Integer Optimization Model and Algorithm for the Stem Cell Culturing Problem

Jongyoon Park[a], Jinil Han[b,*], Kyungsik Lee[a,*]

[a]*Department of Industrial Engineering, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul, Republic of Korea*
[b]*Department of Industrial and Information Systems Engineering, Soongsil University, 369 Sangdo-ro, Dongjak-gu, Seoul, Republic of Korea*

Abstract

In this paper, we present a novel scheduling problem, the stem cell culturing problem (SCP), which is identified in an attempt to improve the productivity of a manufacturing system producing a commercialized autologous stem cell therapeutic product for treating an incurable disease. For a given therapeutic product along with the corresponding manufacturing process, which is called the stem cell culture process, the SCP is the problem of maximizing the number of produced units of a therapeutic product during a given horizon while satisfying the operational constraints stem from unique characteristics of the stem cell culture process and the related resources. After we formally define the SCP, we show that the SCP is NP-hard in the strong sense. Then, we present an integer optimization model based on the concept of a daily mode. The computational performance of the proposed model is analyzed with a general purpose integer optimization software. Moreover, based on this model, we propose an efficient LP-based heuristic algorithm which yields provably good solutions within a short time. Through computational experiments, we demonstrate that the proposed algorithm is both effective and efficient in solving practically-sized instances.

Keywords: Stem cell culturing problem, Resource constrained scheduling, Integer optimization, LP-based heuristic algorithm

1. Introduction

Stem cells refer to cells that have the capability to self-renew and differentiate into multiple types of cells (National Institutes of Health, 2016). Due to the recent technological advances in isolating and culturing stem cells, *stem cell therapeutic products* to treat incurable diseases have been being developed (Daley & Scadden, 2008; Biehl & Russell, 2009; National Institutes of Health, 2016). A number of stem cell therapeutic products are already approved for commercialization in various countries, and many studies are at the late-stage of clinical trials (Syed & Evans, 2013; Corestem, 2018).

A stem cell therapeutic product is manufactured by culturing a small amount of stem cells isolated from origin tissues extracted from a *donor*, and then injected to a *recipient* to treat a disease (Hipp & Atala, 2008; National Institutes of Health, 2016). A stem cell therapeutic product is called *autologous* if the recipient is the same person as the donor, which has been being developed for the treatment of cardiac diseases, nervous system diseases, and so on (Syed & Evans, 2013; Corestem, 2018). Since it uses adult stem cells that can be obtained from a patient's own origin tissues (e.g., bone marrow) without the destruction of

[*]Corresponding authors

Email addresses: `globell@snu.ac.kr` (Jongyoon Park), `jinil.han@ssu.ac.kr` (Jinil Han), `optima@snu.ac.kr` (Kyungsik Lee)

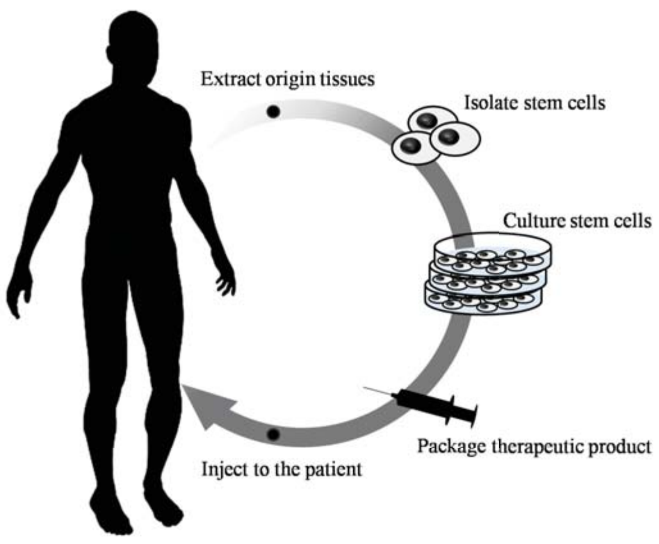


Figure 1: Autologous stem cell therapy procedure

human embryos, not only it can avoid the immunorejection, but also it is free from the ethical dilemma (Volarevic et al., 2018). An autologous stem cell therapy for a specific disease, as illustrated in Figure 1, starts with extracting a patient's origin tissues (e.g., bone marrow extraction through a lumbar puncture) in a hospital. The extracted origin tissues are delivered to a manufacturing facility where stem cells are isolated from the origin tissues and cultured through a specially designed *stem cell culture process*. Finally, the patient's stem cell therapeutic product produced in the manufacturing facility is delivered to the hospital and injected to the patient.

1.1. Stem Cell Culturing Problem

The main resource for the stem cell culture process is the *incubator* (a hardware device in which stem cells are cultured under specially controlled conditions). A manufacturing facility is equipped with a number of incubators (e.g., 20 incubators). An incubator is composed of a few *chambers* of possibly multiple types in terms of the capacity which is defined as the maximum number of *cell lots* that can be accommodated at the same time. A cell lot is a prespecified quantity of stem cells along with a certain amount of a growth medium put in a container such as a flask. Figure 2 is an example of an incubator consisting of 6 chambers of two types (two 24-lot chambers and four 18-lot chambers).

The stem cell culture process, which is approved by a regulatory agency, specifies the culturing duration (the number of days, D), which is typically long (e.g., 29 days), together with the task for each culturing day including dividing or re-dividing one patient's stem cells into a prespecified number of cell lots, the quality tests, and so on. The stem cell culture process begins on the first day (day 1) by dividing the isolated stem cells into a prespecified number of cell lots, and putting them into a number of chambers. On each day from day 2, the cell lots of each patient accommodated in a set of chambers are retrieved, and then again put into a set of chambers, which is not necessarily the same set of chambers as before, after the required task is done. On the last day (day D), the cultured stem cells of one patient are packaged into the patient's therapeutic product. The operational characteristics related to the culture process and the resources in the manufacturing facility are as follows.

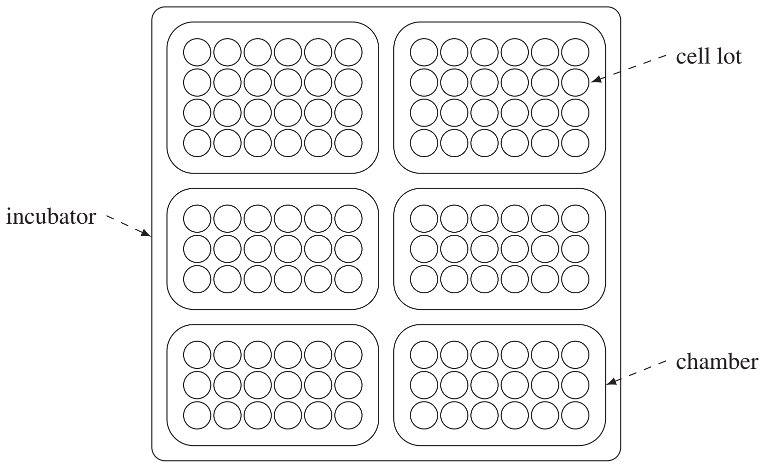


Figure 2: An illustration of an incubator, chambers, and cell lots

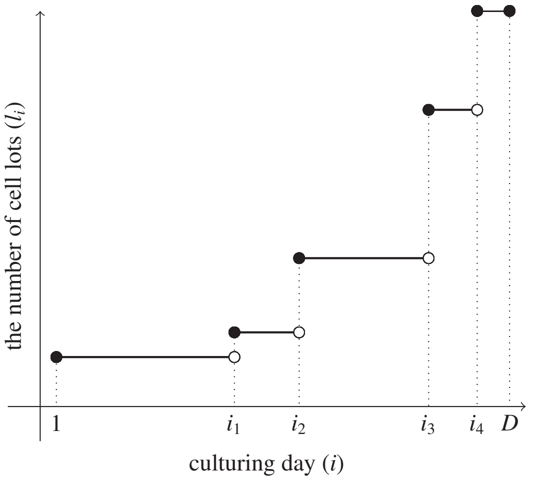


Figure 3: The number of cell lots on each culturing day

(C1) (*Non-preemptive process*) The process must be executed according to the approved specifications. It is not allowed to shorten the culturing period nor to cause a delay. Once the process for one patient starts on day 1, it should be completed exactly on day D.

(C2) (*Non-sharable resource*) On each day of the culturing period, the cell lots of one patient can be put into more than one chamber across several incubators. However, for safety reasons, each chamber must accommodate at most one patient's cell lots on each day. That is, even if a chamber has the remaining capacity after accommodating one patient's cell lots, it is not allowed to accommodate cell lots of other patients.

(C3) (*Multiple combinations of resources*) Since there are (possibly) multiple types of chambers in terms of the capacity, there are, in general, a number of combinations of chambers of different types to accommodate a given number of cell lots for each culturing day due to (C2). For example, 35 cell lots can be accommodated by two 24-lot chambers, or one 24-lot chamber and one 18-lot chamber, or two 18-lot chambers.

(C4) (*Time-varying resource demands and capacities*) As the process proceeds, stem cells proliferate so that the population density of a cell lot increases. To manage the population density, stem cells of each patient are re-divided on specific days into a larger number of cell lots than before. The timing of the re-divisions with the corresponding numbers of cell lots should follow the approved specifications. In general, the number of cell lots of one patient for each culturing day can be regarded as a step function over the culturing period. Figure 3 shows an example of such a function, in which the number of cell lots jumps on each of the four days, i_1, i_2, i_3, and i_4. Therefore, the resource demands vary (more precisely, monotonically increase) with time. The number of available chambers of each type may also vary with time due to the scheduled maintenance of incubators.

In the previously described autologous stem cell therapy procedure (Figure 1), the specific date for the first step (i.e., extracting a patient's origin tissues) is chosen by a doctor considering the patient's health condition from the available dates in a production schedule, which we call the *stem cell culture schedule*, provided by the manager of the manufacturing facility. Let us call a unit of the autologous therapeutic product for one potential patient is *admissible* to the manufacturing facility on a specific day in a planning horizon, if the culture process for it can start on the day and be completed within the planning horizon. A stem cell culture schedule, which is planned by the manufacturing manager, specifies the number of admissible units of the product for each day during a planning horizon (e.g., 6 months). That is, a doctor makes a reservation for a patient by assigning one of available admissible units of the product in the stem cell culture schedule to the patient.

In this paper, we study a novel scheduling problem, the *stem cell culturing problem* (SCP in short), which is described as follows. For a given therapeutic product for a specific disease along with the specifications of the corresponding stem cell culture process, the SCP is the problem of finding a stem cell culture schedule that maximizes the number of admissible units of the product during a given planning horizon while satisfying the operational characteristics, (C1) – (C4), related to the culture process and the resources in a given manufacturing facility. An increased number of admissible units during a given horizon means more patients can be treated over the same time periods. It also means higher throughput and reduced production cost per unit of the manufacturing facility, which leads to a reduced price for a therapeutic product (cf., the current price of one unit for a nervous system disease is around tens of thousands of dollars). Therefore, maximizing the number of admissible units during a given horizon directly contributes to improving the productivity as well as the service level of a medical system providing a commercialized autologous stem cell therapy.

In this study, we propose an integer optimization model and an efficient solution method for the SCP. As far as we know, our study is the first one that considers a scheduling problem arising in the emerging stem cell therapy industry. The rest of the paper is organized as follows. We first review the related literature and summarize the contributions of our study in section 2. In section 3, the computational complexity of the SCP is analyzed, and an integer optimization model for the SCP is presented. Then, we propose an LP-based heuristic algorithm in section 4. The computational performance of the integer optimization model and the heuristic algorithm is reported and discussed in section 5. Finally, concluding remarks are given in section 6.

2. Literature Review and Contributions

2.1. Literature Review

To the best of our knowledge, scheduling problems which have the same characteristics as those of the SCP have not been addressed in the existing literature. However, a few classes of problems sharing some of the characteristics of the SCP can be found in the literature.

Consider a special case where the culturing duration of the given stem cell culture process is one day (i.e., $D = 1$). For this special case, the SCP is reduced to the problem of selecting as many chamber combinations as possible while satisfying the number of available chambers of each type, which can be formulated as a multi-dimensional integer knapsack problem (Kellerer et al., 2004). However, scheduling problems with similar characteristics to those of the SCP have mainly been considered in the context of the resource constrained project scheduling problem (RCPSP).

For a given project consisting of a number of non-preemptive activities and multiple types of resources, the standard RCPSP is the problem of scheduling activities of the project so as to minimize the makespan subject to precedence relations between activities and resource availability constraints, which is shown to be strongly NP-hard by Blazewicz et al. (1983). The standard RCPSP and its practical extensions have been actively studied for over 50 years. For comprehensive reviews, we refer the readers to, for example, Brucker et al. (1999), Herroelen et al. (1998), Herroelen (2005), Kolisch & Padman (2001), Hartmann & Briskorn (2010), Węglarz et al. (2011), Schwindt & Zimmermann (2015), and Hartmann & Briskorn (2022). Among many important extensions of the standard RCPSP, more closely related with the SCP are the multi-mode RCPSP (Węglarz et al., 2011; Mika et al., 2015) and the RCPSP with time-varying resource demands and capacities (Hartmann, 2013, 2015).

In the context of the RCPSP, a *mode* is defined as a specific way of processing an activity, which is characterized by the processing duration of the activity (the number of time periods) and the corresponding resource requests (the number of required units of each resource for each time period during its processing duration). The multi-mode RCPSP is an extension of the standard RCPSP in which each activity of a project can be processed in one of given several different modes which reflect time-cost, time-resource, or resource-resource trade-offs. As far as we know, Pritsker et al. (1969) first considered multiple modes for an activity, which they called alternative methods (or resource combinations). Since then, the multi-mode RCPSP has been broadly studied. A few integer optimization models have been proposed (Talbot, 1982; Maniezzo & Mingozzi, 1999), and exact as well as heuristic algorithms have been devised (Talbot, 1982; Hartmann & Drexl, 1998; Hartmann, 2001; Zhu et al., 2006). The most up-to-date and comprehensive review on the multi-mode RCPSP can be found in Mika et al. (2015).

In the standard RCPSP as well as the multi-mode RCPSP, it is basically assumed that the amount of each resource required to process an activity for each time period during its processing duration and the available amount of each resource for each time period are constant. The RCPSP with time-varying resource demands and capacities (RCPSPt) is an extension of the standard RCPSP where the resource demand of an activity and the capacity (available amount) of each resource may vary over time (Hartmann, 2013, 2015). Compared to other extensions of the standard RCPSP, there have been fewer studies on the RCPSPt. Bartusch et al. (1988) showed that the RCPSPt can be transformed to the RCPSP with constant resource demands and capacities by splitting activities and imposing additional constraints which are called the generalized precedence constraints. Cavalcante et al. (2001) studied a labor constrained scheduling problem where the labor demand varies as a job is processed. Möhring & Uetz (2003) considered the time-varying resource requests in an application arising in the chemical industry. Hartmann (2013) formally defined the RCPSPt and presented its application to the medical research project. For reviews on the recent studies on the RCPSPt, we refer the readers to Hartmann & Briskorn (2010) and Hartmann (2015).

In view of the operational characteristics (C2) and (C3), the SCP has the characteristics of the multi-mode RCPSP with resource-resource trade-off. That is to say, the stem cell culture process for one admissible product can be viewed as an activity of a project which can be executed in one of a number of different modes: a sequence of D combinations of chambers corresponds to one mode for an activity, where the ith element of the sequence corresponds to a combination of chambers, which specifies the number of chambers for each type, to accommodate the cell lots on the ith culturing day. In the literature of the multi-mode RCPSP with resource-resource trade-off, most of the previous studies have considered a small number of modes (less than 5) for each activity (cf., Mika et al. (2015)). However, there exist exponentially many modes (sequences of D combinations of chambers) in the SCP even though the number of combinations for each culturing day is small: only two combinations of chambers for each culturing day lead to 2^D modes.

There have been a few studies on a special case of the multi-mode RCPSP, the so-called multi-skill project scheduling problem, in which a mode for an activity is defined as a subset of the given finite number of *unary* resources so that there may exist an exponential number of modes for each activity (cf., Drezet & Billaut (2008) and Li & Womer (2009)). In the multi-mode RCPSP, a non-sharable resource whose availability for each time period is limited to one unit is called unary (or disjunctive). A typical example of a unary resource is each one of a pool of personnel who can not be assigned to two or more activities at the same time. In Li & Womer (2009), each activity with a specified processing duration requires multiple skills, and each skill requires one individual selected from a pool of personnel who has the skill. Hence, a mode for an activity is defined as a set of individuals which collectively meets the skill requirement of the activity. In Drezet & Billaut (2008), a mode for an activity is similar to that considered in Li & Womer (2009), but the assigned set of individuals to an activity for one period can be different from those for the other periods during its processing duration. Both studies are similar to the SCP in that a mode for an activity is defined as a combination (or a sequence of combinations) of discrete resources. However, they differ from the SCP: only unary resources are considered and the resource demand is given as the number of units of resources, whereas, in the SCP, each type of chamber is not a unary resource and the resource demand is not defined as the number of chambers.

In view of the operational characteristic (C3), the SCP also has the characteristics of the RCPSPt as well as those of the multi-mode RCPSP with resource-resource trade-off. As far as we know, Drezet & Billaut (2008), which is already mentioned before, is the only previous study that considers resource-resource trade-off and time-varying resource demands. In this study, the number of required workers for an activity may vary over its processing duration. However, it is limited to unary resources.

Most of all, in the context of the RCPSP, all activities of a project must be scheduled, whereas the SCP aims to schedule as many stem cell culture processes (corresponding to activities of a project in the RCPSP) as possible within a given horizon. If the stem cell culture process of one admissible unit is regarded as a project rather than an activity of a project, this characteristic of the SCP can be found in the multi-project scheduling problem, where the projects to be carried out are selected from a set of candidate projects and the schedules of all activities of the selected projects are determined simultaneously (cf., Kolisch & Meyer (2006), Chen & Askin (2009), and Shariatmadari et al. (2017)). As far as we know, Kolisch & Meyer (2006) is the only previous study that considers a multi-project scheduling problem with multiple modes for activities. However, they considered multiple modes to reflect time-resource trade-off, not resource-resource trade-off.

2.2. Contributions

From the point of view of practical applications, we identify and present, for the first time, a novel scheduling problem, the SCP, arising in the emerging stem cell therapy industry. Our study on the SCP also contributes to the efficient operation of manufacturing systems producing autologous stem cell therapeutic products. As the stem cell therapy industry is in its early stage and is expected to grow, our study can serve as a starting point for future studies on the design and operation of manufacturing and service systems for providing autologous stem cell therapies.

This study also contributes to the existing literature on resource constrained scheduling problems from the point of view of models and solution methods. As reviewed in section 2.1, the SCP has similar characteristics to those of the multi-mode RCPSP and RCPSPt. However, as far as we know, there have been no previous studies which simultaneously consider 1) resource-resource trade-off with non-sharable non-unary resources, 2) the resource demand which is not given as the number of units of resources, and 3) time-varying resource demands and capacities. Our study fills this gap in the existing literature. The detailed contributions are summarized as follows.

- The computational complexity of the SCP is analyzed. Since the SCP is the problem of maximizing the number of identical activities with no precedence relations which can be completed during a given horizon, it may seem that the SCP is simpler and easier to solve than other scheduling problems studied in the context of the RCPSP. However, we show that the SCP is strongly NP-hard in general.

- We present an integer optimization model for the SCP which is based on the concept of a daily mode. The computational performance of the model including the strength of upper bounds provided by its LP-relaxation is analyzed with a general purpose integer optimization software.

- We propose an efficient LP-based heuristic algorithm which yields provably good solutions quickly. Through computational experiments, we demonstrate that the proposed algorithm is both effective and efficient in solving practically-sized instances.

3. Computational Complexity and Mathematical Formulation

Prior to presenting an integer optimization model for the SCP which has been described in section 1.1, we define parameters and sets to be used to formally state the problem. Recall that the SCP is the problem of maximizing the number of units of the given therapeutic product that can be produced during a given planning horizon while satisfying the operational characteristics (C1) - (C4).

First, the planning horizon of the SCP is discretized into τ time periods, each of which corresponds to one day, and let $T := \{1, \ldots, \tau\}$ be the set of days of the planning horizon. For the given specifications of the culture process of the given therapeutic product, the required number of days (the culturing duration) for each unit of the product is D days. Let $I := \{1, \ldots, D\}$ be the set of days of the culturing duration. Recall that l_i for each $i \in I$ denotes the number of cell lots on the ith culturing day, which is non-decreasing

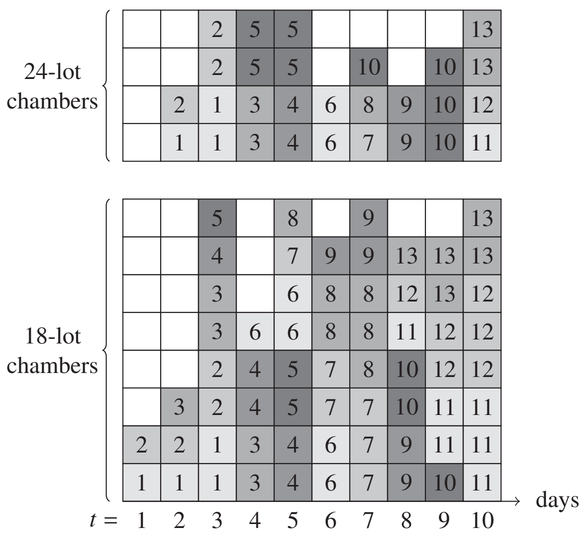


Figure 4: Graphical illustration of a solution of an instance of Example 3.1.

as shown in Figure 3. A manufacturing facility is equipped with a number of incubators, each of which consists of chambers of (possibly) different types in terms of the capacity. Let n be the number of different types of chambers over all the given incubators, and let $R := \{1, \dots, n\}$ be the set of types of chambers. For each $r \in R$, the capacity of a chamber (the maximum number of cell lots that can be accommodated at the same time) is denoted as c_r. The number of chambers of each type may vary over time, so we let a_{rt} be the number of available chambers of type r on day t, for all $r \in R$ and $t \in T$. Since the culture process requires D days, any unit of the product can not be admissible from day $\tau - D + 2$. Hence, we define $\overline{T} := \{1, \dots, \tau - D + 1\}$, the set of days on which the culture process for any admissible unit of the product must start. To facilitate a better understanding of the SCP, a simple but illustrative instance of the SCP is given as follows.

Example 3.1. *Suppose that the planning horizon is set to $\tau = 10$ days, i.e., $T := \{1, \dots, 10\}$, and that the given culture process requires $D = 3$ days, i.e., $I := \{1, 2, 3\}$ and $\overline{T} := \{1, \dots, 8\}$. The number of cell lots on culturing days are given as $l_1 = 10$, $l_2 = 25$, and $l_3 = 75$. Two incubators are available, each of which consists of four 18-lot chambers and two 24-lot chambers. That is, two types of chambers are available, i.e., $n = 2$, $R := \{1, 2\}$, $c_1 = 18$, and $c_2 = 24$. We assume that all the chambers are available for the entire planning horizon. Hence, $a_{1t} = 8$ and $a_{2t} = 4$ for all $t \in T$.*

Figure 4 graphically shows a feasible solution of the instance of the SCP given in Example 3.1. The lower grid consists of 8 rows where each row represents one 18-lot chamber. Each of the 4 rows of the upper grid represents one 24-lot chamber. Each column of the grid corresponds to one day in the planning horizon. The number indicated in each square shows the unit number which occupies a chamber. That is, a square without any numbers is a vacant chamber. For example, the culture process of unit 3, which starts on day 2, occupies one 18-lot chamber (cf., $l_1 = 10$) on its first culturing day (day 2), two 18-lot chambers (cf., $l_2 = 25$) on its second culturing day (day 3), and two 18-lot chambers and two 24-lot chambers (cf., $l_3 = 75$) on its last culturing day (day 4). Each column shows the status of occupancy of chambers on each day in the planning horizon. For example, on day 2, each of the three units (1, 2, and 3) occupies one 18-lot chamber, and two 24-lot chambers are occupied by unit 1 and unit 2. The objective value of the feasible solution shown in Figure 4 is 13, and in fact it is an optimal solution of the example instance.

Now, we analyze the computational complexity of the SCP. It may seem that the SCP has a simpler structure compared to other resource constrained scheduling problems. However, it can be shown that the SCP is NP-hard in the strong sense.

Theorem 3.1. *The SCP is strongly NP-hard.*

Proof. We show that every instance of the well-known 3-PARTITION problem, which is NP-complete in the strong sense (Garey & Johnson, 1979), can be solved by solving an instance of the SCP. An instance of 3-PARTITION consists of a finite set A of $3m$ elements, a bound $B \in \mathbb{Z}_+$, and a weight $w(a) \in \mathbb{Z}_+$ for each $a \in A$ such that $B/4 < w(a) < B/2$ and $\sum_{a\in A} w(a) = mB$. The problem is to determine whether A can be partitioned into m disjoint subsets, $S_1, S_2, \ldots, S_m$, such that $\sum_{a\in S_i} w(a) = B$ for $1 \leq i \leq m$ (Note that each subset must contain exactly three elements).

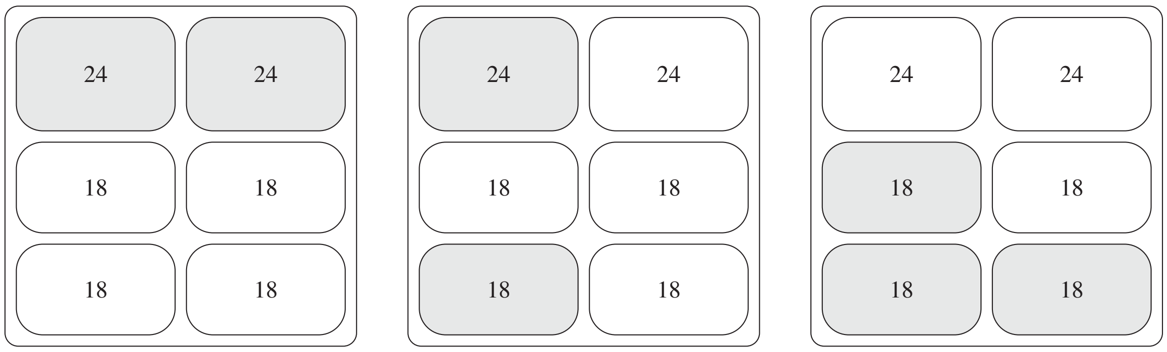
For any given instance of 3-PARTITION, let γ be the number of distinct weights of elements. Then, A can be partitioned into γ disjoint subsets, $A_1, A_2, \ldots, A_\gamma$, each of which contains those elements of A having the same weight. Let $w(A_r)$ be the weight of the elements in A_r for each $r = 1, \cdots, \gamma$. Now the corresponding instance of the SCP can be constructed as follows. First, set the planning horizon $\tau = 1$, and the culturing duration $D = 1$ with $l_1 = B$. Then, set $n = \gamma$, and set $c_r = w(A_r)$ for each $r \in R := \{1, \ldots, n\}$. Finally, set $a_{rt} = |A_r|$ for each pair of $r \in R$ and $t \in T$. From the construction of the instance of the SCP, the objective value of a solution can not exceed m, so that the answer to the given instance of 3-PARTITION is affirmative if and only if the optimal objective value of the SCP is m. Therefore, the result follows. □

Now, we present an integer optimization model which is based on the concept of a *daily mode*. A daily mode p for the ith culturing day is defined as a combination of chambers which is enough to accommodate l_i cell lots, and it is represented by an n-dimensional nonnegative integer vector v^p that satisfies the following *capacity requirement constraint* (1) for the ith culturing day.

$$\sum_{r\in R} c_r v_r^p \geq l_i, \tag{1}$$

where v_r^p represents the number of chambers of type r used by daily mode p. As mentioned in section 2.1, in the existing literature of the RCPSP, the terminology 'mode' is used to refer to a specific way of processing an activity of a project. A daily mode is a similar concept to a 'mode' in the RCPSP in that it specifies one option to process some given task (the task of the culturing process on one specific culturing day). However, if the culture process for one unit of the product is regarded as an activity of a project in the context of the RCPSP, a sequence of D daily modes corresponds to one 'mode' for the activity, which means that a daily mode is a part of one 'mode'. Therefore, we used the terminology 'daily mode' rather than just 'mode'.

To formulate an integer optimization model, we use integer variables each of which represents the number of times that a daily mode is used on some culturing day. Note that the capacity requirement constraint (1) for each culturing day is a part of the constraints of the SCP that should be satisfied in deciding the number of chambers of each type to be used on each culturing day for the culture process of one unit. Therefore, a daily mode can be interpreted as a combinatorial structure that satisfies a part of the constraints of the SCP. Such combinatorial structures relevant to given problems have been frequently used as decision variables to formulate integer optimization models. The primary purpose of those formulations is to obtain stronger LP-relaxation, and hence better performance in a branch-and-bound based algorithm (Vanderbeck & Wolsey, 2010). Examples can be found in various optimization problems: a feasible cutting pattern in the cutting stock problems (Gilmore & Gomory, 1961; Arbib & Marinelli, 2014), a feasible delivery route in the vehicle routing problems (Desrochers et al., 1992; Li et al., 2019), a feasible sequence of flights in



(a) Daily mode p with $v^p = (0, 2)$: two 24-lot chambers.

(b) Daily mode p with $v^p = (1, 1)$: one 24-lot chamber and one 18-lot chamber.

(c) Daily mode p with $v^p = (3, 0)$: three 18-lot chambers.

Figure 5: An example of minimal daily modes for two types of chambers (four 18-lot chambers and two 24-lot chambers) when the number of cell lots is 40.

the airline scheduling problems (Vance et al., 1997; Parmentier & Meunier, 2020; Cacchiani & Salazar-González, 2020; Breugem et al., 2022), a feasible trip sequence for the planning of train units (Gao et al., 2020), a job schedule for a machine in the parallel machine scheduling problems (van den Akker et al., 1999; Wang et al., 2018), to name a few. For an extensive list of references, we refer the readers to Barnhart et al. (1998), Desaulniers et al. (2005), and Vanderbeck & Wolsey (2010).

Suppose that we have four 18-lot chambers and two 24-lot chambers and the number of cell lots on a specific culturing day is 40. Consider a daily mode p for the culturing day with two 18-lot chambers and one 24-lot chamber, i.e., $v^p = (2, 1)$. Note that a chamber combination with one 18-lot chamber and one 24-lot chamber is another daily mode for the culturing day, which uses one less 18-lot chamber. Hence, daily mode p has a superfluous chamber (i.e., one 18-lot chamber). To maximize the number of admissible units during the planning horizon, we do not have to use more chambers than necessary. Therefore, not all but daily modes with no superfluous chamber are sufficient to formulate our integer optimization model for the SCP. We call a daily mode with no superfluous chamber *minimal*, which is formally defined as follows.

Definition 3.1. *For a given daily mode p with the corresponding* $v^p \in \mathbb{Z}^n_+$, *p is minimal if* $c_s(v^p_s - 1) + \sum_{r\in R\setminus\{s\}} c_r v^p_r < l_i$ *for some* $s \in R$ *such that* $v^p_s > 0$.

Figure 5 shows the minimal daily modes with four 18-lot chambers and two 24-lot chambers when the number of cell lots is 40 ($l_i = 40$). Note that there is no minimal daily modes other than the three daily modes shown in the figure.

Let $P(i)$ be the set of all the minimal daily modes for the ith culturing day for each $i \in I$. For each $t \in \overline{T}$, we define $P(i, t)$ as the set of minimal daily modes for the ith culturing day of the culturing process started on day t, i.e., $P(i, t) := \{p \in P(i) : v^p_r \leq a_{r(t+i-1)},\ \forall r \in R\}$ for each $i \in I$. The integer variable λ_{pit}, which we call a *daily mode variable*, represents the number of times that daily mode $p \in P(i, t)$ is used on the culturing day $i \in I$ for the units of the product whose culture processes start on day $t \in \overline{T}$. The integer variables z_t represents the number of selected units whose culture processes start on day $t \in \overline{T}$. Using these variables, our integer optimization model for the SCP, which we call P_{DM}, is formulated as

$$
\begin{aligned}
(P_{DM}) \quad \text{maximize} \quad & \sum_{t\in\overline{T}} z_t && (2)\\
\text{subject to} \quad & z_t = \sum_{p\in P(i,t)} \lambda_{pit} \quad \forall\, i \in I, t \in \overline{T}, && (3)\\
& \sum_{i\in I} \sum_{p\in P(i,t-i+1)} v_r^p \lambda_{pi(t-i+1)} \le a_{rt} \quad \forall\, r \in R, t \in T, && (4)\\
& z_t \in \mathbb{Z}_+, \quad \forall\, t \in \overline{T}, \\
& \lambda_{pit} \in \mathbb{Z}_+ \quad \forall\, p \in P(i,t), i \in I, t \in \overline{T}.
\end{aligned}
$$

The objective function (2) is to maximize the number admissible units during the planning horizon. Constraints (3) ensure that the number of units started on each day $t \in \overline{T}$, z_t, should be equal to the number of times that daily modes are used on their ith culturing day, $\sum_{p\in P(i,t)} \lambda_{pit}$, for all $i \in I$. Constraints (4) mean that the total number of chambers of type r used by daily modes for all the selected units whose culture processes are in progress on day t should not exceed the number of available chambers of the same type, for all $r \in R$ and $t \in T$. Note that, on day t, the culture process started on day $t - i + 1$ is on its ith culturing day. Hence, if $t - i + 1 < 1$ or $t - i + 1 > \tau - D + 1$, $P(i, t - i + 1)$ is regarded as an empty set.

In theory, there may exist an exponential number of minimal daily modes for each culturing day when n (the number of types of chamber) is large, so that it is not viable to enumerate minimal daily modes in advance. However, in practice, the number of types of chamber in a manufacturing facility is small (e.g., 2 types), and the number of minimal daily modes is also small (2 - 5 daily modes for each culturing day). Although the number of minimal daily modes is small, as will be shown in the computational experiments, it is not easy to find optimal solutions within a small amount of time for large-sized instances (e.g., 6-month or 1-year horizon) by solving P_{DM} directly using one of state-of-the-art optimization softwares. Therefore, in the next section, we propose a heuristic algorithm to find good quality solutions quickly based on optimal solutions to the LP-relaxation of P_{DM}.

4. LP-Based Heuristic Algorithm

The proposed heuristic algorithm makes use of optimal solutions of the LP-relaxation of P_{DM} and consists of the following two steps. In the first step, we first construct an integer solution based on the rounded-down LP solutions of the daily mode variables. The idea is that we can expect that daily modes with positive LP solution values are more likely to be used in an optimal culture schedule than other daily modes. We then consider the remaining problem where the available number of chamber has been appropriately updated after assigning chambers according to the integer solution obtained in the first step. In the second step, an improved integer solution is constructed by iteratively assigning remaining chambers to possible units in a greedy manner until no further improvement is possible. We first describe the basic LP-based heuristic algorithm in section 4.1, then present an improved heuristic algorithm in section 4.2.

4.1. Basic LP-based Heuristic

The overall procedure of the basic LP-based heuristic algorithm is described in Algorithm 1, which consists of four parts, i.e., initialization (lines 1 - 2), constructing an initial solution (lines 3 - 4), updating the availability of chambers (lines 5 - 8), and finding an improved solution (lines 9 - 11).

Algorithm 1: BasicLPbasedHeuristic

```
1  $Q(i,t) \leftarrow$ list of daily modes in the set $P(i,t)$ sorted in ascending order of loss, $\forall i \in I, i \in \overline{T}$
2  $Q \leftarrow \{Q(i,t) : \forall i \in I, t \in \overline{T}\}$
3  Solve the LP-relaxation of $P_{DM}$ and get an optimal solution $(\lambda^{LP}, \mathbf{z}^{LP})$
4  $(\lambda^{RA}, \mathbf{z}^{RA}) \leftarrow$ RoundAdjust$(\lambda^{LP}, Q)$
5  $\hat{a}_{rt} \leftarrow 0$, for all $r \in R$ and $t \in T$
6  for $r \in R$ and $t \in T$ do
7      $\hat{a}_{rt} \leftarrow a_{rt} - \sum_{i \in I} \sum_{p \in P(i,t-i+1)} v_r^p \lambda^{RA}_{pi(t-i+1)}$
8  end
9  $(\lambda^{AG}, \mathbf{z}^{AG}) \leftarrow$ Augment$(\hat{\mathbf{a}}, Q)$
10 $\lambda^H \leftarrow \lambda^{RA} + \lambda^{AG}$, $\mathbf{z}^H \leftarrow \mathbf{z}^{RA} + \mathbf{z}^{AG}$
11 return $(\lambda^H, \mathbf{z}^H)$
```

Prior to a detailed explanation of the algorithm, we define the *loss* of a daily mode $p \in P(i,t)$ for each pair of $i \in I$ and $t \in \overline{T}$ as the unused capacity of the daily mode, i.e., $(\sum_{r \in R} c_r v_r^p) - l_i$. For each $i \in I$ and $t \in \overline{T}$, let $Q(i,t)$ be the sorted list of daily modes of $P(i,t)$ in ascending order of the loss of each daily mode of which the dth element is denoted as $Q(i,t)[d]$, for $d = 1, \dots, |P(i,t)|$. We also let Q be the collection of all of those sorted lists. In the first part of BasicLPbasedHeuristic (initialization), the sorted lists, $Q(i,t)$ for all $i \in I$ and $t \in \overline{T}$, and the collection of those sorted lists, Q, are constructed.

Next, in the second part, after solving the LP-relaxation of P_{DM} to obtain an optimal LP solution $(\lambda^{LP}, \mathbf{z}^{LP})$, an integer solution $(\lambda^{RA}, \mathbf{z}^{RA})$ is constructed based on the rounded-down values of λ^{LP} using RoundAdjust procedure, which is described in Procedure 1. Note that $\lfloor \lambda^{LP} \rfloor$ may not yield a feasible integer solution, where $\lfloor \lambda^{LP} \rfloor$ be the integer vector obtained by rounding down every component of λ^{LP}, since constraint (3) can be violated, i.e., the number of daily modes used on the jth culturing day for units started on day t, $\sum_{p \in P(j,t)} \lfloor \lambda^{LP}_{pjt} \rfloor$, may not be the same as that on some other culturing day $i \in I \setminus \{j\}$. To construct a feasible integer solution from $\lfloor \lambda^{LP} \rfloor$, we compute the minimum number of units whose culture processes can start on day t, $z_t^{RA} := \min_{i \in I} \sum_{p \in P(i,t)} \lfloor \lambda^{LP}_{pit} \rfloor$, for each $t \in \overline{T}$ (line 3 of Procedure 1). Then, for each pair of $t \in \overline{T}$ and $i \in I$, we assign daily modes to be used on the ith culturing day for z_t^{RA} units started on day t. To do that, each daily mode $p \in P(i,t)$ is considered, one by one, in the order stored in the list $Q(i,t)$, and set $\lambda^{RA}_{pit} := \lfloor \lambda^{LP}_{pit} \rfloor$ until $\sum_{p \in P(i,t)} \lambda^{RA}_{pit}$ is equal to z_t^{RA} (lines 5 - 10 of Procedure 1).

Then, in the third part, we update the number of available chambers of each type to prepare for the next part. Let $\hat{a}_{rt}$ be the number of available chambers of type r that can be used on day t after assigning chambers of each type according to λ^{RA}, which can be computed as $a_{rt} - \sum_{i \in I} \sum_{p \in P(i,t-i+1)} v_r^p \lambda^{RA}_{pi(t-i+1)}$ for each $r \in R$ and $t \in T$.

Finally, in the fourth part, we examine whether additional units can be produced with those residual chambers over the planning horizon using Augment procedure, which is described in Procedure 2. The idea is to repeatedly find and add additional units using those residual chambers in a greedy manner: repeatedly find the maximum number of additional units (π) with the corresponding day $\hat{t}$ over all $t \in \overline{T}$ on which those π units can start their culture processes, and add those π units on day $\hat{t}$. In Augment$(\hat{\mathbf{a}}, Q)$ procedure, π is heuristically computed. For each planning day $t \in \overline{T}$ and culturing day $i \in I$, we first compute w_{it}, which represents the number of additional units that can be accommodated in the residual chambers on the ith culturing day if their culture processes start on day t (lines 5 - 12 of Procedure 2). Suppose that the current availability of chambers is given as $\bar{a}_{rt}$ for all $r \in R$ and $t \in T$. A daily mode $p \in P(i,t)$, for the

Procedure 1: RoundAdjust($\boldsymbol{\lambda}^{LP}, Q$)

```
1  $\boldsymbol{\lambda}^{RA} \leftarrow \mathbf{0}, \mathbf{z}^{RA} \leftarrow \mathbf{0}$
2  for $t \in \overline{T}$ do
3      $z_t^{RA} \leftarrow \min_{i \in I} \sum_{p \in P(i,t)} \lfloor \lambda_{pit}^{LP} \rfloor$
4      for $i \in I$ do
5          $target \leftarrow z_t^{RA}$
6          for $d := 1, \ldots, |P(i,t)|$ do
7              $p \leftarrow Q(i,t)[d]$
8              $\lambda_{pit}^{RA} \leftarrow \min\{target, \lfloor \lambda_{pit}^{LP} \rfloor\}$
9              $target \leftarrow target - \lambda_{pit}^{RA}$
10         end
11     end
12 end
13 return $(\boldsymbol{\lambda}^{RA}, \mathbf{z}^{RA})$
```

ith culturing day of a unit started on day t, can be used up to $\min_{r \in R} \lfloor \bar{a}_{r(t+i-1)} / v_r^p \rfloor$. Hence, to compute w_{it}, each daily mode $p \in P(i,t)$ is considered, one by one, in the order stored in the list $Q(i,t)$, and increase w_{it} by $\min_{r \in R} \lfloor \bar{a}_{r(t+i-1)} / v_r^p \rfloor$, then update $\bar{a}_{r(t+i-1)}$ for all $r \in R$. Then, for each day $t \in \overline{T}$, the number of units that can be added on the day can be computed as $\min_{i \in I} w_{it}$. After that, $\pi = \max_{t \in \overline{T}} \min_{i \in I} w_{it}$ with the corresponding $\hat{t}$ is computed (line 13 of Procedure 2). Note that it is guaranteed that π additional units can be produced by starting the culture process on day $\hat{t}$ under the given availability of chambers. After adding π units to the current feasible solution ($\boldsymbol{\lambda}^{AG}, \mathbf{z}^{AG}$), we update the availability of chambers, i.e., $\hat{a}_{rt}$ (lines 15 - 27 of Procedure 2). The whole steps are repeated until no more additional units can be accommodated, i.e., $\pi = 0$. The final heuristic solution ($\boldsymbol{\lambda}^{H}$, $\mathbf{z}^{H}$) is obtained, at line 10 of BasicLPbasedHeuristic, by combining feasible solutions ($\boldsymbol{\lambda}^{RA}$, $\mathbf{z}^{RA}$) and ($\boldsymbol{\lambda}^{AG}$, $\mathbf{z}^{AG}$).

4.2. Improvements

For a small-sized instance, BasicLPbasedHeuristic algorithm may find a good solution. For example, for the small instance given in Example 3.1, it found a solution with its objective value being equal to 12 (cf., the optimal objective value is 13). However, as will be shown in the computational results, BasicLPbased-Heuristic algorithm does not perform well in finding good quality feasible solutions possibly because it works in a greedy manner. Hence, in this subsection, we present an improved LP-based heuristic algorithm employing two strategies to improve the solution quality for the large-sized practical instances.

4.2.1. Introduction of the greed coefficient α

Recall that Procedure 2, Augment($\hat{\mathbf{a}}$, Q), adds as many units as possible given residual chambers at each iteration. This is where the greedy nature comes in. However, it is often the case that in various scheduling problems, if the resource consumption in the process of an algorithm is too greedy, the quality of resulting solutions become poor. To remedy this, we introduce the *greed coefficient* α ($0 < \alpha \leq 1$) that controls the level of greed. A modified version of Procedure 2, ControlledAugment($\hat{\mathbf{a}}, Q, \alpha$), is devised by replacing line 15 of Augment($\hat{\mathbf{a}}$, Q) with

$$z_{\hat{t}}^{AG} \leftarrow z_{\hat{t}}^{AG} + \max\{1, \lfloor \alpha\pi \rfloor\}.$$

Procedure 2: AUGMENT($\hat{\mathbf{a}}$, Q)

```
$\lambda^{AG} \leftarrow \mathbf{0}, \mathbf{z}^{AG} \leftarrow \mathbf{0}$
repeat
    $w_{it} \leftarrow 0$ , $\forall i \in I, t \in \overline{T}$
    $\bar{\mathbf{a}} \leftarrow \hat{\mathbf{a}}$
    for $t \in \overline{T}$ and $i \in I$ do
        for $d := 1, \dots, |P(i,t)|$ do
            $p \leftarrow Q(i,t)[d]$
            $temp \leftarrow \min_{r \in R} \lfloor \bar{a}_{r(t+i-1)} / v_r^p \rfloor$
            for $r \in R$ do $\bar{a}_{r(t+i-1)} \leftarrow \bar{a}_{r(t+i-1)} - v_r^p \times temp$
            $w_{it} \leftarrow w_{it} + temp$
        end
    end
    $\pi \leftarrow \max_{t \in \overline{T}} \min_{i \in I} w_{it}$, $\hat{t} \leftarrow argmax_{t \in \overline{T}} \min_{i \in I} w_{it}$
    if $\pi \neq 0$ then
        $z_{\hat{t}}^{AG} \leftarrow z_{\hat{t}}^{AG} + \pi$
        for $i \in I$ do
            $target \leftarrow \pi$
            for $d := 1, \dots, |P(i,t)|$ do
                $p \leftarrow Q(i,t)[d]$
                $temp \leftarrow \min\{target, \min_{r \in R} \lfloor \hat{a}_{r(\hat{t}+i-1)} / v_r^p \rfloor\}$
                $\lambda_{pi\hat{t}}^{AG} \leftarrow \lambda_{pi\hat{t}}^{AG} + temp$
                for $r \in R$ do
                    $\hat{a}_{r(\hat{t}+i-1)} \leftarrow \hat{a}_{r(\hat{t}+i-1)} - v_r^p \times temp$
                end
                $target \leftarrow target - temp$
            end
        end
    end
until $\pi = 0$
return $(\lambda^{AG}, \mathbf{z}^{AG})$
```

Note that, even if $\pi \geq 1$, $\lfloor \alpha\pi \rfloor$ can be 0 for some $0 < \alpha \leq 1$. Hence, we add at each iteration $\max\{1, \lfloor \alpha\pi \rfloor\}$ instead of $\lfloor \alpha\pi \rfloor$ for finite termination. As α increases, the number of added units at each iteration increases. If we set $\alpha = 1$, then the modified procedure is the same as Procedure 2. Our computational results show that the use of the greed coefficient improves the quality of heuristic solution significantly. For the small-sized instance given in Example 3.1, a solution with its objective value being 13 (that is, it is an optimal solution) was found with $\alpha = 0.5$.

4.2.2. Repeated applications of LP-based heuristic

The assignment of daily modes, λ^H_{pit}, for each culturing day i, to z^H_t units that start on day t can be myopic in that Procedure 1 (cf., lines 6 - 10) and Procedure 2 (cf., lines 18 - 26) assign daily modes in ascending order of their losses. If the assignment of daily modes can be altered in such a way that more units can be possibly accommodated, the quality of heuristic solutions can be further improved. One possible way is starting from a different LP solution for a better chamber assignment considering the on-hand heuristic solution $(\boldsymbol{\lambda}^H, \mathbf{z}^H)$. To do that, we add the following bound constraints to P_{DM}.

$$z_t \geq z^H_t, \quad \forall t \in \overline{T},$$

to get an LP solution at least as good as the heuristic solution with possibly more efficient chamber assignment. Since it is expected that daily modes with positive LP solution values are more likely to be used in an optimal solution, applying both Procedure 1 and Procedure 2 again may yield a better heuristic solution. Our computational results show that this strategy also improves the quality of heuristic solution. For the small-sized instance given in Example 3.1, a solution with its objective value being 13 (that is, it is an optimal solution) was also found.

4.3. Improved LP-based Heuristic

By employing two improvement strategies given in the previous subsections, we devised an improved LP-based heuristic, ImprovedLPbasedHeuristic(A, $itrs$), which is described in detail as Algorithm 2, where the first argument A is a given set of greed coefficients and the second one is a given limit on the number of repeated applications. When applying the greed coefficient α, the performance of the algorithm may vary depending on the value of α. Therefore, it can be more effective to select the best one among those heuristic solutions obtained by applying ControlledAugment($\hat{\mathbf{a}}, Q, \alpha$) for multiple α values (lines 11 - 13 of Algorithm 2). That is why a set of greed coefficients is given as an argument. In our computational experiments, we could get high quality solutions when $A := \{0.1, 0.2, 0.3, 0.4, 0.5\}$.

Furthermore, the quality of heuristic solutions can be further improved by applying repeated applications of RoundAdjust($\boldsymbol{\lambda}^{LP}, \mathbf{z}^{LP}$) and ControlledAugment($\hat{\mathbf{a}}, Q, \alpha$) with different LP solutions until no further improvement is possible or the number of iterations reaches a given limit ($itrs$). Hence, Algorithm 2 is devised accordingly (lines 4 - 21). If we set $A := \{1\}$ and $itrs := 1$, then Algorithm 2 is the same as Algorithm 1.

5. Computational Experiments

In this section, we present the results of our computational experiments. General experimental settings are described in section 5.1, and then the computational performance of the proposed integer optimization model is tested in section 5.2. In section 5.3, the performance of the LP-based heuristic algorithm is presented. Finally, we analyze the sensitivity of the computational performance of the LP-based heuristic algorithm to input parameters related to the characteristics of the stem cell culture process and resource configuration in section 5.4.

Algorithm 2: ImprovedLPbasedHeuristic(A, $itrs$)

```
1  $Q(i,t) \leftarrow$ list of daily modes in the set $P(i,t)$ sorted in ascending order of loss, $\forall i \in I, i \in \overline{T}$
2  $Q \leftarrow \{Q(i,t) : \forall i \in I, t \in \overline{T}\}$
3  $counter \leftarrow 0$, $bestval \leftarrow 0$, $improved \leftarrow 0$
4  repeat
5      $counter \leftarrow counter + 1$
6      Solve the LP-relaxation of $P_{DM}$ and get an optimal solution $(\boldsymbol{\lambda}^{LP}, \mathbf{z}^{LP})$
7      $(\boldsymbol{\lambda}^{RA}, \mathbf{z}^{RA}) \leftarrow$ RoundAdjust($\boldsymbol{\lambda}^{LP}$, $Q$)
8      for $t \in T$ and $r \in R$ do
9          $\hat{a}_{rt} = a_{rt} - \sum_{i \in I} \sum_{p \in P(i,t-i+1)} v_r^p \lambda^{RA}_{pi(t-i+1)}$
10     end
11     for $\alpha \in A$ do
12         $(\boldsymbol{\lambda}^{\alpha}, \mathbf{z}^{\alpha}) \leftarrow$ ControlledAugment($\hat{\mathbf{a}}$, $Q$, $\alpha$)
13     end
14     $\hat{\alpha} \leftarrow argmax_{\alpha \in A} \sum_{t \in \overline{T}} z_t^{\alpha}$
15     if $bestval < \sum_{t \in \overline{T}} (z_t^{RA} + z_t^{\hat{\alpha}})$ then
16         $improved \leftarrow 1$
17         $\boldsymbol{\lambda}^H \leftarrow \boldsymbol{\lambda}^{RA} + \boldsymbol{\lambda}^{\hat{\alpha}}$, $\mathbf{z}^H \leftarrow \mathbf{z}^{RA} + \mathbf{z}^{\hat{\alpha}}$
18         $bestval \leftarrow \sum_{t \in \overline{T}} z_t^H$
19         add the constraints $z_t \geq z_t^H, \ \forall t \in \overline{T}$ to $P_{DM}$
20     end
21 until $(improved = 0)$ or $(counter > itrs)$
22 return $(\boldsymbol{\lambda}^H, \mathbf{z}^H)$
```

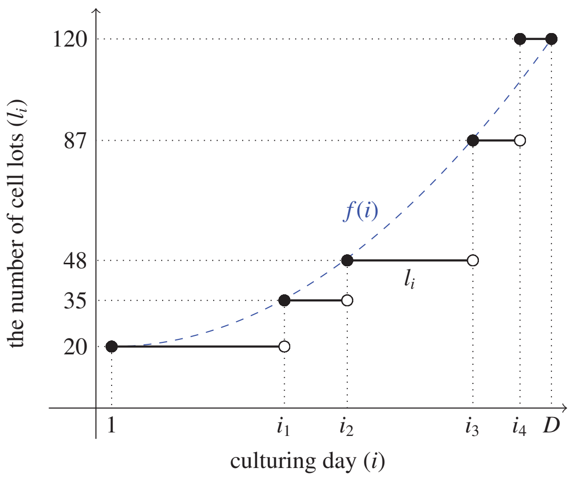


Figure 6: An example of a step function representing a cell lot profile generated with $b = 2$

5.1. Experimental settings

The subsequent computational tests were conducted on artificially generated instances which reflect real situation as closely as possible. First, for the planning horizon, we consider three alternatives in sections 5.2 and 5.3, i.e., $\tau = 90, 180$, and 360 days. In our practical experience, the planning horizon was up to six months. The reason that we consider the planning horizon of 360 days in our experiments is to test the performance of the proposed model and heuristic algorithm on larger-sized instances. For the sensitivity analysis in section 5.4, we set $\tau = 180$ days.

For the specifications of the stem cell culture process of the given therapeutic product, the number of culturing days (D) is set to 29, i.e., $I := \{1, \ldots, 29\}$. The number of cell lots for each culturing day, which we call a *cell lot profile* hereafter, is generated as follows. The numbers of cell lots on the first and the last culturing days, l_1 and l_D, are set to 20 and 120, respectively, and then, the number of cell lots on the other culturing days are set as the following method. As described in section 1, the number of cell lots jumps on each of a couple of culturing days (see Figure 3). Hence, we randomly choose four different culturing days, i_1, i_2, i_3, and i_4, such that $1 < i_1 < i_2 < i_3 < i_4 < D$. Then, the number of cell lots, l_i, for each culturing day $i \in I$, is specified as the following step function similar to the one given in Figure 3:

$$l_i = \begin{cases} f(1) & \text{if } 1 \leq i < i_1, \\ \lfloor f(i_\delta) \rfloor & \text{if } i_\delta \leq i < i_{\delta+1} \text{ for } \delta = 1, 2, 3, \\ f(D) & \text{if } i_4 \leq i \leq D, \end{cases} \tag{5}$$

where $f(i)$ for $i \in [1, D]$ is an increasing function defined as

$$f(i) = (l_D - l_1)\Big(\frac{i-1}{D-1}\Big)^b + l_1, \tag{6}$$

with $b > 0$ being a given *growth parameter*. Among many possible values of b, a cell lot profile generated with $b = 2$ is more similar to the one we encountered in practice so that we used $b = 2$ for the experiments given in sections 5.2 and 5.3. Figure 6 shows an example of a cell lot profile generated with $b = 2$ that shows the resulting l_i values when $i_1 = 12$, $i_2 = 16$, $i_3 = 24$, and $i_4 = 27$. For the sensitivity analysis in section 5.4, instances with various values for b are tested.

To mimic the configuration of a real-world manufacturing facility, we assume that we have tens of incubators, each of which consists of two chambers of type 1 and four chambers of type 2, i.e., $n = 2$ and

Table 1: Headings and their meanings of Table 2

lpval	the optimal value of the LP-relaxation
bestval	the value of the best integer solution obtained at termination
bound	the best upper bound at termination
lpgap	(lpval - bestval) × 100 / (lpval) %
mipgap	(bound - bestval) × 100 / (bound) %
#node	the number of generated nodes in the branch-and-cut tree
lptime	the total time spent in solving the LP-relaxation in seconds
total	the total time spent in solving the formulation in seconds
#opt	the number of instances for which an optimal solution was found out of 10 instances

$R := \{1,2\}$, and the sum of capacities of all chambers of an incubator ($2c_1+4c_2$) is equal to the number of cell lots on the last culturing day (l_D). We also assume that there are no on-going culture processes started before the first day of the planning horizon, and N incubators are available at all times. Hence, all the chambers are available for the entire planning horizon for our computational experiments, i.e., $a_{1t} := 2N$ and $a_{2t} := 4N$ for all $t \in T$. Instances with $N = 20$ and 40 are generated and tested. Let us call a combination of c_1 and c_2 a *capacity configuration*, and denote it as (c_1, c_2) hereafter. For the experiments given in sections 5.2 and 5.3, we set $(c_1, c_2) = (24, 18)$, which is more similar to that in practice. For the sensitivity analysis in section 5.4, we also tested for other capacity configurations.

To test the computational performance of P_{DM} in section 5.2, we implemented a code in C# language using the branch-and-cut routine provided by Xpress 8.9 (Xpress, 2020) with its default parameter setting. The proposed LP-based heuristic algorithm was also implemented in C# language using the LP routine provided by Xpress 8.9. Finally, all the computational experiments were conducted on Intel Core 3.10 GHz PC with 16GB RAM.

5.2. *Performance of the integer optimization model*

We first tested the computational performance of P_{DM} with the test instances generated as follows. First, 10 cell lot profiles are generated according to the method described in section 5.1 with $b = 2$. As mentioned in section 5.1, three values (τ = 90, 180, and 360 days) for the planning horizon, two values for the number of incubators (N = 20 and 40), and one capacity configuration $(24, 18)$ were considered. Then, each combination of a cell lot profile, τ, N, and a capacity configuration corresponds to one test instance, so that 60 instances are generated and tested.

For each of those generated instances, we can find a trivial feasible solution, which we call the *naive* solution. As described in section 5.1, six chambers (2 chambers of type 1 and 4 chambers of type 2) of each incubator can accommodate l_D cell lots at the same time. Hence, if we use each incubator exclusively for one unit for the entire culturing duration, we have a feasible solution with the objective value of $\lfloor \frac{\tau}{D} \rfloor \times N$. In our computational tests, we used the objective value of the naive solution as the initial lower bound of the branch-and-cut algorithm. For each instance, we also set a time limit of 3,600 seconds. The results are reported in Table 2 whose headings from the third column are described in Table 1. Each row of the table reports the average values of the performance of P_{DM} over 10 instances for each combination of τ and N.

From the column 'lpgap', we can see that the LP-relaxations of P_{DM} provides strong upper bounds to the extent that the relative gap to the best integer solution obtained at the termination of the branch-and-cut algorithm ranges from 1.0% to 4.6%. It is also shown in the column 'lptime' that the LP relaxation of P_{DM} can be solved very quickly within 0.1 to 0.6 seconds.

Due to strong upper bounds of the LP-relaxation, the branch-and-cut algorithm with P_{DM} shows good performance in solving test instances. From the column 'mipgap' showing the relative gap between the

Table 2: Computational performance of P_{DM}

Test set		Branch-and-cut algorithm (Xpress with default settings)						Time		
τ	N	lpval	bestval	bound	lpgap	mipgap	#node	lptime	total	#opt
90	20	105.2	103.4	103.8	1.7	0.4	302,165.0	0.1	1,119.0	7
	40	210.4	208.4	209.0	1.0	0.3	723,248.3	0.1	1,825.3	6
180	20	238.1	230.6	237.3	3.1	2.8	127,313.6	0.2	3,600*	0
	40	476.2	468.0	475.5	1.7	1.6	90,195.9	0.2	3,600*	0
360	20	504.0	480.8	503.4	4.6	4.5	3,547.4	0.6	3,600*	0
	40	1,007.9	984.9	1,007.3	2.3	2.2	1,686.6	0.6	3,600*	0

(*) : No integer optimal solution was found within the given time limit of 3,600 seconds.

best integer solution and the best bound obtained at the termination of the branch-and-cut algorithm, it is observed that optimal or near-optimal solutions were found for all the instances within 1 hour. However, from the column '#opt' representing the number of instances for which an optimal solution was found within an hour, we see that it is still hard to find optimal solutions for test instances with $\tau = 180$ and $\tau = 360$ within the time limit.

5.3. Performance of the LP-based heuristic algorithm

Now, the performance of the proposed LP-based heuristic algorithm with the same 60 instances used in the previous subsection is presented in Tables 3 and 4 where each row of the tables reports the average value over corresponding 10 instances. First, the following four options were tested and compared.

- **A1**: BasicLPbasedHeuristic (as described in Algorithm 1)
- **A2**: ImprovedLPbasedHeuristic($A, itrs$) with $A := \{1\}$ and $itrs := \infty$
- **A3**: ImprovedLPbasedHeuristic($A, itrs$) with $A := \{0.1, 0.2, 0.3, 0.4, 0.5\}$ and $itrs := 1$
- **A4**: ImprovedLPbasedHeuristic($A, itrs$) with $A := \{0.1, 0.2, 0.3, 0.4, 0.5\}$ and $itrs := \infty$

The results are shown in Table 3. In the table, the headings 'obj', 'time', 'iter', and '%imp' refer to the objective value of the obtained heuristic solution, the computation time in seconds, the number of times the outer loop (lines 5 - 20 of Algorithm 2 is repeated, and the percentage improvement of the objective value in comparison with that obtained by **A1**, respectively. The results demonstrate that the two improvement strategies greatly improve the performance of the basic LP-based heuristic algorithm. In particular, the impact of the introduction of the greed coefficient α (cf., section 4.2.1), reported in the column 'A3', is greater than that of repeated applications of LP-based heuristic (cf., section 4.2.2), reported in the column 'A2'. Note also that the performance improvement over the basic LP-based heuristic is greatest when both of the two improvement strategies are used together as shown in the column 'A4'. However, since the difference of the objective value between **A3** and **A4** is very small, it is advisable to use **A3** in practice that can find heuristic solutions in a relatively shorter time.

The results in Table 3 show that the LP-based heuristic algorithms find feasible solutions very quickly (within about 10 seconds for instances with $\tau = 360$). Moreover, Table 4 shows that the improved LP-based heuristic algorithm not only has the advantages of short computation time, but also shows very good solution quality. Note that the heuristic solutions reported in the table were obtained by applying **A3**. In the table, Z^{LP} and Z^{BEST} denote the values from the columns 'lpval' and 'bestval' in Table 2, respectively. Z^{RA}

Table 3: Performance improvement of LP-based heuristic algorithms

Instance		A1		A2				A3			A4			
τ	N	obj	time	iter	obj	%imp	time	obj	%imp	time	iter	obj	%imp	time
90	20	93.1	0.3	2.7	94.3	1.29	0.4	100.9	8.38	0.4	2.0	100.9	8.38	0.5
	40	197.2	0.3	2.5	198.4	0.61	0.4	205.8	4.36	0.4	2.1	205.9	4.41	0.5
180	20	191.6	1.1	4.4	205.6	7.31	1.5	225.6	17.75	1.7	2.1	225.7	17.80	2.3
	40	425.9	1.1	3.5	435.6	2.28	1.4	463.3	8.78	1.7	2.0	463.3	8.78	2.3
360	20	387.2	3.5	4.4	423.6	9.40	4.6	473.5	22.29	6.8	2.1	474.2	22.47	9.7
	40	886.7	3.5	3.3	911.1	2.75	4.3	976.8	10.16	6.9	2.2	977.3	10.22	10.3

Table 4: Solution quality of the LP-based heuristic algorithm (A3)

Instance		Solution quality			
τ	N	Z^{RA}/Z^{LP}	Z^{AG}/Z^{LP}	Z^{H}/Z^{LP}	Z^{H}/Z^{BEST}
90	20	72.4%	23.5%	95.9%	97.6%
	40	85.8%	12.0%	97.8%	98.8%
180	20	59.6%	35.1%	94.8%	97.8%
	40	78.6%	18.7%	97.3%	99.0%
360	20	52.2%	41.7%	94.0%	98.5%
	40	75.1%	21.8%	96.9%	99.2%

and Z^{AG} represent the objective values obtained from RoundAdjust(λ^{LP}, Q) procedure and ControlledAugment($\hat{\mathbf{a}}, Q, \alpha$) procedure, respectively. Recall that the final heuristic solution is computed as the sum of the solutions obtained from these two procedures, i.e., $Z^H = Z^{RA} + Z^{AG}$ (the column 'obj' of 'A3' in Table 3). In the table, the columns 'Z^H/Z^{LP}' and 'Z^H/Z^{BEST}' show the percentage of the optimal objective value of the solutions obtained by the improved LP-based heuristic algorithm relative to the optimal objective value of the LP-relaxation of P_{DM} and that relative to the objective value of the best integer solutions obtained within 1 hour by solving P_{DM}, respectively.

Since the optimal solutions are not known for most of the test instances, we compared the quality of heuristic solutions based on the optimal objective value of the LP-relaxation of P_{DM} and the best integer solutions obtained by solving P_{DM}. When compared to the optimal objective value of the LP-relaxation of P_{DM} which is greater than the optimal objective value, the heuristic solutions show a quality of approximately 94.0 to 97.8%. We can also see that our heuristic solutions are comparable (approximately 97.6 to 99.2%) to those best integer solutions obtained by solving P_{DM} for 1 hour, which implies that our heuristic algorithm can be successfully used in practice to find good quality solutions quickly. In addition, the columns 'Z^{RA}/Z^{LP}' and 'Z^{AG}/Z^{LP}' show the qualities of solutions obtained from RoundAdjust(λ^{LP}, Q) procedure and ControlledAugment($\hat{\mathbf{a}}, Q, \alpha$) procedure, respectively, compared to the optimal objective value of the LP-relaxation of P_{DM}. The results show that the quality of Z^{RA} declines with the increase in the length of the planning horizon τ and the decrease in the number of incubators N. However, it can be seen that the worse the performance of RoundAdjust(λ^{LP}, Q) procedure, the greater the improvement of the solution by ControlledAugment($\hat{\mathbf{a}}, Q, \alpha$) procedure.

5.4. Sensitivity analysis

We analyze the sensitivity of the performance of the proposed LP-based heuristic along with the strength of the upper bound provided by the LP-relaxation of P_{DM} to the change of the growth parameter b and the capacity configuration.

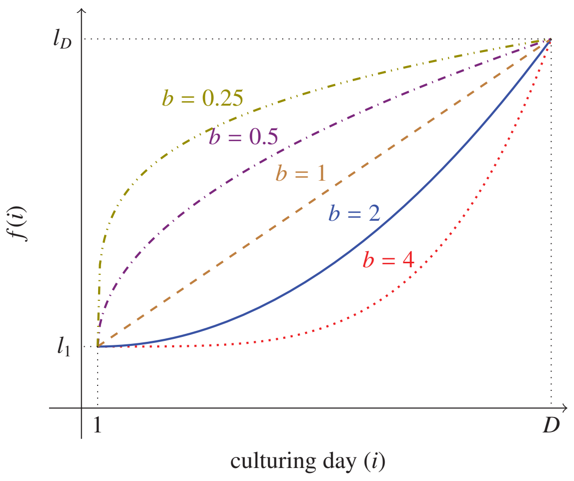


Figure 7: Illustration of function $f(i)$ for different b values

In the previous computational experiments, the cell lot profile of each instance was generated according to functions (5) and (6) with $b = 2$, and the capacity configuration (c_1, c_2) of each instance was set as $(24, 18)$. To analyze the impact of the growth rate of cell lots, we generated 10 cell lot profiles according to functions (5) and (6) for each $b \in \{0.25, 0.5, 1, 2, 4\}$. Figure 7 illustrates the function $f(i)$ defined as (6) for a number of different b values. Note that 10 cell lot profiles with $b = 2$ are the same profiles as those used in the previous experiments. Next, to evaluate the impact of the capacity configuration, we tested 9 capacity configurations (c_1, c_2) from a configuration with $c_1 = c_2$, $(20, 20)$, to one with $c_1 = 3c_2$, $(36, 12)$, as depicted in Figures 8 and 9. As mentioned in section 5.1, we assume that one incubator consists of 2 chambers of type 1 and 4 chambers of type 2, and $2c_1 + 4c_2 = 120$. The first capacity configuration, $(20, 20)$, is actually the case where there is only one type of chambers. For the tests, we set $\tau = 180$ and $N = 40$. For each combination of b and a capacity configuration, 10 instances are tested using ImprovedLPbasedHeuristic$(A, itrs)$ with $A := \{0.1, 0.2, 0.3, 0.4, 0.5\}$ and $itrs := 1$.

Figure 8 shows how the quality of heuristic solution changes according to the change of b and (c_1, c_2). The solution quality is measured by Z^H/Z^{LP}, and the average value over 10 instances for each combination of b and (c_1, c_2) is reported. We observe that the proposed heuristic algorithm is capable of finding good quality solutions for most of the tested combinations. Note that since the unknown optimal values are less than Z^{LP}, if the heuristic solutions are compared with the actual optimal solutions, the percentage values reported in the figure are likely to be better. It also shows that the upper bounds, Z^{LP}, are tight. Except for the cases of $(c_1, c_2) = (36, 12)$ with $b = 1$ and $b = 0.5$ (94.9% and 95.2%, respectively), the average solution quality was over 96%. The results show that as c_1/c_2 increases, the solution quality tends to slightly decrease. However, it can be said that the performance of our algorithm is quite robust to the choice of b and (c_1, c_2) with respect to the solution quality. Through a closer examination of this trend, we were able to observe that the loss of a daily mode increases and the number of daily modes decreases as c_1/c_2 increases. Moreover, the number of daily modes is relatively smaller especially when c_1 is an integer multiple of c_2, e.g., (20,20), (30,15), and (36,12). Since the LP-relaxation of P_{DM} basically will try to (fractionally) use daily modes to minimize losses in those cases, the gap between Z^H which is constructed with a relatively small number of daily modes with relatively larger losses and Z^{LP} tends to increase. However, it might be the case that Z^{LP} is relatively less tight compared to the objective value of unknown optimal solution when c_1/c_2 is large and c_1 is an integer multiple of c_2.

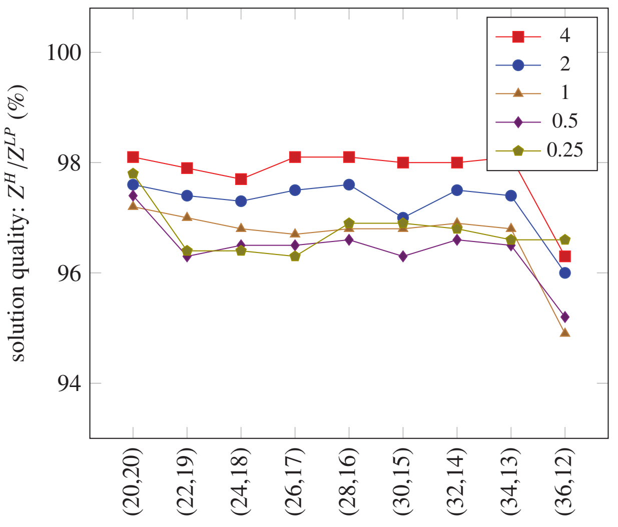


Figure 8: Quality of heuristic solution compared to the LP-relaxation of P_{DM}

The amount of computation time to solve the LP-relaxation of P_{DM} and that taken by the proposed LP-based heuristic algorithm are depicted in Figure 9, where the average value over 10 instances for each combination of b and (c_1, c_2) is reported. First, it can be seen that the proposed heuristic algorithm runs very fast to the extent that it solves the tested instances for each combination of b and (c_1, c_2) within about 2 seconds. It can be said that the performance of our algorithm is also quite robust to the choice of b and (c_1, c_2) with respect to the computation time. From the figure, we also see that the LP solution time and the heuristic solution time show similar trends since the heuristic algorithm is based on solving the LP-relaxation of P_{DM}. We observed that, as b decreases, the number of daily modes increases, and thus the number of decision variables of P_{DM} increases, so that the LP solution time and the heuristic solution time tend to increase. Also, the capacity configuration, (c_1, c_2) greatly influences the solution time. Specifically, as c_1/c_2 increases, the number of daily modes decreases as mentioned before. This reduces the number of decision variables of P_{DM}, and thus the LP solution time and the heuristic solution time decreases. Notice also that, the LP-relaxation of P_{DM} and our heuristic algorithm run much faster when c_1 is an integer multiple of c_2 because in this case the number of daily modes are relatively smaller.

6. Concluding Remarks

In this paper, we presented a novel scheduling problem, the stem cell culturing problem (SCP), which arises in the emerging stem cell therapy industry. We formally defined the SCP, analyzed its characteristics, and showed that the problem is theoretically strongly NP-hard in general. Based on the concept of a daily mode, an integer programming model (P_{DM}) was formulated and computationally tested. Computational experiments showed that the LP-relaxation of P_{DM} gives a strong upper bound on the optimal objective value of the SCP to the extent that the average optimality gap (relative to the best integer solution found with P_{DM}) of the LP-relaxation of P_{DM} over all the 60 test instances was 2.40% (cf., ‘lpgap’ in Table 2). With the help of strong upper bounds provided by the LP-relaxation of P_{DM}, the branch-and-cut algorithm of a state-of-the-art optimization solver was able to find optimal or near-optimal solutions for all 60 test instances within 1 hour. The average optimality gap (relative to the upper bound at termination) of the best integer solution at termination over all the 60 test instances was 1.97% (cf., ‘mipgap’ in Table 2). To

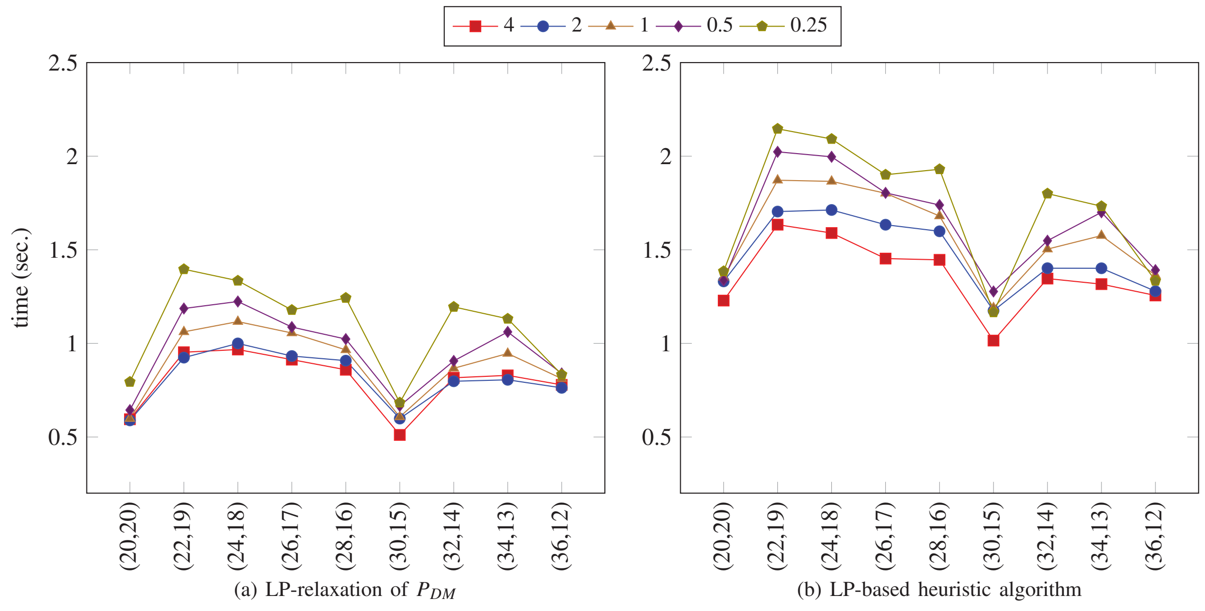


Figure 9: Computation time for each combination of growth parameter b and a capcity configuration (c_1, c_2)

find provably good solutions quickly, we proposed an LP-based heuristic algorithm which utilizes optimal solutions to the LP-relaxations of P_{DM} and iterative augmentation. To mitigate the greedy nature of an initially devised basic LP-based heuristic algorithm, two improvement strategies were proposed, based on which an improved LP-based heuristic algorithm was finally proposed. The improved LP-based heuristic algorithm showed good computational performance. It could find quality solutions within a few seconds (cf., Table 3). The average quality of solutions over all the 60 test instances relative to the LP-relaxation of P_{DM} was 96.11% and that relative to the best integer solutions found with the branch-and-cut algorithm with P_{DM} was 98.48% (cf., Table 4). Moreover, the computational performance of the proposed LP-based heuristic algorithm and the LP-relaxation of P_{DM} was robust to the change of the growth parameter of the cell lot profile and the capacity configuration. Therefore, the proposed integer optimization model P_{DM} and the improved LP-based heuristic can be readily used in practical situations.

The results of our study can be utilized in a number of ways. First, the proposed integer optimization model and the improved LP-based heuristic algorithm can be used for the primary purpose of this study, i.e., planning stem cell culture schedules for production of the stem cell therapeutic product in a manufacturing facility in practice. In an existing practice we have experienced, a manager manually determined a schedule and updated it periodically with the help of spreadsheet software. From a comparative evaluation, we have found that the productivity of a manufacturing facility could be improved up to 20% compared to the existing schedules. For example, for a manufacturing facility consisting of 20 incubators, schedules obtained by our approach accommodate an average of 48 units per month, whereas existing schedules culture an average of 40 units per month. Second, the proposed model and algorithm can be used in making investment decisions of a manufacturing facility by simulating the productivity according to the number of incubators, the capacity configuration, and so on. Since the autologous stem cell therapy industry is at its early stage, new therapeutic products will be developed, and the characteristics of equipment and the operational policies will also evolve. For example, multiple types of therapeutic products with different characteristics could be produced in one manufacturing facility in the near future. In this situation, the stem cell culture schedules need to be established and updated considering the product mix reflecting product demand. In the presence of those possible changes, our results can also serve as a step stone to cope with additional considerations.

Finally, the following topics can be further studied in the near future. First, from a theoretical point of view, to devise methods including strong cutting planes to further strengthen the LP-relaxation of P_{DM} is one way to improve the computational performance of P_{DM}. Second, optimization models and efficient algorithms to plan a culture schedule which is *robust* to the unexpected unavailability of resources (e.g., unscheduled breakdown of incubators) can be further studied. The proposed methods in this study can be used in a reactive manner to re-plan schedules in those situations. However, robust schedules taking into account uncertain events can be helpful for the efficient proactive operation in practice.

Acknowledgement

The authors would like to thank the editors and anonymous reviewers for their valuable comments. The authors would like to thank the Institute for Industrial Systems Innovation of Seoul National University for the administrative support. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (No. 2018R1A2B2003227).

Disclosure statements

No potential conflict of interest was reported by the authors.

References

van den Akker, J. M., Hoogeveen, J. A., & van de Velde, S. L. (1999). Parallel machine scheduling by column generation. *Operations Research*, *47*, 862–872. doi:10.1287/opre.47.6.862.

Arbib, C., & Marinelli, F. (2014). On cutting stock with due dates. *Omega*, *46*, 11–20. doi:10.1016/j.omega.2014.01.004.

Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P., & Vance, P. H. (1998). Branch-and-price: column generation for solving huge integer programs. *Operations Research*, *46*, 316–329. doi:10.1287/opre.46.3.316.

Bartusch, M., Möhring, R. H., & Radermacher, F. J. (1988). Scheduling project networks with resource constraints and time windows. *Annals of Operations Research*, *16*, 199–240. doi:10.1007/BF02283745.

Biehl, J. K., & Russell, B. (2009). Introduction to stem cell therapy. *Journal of Cardiovascular Nursing*, *24*, 98–103. doi:10.1097/JCN.0b013e318197a6a5.

Blazewicz, J., Lenstra, J. K., & Kan, A. H. G. R. (1983). Scheduling subject to resource constraints: classification and complexity. *Discrete Applied Mathematics*, *5*, 11–24. doi:10.1016/0166-218X(83)90012-4.

Breugem, T., van Rossum, B. T., Dollevoet, T., & Huisman, D. (2022). A column generation approach for the integrated crew re-planning problem. *Omega*, *107*, 102555. doi:10.1016/j.omega.2021.102555.

Brucker, P., Drexl, A., Möhring, R. H., Neumann, K., & Pesch, E. (1999). Resource-constrained project scheduling: notation, classification, models, and methods. *European Journal of Operational Research*, *112*, 3–41. doi:10.1016/S0377-2217(98)00204-5.

Cacchiani, V., & Salazar-González, J.-J. (2020). Heuristic approaches for flight retiming in an integrated airline scheduling problem of a regional carrier. *Omega*, *91*, 102028. doi:10.1016/j.omega.2019.01.006.

Cavalcante, C. C. B., de Souza, C. C., Savelsbergh, M. W. P., Wang, Y., & Wolsey, L. A. (2001). Scheduling projects with labor constraints. *Discrete Applied Mathematics*, *112*, 27–52. doi:10.1016/S0166-218X(00)00308-5.

Chen, J., & Askin, R. G. (2009). Project selection, scheduling and resource allocation with time dependent returns. *European Journal of Operational Research*, *193*, 23–34. doi:10.1016/j.ejor.2007.10.040.

Corestem (2018). ALS (NeuroNata-R$^{®}$). URL: http://www.corestem.com/en/m21.php accessed: 2021-02-25.

Daley, G. Q., & Scadden, D. T. (2008). Prospects for stem cell-based therapy. *Cell*, *132*, 544–548. doi:10.1016/j.cell.2008.02.009.

Desaulniers, G., Desrosiers, J., & Solomon, M. M. (Eds.) (2005). *Column Generation.* Springer. doi:10.1007/b135457.

Desrochers, M., Desrosiers, J., & Solomon, M. M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, *40*, 342–354. doi:10.1287/opre.40.2.342.

Drezet, L., & Billaut, J. (2008). A project scheduling problem with labour constraints and time-dependent activities requirements. *International Journal of Production Economics*, *112*, 217–225. doi:10.1016/j.ijpe.2006.08.021.

Gao, Y., Schmidt, M., Yang, L., & Gao, Z. (2020). A branch-and-price approach for trip sequence planning of high-speed train units. *Omega*, *92*, 102150. doi:10.1016/j.omega.2019.102150.

Garey, M. R., & Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company.

Gilmore, P. C., & Gomory, R. E. (1961). A linear programming approach to the cutting-stock problem. *Operations Research*, *9*, 849–859. doi:10.1287/opre.9.6.849.

Hartmann, S. (2001). Project scheduling with multiple modes: a genetic algorithm. *Annals of Operations Research*, *102*, 111–135. doi:10.1023/A:1010902015091.

Hartmann, S. (2013). Project scheduling with resource capacities and requests varying with time: a case study. *Flexible Services and Manufacturing Journal*, *25*, 74–93. doi:10.1007/s10696-012-9141-8.

Hartmann, S. (2015). Time-varying resource requirements and capacities. In Schwindt, C., & Zimmermann, J. (Eds.), *Handbook on Project Management and Scheduling Vol.1* (pp. 163–176). Springer. doi:10.1007/978-3-319-05443-8_8.

Hartmann, S., & Briskorn, D. (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, *207*, 1–14. doi:10.1016/j.ejor.2009.11.005.

Hartmann, S., & Briskorn, D. (2022). An updated survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, *297*, 1–14. doi:10.1016/j.ejor.2021.05.004.

Hartmann, S., & Drexl, A. (1998). Project scheduling with multiple modes: a comparison of exact algorithms. *Networks*, *32*, 283–297. doi:10.1002/(SICI)1097-0037(199812)32:4<283::AID-NET5>3.0.CO;2-I.

Herroelen, W. (2005). Project scheduling—theory and practice. *Production and Operations Management*, *14*, 413–432. doi:10.1111/j.1937-5956.2005.tb00230.x.

Herroelen, W., De Reyck, B., & Demeulemeester, E. (1998). Resource-constrained project scheduling: a survey of recent developments. *Computers & Operations Research*, *25*, 279–302. doi:10.1016/S0305-0548(97)00055-5.

Hipp, J., & Atala, A. (2008). Sources of stem cells for regenerative medicine. *Stem Cell Reviews*, *4*, 3–11. doi:10.1007/s12015-008-9010-8.

Kellerer, H., Pferschy, U., & Pisinger, D. (2004). *Knapsack Problems*. Springer. doi:10.1007/978-3-540-24777-7.

Kolisch, R., & Meyer, K. (2006). Selection and scheduling of pharmaceutical research projects. In Józefowska, J., & Weglarz, J. (Eds.), *Perspectives in Modern Project Scheduling* (pp. 321–344). Springer. doi:10.1007/978-0-387-33768-5_13.

Kolisch, R., & Padman, R. (2001). An integrated survey of deterministic project scheduling. *Omega*, *29*, 249–272. doi:10.1016/S0305-0483(00)00046-3.

Li, C., Gong, L., Luo, Z., & Lim, A. (2019). A branch-and-price-and-cut algorithm for a pickup and delivery problem in retailing. *Omega*, *89*, 71–91. doi:10.1016/j.omega.2018.09.014.

Li, H., & Womer, K. (2009). Scheduling projects with multi-skilled personnel by a hybrid MILP/CP benders decomposition algorithm. *Journal of Scheduling*, *12*, 281–298. doi:10.1007/s10951-008-0079-3.

Maniezzo, V., & Mingozzi, A. (1999). The project scheduling problem with irregular starting time costs. *Operations Research Letters*, *25*, 175–182. doi:10.1016/S0167-6377(99)00050-4.

Mika, M., Waligóra, G., & Węglarz, J. (2015). Overview and state of the art. In Schwindt, C., & Zimmermann, J. (Eds.), *Handbook on Project Management and Scheduling Vol.1* (pp. 445–490). Springer. doi:10.1007/978-3-319-05443-8_21.

Möhring, R. H., & Uetz, M. (2003). Scheduling scarce resources in chemical engineering. In Jäger, W., & Krebs, H.-J. (Eds.), *Mathematics - Key Technology for the Future* (pp. 637–650). Springer. doi:10.1007/978-3-642-55753-8_49.

National Institutes of Health (2016). Introduction to stem cells. URL: https://stemcells.nih.gov/info/basics accessed: 2021-08-01.

Parmentier, A., & Meunier, F. (2020). Aircraft routing and crew pairing: updated algorithms at air france. *Omega*, *93*, 102073. doi:10.1016/j.omega.2019.05.009.

Pritsker, A. A. B., Waiters, L. J., & Wolfe, P. M. (1969). Multiproject scheduling with limited resources: a zero-one programming approach. *Management Science*, *16*, 93–108. doi:10.1287/mnsc.16.1.93.

Schwindt, C., & Zimmermann, J. (Eds.) (2015). *Handbook on Project Management and Scheduling Vol.1*. Springer. doi:10.1007/978-3-319-05443-8.

Shariatmadari, M., Nahavandi, N., Zegordi, S. H., & Sobhiyah, M. H. (2017). Integrated resource management for simultaneous project selection and scheduling. *Computers & Industrial Engineering*, *109*, 39–47. doi:10.1016/j.cie.2017.04.003.

Syed, B. A., & Evans, J. B. (2013). Stem cell therapy market. *Nature Reviews Drug Discovery*, *12*, 185–186. doi:10.1038/nrd3953.

Talbot, F. B. (1982). Resource-constrained project scheduling with time-resource tradeoffs: the nonpreemptive case. *Management*

Science, *28*, 1197–1210. doi:10.1287/mnsc.28.10.1197.

Vance, P. H., Barnhart, C., Johnson, E. L., & Nemhauser, G. L. (1997). Airline crew scheduling: a new formulation and decomposition algorithm. *Operations Research*, *45*, 188–200. doi:10.1287/opre.45.2.188.

Vanderbeck, F., & Wolsey, L. A. (2010). Reformulation and decomposition of integer programs. In Jünger, M., Liebling, T. M., Naddef, D., Nemhauser, G. L., Pulleyblank, W. R., Reinelt, G., Rinaldi, G., & Wolsey, L. A. (Eds.), *50 Years of Integer Programming 1958-2008* (pp. 431–502). Springer. doi:10.1007/978-3-540-68279-0_13.

Volarevic, V., Markovic, B. S., Gazdic, M., Volarevic, A., Jovicic, N., Arsenijevic, N., Armstrong, L., Djonov, V., Lako, M., & Stojkovic, M. (2018). Ethical and safety issues of stem cell-based therapy. *International Journal of Medical Sciences*, *15*, 36–45. doi:10.7150/ijms.21666.

Wang, D., Yin, Y., & Cheng, T. (2018). Parallel-machine rescheduling with job unavailability and rejection. *Omega*, *81*, 246–260. doi:10.1016/j.omega.2018.04.008.

Węglarz, J., Józefowska, J., Mika, M., & Waligóra, G. (2011). Project scheduling with finite or infinite number of activity processing modes - a survey. *European Journal of Operational Research*, *208*, 177–205. doi:10.1016/j.ejor.2010.03.037.

Xpress (2020). Xpress 8.9. URL: http://www.fico.com/en.

Zhu, G., Bard, J. F., & Yu, G. (2006). A branch-and-cut procedure for the multimode resource-constrained project-scheduling problem. *INFORMS Journal on Computing*, *18*, 377–390. doi:10.1287/ijoc.1040.0121.