to_datetime(filtered_logs['StartTime']).dt.time.copy()

# Add weekday
filtered_logs.loc[:, 'weekday'] = pd.to_datetime(filtered_logs['StartTime']).dt.dayofweek.copy()

# Add calendar week
filtered_logs.loc[:, 'calendarweek'] = pd.to_datetime(filtered_logs['StartTime']).dt.isocalendar().week.asty

# Rename the 'DateDiff' column to 'ExecTime'
filtered_logs = filtered_logs.rename(columns={'DateDiff': 'ExecTime'})

# Sort DataFrame by process name and date
filtered_logs = filtered_logs.sort_values(by=['ProcessName', 'date'])

# Add last_ExecTime
filtered_logs.loc[:, 'last_ExecTime'] = filtered_logs.groupby('ProcessName')['ExecTime'].shift(1).copy()

# Add mean_ExecTime
filtered_logs.loc[:, 'mean_ExecTime'] = filtered_logs.groupby('ProcessName')['ExecTime'].transform('mean').c

# Add max_ExecTime
filtered_logs.loc[:, 'max_ExecTime'] = filtered_logs.groupby('ProcessName')['ExecTime'].transform('max').cop

# Add min_ExecTime
filtered_logs.loc[:, 'min_ExecTime'] = filtered_logs.groupby('ProcessName')['ExecTime'].transform('min').cop

# Add median_ExecTime
filtered_logs.loc[:, 'median_ExecTime'] = filtered_logs.groupby('ProcessName')['ExecTime'].transform('median

# Replace NaN with zero on all columns
filtered_logs = filtered_logs.fillna(0)

# filtered_logs.head()
```

```
C:\Users\rbyadmin\AppData\Local\Temp\2\ipykernel_4568\1741696903.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  filtered_logs.loc[:, 'date'] = pd.to_datetime(filtered_logs['StartTime']).dt.date.copy()
C:\Users\rbyadmin\AppData\Local\Temp\2\ipykernel_4568\1741696903.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  filtered_logs.loc[:, 'time'] = pd.to_datetime(filtered_logs['StartTime']).dt.time.copy()
C:\Users\rbyadmin\AppData\Local\Temp\2\ipykernel_4568\1741696903.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  filtered_logs.loc[:, 'weekday'] = pd.to_datetime(filtered_logs['StartTime']).dt.dayofweek.copy()
C:\Users\rbyadmin\AppData\Local\Temp\2\ipykernel_4568\1741696903.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  filtered_logs.loc[:, 'calendarweek'] = pd.to_datetime(filtered_logs['StartTime']).dt.isocalendar().week.as
type('int').copy()
```

In [5]:
```
# Save filtered_logs as csv
#filtered_logs.to_csv("filtered_logs.csv", sep='\t')
```

In [6]:
```
from sklearn.preprocessing import LabelEncoder

# Preprocess the data
filtered_logs['StartTime'] = pd.to_datetime(filtered_logs['StartTime'])
filtered_logs['date'] = pd.to_datetime(filtered_logs['date'])
filtered_logs['weekday'] = filtered_logs['StartTime'].dt.weekday
filtered_logs['calenderweek'] = filtered_logs['StartTime'].dt.isocalendar().week
filtered_logs['calenderweek'] = pd.to_datetime(filtered_logs['StartTime']).dt.isocalendar().week.astype(int)

# Label encoding for categorical features
encoder = LabelEncoder()
filtered_logs['ProcessName'] = encoder.fit_transform(filtered_logs['ProcessName'])

# Define the features and target
X = filtered_logs[['ProcessName', 'weekday', 'calenderweek', 'last_ExecTime', 'mean_ExecTime', 'max_ExecTime
y = filtered_logs['ExecTime']
```

1.4 Random test and train split

The idea behind a test and train split is to evaluate the performance of a machine learning model on unseen data. The aim of machine learning is to learn patterns and relationships from a dataset that can be used to make predictions on new, unseen data. To ensure that the model can generalize well to new data, it is important to test its performance on a separate set of data that was not used for training.

Source: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

In [7]:
```
from sklearn.model_selection import train_test_split, GridSearchCV, KFold, cross_val_score
from sklearn.preprocessing import StandardScaler, LabelEncoder

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Standardize the data
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

1.5 k-fold cross-validation

K-fold cross-validation is a model evaluation technique that helps assess a model's performance on unseen data. It is widely used to reduce the risk of overfitting, estimate the generalization error, and select the best model. The dataset is divided into k equally-sized folds (or partitions), where k is a positive integer (in our case five). The model is trained on k-1 folds and validated on the remaining fold. This process is repeated k times, ensuring that each fold is used as the validation set exactly once.

In scikit-learn's KFold class, the parameters n_splits, shuffle, and random_state have the following meanings:

1. **n_splits**: The number of folds (k) to divide the dataset into. This parameter should be set to a positive integer (typically between 5 and 10). Higher values of k result in a lower bias but higher variance in model evaluation, whereas lower values of k can lead to a higher bias but lower variance.
2. **shuffle**: This is a boolean parameter. When set to True, the dataset will be shuffled before splitting into folds. Shuffling is useful when the dataset has an inherent order that might affect the model's performance during cross-validation. When set to False, the dataset will not be shuffled before splitting, and the folds will be created by sequentially selecting data points.

8. **XGBoost**: XGBoost is an optimized implementation of gradient boosting that uses a variety of techniques (such as regularization, early stopping, and parallel processing) to improve the performance and scalability of the algorithm. It is often used in machine learning competitions and has been shown to achieve state-of-the-art results on a wide range of datasets.

Hyperparameter tuning: Hyperparameter tuning is the process of finding the best set of hyperparameters for a machine learning model to optimize its performance. Hyperparameters are configuration variables that govern the training process and cannot be learned directly from the data. They can have a significant impact on the model's performance, and thus, finding the right combination is crucial for building a successful model. In this case the hyperparameter tuning is done with GridSearchCV. Source: https://scikit-learn.org/stable/modules/grid_search.html

In [10]:
```
# every regression model with its hyperparameter grids
models = [
    {'name': 'Linear', 'model': LinearRegression(), 'param_grid': {}}, # no hyperparamters
    {'name': 'Ridge', 'model': Ridge(), 'param_grid': {'alpha': [0.1, 1, 10]}},
    {'name': 'Lasso', 'model': Lasso(), 'param_grid': {'alpha': [0.1, 1, 10]}},
    {'name': 'Elastic Net', 'model': ElasticNet(), 'param_grid': {'alpha': [0.1, 1, 10], 'l1_ratio': [0.1, 0
    {'name': 'Decision Tree', 'model': DecisionTreeRegressor(), 'param_grid': {'max_depth': [None, 5, 10]}},
    {'name': 'Random Forest', 'model': RandomForestRegressor(), 'param_grid': {'n_estimators': [10, 50, 100]
    {'name': 'Gradient Boosting', 'model': GradientBoostingRegressor(), 'param_grid': {'n_estimators': [50,
    {
        'name': 'XGBoost',
        'model': xgb.XGBRegressor(objective='reg:squarederror'),
        'param_grid': {
            'n_estimators': [50, 100, 150],
            'learning_rate': [0.01, 0.1, 0.2],
            'max_depth': [3, 5, 7],
            'subsample': [0.8, 1.0]
        }
    }
]
```

In [11]:
```
# create a dataframe to save results
results = {
    'Model': [],
    'Mean Squared Error': [],
    'R2 Score': [],
    'Learning Time': []
}

results_df = pd.DataFrame(results)
```

2.3 Cross-validation with GridSearchCV

We train and evaluate multiple models using cross-validation and grid search for hyperparameter tuning.

Source: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV

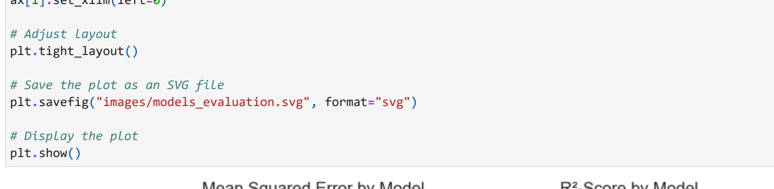
In [12]:
```
models_dict = {}

for m in models:
    print(f"Training and evaluating {m['name']}...")
    grid_search = GridSearchCV(m['model'], m['param_grid'], scoring='neg_mean_squared_error', cv=kfold, verb
    start_time = time.time()
    grid_result = grid_search.fit(X_train, y_train)
    best_model = grid_result.best_estimator_
    y_pred = best_model.predict(X_test)

    # Calculate evaluation metrics
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)

    # Measure the time it takes to train the model
    end_time = time.time()
    learning_time = end_time - start_time

    # Assign values
    key = f"{m['name']}"
    value = best_model
    models_dict[key] = value

    print(f"{m['name']}:")
    print(f"  Mean Squared Error: {mse:.2f}")
    print(f"  R2 Score: {r2:.2f}")
    print(f"  Learning Time: {learning_time:.2f}s")
    print()

    # Create a new DataFrame with the results for this model
    model_results = pd.DataFrame({'Model': key, 'Mean Squared Error': [mse], 'R2 Score': [r2], 'Learning Tim

    # Concatenate the new DataFrame to the existing results DataFrame
    results_df = pd.concat([results_df, model_results], ignore_index=True)

print(results_df)
```

```
Training and evaluating Linear...
Linear:
  Mean Squared Error: 296.06
  R2 Score: 0.73
  Learning Time: 0.49s

Training and evaluating Ridge...
Ridge:
  Mean Squared Error: 296.06
  R2 Score: 0.73
  Learning Time: 0.78s

Training and evaluating Lasso...
Lasso:
  Mean Squared Error: 296.06
  R2 Score: 0.73
  Learning Time: 1.03s

Training and evaluating Elastic Net...
Elastic Net:
  Mean Squared Error: 296.10
  R2 Score: 0.73
  Learning Time: 2.63s

Training and evaluating Decision Tree...
Decision Tree:
  Mean Squared Error: 244.71
  R2 Score: 0.77
  Learning Time: 6.82s

Training and evaluating Random Forest...
Random Forest:
  Mean Squared Error: 237.34
  R2 Score: 0.78
  Learning Time: 706.63s

Training and evaluating Gradient Boosting...
Gradient Boosting:
  Mean Squared Error: 235.58
  R2 Score: 0.78
  Learning Time: 3307.55s

Training and evaluating XGBoost...
XGBoost:
  Mean Squared Error: 235.17
  R2 Score: 0.78
  Learning Time: 1263.44s

               Model  Mean Squared Error  R2 Score  Learning Time
0             Linear          296.063872  0.727279       0.489583
1              Ridge          296.063826  0.727279       0.780494
2              Lasso          296.064447  0.727279       1.034233
3        Elastic Net          296.102691  0.727244       2.629630
4      Decision Tree          244.713156  0.774581       6.819646
5      Random Forest          237.342286  0.781371     706.628110
6  Gradient Boosting          235.579804  0.782995    3307.550284
7            XGBoost          235.165783  0.783376    1263.440491
```

2.4 Model performance evaluation

Visualize the results from the GridSearch.

In [13]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Set plot style
sns.set(style='whitegrid')

# Create subplots
fig, ax = plt.subplots(1, 2, figsize=(8, 3))

# Plot Mean Squared Error
sns.barplot(x='Mean Squared Error', y='Model', data=results_df, ax=ax[0], palette='Blues_d')
ax[0].set_xlabel('Mean Squared Error')
ax[0].set_ylabel('Regression Model')
ax[0].set_title('Mean Squared Error by Model')

# Plot R2 Score
sns.barplot(x='R2 Score', y='Model', data=results_df, ax=ax[1], palette='Greens_d')
ax[1].set_xlabel('R²-Score')
#ax[1].set_ylabel('Model')
ax[1].set_title('R²-Score by Model')
ax[1].set_ylabel('')
ax[1].set_yticklabels([])
ax[1].set_xlim(left=0)

# Adjust layout
plt.tight_layout()

# Save the plot as an SVG file
plt.savefig("images/models_evaluation.svg", format="svg")

# Display the plot
plt.show()
```



2.5 Select best model

In [14]:
```
# Create a new column with the product of "Mean Squared Error" and "R2 Score"
results_df['R2 x 1/MSE'] = results_df['R2 Score'] * 1 / results_df['Mean Squared Error']

# Find the index of the row with the highest product value
idx_max_mse_r2 = results_df['R2 x 1/MSE'].idxmax()

# Get the model name for the highest-scoring row
best_model_name = list({results_df.loc[idx_max_mse_r2, 'Model']})[0]

# Print the model name and the product value for the highest-scoring row
print(f"Model with highest R2 x 1/MSE: " + best_model_name)
print(f"Highest R2 x 1/MSE value: {results_df.loc[idx_max_mse_r2, 'R2 x 1/MSE']:.2f} * 1e8")

# Assign the best model
model = models_dict[best_model_name]

# Assign mse
model_mse = results_df.loc[idx_max_mse_r2, 'Mean Squared Error']
print(model, model_mse)
```

```
Model with highest R2 x 1/MSE: XGBoost
Highest R2 x 1/MSE value: 0.00 * 1e8
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=None, gpu_id=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=0.1, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=7, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             n_estimators=150, n_jobs=None, num_parallel_tree=None,
             predictor=None, random_state=None, ...) 235.16578328450615
```

2.6 Predict future values

In [15]:
```
import datetime

# Load scheduled job data into a Pandas DataFrame
future_jobs = pd.read_csv('data/weekly_releases.csv')

# Sort by Name
future_jobs = future_jobs.sort_values(by=['Name'], ascending=True)

# Label encoding for categorical features
```

```
Set parameter TimeLimit to value 6000
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2|AVX512]
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Academic license - for non-commercial use only - registered to nils.heilemann@stud-mail.uni-wuerzburg.de
Optimize a model with 3274 rows, 756003 columns and 1515024 nonzeros
Model fingerprint: 0x5f094880
Variable types: 0 continuous, 756003 integer (756003 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 3.0000000
Presolve time: 3.38s
Presolved: 3274 rows, 756003 columns, 1515024 nonzeros
Variable types: 0 continuous, 756003 integer (756003 binary)

Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.4801587e-01   0.000000e+00   0.000000e+00      5s

Use crossover to convert LP symmetric solution to basic solution...

Root crossover log...

  754995 PPushes remaining with PInf 0.0000000e+00                 5s
       0 PPushes remaining with PInf 0.0000000e+00                 9s

       0 DPushes remaining with DInf 0.0000000e+00                 9s

  Push phase complete: Pinf 0.0000000e+00, Dinf 8.7902775e-15      9s


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
  754998    2.4801587e-01   0.000000e+00   0.000000e+00      9s

Root relaxation: objective 2.480159e-01, 754998 iterations, 5.33 seconds (3.59 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.24802    0 3272    3.00000    0.24802  91.7%     -   47s
H    0     0                       1.0000000    0.24802  75.2%     -   48s
     0     0    0.24802    0 3272    1.00000    0.24802  75.2%     -   48s

Explored 1 nodes (764117 simplex iterations) in 48.36 seconds (34.77 work units)
Thread count was 4 (of 4 available processors)

Solution count 2: 1 3

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
Optimal solution found.
Minimum number of workers: 1
Worker 1:
        Job 235 (duration: 10 minutes): Start at Day 0, 00:00 - End at Day 0, 00:10
        Job 136 (duration: 10 minutes): Start at Day 0, 00:10 - End at Day 0, 00:20
        Job 228 (duration: 10 minutes): Start at Day 0, 00:20 - End at Day 0, 00:30
        Job 108 (duration: 10 minutes): Start at Day 0, 00:40 - End at Day 0, 00:50
        Job 74 (duration: 10 minutes): Start at Day 0, 00:50 - End at Day 0, 01:00
        Job 206 (duration: 10 minutes): Start at Day 0, 01:00 - End at Day 0, 01:10
        Job 218 (duration: 10 minutes): Start at Day 0, 01:10 - End at Day 0, 01:20
        Job 199 (duration: 10 minutes): Start at Day 0, 01:20 - End at Day 0, 01:30
        Job 200 (duration: 10 minutes): Start at Day 0, 01:30 - End at Day 0, 01:40
        Job 147 (duration: 10 minutes): Start at Day 0, 01:50 - End at Day 0, 02:00
        Job 89 (duration: 10 minutes): Start at Day 0, 02:00 - End at Day 0, 02:10
        Job 195 (duration: 10 minutes): Start at Day 0, 02:10 - End at Day 0, 02:20
        Job 157 (duration: 10 minutes): Start at Day 0, 02:20 - End at Day 0, 02:30
        Job 101 (duration: 10 minutes): Start at Day 0, 02:30 - End at Day 0, 02:40
        Job 223 (duration: 10 minutes): Start at Day 0, 02:40 - End at Day 0, 02:50
        Job 29 (duration: 10 minutes): Start at Day 0, 02:50 - End at Day 0, 03:00
        Job 201 (duration: 10 minutes): Start at Day 0, 03:00 - End at Day 0, 03:10
        Job 221 (duration: 10 minutes): Start at Day 0, 03:10 - End at Day 0, 03:20
        Job 65 (duration: 10 minutes): Start at Day 0, 03:20 - End at Day 0, 03:30
        Job 86 (duration: 10 minutes): Start at Day 0, 03:30 - End at Day 0, 03:40
        Job 181 (duration: 10 minutes): Start at Day 0, 03:40 - End at Day 0, 03:50
        Job 115 (duration: 10 minutes): Start at Day 0, 03:50 - End at Day 0, 04:00
        Job 179 (duration: 10 minutes): Start at Day 0, 04:00 - End at Day 0, 04:10
        Job 161 (duration: 10 minutes): Start at Day 0, 04:10 - End at Day 0, 04:20
        Job 248 (duration: 10 minutes): Start at Day 0, 04:20 - End at Day 0, 04:30
        Job 163 (duration: 10 minutes): Start at Day 0, 04:30 - End at Day 0, 04:40
        Job 77 (duration: 10 minutes): Start at Day 0, 04:40 - End at Day 0, 04:50
        Job 202 (duration: 10 minutes): Start at Day 0, 04:50 - End at Day 0, 05:00
        Job 229 (duration: 10 minutes): Start at Day 0, 05:00 - End at Day 0, 05:10
        Job 15 (duration: 10 minutes): Start at Day 0, 05:10 - End at Day 0, 05:20
        Job 110 (duration: 10 minutes): Start at Day 0, 05:20 - End at Day 0, 05:30
        Job 238 (duration: 10 minutes): Start at Day 0, 05:30 - End at Day 0, 05:40
        Job 171 (duration: 10 minutes): Start at Day 0, 05:50 - End at Day 0, 06:00
        Job 35 (duration: 10 minutes): Start at Day 0, 06:00 - End at Day 0, 06:10
        Job 234 (duration: 10 minutes): Start at Day 0, 06:10 - End at Day 0, 06:20
        Job 139 (duration: 10 minutes): Start at Day 0, 06:20 - End at Day 0, 06:30
        Job 53 (duration: 10 minutes): Start at Day 0, 06:30 - End at Day 0, 06:40
        Job 189 (duration: 10 minutes): Start at Day 0, 06:40 - End at Day 0, 06:50
        Job 69 (duration: 10 minutes): Start at Day 0, 06:50 - End at Day 0, 07:00
        Job 190 (duration: 10 minutes): Start at Day 0, 07:00 - End at Day 0, 07:10
        Job 26 (duration: 10 minutes): Start at Day 0, 07:20 - End at Day 0, 07:30
        Job 210 (duration: 10 minutes): Start at Day 0, 07:30 - End at Day 0, 07:40
        Job 103 (duration: 10 minutes): Start at Day 0, 07:40 - End at Day 0, 07:50
        Job 165 (duration: 10 minutes): Start at Day 0, 07:50 - End at Day 0, 08:00
        Job 247 (duration: 10 minutes): Start at Day 0, 08:00 - End at Day 0, 08:10
        Job 174 (duration: 10 minutes): Start at Day 0, 08:10 - End at Day 0, 08:20
        Job 182 (duration: 10 minutes): Start at Day 0, 08:20 - End at Day 0, 08:30
        Job 170 (duration: 10 minutes): Start at Day 0, 08:30 - End at Day 0, 08:40
        Job 122 (duration: 10 minutes): Start at Day 0, 08:50 - End at Day 0, 09:00
        Job 191 (duration: 10 minutes): Start at Day 0, 09:00 - End at Day 0, 09:10
        Job 178 (duration: 10 minutes): Start at Day 0, 09:10 - End at Day 0, 09:20
        Job 70 (duration: 10 minutes): Start at Day 0, 09:20 - End at Day 0, 09:30
        Job 134 (duration: 10 minutes): Start at Day 0, 09:30 - End at Day 0, 09:40
        Job 245 (duration: 10 minutes): Start at Day 0, 09:40 - End at Day 0, 09:50
        Job 107 (duration: 10 minutes): Start at Day 0, 10:00 - End at Day 0, 10:10
        Job 68 (duration: 10 minutes): Start at Day 0, 10:10 - End at Day 0, 10:20
        Job 185 (duration: 10 minutes): Start at Day 0, 10:20 - End at Day 0, 10:30
        Job 154 (duration: 10 minutes): Start at Day 0, 10:30 - End at Day 0, 10:40
        Job 90 (duration: 10 minutes): Start at Day 0, 10:40 - End at Day 0, 10:50
        Job 216 (duration: 10 minutes): Start at Day 0, 11:00 - End at Day 0, 11:10
        Job 111 (duration: 10 minutes): Start at Day 0, 11:10 - End at Day 0, 11:20
        Job 184 (duration: 10 minutes): Start at Day 0, 11:30 - End at Day 0, 11:40
        Job 93 (duration: 10 minutes): Start at Day 0, 11:40 - End at Day 0, 11:50
        Job 142 (duration: 10 minutes): Start at Day 0, 11:50 - End at Day 0, 12:00
        Job 244 (duration: 10 minutes): Start at Day 0, 12:00 - End at Day 0, 12:10
        Job 61 (duration: 10 minutes): Start at Day 0, 12:10 - End at Day 0, 12:20
        Job 172 (duration: 10 minutes): Start at Day 0, 12:20 - End at Day 0, 12:30
        Job 241 (duration: 10 minutes): Start at Day 0, 12:30 - End at Day 0, 12:40
        Job 187 (duration: 10 minutes): Start at Day 0, 12:40 - End at Day 0, 12:50
        Job 23 (duration: 10 minutes): Start at Day 0, 13:00 - End at Day 0, 13:10
        Job 231 (duration: 10 minutes): Start at Day 0, 13:20 - End at Day 0, 13:30
        Job 20 (duration: 10 minutes): Start at Day 0, 13:30 - End at Day 0, 13:40
        Job 220 (duration: 10 minutes): Start at Day 0, 13:40 - End at Day 0, 13:50
        Job 237 (duration: 10 minutes): Start at Day 0, 13:50 - End at Day 0, 14:00
        Job 186 (duration: 10 minutes): Start at Day 0, 14:00 - End at Day 0, 14:10
        Job 51 (duration: 10 minutes): Start at Day 0, 14:10 - End at Day 0, 14:20
        Job 45 (duration: 10 minutes): Start at Day 0, 14:20 - End at Day 0, 14:30
        Job 141 (duration: 10 minutes): Start at Day 0, 14:30 - End at Day 0, 14:40
        Job 197 (duration: 10 minutes): Start at Day 0, 14:40 - End at Day 0, 14:50
        Job 118 (duration: 10 minutes): Start at Day 0, 14:50 - End at Day 0, 15:00
        Job 192 (duration: 10 minutes): Start at Day 0, 15:00 - End at Day 0, 15:10
        Job 79 (duration: 10 minutes): Start at Day 0, 15:10 - End at Day 0, 15:20
        Job 109 (duration: 10 minutes): Start at Day 0, 15:20 - End at Day 0, 15:30
        Job 119 (duration: 10 minutes): Start at Day 0, 15:40 - End at Day 0, 15:50
        Job 41 (duration: 10 minutes): Start at Day 0, 15:50 - End at Day 0, 16:00
        Job 233 (duration: 10 minutes): Start at Day 0, 16:00 - End at Day 0, 16:10
        Job 67 (duration: 10 minutes): Start at Day 0, 16:10 - End at Day 0, 16:20
        Job 151 (duration: 10 minutes): Start at Day 0, 16:30 - End at Day 0, 16:40
        Job 162 (duration: 10 minutes): Start at Day 0, 16:40 - End at Day 0, 16:50
        Job 159 (duration: 10 minutes): Start at Day 0, 16:50 - End at Day 0, 17:00
        Job 80 (duration: 10 minutes): Start at Day 0, 17:00 - End at Day 0, 17:10
        Job 150 (duration: 10 minutes): Start at Day 0, 17:20 - End at Day 0, 17:30
        Job 215 (duration: 10 minutes): Start at Day 0, 17:30 - End at Day 0, 17:40
        Job 156 (duration: 10 minutes): Start at Day 0, 17:50 - End at Day 0, 18:00
        Job 102 (duration: 10 minutes): Start at Day 0, 18:10 - End at Day 0, 18:20
        Job 82 (duration: 10 minutes): Start at Day 0, 18:20 - End at Day 0, 18:30
        Job 120 (duration: 10 minutes): Start at Day 0, 19:20 - End at Day 0, 19:30
        Job 212 (duration: 10 minutes): Start at Day 0, 19:40 - End at Day 0, 19:50
        Job 246 (duration: 10 minutes): Start at Day 0, 20:20 - End at Day 0, 20:30
        Job 236 (duration: 10 minutes): Start at Day 0, 20:50 - End at Day 0, 21:00
        Job 39 (duration: 10 minutes): Start at Day 0, 21:10 - End at Day 0, 21:20
        Job 22 (duration: 10 minutes): Start at Day 0, 21:20 - End at Day 0, 21:30
        Job 28 (duration: 10 minutes): Start at Day 0, 21:30 - End at Day 0, 21:40
        Job 135 (duration: 10 minutes): Start at Day 0, 22:20 - End at Day 0, 22:30
        Job 125 (duration: 10 minutes): Start at Day 0, 22:30 - End at Day 0, 22:40
        Job 230 (duration: 10 minutes): Start at Day 0, 22:40 - End at Day 0, 22:50
        Job 94 (duration: 10 minutes): Start at Day 0, 22:50 - End at Day 0, 23:00
        Job 114 (duration: 10 minutes): Start at Day 0, 23:10 - End at Day 0, 23:20
        Job 31 (duration: 10 minutes): Start at Day 0, 23:20 - End at Day 0, 23:30
        Job 243 (duration: 10 minutes): Start at Day 0, 23:30 - End at Day 0, 23:40
        Job 63 (duration: 10 minutes): Start at Day 0, 23:50 - End at Day 1, 00:00
        Job 91 (duration: 10 minutes): Start at Day 1, 00:00 - End at Day 1, 00:10
        Job 242 (duration: 10 minutes): Start at Day 1, 00:30 - End at Day 1, 00:40
        Job 198 (duration: 10 minutes): Start at Day 1, 00:40 - End at Day 1, 00:50
        Job 194 (duration: 10 minutes): Start at Day 1, 01:20 - End at Day 1, 01:30
        Job 240 (duration: 10 minutes): Start at Day 1, 01:40 - End at Day 1, 01:50
        Job 50 (duration: 10 minutes): Start at Day 1, 01:50 - End at Day 1, 02:00
        Job 99 (duration: 10 minutes): Start at Day 1, 02:20 - End at Day 1, 02:30
        Job 205 (duration: 10 minutes): Start at Day 1, 02:40 - End at Day 1, 02:50
        Job 44 (duration: 10 minutes): Start at Day 1, 02:50 - End at Day 1, 03:00
        Job 46 (duration: 10 minutes): Start at Day 1, 04:00 - End at Day 1, 04:10
        Job 204 (duration: 10 minutes): Start at Day 1, 04:40 - End at Day 1, 04:50
        Job 95 (duration: 10 minutes): Start at Day 1, 04:50 - End at Day 1, 05:00
        Job 132 (duration: 10 minutes): Start at Day 1, 05:00 - End at Day 1, 05:10
        Job 208 (duration: 10 minutes): Start at Day 1, 05:10 - End at Day 1, 05:20
        Job 173 (duration: 10 minutes): Start at Day 1, 05:50 - End at Day 1, 06:00
        Job 158 (duration: 10 minutes): Start at Day 1, 06:00 - End at Day 1, 06:10
        Job 127 (duration: 10 minutes): Start at Day 1, 06:10 - End at Day 1, 06:20
        Job 105 (duration: 10 minutes): Start at Day 1, 06:20 - End at Day 1, 06:30
        Job 209 (duration: 10 minutes): Start at Day 1, 06:40 - End at Day 1, 06:50
        Job 87 (duration: 10 minutes): Start at Day 1, 07:10 - End at Day 1, 07:20
        Job 219 (duration: 10 minutes): Start at Day 1, 07:40 - End at Day 1, 07:50
        Job 196 (duration: 10 minutes): Start at Day 1, 08:00 - End at Day 1, 08:10
        Job 217 (duration: 10 minutes): Start at Day 1, 09:00 - End at Day 1, 09:10
        Job 224 (duration: 10 minutes): Start at Day 1, 09:30 - End at Day 1, 09:40
        Job 226 (duration: 10 minutes): Start at Day 1, 09:40 - End at Day 1, 09:50
        Job 88 (duration: 10 minutes): Start at Day 1, 09:50 - End at Day 1, 10:00
        Job 222 (duration: 10 minutes): Start at Day 1, 10:00 - End at Day 1, 10:10
        Job 148 (duration: 10 minutes): Start at Day 1, 10:20 - End at Day 1, 10:30
        Job 106 (duration: 10 minutes): Start at Day 1, 10:30 - End at Day 1, 10:40
        Job 180 (duration: 10 minutes): Start at Day 1, 10:40 - End at Day 1, 10:50
        Job 52 (duration: 10 minutes): Start at Day 1, 10:50 - End at Day 1, 11:00
        Job 137 (duration: 10 minutes): Start at Day 1, 11:00 - End at Day 1, 11:10
        Job 152 (duration: 10 minutes): Start at Day 1, 11:20 - End at Day 1, 11:30
        Job 130 (duration: 10 minutes): Start at Day 1, 11:30 - End at Day 1, 11:40
        Job 166 (duration: 10 minutes): Start at Day 1, 11:40 - End at Day 1, 11:50
        Job 57 (duration: 10 minutes): Start at Day 1, 11:50 - End at Day 1, 12:00
        Job 62 (duration: 10 minutes): Start at Day 1, 12:00 - End at Day 1, 12:10
        Job 123 (duration: 10 minutes): Start at Day 1, 14:10 - End at Day 1, 14:20
        Job 116 (duration: 10 minutes): Start at Day 1, 15:20 - End at Day 1, 15:30
        Job 232 (duration: 10 minutes): Start at Day 1, 15:40 - End at Day 1, 15:50
        Job 155 (duration: 10 minutes): Start at Day 1, 16:00 - End at Day 1, 16:10
        Job 16 (duration: 10 minutes): Start at Day 1, 16:10 - End at Day 1, 16:20
        Job 66 (duration: 10 minutes): Start at Day 1, 16:50 - End at Day 1, 17:00
        Job 92 (duration: 10 minutes): Start at Day 1, 17:10 - End at Day 1, 17:20
        Job 177 (duration: 10 minutes): Start at Day 1, 17:20 - End at Day 1, 17:30
        Job 124 (duration: 10 minutes): Start at Day 1, 17:30 - End at Day 1, 17:40
        Job 129 (duration: 10 minutes): Start at Day 1, 17:50 - End at Day 1, 18:00
        Job 239 (duration: 10 minutes): Start at Day 1, 19:00 - End at Day 1, 19:10
        Job 21 (duration: 10 minutes): Start at Day 1, 19:30 - End at Day 1, 19:40
        Job 214 (duration: 10 minutes): Start at Day 1, 19:40 - End at Day 1, 19:50
        Job 164 (duration: 10 minutes): Start at Day 1, 19:50 - End at Day 1, 20:00
        Job 19 (duration: 10 minutes): Start at Day 1, 20:10 - End at Day 1, 20:20
        Job 193 (duration: 10 minutes): Start at Day 1, 20:20 - End at Day 1, 20:30
        Job 168 (duration: 10 minutes): Start at Day 1, 20:40 - End at Day 1, 20:50
        Job 227 (duration: 10 minutes): Start at Day 1, 21:10 - End at Day 1, 21:20
        Job 145 (duration: 10 minutes): Start at Day 1, 21:30 - End at Day 1, 21:40
        Job 81 (duration: 10 minutes): Start at Day 1, 21:40 - End at Day 1, 21:50
        Job 30 (duration: 10 minutes): Start at Day 1, 22:00 - End at Day 1, 22:10
        Job 4 (duration: 10 minutes): Start at Day 1, 22:20 - End at Day 1, 22:30
        Job 167 (duration: 10 minutes): Start at Day 1, 22:50 - End at Day 1, 23:00
        Job 84 (duration: 10 minutes): Start at Day 1, 23:10 - End at Day 1, 23:20
        Job 213 (duration: 10 minutes): Start at Day 2, 00:10 - End at Day 2, 00:20
        Job 8 (duration: 10 minutes): Start at Day 2, 00:30 - End at Day 2, 00:40
        Job 78 (duration: 10 minutes): Start at Day 2, 00:40 - End at Day 2, 00:50
        Job 140 (duration: 10 minutes): Start at Day 2, 01:20 - End at Day 2, 01:30
        Job 76 (duration: 10 minutes): Start at Day 2, 01:40 - End at Day 2, 01:50
        Job 100 (duration: 10 minutes): Start at Day 2, 02:10 - End at Day 2, 02:20
        Job 17 (duration: 10 minutes): Start at Day 2, 02:40 - End at Day 2, 02:50
        Job 104 (duration: 10 minutes): Start at Day 2, 03:10 - End at Day 2, 03:20
        Job 211 (duration: 10 minutes): Start at Day 2, 03:40 - End at Day 2, 03:50
        Job 175 (duration: 10 minutes): Start at Day 2, 03:50 - End at Day 2, 04:00
        Job 121 (duration: 10 minutes): Start at Day 2, 04:00 - End at Day 2, 04:10
        Job 10 (duration: 10 minutes): Start at Day 2, 05:00 - End at Day 2, 05:10
        Job 36 (duration: 10 minutes): Start at Day 2, 05:10 - End at Day 2, 05:20
        Job 203 (duration: 10 minutes): Start at Day 2, 06:00 - End at Day 2, 06:10
        Job 48 (duration: 10 minutes): Start at Day 2, 07:10 - End at Day 2, 07:20
        Job 98 (duration: 10 minutes): Start at Day 2, 07:40 - End at Day 2, 07:50
        Job 176 (duration: 10 minutes): Start at Day 2, 08:20 - End at Day 2, 08:30
        Job 85 (duration: 10 minutes): Start at Day 2, 09:00 - End at Day 2, 09:10
        Job 133 (duration: 10 minutes): Start at Day 2, 09:30 - End at Day 2, 09:40
        Job 27 (duration: 10 minutes): Start at Day 2, 10:40 - End at Day 2, 10:50
        Job 169 (duration: 10 minutes): Start at Day 2, 11:00 - End at Day 2, 11:10
        Job 54 (duration: 10 minutes): Start at Day 2, 11:20 - End at Day 2, 11:30
        Job 138 (duration: 10 minutes): Start at Day 2, 13:20 - End at Day 2, 13:30
        Job 144 (duration: 10 minutes): Start at Day 2, 13:50 - End at Day 2, 14:00
        Job 37 (duration: 10 minutes): Start at Day 2, 14:00 - End at Day 2, 14:10
        Job 71 (duration: 10 minutes): Start at Day 2, 14:10 - End at Day 2, 14:20
        Job 47 (duration: 10 minutes): Start at Day 2, 14:30 - End at Day 2, 14:40
        Job 249 (duration: 10 minutes): Start at Day 2, 15:00 - End at Day 2, 15:10
        Job 25 (duration: 10 minutes): Start at Day 2, 16:00 - End at Day 2, 16:10
        Job 55 (duration: 10 minutes): Start at Day 2, 16:30 - End at Day 2, 16:40
        Job 143 (duration: 10 minutes): Start at Day 2, 17:10 - End at Day 2, 17:20
        Job 75 (duration: 10 minutes): Start at Day 2, 17:20 - End at Day 2, 17:30
        Job 96 (duration: 10 minutes): Start at Day 2, 17:50 - End at Day 2, 18:00
        Job 49 (duration: 10 minutes): Start at Day 2, 18:30 - End at Day 2, 18:40
        Job 153 (duration: 10 minutes): Start at Day 2, 18:50 - End at Day 2, 19:00
        Job 113 (duration: 10 minutes): Start at Day 2, 19:10 - End at Day 2, 19:20
        Job 33 (duration: 10 minutes): Start at Day 2, 20:30 - End at Day 2, 20:40
        Job 183 (duration: 10 minutes): Start at Day 2, 21:20 - End at Day 2, 21:30
        Job 34 (duration: 10 minutes): Start at Day 2, 22:00 - End at Day 2, 22:10
        Job 117 (duration: 10 minutes): Start at Day 2, 23:00 - End at Day 2, 23:10
        Job 38 (duration: 10 minutes): Start at Day 2, 23:50 - End at Day 3, 00:00
        Job 188 (duration: 10 minutes): Start at Day 3, 00:20 - End at Day 3, 00:30
        Job 42 (duration: 10 minutes): Start at Day 3, 00:40 - End at Day 3, 00:50
        Job 149 (duration: 10 minutes): Start at Day 3, 00:50 - End at Day 3, 01:00
        Job 18 (duration: 10 minutes): Start at Day 3, 01:20 - End at Day 3, 01:30
        Job 11 (duration: 10 minutes): Start at Day 3, 01:40 - End at Day 3, 01:50
        Job 43 (duration: 10 minutes): Start at Day 3, 02:20 - End at Day 3, 02:30
        Job 7 (duration: 10 minutes): Start at Day 3, 02:50 - End at Day 3, 03:00
        Job 83 (duration: 10 minutes): Start at Day 3, 04:20 - End at Day 3, 04:30
        Job 56 (duration: 10 minutes): Start at Day 3, 04:40 - End at Day 3, 04:50
        Job 40 (duration: 10 minutes): Start at Day 3, 05:30 - End at Day 3, 05:40
        Job 207 (duration: 10 minutes): Start at Day 3, 06:30 - End at Day 3, 06:40
        Job 146 (duration: 10 minutes): Start at Day 3, 07:10 - End at Day 3, 07:20
        Job 72 (duration: 10 minutes): Start at Day 3, 07:30 - End at Day 3, 07:40
        Job 64 (duration: 10 minutes): Start at Day 3, 09:30 - End at Day 3, 09:40
        Job 59 (duration: 10 minutes): Start at Day 3, 11:50 - End at Day 3, 12:00
        Job 225 (duration: 10 minutes): Start at Day 3, 12:50 - End at Day 3, 13:00
        Job 126 (duration: 10 minutes): Start at Day 3, 13:20 - End at Day 3, 13:30
        Job 128 (duration: 10 minutes): Start at Day 3, 19:00 - End at Day 3, 19:10
        Job 14 (duration: 10 minutes): Start at Day 3, 20:20 - End at Day 3, 20:30
        Job 5 (duration: 10 minutes): Start at Day 3, 20:30 - End at Day 3, 20:40
        Job 160 (duration: 10 minutes): Start at Day 3, 22:20 - End at Day 3, 22:30
        Job 97 (duration: 10 minutes): Start at Day 3, 23:30 - End at Day 3, 23:40
        Job 12 (duration: 10 minutes): Start at Day 4, 00:40 - End at Day 4, 00:50
        Job 60 (duration: 10 minutes): Start at Day 4, 01:00 - End at Day 4, 01:10
        Job 131 (duration: 10 minutes): Start at Day 4, 02:00 - End at Day 4, 02:10
        Job 58 (duration: 10 minutes): Start at Day 4, 03:40 - End at Day 4, 03:50
        Job 13 (duration: 10 minutes): Start at Day 4, 06:10 - End at Day 4, 06:20
        Job 9 (duration: 10 minutes): Start at Day 4, 07:20 - End at Day 4, 07:30
        Job 73 (duration: 10 minutes): Start at Day 4, 15:50 - End at Day 4, 16:00
        Job 32 (duration: 10 minutes): Start at Day 4, 22:10 - End at Day 4, 22:20
        Job 24 (duration: 10 minutes): Start at Day 5, 03:30 - End at Day 5, 03:40
        Job 112 (duration: 10 minutes): Start at Day 5, 05:40 - End at Day 5, 05:50
        Job 0 (duration: 10 minutes): Start at Day 5, 07:10 - End at Day 5, 07:20
        Job 6 (duration: 10 minutes): Start at Day 5, 13:50 - End at Day 5, 14:00
        Job 2 (duration: 10 minutes): Start at Day 5, 14:00 - End at Day 5, 14:10
        Job 3 (duration: 10 minutes): Start at Day 5, 20:40 - End at Day 5, 20:50
        Job 1 (duration: 10 minutes): Start at Day 6, 10:20 - End at Day 6, 10:30
```

3.2 Smart Stochastic scheduling (SSS)

3.2.1 Loss function

We use the Mean Squared Error (MSE) because it's sensitive to large errors and works well for continuous values like runtimes.

loss error = the decision error induced by a prediction

In [22]:

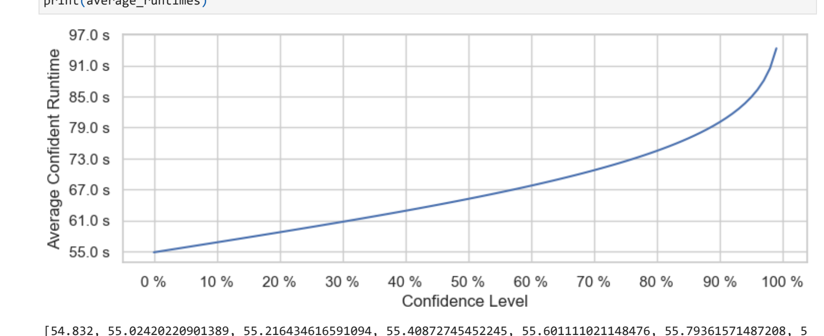
```python
import numpy as np
import scipy.stats as st

def loss_function(data,confidence_level):
    """
    This function can be used to determine confident runtimes for scheduling processes in a calendar,
    taking into account the uncertainty in the predictions based on the model's MSE and the desired confiden
    The resulting confident runtimes represent the upper bounds of the confidence intervals for the predicte
    ensuring that the actual runtimes are likely to be within the specified confidence level below these con

    :param jobs: A dataframe with the future jobs.
    :param confidence_level: Input confidence level.
    :return: a list of the confident runtimes for each job.
    """

    # Assume your predictions and the MSE are stored in the following variables:
    predictions = data["ExecTime"].astype(int)
    mse = model_mse

    # Calculate the standard deviation of the prediction errors (RMSE)
    rmse = np.sqrt(mse)

    # Assign the confidence level
    z_value = st.norm.ppf((1 + confidence_level) / 2)

    # Calculate the confidence interval for each prediction
    confident_runtimes = predictions + z_value * rmse

    return confident_runtimes

# Call function
confident_runtimes = loss_function(future_jobs,0.99)
print(confident_runtimes)
```

```
0      108.500641
1      113.500641
2       57.500641
3       49.500641
4       87.500641
          ...
245     49.500641
246     49.500641
247     49.500641
248     49.500641
249     49.500641
Name: ExecTime, Length: 250, dtype: float64
```

Create a graph to visualize the relation between confidence levels and runtimes.

In [23]:

```python
# Define a range of confidence levels from 50% to 99.999%
confidence_levels = np.linspace(0, 0.99, 100)

# Calculate confident runtimes for each confidence level
runtimes = [loss_function(future_jobs, cl) for cl in confidence_levels]

# Calculate the average confident runtime for each confidence level
average_runtimes = [np.mean(rt) for rt in runtimes]

# Set the size of the plot (width, height) in inches
plt.figure(figsize=(9, 3))

# Plot the sensitivity analysis
plt.plot(confidence_levels * 100, average_runtimes)

# Customize x-axis tick labels
x_ticks = np.arange(0, 101, 10)  # Define tick positions
x_tick_labels = [f'{t} %' for t in x_ticks]  # Add % symbol to tick labels
plt.xticks(x_ticks, x_tick_labels)

# Customize y-axis tick labels
y_ticks = np.arange(np.ceil(np.min(average_runtimes)), np.ceil(np.max(average_runtimes)) + 5, 6)  # Define t
y_tick_labels = [f'{t} s' for t in y_ticks]  # Add 's' unit to tick labels
plt.yticks(y_ticks, y_tick_labels)

plt.xlabel("Confidence Level")
plt.ylabel("Average Confident Runtime")
#plt.title("Sensitivity Analysis of the Loss Function")

# Save the plot as an SVG file
plt.savefig("images/sensitivity_lossfunction.svg", format="svg")

# Display the plot
plt.show()


print(average_runtimes)
```




```
[54.832, 55.02420220901389, 55.216434616591094, 55.40872745452245, 55.601111021148476, 55.79361571487208, 5
5.98627206795725, 56.17911078071268, 56.37216275615946, 56.56545913528609, 56.75903133299654, 56.95291107486
117, 57.147130434784536, 57.34172187371008, 57.536718279486536, 57.732153008028604, 57.92805992591126, 58.12
447345454617, 58.321428616097606, 58.51896108130663, 58.71710721940333, 58.91590415030071, 59.1153897992782
1, 59.31560295437908, 59.51658333276424, 59.71837161428488, 59.921009568559164, 60.12454006586219, 60.329007
18216748, 60.53445627270762, 60.74093405645718, 60.94848870597905, 61.15716994311766, 61.367029141071484, 6
1.57811943343015, 61.79049583082189, 62.00421534588478, 62.21933712735154, 62.43592260412242, 62.65403564029
887, 62.87374270225916, 63.09511303898079, 63.31821887695634, 63.543133631208244, 63.769942134090826, 63.998
72088377595, 64.22955831455731, 64.46254509138113, 64.69777643132636, 64.93535245511909, 65.17537857218646,
65.41796590323939, 65.66323174494013, 65.91130008186758, 66.16230215176456, 66.41637707095335, 66.6736725278
7028, 66.93434555392491, 67.1985633823783, 67.46650440770574, 67.73835926002508, 68.01433201170867, 68.29464
153635391, 68.57952304398647, 68.86922982086605, 69.16403520776127, 69.46423485730968, 69.77014931941609, 7
0.08212701400338, 70.40054766337938, 70.725826272782, 71.05841776831491, 71.39882242785089, 71.747592274396,
72.10533864541878, 72.47274120926245, 72.85055877591229, 73.23964235109754, 73.64095102006029, 74.0555714350
5794, 74.48474194063361, 74.92988273582236, 75.39263399331266, 75.87490461129161, 76.37893539089406, 76.9073
8211835581, 77.46342663540227, 78.05092810966447, 78.67463345481423, 79.3404772198353, 80.05602120187453, 8
0.8311205836187, 81.67897401369075, 82.61786038392064, 83.67418818685007, 84.88827509395853, 86.326477816312
19, 88.11058770973291, 90.50681454656704, 94.33264121240137]
```

3.2.2 Stochastic model

We construct a stochastic model which builds upon the foundational model delineated in chapter 3.1. However, this enhanced model accommodates confidence levels and employs the `loss_function` to schedule tasks according to their associated confident job-runtimes.

Initially, we need to establish a helper function that facilitates the conversion of time from seconds into units of one-sixth of an hour:

In [24]:

```python
def convert_time(data_min):
    data_conv = data_min.copy()

    # Convert seconds to 1/6 hours
    data_conv['ExecTime'] = np.ceil(data_min['ExecTime'] / 60 / 10)
    data_conv['last_ExecTime'] = np.ceil(data_min['last_ExecTime'] / 60 / 10)
    data_conv['mean_ExecTime'] = np.ceil(data_min['mean_ExecTime'] / 60 / 10)
    data_conv['max_ExecTime'] = np.ceil(data_min['max_ExecTime'] / 60 / 10)
    data_conv['min_ExecTime'] = np.ceil(data_min['min_ExecTime'] / 60 / 10)
    data_conv['median_ExecTime'] = np.ceil(data_min['median_ExecTime'] / 60 / 10)
    data_conv['confExecTime'] = np.ceil(data_min['confExecTime'] / 60 / 10)
    return data_conv

#print(convert_time(future_jobs))
```

Following that, we can develop the `smart_stochastic_scheduling` function that incorporates the `confidence_level` as an input parameter:

In [25]:

```python
def smart_stochastic_scheduling(data, confidence_level):

    data_intern = data.copy()
    data_loss_tenth = []
    confExecTime_sum = []

    # Get execution time for the given confidence level
    data_intern['confExecTime'] = loss_function(data_intern,confidence_level)

    # Convert timeformat
    data_loss_tenth = convert_time(data_intern)

    # Create list of tuples with (Id, ExecTime)
    smart_jobs = []
    for index, row in data_loss_tenth.iterrows():
        smart_jobs.append((row['Id'], int(row['confExecTime'])))

    confExecTime_sum = data_intern['confExecTime'].sum()
    print("Ausführungszeit: " + str(confExecTime_sum / 60) + " min")

    # Set variables
    max_timesteps = 7 * 24 * 6  # One week in minutes
    max_workers = 6  # Maximum number of workers allowed
    time_limit = 6000  # Time limit for the solver in seconds
    n_jobs = len(smart_jobs)

    # Create a Gurobi model
    smartModel = Model(env=env)

    # Set a time limit for the solver
    smartModel.setParam(GRB.Param.TimeLimit, time_limit)

    # Variables
    x = {}  # Binary variables to indicate if worker i starts job j at time t
    y = {}  # Binary variables to indicate if worker i is being used

    # Add variables to the model
    for i in range(max_workers):
        y[i] = smartModel.addVar(vtype=GRB.BINARY)
        for j in range(n_jobs):
            for t in range(max_timesteps):
                x[i, j, t] = smartModel.addVar(vtype=GRB.BINARY)

    # Constraints
    # Each job has to be executed once
    for j in range(n_jobs):
        job_length = min(smart_jobs[j][1],max_timesteps)
        constraint_sum = sum(x[i, j, t] for i in range(max_workers) for t in range(max_timesteps) if t <= ma
        smartModel.addConstr(constraint_sum == 1, f"job_{j}_executed_once")

        if max_timesteps - smart_jobs[j][1] < 0:
            print(f"Warning: Execution time for job {j} exceeds max_timesteps.")


    # A worker can only execute one job simultaneously
    for i in range(max_workers):
        for t in range(max_timesteps):
            smartModel.addConstr(sum(x[i, j, t - k] for j in range(n_jobs) for k in range(smart_jobs[j][1])

    # Objective
    smartModel.setObjective(sum(y[i] for i in range(max_workers)), GRB.MINIMIZE)

    # Solve the model
    smartModel.optimize()

    # Error handling
    if smartModel.status == GRB.INFEASIBLE:
        print("Model is infeasible.")
        smartModel.computeIIS()
        smartModel.write("infeasible.ilp")

    # Print results and generate timetables
    if smartModel.status == GRB.OPTIMAL:
        print("Optimal solution found.")
        #return int(smartModel.objVal)
        return int(confExecTime_sum)

    else:
        return None
```

Subsequently, we'll conduct an initial test using our function with a confidence level set at 80%:

In [26]:

```python
# Test function
smart_stochastic_scheduling(future_jobs, 0.8)
```

```
Ausführungszeit: 310.35309141930674 min
Set parameter TimeLimit to value 6000
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2|AVX512]
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Academic license - for non-commercial use only - registered to nils.heilemann@stud-mail.uni-wuerzburg.de
Optimize a model with 6298 rows, 1512006 columns and 3030048 nonzeros
Model fingerprint: 0x4d47dda5
Variable types: 0 continuous, 1512006 integer (1512006 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 6.0000000
Presolve removed 0 rows and 0 columns (presolve time = 5s) ...
Presolve time: 6.70s
Presolved: 6298 rows, 1512006 columns, 3030048 nonzeros
Variable types: 0 continuous, 1512006 integer (1512006 binary)

Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.4801587e-01   0.000000e+00   0.000000e+00     10s

Use crossover to convert LP symmetric solution to basic solution...

Root crossover log...

 1510998 PPushes remaining with PInf 0.0000000e+00                10s
  819814 PPushes remaining with PInf 0.0000000e+00                15s
  251183 PPushes remaining with PInf 0.0000000e+00                20s
       0 PPushes remaining with PInf 0.0000000e+00                22s

       0 DPushes remaining with DInf 0.0000000e+00                22s

  Push phase complete: Pinf 0.0000000e+00, Dinf 4.3723705e-15     22s


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
 1511001    2.4801587e-01   0.000000e+00   0.000000e+00     22s

Root relaxation: objective 2.480159e-01, 1511001 iterations, 14.56 seconds (11.20 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.24802    0 6296    6.00000    0.24802  95.9%     -  158s
H    0     0                       1.0000000    0.24802  75.2%     -  159s
     0     0    0.24802    0 6296    1.00000    0.24802  75.2%     -  159s

Explored 1 nodes (1513446 simplex iterations) in 159.88 seconds (83.79 work units)
Thread count was 4 (of 4 available processors)

Solution count 2: 1 6

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
Optimal solution found.
```

Out[26]: 18621

4. Sensitivity analysis

The following Python script performs a sensitivity analysis on the `smart_stochastic_scheduling` function with different confidence levels.

The script begins by defining a range of confidence levels from 95% to 99.99%, generating 30 equally spaced values between these limits. This range forms the basis of our sensitivity analysis, representing a span from a relatively high confidence level to an extremely high one.

In [ ]:

```python
confidence_levels = np.linspace(0.95, 0.9999, 30) # 1 year to 100 years
results = []
# Perform sensitivity analysis
for confidence_level in confidence_levels:
    min_workers = smart_stochastic_scheduling(future_jobs, confidence_level)

    if min_workers is not None:
        print(f"Minimum number of workers for {confidence_level * 100}% completion: {min_workers}")
        results.append((confidence_level * 100,min_workers))
    else:
        print(f"No optimal solution found for {confidence_level * 100}% completion.")
        results.append((confidence_level * 100,0))
print(results)
```

```
Ausführungszeit: 353.701146224827 min
Set parameter TimeLimit to value 6000
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2|AVX512]
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Academic license - for non-commercial use only - registered to nils.heilemann@stud-mail.uni-wuerzburg.de
Optimize a model with 6298 rows, 1512006 columns and 3030048 nonzeros
Model fingerprint: 0x4d47dda5
Variable types: 0 continuous, 1512006 integer (1512006 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 6.0000000
Presolve removed 0 rows and 0 columns (presolve time = 5s) ...
Presolve time: 6.56s
Presolved: 6298 rows, 1512006 columns, 3030048 nonzeros
Variable types: 0 continuous, 1512006 integer (1512006 binary)

Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.4801587e-01   0.000000e+00   0.000000e+00     10s

Use crossover to convert LP symmetric solution to basic solution...

Root crossover log...

 1510998 PPushes remaining with PInf 0.0000000e+00                10s
 1494577 PPushes remaining with PInf 0.0000000e+00                10s
  819814 PPushes remaining with PInf 0.0000000e+00                15s
  249505 PPushes remaining with PInf 0.0000000e+00                20s
       0 PPushes remaining with PInf 0.0000000e+00                22s

       0 DPushes remaining with DInf 0.0000000e+00                22s

  Push phase complete: Pinf 0.0000000e+00, Dinf 4.3723705e-15     22s


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
 1511001    2.4801587e-01   0.000000e+00   0.000000e+00     22s

Root relaxation: objective 2.480159e-01, 1511001 iterations, 14.59 seconds (11.20 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.24802    0 6296    6.00000    0.24802  95.9%     -  160s
H    0     0                       1.0000000    0.24802  75.2%     -  160s
     0     0    0.24802    0 6296    1.00000    0.24802  75.2%     -  160s

Explored 1 nodes (1513446 simplex iterations) in 161.10 seconds (83.79 work units)
Thread count was 4 (of 4 available processors)

Solution count 2: 1 6

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
Optimal solution found.
Minimum number of workers for 95.0% completion: 21222
Ausführungszeit: 354.6556079066226 min
Set parameter TimeLimit to value 6000
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2|AVX512]
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Academic license - for non-commercial use only - registered to nils.heilemann@stud-mail.uni-wuerzburg.de
Optimize a model with 6298 rows, 1512006 columns and 3030048 nonzeros
Model fingerprint: 0x4d47dda5
Variable types: 0 continuous, 1512006 integer (1512006 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 6.0000000
Presolve removed 0 rows and 0 columns (presolve time = 5s) ...
Presolve time: 6.60s
Presolved: 6298 rows, 1512006 columns, 3030048 nonzeros
Variable types: 0 continuous, 1512006 integer (1512006 binary)

Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.4801587e-01   0.000000e+00   0.000000e+00     10s

Use crossover to convert LP symmetric solution to basic solution...

Root crossover log...

 1510998 PPushes remaining with PInf 0.0000000e+00                10s
 1510151 PPushes remaining with PInf 0.0000000e+00                10s
  835149 PPushes remaining with PInf 0.0000000e+00                15s
  251755 PPushes remaining with PInf 0.0000000e+00                20s
       0 PPushes remaining with PInf 0.0000000e+00                22s

       0 DPushes remaining with DInf 0.0000000e+00                22s

  Push phase complete: Pinf 0.0000000e+00, Dinf 4.3723705e-15     22s


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
 1511001    2.4801587e-01   0.000000e+00   0.000000e+00     23s

Root relaxation: objective 2.480159e-01, 1511001 iterations, 14.74 seconds (11.20 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.24802    0 6296    6.00000    0.24802  95.9%     -  157s
H    0     0                       1.0000000    0.24802  75.2%     -  158s
     0     0    0.24802    0 6296    1.00000    0.24802  75.2%     -  158s

Explored 1 nodes (1513446 simplex iterations) in 158.56 seconds (83.79 work units)
Thread count was 4 (of 4 available processors)

Solution count 2: 1 6

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
Optimal solution found.
Minimum number of workers for 95.17206896551724% completion: 21279
Ausführungszeit: 355.63908645115987 min
Set parameter TimeLimit to value 6000
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2|AVX512]
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Academic license - for non-commercial use only - registered to nils.heilemann@stud-mail.uni-wuerzburg.de
Optimize a model with 6298 rows, 1512006 columns and 3030048 nonzeros
Model fingerprint: 0x4d47dda5
Variable types: 0 continuous, 1512006 integer (1512006 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 6.0000000
Presolve removed 0 rows and 0 columns (presolve time = 5s) ...
Presolve time: 6.52s
Presolved: 6298 rows, 1512006 columns, 3030048 nonzeros
Variable types: 0 continuous, 1512006 integer (1512006 binary)

Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.4801587e-01   0.000000e+00   0.000000e+00     10s

Use crossover to convert LP symmetric solution to basic solution...

Root crossover log...

 1510998 PPushes remaining with PInf 0.0000000e+00                10s
 1494577 PPushes remaining with PInf 0.0000000e+00                10s
  819814 PPushes remaining with PInf 0.0000000e+00                15s
  251131 PPushes remaining with PInf 0.0000000e+00                20s
       0 PPushes remaining with PInf 0.0000000e+00                22s

       0 DPushes remaining with DInf 0.0000000e+00                22s

  Push phase complete: Pinf 0.0000000e+00, Dinf 4.3723705e-15     22s


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
 1511001    2.4801587e-01   0.000000e+00   0.000000e+00     22s

Root relaxation: objective 2.480159e-01, 1511001 iterations, 14.74 seconds (11.20 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.24802    0 6296    6.00000    0.24802  95.9%     -  157s
H    0     0                       1.0000000    0.24802  75.2%     -  157s
     0     0    0.24802    0 6296    1.00000    0.24802  75.2%     -  157s

Explored 1 nodes (1513446 simplex iterations) in 158.09 seconds (83.79 work units)
Thread count was 4 (of 4 available processors)

Solution count 2: 1 6

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
Optimal solution found.
Minimum number of workers for 95.34413793103448% completion: 21338
Ausführungszeit: 356.65364971018647 min
Set parameter TimeLimit to value 6000
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2|AVX512]
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Academic license - for non-commercial use only - registered to nils.heilemann@stud-mail.uni-wuerzburg.de
Optimize a model with 6298 rows, 1512006 columns and 3030048 nonzeros
Model fingerprint: 0x4d47dda5
Variable types: 0 continuous, 1512006 integer (1512006 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 6.0000000
Presolve removed 0 rows and 0 columns (presolve time = 5s) ...
Presolve time: 6.49s
Presolved: 6298 rows, 1512006 columns, 3030048 nonzeros
Variable types: 0 continuous, 1512006 integer (1512006 binary)

Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.4801587e-01   0.000000e+00   0.000000e+00     10s

Use crossover to convert LP symmetric solution to basic solution...

Root crossover log...

 1510998 PPushes remaining with PInf 0.0000000e+00                10s
 1475939 PPushes remaining with PInf 0.0000000e+00                10s
  790147 PPushes remaining with PInf 0.0000000e+00                15s
  230606 PPushes remaining with PInf 0.0000000e+00                20s
       0 PPushes remaining with PInf 0.0000000e+00                22s

       0 DPushes remaining with DInf 0.0000000e+00                22s

  Push phase complete: Pinf 0.0000000e+00, Dinf 4.3723705e-15     22s


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
 1511001    2.4801587e-01   0.000000e+00   0.000000e+00     22s

Root relaxation: objective 2.480159e-01, 1511001 iterations, 14.52 seconds (11.20 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.24802    0 6296    6.00000    0.24802  95.9%     -  157s
H    0     0                       1.0000000    0.24802  75.2%     -  158s
     0     0    0.24802    0 6296    1.00000    0.24802  75.2%     -  158s

Explored 1 nodes (1513446 simplex iterations) in 158.77 seconds (83.79 work units)
Thread count was 4 (of 4 available processors)

Solution count 2: 1 6

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
Optimal solution found.
Minimum number of workers for 95.51620689655172% completion: 21399
Ausführungszeit: 357.7016000399971 min
Set parameter TimeLimit to value 6000
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2|AVX512]
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Academic license - for non-commercial use only - registered to nils.heilemann@stud-mail.uni-wuerzburg.de
Optimize a model with 6298 rows, 1512006 columns and 3030048 nonzeros
Model fingerprint: 0x4d47dda5
Variable types: 0 continuous, 1512006 integer (1512006 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 6.0000000
Presolve removed 0 rows and 0 columns (presolve time = 5s) ...
Presolve time: 6.50s
Presolved: 6298 rows, 1512006 columns, 3030048 nonzeros
Variable types: 0 continuous, 1512006 integer (1512006 binary)

Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.4801587e-01   0.000000e+00   0.000000e+00     10s

Use crossover to convert LP symmetric solution to basic solution...

Root crossover log...

 1510998 PPushes remaining with PInf 0.0000000e+00                10s
 1475939 PPushes remaining with PInf 0.0000000e+00                10s
  800904 PPushes remaining with PInf 0.0000000e+00                15s
  233065 PPushes remaining with PInf 0.0000000e+00                20s
       0 PPushes remaining with PInf 0.0000000e+00                22s

       0 DPushes remaining with DInf 0.0000000e+00                22s

  Push phase complete: Pinf 0.0000000e+00, Dinf 4.3723705e-15     22s


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
 1511001    2.4801587e-01   0.000000e+00   0.000000e+00     22s

Root relaxation: objective 2.480159e-01, 1511001 iterations, 14.54 seconds (11.20 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.24802    0 6296    6.00000    0.24802  95.9%     -  160s
H    0     0                       1.0000000    0.24802  75.2%     -  161s
     0     0    0.24802    0 6296    1.00000    0.24802  75.2%     -  161s

Explored 1 nodes (1513446 simplex iterations) in 161.15 seconds (83.79 work units)
Thread count was 4 (of 4 available processors)

Solution count 2: 1 6

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
Optimal solution found.
Minimum number of workers for 95.68827586206896% completion: 21462
Ausführungszeit: 358.78551157288325 min
Set parameter TimeLimit to value 6000
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2|AVX512]
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Academic license - for non-commercial use only - registered to nils.heilemann@stud-mail.uni-wuerzburg.de
Optimize a model with 6298 rows, 1512006 columns and 3030048 nonzeros
Model fingerprint: 0x4d47dda5
Variable types: 0 continuous, 1512006 integer (1512006 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 6.0000000
Presolve removed 0 rows and 0 columns (presolve time = 5s) ...
Presolve time: 6.54s
Presolved: 6298 rows, 1512006 columns, 3030048 nonzeros
Variable types: 0 continuous, 1512006 integer (1512006 binary)

Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.4801587e-01   0.000000e+00   0.000000e+00     10s

Use crossover to convert LP symmetric solution to basic solution...

Root crossover log...

 1510998 PPushes remaining with PInf 0.0000000e+00                10s
 1494577 PPushes remaining with PInf 0.0000000e+00                10s
  800904 PPushes remaining with PInf 0.0000000e+00                15s
  236806 PPushes remaining with PInf 0.0000000e+00                20s
       0 PPushes remaining with PInf 0.0000000e+00                22s

       0 DPushes remaining with DInf 0.0000000e+00                22s

  Push phase complete: Pinf 0.0000000e+00, Dinf 4.3723705e-15     22s


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
 1511001    2.4801587e-01   0.000000e+00   0.000000e+00     22s

Root relaxation: objective 2.480159e-01, 1511001 iterations, 14.51 seconds (11.20 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.24802    0 6296    6.00000    0.24802  95.9%     -  156s
H    0     0                       1.0000000    0.24802  75.2%     -  157s
     0     0    0.24802    0 6296    1.00000    0.24802  75.2%     -  157s

Explored 1 nodes (1513446 simplex iterations) in 157.62 seconds (83.79 work units)
Thread count was 4 (of 4 available processors)

Solution count 2: 1 6

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
Optimal solution found.
Minimum number of workers for 95.8603448275862% completion: 21527
Ausführungszeit: 359.9082752514463 min
Set parameter TimeLimit to value 6000
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2|AVX512]
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Academic license - for non-commercial use only - registered to nils.heilemann@stud-mail.uni-wuerzburg.de
Optimize a model with 6298 rows, 1512006 columns and 3030048 nonzeros
Model fingerprint: 0x4d47dda5
Variable types: 0 continuous, 1512006 integer (1512006 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 6.0000000
Presolve removed 0 rows and 0 columns (presolve time = 5s) ...
Presolve time: 6.55s
Presolved: 6298 rows, 1512006 columns, 3030048 nonzeros
Variable types: 0 continuous, 1512006 integer (1512006 binary)

Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.4801587e-01   0.000000e+00   0.000000e+00     10s

Use crossover to convert LP symmetric solution to basic solution...

Root crossover log...

 1510998 PPushes remaining with PInf 0.0000000e+00                10s
 1475939 PPushes remaining with PInf 0.0000000e+00                10s
  800904 PPushes remaining with PInf 0.0000000e+00                15s
  228665 PPushes remaining with PInf 0.0000000e+00                20s
       0 PPushes remaining with PInf 0.0000000e+00                22s

       0 DPushes remaining with DInf 0.0000000e+00                22s

  Push phase complete: Pinf 0.0000000e+00, Dinf 4.3723705e-15     22s


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
 1511001    2.4801587e-01   0.000000e+00   0.000000e+00     22s

Root relaxation: objective 2.480159e-01, 1511001 iterations, 14.43 seconds (11.20 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.24802    0 6296    6.00000    0.24802  95.9%     -  156s
H    0     0                       1.0000000    0.24802  75.2%     -  157s
     0     0    0.24802    0 6296    1.00000    0.24802  75.2%     -  157s

Explored 1 nodes (1513446 simplex iterations) in 157.57 seconds (83.79 work units)
Thread count was 4 (of 4 available processors)

Solution count 2: 1 6

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
Optimal solution found.
Minimum number of workers for 96.03241379310344% completion: 21594
Ausführungszeit: 361.07315365946033 min
Set parameter TimeLimit to value 6000
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2|AVX512]
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Academic license - for non-commercial use only - registered to nils.heilemann@stud-mail.uni-wuerzburg.de
Optimize a model with 6298 rows, 1512006 columns and 3030048 nonzeros
Model fingerprint: 0x4d47dda5
Variable types: 0 continuous, 1512006 integer (1512006 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 6.0000000
Presolve removed 0 rows and 0 columns (presolve time = 5s) ...
Presolve time: 6.48s
Presolved: 6298 rows, 1512006 columns, 3030048 nonzeros
Variable types: 0 continuous, 1512006 integer (1512006 binary)

Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.4801587e-01   0.000000e+00   0.000000e+00     10s

Use crossover to convert LP symmetric solution to basic solution...

Root crossover log...

 1510998 PPushes remaining with PInf 0.0000000e+00                10s
 1475939 PPushes remaining with PInf 0.0000000e+00                10s
  790147 PPushes remaining with PInf 0.0000000e+00                15s
  219543 PPushes remaining with PInf 0.0000000e+00                20s
       0 PPushes remaining with PInf 0.0000000e+00                22s

       0 DPushes remaining with DInf 0.0000000e+00                22s

  Push phase complete: Pinf 0.0000000e+00, Dinf 4.3723705e-15     22s


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
 1511001    2.4801587e-01   0.000000e+00   0.000000e+00     22s

Root relaxation: objective 2.480159e-01, 1511001 iterations, 14.46 seconds (11.20 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.24802    0 6296    6.00000    0.24802  95.9%     -  157s
H    0     0                       1.0000000    0.24802  75.2%     -  158s
     0     0    0.24802    0 6296    1.00000    0.24802  75.2%     -  158s

Explored 1 nodes (1513446 simplex iterations) in 158.74 seconds (83.79 work units)
Thread count was 4 (of 4 available processors)

Solution count 2: 1 6

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
Optimal solution found.
Minimum number of workers for 96.20448275862068% completion: 21664
Ausführungszeit: 362.2838483350577 min
Set parameter TimeLimit to value 6000
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2|AVX512]
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Academic license - for non-commercial use only - registered to nils.heilemann@stud-mail.uni-wuerzburg.de
Optimize a model with 6298 rows, 1512006 columns and 3030048 nonzeros
Model fingerprint: 0x4d47dda5
Variable types: 0 continuous, 1512006 integer (1512006 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 6.0000000
Presolve removed 0 rows and 0 columns (presolve time = 5s) ...
Presolve time: 6.48s
Presolved: 6298 rows, 1512006 columns, 3030048 nonzeros
Variable types: 0 continuous, 1512006 integer (1512006 binary)

Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.4801587e-01   0.000000e+00   0.000000e+00     10s

Use crossover to convert LP symmetric solution to basic solution...

Root crossover log...

 1510998 PPushes remaining with PInf 0.0000000e+00                10s
 1464963 PPushes remaining with PInf 0.0000000e+00                10s
  790147 PPushes remaining with PInf 0.0000000e+00                15s
  228665 PPushes remaining with PInf 0.0000000e+00                20s
       0 PPushes remaining with PInf 0.0000000e+00                22s

       0 DPushes remaining with DInf 0.0000000e+00                22s

  Push phase complete: Pinf 0.0000000e+00, Dinf 4.3723705e-15     22s


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
 1511001    2.4801587e-01   0.000000e+00   0.000000e+00     22s

Root relaxation: objective 2.480159e-01, 1511001 iterations, 14.48 seconds (11.20 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.24802    0 6296    6.00000    0.24802  95.9%     -  157s
H    0     0                       1.0000000    0.24802  75.2%     -  158s
     0     0    0.24802    0 6296    1.00000    0.24802  75.2%     -  158s

Explored 1 nodes (1513446 simplex iterations) in 158.72 seconds (83.79 work units)
Thread count was 4 (of 4 available processors)

Solution count 2: 1 6

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
Optimal solution found.
Minimum number of workers for 96.37655172413793% completion: 21737
Ausführungszeit: 363.54458315519514 min
Set parameter TimeLimit to value 6000
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2|AVX512]
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Academic license - for non-commercial use only - registered to nils.heilemann@stud-mail.uni-wuerzburg.de
Optimize a model with 6298 rows, 1512006 columns and 3030048 nonzeros
Model fingerprint: 0x4d47dda5
Variable types: 0 continuous, 1512006 integer (1512006 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 6.0000000
Presolve removed 0 rows and 0 columns (presolve time = 5s) ...
Presolve time: 6.58s
Presolved: 6298 rows, 1512006 columns, 3030048 nonzeros
Variable types: 0 continuous, 1512006 integer (1512006 binary)

Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.4801587e-01   0.000000e+00   0.000000e+00     10s

Use crossover to convert LP symmetric solution to basic solution...

Root crossover log...

 1510998 PPushes remaining with PInf 0.0000000e+00                10s
  835149 PPushes remaining with PInf 0.0000000e+00                15s
  266290 PPushes remaining with PInf 0.0000000e+00                20s
       0 PPushes remaining with PInf 0.0000000e+00                23s

       0 DPushes remaining with DInf 0.0000000e+00                23s

  Push phase complete: Pinf 0.0000000e+00, Dinf 4.3723705e-15     23s


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
 1511001    2.4801587e-01   0.000000e+00   0.000000e+00     23s

Root relaxation: objective 2.480159e-01, 1511001 iterations, 14.78 seconds (11.20 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.24802    0 6296    6.00000    0.24802  95.9%     -  157s
H    0     0                       1.0000000    0.24802  75.2%     -  158s
     0     0    0.24802    0 6296    1.00000    0.24802  75.2%     -  158s

Explored 1 nodes (1513446 simplex iterations) in 158.36 seconds (83.79 work units)
Thread count was 4 (of 4 available processors)

Solution count 2: 1 6

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
Optimal solution found.
Minimum number of workers for 96.54862068965517% completion: 21812
Ausführungszeit: 364.86020864913763 min
Set parameter TimeLimit to value 6000
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2|AVX512]
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Academic license - for non-commercial use only - registered to nils.heilemann@stud-mail.uni-wuerzburg.de
Optimize a model with 6298 rows, 1512006 columns and 3030048 nonzeros
Model fingerprint: 0x4d47dda5
Variable types: 0 continuous, 1512006 integer (1512006 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 6.0000000
Presolve removed 0 rows and 0 columns (presolve time = 5s) ...
Presolve time: 6.56s
Presolved: 6298 rows, 1512006 columns, 3030048 nonzeros
Variable types: 0 continuous, 1512006 integer (1512006 binary)

Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.4801587e-01   0.000000e+00   0.000000e+00     10s

Use crossover to convert LP symmetric solution to basic solution...

Root crossover log...

 1510998 PPushes remaining with PInf 0.0000000e+00                10s
 1509295 PPushes remaining with PInf 0.0000000e+00                10s
  819814 PPushes remaining with PInf 0.0000000e+00                15s
  251079 PPushes remaining with PInf 0.0000000e+00                20s
       0 PPushes remaining with PInf 0.0000000e+00                22s

       0 DPushes remaining with DInf 0.0000000e+00                22s

  Push phase complete: Pinf 0.0000000e+00, Dinf 4.3723705e-15     22s


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
 1511001    2.4801587e-01   0.000000e+00   0.000000e+00     22s

Root relaxation: objective 2.480159e-01, 1511001 iterations, 14.65 seconds (11.20 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.24802    0 6296    6.00000    0.24802  95.9%     -  160s
H    0     0                       1.0000000    0.24802  75.2%     -  161s
     0     0    0.24802    0 6296    1.00000    0.24802  75.2%     -  161s

Explored 1 nodes (1513446 simplex iterations) in 161.47 seconds (83.79 work units)
Thread count was 4 (of 4 available processors)

Solution count 2: 1 6

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
Optimal solution found.
Minimum number of workers for 96.72068965517241% completion: 21891
Ausführungszeit: 366.23633390878507 min
Set parameter TimeLimit to value 6000
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2|AVX512]
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Academic license - for non-commercial use only - registered to nils.heilemann@stud-mail.uni-wuerzburg.de
Optimize a model with 6298 rows, 1512006 columns and 3030048 nonzeros
Model fingerprint: 0x4d47dda5
Variable types: 0 continuous, 1512006 integer (1512006 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 6.0000000
Presolve removed 0 rows and 0 columns (presolve time = 5s) ...
Presolve time: 6.48s
Presolved: 6298 rows, 1512006 columns, 3030048 nonzeros
Variable types: 0 continuous, 1512006 integer (1512006 binary)

Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.4801587e-01   0.000000e+00   0.000000e+00     10s

Use crossover to convert LP symmetric solution to basic solution...

Root crossover log...

 1510998 PPushes remaining with PInf 0.0000000e+00                10s
  835149 PPushes remaining with PInf 0.0000000e+00                15s
  252650 PPushes remaining with PInf 0.0000000e+00                20s
       0 PPushes remaining with PInf 0.0000000e+00                23s

       0 DPushes remaining with DInf 0.0000000e+00                23s

  Push phase complete: Pinf 0.0000000e+00, Dinf 4.3723705e-15     23s


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
 1511001    2.4801587e-01   0.000000e+00   0.000000e+00     23s

Root relaxation: objective 2.480159e-01, 1511001 iterations, 14.72 seconds (11.20 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.24802    0 6296    6.00000    0.24802  95.9%     -  158s
H    0     0                       1.0000000    0.24802  75.2%     -  159s
     0     0    0.24802    0 6296    1.00000    0.24802  75.2%     -  159s

Explored 1 nodes (1513446 simplex iterations) in 159.29 seconds (83.79 work units)
Thread count was 4 (of 4 available processors)

Solution count 2: 1 6

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
Optimal solution found.
Minimum number of workers for 96.89275862068966% completion: 21974
Ausführungszeit: 367.6794953901616 min
Set parameter TimeLimit to value 6000
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2|AVX512]
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Academic license - for non-commercial use only - registered to nils.heilemann@stud-mail.uni-wuerzburg.de
Optimize a model with 6298 rows, 1512006 columns and 3030048 nonzeros
Model fingerprint: 0x4d47dda5
Variable types: 0 continuous, 1512006 integer (1512006 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 6.0000000
Presolve removed 0 rows and 0 columns (presolve time = 5s) ...
Presolve time: 6.49s
Presolved: 6298 rows, 1512006 columns, 3030048 nonzeros
Variable types: 0 continuous, 1512006 integer (1512006 binary)

Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.4801587e-01   0.000000e+00   0.000000e+00     10s

Use crossover to convert LP symmetric solution to basic solution...

Root crossover log...

 1510998 PPushes remaining with PInf 0.0000000e+00                10s
 1475939 PPushes remaining with PInf 0.0000000e+00                10s
  819814 PPushes remaining with PInf 0.0000000e+00                15s
  251235 PPushes remaining with PInf 0.0000000e+00                20s
       0 PPushes remaining with PInf 0.0000000e+00                22s

       0 DPushes remaining with DInf 0.0000000e+00                22s

  Push phase complete: Pinf 0.0000000e+00, Dinf 4.3723705e-15     22s


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
 1511001    2.4801587e-01   0.000000e+00   0.000000e+00     23s

Root relaxation: objective 2.480159e-01, 1511001 iterations, 14.82 seconds (11.20 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.24802    0 6296    6.00000    0.24802  95.9%     -  158s
H    0     0                       1.0000000    0.24802  75.2%     -  158s
     0     0    0.24802    0 6296    1.00000    0.24802  75.2%     -  158s

Explored 1 nodes (1513446 simplex iterations) in 159.06 seconds (83.79 work units)
Thread count was 4 (of 4 available processors)

Solution count 2: 1 6

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
```

```
Optimal solution found.
Minimum number of workers for 97.06482758620689% completion: 22060
Ausführungszeit: 369.1973757832801 min
Set parameter TimeLimit to value 6000
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2|AVX512]
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Academic license - for non-commercial use only - registered to nils.heilemann@stud-mail.uni-wuerzburg.de
Optimize a model with 6298 rows, 1512006 columns and 3030048 nonzeros
Model fingerprint: 0x4d47dda5
Variable types: 0 continuous, 1512006 integer (1512006 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 6.0000000
Presolve removed 0 rows and 0 columns (presolve time = 5s) ...
Presolve time: 6.51s
Presolved: 6298 rows, 1512006 columns, 3030048 nonzeros
Variable types: 0 continuous, 1512006 integer (1512006 binary)

Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.4801587e-01   0.000000e+00   0.000000e+00     10s

Use crossover to convert LP symmetric solution to basic solution...

Root crossover log...

 1510998 PPushes remaining with PInf 0.0000000e+00                10s
 1485750 PPushes remaining with PInf 0.0000000e+00                10s
  800904 PPushes remaining with PInf 0.0000000e+00                15s
  241180 PPushes remaining with PInf 0.0000000e+00                20s
       0 PPushes remaining with PInf 0.0000000e+00                22s

       0 DPushes remaining with DInf 0.0000000e+00                22s

  Push phase complete: Pinf 0.0000000e+00, Dinf 4.3723705e-15     22s


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
 1511001    2.4801587e-01   0.000000e+00   0.000000e+00     22s

Root relaxation: objective 2.480159e-01, 1511001 iterations, 14.55 seconds (11.20 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.24802    0 6296    6.00000    0.24802  95.9%     -  156s
H    0     0                         1.0000000    0.24802  75.2%     -  157s
     0     0    0.24802    0 6296    1.00000    0.24802  75.2%     -  157s

Explored 1 nodes (1513446 simplex iterations) in 157.57 seconds (83.79 work units)
Thread count was 4 (of 4 available processors)

Solution count 2: 1 6

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
Optimal solution found.
Minimum number of workers for 97.23689655172414% completion: 22151
Ausführungszeit: 370.79909198766256 min
Set parameter TimeLimit to value 6000
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2|AVX512]
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Academic license - for non-commercial use only - registered to nils.heilemann@stud-mail.uni-wuerzburg.de
Optimize a model with 6298 rows, 1512006 columns and 3030048 nonzeros
Model fingerprint: 0x4d47dda5
Variable types: 0 continuous, 1512006 integer (1512006 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 6.0000000
Presolve removed 0 rows and 0 columns (presolve time = 5s) ...
Presolve time: 6.62s
Presolved: 6298 rows, 1512006 columns, 3030048 nonzeros
Variable types: 0 continuous, 1512006 integer (1512006 binary)

Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.4801587e-01   0.000000e+00   0.000000e+00     10s

Use crossover to convert LP symmetric solution to basic solution...

Root crossover log...

 1510998 PPushes remaining with PInf 0.0000000e+00                10s
 1504980 PPushes remaining with PInf 0.0000000e+00                10s
  819814 PPushes remaining with PInf 0.0000000e+00                15s
  245361 PPushes remaining with PInf 0.0000000e+00                20s
       0 PPushes remaining with PInf 0.0000000e+00                22s

       0 DPushes remaining with DInf 0.0000000e+00                22s

  Push phase complete: Pinf 0.0000000e+00, Dinf 4.3723705e-15     22s


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
 1511001    2.4801587e-01   0.000000e+00   0.000000e+00     22s

Root relaxation: objective 2.480159e-01, 1511001 iterations, 14.55 seconds (11.20 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.24802    0 6296    6.00000    0.24802  95.9%     -  157s
H    0     0                         1.0000000    0.24802  75.2%     -  158s
     0     0    0.24802    0 6296    1.00000    0.24802  75.2%     -  158s

Explored 1 nodes (1513446 simplex iterations) in 158.90 seconds (83.79 work units)
Thread count was 4 (of 4 available processors)

Solution count 2: 1 6

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
Optimal solution found.
Minimum number of workers for 97.40896551724137% completion: 22247
Ausführungszeit: 372.49558028072136 min
Set parameter TimeLimit to value 6000
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2|AVX512]
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Academic license - for non-commercial use only - registered to nils.heilemann@stud-mail.uni-wuerzburg.de
Optimize a model with 6298 rows, 1512006 columns and 3030048 nonzeros
Model fingerprint: 0x4d47dda5
Variable types: 0 continuous, 1512006 integer (1512006 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 6.0000000
Presolve removed 0 rows and 0 columns (presolve time = 5s) ...
Presolve time: 6.55s
Presolved: 6298 rows, 1512006 columns, 3030048 nonzeros
Variable types: 0 continuous, 1512006 integer (1512006 binary)

Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.4801587e-01   0.000000e+00   0.000000e+00     10s

Use crossover to convert LP symmetric solution to basic solution...

Root crossover log...

 1510998 PPushes remaining with PInf 0.0000000e+00                10s
 1475939 PPushes remaining with PInf 0.0000000e+00                10s
  800904 PPushes remaining with PInf 0.0000000e+00                15s
  230606 PPushes remaining with PInf 0.0000000e+00                20s
       0 PPushes remaining with PInf 0.0000000e+00                22s

       0 DPushes remaining with DInf 0.0000000e+00                22s

  Push phase complete: Pinf 0.0000000e+00, Dinf 4.3723705e-15     22s


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
 1511001    2.4801587e-01   0.000000e+00   0.000000e+00     22s

Root relaxation: objective 2.480159e-01, 1511001 iterations, 14.42 seconds (11.20 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.24802    0 6296    6.00000    0.24802  95.9%     -  156s
H    0     0                         1.0000000    0.24802  75.2%     -  157s
     0     0    0.24802    0 6296    1.00000    0.24802  75.2%     -  157s

Explored 1 nodes (1513446 simplex iterations) in 157.64 seconds (83.79 work units)
Thread count was 4 (of 4 available processors)

Solution count 2: 1 6

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
Optimal solution found.
Minimum number of workers for 97.58103448275862% completion: 22349
Ausführungszeit: 374.30012110721276 min
Set parameter TimeLimit to value 6000
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2|AVX512]
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Academic license - for non-commercial use only - registered to nils.heilemann@stud-mail.uni-wuerzburg.de
Optimize a model with 6298 rows, 1512006 columns and 3030048 nonzeros
Model fingerprint: 0x4d47dda5
Variable types: 0 continuous, 1512006 integer (1512006 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 6.0000000
Presolve removed 0 rows and 0 columns (presolve time = 5s) ...
Presolve time: 6.47s
Presolved: 6298 rows, 1512006 columns, 3030048 nonzeros
Variable types: 0 continuous, 1512006 integer (1512006 binary)

Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.4801587e-01   0.000000e+00   0.000000e+00     10s

Use crossover to convert LP symmetric solution to basic solution...

Root crossover log...

 1510998 PPushes remaining with PInf 0.0000000e+00                10s
 1475939 PPushes remaining with PInf 0.0000000e+00                10s
  800904 PPushes remaining with PInf 0.0000000e+00                15s
  233796 PPushes remaining with PInf 0.0000000e+00                20s
       0 PPushes remaining with PInf 0.0000000e+00                22s

       0 DPushes remaining with DInf 0.0000000e+00                22s

  Push phase complete: Pinf 0.0000000e+00, Dinf 4.3723705e-15     22s


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
 1511001    2.4801587e-01   0.000000e+00   0.000000e+00     22s

Root relaxation: objective 2.480159e-01, 1511001 iterations, 14.57 seconds (11.20 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.24802    0 6296    6.00000    0.24802  95.9%     -  159s
H    0     0                         1.0000000    0.24802  75.2%     -  159s
     0     0    0.24802    0 6296    1.00000    0.24802  75.2%     -  159s

Explored 1 nodes (1513446 simplex iterations) in 159.98 seconds (83.79 work units)
Thread count was 4 (of 4 available processors)

Solution count 2: 1 6

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
Optimal solution found.
Minimum number of workers for 97.75310344827585% completion: 22458
Ausführungszeit: 376.22906931472875 min
Set parameter TimeLimit to value 6000
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2|AVX512]
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Academic license - for non-commercial use only - registered to nils.heilemann@stud-mail.uni-wuerzburg.de
Optimize a model with 6298 rows, 1512006 columns and 3030048 nonzeros
Model fingerprint: 0x4d47dda5
Variable types: 0 continuous, 1512006 integer (1512006 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 6.0000000
Presolve removed 0 rows and 0 columns (presolve time = 5s) ...
Presolve time: 6.49s
Presolved: 6298 rows, 1512006 columns, 3030048 nonzeros
Variable types: 0 continuous, 1512006 integer (1512006 binary)

Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.4801587e-01   0.000000e+00   0.000000e+00     10s

Use crossover to convert LP symmetric solution to basic solution...

Root crossover log...

 1510998 PPushes remaining with PInf 0.0000000e+00                10s
 1475939 PPushes remaining with PInf 0.0000000e+00                10s
  800904 PPushes remaining with PInf 0.0000000e+00                15s
  227748 PPushes remaining with PInf 0.0000000e+00                20s
       0 PPushes remaining with PInf 0.0000000e+00                22s

       0 DPushes remaining with DInf 0.0000000e+00                22s

  Push phase complete: Pinf 0.0000000e+00, Dinf 4.3723705e-15     22s


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
 1511001    2.4801587e-01   0.000000e+00   0.000000e+00     22s

Root relaxation: objective 2.480159e-01, 1511001 iterations, 14.46 seconds (11.20 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.24802    0 6296    6.00000    0.24802  95.9%     -  158s
H    0     0                         1.0000000    0.24802  75.2%     -  159s
     0     0    0.24802    0 6296    1.00000    0.24802  75.2%     -  159s

Explored 1 nodes (1513446 simplex iterations) in 159.41 seconds (83.79 work units)
Thread count was 4 (of 4 available processors)

Solution count 2: 1 6

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
Optimal solution found.
Minimum number of workers for 97.9251724137931% completion: 22573
Ausführungszeit: 378.30289511682133 min
Set parameter TimeLimit to value 6000
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2|AVX512]
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Academic license - for non-commercial use only - registered to nils.heilemann@stud-mail.uni-wuerzburg.de
Optimize a model with 6298 rows, 1512006 columns and 3030048 nonzeros
Model fingerprint: 0x4d47dda5
Variable types: 0 continuous, 1512006 integer (1512006 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 6.0000000
Presolve removed 0 rows and 0 columns (presolve time = 5s) ...
Presolve time: 6.50s
Presolved: 6298 rows, 1512006 columns, 3030048 nonzeros
Variable types: 0 continuous, 1512006 integer (1512006 binary)

Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.4801587e-01   0.000000e+00   0.000000e+00     10s

Use crossover to convert LP symmetric solution to basic solution...

Root crossover log...

 1510998 PPushes remaining with PInf 0.0000000e+00                10s
 1510743 PPushes remaining with PInf 0.0000000e+00                10s
  819814 PPushes remaining with PInf 0.0000000e+00                15s
  251079 PPushes remaining with PInf 0.0000000e+00                20s
       0 PPushes remaining with PInf 0.0000000e+00                22s

       0 DPushes remaining with DInf 0.0000000e+00                22s

  Push phase complete: Pinf 0.0000000e+00, Dinf 4.3723705e-15     22s


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
 1511001    2.4801587e-01   0.000000e+00   0.000000e+00     22s

Root relaxation: objective 2.480159e-01, 1511001 iterations, 14.56 seconds (11.20 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.24802    0 6296    6.00000    0.24802  95.9%     -  160s
H    0     0                         1.0000000    0.24802  75.2%     -  161s
     0     0    0.24802    0 6296    1.00000    0.24802  75.2%     -  161s

Explored 1 nodes (1513446 simplex iterations) in 161.22 seconds (83.79 work units)
Thread count was 4 (of 4 available processors)

Solution count 2: 1 6

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
Optimal solution found.
Minimum number of workers for 98.09724137931035% completion: 22698
Ausführungszeit: 380.54771018860157 min
Set parameter TimeLimit to value 6000
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2|AVX512]
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Academic license - for non-commercial use only - registered to nils.heilemann@stud-mail.uni-wuerzburg.de
Optimize a model with 6298 rows, 1512006 columns and 3030048 nonzeros
Model fingerprint: 0x4d47dda5
Variable types: 0 continuous, 1512006 integer (1512006 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 6.0000000
Presolve removed 0 rows and 0 columns (presolve time = 5s) ...
Presolve time: 6.51s
Presolved: 6298 rows, 1512006 columns, 3030048 nonzeros
Variable types: 0 continuous, 1512006 integer (1512006 binary)

Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.4801587e-01   0.000000e+00   0.000000e+00     10s

Use crossover to convert LP symmetric solution to basic solution...

Root crossover log...

 1510998 PPushes remaining with PInf 0.0000000e+00                10s
 1475939 PPushes remaining with PInf 0.0000000e+00                10s
  800904 PPushes remaining with PInf 0.0000000e+00                15s
  236806 PPushes remaining with PInf 0.0000000e+00                20s
       0 PPushes remaining with PInf 0.0000000e+00                22s

       0 DPushes remaining with DInf 0.0000000e+00                22s

  Push phase complete: Pinf 0.0000000e+00, Dinf 4.3723705e-15     22s


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
 1511001    2.4801587e-01   0.000000e+00   0.000000e+00     22s

Root relaxation: objective 2.480159e-01, 1511001 iterations, 14.55 seconds (11.20 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.24802    0 6296    6.00000    0.24802  95.9%     -  157s
H    0     0                         1.0000000    0.24802  75.2%     -  158s
     0     0    0.24802    0 6296    1.00000    0.24802  75.2%     -  158s

Explored 1 nodes (1513446 simplex iterations) in 158.78 seconds (83.79 work units)
Thread count was 4 (of 4 available processors)

Solution count 2: 1 6

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
Optimal solution found.
Minimum number of workers for 98.26931034482759% completion: 22832
Ausführungszeit: 382.9975798775316 min
Set parameter TimeLimit to value 6000
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2|AVX512]
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Academic license - for non-commercial use only - registered to nils.heilemann@stud-mail.uni-wuerzburg.de
Optimize a model with 6298 rows, 1512006 columns and 3030048 nonzeros
Model fingerprint: 0x4d47dda5
Variable types: 0 continuous, 1512006 integer (1512006 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 6.0000000
Presolve removed 0 rows and 0 columns (presolve time = 5s) ...
Presolve time: 6.92s
Presolved: 6298 rows, 1512006 columns, 3030048 nonzeros
Variable types: 0 continuous, 1512006 integer (1512006 binary)

Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.4801587e-01   0.000000e+00   0.000000e+00     10s

Use crossover to convert LP symmetric solution to basic solution...

Root crossover log...

 1510998 PPushes remaining with PInf 0.0000000e+00                11s
  905765 PPushes remaining with PInf 0.0000000e+00                15s
  350364 PPushes remaining with PInf 0.0000000e+00                20s
       0 PPushes remaining with PInf 0.0000000e+00                23s

       0 DPushes remaining with DInf 0.0000000e+00                23s

  Push phase complete: Pinf 0.0000000e+00, Dinf 4.3723705e-15     23s


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
 1511001    2.4801587e-01   0.000000e+00   0.000000e+00     23s

Root relaxation: objective 2.480159e-01, 1511001 iterations, 15.15 seconds (11.20 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.24802    0 6296    6.00000    0.24802  95.9%     -  166s
H    0     0                         1.0000000    0.24802  75.2%     -  167s
     0     0    0.24802    0 6296    1.00000    0.24802  75.2%     -  167s

Explored 1 nodes (1513446 simplex iterations) in 167.40 seconds (83.79 work units)
Thread count was 4 (of 4 available processors)

Solution count 2: 1 6

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
Optimal solution found.
Minimum number of workers for 98.44137931034483% completion: 22979
Ausführungszeit: 385.69816671662943 min
Set parameter TimeLimit to value 6000
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2|AVX512]
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Academic license - for non-commercial use only - registered to nils.heilemann@stud-mail.uni-wuerzburg.de
Optimize a model with 6298 rows, 1512006 columns and 3030048 nonzeros
Model fingerprint: 0x4d47dda5
Variable types: 0 continuous, 1512006 integer (1512006 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 6.0000000
Presolve removed 0 rows and 0 columns (presolve time = 5s) ...
Presolve time: 6.56s
Presolved: 6298 rows, 1512006 columns, 3030048 nonzeros
Variable types: 0 continuous, 1512006 integer (1512006 binary)

Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.4801587e-01   0.000000e+00   0.000000e+00     10s

Use crossover to convert LP symmetric solution to basic solution...

Root crossover log...

 1510998 PPushes remaining with PInf 0.0000000e+00                10s
 1510151 PPushes remaining with PInf 0.0000000e+00                10s
  819814 PPushes remaining with PInf 0.0000000e+00                15s
  241180 PPushes remaining with PInf 0.0000000e+00                20s
       0 PPushes remaining with PInf 0.0000000e+00                22s

       0 DPushes remaining with DInf 0.0000000e+00                22s

  Push phase complete: Pinf 0.0000000e+00, Dinf 4.3723705e-15     22s


Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
 1511001    2.4801587e-01   0.000000e+00   0.000000e+00     22s

Root relaxation: objective 2.480159e-01, 1511001 iterations, 14.45 seconds (11.20 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.24802    0 6296    6.00000    0.24802  95.9%     -  162s
H    0     0                         1.0000000    0.24802  75.2%     -  162s
     0     0    0.24802    0 6296    1.00000    0.24802  75.2%     -  162s

Explored 1 nodes (1513446 simplex iterations) in 162.94 seconds (83.79 work units)
Thread count was 4 (of 4 available processors)

Solution count 2: 1 6

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
Optimal solution found.
Minimum number of workers for 98.61344827586207% completion: 23141
Ausführungszeit: 388.71275196198536 min
Set parameter TimeLimit to value 6000
```

Following that, we'll construct a plot to visually represent the outcomes of the sensitivity analysis:

In [ ]:

```
import matplotlib.pyplot as plt
from matplotlib.ticker import FixedLocator, FuncFormatter
import locale

# Set to German locale to get comma as thousand separator
locale.setlocale(locale.LC_ALL, 'en_US')

# Unpack the data into x and y lists
x, y = zip(*results)

# Create a new figure with the specified figsize
plt.figure(figsize=(8, 3))

# Create a scatter plot
plt.scatter(x, y)

# Alternatively, create a line plot
plt.plot(x, y)

# Custom y-axis tick labels with "s"
def format_y_ticks(value, tick_number):
    # Use locale to format the number with comma as thousand separator
    formatted = locale.format_string("%d", value, grouping=True)
    return f"{formatted} s"

ax = plt.gca()
ax.yaxis.set_major_formatter(FuncFormatter(format_y_ticks))

# Add percentage unit and detailed ticks to the x-axis
ax.xaxis.set_major_locator(FixedLocator(x))
ax.set_xticklabels(['{:.2f}%'.format(val) for val in x], rotation=45, fontsize=8)

# Add labels to the axes
plt.xlabel('Confidence Level')
plt.ylabel('Execution Time')

# Remove extra white space
plt.tight_layout()

# Save the plot as an SVG file
plt.savefig("images/sensitivity_confidence.svg", format="svg")

# Show the plot
plt.show()
```

5. Graphical User Interface (GUI)

In order to enhance user comprehension of the Smart Stochastic Scheduling approach, we're going to build a GUI for the framework. We'll employ the `ipywidgets` library to craft an interactive user interface, enabling the execution of the `smart_stochastic_scheduling` function in response to a user-defined confidence level.

In [2]:

```
import ipywidgets as widgets
from IPython.display import display, clear_output, Javascript

# Create a label for the headline
headline = widgets.HTML(value='<h2 style="font-weight: bold;">Smart Stochastic Scheduling (SSS)</h2>', layou

# Create a label for the description
description = widgets.Label(value='Please select the desired confidence level and click the button to run th

# Create a slider for the confidence_level
confidence_slider = widgets.FloatSlider(
    value=95.5,
    min=95,
    max=99.99,
    step=0.01,
    description='Confidence Level (%):',
    readout_format='.2f',
    style={'description_width': 'initial'},
)

# Create a text input field for the confidence_level
confidence_text = widgets.FloatText(
    value=95.5,
    step=0.01,
    description='Confidence Level (%):',
    style={'description_width': 'initial'},
)

# Link the slider and text input field
widgets.link((confidence_slider, 'value'), (confidence_text, 'value'))


# Create a button widget
button = widgets.Button(description='Run optimization',button_style='success')

# Create an animated loading widget using HTML and CSS
loading_widget = widgets.HTML(
    value='<div style="font-size: 14px;">Optimizing: <i class="fa fa-spinner fa-spin"></i></div>',
    layout=widgets.Layout(visibility='hidden')  # Hide the loading widget initially
)

# Create a separator
separator = widgets.HTML(value='<hr>')

# Create an Output Widget with a limited height
output = widgets.Output(layout=widgets.Layout(height='200px', overflow='auto'))

def on_button_clicked(b):
    with output:
        # Disable the button while the function is running
        button.disabled = True

        # Display the loading widget
        loading_widget.layout.visibility = 'visible'

        # clear output
        clear_output()

        # run function
        result = smart_stochastic_scheduling(future_jobs, confidence_slider.value / 100)  # Divide by 100 to

        # Enable button and hide the loading animation
        button.disabled = False
        loading_widget.layout.visibility = 'hidden'

        # print result
        print(f'')
        print(f'______________________________')
        print(f'')
        print(f'Total runtime: {int(result)} seconds.')

# Adds the event-handler to the button
button.on_click(on_button_clicked)

# Display the widgets
display(headline, description, confidence_slider, confidence_text, button, loading_widget, separator, output
```

Smart Stochastic Scheduling (SSS)

Please select the desired confidence level and click the button to run the optimization.

Confidence Level (%): 98.50

Confidence Level (%): 98,5

Run optimization

```
Explored 1 nodes (1514144 simplex iterations) in 62.41 seconds (84.42 work units)
Thread count was 10 (of 10 available processors)

Solution count 2: 1 6

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
Optimal solution found.

______________________________

Total runtime: 23084 seconds.
```