ISSN 1392-3730/eISSN 1822-3605

JOURNAL of CIVIL ENGINEERING and MANAGEMENT

2024
Volume 30
Issue 3
Pages 220–233
https://doi.org/10.3846/jcem.2024.20952

ADAPTIVE LOB SCHEDULING FOR OPTIMIZING RESOURCE LEVELING AND CONSUMPTION

Ayman ALTUWAIM[1✉], Eid ALAGHA[1],
Abdulrahman Bin MAHMOUD[1], Dina A. SAAD[2]

[1] *Department of Civil Engineering, King Saud University, Riyadh 12372, Saudi Arabia*
[2] *Structural Engineering Department, Cairo University, Giza, Egypt*

Article History:
- received 12 September 2023
- accepted 4 December 2023

Abstract. Several techniques were developed to improve resource allocation in construction projects, yet few in repetitive construction projects. Accordingly, this research aims at minimizing resource consumption and fluctuations using Line-of-Balance (LOB) repetitive scheduling technique. An optimization model has been developed consisting of three modules: LOB schedule module, resource leveling module, and optimization module. The model helps determine the optimum (a) start time for each activity considering any needed delays to level resources, (b) number of crews travelling from one unit to another, (c) unit to change the number of crews, and (d) the new changed number of crews. Unlike the existing efforts, the developed model provides the capabilities of (a) allowing the change of number of crews of an activity at a certain repetitive unit to increase or decrease the progress rate; and (b) accommodates non-serial repetitive projects to enhance the model's practicality. Using a pipeline project, the model outperformed the existing LOB-based models in minimizing resource consumption and fluctuations within the desired project duration. This study offers the project planners a useful tool to efficiently utilize their projects' resources and avoid hidden costs due to inefficient resource utilization on-site as well as overcoming shortage in resource availability.

Keywords: repetitive scheduling, resource-constrained scheduling, resource leveling, optimization, resource allocation, line of balance.

✉Corresponding author. E-mail: *aaltuwaim@ksu.edu.sa*

1. Introduction

In construction projects, efficient utilization of resources is a key for a successful completion that meets both time and cost goals. However, construction schedules developed by the commonly used critical path method (CPM) often generate resource fluctuations and impractical consumption that are quite costly to implement (El-Rayes & Jun, 2009). Thus, it is essential to consider concurrently when scheduling both the resource utilization and the logical relationships between activities. Accordingly, several research efforts have tried to address the problem of resource management in construction projects whether using resource-constrained scheduling techniques or resource leveling techniques (García-Nieves et al., 2018).

However, it is even more challenging in repetitive construction projects, including: linear projects like tunnels and roads, vertical projects like high-rise buildings, or scattered projects like housing projects. Since the work is repeated across repetitive units, resources are required to move from one unit to another continuously to avoid being idle and to benefit from the learning curve. Using CPM scheduling for this type of projects leads to disruption of the resources' work flow across the repetitive units, and inability to visually present the crews' progress across the units (Hegazy et al., 2021; Harris & Ioannou, 1998). Accordingly, various scheduling techniques designated for repetitive construction projects were developed to overcome those drawbacks (Ioannou & Yang, 2016).

The most common repetitive scheduling methods (RSM) are the line of balance (LOB) and the linear scheduling method (LSM) (García-Nieves et al., 2018). In repetitive scheduling method (RSM), activities are presented graphically in the form of flow lines that represent the progress rate across the repetitive units. They are constrained with logical relationships and resource work continuity from one unit to another. However, in RSM, activities diverge if an activity with slow production rate follows an activity with higher production rate, and converge if the contrary occurs. Accordingly, to respect the precedence constraints across the repetitive units and to avoid convergence of activities with different production rates, activities are controlled by the latest finish of the first unit in all preceding activities in case of diverging activities, and by the latest finish of the last unit in all preceding activities in case of converging activities (Harris & Ioannou, 1998). Thus, despite the benefit of repetitive scheduling in maintaining the crews' synchronization and the flow of resources uninterrupted from one unit to another; it may increase the project duration and subsequently the indirect costs (Altuwaim & El-Rayes, 2018).

To alleviate these problems, slower activities can be accelerated by utilizing more resources by increasing the crew size to increase the productivity, however, the crew size cannot be increased indefinitely. Therefore, multiple crews travelling across the units can be used to shorten the activity's duration across the units (Saad et al., 2021; Hegazy & Wassef, 2001). However, assigning crews to the activities randomly can lead to huge resource fluctuations and inefficient resource consumption. Consequently, it leads to increased project total cost and the requirement of providing a high level of resource availability. Accordingly, this research aims to address these limitations by considering the resource utilization profile when scheduling repetitive construction projects using LOB, and identifying the optimum number of travelling crews across the units to arrive at the desired project duration with efficient resources management, as described later in the research objective section. The study outcomes will help construction planners reduce the total cost of their repetitive construction projects by improving their resource management practice.

2. Literature review

In the literature of repetitive scheduling, several research efforts tried to address the problem of reducing the project duration while maintaining resource continuity. Others tackled the common scheduling problem of time cost trade-off (TCT), by trying to find the optimum balance between the reduced duration and the associated increased costs. For example, Hegazy et al. (2020) introduced an enhancement to LOB scheduling to reduce project duration considering time deadline, limited resources, and the existence of non-identical units. New formulations have been proposed to compute the necessary number of crews and the interruption time needed to synchronize the activities' production rates to generate densely packed schedules. Altuwaim and El-Rayes (2018) developed an optimization model that minimizes the costs associated with the forced interruptions to minimize delay caused by the converging activities, considering the delay of each repetitive activity from its earliest start time. Bakry et al. (2014) developed a repetitive schedule optimization model that determines the optimum acceleration strategy (e.g., working double shifts, increasing number of resources) for each activity in each repetitive unit as a separate segment to reduce the overall project duration using the minimum cost slope. Hegazy and Kamarah (2008) developed an optimization model to reduce the project duration by incorporating acceleration strategies, including: increasing working time, increasing number of resources, relaxing converging activities, and introducing work interruptions. Ipsilandis (2007) developed a repetitive schedule supported with a multi-objective optimization model to minimize: project duration, work-break time, cost of work break due to resource idleness, cost of delayed completion of units as a result of delayed payments, and the tradeoff between the costs of project delays and resource delays. El-Rayes and Moselhi (2001) developed an optimization model to determine the optimum crew size and interruption strategy to minimize project duration in repetitive scheduling. Despite the usefulness of the above time-driven techniques, however, they do not provide practical solutions to the problems associated with the resource consumption and limitations.

2.1. Resource-constrained repetitive scheduling

Resource-constrained scheduling, which is often referred to it as resource allocation scheduling technique, addresses the problem of resource limitation (García-Nieves et al., 2019). It aims at minimizing the project duration while taking into consideration the resource availability limits. Some research efforts developed resource-constrained repetitive scheduling models using different approaches. For instance, Leu and Hwang (2001) developed a GA-based repetitive scheduling optimization model to arrive at the optimum trade-off between the resource availability and project duration, specifically for precast production. The model considers the resource limitations in terms of crew size and resource sharing of rare resources like cranes. Hyari and El-Rayes (2006) developed a multi-objective optimization model to arrive at the optimum repetitive schedule considering trade-off between project duration and crew continuity. To provide practical schedules, the model incorporates the crews' formation and availability in addition to the job logical conditions. García-Nieves et al. (2019) developed a repetitive scheduling optimization model that minimizes the project make-span or the project cost, considering all four relationships, optional continuity between sub activities, multiple execution modes to allow acceleration and deceleration of activities, controlled maximum shifts, the flexibility of utilizing multiple crews, and resource consumption constraint per period.

2.2. Resource leveling

As opposed to resource allocation techniques, resource leveling is concerned with the resource consumption (García-Nieves et al., 2019). Its objective is to minimize the fluctuations, the peaks and valleys, in the resource demand curves without changing the project duration to reduce resource logistics and costs (Hegazy, 1999; Damci et al., 2013a; Tang et al., 2014). Several research efforts have tackled resource leveling in CPM analysis by developing optimization models to minimize the resource fluctuations and consumption to meet the imposed resource constraints (El-Rayes & Jun, 2009; Hegazy, 1999). For instance, Senouci and Eldin (2004) developed a GA-based optimization model that addresses resource allocation and leveling simultaneously in scheduling, considering all logical relationships, multiple crew strategies, total project cost minimization, and time-cost trade-off. In the model, the absolute difference between resource consumption in consecutive time periods is constrained to meet a preset desired limit to level resource consumption. Christodoulou et al. (2010) developed a CPM-based resource leveling algorithm based on the concept of entropy maximization rather than the commonly used method of minimum moment algorithm. The developed technique takes into consideration as well the avaibility of resources, and the possibility of stretching or pressing the activities' duration. Jun and El-Rayes (2011) developed a multi-objective optimization model using genetic algorithms that arrives at the optimum schedule that minimizes project duration and the resource release and rehire or resource idle days, considering resource availability. In this model, two metrics that rely on the maximum resource consumption and the difference between consecutive daily consumption were used to reduce the resource fluctuations. Koulinas and Anagnostopoulos (2012) developed a hyperheurisitc algorithm that uses multi-level heuristic or metaheuristic algorithms to address both resource leveling and allocation simultaneously. It minimizes the moment and the project duration under constrained resource availability. El-Abbasy et al. (2016) developed a multi-objective optimization model that consists of three sub modules. Among those modules, one that addresses resource leveling by computing two metrics; Release and Re-Hire (RRH) and Resource Idle Days (RID). The first metric is used to minimize the resource demand peaks and valleys, while the second metric is to reduce the resource idle times. Also, the Abdel-Basset et al. (2020) developed a neutrosophic heuristic procedure for resource leveling, however, it relies on shifting the activities within their float to minimize the sum of squares of resources usage, and thus minimize resource fluctuations.

It can be noted that most of the existing efforts have tried leveling the resources using the following algorithms or its enhancements: 1) the minimum moment algorithm or squared sum of resources, 2) the minimum absolute difference between resource consumption in consecutive time periods, 3) reduction of resource demand or resource idleness, or 4) minimum deviation of the daily resource consumption from the overall average.

However, few research efforts have applied those algorithms to address the problem of resource consumption and fluctuations in repetitive scheduling. Among those efforts, Damci et al. (2013a) developed a genetic algorithm-based multi-resource leveling model to generate an optimum LOB schedule with optimum resource consumption profile. The model considered the multiple crews that can be utilized in a given activity (e.g., excavation crew and the helping independent crew), and the dominancy among the crews. It consists of two modules: scheduling module and leveling module. The schedule module is responsible for producing the schedule and the resource histograms. The leveling module incorporates a genetic algorithm optimization model that arrives at the optimum leveled profile through minimizing the sum of the absolute deviations between each daily resource consumption and the average resource consumption. The decision variable is the number of dominant crews assigned in each eligible activity in each unit. The constraints include the number of crews limit, the logical sequence between the activities, and number of crews following the principal of natural rhythm. Following the same methodology, Damci et al. (2013b) developed a genetic algorithms-based resource-leveling optimization model for LOB scheduling, yet considering only single crews. Tang et al. (2014) developed a LSM supported with a constraint programming optimization model using two stages to arrive at an optimum schedule and optimum resource utilization curve while considering a fixed time constraint. The first stage generates the optimum schedule considering the logical sequence between activities and their productivities. The second stage works on the generated schedule from the first stage to improve the schedule by fixing the controlling segments in the activities and changing the slope of the non-controlling segments by changing the number of crews. To arrive at the optimum leveled resource profile, the model's objective function is set to minimize the deviation of the sum of the absolute values of the differences between the resource consumptions of every 2 consecutive days. The model was able to reduce the resource fluctuations, yet it didn't decrease the resource consumption.

Tang et al. (2018) developed a comprehensive LOB-based optimization model that addresses the three main scheduling problems: resource constraints, time-cost trade-off, and resource leveling. Thus, the model has multiple objective functions, including: minimize project duration, minimize the total cost, minimize the sum of the absolute of the difference between the resources utilized in every two consecutive days considering the case of utilizing multiple resources, minimize the peak resource usage, and minimize the total resource usage. The model allows having different unit production rates among the units, and non-continuous flow of crews between the repetitive units by dividing the activity into sub-activities.

García-Nieves et al. (2018) developed LSM-based optimization model that minimizes the project duration, considering resource constraints. It incorporates logical sequence constraints between sub-activities of the same activity and of interdependent activities to allow activities to be executed discretionally in a fragmented way or in a continuous way. Moreover, it allows multiple modes of execution for each activity in the form of the crew size to help accelerate the model. However, those modes are controlled to prevent any inefficient hiring and firing conditions by using several indices that measures the efficiency of resource consumption and resource levelling of the optimal solutions, including: resource improvement coefficient, the resource idle days, resource surplus-consumption ratio, and resource consumption-availability ratio. Wang et al. (2020) developed a two-staged LSM-based optimization model using particle swarm optimization for resource leveling using a third type of float. In the first stage, an optimum LSM schedule is developed considering activity type, resource usage, duration constraint and logical constraints between activities. In the second stage, the non-controlling activities are optimized by changing the start time and construction rate within the newly defined third type of float to achieve better resource profile in terms of resource fluctuations. The model uses the minimum absolute difference between resource consumption in consecutive time periods as an objective function to minimize resource consumption.

3. Research objective

From the literature review, it can be noted that the earlier efforts that have addressed the resource leveling problem in repetitive construction projects can be classified into two groups: 1) LSM-based repetitive models, and 2) LOB-based repetitive models. It has been noted that the existing research attempts, to optimize the resource leveling in LOB-based schedules, are incapable to guarantee generating the optimal solution. They are suffering from several limitations, including: (1) being limited to finding an optimum leveled resource profile for a fixed input LOB schedule leading to excluding a research space of a huge number of possible optimal solutions (e.g., Damci et al., 2013a; Koulinas & Anagnostopoulos, 2012), (2) using a fixed number of crews across the units and thus limiting the number of possible optimal solutions that can reduce the project duration and resource fluctuations (e.g., Tang et al., 2018); and (3) being limited to serial repetitive construction projects where each activity is assumed to have only one predecessor (e.g., Damci et al., 2013a; Tang et al., 2018). Accordingly, the objective of this research is to address those limitations by developing a LOB-based optimization model that aims at arriving at the optimum schedule that minimizes resource consumption and fluctuations, while satisfying the desired project duration. As opposed to the existing LOB models, the proposed model allows changing the number of crews across the repetitive units, and subsequently the activity's progress rate to facilitate finding the optimum solution. Additionally, the proposed model accommodates non-serial repetitive projects; such that, it allows each activity to have one or more predecessors in order to enhance the model's practicality.

4. Proposed LOB-RL Optimization Model

This research proposes an enhancement to the earlier LOB models developed by Damci et al. (2013a, 2013b) and Tang et al. (2018). As opposed to the earlier efforts, the proposed LOB-RL optimization model has a built-in resource leveling (RL) feature that can generate a LOB schedule that meets a desired project duration with minimum resource consumption and fluctuations, yet with the flexibility of changing the number of crews across the repetitive units. The model consists of three modules, as shown in Figure 1: 1) a scheduling module that generates an initial LOB schedule given the project data (activities, logical sequence, crews productivity rates, etc.), 2) a resource leveling module that computes the daily resource usage and fluctuations in addition to generating a resource profile, and 3) an optimization module that helps determine the optimum schedule and resource profile in an iterative process considering the input from the first two modules. Accordingly, the output of the model is an optimum LOB schedule that meets the required project duration with minimum resource consumption and fluctuations. The following sections explain each module in detail and the application of the model to a case study.

4.1. LOB schedule module

To develop a LOB schedule for any given repetitive project, it is necessary to determine the progress rate of the crews travelling from one unit to another in each activity. Therefore, the unit progress rate relies on the number of crews assigned to each activity travelling across the units, and the activity's duration to finish one repetitive unit, as shown in Eqn (1). The duration in turn relies on the crews' productivity rate, and the quantity of work or the required working hours to finish a given activity, as shown in Eqn (2).

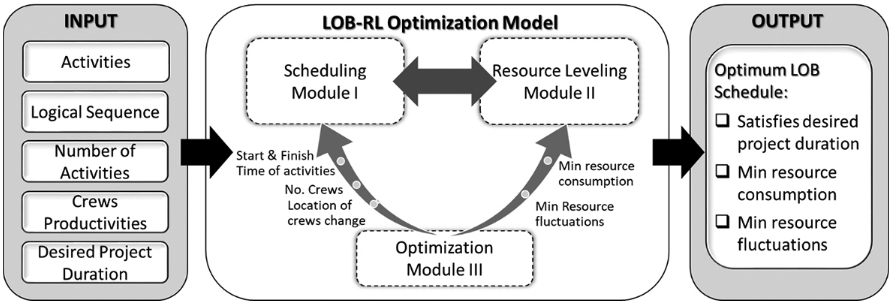


Figure 1. Research methodology framework

Figure 2 shows an example, on the left hand side, for a LOB developed for a repetitive activity “A” progressing through discrete repetitive units using single crew (CR-1). The flow line connecting the activity in each unit represents the unit progress rate (*upr*). It should be noted that in this research, only the solid flow line (bold line in Figure 2) is used in developing the schedules to facilitate viewing the schedule.

$$upr_{ij} = \frac{1}{D_{ij}} \times Cn_i \; ; \; \forall i \in \{1,\ldots,m\}, \forall j \in \{1,\ldots,N\} \; ; \tag{1}$$

$$D_{ij} = {Q_{ij}}\big/{cpr_{ij}} \; ; \; \forall i \in \{1,\ldots,m\}, \forall j \in \{1,\ldots,N\} \, , \tag{2}$$

where, upr_{ij} is the unit progress rate of each activity i (i = 1 to m activities) in each unit j (j = 1 to N activities), D_{ij} is the duration of activity (i) in unit (j), Cn_i is the number of crews travelling from one unit to another, cpr_{ij} is the crew production rate of activity (i) at unit (j) which relies on the crew size, Q_{ij} is the quantity of work of activity (i) at unit (j) which is constant across the units for identical repetitive units, yet it is flexible to accommodate any irregular or non-identical units.

In this research, the schedule module allows changing (increasing or decreasing) the number of crews working across the repetitive units to improve the schedule (28), and overcome any delays due to leveling resources. Yet, using only one crew in a given repetitive unit. Currently, the existing model enable one activity to have two different unit progress rates by increasing the number of crews to a new number (Cn_i^*) at a given unit (U_i) to minimize delays when leveling resources. For instance, on the right hand side of Figure 2, the unit progress rate has increased after the completion of unit 2 (i.e., U_i = 2) due to adding a second crew (CR-2) to work concurrently with the first crew (CR-1). Accordingly, the unit progress rate formulated in Eqn (1) will be modified to adapt the change in the number of crews as follows:

$$upr_{ij} = \begin{cases} \dfrac{1}{D_{ij}} \times Cn_i, & j \le U_i \\ \\ \dfrac{1}{D_{ij}} \times Cn_i^*, & j > U_i \end{cases} \; . \tag{3}$$

Usually to develop a LOB schedule, there is a need to identify the cases where the activities are diverging or converging to avoid logical conflicts between activities when computing the start and finish times (Harris & Ioannou, 1998). An activity would be diverging away from its logical predecessors, if the activity’s unit progress rate is lower than the rate of its predecessors. In this case, the activity’s start time in the first repetitive unit, considering the preceding activities, is computed from the finish time of the predecessor activity in the first unit in addition to any necessary time buffer. On the other hand, an activity would be converging towards its predecessors, if the activity’s unit progress rate is higher than the rate of the preceding activities. In this case, the activity’s start time in the last repetitive unit is constrained by the finish time of the predecessor in the last repetitive unit.

However, since in the proposed schedule module, the activity can have different number of crews across the units, and thus different unit progress rates, it would be quite complex to determine whether the activity is diverging away or converging towards its predecessor. Therefore, this research adopted a variation of the shift method previously utilised using different variations in the literature (Hegazy et al., 2020; Hegazy & Kamarah, 2008; Long & Ohsato, 2009).

In this method, the activity’s start time in the first repetitive unit is constrained with the finish time of the predecessor in the first unit in addition to any necessary time buffer. To facilitate identifying the predecessors of each activity in the formulations, they have been ranked from k = 1 to K. Accordingly, the predecessor (p) of activity (i) that falls in a given rank (k) is defined as ($p_{i,k}$). The activity’s start time ($St_{i,1}^{k}$) is first computed with respect to each ranked predecessor (k) using Eqn (4). Afterwards, the earliest possible start time ($St_{i,1}$) for a given activity (i) in the first repetitive unit (j = 1), is determined from the maximum start time among all predecessors, as shown in Eqn (5). The finish time of activity (i) in the first unit ($Ft_{i,1}$) would be computed by adding the activity’s duration (D_i) to the start time, as formulated in Eqn (6). After computing the activity’s finish time in the first unit, the activity’s start and finish times at any subsequent unit can be computed using Eqns (7) and (8) below.

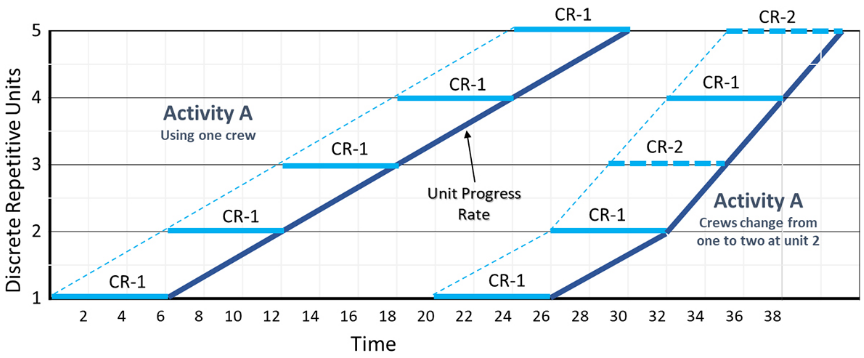


Figure 2. An example of a developed LOB for a given activity

$$St_{i,1}^{k} = Ft_{p_{i,k},1} + buf_{i,p_{i,k}};\ \forall k \in \{1,\ldots,K\}; \tag{4}$$

$$St_{i,1} = \max\left\{St_{i,1}^{1}, St_{i,1}^{2}, St_{i,1}^{k},\ldots., St_{i,1}^{K}\right\};\ \forall k \in \{1,\ldots,K\}; \tag{5}$$

$$Ft_{i,1} = St_{i,1} + D_{i,1}; \tag{6}$$

$$Ft_{i,j} = Ft_{i,j-1} + \left(\frac{1}{upr_{ij}}\right); \tag{7}$$

$$St_{i,j} = Ft_{i,j} - D_{i,j}, \tag{8}$$

where, $Ft_{p_{i,k},1}$ is the finish time of the preceding activity ($p_{i,k}$) in the first repetitive unit (j = 1), $buf_{i,p_{i,k}}$ is the time buffer between activity (i) and the preceding activity ($p_{i,k}$), if any, $Ft_{i,j}$ is the finish time of activity (i) in unit (j), upr_{ij} is the unit progress rate of activity (i) in unit (j), and $St_{i,j}$ is the start time of activity (i) in unit (j).

To avoid any logical conflicts between the activity and its predecessors due to the change in progress rates after changing the number of crews, a delta-shift time ($\Delta_{i,j}^{k}$) is calculated at each unit between the potential start time of the activity at each unit ($St_{i,j}$) and the finish time of the ranked predecessor at the same unit ($Ft_{p_{i,k},j}$), as shown in Eqn (9). Afterwards, the computed start time in Eqn (8) is modified or rescheduled by shifting it forward using the maximum delta-shift time value as formulated in Eqns (10) and (11).

$$\Delta_{i,j}^{k} = Ft_{p_{i,k},j} - St_{i,j},\ \forall k \in \{1,\ldots,K\}, \forall j \in \{1,\ldots,N\}; \tag{9}$$

$$\Delta_{i} = \max\left\{0, \Delta_{i,1}^{1},\ldots, \Delta_{i,N}^{1}, \Delta_{i,1}^{2},\ldots, \Delta_{i,N}^{2}, \Delta_{i,j}^{k},\ldots., \Delta_{i,N}^{K}\right\},$$
$$\forall k \in \{1,\ldots,K\}, \forall j \in \{1,\ldots,N\}; \tag{10}$$

$$St_{i,1}^{'} = St_{i,1} + \Delta_{i}, \tag{11}$$

where, Δ_i is the maximum delta-shift time computed from the difference between the start time of activity (i) and the finish time of each ranked predecessor ($p_{i,k}$) in each unit unit (j), and $St_{i,1}^{'}$ is the rescheduled start time of activity (i) in first repetitive unit (j = 1) after being shifted with the amount of Δ_i to avoid conflicts with the predecessors due to the change in progress rates.

However, to allow leveling the resources, activities are allowed to be delayed from their earliest start time to minimize resource consumption and fluctuations. Accordingly, a delay variable (D_{level}) is added to the earliest start time ($St_{i,1}^{'}$) computed in Eqn (11), for a given activity (i) in the first unit to accommodate the leveling objective, as shown for ($St_{i,1}^{''}$) in Eqn (12) below. Accordingly, the finish time ($Ft_{i,1}$) of the first repetitive unit (j = 1) in each activity (i) that was computed in Eqn (6) will be rescheduled as shown for ($Ft_{i,1}^{''}$) in Eqn (13) below. Finally, the project completion time can be computed from the latest finish time among all activities (i = 1 to m) in the last unit (j = N), as shown in Eqn (14). There is a need to determine the optimum delay "D_{level}" for each activity to avoid exceeding the desired completion time, therefore, it has been set as a decision variable as will be further described in Section 4.3.1 of the optimization module.

$$St_{i,1}^{''} = St_{i,1}^{'} + D_{level_i}; \tag{12}$$

$$Ft_{i,1}^{''} = St_{i,1}^{''} + D_{i,j}; \tag{13}$$

$$CT = \max\{Ft_{1,N}, Ft_{2,N},\ldots Ft_{i,N}..Ft_{m,N}\};\ \forall i \in \{1,\ldots,m\}, \tag{14}$$

where, $St_{i,1}^{''}$ is the modified start time of activity (i) in the first unit (j = 1) after being delayed with the amount of D_{level_i} to meet the leveling objective, CT is the project completion time, $Ft_{i,1}^{''}$ the modified finish time of activity (i) in the first unit (j = 1) following the modification in the start time, and ($Ft_{i,N}$) is the finish time of activity (i) in the last unit (j = N).

To facilitate tracking the computation of the start and finish time of each activity in each given unit using the aforementioned equations, a flow chart has been developed as shown in Figure 3. In the figure, the steps of the scheduling module are divided into three phases. The first phase is the initial phase where the start and finish times of each activity in each unit are computed, using Eqns (1) to (8), regardless the impact of changing the number of crews in a given activity. In the second phase, the calculated start and finish times in phase 1 are rescheduled to consider the impact of changing the number of crews by shifting the activities using Eqns (9) to (11). In the third phase, a delay is add to the start time of each activity, if needed, to accommodate the resource leveling objective. The finish time of each activity and the project's completion time are modified accordingly, as shown in Eqns (12) to (14). It should be noted that the LOB scheduling algorithm is designed in this module to maintain strictly the crew work continuity for each crew utilized in each activity where each crew moved from one unit to another without being idle for waiting predecessor crew to finish its work.

4.2. Resource leveling module

As previously mentioned in the LOB schedule module, activities are allowed to be shifted from their earliest start time in an effort to minimize the resource consumption and fluctuations. However, to evaluate the effectiveness of this shift, there is a need to compute the corresponding resource consumption and fluctuations to this shift. Therefore, the main objective of this module is to: 1) generate the resource profile or histogram, 2) compute the maximum daily resource consumption, and 3) capture the resource fluctuations. Accordingly, the daily resource consumption (R_t) and maximum daily value ($R_{\max}$) are computed as shown below in Eqns (15) and (16):

$$R_t = \sum_{i=1}^{m}\sum_{j=1}^{N}(Nc_{ij} \times r_i)_t,\ Nc_{ij} = \begin{cases} Cn_i, & j \le U_i \\ Cn_i^{*}, & j > U_i \end{cases}; \tag{15}$$

$$R_{\max} = \max\{R_1, R_2, R_t \ldots.. R_T\}, \tag{16}$$

where, R_t is the resource consumption in a given day (t), Nc_{ij} is the number of crews of activity (i) at unit (j) at a given time (t) considering the possible change in the number of crews from (Cn_i) to (Cn_i^{*}), and r_i is the number of labors existing in the crew of activity (i).

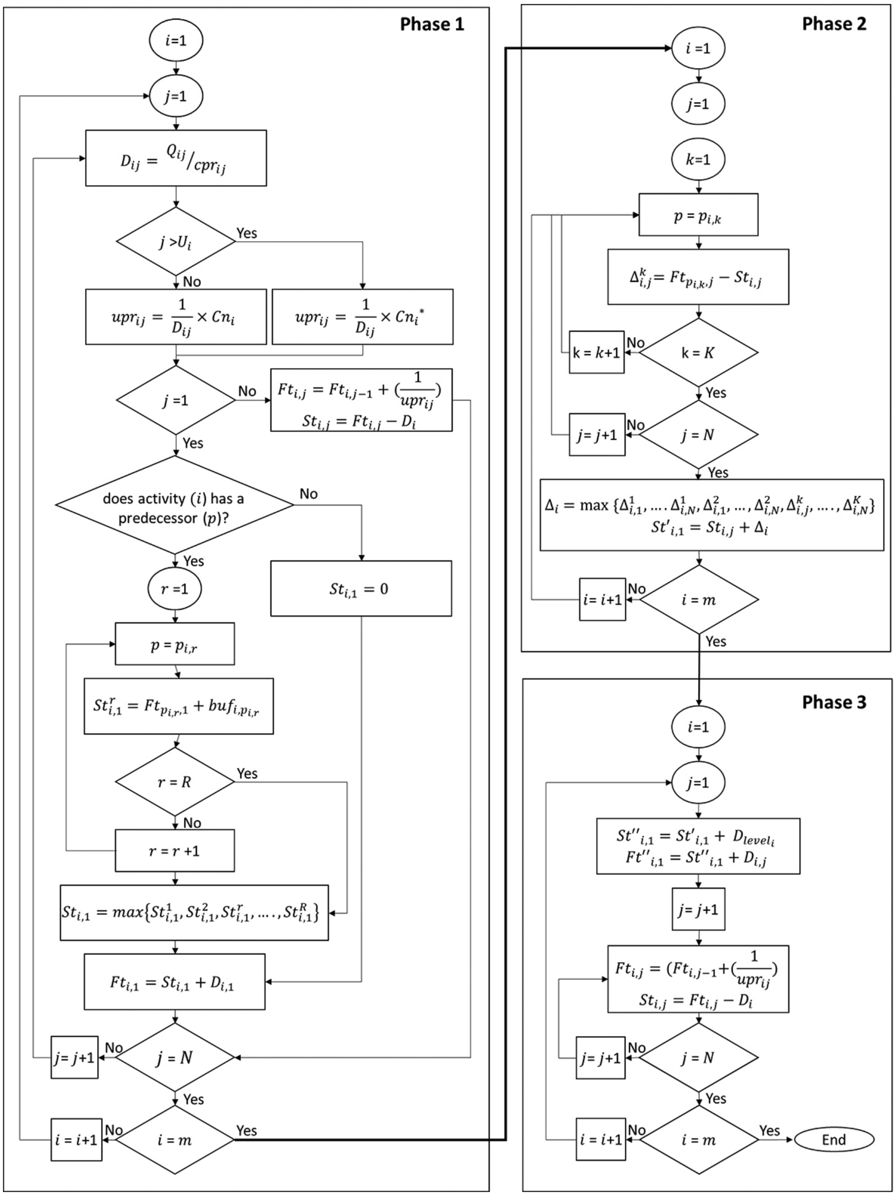


Figure 3. LOB scheduling module steps

To capture the resource fluctuations, this research adopts the method of computing the deviation of the daily resource consumption from the overall average consumption (Damci et al., 2013a). The average daily consumption is computed using Eqn (17) below. Afterwards, the overall deviation of the daily resource consumptions from the average is computed as shown in Eqn (18). Figure 4 illustrates the adopted methodology to formulate the resource consumption and fluctuations. The smaller the deviation is, the less the fluctuations are. Therefore, the objective of the optimization model is to minimize this deviation and the peak resource consumption, as described in the following section.

$$R_{avg} = \frac{\sum_{t=1}^{T} R_t}{TD}; \tag{17}$$

$$Dv = \sum_{t=1}^{T} Dv_t = \sum_{t=1}^{T} \left| R_{avg} - R_t \right|, \tag{18}$$

where, R_{avg} is the average daily consumption, R_t is the resource consumption in a given day (t), TD is total project duration computed as shown in Eqn (17), and (Dv) is the overall deviation of the daily resource consumption deviations from the average consumption.

4.3. Optimization module

The main purpose of this module is to find an optimal or near-optimal schedule for repetitive construction projects that minimize the daily resource consumption and fluctuations. As previously described, activities are allowed to be delayed from their earliest start time to level the resources. As a result, the project duration might get extended.

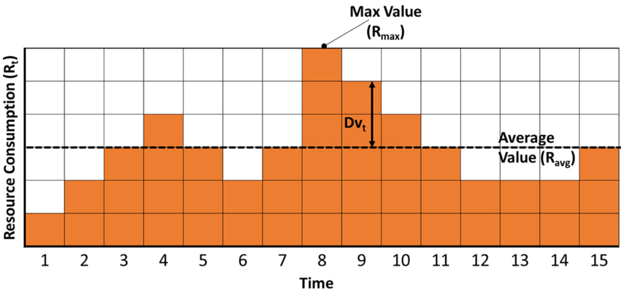


Figure 4. Illustration of the resource leveling methodology

To avoid exceeding the desired project duration, the model allows changing the number of crews assigned to a given activity to increase the progress rate. It allows as well increasing the number of crews at a certain repetitive unit to further enhance the progress rate. Accordingly, the aim of the developed optimization module is to determine for each activity (*i*) at each repetitive section (*j*): 1) near-optimal start time by determining the optimum leveling delay, 2) near-optimal number of crews for every activity, 3) the optimal unit where the number of crews can change to achieve optimum progress rate that satisfies the desired duration, and 4) Near-optimal new number of crews for every activity after change. The optimisation model is formulated as a linear integer programming model. The model's parameters and formulations are shown in the following sections.

4.3.1. Decision variables

To achieve the four goals of the optimization model as previously described, four decision variables have been identified along with their upper and lower boundaries in the optimization model, as follows:

1) "D_{level_i}" is the delay allowed for every activity (*i*) from its earliest start time to serve the leveling purpose, as formulated in Eqn (12):

$$0 \le D_{level_i} \le \frac{N-1}{upr_i}. \tag{19}$$

2) "Cn_i" is the number of crews assigned to activity (*i*) which determines the unit progress rate as formulated in Eqn (1):

$$1 \le Cn_i \le \max Cn_i, \tag{20}$$

where, "max Cn_i" is the maximum available number of crews that can be assigned to activity (*i*). The maximum theoretical value of max Cn_i is "*N*" which represents the total number of units assuming one crew working in each unit.

3) "U_i" is the unit number where the number of crews changes in activity (*i*) to achieve the desired progress rate, as formulated in Eqn (3):

$$1 \le U_i \le N. \tag{21}$$

4) "Cn_i^*" is the new number of crews assigned to activity (*i*) after the change at unit "U_i" to achieve the desired progress rate, as shown in Eqn (3):

$$1 \le Cn_i^* \le \max Cn_i. \tag{22}$$

The upper and lower boundaries of the variable "Cn_i^*" is based on the assumption that only one crew can be assigned to any given unit, and that the crew is not released until the assigned unit is completed.

4.3.2. Constraints

The lower and upper boundaries of the previously described decision variables have been set as constraints in the developed optimization model, as formulated in Eqns (19) to (22). Construction projects are usually constrained with a desired project duration. Accordingly, the project completion, computed using Eqn (14), has been constrained to meet a desired project duration (DD), as follows:

$$CT \le DD. \tag{23}$$

4.3.3. Objective function

To minimize resource consumption and fluctuations, there is a need to minimize the maximum resource daily consumption computed using Eqn (16) and the overall deviation from the average daily consumption computed using Eqn (18). Accordingly, to satisfy both objectives, a multi-objective function has been formulated to minimize both using the following equation:

$$\text{Min:}Y = (Dv + P \times R_{\max}), \tag{24}$$

where, Dv is the overall deviation, $R_{\max}$ is the maximum daily resource consumption, and P is a penalty variable that helps penalize any increase in the peak resource consumption and thus helps minimize $R_{\max}$.

4.3.4. Optimization solver

To arrive at the optimum solution, different combinations of the decision variables need to be examined. Accordingly, it is a combinatorial problem. To solve this problem, genetic algorithms (GA) has been utilized to arrive at the optimum solution, as it is quite efficient in solving problems with medium and large solution spaces (Saad et al., 2021; Hegazy & Kamarah, 2008; Senouci & Eldin, 2004). GA is a metaheuristic optimization solver that replicates the natural evolution of genes, survival of the fittest, when searching for the optimum solution. The GA process consists mainly of three operations: reproduction, crossover, and mutation. Reproduction is the process of selecting the parents for the crossover and mutation operations. The crossover operation is responsible for exchanging genes between two parents to generate a new population. To allow searching the whole solution space, the mutation operation is responsible of enforcing diversity by introducing random changes in the generated solutions.

Thus, GA process follows sequential steps based on the defined objective function, decision variables, and constraints (Senouci & Eldin, 2004): 1) generation of chromosomes based on the decision variables, 2) generation of the initial parent population of chromosomes, 3) selection of an offspring generation based on the fitness values of the strings in the parent population, then (4) generate the next generation of solution strings through crossover and mutation. Figure 5 illustrates a flowchart that shows the GA process in the proposed optimization module to arrive at the optimum LOB schedule. Moreover, it demonstrates the interaction between the three developed modules: scheduling module, resource leveling module, and the optimization module.

5. Application example

To verify and validate the proposed model and illustrate its capabilities, a comparative analysis has been conducted using a case study from the literature review. The study was originally used by Tokdemir et al. (2006) to generate LOB schedule without attempting to consider the leveling of resource; and later was used by Damci et al. (2013b) and Tang et al. (2018) to illustrate the capabilities of their developed models in optimizing LOB schedules while considering the leveling of resource. The case study is for a 26 Km pipeline construction project that included 7 activities (i = 1 to 7) and 26 repetitive units (each unit consists of 1 Km, j = 1 to 26). Table 1 shows the data of the original example for each activity including: activity ID, activity description, logical sequence between the activities (IPA), quantity of work in terms of working hours required to finish the activity, the number of workers per crew, crew production rate considering the number of workers and working hours in a day, the duration required to complete one unit, the number of crews travelling across the units as in original case, and the corresponding unit progress rate (upr_i).

To verify the mathematical formulation of the developed model and the embedded scheduling module, the generated LOB schedule and resource profile have been compared to the ones generated by Tokdemir et al. (2006), without running the optimization model. Accordingly, the LOB schedule was generated using the LOB Schedule Module by applying Eqns (3) to (14), as shown in Figure 6. The resource profile associated with the developed LOB schedule is generated using Resource Leveling Module by applying Eqns (15) to (18) as shown in Figure 7. In this comparison analysis, it has been assumed that: (1) the number of crews for each activity is similar to the original case presented by Tokdemir et al. (2006) (see column 7 in Table 1) as well as keeping number of crews fixed from the first repetitive unit to the last unit, and (2) the activity starts at its earliest possible start which means that D_{level_i} is assumed to be equal to zero. The outcome of this analysis confirms that the results generated by the developed model are identical to the one generated by the original example presented by Tokdemir et al. (2006) with project duration of 65 days and the daily resource consumption with a maximum value ($R_{\max}$) of 102 and a minimum value of 6.

It can be noted from the results of original case that was generated without optimizing the level of resource that there are undesirable deep valleys in the resource profile; the resource number increase to 72 then decreases to 14, and afterwards increase one more time to 102 and so forth. Accordingly, there is a need to produce an optimal LOB schedule that reduces those fluctuations whenever possible while maintaining the project duration at 65 days.

To illustrate the capabilities of the proposed optimization model and compare its performance against the existing models in terms of generating the optimal/near optimal LOB schedule that reduces resource fluctuations and consumption, two scenarios have been carried out.

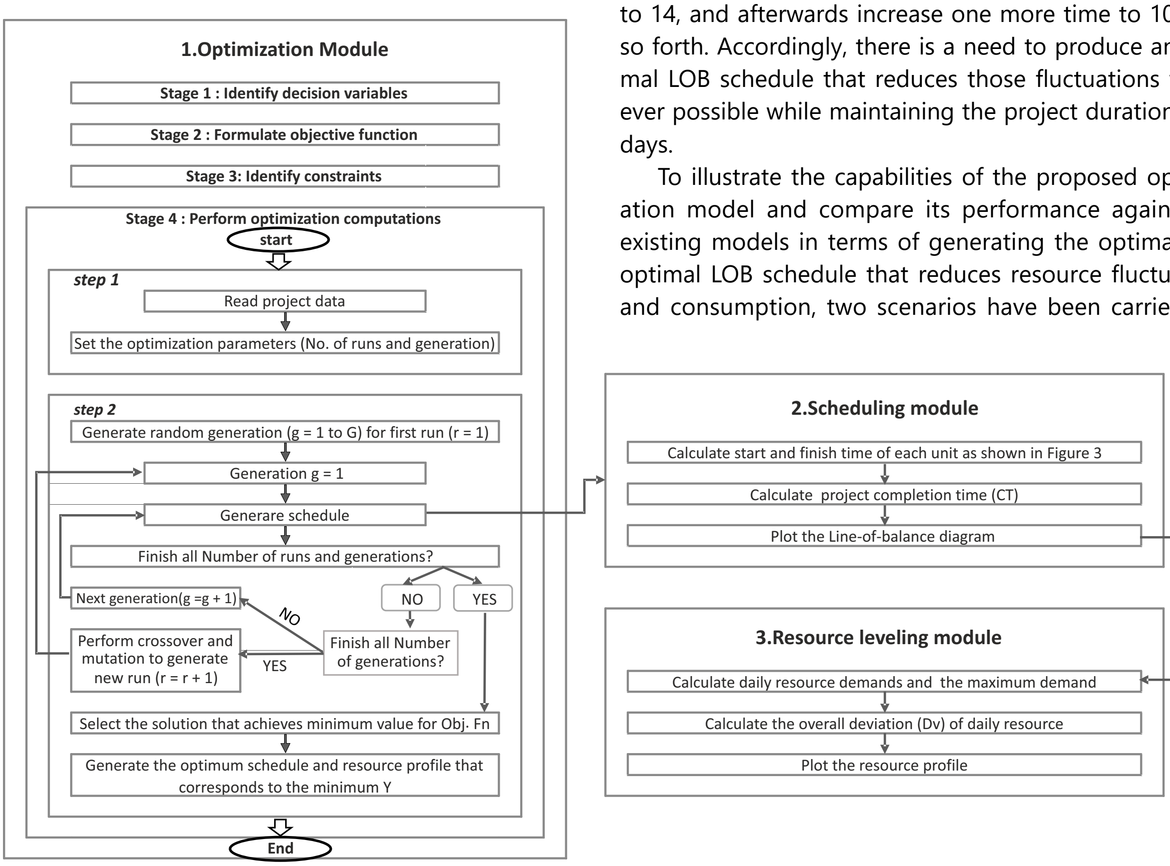


Figure 5. Sequential steps of the proposed LOB-RL Optimization Model

Table 1. Original case study description

Act	Activity Description	IPA	Quantity (Q_i) w-hrs	No. of workers per crew	Crew Productivity (cpr_i) hr/day	No. of crews (Cn_i)	Duration days	Unit progress rate (upr_i)
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
A	Locating & clearing	–	96	6	48	2	2	1
B	Excavating	A	64	8	64	1	1	1
C	Laying aggregate	B	80	10	80	1	1	1
D	Laying pipes	C	84	7	56	2	1.5	1.33
E	Testing	D	80	10	80	1	1	1
F	Backfilling	E	96	6	48	2	2	1
G	Compacting	F	144	9	72	2	2	1

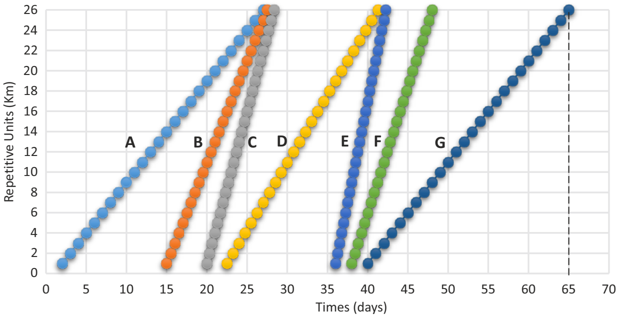


Figure 6. LOB schedule based on data of original case (before resource leveling)

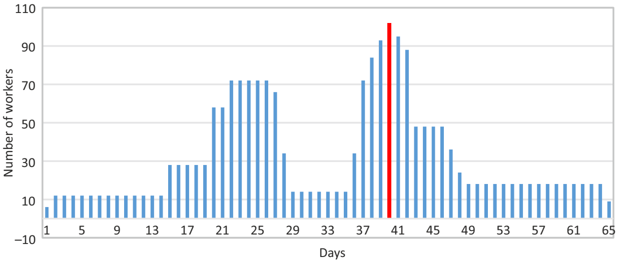


Figure 7. Resource profile based on data of original case (before resource leveling)

In the first scenario, the objective of the optimization model is to determine the optimum start time of each activity and the optimum number of crews travelling across the units that would minimize the maximum resource consumption and the overall deviation. In this scenario, the number of crews will remain fixed across the units. Accordingly, only the first two decision variables of the optimization model are used in this scenario. In the second scenario, on the other hand, the number of crews is allowed to change at a certain unit within the same activity. Therefore, in this scenario, the objective of the optimization model is to determine the optimum unit to change the number of crews, the optimum number of crews before and after change, and the optimum start time of the activity. Thus, the all four decision variables previously described in the optimization module are used in this scenario. To validate the model, the results of those two scenarios will be compared against the earlier efforts of Damci et al. (2013b) and Tang et al. (2018) that attempted as well to minimize resource consumption fluctuations.

The optimization setup has been developed in an excel spreadsheet, where the solver "SolveXL", which is a widely used Excel Add-in in solving engineering problems, has been used. To run the optimization model, GA parameters in "SolveXL" have been identified based on the solver's user guide and the literature review (Hassanat et al., 2019; SolveXL, 2018), including: problem type, population size, crossover rate, and mutation rate. The most influential parameter is the population size, if it is too small, GA may fail to reach an optimum solution, and if it is too large, GA will take longer time to complete a run and reach a solution. Accordingly, a population size of 100 has been selected which is quite reasonable for a problem of medium solution space (Saad et al., 2021). The number of generations and runs have been set to 1000, and 100, respectively. For the crossover rate, it has been set to the solver's standard value of 0.95, using the "Simple one point" setting. This rate controls the crossover operation where a child is generated from two selected parents by exchanging genes based on the selected value. The "simple one point" setting allows dividing each chromosome at a particular point where the crossover takes place by recombining one section of one of the chromosomes with the opposite section of the other. To generate new population, the "Roulette" setting of the solver has been selected. This setting gives a higher chance for an individual to be selected in a roulette for further mutation if it has a high fitness ratio with respect to the whole population fitness. The mutation rate has been set to 0.05, using the "Simple Mutator" setting. It allows only one gene in a given chromosome to change randomly during the crossover operation based on the selected value to avoid getting stuck in local optimal solutions. The "generational" algorithm type has been selected in the solver, which permits at each iteration, an entire new population to be generated from the old one through mutation and crossover. The problem type has been set as a single objective, since the solver tries to minimize a single value "Y" as formulated in Eqn (24). In both scenarios, the maximum number of crews "$\max Cn_i$" that is allowed to be used for any given activity has been set to four crews; and the penalty variable in the objective function formulated in Eqn (24), has been set to 100 after several trials to arrive at better results.

5.1. Scenario I

As stated earlier, in this scenario, the number of crews will be enforced to remain fixed across all the activity units. Accordingly, only the first two decision variables of the optimization model are used in this scenario. The objective of this scenario is to determine (a) the optimum start time of each activity, and (b) the optimum number of crews travelling across the units.

Since the project in the case study has 7 activities, the number of decision variables in the optimization model is 13 decision variables; 6 variables for the leveling delay time "D_{level_i}" for each activity excluding activity A that should start at time 0, and 7 variables for the number of crews Cn_i used in each activity. After running the optimization model for this scenario, the optimal/near optimal LOB schedule is generated as shown in Figure 8, where it can be noticed that the project duration is maintained at almost 65 days. Table 2 shows the optimum results for the start time and the number of crews for each activity. It can be noticed that the progress rates of activities A, D, E, and G are increased by increasing the number of crews to overcome the delays caused due to leveling and thus avoid exceeding the desired project duration. The resource profile associated with the generated LOB schedule is presented in Figure 10a showing minimum and maximum resource consumptions ($R_{\min}$ and $R_{\max}$) of 12 and 39, respectively.

Table 2. Results of scenario 1

Act	Activity Description	Optimum Start time ($St_{i,1}$)	Optimum No. crews (Cn_i)
A	Locating and clearing	0	3
B	Excavating	2	1
C	Laying aggregate	5	1
D	Laying pipes	19	3
E	Testing	31	2
F	Backfilling	35	2
G	Compacting	47	3

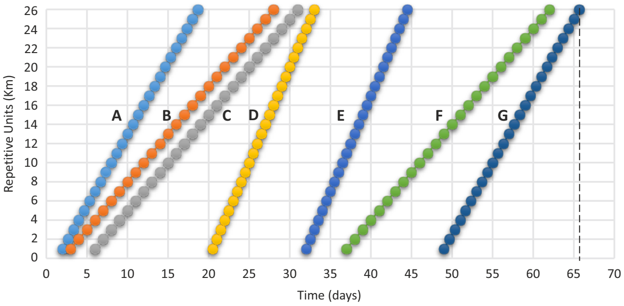


Figure 8. LOB schedule after implementing scenario I

5.2. Scenario II

As stated earlier, in this scenario, the number of crews is allowed to change at a certain unit within the same activity. Therefore, the all four decision variables previously described in the optimization module are used in this scenario. The objective of the optimization model in this scenario is to determine the optimum unit to change the number of crews, the optimum number of crews before and after change, and the optimum start time of the activity.

The number of decision variables in this scenario is 27 decision variables; 6 variables for the leveling delay time of the activities D_{level_i} (activity A is not included because it should start at time 0); 7 variables for number of crews Cn_i used in each activity; 7 variables for unit number U_i where the number of crews change; and 7 variables for new number of crews Cn_i^* after change. After running the optimization model for this scenario, the optimal/near optimal LOB schedule is generated as shown in Figure 9, where it can be noticed that the project duration is maintained at exactly 65 days. Table 3 shows the optimum values for each decision variable after running the model.

Table 3. Results of scenario II

Act	Activity Description	Start time ($St_{i,1}$)	No. crews (Cn_i)	Unit of crew change (U_i)	New number of crews Cn_i^*
A	Locating and clearing	0	2	8	3
B	Excavating	2	1	–	–
C	Laying aggregate	3	1	–	–
D	Laying pipes	21	2	–	–
E	Testing	28	1	3	2
F	Backfilling	40	2	5	3
G	Compacting	42	2	18	4

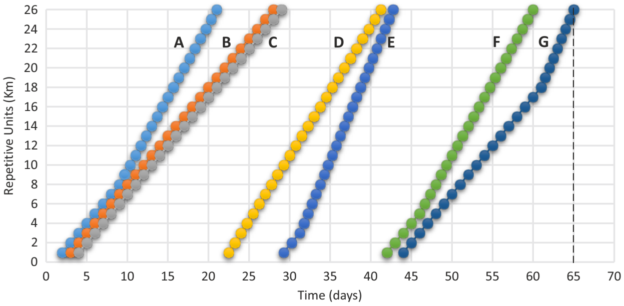


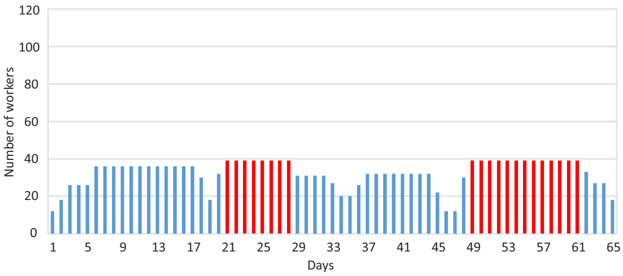
Figure 9. LOB schedule after implementing scenario II

It can be noted in Figure 9 that activities A, E, F and G are changing crews at units 8, 3, 5, and 18, respectively. The remaining activities are using fixed number of crews across the units. In comparison with scenario 1, the start of activity F is delayed 5 days. To overcome this delay, Activity G utilized more resources by changing from 3 to 4 crews at the 18th unit. The resource profile associated with the generated LOB schedule is presented in Figure 10b showing minimum and maximum resource consumptions ($R_{\min}$ and $R_{\max}$) of 6 and 36, respectively.

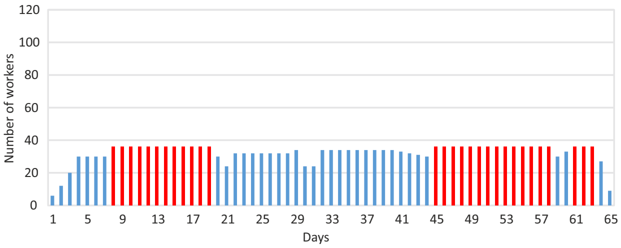
5.3. Discussion

To further validate the proposed LOB-RL optimization model, a comparison has been conducted between the results of the proposed model and the ones generated by the earlier efforts of Damci et al. (2013b) and Tang et al. (2018) using same case study. Figure 10 shows the resource profiles generated using the proposed model in case of scenario one and two, and the resource profiles after implementing the previously mentioned earlier efforts. It can be noticed that all four models resulted in a better resource profile in terms of daily resource fluctuations compared to the original case. Table 4 summarizes the results of the four models in terms of: 1) total duration, 2) the maximum resource consumption (R_{max}), 3) the minimum resource consumption (R_{min}), and 4) total deviation Dv of the daily resource consumption deviations from the average consumption.

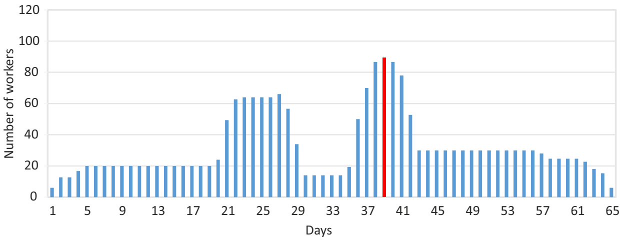
a) Proposed model (scenario I)



b) Proposed model (scenario II)



c) Damci et al. (2013b)



d) Tang et al. (2018

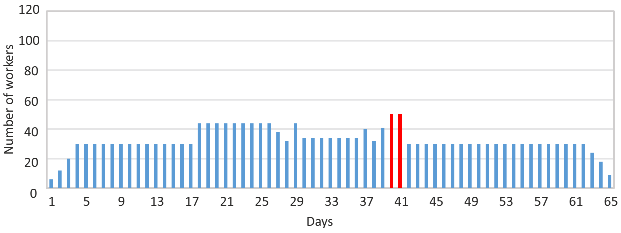


Figure 10. Resource profile using: a – scenario I of the proposed model; b – scenario II of the proposed model; c – model of Damci et al. (2013b); d – model of Tang et al. (2018)

From Table 4, that all four models maintained the desired project duration. The models of Damci et al. (2013b) and Tang et al. (2018) reduced the maximum resource consumption (R_{max}) from 102 to 89 (–13%) and 50 (–51%), respectively. However, the proposed model has arrived at better results; it reduced the maximum resource consumption (R_{max}) to 39 (–62%) in scenario I and 36 (–65%) in scenario II. Also, the valleys in the resource profile have been minimized in the resource profiles generated by the proposed model in comparison with the earlier efforts. Such that, Damci et al. (2013b) and Tang et al. (2018) reduced the total deviation from 1390 to 1037 (–25%) and to 374 (–73%), respectively. However, the proposed model has arrived at better results; it reduced the total deviation to 378 (–73%) in scenario I and 260 (–8%) in scenario II.

Accordingly, the conducted comparative analysis confirm that the proposed model outperforms the earlier efforts in terms of the overall deviation from the average consumption, and the maximum daily resource consumption especially in the second scenario where the feature that allows the crews to change within the same activity is used. Moreover, the difference between the maximum and the minimum daily resource consumption is minimized in the proposed model in comparison with the earlier efforts, which is quite effective in terms of resource procurement. In addition, the proposed model is designed to consider a non-serial repetitive construction projects where an activity is able to have one or more predecessors.

6. Conclusions

A new optimization model has been developed for improving the levelling of resource consumption in repetitive schedules. The model helps arrive at optimum LOB schedules with minimum resource fluctuations and daily resource consumption. Unlike the existing efforts, it minimizes the deviation of the daily resource consumption

Table 4. Comparing the performance of the leveled resource profiles using the proposed model and the existing models

Model	Total duration	Total deviation from average Dv	Minimum resource usage R_{min}	Maximum resource usage R_{max}
Tokdemir et al. (2006) (Original case *w/o* leveling)	65	1,390	6	102
Damci et al. (2013b)	65	1,037 (–25%)	6	89 (–13%)
Tang et al. (2018)	65	374 (–73%)	6	50 (–51%)
Proposed model (scenario I)	65	378 (–73%)	12	39 (–62%)
Proposed model (scenario II)	65	260 (–81%)	6	36 (–65%)

from the average consumption, while having the flexibility of changing the number of crews within the same activity to compensate for any delays due to the levelling process. The model consists of three modules: 1) LOB scheduling module, 2) resource levelling module, and 3) optimization module. It determines for each activity the optimum start time by determining the optimum levelling delay, the optimum number of crews traveling across the units, the optimum unit where the number of crews can be changed, and the optimum new number of crews after the change. To validate the performance of the model, an example from the literature has been used to compare the performance against earlier efforts. The results proved the outperformance of the proposed optimization model in comparison with the existing efforts in terms of resource consumption and fluctuations.

In essence, the proposed model has proved its efficiency in producing LOB schedules for repetitive projects that can result in more efficient and practical resource profiles. Thus, it can reduce hidden costs due to inefficient resource utilization on site.

7. Future work and limitations

Despite the novelty of the present optimization model in the field of resource management optimization in linear repetitive projects, number of future research aspects is identified to evolve the proposed model and eliminate its limitations. Currently, the proposed optimization model is formulated to meet a desired duration as a constraint. It can be extended in the future to a multi-objective model, where the duration can be minimized as well. Also, it can be extended to include the costs of hiring and firing (mobilization and demobilization) resources and being idle on-site due to inefficient scheduled resource consumption. In future research, the proposed methodology should be tested using a larger real-life case study of different repetitive project types to validate its merits. While this study investigates resource management under normal scheduling conditions, future work needs to extend the capability of the proposed approach to be dynamic for using it in monitoring and controlling the project in real-time and investigate resource management under crashing scheduling conditions, as well as uncertainties in large linear projects. In the current work, only a single crew resource type was used; future work may consider scheduling multiple crew resources in the project activities.

Availability of data

Data generated or analyzed during the study are available from the corresponding author by request.

Acknowledgements

Researchers Supporting Project number (RSP2023R280), King Saud University, Riyadh, Saudi Arabia.

References

Abdel-Basset, M., Ali, M., & Atef, A. (2020). Resource levelling problem in construction projects under neutrosophic environment. *Journal of Supercomputing*, *76*, 964–988. https://doi.org/10.1007/s11227-019-03055-6

Altuwaim, A., & El-Rayes, K. (2018). Optimizing the scheduling of repetitive construction to minimize interruption cost. *Journal of Construction Engineering and Management*, *144*(7), Article 4018051. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001510

Bakry, I., Moselhi, O., & Zayed, T. (2014). Optimized acceleration of repetitive construction projects. *Automation in Construction*, *39*, 141–151. https://doi.org/10.1016/j.autcon.2013.07.003

Christodoulou, S. E., Ellinas, G., & Michaelidou-Kamenou, A. (2010). Minimum moment method for resource leveling using entropy maximization. *Journal of Construction Engineering and Management*, *136*(5), 518–527. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000149

Damci, A., Arditi, D., & Polat, G. (2013a). Multi resource leveling in line of balance scheduling. *Journal of Construction Engineering and Management*, *139*(9), 1108–1116. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000716

Damci, A., Arditi, D., & Polat, G. (2013b). Resource leveling in line of balance scheduling. *Computer-Aided Civil and Infrastructure Engineering*, *28*(9), 679–692. https://doi.org/10.1111/mice.12038

El-Abbasy, M. S., Elazouni, A., & Zayed, T. (2016). MOSCOPEA: Multi-objective construction scheduling optimization using elitist non-dominated sorting genetic algorithm. *Automation in Construction*, *71*, 153–170. https://doi.org/10.1016/j.autcon.2016.08.038

El-Rayes, K., & Moselhi, O. (2001). Optimizing resource utilization for repetitive construction projects. *Journal of Construction Engineering and Management*, *127*(1), 18–27. https://doi.org/10.1061/(ASCE)0733-9364(2001)127:1(18)

El-Rayes, K., & Jun, D. (2009). Optimizing resource leveling in construction projects. *Journal of Construction Engineering and Management*, *135*(11), 1172–80. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000097

García-Nieves, J. D., Ponz-Tienda, J. L., Salcedo-Bernal, A., & Pellicer, E. (2018). The multimode resource-constrained project scheduling problem for repetitive activities in construction projects. *Computer-Aided Civil and Infrastructure Engineering*, *33*, 655–671. https://doi.org/10.1111/mice.12356

García-Nieves, J. D., Ponz-Tienda, J. L., Ospina-Alvarado, A., & Bonilla-Palacios, M. (2019). Multipurpose linear programming optimization model for repetitive activities scheduling in construction projects. *Automation in Construction*, *105*, Article 102799. https://doi.org/10.1016/j.autcon.2019.03.020

Harris, R. B., & Ioannou, P. G. (1998). Scheduling projects with repeating activities. *Journal of Construction Engineering and Management*, *124*(4), 269–278. https://doi.org/10.1061/(ASCE)0733-9364(1998)124:4(269)

Hassanat, A., Almohammadi, K., Alkafaween, E., Abunawas, E., Hammouri, A., & Prasath, V. B. S. (2019). Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach. *Information*, *10*(12), Article 390. https://doi.org/10.3390/info10120390

Hegazy, T. (1999). Optimization of resource allocation and leveling using genetic algorithms. *Journal of Construction Engineering and Management*, *125*(3), 167–175. https://doi.org/10.1061/(ASCE)0733-9364(1999)125:3(167)

Hegazy, T., & Kamarah, E. (2008). Efficient repetitive scheduling for high-rise construction. *Journal of Construction Engineering and Management*, *134*(4), 253–264. https://doi.org/10.1061/(ASCE)0733-9364(2008)134:4(253)

Hegazy, T., & Wassef, N. (2001). Cost optimization in projects with repetitive non-serial activities. *Journal of Construction Engineering and Management*, *127*(3), 183–191. https://doi.org/10.1061/(ASCE)0733-9364(2001)127:3(183)

Hegazy, T., Saad, D. A., & Mostafa, K. (2020). Enhanced repetitive-scheduling computation and visualization. *Journal of Construction Engineering and Management*, *146*(10), Article 04020118. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001911

Hegazy, T., Mostafa, K., & Ojulari, S. (2021). Tetris-inspired approach for generating tightly-packed repetitive schedules. *Automation in Construction*, *124*, Article 103601. https://doi.org/10.1016/j.autcon.2021.103601

Hyari, K., & El-Rayes, K. (2006). Optimal planning and scheduling for repetitive construction projects. *Journal of Management in Engineering*, *22*(1), 11–19. https://doi.org/10.1061/(ASCE)0742-597X(2006)22:1(11)

Ioannou, P. G., & Yang, I. T. (2016). Repetitive scheduling method: Requirements, modeling, and implementation. *Journal of Construction Engineering and Management*, *152*(5), Article 04016002. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001107

Ipsilandis, P. G. (2007). Multiobjective linear programming model for scheduling linear repetitive projects. *Journal of Construction Engineering Management*, *133*(6), 417–424. https://doi.org/10.1061/(ASCE)0733-9364(2007)133:6(417)

Jun, D. H., & El-Rayes, K. (2011). Multi-objective optimization of resource leveling and allocation during construction scheduling. *Journal of Construction Engineering and Management*, *137*(12), 1080–1088. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000368

Koulinas, G. K., & Anagnostopoulos, K. P. (2012). Construction resource allocation and leveling using a threshold accepting–based hyper heuristic algorithm. *Journal of Construction Engineering and Management*, *138*(7), 854–863. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000492

Leu, S., & Hwang, S. (2001). Optimal repetitive scheduling model with shareable Resource constraint. *Journal of Construction Engineering and Management*, *127*(4), 270–280. https://doi.org/10.1061/(ASCE)0733-9364(2001)127:4(270)

Long, L. D., & Ohsato, A. (2009). A genetic algorithm-based method for scheduling repetitive construction projects. *Automation in Construction*, *18*(4), 499–511. https://doi.org/10.1016/j.autcon.2008.11.005

Saad, D. A., Masoud, M., & Osman, H. (2021). Multi-objective optimization of lean-based repetitive scheduling using batch and pull production. *Automation in Construction*, *127*, Article 103696. https://doi.org/10.1016/j.autcon.2021.103696

Senouci, A. B., & Eldin, N. N. (2004). Use of genetic algorithms in resource scheduling of construction projects. *Journal of Construction Engineering and Management*, *130*(6), 869–877. https://doi.org/10.1061/(ASCE)0733-9364(2004)130:6(869)

SolveXL. (2018). *SolveXL user manual*. Exeter Advanced Analytics LLP.

Tang, Y. J., Liu, R. K., & Sun, Q. X. (2014). Two-stage scheduling model for resource-leveling of linear projects. *Journal of Construction Engineering and Management*, *140*(7), Article 04014022. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000862

Tang, Y., Sun, Q., Liu, R., & Wang, F. (2018). Resource leveling based on line of balance and constraint programming. *Computer-Aided Civil and Infrastructure Engineering*, *33*(10), 864–884. https://doi.org/10.1111/mice.12383

Tokdemir, O. B., Arditi, D. & Balcik, C. (2006). ALISS: advanced linear scheduling system. *Construction Management and Economics*, *24*(12), 1253–1267. https://doi.org/10.1080/01446190600953706

Wang, Z., Hu, Z., & Tang, Y. (2020). Float-based resource leveling optimization of linear projects. *IEEE Access*, *8*, 176997–177020. https://doi.org/10.1109/ACCESS.2020.3027058