Scenario Based Robust Line Balancing: Computational Complexity

Alexandre Dolgui\textsuperscript{a,∗}, Sergey Kovalev\textsuperscript{a}

\textsuperscript{a}Industrial Engineering and Computer Science Center (G2I), Ecole des Mines de Saint-Etienne, 158, cours Fauriel, 42023 Saint-Etienne Cedex 2, France

Abstract

This paper studies the following line balancing problem with uncertain operation execution times. Operations on the same product have to be assigned to the stations of a transfer line. The product moves along the stations in the same direction, and operations assigned to the same station are executed sequentially. Exclusion, inclusion and precedence relations are given on the set of operations. Operation execution times are uncertain in the sense that their set belongs to a given set of scenarios. The objective is to minimize the line cycle time, which is equal to the maximum total execution time of operations of the same station, for the worst scenario. An approach to reducing the scenario set is described. Several special cases of the problem are proved NP-hard and strongly NP-hard. Enumerative dynamic programming algorithms and problem-specific polynomial time algorithms are suggested for some cases.

Keywords: Line balancing, Uncertainty, Scenario, Min-max approach, Computational complexity.

1. Introduction

1.1. Deterministic formulation

The following problem is studied. A set \( N = \{1, \ldots, n\} \) of operations on the same product has to be executed repeatedly on a transfer line consisting of \( m, m \geq 2 \), stations. Operations assigned to a station are executed sequentially, and operations assigned to station \( l \) are executed earlier than those assigned to station \( l + 1, l = 1, \ldots, m - 1 \). Plural exclusion and inclusion relations as well as binary precedence relations can be imposed on the set \( N \).

Exclusion relations are represented by a collection \( E \) of subsets \( E' \subset N \) such that \( |E'| \geq 2 \) and all operations of \( E', E' \in E \), cannot be assigned to the same station but any proper

\*Corresponding author

Email addresses: dolgui@emse.fr (Alexandre Dolgui), kovalev@emse.fr (Sergey Kovalev)
subset of $E'$ can be assigned to the same station. These constraints are needed for resolving situations in which some resources cannot be simultaneously used on the same station due to a conflict of their characteristics, for example, dimensions. Without incurring any loss of generality, we assume that there are no two sets in $E$ containing one another.

Inclusion relations are represented by a collection $I$ of subsets $I' \subset N$ such that all operations of $I', I' \in I$, must be assigned to the same station. These relations represent situations in which the required precision of some operations can be lost if the product is moved between these operations. We assume without loss of generality that all the sets in $I$ are non-intersecting and that every operation belongs to exactly one such set. If inclusion relations are not specified, then, by the definition, $I$ consists of the sets $\{j\}$, $j = 1, \ldots, n$.

If operation $i$ precedes operation $j$, then $i$ should be executed earlier than $j$, on a station with a smaller index or before $j$ on the same station. Precedence relations reflect technological requirements for operations. They can be represented by a digraph $G = (N, U)$, in which there is an arc $(i, j) \in U$ if and only if $i$ precedes $j$.

The problem is to assign operations to the stations and to determine the order of their execution on the stations such that the exclusion, inclusion and precedence relations are satisfied and line cycle time is minimized. Line cycle time is equal to the maximum station load, and a station load is the total execution time of operations assigned to the station.

1.2. Execution time uncertainty

The majority of the literature on line balancing problems assumes that the numerical input parameters are given and precise, see the latest classification of line balancing problems in Boysen et al. [9] and Scholl et al. [36]. However, this assumption is often unrealistic because numerical parameters admit variations or they are subject to unpredictable changes, especially when human factor or tool deterioration is involved.

The principal reason for uncertainty in line balancing is imprecise or uncertain execution times of operations, which is the aspect studied in this paper. To describe uncertainty, we assume that a collection of real operation execution times is an element of a given set $S$ of scenarios. Let $p_{lj}$ denote the real execution time of operation $j$ on station $l$, and let $p$ denote the $m \times n$ matrix with entries $p_{lj}$. We assume that $p \in S$. Since the set of operation execution times is not unique, a question appears as to which solution is appropriate. The best solution for a specific set of operation execution times can be the worst for another set. One of the classic approaches to hedge against uncertainty is the absolute robustness (or min-max)
approach, which is to construct the best solution for the worst scenario, see Kouvelis and Yu [24] for definitions, Aissi, Bazgan and Vanderpooten [3] for a survey of the applications to discrete optimization problems, and the most recent publications of Minoux [27], Averbakh [4], Aissi et al. [2] and Berman and Wang [7]. In this paper, we will apply the min-max approach to the line balancing problem.

We assume that the set of scenarios $S$ is a finite collection of subsets of interval scenarios: $S = S^{(1)} \cup \ldots \cup S^{(k)}$, and each scenario $p \in S^{(i)}$ is specified by lower and upper bounds on operation execution times: $S^{(i)} = \{ p \mid a_{ij}^{(i)} \leq p_{lj} \leq b_{ij}^{(i)}, l = 1, \ldots, m, j = 1, \ldots, n \}$, where all $a_{ij}^{(i)}$ and $b_{ij}^{(i)}$, $0 \leq a_{ij}^{(i)} \leq b_{ij}^{(i)}$, are given numbers.

This assumption is relevant in situations where variations of execution times are due to the human factor or tool deterioration. In both cases, precise execution times or their probabilistic characteristics are difficult to predict but possible to evaluate based on the shift schedule or tool maintenance (replacement) timetable. Dynamics of the shifts and tool maintenance lead to the variability of the scenario subsets such that a specific subset $S^{(i)}$ corresponds to a specific shift or time period in which no tool maintenance takes place.

In reality variations of execution times in the same subset $S^{(i)}$ (the same shift) can be small and they can be large between the subsets (different shifts). It may not be optimal to account for variations between the shifts by voluntary widening station borders – the employed resources can be idle for a long time in shifts with small operation execution times.

It should be mentioned that there exist various approaches to represent uncertainty and to define robust decisions, see an extended discussion of these issues in Roy [33]. Our choice of the scenario based approach results from the assumption that uncertainty applies only to the operation execution times, whose probabilistic characteristics are unknown but real values can be adequately numerically evaluated.

The choice of the min-max objective function is motivated by the production environment, in which a transfer line supplies succeeding manufacturing process on a regular basis such that the time between two successive deliveries, which is the line cycle time, is supposed to be the same. Therefore, the objective is to balance the line so that whichever operation execution times are, the planned cycle time is guaranteed not to be exceeded. The minimization part of the min-max objective removes unnecessary line idle times and allows smooth loading of the stations, in contrast to the models in which it is replaced by an upper bounding constraint, which permits a disbalance between the stations.

The min-max balancing decision is supposed to be used over several shifts. Its efficiency
can be verified for scenarios different from the worst scenario. If the worst-case decision is not satisfactory in some sense, for example, if line idle times are too big, the scenario set can be adjusted and the balance can be re-calculated.

1.3. Problem simplification and formal statement

Observe that a feasible solution to the considered problem exists if precedence relations are non-contradictory such that graph $G$ is acyclic. This condition can be verified in $O(n + |U|)$ time by the well-known topological ordering procedure of graph theory, see for example, Kahn [23]. Assume that precedence relations are non-contradictory, i.e., graph $G = (N, U)$ is acyclic.

If there exist arcs $(i, j) \in U$ and $(k, l) \in U$ such that operations $i$ and $l$ are from the same inclusion set $I_1$ and operations $j$ and $k$ are from the same inclusion set $I_2$, then all operations of the two sets $I_1$ and $I_2$ must be assigned to the same station in any feasible solution. In this case, sets $I_1$ and $I_2$ can be united. Unification of all such sets can be done in $O(n + |U|)$ time by using the following two-stage procedure. Firstly, construct a precedence graph $G_I = (I, U_I)$ with vertices being inclusion sets and arcs $(I_1, I_2) \in U_I$ such that there exist operations $i \in I_1$, $j \in I_2$ and arc $(i, j) \in U$ in the original precedence graph. Secondly, apply an algorithm of finding strongly connected components in graph $G_I$, for example, Tarjan’s [38] algorithm. Vertices of each strongly connected component can be united to form a single inclusion set. Now assume that the unification procedure has been applied, which implies that the graph $G_I$ is acyclic.

Further observe that inclusion and exclusion relations must be consistent in any feasible solution: no set $I' \in I$ can include all operations of a set $E' \in E$. This condition can be verified in $O(\sum_{E' \in E} |E'|) \leq O(n|E|)$ time by scanning operations of the same exclusion set and checking if all these operations belong to the same inclusion set. Assume that inclusion and exclusion relations are consistent.

It should also be noted that the order of operations assigned to a station does not affect the line cycle time as well as the inclusion, exclusion and precedence relations with respect to the other operations. Therefore, once the sets of operations on the stations are determined, it suffices to provide topological ordering for each such set with respect to the precedence relations of this set, which takes $O(n + |U|)$ time, as we already mentioned.

From now on, we relax the problem formulation by omitting the requirement of the ordering decision for operations of the same station and assume that operations assigned to any station
are always feasibly ordered.

Application of all the above simplifications makes the original problem less constrained. Furthermore, subsets \( I' \subseteq I \) can be treated as composite operations, whose execution times are sums of execution times of the corresponding original operations, and the problem can be reformulated such that operations are replaced by the composite operations. In this case, precedence relations can be given on the set \( I \) rather than on the set \( N \), and they can be represented by the acyclic digraph \( G_I = (I, U_I) \). However, we prefer not to reformulate the problem for the practical reader convenience and for the purposes of computational complexity classification because there are special cases whose description requires numerical characteristics of single operations and not composite operations, for example, those with non-empty precedence constraints inside the inclusion sets or with inclusion sets which intersect with several exclusion sets.

Let \( x \) denote a 0-1 assignment matrix with \( m \) rows, \( n \) columns and entries \( x_{lj} \) such that \( x_{lj} = 1 \) if operation \( j \) is assigned to station \( l \), and \( x_{lj} = 0 \), otherwise. Let \( X \) be the set of all assignment matrices \( x \). Since each operation must be executed exactly once, we require

\[
\sum_{l=1}^{m} x_{lj} = 1, \quad j = 1, \ldots, n,
\]

for each assignment matrix \( x \in X \). Given \( p \in S \) and \( x \in X \), the corresponding line cycle time is equal to

\[
F(p, x) := \max_{1 \leq l \leq m} \left\{ \sum_{j=1}^{n} p_{lj} x_{lj} \right\}.
\]

Following the min-max approach, the problem is to determine an optimal assignment matrix \( x^* \in X \subseteq \bar{X} \) such that

\[
x^* = \arg \min_{x \in X} \max_{p \in S} F(p, x) = \arg \min_{x \in X} \max_{p \in S} \max_{1 \leq l \leq m} \left\{ \sum_{j=1}^{n} p_{lj} x_{lj} \right\}.
\]

Here \( X \) is a set of feasible assignment matrices, for which the above specified exclusion, inclusion and precedence relations on the set \( N \) of operations are satisfied. This problem will be denoted as \( \text{SCEN-L} \), where the abbreviation \( L \) stands for Line.

We now show how the original problem with an infinite number of scenarios can be reduced to an equivalent problem with a finite number of maximal (worst) scenarios.

Matrix \( \hat{p} \in S \) is called maximal if for any \( p \in S \) relations \( \hat{p}_{lj} \leq p_{lj}, \ l = 1, \ldots, m, \ j = 1, \ldots, n \), imply \( p = \hat{p} \). Let \( \hat{S} \) denote the set of all maximal matrices in \( S \). It should be noted that for any \( p \in S \) there exists \( \hat{p} \in \hat{S} \) such that \( \hat{p}_{lj} \geq p_{lj}, \ l = 1, \ldots, m, \ j = 1, \ldots, n \), and that all maximal matrices are mutually incomparable such that, for any \( \hat{p} \in \hat{S} \) and \( \hat{p}' \in \hat{S} \), there exist indices \( a, b \) and \( c, d \) such that \( \hat{p}_{ab} > \hat{p}'_{ab} \) and \( \hat{p}_{cd} < \hat{p}'_{cd} \).

Set \( \hat{S} \) can be calculated as follows. Consider the original set \( S = S^{(1)} \cup \cdots \cup S^{(k)} \). Observe that each subset \( S^{(i)} \) contains a unique maximal matrix \( p^{(i)} \) such that \( p^{(i)}_{lj} = b^{(i)}_{lj}, \ l = 1, \ldots, m, \ j = 1, \ldots, n \),
Thus, $\hat{S} \subseteq W := \{p^{(1)}, \ldots, p^{(k)}\}$. Furthermore, it is easy to see that set $\hat{S}$ can be constructed in $O(mnk^2)$ time by comparing each matrix of the set $W$ against every other matrix in this set.

By the definition of the set $\hat{S}$, we have

$$\max_{p \in S} \max_{1 \leq l \leq m} \sum_{j=1}^{n} p_{lj} x_{lj} = \max_{\hat{p} \in \hat{S}} \max_{1 \leq l \leq m} \sum_{j=1}^{n} \hat{p}_{lj} x_{lj}.$$

Thus, the mathematical model for the original problem Scen-L and the problem with the set $\hat{S}$ of maximal matrices is the same. For the conformity of presentation, let $S = \hat{S} = \{p^{(1)}, \ldots, p^{(k)}\}$. Given assignment matrix $x$, the worst value of the objective function can be calculated as follows.

$$\max_{p \in S} \max_{1 \leq l \leq m} \sum_{j=1}^{n} p_{lj} x_{lj} = \max_{1 \leq i \leq k} \max_{1 \leq l \leq m} \sum_{j=1}^{n} p_{lj} x_{lj} = \max \left\{ \sum_{j=1}^{n} p_{lj} x_{lj}, \ldots, \sum_{j=1}^{n} p_{lj} x_{lj} \right\} = \max_{1 \leq l \leq m} \{T_l(p^{(1)}, \ldots, p^{(k)}, x)\},$$

where

$$T_l(p^{(1)}, \ldots, p^{(k)}, x) := \max \sum_{j=1}^{n} p_{lj} x_{lj}$$

can be treated as a $k$-dimensional load of station $l$ with respect to the operation execution time matrices (scenarios) $p^{(1)}, \ldots, p^{(k)}$.

1.4. Related models and literature

Problem Scen-L can be viewed as a generalization of the single-model Simple Assembly Line Balancing Problem with the objective of minimizing the cycle time, denoted as SALB-2 in Boysen et al. [9], see also Battini et al. [6]. Besides the scenario uncertainty and the min-max objective, the generalizations include station dependent execution times, inclusion and exclusion relations. Scholl [35] and Boysen et al. [9] provided a classification of assembly line balancing problems. Among the most important characteristics they note the type of the workpiece movement, by which paced/unpaced and synchronous/asynchronous lines are distinguished. Any of these types fits into the formulation of problem Scen-L.

To the best of our knowledge, the min-max approach to line balancing problems was applied only by Xu and Xiao [41, 42] and Mirzapour Al-E-Hashem et al. [29], who studied real-life mixed-model problems, in which operation execution time can take any value from a given interval independently of the other operations. The suggested min-max model is resolved by
a genetic algorithm in [41, 42] and an optimal mathematical programming algorithm in [29].
Computational complexity aspects are not studied in these papers. This is the main focus of
our article.

Line balancing problems in which operation execution times vary depending on the assign-
ment of workers are studied by Hopp et al. [22], Miralles et al. [28] and Corominas et al. [13].
The variability of the operation execution times is modeled deterministically in these studies:
one once a worker is assigned to an operation, the operation execution time is fully specified.

For the case in which precedence relations are in effect only for operations of the same
station problem Scen-L can be viewed as a generalization of the problem of scheduling $n$ jobs
on $m$ unrelated parallel machines to minimize the makespan, which is denoted as $R | \beta | C_{\text{max}}$
in the scheduling literature, or the problem of packing $n$ items into $m$ bins to minimize the load
of the most loaded bin. The field $\beta$ specifies job characteristics, which are inclusion, exclusion
and machine dependent precedence relations in the considered case. We are unaware of the
min-max models for the above mentioned scheduling and bin packing problems.

The rest of this paper is organized as follows. In Section 2, we prove that problem Scen-
L is NP-hard in the strong sense even if no inclusion, exclusion or precedence relation is
given and 1) $k = 1$, $m$ is variable and operation execution times are station independent
($p_{lj} = p_j$), or 2) $m = 2$ and $k$ is variable. If $k$ and $m$ are constants and there is no inclusion,
exclusion or precedence relation, then the problem is NP-hard in the ordinary sense and
pseudo-polynomially solvable by a dynamic programming algorithm. Thus, in the general case
and for where operation execution times depend solely on the operation index, computational
complexity of the problem Scen-L is settled. In Section 3, computational complexity and
algorithmic results are presented for when operation execution times depend solely on the
station index, which completes computational complexity classification of all special cases of
the problem Scen-L with regard to the operation execution times and inclusion, exclusion
and precedence relations, except for the case with a constant number of stations, non-empty
inclusion relations and graph $G_I$ being not a collection of chains. The paper concludes with a
summary of the results and suggestions for future research.

2. Complexity of the general problem

In this section we show that problem Scen-L is NP-hard in the strong sense in the following
cases: 1) $k = 1$, $m$ is variable and $p_{lj} = p_j$ for all $l$ and $j$, and 2) $m = 2$ and $k$ is variable. In
both cases, no inclusion, exclusion or precedence relation is assumed to be given on the set $N$
of operations, i.e., \( X = X \).

**Theorem 1.** If \( k = 1 \), \( X = X \) and \( p_{lj} = p_j \), \( l = 1, \ldots, m \), \( j = 1, \ldots, n \), for the unique vector \( p = (p_1, \ldots, p_n) \in S \), then problem \( \text{Scen-L} \) is NP-hard in the strong sense for variable \( m \) and it is NP-hard in the ordinary sense for a constant \( m \).

**Proof:** Under the conditions of the theorem, problem \( \text{Scen-L} \) is to minimize 
\[
\max_{1 \leq l \leq m} \{ \sum_{j=1}^{n} p_{lj} x_{lj} \}
\]
on the set of 0-1 matrices \( X \), for a given vector \( p = (p_1, \ldots, p_n) \). This problem coincides with the well known scheduling problem \( P||C_{\max} \), in which \( n \) non-preemptive jobs with processing times \( p_j \) have to be scheduled on \( m \) identical parallel machines to minimize the completion time of the latest job. Problem \( P||C_{\max} \) is NP-hard in the strong sense if \( m \) is variable and it is NP-hard in the ordinary sense if \( m \) is a constant, see problem \( \text{Multiprocessor Scheduling} \) in Garey and Johnson [17].

**Theorem 2.** Problem \( \text{Scen-L} \) is NP-hard in the strong sense if \( m = 2 \), \( X = X \) and \( p_{lj}^{(i)} \in \{0, p_{lj}^{(i)}\} \), \( l = 1, 2 \), \( i = 1, \ldots, k \).

**Proof:** If \( m = 2 \), then there is one to one correspondence between set \( X \) and the set of all 0-1 vectors \( y = (y_1, \ldots, y_n) \): \( y_j = 1 \) if operation \( j \) is assigned to station 1, and \( y_j = 0 \) if it is assigned to station 2. Then, under the conditions of the theorem, problem \( \text{Scen-L} \) is to minimize 
\[
f(y) := \max_{1 \leq l \leq k} \{ \sum_{j=1}^{n} p_{lj}^{(i)} y_j, \sum_{j=1}^{n} p_{lj}^{(i)} (1 - y_j) \}
\]
subject to \( y_j \in \{0, 1\} \), \( j = 1, \ldots, n \). We show that a decision version of problem \( \text{Scen-L} \) is NP-hard by a reduction from the NP-hard in the strong sense problem \( \text{Exact Cover by 3-Sets (X3C)} \), see Garey and Johnson [17].

**X3C:** Given a family \( A = \{A_1, \ldots, A_a\} \) of 3-element subsets of the set \( B = \{1, \ldots, 3b\} \), does \( A \) contain an exact cover of \( B \), i.e., a subfamily \( Y \subseteq A \) such that each \( i \in B \) belongs to exactly one 3-element set in \( Y \)? We assume \( a > b \) because otherwise there exists a trivial solution.

Given an instance of X3C, we construct the following instance of the decision version of problem \( \text{Scen-L} \). Define the number of operations \( n = a \) and the number of scenarios \( k = 3b + 1 \). Set \( D = a(a - b) \). Define \( p_{2j}^{(i)} = 1 \), \( j = 1, \ldots, n \), \( i = 1, \ldots, 3b \). Calculate 
\[
p_{lj}^{(i)} = D \cdot h_{lj}^{(i)}
\]
where \( h_{lj}^{(i)} = 1 \) if element \( i \in B \) is present in the subset \( A_j \), and \( h_{lj}^{(i)} = 0 \), otherwise, \( j = 1, \ldots, n \). Define \( p_{lj}^{(3b+1)} = 1 \) and \( p_{2j}^{(3b+1)} = a \), \( j = 1, \ldots, n \).

It can be shown that for the considered instance, a solution to X3C exists if and only if there exists a solution to \( \text{Scen-L} \) with value \( f(y) \leq D \). It is easy to see that our reduction is polynomial and pseudo-polynomial at the same time.
Part “if”. Assume that there exists a solution $y$ to Scen-L with value $f(y) \leq D$. This relation implies (i) $D \sum_{j=1}^{n} h_j^{(i)} y_j \leq D$, $i = 1, \ldots, 3b$, (ii) $\sum_{j=1}^{n} (1 - y_j) \leq D$, $i = 1, \ldots, 3b$, (iii) $\sum_{j=1}^{n} y_j \leq D$, and (iv) $a \sum_{j=1}^{n} (1 - y_j) \leq D$. Observe that relations in (ii) and (iii) are always satisfied because their left hand sides do not exceed $n = a \leq D = a(a - b)$. Relations in (iv) and (i) are equivalent, respectively, to

$$\sum_{j=1}^{n} y_j \geq b,$$

$$\sum_{j=1}^{n} h_j^{(i)} y_j \leq 1, \quad i = 1, \ldots, 3b.$$

Construct a family $Y$ of 3-element subsets $A_j \in A$ such that $A_j \in Y$ if and only if $y_j = 1$. Relation (1) ensures that $|Y| \geq b$. Relations in (2) ensure that each $i \in B$ belongs to at most one 3-element set $A_j$ in $Y$, which implies $|Y| \leq b$. Otherwise, the number of elements in the sets of $Y$ is at least $3(b + 1)$. Since there are $3b$ distinct elements among them, at least one of them must appear twice in the sets of $Y$, i.e., one of the relations (2) must be violated. We deduce that $|Y| = b$. Hence, each $i \in B$ should belong to exactly one 3-element set in $Y$. The last statement proves that $Y$ is a solution to X3C. Part “if” is proven.

Part “only if”. If there exists a solution $Y$ to X3C, then we define $y_j = 1$ if and only if $A_j \in Y$. For $y = (y_1, \ldots, y_n)$, relations in (1) and (2), and, hence, (iv) and (i), are satisfied as equalities. Since relations in (ii) and (iii) are obviously satisfied for $y$, it is a solution to Scen-L with value $f(y) = D$, as required.

If there are a constant number of scenarios and a constant number of stations, i.e., if $k$ and $m$ are constants, then problem Scen-L with no inclusion, exclusion or precedence relation, which we denote as Scen-L-Simple, can be solved in pseudo-polynomial time by a straightforward dynamic programming algorithm. For completeness, we present such an algorithm, which we denote as DP-Simple. Introduce $(k \cdot m)$-dimensional vector of state variables (state): $T = (T_1^{(1)}, \ldots, T_1^{(k)}, \ldots, T_m^{(1)}, \ldots, T_m^{(k)})$, where $T_i^{(i)}$ is the load of station $l$ with respect to scenario $i$. Introduce sets $Z_1, \ldots, Z_n$ of feasible states such that $Z_j$ consists of states $T$ each of which corresponds to a feasible assignment of operations $1, \ldots, j$ to the $m$ stations.

Algorithm DP-Simple (for problem Scen-L-Simple)

Step 1 (Initialization) Set $Z_0 = \{(0, \ldots, 0)\}$.

Step 2 (Recursion) Calculate $Z_j = \{T + \Delta_{lj} \mid T \in Z_{j-1}, \ l = 1, \ldots, m\}$, where $\Delta_{lj}$ is the $(k \cdot m)$-dimensional vector, in which $k$ coordinates $k(l-1)+1, k(l-1)+2, \ldots, kl$ associated
with station \( l \) are equal to \( p_{ij}^{(1)}, p_{ij}^{(2)}, \ldots, p_{ij}^{(k)} \), respectively, and the remaining coordinates are equal to zero. We assume that \( Z_j \) is not a multi-set, i.e., it contains only one copy of each appropriate vector \( T \). Vector summation \( T + \Delta_{ij} \) corresponds to the assignment of operation \( j \) to station \( l \) according to which the load of this station with respect to scenario \( i \) is increased by \( p_{ij}^{(i)} \), \( i = 1, \ldots, k \).

If \( j = n \), then go to Step 3. Otherwise, re-set \( j := j + 1 \) and repeat Step 2.

**Step 3** (Optimal solution) Calculate optimal solution value

\[
\min \{\max \{T_1^{(1)}, \ldots, T_1^{(k)}, \ldots, T_m^{(1)}, \ldots, T_m^{(k)}\} \mid (T_1^{(1)}, \ldots, T_1^{(k)}, \ldots, T_m^{(1)}, \ldots, T_m^{(k)}) \in Z_n\}
\]

and determine corresponding optimal assignment of operations by backtracking.

Denote \( P_{ij}^{(i)} = \sum_{r=1}^{j} p_{ir}^{(i)} \), \( i = 1, \ldots, k \), \( l = 1, \ldots, m \), \( j = 1, \ldots, n \), and \( P = \max \{P_{1j}^{(i)}\} \) where this maximum is taken over all \( i \) and \( l \). Since \( T_{ij}^{(i)} \leq P_{ij}^{(i)} \) in iteration \( j \) of Step 2, the time complexity of the algorithm DP-Simple can be evaluated as \( O(km \sum_{j=1}^{n} \prod_{i=1}^{k} \prod_{l=1}^{m} P_{ij}^{(i)}) \leq O(mnkP^{km}) \). Taking into account the time needed to construct the set of maximal scenarios, problem SCEN-L-SIMPLE can be solved in \( O(mnk^2 + mnkP^{km}) \) which is pseudo-polynomial if \( k \) and \( m \) are constants.

**3. Operation independent execution times, \( p_{ij} = p_l \)**

In this section, we study a special case of problem SCEN-L, in which \( p_{ij} = p_l \), \( j = 1, \ldots, n \), for all \( p \in S \), and \( S^{(i)} = \{p \mid a^{(i)}_l \leq p_l \leq b^{(i)}_l, l = 1, \ldots, m\} \). We denote this special case as SCEN-L(\( p_{ij} = p_l \)). It can be used as a model in situations when duration of any operation assigned to the same station is the same or nearly so. For example, durations of drilling, threading and screwing operations associated with a hole of the same depth and diameter are usually the same and depend solely on the speed of the revolving head.

The study of the problem SCEN-L(\( p_{ij} = p_l \)) is also important because its optimal solution value is obviously a lower bound for the original problem if inequalities \( b^{(i)}_l \leq \max_{1 \leq i \leq l} b^{(i)}_l \) are satisfied for all \( i, l \) and \( j \). The closer are the values \( b^{(i)}_l \) and \( b^{(i)}_l \) the better is the lower bound.

For a given 0-1 matrix \( x \in \overline{X} \), introduce aggregate variables \( x_l := \sum_{j=1}^{n} x_{ij}, l = 1, \ldots, m \), which satisfy \( x_l \in \{0, 1, \ldots, n\} \), \( l = 1, \ldots, m \), and \( \sum_{l=1}^{m} x_l = n \). Denote \( p^*_l := \max_{p \in S} \{p_l\} = \max_{1 \leq i \leq l} b^{(i)}_l \), \( l = 1, \ldots, m \). Since \( |S| = k \), values \( p^*_l \) can be computed in \( O(mk) \) time. Simplify

\[
\max_{p \in S} F(p, x) = \max_{p \in S} \{p_1 x_1, \ldots, p_m x_m\} = \max\{x_1 \max_{p \in S} \{p_1\}, \ldots, x_m \max_{p \in S} \{p_m\}\} = \max_{1 \leq i \leq m} \{p^*_i x_i\}.
\]
It follows that problem SCEN-L($p_{ij} = p_l$) with $k$ scenarios boils down to the same problem with one scenario, in which $p_{ij} = p_l^*, l = 1, \ldots, m$, $j = 1, \ldots, n$.

3.1. No inclusion, exclusion or precedence relation

Denote special case in this subsection as SCEN-L($p_{ij} = p_l$)-SIMPLE. Consider a continuous relaxation of this problem:

$$\min F(x) = \max_{1 \leq l \leq m} \{p_l^* x_l\}, \quad \text{ (3)}$$
$$\sum_{l=1}^{m} x_l = n, \quad \text{ (4)}$$
$$x_l \geq 0, \quad l = 1, \ldots, m. \quad \text{ (5)}$$

Denote an optimal solution of problem (3)-(5) and its optimal value by $x^0$ and $F^0$, respectively. It is easy to show that equalities $p_l^* x_l^0 = F^0, l = 1, \ldots, m$, must be satisfied. Then, $x_l^0 = F^0/p_l^*, l = 1, \ldots, m$, and from $\sum_{l=1}^{m} x_l^0 = n$ we obtain

$$F^0 = \frac{n \prod_{l=1}^{m} p_l^*}{\sum_{r=1}^{m}(\prod_{l=1}^{m} p_l^*)/p_r^*} \quad \text{and} \quad x_l^0 = \frac{n \prod_{l=1}^{m} p_l^*}{p_l^* \sum_{r=1}^{m}(\prod_{l=1}^{m} p_l^*)/p_r^*}, \quad l = 1, \ldots, m.$$  

Values $x_l^0, l = 1, \ldots, m$, can be computed in $O(m)$ time.

Observe that function $F(x)$ is convex because it is a maximum of convex (linear) functions, and feasible domain (4)-(5) is convex as well. Therefore, an optimal solution $x^*$ for the more constrained integer problem SCEN-L($p_{ij} = p_l$)-SIMPLE must be an integer point in the closest feasible neighborhood of $x^*$, i.e., it must satisfy $x_l^* \in \{\lfloor x_l^0 \rfloor, \lceil x_l^0 \rceil\}, l = 1, \ldots, m$, and $\sum_{l=1}^{m} x_l^* = n$. It can be shown that $x_l^* = \lfloor x_l^0 \rfloor$ should correspond to larger $p_l^*$ and $x_l^* = \lceil x_l^0 \rceil$ should correspond to smaller $p_l^*$. Re-number values $p_l^*$ so that $p_1^* \geq \cdots \geq p_m^*$, which takes $O(m \log m)$ time. For an optimal solution $x^*$, there exists a unique index $u$, $0 \leq u \leq m$, such that

$$x_l^* = \lfloor x_l^0 \rfloor, \quad l = 1, \ldots, u, \quad \text{and} \quad x_l^* = \lceil x_l^0 \rceil, \quad l = u + 1, \ldots, m.$$  

Index $u$ should satisfy $A_{1,u} + B_{u+1,m} = n$, where $A_{1,u} = \sum_{l=1}^{u} \lfloor x_l^0 \rfloor$ and $B_{u+1,m} = \sum_{l=u+1}^{m} \lceil x_l^0 \rceil$. All values $A_{1,l}$ and $B_{l+1,m}, l = 0, 1, \ldots, m$, can be calculated in $O(m)$ time. Equation $A_{1,l} + B_{l+1,m} = n$ for $0 \leq l \leq m$ can be solved in $O(m)$ time as well. We deduce that problem SCEN-L($p_{ij} = p_l$)-SIMPLE can be solved in $O(mk + m \log m)$ time, where $O(mk)$ is needed to compute scenario $p^*$. 
3.2. Given precedence and no inclusion or exclusion relation

Denote special case in this subsection as SCEN-L($p_{lj} = p_l$)-Prec. Recall that the precedence relations are represented by an acyclic digraph $G = (N, U)$. Given an arbitrary instance of SCEN-L($p_{lj} = p_l$)-Prec, consider the corresponding instance of the problem SCEN-L($p_{lj} = p_l$)-Simple with no precedence relations. Let $x^*$ be an optimal solution for the latter instance.

In $O(n + |U|)$ time, determine an arbitrary topologically feasible sequence of operations with respect to graph $G$. Without loss of generality, let it be $(1, \ldots, n)$. Determine a solution to the problem SCEN-L($p_{lj} = p_l$)-Prec as follows. Assign operations $1, 2, \ldots, x^*_1$ to the first station, operations $x^*_1 + 1, \ldots, x^*_1 + x^*_2$ to the second station, and so on. This solution is feasible for SCEN-L($p_{lj} = p_l$)-Prec and has the same objective value as the corresponding instance of the less constrained problem SCEN-L($p_{lj} = p_l$)-Simple. Therefore, it is optimal for the problem SCEN-L($p_{lj} = p_l$)-Prec. Its construction requires $O(n + |U| + mk + m \log m)$ time, which is polynomial.

3.3. Given inclusion and no exclusion or precedence relation

Denote special case in this subsection as SCEN-L($p_{lj} = p_l$)-Incl. Let $I_1, \ldots, I_v$, $v \leq n$, be all subsets of the set $I$, thus, $v = |I|$. Introduce variables $y_{lr}$, $l = 1, \ldots, m$, $r = 1, \ldots, v$, such that $y_{lr} = 1$ if all operations of the set $I_r$ are assigned to station $l$, and $y_{lr} = 0$, otherwise. Denote $|I_r| = g_r$, $r = 1, \ldots, v$. We have $\sum_{r=1}^{v} g_r = n$. Problem SCEN-L($p_{lj} = p_l$)-Incl can be formulated as follows.

\[
\min \ F(y) = \max_{1 \leq l \leq m} \left\{ p_l^* \sum_{r=1}^{v} g_r y_{lr} \right\}, \quad (6)
\]

\[
\sum_{l=1}^{m} y_{lr} = 1, \quad r = 1, \ldots, v, \quad (7)
\]

\[
y_{lr} \in \{0, 1\}, \quad l = 1, \ldots, m, \quad r = 1, \ldots, v. \quad (8)
\]

Problem (6)-(8) can be interpreted as the well known scheduling problem $Q||C_{max}$, in which there are $v$ non-preemptive jobs to be scheduled for processing on $m$ uniform parallel machines such that the processing time of job $r$ on machine $l$ is equal to $p_l^* g_r$, and the objective is to minimize the completion time of the latest job. Problem $Q||C_{max}$ is NP-hard in the strong sense if $m$ is variable and $p_l^* = 1$, $l = 1, \ldots, m$, and it is NP-hard in the ordinary sense if $m$ is a constant greater than or equal to two and $p_l^* = 1$, $l = 1, \ldots, m$, see Lawler et al. [26]. It can be solved in $O(vmn^{m-1})$ time by a modification of algorithm DP-Simple, in which there
are states \((T_1, \ldots, T_{m-1}) \in \mathbb{Z}_j\) such that \(T_l\) is the summation of \(g_r\) for all operations among \(1, \ldots, j\) assigned to station \(l\).

We can deduce that problem \(\text{SCEN-L}(p_{lj} = p_l)\)-INCL is NP-hard in the strong sense if \(m\) is variable and \(p_l = 1, l = 1, \ldots, m\). If \(m\) is a constant, its computational complexity depends on the presentation of its input. If sets \(I' \in I\) are given as arbitrary collections of operations, then the length of the input is \(O(n)\), and problem \(\text{SCEN-L}(p_{lj} = p_l)\)-INCL is polynomially solvable for any constant \(m\). If sets \(I' \in I\) are given by their cardinalities \(g_r, r = 1, \ldots, v\), i.e., if the so-called \emph{id-encoding} (Rinnooy Kan [32]) or, in other terminology, \emph{high-multiplicity encoding} (Hochbaum and Shamir [21]) is used, then the length of the input is \(O(v)\) and the problem \(\text{SCEN-L}(p_{lj} = p_l)\)-Incl is NP-hard in the ordinary sense even if \(p_l = 1, l = 1, \ldots, m\), and it is pseudo-polynomially solvable for any constant \(m\).

### 3.4. Given inclusion and precedence relations and no exclusion relation

Denote special case in this subsection as \(\text{SCEN-L}(p_{lj} = p_l)\)-INCL-PREC. Again, let \(I_1, \ldots, I_v, v \leq n\), be all subsets of the set \(I\). Consider the digraph \(G_I = (I, U_I)\), see Section 1.3. Due to the strong NP-hardness of the problem \(\text{SCEN-L}(p_{lj} = p_l)\)-INCL for variable \(m\), only the case with a constant \(m\) is of interest with respect to computational complexity. In this paper, we were able to establish polynomiality of a sub-case of this case, in which graph \(G_I\) is a collection of chains. Computational complexity of the remaining sub-cases rests unknown.

We now present a dynamic programming algorithm, denoted as DP-Chains, for problem \(\text{SCEN-L}(p_{lj} = p_l)\)-INCL-CHAINS. Let \(C_1, \ldots, C_c\) be all chains. Consider an arbitrary chain in \(G_I\). Without loss of generality, let it be \(C_i = (I_1, I_2, \ldots, I_r)\) such that \(I_j\) precedes \(I_{j+1}\), \(j = 1, \ldots, r - 1\). Observe that for any feasible solution, there exists a partition of \(C_i\) into at most \(m\) sub-chains such that \(C_i = (C_{1i}, C_{2i}, \ldots, C_{hi})\), \(h_i \leq m\), and a sub-chain \(C_{li}\) with a smaller index \(l\) contains consecutively indexed sets \(I_j\) with smaller indices and it is assigned to a station with a smaller index. Sub-chains of different chains assigned to the same station can be sequenced arbitrarily. Denote the number of operations (of all sets \(I_j\)) in the chain \(C_i\) as \(d_i\), and the number of operations in the sub-chain \(C_{li}\) as \(d_{li}\). Any assignment of these sub-chains to the stations is completely characterized by an \((m - 1)\)-dimensional assignment vector \(a_i := (a_{1i}, a_{2i}, \ldots, a_{m-1,i})\), where \(a_{li} = d_{li}\) if the sub-chain \(C_{li}\) is assigned to the station \(l\), and \(a_{li} = 0\), otherwise, \(l = 1, \ldots, m - 1\). The number of operations of chain \(C_i\) assigned to station \(m\) is equal to \(d_i - \sum_{l=1}^{m-1} a_{li}\). Denote by \(A_i\) the set of all feasible assignment vectors.
for chain $C_i$. Since $a_{li} \leq d_{li}$, $l = 1, \ldots, m - 1$, and $\sum_{l=1}^{m-1} d_{li} \leq d_i$, we have $|A_i| \leq O(d_i^{m-1})$. All sets $A_i$, $i = 1, \ldots, c$, can be calculated in $O(n^{m-1})$ time.

Introduce $(m-1)$-dimensional state $V = (V_1, \ldots, V_{m-1})$, where $V_l$ is the number of operations assigned to station $l$. Introduce sets $Y_1, \ldots, Y_c$ of feasible states such that $Y_i$ consists of states $U$ each of which corresponds to a feasible assignment of the sub-chains of the chains $C_1, \ldots, C_i$ to the stations $1, \ldots, m - 1$.

**Algorithm DP-Chains** (for problem SCEN-L($p_{lj} = p_l$)-INCL-CHAINS)

**Step 1** (Initialization) Set $Y_0 = \{(0, \ldots, 0)\}$.

**Step 2** (Recursion) Calculate $Y_i = \{V + a_i \mid V \in Z_{i-1}, a_i \in A_i\}$. As in the algorithm DP-Simple, we assume that $Y_i$ is not a multi-set, i.e., it contains only one copy of each appropriate vector $V$. Vector summation $V + a_i$ corresponds to the assignment of sub-chains of the chain $C_i$, which are determined by the assignment vector $a_i$, to the stations, according to which the number of operations on station $l$ is increased by $a_{li}$, $l = 1, \ldots, m - 1$.

If $i = c$, then go to Step 3. Otherwise, re-set $i := i + 1$ and repeat Step 2.

**Step 3** (Optimal solution) Calculate optimal solution value

$$\min \{\max\{p^*_1V_1, \ldots, p^*_mV_{m-1}, p^*_m(n - \sum_{l=1}^{m-1} V_l)\} \mid (V_1, \ldots, V_{m-1}) \in Y_c\}$$

and determine corresponding optimal assignment of sub-chains to the stations by backtracking.

Since $V_l \leq n$, $l = 1, \ldots, m - 1$, in iteration $i$ of Step 2, the time complexity of the algorithm DP-Chains can be evaluated as $O(n \sum_{i=1}^{c} d_i^{m-1}) \leq O(n^m)$. Thus, problem SCEN-L($p_{lj} = p_l$)-INCL-CHAINS can be solved in $O(mk + |U| + n^m)$ time, which is polynomial if $m$ is a constant. Here $O(mk)$ time is needed to calculate values $p^*_l$ and $O(n + |U|)$ time is needed to find feasible sequences of operations inside the inclusion sets.

Note that, unlike problem SCEN-L($p_{lj} = p_l$)-INCL and scheduling problem $Q||C_{max}$, problem SCEN-L($p_{lj} = p_l$)-PREC or problem SCEN-L($p_{lj} = p_l$)-INCL-PREC and scheduling problem $Q|p_{lj} = p_l, prec|C_{max}$ with precedence constraints on the the set of jobs are different, because of the different effects of precedence constraints. Computational complexity of problem $Qm|p_{lj} = p_l, chains|C_{max}$ with a constant number of machines is open (Brucker and
Knust [11]) and problem $Q[p_{ij} = p_l, \text{chains}]C_{\text{max}}$ with variable number of machines is NP-hard in the strong sense (Kubiak [25]), while we have shown that problem $\text{SCEN-L}(p_{ij} = p_l)\text{-INCL-CHAINS}$ with a constant number of stations and problem $\text{SCEN-L}(p_{ij} = p_l)\text{-PREC}$ with variable number of stations are polynomially solvable.

3.5. Given exclusion and no inclusion or precedence relation

Denote the special case in this subsection as $\text{SCEN-L}(p_{ij} = p_l)\text{-Excl}$.

**Theorem 3.** Problem $\text{SCEN-L}(p_{ij} = p_l)\text{-Excl}$ is NP-hard in the strong sense if $m$ is variable and $|E'| = 2$ for each $E' \in E$.

**Proof:** We use a reduction from the problem Exact Cover by 3-Sets (X3C), see Theorem 2. Given an instance of X3C, we construct the following instance of the decision version of problem $\text{SCEN-L}(p_{ij} = p_l)\text{-Excl}$ with the unique vector $p = (p_1, \ldots, p_m)$ of operation execution times. Define the number of operations $n = a$ and the number of stations $m = a - b + 1$. Operation $j$ is associated with the 3-element subset $A_j$, $j = 1, \ldots, a$. Define $p_{lj} = b/2 + 1$, $l = 1, \ldots, m-1$, and $p_{mj} = 1$, $j = 1, \ldots, n$. Exclusion relations are determined as follows. If element $e \in B$ appears in the 3-element subsets $A_i$ and $A_r$, then operations $i$ and $r$ exclude each other, i.e., a set $E_{e,\{i,r\}} = \{i, r\} \in E$ is introduced. For a given $e \in B$, the number of the corresponding sets $E_{e,\{i,r\}}$ does not exceed $O(a^2)$, thus, the total number of such sets is limited by $O(ba^2)$.

We will show that for the constructed instance, a solution to X3C exists if and only if there exists a solution to $\text{SCEN-L}(p_{ij} = p_l)\text{-Excl}$ with value $F(p, x) \leq b$. It can be seen that our reduction is polynomial and pseudo-polynomial at the same time.

**Part “if”**. Assume that there exists a solution $x$ to $\text{SCEN-L}(p_{ij} = p_l)\text{-Excl}$ with value $F(p, x) \leq b$. This implies that at most one operation is assigned to each of the stations $1, \ldots, m - 1$, where $m - 1 = a - b$, and at most $b$ operations are assigned to the station $m = a - b + 1$. Since there are $a$ operations, it follows that exactly one operation is assigned to each of the stations $1, \ldots, m - 1$ and exactly $b$ operations are assigned to the station $m$. Furthermore, no two operations $i$ and $r$ exist such that $\{i, r\} \in E$ are assigned to the station $m$. Therefore, no two subsets $A_i$ and $A_r$ corresponding to the operations assigned to the station $m$ share the same element. We deduce that the subsets $A_j$ corresponding to the $b$ operations assigned to station $m$ constitute an exact cover of $B$. Part “if” is proven.

**Part “only if”**. If there exists a solution $Y$ to X3C, then assign operation $j$ to station $m$ if and only if $A_j \in Y$. Each of the remaining $a - b$ operations is assigned to one of the stations
1, \ldots, m - 1, one operation to one station. Such an assignment, \( x \), is feasible with respect to the exclusion relations, and \( F(p, x) = b \), as required.

Let \( E = \{E_1, \ldots, E_u\} \), thus \( u = |E| \). For conformity of presentation, introduce artificial sets \( E_r = \{i_r\}, r = u + 1, \ldots, w \), where \( i_{a+1}, \ldots, i_w \) are all operations not involved in the exclusion relations. Denote \( e_r = |E_r|, r = 1, \ldots, w \). We have \( \sum_{r=1}^{w} e_r \leq u + n \). Introduce variable \( z_{lr} \), which is equal to the number of operations of the set \( E_r \) assigned to station \( l \), for \( l = 1, \ldots, m \) and \( r = 1, \ldots, w \). Problem \( \text{Scen-L}(p_{lj} = p_l)\)-EXCL can be formulated as follows.

\[
\min \quad F(z) = \max_{1 \leq l \leq m} \left\{ p_l^* \sum_{r=1}^{w} z_{lr} \right\},
\]
\[
\sum_{l=1}^{m} z_{lr} = e_r, \quad r = 1, \ldots, w,
\]
\[
z_{lr} \in \{0, 1, \ldots, \max\{1, e_r - 1\}\}, \quad l = 1, \ldots, m, \quad r = 1, \ldots, w.
\]

Problem (9)-(11) can be solved in \( O(m(u + n)^m) \) time by a modification of algorithm DP-Simple, in which there are states \((T_1, \ldots, T_{m-1}) \in Z_j \) such that \( T_l = \sum_{r=1}^{j} z_{lr} \). Thus, problem \( \text{Scen-L}(p_{lj} = p_l)\)-EXCL is solvable in \( O(mk + m(u + n)^m) \) time, which is polynomial for any constant \( m \).

4. Conclusions and suggestions for future research

The results of this paper are summarized in Table 1. There, “sNP” and “NP” are abbreviations for “strongly NP-hard” and “NP-hard”, respectively, and \( P = \max_{i,l} \left\{ \sum_{r=1}^{n} p_{lr}^{(i)} \right\} \).

<table>
<thead>
<tr>
<th>Problem</th>
<th>Complexity</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scen-L</td>
<td>sNP</td>
<td>Th. 1,2,3</td>
</tr>
<tr>
<td>Scen-L, const ( k ) and ( m )</td>
<td>NP, ( O(mnk^2 + mnkP^{km}) )</td>
<td>Sec. 2</td>
</tr>
<tr>
<td>Scen-L, ( k = 1 ), ( X = \overline{X} ), ( p_{lj} = p_j )</td>
<td>sNP</td>
<td>Th. 1</td>
</tr>
<tr>
<td>Scen-L, ( m = 2 ), ( X = \overline{X} ), ( p_{lj}^{(i)} \in {0, p_l^{(i)}} )</td>
<td>sNP</td>
<td>Th. 2</td>
</tr>
<tr>
<td>Scen-L((p_{lj} = p_l))-EXCL, (</td>
<td>E'</td>
<td>= 2, \forall E' \in E )</td>
</tr>
<tr>
<td>Scen-L((p_{lj} = p_l))-SIMPLE</td>
<td>( O(mk + m \log m) )</td>
<td>Sec. 3.1</td>
</tr>
<tr>
<td>Scen-L((p_{lj} = p_l))-PREC</td>
<td>( O(n +</td>
<td>U</td>
</tr>
<tr>
<td>Scen-L((p_{lj} = 1))-INCL</td>
<td>sNP</td>
<td>Sec. 3.3</td>
</tr>
</tbody>
</table>
Table 1: Computational complexity of problem Scen-L.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Complexity</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scen-L($p_{ij} = 1$)-Incl, const $m$, id-encoding</td>
<td>NP</td>
<td>Sec. 3.3</td>
</tr>
<tr>
<td>Scen-L($p_{ij} = p_l$)-Incl, const $m$</td>
<td>$O(k +</td>
<td>I</td>
</tr>
<tr>
<td>Scen-L($p_{ij} = p_l$)-Incl-Chains, const $m$</td>
<td>$O(mk +</td>
<td>U</td>
</tr>
<tr>
<td>Scen-L($p_{ij} = p_l$)-Excl, const $m$</td>
<td>$O(mk + (</td>
<td>E</td>
</tr>
</tbody>
</table>

NP-hardness proofs imply that it is unlikely to derive efficient algorithms for the general problem Scen-L. Therefore, practically relevant special cases should be handled by the problem specific algorithms, whose development is a subject of future research. We will concentrate on heuristic algorithms and efficient mathematical programming formulations, which can be handled by commercial solvers.

Polynomial time algorithms of low complexity are presented for problems Scen-L($p_{ij} = p_l$)-Simple and Scen-L($p_{ij} = p_l$)-Prec. They can be used as lower bound procedures for more general cases of the problem Scen-L.

Problem Scen-L($p_{ij} = p_l$)-Incl-Prec with a constant number of stations requires additional investigation to determine computational complexity of the cases in which graph $G_I$ is not a collection of chains.

Finally, models with min-max \textit{regret} and min-max \textit{relative regret} objectives, see Aissi, Bazgan and Vanderpooten [3] for definitions, can be appropriate for line balancing problems, and they are worth to be studied.

References


