Integer Programming Project - Modeling the Single Machine Scheduling Problems by Mixed-Integer Linear Programming

Jiachen Zhang

February 1, 2022

1 Introduction

Scheduling is motivated by problems that arise in industrial planning, in production processes, in telecommunication and generally in all situations where limited resources have to be allocated to activities over time.

A large group of scheduling problems can be properly modeled as mixed-integer linear programs (MILP), quadratic integer programs (QIP) and constraint programs (CP), then be solved by commercial solvers such as CPLEX and Gurobi. Single-machine scheduling problems (SMS) is one of such problems with many variations. Moreover, SMS is more basic than the famous job-shop scheduling problems (JSP) and vehicle routing problems (VRP) in the context of optimization without losing the essence of scheduling problems.

In this project, the author specifically study two types of SMS that have been proved NP-hard [Lenstra et al. (1977), Kise et al. (1978), Lawler (1977)]. The first SMS is with release dates and the (weighted) sum-of-completion-time objective and the second is with due dates and the (weighted) sum-of-tardiness objective. For each of the two SMS, four different MILP formulations including disjunctive MILP formulation, time-index formulation, rank-based formulation, and linear ordering formulation [Ku and Beck (2016), Keha et al. (2009)] are applied to formulate the problems. The models are then implemented in Python and problem instances are solved by Gurobi for results comparison. With the comparison of the performances of the four formulations, the author try to improve them in terms of number of decision variables and constraints. Better LP relaxations will also be taken into consideration during the model refinement process.

2 Problem Definition

A single-machine scheduling problem (SMS) consists of a set of activities $\mathcal{A} = \{a_1, \ldots, a_n\}$ to be scheduled on one machine. Each activity a_i has a positive duration d_i and has to be processed on the machine exactly once. Let d_{max} be the maximum duration of activities. We assume no activities can overlap with each other in their execution on the machine. We also assume non-preemption, i.e., a job that has started processing on a machine cannot be interrupted.

In the first type of SMS, each activity is associated with a release date, before which the activity cannot be processed. The set of release dates is represented by $R = \{R_1, R_2, ..., R_n\}$. Let R_{min} be the earliest release time and R_{max} the latest release time. The objective is to minimize the (weighted) sum of the completion time of each activity, i.e. $\min(\sum_{i=1}^{n} C_i)$, where C_i represents the completion time of activity i. Let T be the upper bound of the completion time of the last activity on the machine. In this report, we assign $T = n * d_{max} + R_{max}$, since the completion time of any activity cannot exceed the summation of the worst-case processing times of all the activities plus the latest release date. The weight for the completion time of each activity i is Wt_i. The first type of SMS is called SMSR in this report.

In the second type of SMS, each activity is associated with a due date. The set of due dates is represented by $D = \{D_1, D_2, ..., D_n\}$. Let D_{min} be the earliest due date and D_{max} the latest due date.

If the activity i is finished after its due date, it would be considered late, which brings in the tardiness, which can be computed by $T_i = C_i - D_i$. The objective is to minimize the (weighted) sum of the tardiness of each activity, i.e. $\min(\sum_{i=1}^{n} T_i)$. The weight for the tardiness of each activity i is Wt_i. The second type of SMS is called SMSD in this report.

In the next section, the four MILP formulations of SMSR are introduced and analyzed.

3 MILP Formulations of SMSR

3.1 The Disjunctive MILP Model

In this section, we present a disjunctive mixed-integer linear programming (MILP) formulation of single-machine scheduling problems with release dates and the (weighted) sum-of-completion time objective. The decision variables of the disjunctive model are:

- x_i - integer start time of activity i.
- $z_{i,j}$ - an indicator of whether activity i precedes activity j.

The disjunctive MILP model is given by:

$$\min_{x,z} \quad \sum_{i=1}^{n} (x_i + d_i) \cdot Wt_i, \tag{1a}$$

$$\text{s.t.} \quad x_i \geq x_j + d_j - M \cdot z_{i,j}, \qquad \forall i, j = 1, ..., n, i < j, \tag{1b}$$

$$x_j \geq x_i + d_i - M \cdot (1 - z_{i,j}), \qquad \forall i, j = 1, ..., n, i < j, \tag{1c}$$

$$x_i \geq R_i, \qquad \forall i = 1, ..., n, \tag{1d}$$

$$x_i \geq 0, \qquad \forall i = 1, ..., n, \tag{1e}$$

$$z_{i,j} \in \{0, 1\}, \qquad \forall i, j = 1, ..., n, i < j. \tag{1f}$$

The objective (1a) represents the sum of weighted completion time of each activity. Constraints (1b) and (1c) ensure that no two activities can be executed on the machine at the same time. M has to be large enough to guarantee the correctness of (1b) and (1c). In our model, we assign $M = T + 1$. We use the same M value in the rest of the report. Constraint (1d) ensures that all activities are scheduled after their release dates. Constraints (1e) make sure that the start time of each activity is greater than 0. Constraint (1f) enforces that two activities on the same machine are in sequential rather than in parallel [Ku and Beck (2016)].

The disjunctive MILP formulation has $n(n-1)/2$ binary decision variables and n integer variables. Besides, this formulation has $n(n-1)$ disjunctive constraints and n constraints for release dates. In total, there are $(n^2+n)/2$ decision variables and n^2 constraints.

3.2 The Time-indexed MILP Model

In this part, we present a time-indexed MILP formulation of single-machine scheduling problems with release dates and the (weighted) sum-of-completion-time objective.
The decision variables of the time-indexed model are:

- $x_{i,t}$ - an indicator of whether activity i starts at time t.

The time-indexed MILP model is given by:

$$\min_{x} \quad \sum_{i=1}^{n} Wt_i \cdot \sum_{t=1}^{T}(t+d_i)\cdot x_{i,t}, \tag{2a}$$

$$\text{s.t.} \quad \sum_{t=1}^{T}(t\cdot x_{i,t}) \geq R_i, \qquad \forall i=1,...,n, \tag{2b}$$

$$\sum_{t=1}^{T} x_{i,t} = 1, \qquad \forall i=1,...,n, \tag{2c}$$

$$\sum_{i=1}^{n}\sum_{t'\in T_{i,t}} x_{i,t'} \leq 1, \qquad \forall t=1,...,T, T_{i,t}=\{t-d_i+1,...,t\} \tag{2d}$$

$$x_{i,t}\in\{0,1\}, \qquad \forall i=1,...,n, \forall t=1,...,T. \tag{2e}$$

The objective (2a) represents the weighted sum of completion time of each activity. Constraint (2b) ensures that all activities are scheduled after their release dates. Constraint (2c) guarantees that every activity starts only once. Constraint (2d) are resource constraints ensuring that no two activities can be executed on the machine at the same time.

The time-indexed MILP formulation has nT binary decision variables. Besides, this formulation has n constraints for release dates and n constraints (2c). Also, there are T resource constraints. In total, there are nT decision variables and $2n+T$ constraints.

3.3 The Linear-ordering MILP Model

In this section, we present a linear-ordering MILP formulation of single-machine scheduling problems with release dates and the (weighted) sum-of-completion-time objective.
The decision variables of the linear-ordering model are:

- x_i - integer idle time before the start of activity i.
- $z_{i,j}$ - linear ordering decision variable indicating that whether activity i precedes activity j.

The disjunctive MILP model is given by:

$$\min_{x,z} \quad \sum_{j=1}^{n}(d_j + x_j + \sum_{i,i\neq j} z_{i,j}\cdot d_i)\cdot Wt_j, \tag{3a}$$

$$\text{s.t.} \quad z_{i,j}+z_{j,i}=1 \qquad \forall i,j=1,...,n, i<j, \tag{3b}$$

$$z_{i,j}+z_{j,k}+z_{k,i}\leq 2 \qquad \forall i,j,k=1,...,n, i\neq j, j\neq k, k\neq i, \tag{3c}$$

$$x_j + \sum_{i,i\neq j} z_{i,j}\cdot d_i \geq R_j, \qquad \forall j=1,...,n, \tag{3d}$$

$$x_j - x_i \geq M\cdot(z_{i,j}-1), \qquad \forall i,j=1,...,n, i\neq j, \tag{3e}$$

$$x_i\geq 0, \qquad \forall i=1,...,n, \tag{3f}$$

$$z_{i,j}\in\{0,1\}, \qquad \forall i,j=1,...,n, i\neq j. \tag{3g}$$

The objective (3a) represents the sum of weighted completion time of each activity. Constraint (3b) ensures that two activities must be in sequential but not in parallel. Constraint (3c) ensures that there is no cycle between three activities. Constraints (3d) are the release date constraints. Constraints (3e) link the integer idle time variables and the linear-ordering variables in a sense that the idle time before activity j is greater than the idle time before activity i if activity i precedes activity j. Constraint (3f) ensures that the idle time of each activity is greater than 0. Constraint (3g) enforces that two activities on the same machine are in sequential rather than in parallel [Keha et al. (2009)].

The linear-ordering MILP formulation has $n(n-1)$ binary decision variables and n integer variables. Besides, this formulation has $n(n-1)/2$ constraints (3b) and $n(n-1)(n-2)/6$ constraints (3c). This formulation also has n release date constraints and $n(n-1)$ constraints. In total, there are n^2 decision variables and $(n^3+6n^2-n)/6$ constraints.

3.4 The Rank-based MILP Model

In this section, we present a rank-based MILP formulation of single-machine scheduling problems with release dates and the sum-of-completion-time objective.
The decision variables of the rank-based model are:

- $x_{i,k}$ - an indicator of whether activity i is scheduled at the k-th position of the machine.
- h_k - the start time of the activity i at the k-th position of the machine sequence.

The rank-based MILP model is given by:

$$
\begin{aligned}
\min_{x,h} \quad & \sum_{k=1}^{n}(h_k + \sum_{i=1}^{n} x_{i,k}\cdot d_i), && && \text{(4a)}\\
\text{s.t.} \quad & \sum_{i=1}^{n} x_{i,k} = 1, && \forall k = 1,...,n, && \text{(4b)}\\
& \sum_{k=1}^{n} x_{i,k} = 1, && \forall i = 1,...,n, && \text{(4c)}\\
& h_k + \sum_{i=1}^{n} x_{i,k}\cdot d_i \le h_{k+1}, && \forall k = 1,...,n-1, && \text{(4d)}\\
& h_k \ge \sum_{i=1}^{n} x_{i,k}\cdot R_i, && \forall k = 1,...,n, && \text{(4e)}\\
& h_k \ge 0, && \forall k = 1,...,n, && \text{(4f)}\\
& x_{i,k} \in \{0,1\}, && \forall i = 1,...,n, \forall k = 1,...,n. && \text{(4g)}
\end{aligned}
$$

The objective (4a) represents the sum of completion time of each activity. Constraint (4b) ensures that only one activity can be placed on a position of the machine. Constraint (4c) ensures that only one position of the machine can be assigned to an activity. Constraint (4d) guarantees that there is no overlap between any two activities on the machine. Constraint (4e) ensures that an activity is only allowed to be scheduled after its release date.

The rank-based MILP formulation has n^2 binary decision variables and n integer variables. Besides, this formulation has n constraints (4b) and n constraints (4c). This formulation also has $n-1$ resource constraints and n release date constraints. In total, there are n^2+n decision variables and $4n-1$ constraints.

Some careful readers may find that there is no weight in the objective (4a) of the rank-based MILP formulation. Since the completion time of an activity is represented according to its position in the schedule, it's very hard to associate a weight to an activity without introducing a quadratic term in the objective. For example, if the weight needs to be incorporated in the objective, the objective function would become

$$\sum_{k=1}^{n}((h_k + \sum_{i=1}^{n} x_{i,k}\cdot d_i)\cdot \sum_{i=1}^{n}(x_{i,k}\cdot Wt_i))$$

which is a quadratic term with respect to decision variables. Thus, weights of activities cannot be easily represented in the objective (4a) of the rank-based MILP formulation.

3.5 Improvement of the Rank-based MILP Model

First of all, the integer variable h_k that representing the starting time of the activity at position k of the sequence seem to need too many binary variables for expansion, since h_k has a large upper bound if k is large. Therefore, we can use new integer variables $w_k, \forall k = 1, ..., n$ representing the idle time between the activities at position $k-1$ and k of the sequence to indirectly replace h_k. The existence of non-zero idle time between two activities is due to the release time of the latter activity is greater than the completion time of the former activity. We can simply use the following equation to replace h_k.

$$h_k = \sum_{j=1}^{k}(\sum_{i=1}^{n} x_{i,j} \cdot d_i + w_j) + w_k \tag{5}$$

The first advantage of w_k is that the idle times should often be small in high-quality solutions. The second advantage of w_k is that we can delete constraints (4d) and express them in the objective function of the MILP model. We will see it later.

Besides, the number of integer variables can be reduced. In a solution, it is often the case that the first few idle times are non-zero but the last few are all zero, since the starting times of the last few activities have exceeded the largest release time. If we can estimate a position m of the sequence such that all activities after that are scheduled compactly (without idle time), we can delete the idle time variables w_k for any $k > m$. Moreover, for any $k > m$, the corresponding constraints (4e) can be deleted since they would be automatically satisfied. We will see it later, too.

In order to find the position m, we sort all the processing time d_i in an non-decreasing order. We use $d_i^{'}$ to represented the sorted processing time, namely $d_1^{'} \leq d_1^{'} \leq ... \leq d_n^{'}$. Thus, the most conservative position index m is the smallest k such that

$$R_{min} + \sum_{i=1}^{k} d_i^{'} \leq R_{max}. \tag{6}$$

The actual position index in the optimal solution could be much smaller but never greater than m.

We provide the improved rank-based MILP model below.

The decision variables of the improved rank-based model are:

- $x_{i,k}$ - an indicator of whether activity i is scheduled at the k-th position of the machine.
- w_k - the idle time between the activities at the $(k-1)$-th and k-th positions of the machine sequence.

The improved rank-based MILP model is given by:

$$\min_{x,w} \quad \sum_{k=1}^{m}(n-k+1) \cdot (w_k + \sum_{i=1}^{n} x_{i,k} \cdot d_i) \quad + \quad \sum_{k=m+1}^{n} (n-k+1) \cdot \sum_{i=1}^{n} x_{i,k} \cdot d_i, \tag{7a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} x_{i,k} = 1, \qquad \forall k = 1, ..., n, \tag{7b}$$

$$\sum_{k=1}^{n} x_{i,k} = 1, \qquad \forall i = 1, ..., n, \tag{7c}$$

$$w_1 = \sum_{i=1}^{n} x_{i,1} \cdot R_i, \tag{7d}$$

$$w_k + \sum_{j=1}^{k-1}(w_j + \sum_{i=1}^{n} x_{i,j} \cdot d_i) \geq \sum_{i=1}^{n} x_{i,k} \cdot R_i, \qquad \forall k = 2, ..., m, \tag{7e}$$

$$w_k \geq 0, \qquad \forall k = 1, ..., m, \tag{7f}$$

$$x_{i,k} \in \{0, 1\}, \qquad \forall i = 1, ..., n, \forall k = 1, ..., n. \tag{7g}$$

The objective (7a) represents the sum of completion time of each activity. Constraint (7b) ensures that only one activity can be placed on a position of the machine. Constraint (7c) ensures that only one position of the machine can be assigned to an activity. Constraint (7d) guarantees that the idle time before the first activity on the machine equals to the activity's release date. Constraint (7e) ensures that an activity at a position before $m + 1$ is only allowed to be scheduled after its release date. The term $\sum_{j=1}^{k-1}(w_j + \sum_{i=1}^{n} x_{i,j} \cdot d_i)$ is the completion time of the $(k-1)$-th activity. In the objective function, the first term $\sum_{k=1}^{m}(n-k+1) \cdot (w_k + \sum_{i=1}^{n} x_{i,k} \cdot d_i)$ is the sum of completion times of the first m activities, and the second term $\sum_{k=m+1}^{n}(n-k+1) \cdot \sum_{i=1}^{n} x_{i,k} \cdot d_i$ is the sum of completion times of the last $n - m$ activities.

The improved rank-based MILP formulation has n^2 binary decision variables and m integer variables. Besides, this formulation has n constraints (7b) and n constraints (7c). This formulation also has m release date constraints. In total, there are $n^2 + m$ decision variables and $2n + m$ constraints.

Ideally, in the improved rank-based MILP model, we need less integer variables and less constraints. However, since the completion time of each activity is still computed according to its position in the schedule, it is still hard to represent the weights in the objective of this improved rank-based MILP formulation without introducing quadratic form.

3.6 Model Comparison

Here we summarize the number of variables and constraints as well as the possibility to incorporate weights in the following table.

Table 1: Properties of the five MILP models of SMSR

Properties	Disjunctive	Time-indexed	Linear-ordering	Rank-based	Improved rank-based
Variables	$(n^2+n)/2$	nT	n^2	n^2+n	n^2+m
Constraints	n^2	$2n+T$	$(n^3+6n^2-n)/6$	$4n-1$	$2n+m$
Weights	Yes	Yes	Yes	No	No

According to the table, we can conclude that rank-based and the improved rank-based MILP models should be most efficient since they have the fewest number of decision variables and constraints. Although the disjunctive and the linear-ordering MILP models have the fewest numbers of variables, they have too many constraints. While model size is an important aspect of the quality of a formulation, it is well-known that other characteristics such as the tightness of the linear relaxation can be equally or more important. Thus, we run experiments to compare performances of the five MILP models in the following section.

4 Experimental Performances of Four MILP Formulations of SMSR

In this section, a number of SMSR problem instances have been tested by using Gurobi for each MILP model. We have selected problem instances with 10, 20, 40, and 80 activities. The duration is always in the range $[1, 100]$; the release dates are always in the range $[1, 10n]$, where n is the number of activities. We treat the problems with weighted objective and unweighted objective separately. All the experiments are run on a computer with Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz and 8GB RAM. The time limit for all experiments is 600 seconds. The results are shown below.

Table 2: Performances of the five MILP models of unweighted SMSR with 10 activities

Properties	Disjunctive	Time-indexed	Linear-ordering	Rank-based	Improved rank-based
Time to Opt	2.40	0.93	0.077	0.016	0.0156
Opt Gap	0	0	0	0	0

Table 3: Performances of the five MILP models of unweighted SMSR with 20 activities

Properties	Disjunctive	Time-indexed	Linear-ordering	Rank-based	Improved rank-based
Time to Opt	600+	11.561	57.103	0.155	0.176
Opt Gap	11.68%	0	0	0	0

Table 4: Performances of the five MILP models of unweighted SMSR with 40 activities

Properties	Disjunctive	Time-indexed	Linear-ordering	Rank-based	Improved rank-based
Time to Opt	600+	217.757	476.382	4.217	18.481
Opt Gap	27.726%	0	1.348%	0	0

Table 5: Performances of the five MILP models of unweighted SMSR with 80 activities

Properties	Disjunctive	Time-indexed	Linear-ordering	Rank-based	Improved rank-based
Time to Opt	600+	440.118	600+	82.565	390.336
Opt Gap	39.726%	2.078%	5.469%	0	0

From the result tables we can find that the rank-based MILP models have a significant advantage over other MILP models for unweighted objective in terms of solution time. The reason could be that the rank-base MILP models have the fewest number of constraints, namely the polyhedron of the LP relaxation is not very complicated. Within rank-based models, the improved model has a slightly worse performance than the original rank-based model, which surprises the author. The reason could lie in the complexity of constraints (7e), which incorporates too many variables. It seems that the fewer number of variables and constraints do not necessarily matter if the structure of constraints has changed. Since it is hard to represent weights in the objective of rank-based MILP models, the good performance might serve as the other side of the trade-off.

The time-index MILP model has a satisfactory performance, though not the best. In this model, the numbers of the variables and constraints are apparently the highest among the four MILP models, so the polyhedron of the time-indexed model is of the highest dimension. However, since the variables are all binary, each variable only has two options for branching, which reduces the complexity of branch-and-bound procedure on this MILP model. That's why the performance of time-indexed MILP model is better than both the disjunctive MILP model and the linear-ordering MILP model.

The disjunctive MILP model has the worst performance in finding the optimal solution and proving optimality. Though the number of constraints is much more than the disjunctive MILP model, the linear-ordering MILP model has a better performance than the disjunctive MILP model. Given that the two models all have integer ordering variables and big-M constraints linking binary and integer variables, the experimental results show that the linear-ordering model provides a better way to take advantage of big-M constraints. Since the big-M constraints are all considered as disjunctive constraints in both models, the author suspects that the constraints (3b) and (3c) are the keys to accelerate the solving process. Since the two types of constraints are redundant for the disjunctive MILP model, the author has created a new disjunctive MILP model with constraints (3b) and (3c) as redundant constraints. However, the results show that there is no gain but more pain after adding these constraints, which indicates that our guess is not correct. Some of the results are shown below.

Table 6: Performances of the original and the new disjunctive MILP models of unweighted SMSR with 20 activities

Properties	Original disjunctive	Disjunctive with (3b) and (3c)
Time to Opt	$600+$	$600+$
Opt Gap	11.771%	13.125%

Thus, the better performance of the linear-ordering might be due to its superior structure of integer variables. The range of the integer variable representing idle time before each activity starts is much less than the range of the integer variable representing start time of each activity. The linear-ordering binary variables are all the same in the two models, the better integer variables might be the key to a better MILP model.

The results of the problems with weighted objective are shown below.

Table 7: Performances of the three MILP models of weighted SMSR with 10 activities

Properties	Disjunctive	Time-indexed	Linear-ordering
Time to Opt	0.692	1.219	0.0823
Opt Gap	0	0	0

Table 8: Performances of the three MILP models of weighted SMSR with 20 activities

Properties	Disjunctive	Time-indexed	Linear-ordering
Time to Opt	$600+$	13.124	481.804
Opt Gap	8.124%	0	1.56%

Table 9: Performances of the three MILP models of weighted SMSR with 40 activities

Properties	Disjunctive	Time-indexed	Linear-ordering
Time to Opt	$600+$	349.643	$600+$
Opt Gap	16.606%	0	1.665%

Table 10: Performances of the three MILP models of weighted SMSR with 80 activities

Properties	Disjunctive	Time-indexed	Linear-ordering
Time to Opt	$600+$	$600+$	$600+$
Opt Gap	24.613%	1.531%	5.462%

For the problems with weighted objective, the rank-based model cannot be applied. The performances of the disjunctive MILP model and the linear-ordering MILP model are better than the time-indexed model when the problem sizes are small and are much worse when the problem sizes are large. This result is kind of surprising since the time-indexed MILP model is well-known for its poor scalability. Our guess is that the problem sizes of weighted SMSR are still not large enough to breakdown the time-indexed MILP model. We hence have tried extremely large problem instances with 160 activities. The results are as follows.

Table 11: Performances of the three MILP models of weighted SMSR with 160 activities

Properties	Disjunctive	Time-indexed	Linear-ordering
Time to Opt	600+	600+	600+
Opt Gap	36.406%	4.437%	5.462%

However, the results remain the same, which implies that the time-indexed MILP model is still better than the disjunctive and linear-ordering MILP models when the problem sizes get very large.

5 MILP Formulations of SMSD

5.1 The Disjunctive MILP Model

In this section, we present a disjunctive mixed-integer linear programming (MILP) formulation of single-machine scheduling problems with due dates and the (weighted) sum-of-tardiness objective.
The decision variables of the disjunctive model are:

- x_i - integer start time of activity i.
- $z_{i,j}$ - an indicator of whether activity i precedes activity j.
- T_i - integer tardiness of activity i.

The disjunctive MILP model is given by:

$$\min_{x,z,T} \quad \sum_{i=1}^{n} T_i \cdot Wt_i, \tag{8a}$$

$$\begin{aligned}
\text{s.t.} \quad & x_i \geq x_j + d_j - M \cdot z_{i,j}, && \forall i,j = 1,...,n, i < j, && \text{(8b)}\\
& x_j \geq x_i + d_i - M \cdot (1 - z_{i,j}), && \forall i,j = 1,...,n, i < j, && \text{(8c)}\\
& T_i \geq x_i + d_i - D_i, && \forall i = 1,...,n, && \text{(8d)}\\
& x_i \geq 0, && \forall i = 1,...,n, && \text{(8e)}\\
& T_i \geq 0, && \forall i = 1,...,n, && \text{(8f)}\\
& z_{i,j} \in \{0,1\}, && \forall i,j = 1,...,n, i < j. && \text{(8g)}
\end{aligned}$$

The objective (8a) represents the sum of weighted tardiness of each activity. Constraints (8b) and (8c) ensure that no two activities can be executed on the machine at the same time. Constraint (8d) ensures that the tardiness of activity i is always greater than the difference between the completion time and the due date of activity i. Since the objective is to minimize the tardiness, the inequality in (8d) would become equality due to the optimization. Constraints (8e) and (8f) make sure that the start time and the tardiness of each activity is greater than 0. Constraint (8g) enforces that two activities on the same machine are in sequential rather than in parallel [Ku and Beck (2016)].

The disjunctive MILP formulation has $n(n-1)/2$ binary decision variables and $2n$ integer variables. Besides, this formulation has $n(n-1)$ disjunctive constraints and n constraints for tardiness. In total, there are $(n^2 + 3n)/2$ decision variables and n^2 constraints.

5.2 The Time-indexed MILP Model

In this section, we present a time-indexed mixed-integer linear programming (MILP) formulation of single-machine scheduling problems with due dates and the (weighted) sum-of-tardiness objective. The decision variables of the time-indexed model are:

- $x_{i,t}$ - an indicator of whether activity i starts at time t.
- T_i - integer tardiness of activity i.

The time-indexed MILP model is given by:

$$\min_{x,T} \quad \sum_{i=1}^{n} Wt_i \cdot T_i, \tag{9a}$$

$$\text{s.t.} \quad T_i \geq \sum_{t=1}^{T} t \cdot x_{i,t} + d_i - D_i, \qquad \forall i = 1, ..., n, \tag{9b}$$

$$\sum_{t=1}^{T} x_{i,t} = 1, \qquad \forall i = 1, ..., n, \tag{9c}$$

$$\sum_{i=1}^{n} \sum_{t' \in T_{i,t}} x_{i,t'} \leq 1, \qquad \forall t = 1, ..., T, T_{i,t} = \{t - d_i + 1, ..., t\} \tag{9d}$$

$$T_i \geq 0, \qquad \forall i = 1, ..., n, \tag{9e}$$

$$x_{i,t} \in \{0, 1\}, \qquad \forall i = 1, ..., n, \forall t = 1, ..., T. \tag{9f}$$

The objective (9a) represents the weighted sum of tardiness of each activity. Constraint (9b) ensures that the tardiness of activity i is always greater than the difference between the completion time and the due date of activity i. Constraint (9c) guarantees that every activity starts only once. Constraint (9d) are resource constraints ensuring that no two activities can be executed on the machine at the same time.

The time-indexed MILP formulation has nT binary decision variables and n integer variables. Besides, this formulation has n constraints for release dates and n constraints (2c). Also, there are T resource constraints. In total, there are $nT + n$ decision variables and $2n + T$ constraints.

5.3 The Linear-ordering MILP Model

In this section, we present a time-indexed mixed-integer linear programming (MILP) formulation of single-machine scheduling problems with due dates and the (weighted) sum-of-tardiness objective. The decision variables of the linear-ordering model are:

- x_i - integer idle time before the start of activity i.
- $z_{i,j}$ - linear ordering decision variable indicating that whether activity i precedes activity j.
- T_i - integer tardiness of activity i.

The disjunctive MILP model is given by:

$$
\begin{aligned}
\min_{x,z,T} \quad & \sum_{j=1}^{n} T_j \cdot Wt_j, && && \text{(10a)}\\
\text{s.t.} \quad & z_{i,j} + z_{j,i} = 1 && \forall i,j = 1,...,n, i < j, && \text{(10b)}\\
& z_{i,j} + z_{j,k} + z_{k,i} \leq 2 && \forall i,j,k = 1,...,n, i \neq j, j \neq k, k \neq i, && \text{(10c)}\\
& T_j \geq d_j + x_j + \sum_{i, i \neq j} z_{i,j} \cdot d_i - D_j, && \forall j = 1,...,n, && \text{(10d)}\\
& x_j - x_i \geq M \cdot (z_{i,j} - 1), && \forall i,j = 1,...,n, i \neq j, && \text{(10e)}\\
& x_i \geq 0, && \forall i = 1,...,n, && \text{(10f)}\\
& T_i \geq 0, && \forall i = 1,...,n, && \text{(10g)}\\
& z_{i,j} \in \{0,1\}, && \forall i,j = 1,...,n, i \neq j. && \text{(10h)}
\end{aligned}
$$

The objective (10a) represents the sum of weighted tardiness of each activity. Constraint (10b) ensures that two activities must be in sequential but not in parallel. Constraint (10c) ensures that there is no cycle between three activities. Constraints (10d) are the tardiness constraints. Constraints (10e) link the integer idle time variables and the linear-ordering variables in a sense that the idle time before activity j is greater than the idle time before activity i if activity i precedes activity j [Keha et al. (2009)].

The linear-ordering MILP formulation has $n(n-1)$ binary decision variables and $2n$ integer variables. Besides, this formulation has $n(n-1)/2$ constraints (3b) and $n(n-1)(n-2)/6$ constraints (3c). This formulation also has n release date constraints and $n(n-1)$ constraints. In total, there are n^2+n decision variables and $(n^3+6n^2-n)/6$ constraints.

5.4 The Rank-based MILP Model

In this section, we present a rank-based MILP formulation of single-machine scheduling problems with due dates and the sum-of-tardiness objective.
The decision variables of the rank-based model are:

- $x_{i,k}$ - an indicator of whether activity i is scheduled at the k-th position of the machine.
- h_k - the start time of the activity i at the k-th position of the machine sequence.
- T_k - integer tardiness of the activity at the position k-th position of the machine sequence.

The rank-based MILP model is given by:

$$\min_{x,h} \quad \sum_{k=1}^{n} T_k, \tag{11a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} x_{i,k} = 1, \qquad \forall k = 1, ..., n, \tag{11b}$$

$$\sum_{k=1}^{n} x_{i,k} = 1, \qquad \forall i = 1, ..., n, \tag{11c}$$

$$h_k + \sum_{i=1}^{n} x_{i,k} \cdot d_i \leq h_{k+1}, \qquad \forall k = 1, ..., n-1, \tag{11d}$$

$$T_k \geq h_k + \sum_{i=1}^{n} x_{i,k} \cdot (d_i - D_i), \qquad \forall k = 1, ..., n, \tag{11e}$$

$$h_k \geq 0, \qquad \forall k = 1, ..., n, \tag{11f}$$

$$T_k \geq 0, \qquad \forall k = 1, ..., n, \tag{11g}$$

$$x_{i,k} \in \{0, 1\}, \qquad \forall i = 1, ..., n, \forall k = 1, ..., n. \tag{11h}$$

The objective (11a) represents the sum of tardiness of each activity. Constraint (11b) ensures that only one activity can be placed on a position of the machine. Constraint (11c) ensures that only one position of the machine can be assigned to an activity. Constraint (11d) guarantees that there is no overlap between any two activities on the machine. Constraints (11e) are the tardiness constraints.

The rank-based MILP formulation has n^2 binary decision variables and $2n$ integer variables. Besides, this formulation has n constraints (4b) and n constraints (4c). This formulation also has $n-1$ resource constraints and n release date constraints. In total, there are $n^2 + 2n$ decision variables and $4n - 1$ constraints.

Similar to the rank-based MILP formulation in Section 4, there is no weight in the objective (11a) of the rank-based MILP formulation. Since the tardiness of an activity is represented according to its position in the schedule, it's very hard to associate a weight to an activity without introducing a quadratic term in the objective. For example, if the weight needs to be incorporated in the objective, the objective function would become

$$\sum_{k=1}^{n} (T_k \cdot \sum_{i=1}^{n} (x_{i,k} \cdot Wt_i))$$

which is a quadratic term with respect to decision variables. Thus, weights of activities cannot be easily represented in the objective (11a) of the rank-based MILP formulation.

5.5 Improvement of the Rank-based MILP Model

Similar to Section 3.5, the integer variable h_k can be simply replaced by the following equation.

$$h_k = \sum_{j=1}^{k} (\sum_{i=1}^{n} x_{i,j} \cdot d_i + w_j) + w_k \tag{12}$$

The first advantage of w_k is that the idle times should often be small in high-quality solutions. Thus the overall number of binary variables for the binary expansion of w_k is less than the number for the binary expansion of h_k. The second advantage of w_k is that we can delete constraints (11d) and express them in the objective function of the MILP model. We will see it later.

Since the objective in this section is no longer the sum-of-completion-time, there might be an idle time for each activity as the objective is to minimize the sum-of-tardiness. Thus, we won't look for the position m as in Section 3. We provide the improved rank-based MILP model below.
The decision variables of the improved rank-based model are:

- $x_{i,k}$ - an indicator of whether activity i is scheduled at the k-th position of the machine.
- w_k - the idle time between the activities at the $(k-1)$-th and k-th positions of the machine sequence.
- T_k - integer tardiness of the activity at the position k-th position of the machine sequence.

The improved rank-based MILP model is given by:

$$
\begin{aligned}
\min_{x,w} \quad & \sum_{k=1}^{n} T_k, && && (13a)\\
\text{s.t.} \quad & \sum_{i=1}^{n} x_{i,k} = 1, && \forall k = 1, ..., n, && (13b)\\
& \sum_{k=1}^{n} x_{i,k} = 1, && \forall i = 1, ..., n, && (13c)\\
& T_1 \geq w_1 + \sum_{i=1}^{n} x_{i,1} \cdot (d_i - D_i), && && (13d)\\
& T_k \geq w_k + \sum_{j=1}^{k-1} (w_j + \sum_{i=1}^{n} x_{i,j} \cdot d_i) + \sum_{i=1}^{n} x_{i,k} \cdot (d_i - D_i), && \forall k = 2, ..., n, && (13e)\\
& w_k \geq 0, && \forall k = 1, ..., n, && (13f)\\
& T_k \geq 0, && \forall k = 1, ..., n, && (13g)\\
& x_{i,k} \in \{0, 1\}, && \forall i = 1, ..., n, \forall k = 1, ..., n. && (13h)
\end{aligned}
$$

The objective (13a) represents the sum of tardiness of each activity. Constraint (13b) ensures that only one activity can be placed on a position of the machine. Constraint (13c) ensures that only one position of the machine can be assigned to an activity. Constraint (13d) is the tardiness constraint for the activity at the first position of the schedule. Constraints (13e) are the tardiness constraints for the activity at the other positions of the schedule.

The improved rank-based MILP formulation has n^2 binary decision variables and $2n$ integer variables. Besides, this formulation has n constraints (7b) and n constraints (7c). This formulation also has n tardiness constraints. In total, there are $n^2 + 2n$ decision variables and $3n$ constraints.

Ideally, in the improved rank-based MILP model, we need less integer variables and less constraints. However, since the completion time of each activity is still computed according to its position in the schedule, it is still hard to represent the weights in the objective of this improved rank-based MILP formulation without introducing quadratic form.

5.6 Model Comparison

Here we summarize the number of variables and constraints as well as the possibility to incorporate weights in the following table.

Table 12: Properties of the four MILP models of SMSD

Properties	Disjunctive	Time-indexed	Linear-ordering	Rank-based	Improved rank-based
Variables	$(n^2 + 3n)/2$	$nT + n$	$n^2 + n$	$n^2 + 2n$	$n^2 + 2n$
Constraints	n^2	$2n + T$	$(n^3 + 6n^2 - n)/6$	$4n - 1$	$3n$
Weights	Yes	Yes	Yes	No	No

According to the table, we can conclude that rank-based and the improved rank-based MILP models should be most efficient since they have the fewest number of decision variables and constraints. Although the disjunctive and the linear-ordering MILP models have the fewest numbers of variables, they have too many constraints. While model size is an important aspect of the quality of a formulation, it is well-known that other characteristics such as the tightness of the linear relaxation can be equally or more important. Thus, we run experiments to compare performances of the five MILP models in the following section.

6 Experimental Performances of Five MILP Formulations of SMSD

In this section, a number of SMSD problem instances have been tested by using Gurobi for each MILP model. We have selected problem instances with 10, 20, 40, and 80 activities. The duration is always in the range $(1, 100)$; the release dates are always in the range $(1, 10n)$, where n is the number of activities. We treat the problems with weighted objective and unweighted objective separately. All the experiments are run on a computer with Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz and 8GB RAM. The time limit for all experiments is 600 seconds. The results are shown below.

Table 13: Gurobi Performances on SMSD with 10 activities and unweighted objective

Properties	Disjunctive	Time-indexed	Linear-ordering	Rank-based	Improved rank-based
Variables	0.451	3.265	0.266	0.125	0.102
Constraints	0	0	0	0	0

Table 14: Gurobi Performances on SMSD with 20 activities and unweighted objective

Properties	Disjunctive	Time-indexed	Linear-ordering	Rank-based	Improved rank-based
Variables	600+	82.692	34.802	6.47	7.826
Constraints	81.25%	0	0	0	0

Table 15: Gurobi Performances on SMSD with 40 activities and unweighted objective

Properties	Disjunctive	Time-indexed	Linear-ordering	Rank-based	Improved rank-based
Variables	600+	600+	600+	600+	600+
Constraints	94.804%	5.763%	1.933%	5.91%	6.548%

Table 16: Gurobi Performances on SMSD with 80 activities and unweighted objective

Properties	Disjunctive	Time-indexed	Linear-ordering	Rank-based	Improved rank-based
Variables	600+	600+	600+	600+	600+
Constraints	99.585%	99.98%	3.138%	7.582%	8.237%

From the result tables we can find that the overall performances of the five models for unweighted SMSD are worse than the performances for unweighted SMSR, which suggests that the unweighted SMSD is harder than unweighted SMSD for MILP.

Among the five models, the rank-based MILP models have a significant advantage over other MILP models for unweighted objective in terms of solution time when problem sizes are relatively small.

The reason could be that the rank-base MILP models have the fewest number of constraints, namely the polyhedron of the LP relaxation is not very complicated. Within rank-based models, the improved model has a slightly worse performance than the original rank-based model. The reason could lie in the complexity of constraints (13e), which incorporates too many variables. It seems that the fewer number of variables and constraints do not necessarily matter if the structure of constraints has changed. Since it is hard to represent weights in the objective of rank-based MILP models, the good performance might serve as the other side of the trade-off.

The time-index MILP model has a satisfactory performance, though not the best. In this model, the numbers of the variables and constraints are apparently the highest among the four MILP models, so the polyhedron of the time-indexed model is of the highest dimension. However, since the variables are all binary, each variable only has two options for branching, which reduces the complexity of branch-and-bound procedure on this MILP model.

The disjunctive MILP model has the worst performance in finding the optimal solution and proving optimality. Though the number of constraints is much more than the disjunctive MILP model, the linear-ordering MILP model has a better performance than the disjunctive MILP model. When the problem sizes are getting larger, the performance of linear-ordering MILP model is becoming the best among the five models, which surprises the author.

Given that the two models all have integer ordering variables and big-M constraints linking binary and integer variables, the experimental results show that the linear-ordering model provides a better way to take advantage of big-M constraints. The better performance of the linear-ordering might also be due to its superior integer variables. The range of the integer variable representing idle time before each activity starts is much less than the range of the integer variable representing start time of each activity. The linear-ordering binary variable are all the same in the two models, the better integer variables might be the key to a better MILP model.

The results of the problems with weighted objective are shown below.

Table 17: Gurobi Performances on SMSD with 10 activities and weighted objective

Properties	Disjunctive	Time-indexed	Linear-ordering
Variables	1.201	2.94	0.107
Constraints	0	0	0

Table 18: Gurobi Performances on SMSD with 20 activities and weighted objective

Properties	Disjunctive	Time-indexed	Linear-ordering
Variables	600+	144.54	3.124
Constraints	58.32%	0	0

Table 19: Gurobi Performances on SMSD with 40 activities and weighted objective

Properties	Disjunctive	Time-indexed	Linear-ordering
Variables	600+	600+	600+
Constraints	94.804%	5.763%	1.933%

Table 20: Gurobi Performances on SMSD with 80 activities and weighted objective

Properties	Disjunctive	Time-indexed	Linear-ordering
Variables	600+	176.624	600+
Constraints	99.211%	16.725%	11.512%

The overall performances of the three MILP models for the weighted SMSD are much worse than the performances for the weighted SMSR, which suggests that the weighted SMSD is harder than weighted SMSD for MILP.

For the problems with weighted objective, the rank-based model cannot be applied. The performances of the disjunctive MILP model and the linear-ordering MILP model are better than the time-indexed model when the problem sizes are small. However, when the problem sizes are getting larger, the linear-ordering MILP model has the best performance though the disjunctive MILP model has the worst performance. The results show that the linear-ordering MILP model differs significantly from the disjunctive MILP model in terms of computational behaviour, though the two models look similar. We have run experiments on problem instances with 160 activities to see the further performance. Some of the results are shown below.

Table 21: Gurobi Performances on SMSD with 160 activities and weighted objective

Properties	Disjunctive	Time-indexed	Linear-ordering
Variables	600+	600+	600+
Constraints	100%	96.845%	100%

According to the above table, there is no change of the relative performances between the three MILP models for problem instances with larger sizes.

7 Conclusion

In this project, the author has developed disjunctive, time-indexed, linear-ordering, rank-based and improved rank-based mixed integer linear programming models for unweighted single-machine scheduling problems with release dates and sum-of-completion-time objective as well as unweighted single-machine scheduling problems with due dates and sum-of-tardiness objective. Due to the incompatibility of rank-based MILP models and weights on objective functions, the author only develops disjunctive, time-indexed and linear-ordering MILP models for weighted single-machine scheduling problems with release dates and sum-of-completion-time objective as well as unweighted single-machine scheduling problems with due dates and sum-of-tardiness objective.

The author has compared the performances of the above MILP models for a number of single-machine scheduling problem instances by using the Gurobi MILP solver. It turns out that the original rank-based MILP model has the best performances for both types of unweighted single-machine scheduling problems since the model has the fewest number of decision variables and constraints without introducing too much complexity to constraints. For the weighted single-machine scheduling problems, the time-indexed MILP model surprisingly outperforms both the disjunctive and the linear-ordering MILP models when problem sizes are large, since most variables in a time-indexed model are binary. Besides, the linear-ordering MILP model always have a better performance than the disjunctive MILP model due to the superior structure of integer variables in linear-ordering MILP models.

References

Keha, A. B., Khowala, K., and Fowler, J. W. (2009). Mixed integer programming formulations for single machine scheduling problems. *Computers & Industrial Engineering*, 56(1):357–367. 1, 3, 11

Kise, H., Ibaraki, T., and Mine, H. (1978). A solvable case of the one-machine scheduling problem with ready and due times. *Operations Research*, 26(1):121–126. 1

Ku, W.-Y. and Beck, J. C. (2016). Mixed integer programming models for job shop scheduling: A computational analysis. *Computers & Operations Research*, 73:165–173. 1, 2, 9

Lawler, E. L. (1977). A "pseudopolynomial" algorithm for sequencing jobs to minimize total tardiness. In *Annals of discrete Mathematics*, volume 1, pages 331–342. Elsevier. 1

Lenstra, J. K., Kan, A. R., and Brucker, P. (1977). Complexity of machine scheduling problems. *Studies in integer programming*, 1:343–362. 1