Journal of Civil Engineering and Management
ISSN 1392-3730 / eISSN 1822-3605
2023 Volume 29 Issue 2: 93–106
https://doi.org/10.3846/jcem.2023.17568

RESOURCE LEVELLING IN REPETITIVE CONSTRUCTION PROJECTS WITH INTERRUPTIONS: AN INTEGRATED APPROACH

Guyu DAI, Mingjuan LIAO, Renqian ZHANG*

School of Economics and Management, Beihang University, Beijing, China

Received 18 November 2021; accepted 2 June 2022

Abstract. Despite the significance of resource levelling, project managers lack various ways to smooth resource usage fluctuation of a repetitive construction project besides changing resource usage. Tolerating interruptions is an effective way to provide flexibility for a schedule but is ignored when solving resource levelling problems. Therefore, this paper investigates the impacts of interruptions on resource usage fluctuation and develops an integrated approach that simultaneously integrates two scheduling adjusting processes: changing resource usage and tolerating interruptions. In this paper, two interruption conditions are proposed to identify which activities are suitable to be interrupted for smoothing resource usage fluctuation. The traditional resource levelling model is modified to a new scheduling model by incorporating interruptions. A two-stage GA-based scheduling algorithm is developed by integrating changing resource usage and tolerating interruptions. A commonly used pipeline project is adopted to illustrate the steps of the proposed approach and demonstrate its effectiveness and superiority through comparison with previous studies. A large-scale project further verifies the usability of the proposed approach. The results confirmed the feasibility to smooth resource usage fluctuation by interruptions, and the integrated approach can achieve a more competitive resource levelling result.

Keywords: construction management, repetitive construction project, scheduling, resource levelling, work interruption, optimization.

Introduction

Repetitive construction projects represent a particular type of construction project that requires employed crews to repeat their work in all units of the same project and move from one unit to the next (Podolski, 2017; Tran et al., 2019). Examples of typical repetitive construction projects include highways, bridges, multi-story buildings and pipeline networks, which form a large portion of the construction industry (Podolski & Sroka, 2019). Massive construction resources, such as manpower, machinery and material, need to be frequently used when a repetitive construction project is under construction. However, frequent resource usage fluctuation inevitably brings many difficulties to project management, such as increasing management costs, complicating crew management and reducing the benefits of learning curve effects. Thus, more and more attention is paid to the resource levelling problem that concerns the smoothing-out of the resource usage fluctuation over the course of a project (Georgy, 2008). In order to achieve more efficient project execution, schedule adjusting processes need to be performed to reduce unnecessary resource usage fluctuations (Cheng et al., 2017). As a kind of scheduling adjusting process, changing resource usage smooths resource usage fluctuation by reallocating resource usage in different activities. Due to its convenience and effectiveness, changing resource usage has been widely used to smooth resource usage fluctuation.

In a repetitive construction project, crews need to repeat the same work in various units, continuously moving from one unit to another (Hyari & El-Rayes, 2006; Hassan et al., 2021). This unique requirement for repetitive construction projects is referred to as work continuity. Scheduling repetitive construction projects can benefit from work continuity due to: (1) reducing firing and hiring of labour, (2) minimizing the idle times of equipment or labour, (3) maximizing the benefits from learning curve effects (Hyari & El-Rayes, 2006; Altuwaim & El-Rayes, 2018a; Ammar, 2022). However, maintaining work continuity has its limitation. Some researchers have pointed out that strict compliance with work continuity

*Corresponding author. E-mail: *zhangrenqian@buaa.edu.cn*

may generate a schedule with a longer project duration (Hegazy & Wassef, 2001; Ammar, 2022). As another kind of scheduling adjusting process, interruptions reschedule some units and activities by breaking the work continuity of a schedule. In fact, tolerating interruptions can provide more flexibility for a schedule and brings two potential benefits to management: (1) reducing duration. Tolerating interruptions in a schedule allows some units to start earlier than the schedule compliant with work continuity and accordingly reduces the project duration. This benefit has been proved by previous studies (Hyari & El-Rayes, 2006; Ammar, 2022); (2) smoothing resource usage fluctuation. When a project manager decides to reschedule some units and activities by tolerating interruptions, the resource usage also needs to be rearranged to make the new schedule feasible. These changes naturally have an impact on the resource usage distribution curve. Therefore, tolerating interruptions provides a possible way for project managers to smooth resource usage fluctuation if interruptions can be applied to the right units.

Despite an effective way to smooth resource usage fluctuation, changing resource usage may not be appropriate for all contexts. Various scheduling adjusting processes should be developed and applied to achieve more efficient project execution. Tolerating interruptions can provide more flexibility for a schedule, but its impacts on resource usage fluctuation are still unclear. To address the above challenges, this paper investigates the resource levelling problem of repetitive construction projects based on the LOB (line of balance) technique. This paper mainly aims to investigate the impact of interruptions on resource usage fluctuation and develop a novel interrupted-based scheduling approach for solving resource levelling problems. Not limited to interruptions, changing resource usage is also considered for developing an integrated approach.

The contributions of this study are as follows: (1) the impacts of interruptions on resource usage fluctuation are investigated and revealed for the first time; (2) two interruption conditions are proposed to decide which activities can be selected as candidates for interruptions; (3) a new resource levelling model is formulated by incorporating interruption factors; (4) the proposed approach achieves a successful integration of changing resource usage and tolerating tolerations, and results in more efficient schedules compared to existing scheduling models.

The remainder of this paper is organized as follows. Section 1 presents a literature review. Section 2 presents two research bases related to this paper: LOB technique and interruption impacts. Section 3 presents a new resource levelling model by incorporating interruption factors. Section 4 develops a two-stage GA-based algorithm by integrating changing resource usage and tolerating interruptions. Section 5 conducts several case studies to illustrate the proposed approach and verify its superiority and usability. Finally, the conclusions are presented. In order to illustrate the structure of this research clearly, a general flowchart is presented, as shown in Figure 1.

1. Literature review

Network-based techniques, such as CPM (critical path method), have traditionally been used for planning, scheduling, and monitoring construction projects since the late 1950s (Ammar, 2020). Despite the wide applications of CPM, it fails to schedule repetitive construction projects due to the following shortcomings: (1) CPM requires a large number of nodes to represent a repetitive construction project; (2) CPM cannot demonstrate the production rate of activities; (3) CPM does not ensure resource continuity (Long & Ohsato, 2009; Altuwaim & El-Rayes, 2018a; Jaskowski & Biruk, 2020). Therefore, many more practical scheduling techniques have been developed for scheduling repetitive construction projects, such as Line of balance (LOB) (Arditi et al., 2001), linear scheduling method (LSM) (Harmelink & Rowings, 1998; Rogalska & Hejducki, 2007), and production scheduling method (PSM) (Lucko, 2008). These scheduling techniques commonly describe a repetitive construction project schedule by employing a two-dimensional coordinate system:

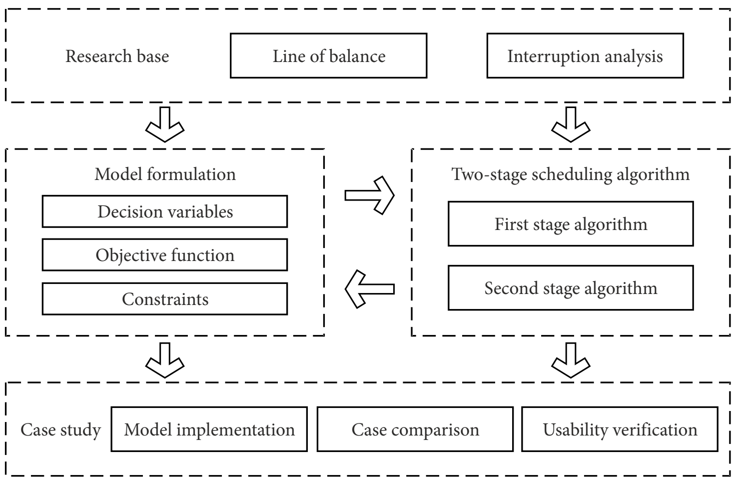

Figure 1. Research flowchart

time and distance(unit) (Tang et al., 2018a). Among these scheduling techniques, LOB is the most frequently used. As a resource-driven scheduling method, the main objective of LOB is to determine a balanced mix of crews and synchronize their work so that the unit works can be conducted smoothly (Ammar, 2013; Cho et al., 2013). Zhang et al. (2017) concluded that LOB is the best method for scheduling repetitive construction projects because it can provide a schedule with continuous use of resources. Therefore, this paper uses the LOB technique to schedule repetitive construction projects. The following section illustrates two research topics related to this paper: resource levelling and interruptions.

The resource levelling problem is generally described as a task aiming at minimizing resource usage fluctuation over the course of a project (Damci & Polat, 2014; Jaskowski & Biruk, 2018). In resource levelling, the total project duration is usually kept unchanged with no limitation on the maximum number of resources used at any point in time throughout the project duration (Georgy, 2008). In recent years, many studies have focused on the resource levelling problem in repetitive construction projects. Previous studies mainly solved this problem in two ways respectively: (1) changing the resource usage in these activities that have no impact on project duration: Mattila and Abraham (1998) formulated the resource levelling problem as an integer programming model based on a given initial schedule. The strategy for smoothing resource fluctuation is adjusting the resource usage in non-controlling activities. Damci et al. (2013a) investigated the resource levelling problem based on LOB and proposed a genetic algorithm to improve the resource levelling object by decreasing the number of crews. This algorithm selected the activities that have no impact on project duration as candidates for smoothing resource usage fluctuation; (2) deciding the resource usage of all the activities. Georgy (2008) presented a genetic algorithm for resource levelling using a linear scheduling method. In this study, every activity is considered in its entirety for resource adjustments. Ammar (2020) formulated the resource levelling problem at the activity level using integer programming and obtained a better result than previous studies. In conclusion, the above two ways both smooth resource usage fluctuation by changing resource usage in some activities or units, whether in non-critical activities or not. The first way can quickly search for an effective schedule, and the second way can obtain a better schedule by spending more time.

Based on the two ways mentioned above, some studies further extended the content of the resource levelling problem of repetitive construction projects. Lucko (2011) used a new function called the singularity function to solve the resource levelling problem. Damci et al. (2016) compared the impacts of 10 different objective functions that are commonly used in CPM on the resource levelling problem of repetitive construction projects. Some studies considered the application of multi-resource and multi-objective can make resource levelling results more practical (Damci et al., 2013b; Zhang et al., 2017; Tang et al., 2018b). Tang et al. (2014) concluded that separating using the above two ways is not sufficiently flexible and proposed a novel scheduling model by integrating them. Although these studies have made significant progress in resource levelling, they all smooth resource usage fluctuation by changing resource usage. However, whether there is another possible way to smooth resource usage fluctuation is still not investigated. This may lead to a dilemma that project managers lack various tools to deal with resource levelling problems in different situations.

As stated above, the strict application of work continuity may lead to a longer overall project duration. Tolerating interruptions can provide more flexibility for a schedule by allowing some units to change their start time. Therefore, many researchers have regarded tolerating interruptions as an effective way to reduce project duration and tried to investigate its impacts on repetitive construction projects. Hegazy and Wassef (2001) concluded that interruptions could positively impact project duration because they allow some units of a schedule to start earlier than the planned schedule. Amini and Heravi (2009) deduced specific calculation formulas for interruptions, including the number of required interruptions, the unit for applying interruptions, and the optimal duration of each interruption. Agrama (2014) concluded that different types of interruptions might have different effects on a schedule: some interruptions are beneficial for scheduling because of the ability to compress durations, but some are not. Altuwaim and El-Rayes (2018a) analyzed the impact of interruptions based on float analysis. They proposed two new types of work continuity floats which consider the impact of delaying the early start time of activities on work continuity.

Inspired by the above literature, many researchers developed effective scheduling models by tolerating interruptions. The objectives of these proposed scheduling models contain minimizing project duration (El-Rayes & Moselhi, 2001; Liu & Wang, 2007; Ammar, 2022; Hegazy et al., 2021) and minimizing costs (Hegazy & Wassef, 2001; Altuwaim & El-Rayes, 2018b). The results of these studies proved that tolerating interruptions can obtain better schedules. However, Bragadin (2010) pointed out that allowing some interruptions can reduce total project duration and indirect cost but increase direct cost because of the idle time. Therefore, trade-off analysis between conflicting objectives was further investigated, such as project duration and costs (Long & Ohsato, 2009; Zou et al., 2022), costs and work continuity (Zou et al., 2018), project duration and work continuity (Hyari & El-Rayes, 2006; Altuwaim & El-Rayes, 2018a) and all the three objectives (Zou et al., 2021; Arabpour & Moselhi, 2021). These studies aim at achieving an optimal trade-off between conflicting objectives when interruptions are tolerated. Eid et al. (2021) considered the impact of delay and presented a multi-objective scheduling model aiming at achieving a trade-off between minimizing project duration, cost, work interruptions and unit delivery delays.

In conclusion, the impact of interruptions on project duration has been thoroughly investigated. The performance of some optimization objectives, including project duration and costs, has been improved significantly by integrating interruptions into scheduling models. In fact, resource levelling also plays a vital role in successfully implementing a repetitive construction project. Despite the contributions of available literature, how to solve the resource levelling problem by using interruptions is not investigated until now, which limits the further improvement of resource levelling results.

2. Research base

Two research bases are involved in this paper. The first research base is the LOB technique because the following scheduling model is established based on it. The first subsection will present a brief explanation for LOB and illustrate how to generate a schedule maintaining work continuity by this scheduling technique. The second research base is the impact of interruptions on resource usage fluctuation. The second subsection will analyze the impact of interruptions on resource usage and provide some evidence to show which kinds of interruption can be applied to smooth resource usage fluctuation.

2.1. LOB calculation

In this paper, a repetitive construction project is scheduled by the LOB technique. Figure 2 shows the most common graphical description of a LOB schedule, where each activity is represented by a bar and each unit is represented by a horizontal line (Ammar, 2013). The unit duration is represented by the width of the bar, which is assumed constant along with all units. The intersection of a horizontal line and the left side of the bar represents the start time of a unit, and the intersection of a horizontal line and the right side of the bar represents the finish time. The slope of a bar represents the production rate of an activity. Some project data are required to establish a LOB schedule, such as the number of units, man-hour estimate, daily working hours, and unit duration of each activity. Having the project data, a LOB schedule can be generated by the following calculation procedure.

The calculation procedure identifies the start time (s_{ij}) and finish time (f_{ij}) of activity i in unit j. With reference to Figure 2, the production rate (r_i) links the start time between different units in an activity. The relation of start time between adjacent units is expressed by Eqn (1):

$$s_{ij} = s_{i(j-1)} + 1 / r_i. \tag{1}$$

To maintain the work continuity of a repetitive construction project, crews' movement should be synchronised between repetitive units within each activity (Ammar, 2020). As shown in Figure 2, when crew 1 employed by Activity B completes unit 1, it needs to move to unit 3 for construction then. Therefore, the work continuity forces the production rate r_i of Activity i to be equal to the ratio of the number of crews (c_i) and its unit duration (d_i). Substituting r_i by the ratio (c_i/d_i), Eqn (1) is reformulated by Eqn (2):

$$s_{ij} = s_{i(j-1)} + d_i / c_i. \tag{2}$$

The finish time (f_{ij}) is equal to the start time (s_{ij}) plus the unit duration (d_i) and is calculated by Eqn (3):

$$f_{ij} = s_{ij} + d_i. \tag{3}$$

Precedence relations guarantee that each activity in a repetitive construction project can be completed in order. If Activity j is the succeeding activity of Activity i, the start time of Activity j must be greater than or equal to the finish time of Activity i at all units $\left(s_{ik} + d_i \le s_{jk}\right)$. In LOB, if the precedence relation is preserved at the first or the last unit, the precedence relations at other units can be satisfied automatically when the resource continuity is maintained. Taking the case shown by Figure 1 as an example, precedence relations between Activity A and B (Activity B and C) at all units are satisfied by only preserving the precedence relation at the first unit (the last unit, defined as unit m). Obeying the precedence relation, the start time of Activity i at unit 1 can be calculated by Eqn (4):

$$s_{i1} = \max(f_{(i-1)1}, f_{(i-1)m} - (m-1)d_i / c_i). \tag{4}$$

Assuming there is a dummy activity named Activity 0, each unit's start time and end time in this activity are set as zero. Then, a schedule that maintains work continuity and precedence relation can be generated by combining Eqns (2), (3) and (4). Assuming a repetitive construction project containing n activities and m units, the scheduling algorithm presented in Figure 3 shows the specific procedure that how to generate a LOB schedule.

2.2. Interruption analysis

An interruption is defined as a planned break in work continuity, which can be directly demonstrated by the delay of the start of an activity at a specific unit from its planned start time based on the work continuity (Hegazy & Wassef, 2001; Ammar, 2022). The application of interruptions

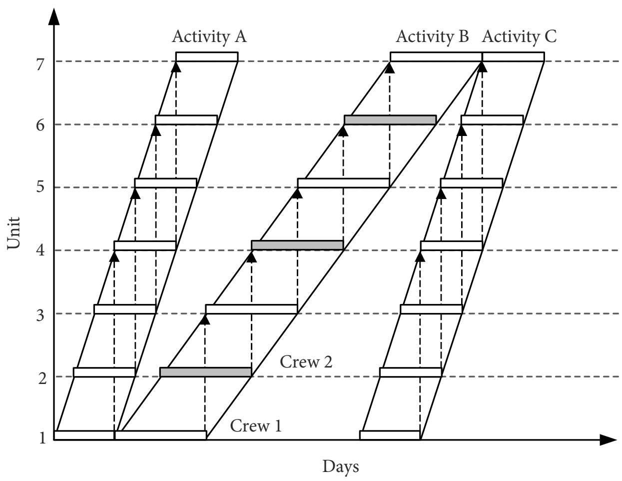

Figure 2. LOB representation

aims at improving the performance of some project objectives (Ammar, 2022). Figure 4 shows a schedule that is interrupted in Activity B at unit *j*. The schedule benefits from introducing interruptions by allowing earlier units of Activity B (the bottom part of B in Figure 4) to start earlier than the planned schedule (the dotted part of B). Accordingly, the succeeding Activity C can be scheduled to start at an earlier date as well (Hegazy & Wassef, 2001). It should also be noted that interruptions can reduce the project duration only if they are applied to the activities with a higher progress rate than their preceding and succeeding activities (Ammar, 2022). Otherwise, interruptions will prolong project duration if they are applied to the wrong activities (Agrama, 2014).

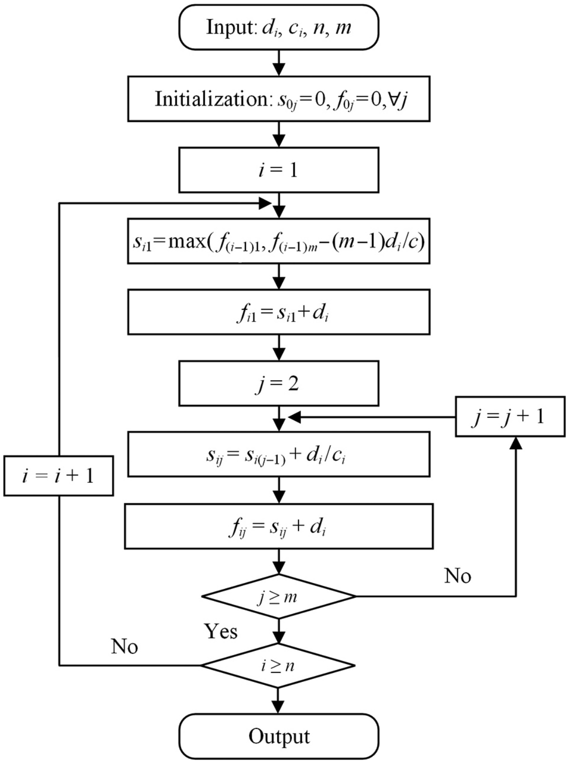

Figure 3. LOB calculation

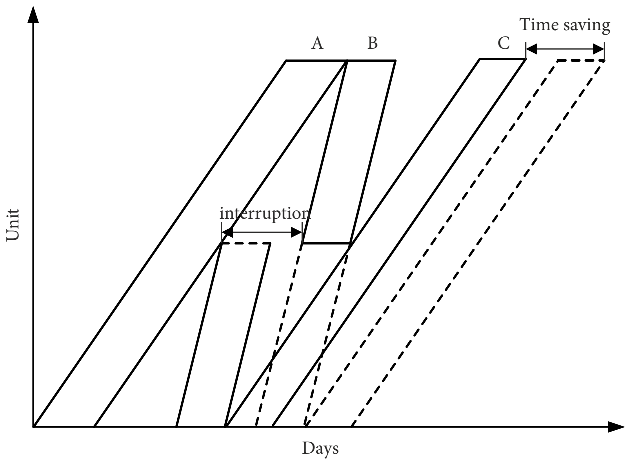

Figure 4. Effect of interruption on project duration (Hegazy & Wassef, 2001)

The above discussion demonstrates that some special interruptions can reduce the project duration of a schedule. Despite the successful application to duration compression and cost-saving, the interruption presented in Figure 4 cannot be applied to smoothing resource fluctuation. In a resource levelling problem, a fixed project duration should be maintained, but the interruptions presented in Figure 4 inevitably lead to changeable project duration. Therefore, an interruption applied to resource levelling should satisfy two conditions: (1) the resource usage fluctuation becomes smoother after the interruption; (2) the project duration does not change. The following of this section proposes two cases that explain which kinds of interruptions can be used for smoothing resource usage fluctuation.

Figure 5 presents the first case that an interruption has no impact on the project duration and can smooth the resource usage fluctuation. As shown in Figure 5a, this case has three activities: Activity A, B and C. The feature of Activity B is that it has a finish-to-start relationship with its preceding and succeeding activity in the last unit. Figure 5b shows that the project duration does not change even if Activity B is interrupted in a certain unit. Figure 5c presents the resource usage curve of this case. It should be noticed that there is a resource usage peak when the three activities are carried out almost simultaneously. In fact, when Activity B's work continuity is interrupted, as shown in Figure 5b, some resources in the peak can be moved to other periods and the resource usage curve presented in Figure 5d becomes smoother compared to Figure 5c. From this case, it can be concluded that the activities share a similar feature with Activity B in Figure 5 can be regarded as candidates for the application of interruption to smooth resource usage fluctuation.

Figure 6 presents the second case that an interruption also has no impact on the project duration and can smooth the resource usage fluctuation. Different from the first case, the feature of Activity B in the second case is that the finish-to-start relationship with its preceding and succeeding activity is preserved at the first unit, as shown in Figure 6a. Figure 6b shows that the project duration does not change when Activity B is interrupted. Figures 5c and 5d demonstrate that the resource usage curve becomes smoother by applying the interruption. The second case indicates that the activities share a similar feature with Activity B in Figure 6 can be regarded as candidates for the application of interruption to smooth resource usage fluctuation.

The above discussion illustrates that some special interruptions can smooth resource usage fluctuation without changing project duration. Concluded from the two cases above, an activity can be selected as a candidate for interruptions if it satisfies one of the two following interruption conditions: (1) the finish-to-start relationship with the preceding and succeeding activities is preserved at the last unit and there are intervals between this activity and its preceding activities; (2) the finish-to-start relationship with the preceding and succeeding activities is preserved

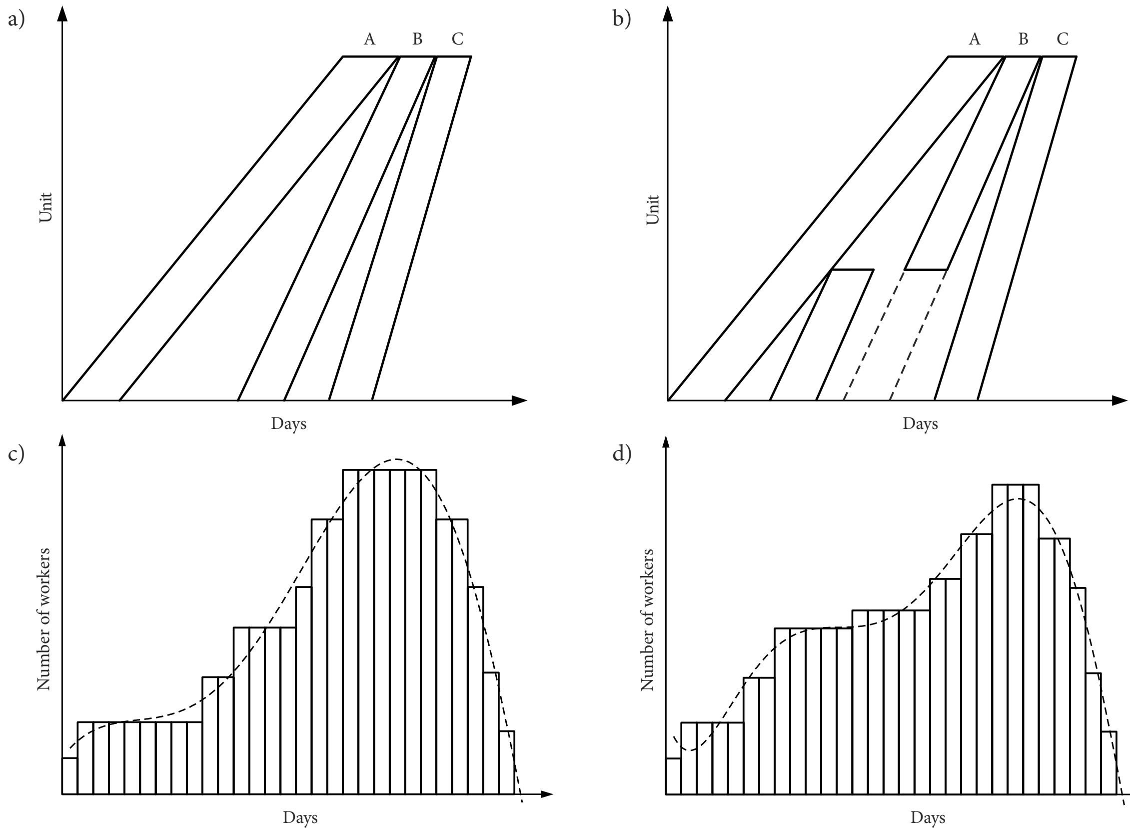

Figure 5. The first case: a – initial LOB schedule; b – LOB schedule with an interruption; c – resource usage curve of the initial LOB schedule; d – resource usage curve of the LOB schedule with an interruption

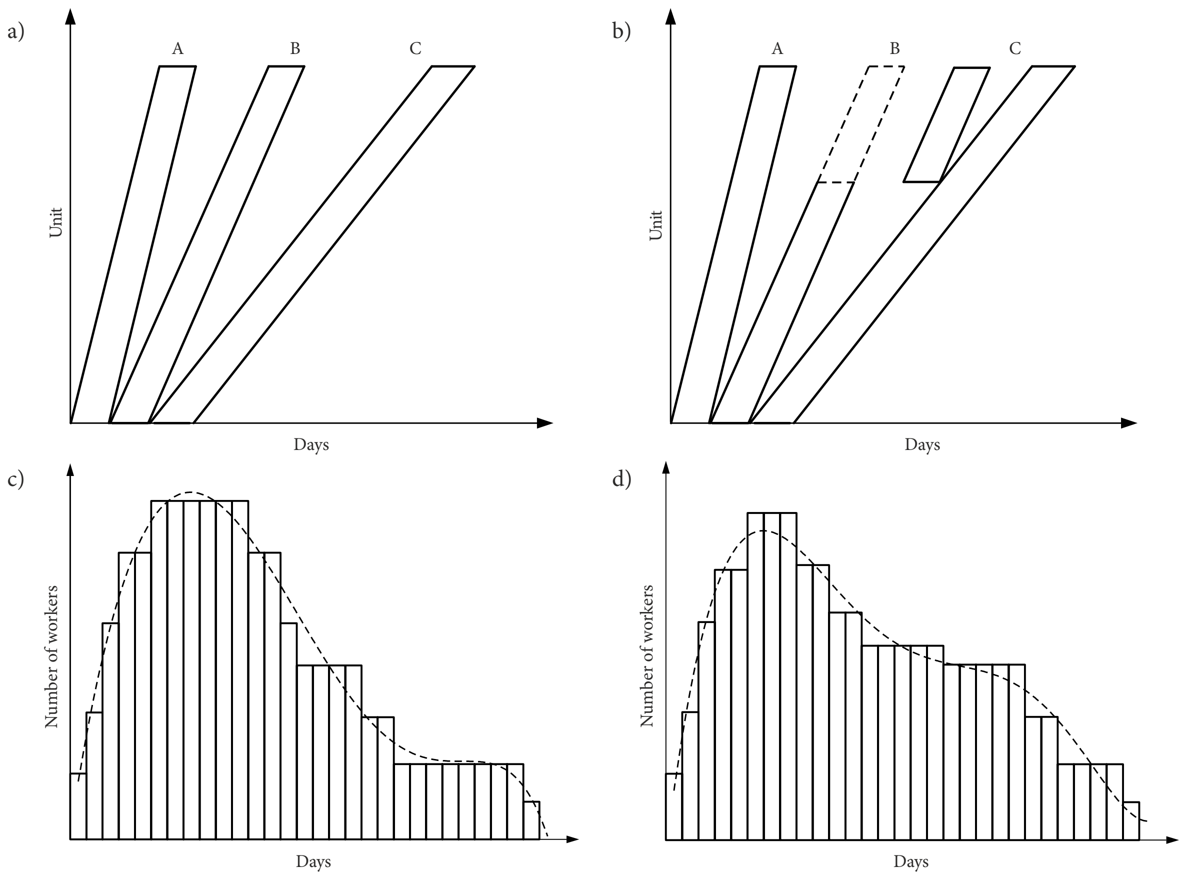

Figure 6. The second case: a – initial LOB schedule; b – LOB schedule with an interruption; c – resource usage curve of the initial LOB schedule; d – resource usage curve of the LOB schedule with an interruption

at the first unit and there are intervals between this activity and its succeeding activities. Project managers can use the two conditions to identify whether an activity can be interrupted to smooth resource usage fluctuation.

3. Model formulation

The objective of the proposed model is to minimize resource usage fluctuation allowing for interruptions. The following constraints are considered in this model: (1) crew work continuity constraints, (2) interruption time limitation constraints, (3) the fixed duration constraint, (4) precedence relation constraints and (5) resource availability constraints. The model requires as input the data of a repetitive construction project, including typical activities, precedence relationships, unit duration of activities and fixed duration. The challenge of the model is to decide the number of crews and interruption values for each activity to minimize resource usage fluctuation value. Therefore, the following decision variables are used: (1) c_i is the number of crews allocated to Activity i, (2) I_{ij} is the interruption time of activity i at unit j and (3) x_{ij} is a 0–1 decision variable representing whether Activity i is interrupted at unit j.

3.1. Objective function

The object of resource levelling problems is to minimize the resource usage fluctuation of a repetitive construction project. In this paper, minimizing the sum of the absolute deviations between daily resource usage and the average resource usage is used as the objective function to measure the resource usage fluctuation. This is one of the most commonly used objective functions for resource levelling problems (Damci et al., 2013a). The mathematical expression of this object is presented as follows:

$$\min \sum_{i=1}^{T} \left| R_i - Ave \right|, \tag{5}$$

where T is the fixed duration of a repetitive construction project; R_i is the resource usage on day i; and Ave is the average resource usage.

3.2. Constraints

Traditional resource levelling models assume that the work continuity of a repetitive construction project is maintained. In this paper, interruptions are tolerated in a schedule to obtain a better resource levelling object function performance. To avoid too much idle time, this paper assumes that an activity can be interrupted only once. Then, the original work continuity constraint Eqn (2) can be modified to a new constraint which is presented by Eqn (6):

$$s_{ij} = s_{i(j-1)} + d_i / c_i + I_{ij} x_{ij}. \tag{6}$$

The interruption time should vary within a certain range. By setting an upper bound and a lower bound, the variation range is described as follows:

$$x_{ij} I_{\min} \le I_{ij} \le x_{ij} I_{\max}, \tag{7}$$

where $I_{\min}$ and $I_{\max}$ denotes the minimum and maximum days of an interruption. Normally, the value of $I_{\min}$ is 0.

In a resource levelling problem, a repetitive construction project needs to be completed within a fixed duration. Because the project duration of a repetitive construction project can be evaluated by the finish time of the last activity (Activity n) at the last unit (unit m), the fixed duration constraint is described as follows:

$$f_{nm} \le T. \tag{8}$$

Precedence relation constraints between each pair of activities should be maintained at all units. If Activity j is the succeeding activity of Activity i, the start time of Activity j must be greater than or equal to the finish time of Activity i at all units. Then, the precedence relation constraints are described as follows:

$$s_{ik} + d_i \le s_{jk}. \tag{9}$$

The resource availability requires that the resource of each activity is limited. The number of crews employed by each activity must not exceed the maximal number of available crews. The resource availability constraints ae described as follows:

$$c_i \le \overline{c_i}, \tag{10}$$

where $\overline{c_i}$ is the crew availability limit of Activity i.

4. Two-stage scheduling algorithm

The proposed resource levelling model involves many factors that need to be decided by project managers, including the number of employed crews, the start time of each activity, the location of interrupted units and the days of each interruption. It is difficult for project managers to only decide on these factors based on their experience. This paper develops a two-stage scheduling algorithm that can automatically establish a schedule with smooth resource usage. The first stage algorithm generates an initial schedule with uninterrupted work continuity by deciding the resource usage allocated to each activity. This schedule will be used as the input of the second stage algorithm. The second stage algorithm further smooths resource usage fluctuation by tolerating interruptions. The specific scheduling procedure of each stage is organized by the genetic algorithm (GA), which can provide a high-quality solution to an optimization model. The proposed two-stage algorithm can significantly improve the performance of the resource levelling object by integrating changing resource usage and tolerating interruptions.

4.1. First stage algorithm

Figure 7 presents the GA-based scheduling algorithm for the first stage. The basic to GA algorithm is the definition of chromosomes to represent different solutions of the resource levelling model. In the first stage algorithm, a chromosome (C) consisting of many genes is defined as an array, and each element of the array represents the number of crews employed by an activity. For example, an array [2, 4] represents a chromosome containing two activities. The first activity employs two crews and the second activity employs four crews. The value of each gene is an integer ranged between one crew (the minimum number of crews) and the crew availability limit ($\overline{c}_i$). Once a suitable representation of chromosomes is decided, an initial population composed of N_1 chromosomes is generated randomly to serve as the starting point for GA.

The next step is translating chromosomes to feasible schedules. The LOB calculation procedure mentioned above forms a one-to-one match between chromosomes and schedules. When a chromosome is determined, a schedule with uninterrupted work continuity corresponding to this chromosome can be generated by using the LOB calculation. The "LOB calculation" section has specified the procedure for generating a LOB schedule.

It is a key step to determine the fitness function of chromosomes. The fitness function is defined on chromosomes and its value reflects the performance of a chromosome. Generally, a chromosome with a better fitness function value means that the corresponding schedule is also better. Based on the resource levelling model, the sum of the absolute deviations between daily resource usage and the average resource usage is used as the fitness function and is presented as follows:

$$f(C) = \sum_{i=1}^{T} \left| R_i - Ave \right|. \tag{11}$$

Then, three genetic operators: selection, crossover and mutation, are performed to produce children. The three operators provide an evolutionary mechanism that adopts the survival of the fittest principle to generate schedules with better performance of the value of the fitness function. The selection operator randomly selects parent chromosomes from the population based on the roulette selection principle. This principle states that a chromosome with a better fitness function value is more likely to be selected with a higher probability. The crossover operator specifies how GA combines two parents to form a new child for the next generation. The single point crossover that exchanges one gene between different chromosomes at the same location is used in the crossover operator. The mutation operator makes small random changes to the chromosomes in the population to create mutation children. The uniform mutation is used in the mutation operator, which a fraction of the chromosome for mutation and replaces each selected gene by a random number selected uniformly from the range of genes.

The above algorithm needs to be carried out repeatedly until one of the two following termination conditions is satisfied: (1) the best value of fitness function cannot be improved by carrying out the algorithm multiple times; (2) the algorithm execution times reaches the maximum. The output of the first stage is an uninterrupted schedule that achieves the optimal resource usage fluctuation.

4.2. Second stage algorithm

The operation flowchart of the second stage algorithm is presented in Figure 8. The first stage provides a schedule with minimum resource usage fluctuation when the work continuity is maintained, and this schedule is used as the input for the second stage algorithm. Based on the

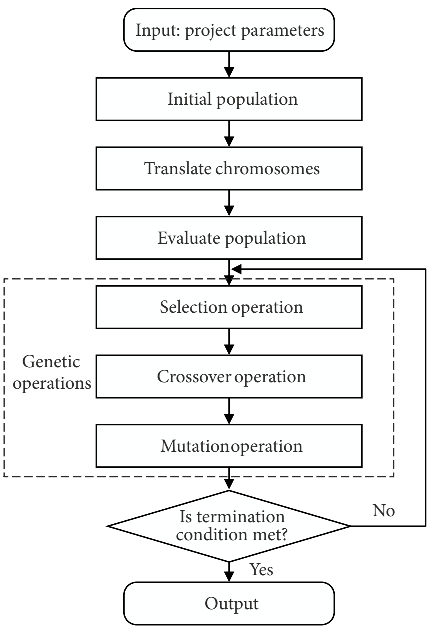

Figure 7. The first stage algorithm

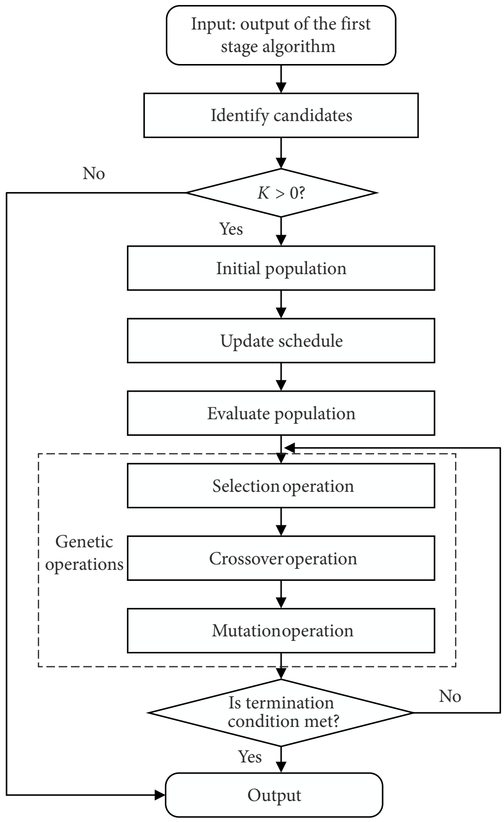

Figure 8. The second stage algorithm

interruption conditions proposed in the "Interruption analysis" section, the second stage algorithm needs to identify which activities can be selected as candidates for interruptions. The number of candidates is defined as K and all candidates constitute a set $A = \{a_1, \ldots, a_K\}$ where a_i demonstrates that Activity a_i is selected as a candidate. If there are some activities that qualify for the interruption conditions ($K > 0$), they will be used in the following algorithm. Otherwise, the second stage algorithm stops and no interruption can be applied to smooth resource usage fluctuation.

In the second stage algorithm, the chromosome is composed of two arrays: (1) the interrupted unit of each candidate, defined as U; (2) the days of each interruption, defined as I. For example, assuming Activities 2 and 4 are selected as two candidates, a chromosome [(10,8), (5,6)] means that an interruption is applied to the first candidate (Activity 2) at unit 10 and lasts for five days, and the other interruption is applied to the second candidate (Activity 4) at unit 8 and lasts for six days. Then, an initial population composed of N_2 chromosomes is generated randomly to serve as the starting point. Considering the precedence relationships, the days of each interruption should not exceed the interval between adjoint activities at the same unit.

In the second stage, a one-to-one match between chromosomes and schedules is also need to be formed. After a chromosome is determined, a new schedule should be generated by inserting interruptions to the schedule provided by the first stage algorithm. Figure 9 presents the updating procedure which can achieve this goal. When an interruption is applied to a candidate, assuming Activity a_k, the original start time $s_{a_k j}$ needs to be updated when unit j is constructed behind the interrupted unit u_k. The new start time $s'_{a_k j}$ is equal to the original start time $s_{a_k j}$ plus the days of interruption I_k.

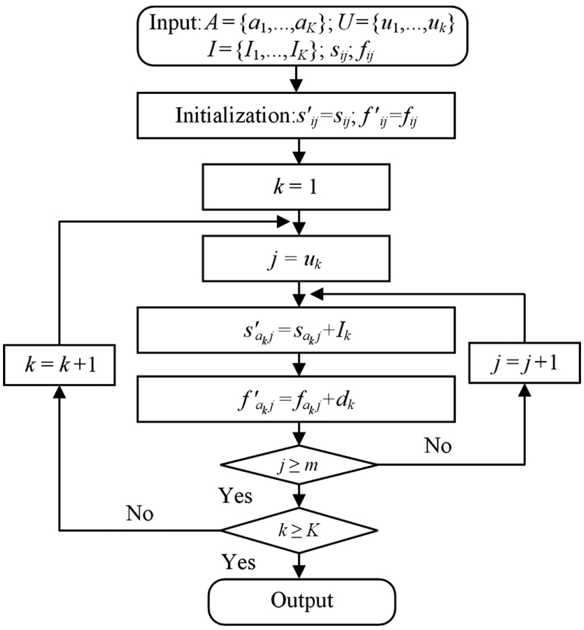

Figure 9. Update schedule

Then, three genetic operators: selection, crossover and mutation, are performed to produce children. The roulette selection principle, single point crossover and uniform mutation are also applied to the three genetic operators. Different from the first stage, the crossover operator of the second stage algorithm may produce infeasible children because the days of interruptions may exceed its limitations. Therefore, the second stage algorithm adopts a modification strategy: if the days of some interruption exceed the interval between the corresponding adjoint activities at the same unit, then replace it with the interval to make children feasible.

The same as the first stage algorithm, termination conditions are as follows: (1) the best value of fitness function cannot be improved by carrying out the algorithm multiple times; (2) the algorithm execution times repeats reaches the maximum. When a termination condition is reached, the second stage algorithm stops and the optimal result are obtained. Otherwise, the algorithm continues.

5. Case study

This section first adopts a shared instance of a pipeline project to illustrate the steps of the proposed approach. This instance was first introduced by Damci et al. (2013a) and was widely used as a benchmark by many resource levelling studies under LOB (Damci et al., 2013a, 2013b; Zhang et al., 2017; Ammar, 2020). Then, this instance is further used to demonstrate the effectiveness and superiority of the proposed approach through the comparison with the results given by Damci et al. (2013a) and Ammar (2020). Finally, a large-scale project presented by Dolabi et al. (2014) is adopted to verify the usability of the proposed approach.

5.1. Model implementation

The pipeline project presented by Damci et al. (2013a) consists of seven consecutive activities and has a length of 26 km. These seven activities are locating and clearing (Activity A), excavating (Activity B), laying aggregate (Activity C), laying pipes (Activity D), testing (Activity E), backfilling (Activity F) and compacting (Activity G). Each kilometre of this pipeline project is regarded as a unit, and each activity needs to be repeatedly constructed from one unit to the other throughout this project. The data of this example project is shown in Table 1, including worker hours, crew sizes, daily working hours, and crews' availability limits.

The data presented in Table 1 is used as the inputs of the first stage algorithm. In the specific calculation process, the chromosome consists of seven genes, and each gene represents the number of crews employed by each activity. Then, the population was modified by using selection, crossover, and mutation operations. Setting the maximum execution times as 250 and the initial population scale as 150, the optimal resource usage allocation and the corresponding schedule are obtained by using the first stage algorithm on a personal computer. Table 2 shows the results of the number of crews and the corresponding schedule. Table 2 indicates that the schedule satisfies the constraints of the 65 days fixed duration. The corresponding resource levelling objective function value is calculated to be 657.33. Figure 10 shows the corresponding LOB diagram of the first stage algorithm result.

The second stage algorithm uses the schedule provided by the first stage algorithm as the input and further smooths its resource usage fluctuation by applying interruptions. The first thing of the second stage algorithm is to identify which activity satisfies the interruption conditions. Based on the data in Table 2, Activity C is selected as the candidate for interruption because this activity satisfies the interruption condition (2) that the finish-to-start relationship of Activity C with its preceding activity (Activity B) and succeeding activity (Activity D) is preserved at the first unit and there are intervals between Activity C and its succeeding activity (Activity D). Therefore, the second stage algorithm contains two decision variables: the interrupted unit of Activity C and the interruption days. In the specific calculation process, the maximum execution times is set as 250, and the initial population scale is set as 150. The population was improved by using the selection, crossover, and mutation operations of the second stage algorithm. The result shows that an interruption is applied to Activity C at unit 16 with five days. The corresponding resource levelling objective function value is reduced from 657.33 to 609. Figure 11 shows the corresponding LOB diagram of the second stage algorithm result.

Table 1. Information about the pipeline project

Activity name	Required worker hours to finish unit	Number of workers	Daily working hours	Max. no. of crews	Duration (days)
(A) Locating and clearing	96	6	8	2	2.0
(B) Excavating	64	8	8	2	1.0
(C) Laying aggregate	80	10	8	3	1.0
(D) Laying pipes	84	7	8	2	1.5
(E) Testing	80	10	8	4	1.0
(F) Backfilling	96	6	8	5	2.0
(G) Compacting	144	9	8	2	2.0

Table 2. First stage algorithm results

Activity name	Number of crews	Start time of the first unit	End time of the first unit	Start time of the last unit	End time of the last unit
(A) Locating and clearing	2	0	2	25	27
(B) Excavating	1	2	3	27	28
(C) Laying aggregate	1	3	4	28	29
(D) Laying pipes	1	4	5.5	41.5	43
(E) Testing	2	34.67	35.67	43	44
(F) Backfilling	2	35.67	37.67	60.67	62.67
(G) Compacting	2	37.67	39.67	62.67	64.67

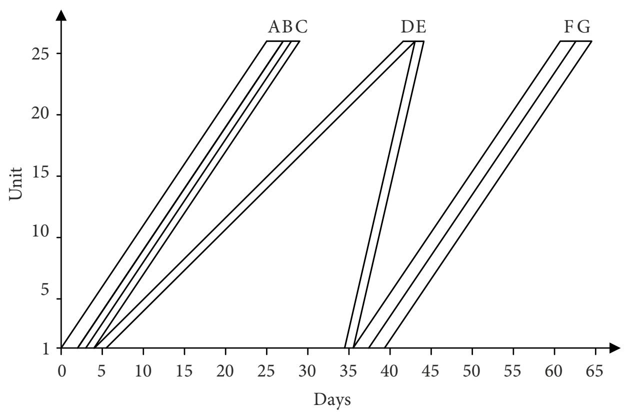

Figure 10. First stage algorithm result

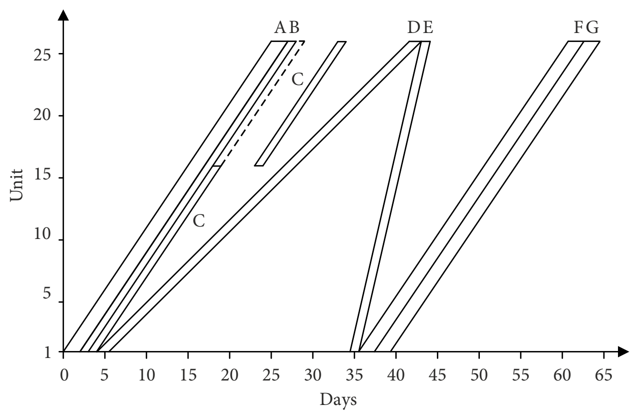

Figure 11. Second stage algorithm result

5.2. Case comparison

Damci et al. (2013a) proposed a scheduling algorithm to smooth resource usage fluctuation by changing the resource usage in some special activities. Damci et al. (2013a) used this algorithm to solve the above pipeline project and generated a schedule of which the resource levelling function value is 1037. In this schedule, the number of crews employed by each activity is [2,2,3,2,4,5,2] and is lower at some units. The proposed two-stage algorithm provided a different schedule and has been presented in subsection 5.1. After calculating the objective function value and indexes, the comparison between the results of Damci et al. (2013a) and this paper is summarized in Table 3. It can be seen that the resource levelling result is improved twice. In the first stage, the total deviation is reduced from 1037 to 657 because the proposed algorithm finds a better resource allocation to each activity. In the second stage, the total deviation is further reduced from 656 to 607 by applying an interruption.

Figure 12 presents the resource usage histogram of Damci et al. (2013a) and this paper. As shown in Figure 12a, the resource usage histogram of Damci et al. (2013a) has two resource usage peaks leading to an unsmooth resource usage curve. By applying the proposed two-stage algorithm, the resource usage histogram of the result only has one resource usage peak, and the maximum value of resource usage is lower, as shown in Figure 12b. This comparison proves the effectiveness and superiority of the proposed method and shows that the integrated approach can achieve a smooth resource curve to the most extent.

Ammar (2020) tried to obtain optimal resource allocation by modelling the resource levelling problem under LOB as an optimization problem. Using the same pipeline project, Ammar (2020) improves the result of Damci et al. (2013a) and provides a new schedule with a shorter project duration (48 days) and smoother resource usage fluctuation (592). In this schedule, the number of crews employed by each activity is [2,1,1,1,1,2,2]. By setting the project duration as 48 days, the proposed two-stage algorithm is used to solve the new case. The first stage algorithm provides the same resource allocation, and the second stage algorithm applies two interruptions to this project: Activity C is interrupted at unit 19 with nine days, and Activity E is interrupted at unit 10 with eight days.

After calculating the objective function value and indexes, Table 4 shows the comparison between the results of Ammar (2020) and this paper. As shown in Table 4, all the indexes of the first stage result are the same as that given by Ammar (2020). This is because the two approaches generate the same resource allocation schedule. However, after executing the second stage algorithm, two interruptions are applied and the performance of total deviation and the maximal number of workers both become better.

Figure 13 presents the resource usage histogram of Ammar (2020) and this paper. Although Ammar (2020) found a schedule with better resource levelling performance than Damci et al. (2013a), the resource usage histogram of the result still has one resource usage peak, as shown in Figure 13a. Through the application of interruptions, the resource usage peak is almost entirely cut

Table 3. Comparison between Damci et al. (2013a) and this paper

	Damci et al. (2013a)	This paper		Improvement	
		First stage	Second Stage	First stage	Second stage
Project duration (days)	65	65	65	–	–
Total number of workers	2093	2093	2093	–	–
Max. number of workers	89	67	67	28.09%	28.09%
Total deviation from avg	1037	657	609	36.64%	41.27%

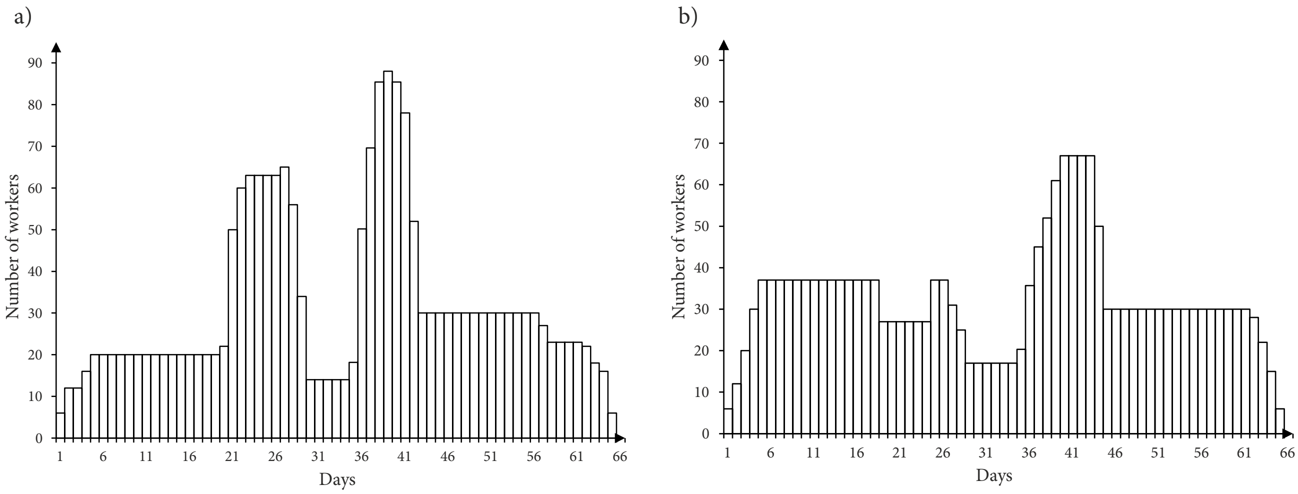

Figure 12. The first comparison: a – resource usage histogram of Damci et al. (2013a); b – resource usage histogram of the proposed method

and the resource usage histogram becomes smoother, as shown in Figure 13b. This comparison proves the effectiveness and superiority of the proposed method again and shows that the application of interruptions can further smooth the resource usage fluctuation.

5.3. Usability verification

To verify the usability of the proposed approach, a large-scale project presented by Dolabi et al. (2014) is adopted. This project is a highway with more activities and a longer project duration compared to that in the previous case studies. The highway project consists of 24 activities with no buffers and is divided into 10 repetitive units. The original schedule prescribed a project duration of 406 days. Detailed information about the project and original schedule can be found in Dolabi et al. (2014), such as activities description, crew sizes and unit durations.

By calculating the original schedule's resource usage, the total deviation and the maximum number of works are 5182 and 64. Then, the proposed two-stage algorithm is used to solve the large-scale project. In order to ensure comparability, the project duration is also set as 406 days.

Table 4. Comparison between Ammar (2020) and this paper

	Ammar (2020)	This paper		Improvement	
		First stage	Second Stage	First stage	Second stage
Project duration (days)	48	48	48	–	–
Total number of workers	2093	2093	2093	–	–
Max. number of workers	77	77	67	0%	12.99%
Total deviation from avg	592	592	479	0%	19.09%

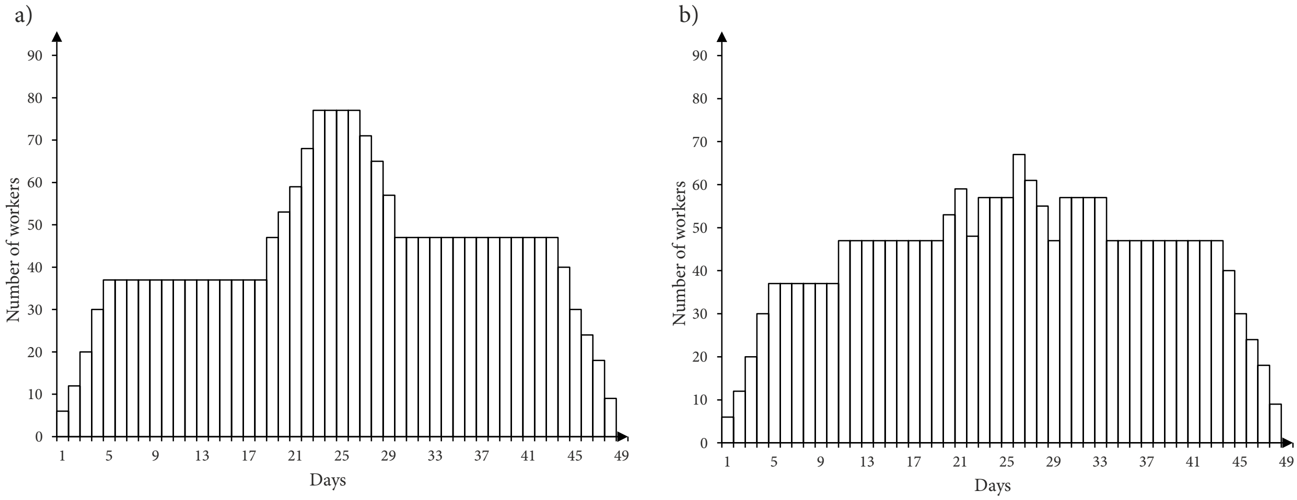

Figure 13. The second comparison: a – resource usage histogram of Ammar (2020);
b – resource usage histogram of the proposed method

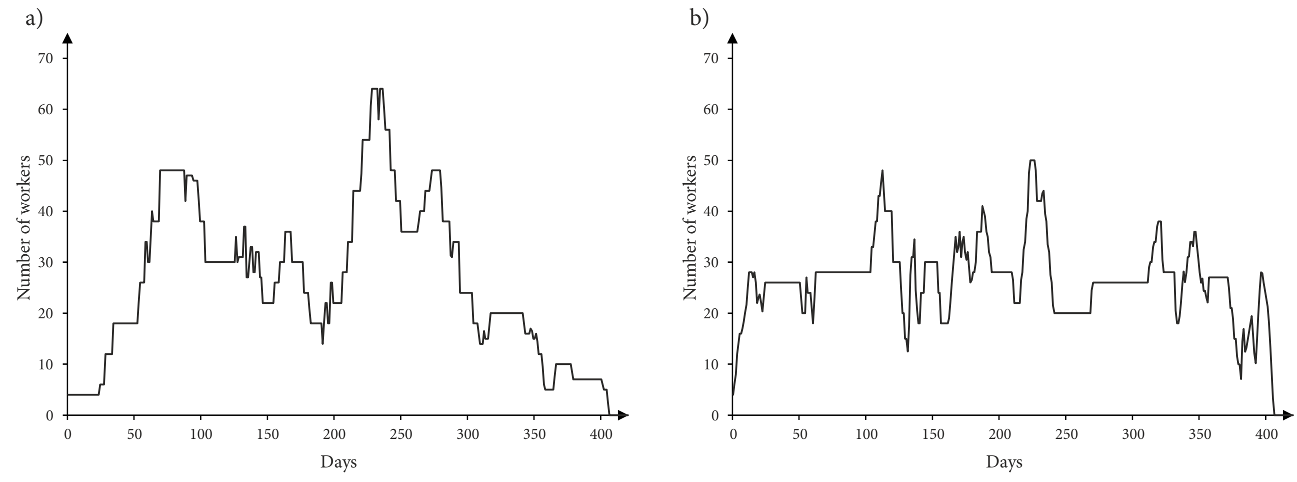

Figure 14. The second comparison: a – resource usage cure of Dolabi et al. (2014);
b – resource usage cure of the proposed approach

Given the large scale of this case, the maximum execution times is set as 500 and the initial population scale is set as 300. While satisfying constraints and maintaining the fixed project duration, the proposed approach can also solve this project in several minutes and provides a schedule with smoother resource usage. The result shows that the total deviation is 1,797 and the maximum number of works is 50. Both of the resource levelling indexes are significantly reduced. Figure 14 compares the resource usage curves of the original schedule and the schedule provided by the proposed approach. The resource usage curve of the proposed method is much smoother compared to the original schedule, thereby indicating its usability and superiority.

Conclusions

Resource levelling plays a vital role in successfully implementing a repetitive construction project. Frequent resource usage changes in repetitive construction projects require that project managers should have various scheduling adjusting processes to smooth resource usage fluctuation. However, changing resource usage is currently the only scheduling adjusting process that project managers can utilize to smooth resource usage fluctuation. Given the flexibility interruptions provide, this paper analyzed the impact of interruptions on resource usage and proved that two kinds of interruptions could be used to smooth resource usage fluctuation. In order to achieve a better resource levelling performance, this paper developed a scheduling approach that integrates the two scheduling adjusting processes: changing resource usage and tolerating interruptions. The effectiveness of the proposed approach was verified by a shared instance of a pipeline project. The comparison shows that the results of the proposed approach are superior to those of previous studies, and the integration of the two scheduling adjusting processes can significantly improve resource levelling results. The proposed approach is still effective even when solving a large-scale project.

Although the proposed approach achieves a significant improvement in resource levelling results, this paper still has some limits to be perfected. As stated in some previous studies, interrupting the work continuity may increase the management cost due to the idle time of resources. Consideration of the trade-off between resource usage fluctuation and interruption costs is a way worthy of exploration in future works.

Acknowledgements

The author would like to thank the referees for their constructive criticism on an earlier version of this paper.

Funding

This work was supported by the National Natural Science Foundation of China under Grant 71971010.

Disclosure statement

Authors have not any competing financial, professional, or personal interests from other parties.

References

Agrama, F. A. (2014). Multi-objective genetic optimization for scheduling a multi-storey building. *Automation in Construction*, *44*(8), 119–128. https://doi.org/10.1016/j.autcon.2014.04.005

Altuwaim, A., & El-Rayes, K. (2018a). Minimizing duration and work interruptions of repetitive construction projects. *Automation in Construction*, *88*, 59–72. https://doi.org/10.1016/j.autcon.2017.12.024

Altuwaim, A., & El-Rayes, K. (2018b). Optimizing the scheduling of repetitive construction to minimize interruption cost. *Journal of Construction Engineering and Management*, *144*(7), 04018051. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001510

Amini, S. M., & Heravi, G. (2009). Schedule compression for construction projects by interruption in LOB scheduling. *AACE International Transactions*, *4*, 1–18.

Ammar, M. A. (2013). LOB and CPM integrated method for scheduling repetitive construction projects. *Journal of Construction Engineering and Management*, *139*(1), 44–50. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000569

Ammar, M. A. (2020). Resource optimisation in line of balance scheduling. *Construction Management and Economics*, *38*(8), 715–725. https://doi.org/10.1080/01446193.2019.1606924

Ammar, M. A. (2022). Optimization of line of balance scheduling considering work interruption. *International Journal of Construction Management*, *22*(2), 305–316. https://doi.org/10.1080/15623599.2019.1624003

Arabpour, R. M., & Moselhi, O. (2021). Optimized crew selection for scheduling of repetitive projects. *Engineering, Construction and Architectural Management*, *28*(6), 1517–1540. https://doi.org/10.1108/ECAM-10-2019-0590

Arditi, D., Tokdemir, O. B., & Sun, K. (2001). Scheduling system for repetitive unit construction using line-of-balance technology. *Engineering, Construction and Architectural Management*, *8*(2), 90–103. https://doi.org/10.1108/eb021173

Bragadin, M. (2010, May). Heuristic repetitive activity scheduling process for networking techniques. In *TG65 & W065-Special Track 18th CIB World Building Congress* (pp. 331–342), Salford, United Kingdom.

Cheng, M. Y., Tran, D., & Hoang, N. D. (2017). Fuzzy clustering chaotic-based differential evolution for resource leveling in construction projects. *Journal of Civil Engineering and Management*, *23*(1), 113–124. https://doi.org/10.3846/13923730.2014.982699

Cho, K., Hong, T., & Hyun, C. T. (2013). Space zoning concept-based scheduling model for repetitive construction process. *Journal of Civil Engineering and Management*, *19*(3), 409–421. https://doi.org/10.3846/13923730.2012.757561

Damci, A., & Polat, G. (2014). Impacts of different objective functions on resource leveling in construction projects: A case study. *Journal of Civil Engineering and Management*, *20*(4), 537–547. https://doi.org/10.3846/13923730.2013.801909

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

Damci, A., Arditi, D., & Polat, G. (2013b). Multiresource 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. (2016). Impacts of different objective functions on resource leveling in line-of-balance scheduling. *KSCE Journal of Civil Engineering*, *20*(1), 58–67. https://doi.org/10.1007/s12205-015-0578-7

Dolabi, H. R. Z., Afshar, A., & Abbasnia, R. (2014). CPM/LOB scheduling method for project deadline constraint satisfaction. *Automation in Construction*, *48*, 107–118. https://doi.org/10.1016/j.autcon.2014.09.003

Eid, M., Elbeltagi, E., & El-Adaway, I. (2021). Simultaneous multicriteria optimization for scheduling linear infrastructure projects. *International Journal of Construction Management*, *21*(1), 41–55. https://doi.org/10.1080/15623599.2018.1505027

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)

Georgy, M. E. (2008). Evolutionary resource scheduler for linear projects. *Automation in Construction*, *17*(5), 573–583. https://doi.org/10.1016/j.autcon.2007.10.005

Harmelink, D. J., & Rowings, J. E. (1998). Linear scheduling model: Development of controlling activity path. *Journal of Construction Engineering and Management*, *124*(4), 263–268. https://doi.org/10.1061/(ASCE)0733-9364(1998)124:4(263)

Hassan, A., El-Rayes, K., & Attalla, M. (2021). Optimizing the scheduling of crew deployments in repetitive construction projects under uncertainty. *Engineering, Construction and Architectural Management*, *28*(6), 1615–1634. https://doi.org/10.1108/ECAM-05-2020-0304

Hegazy, T., & Wassef, N. (2001). Cost optimization in projects with repetitive nonserial 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., Mostafa, K., & Ojulari, S. (2021). Tetris-inspired approach for generating tightly-packed repetitive schedules. *Automation in Construction*, *124*, 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)

Jaskowski, P., & Biruk, S. (2018). Reducing renewable resource demand fluctuation using soft precedence relations in project scheduling. *Journal of Civil Engineering and Management*, *24*(4), 355–363. https://doi.org/10.3846/jcem.2018.3043

Jaskowski, P., & Biruk, S. (2020). Scheduling of repetitive construction processes with concurrent work of similarly specialized crews. *Journal of Civil Engineering and Management*, *26*(6), 579–589. https://doi.org/10.3846/jcem.2020.12914

Liu, S. S., & Wang, C. J. (2007). Optimization model for resource assignment problems of linear construction projects. *Automation in Construction*, *16*(4), 460–473. https://doi.org/10.1016/j.autcon.2006.08.004

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

Lucko, G. (2008). Productivity scheduling method compared to linear and repetitive construction project scheduling methods. *Journal of Construction Engineering and Management*, *134*(9), 711–720. https://doi.org/10.1061/(ASCE)0733-9364(2008)134:9(711)

Lucko, G. (2011). Integrating efficient resource optimization and linear schedule analysis with singularity functions. *Journal of Construction Engineering and Management*, *137*(1), 45–55. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000244

Mattila, K. G., & Abraham, D. M. (1998). Resource leveling of linear schedules using integer linear programming. *Journal of Construction Engineering and Management*, *124*(3), 232–244. https://doi.org/10.1061/(ASCE)0733-9364(1998)124:3(232)

Podolski, M. (2017). Management of resources in multiunit construction projects with the use of a tabu search algorithm. *Journal of Civil Engineering and Management*, *23*(2), 263–272. https://doi.org/10.3846/13923730.2015.1073616

Podolski, M., & Sroka, B. (2019). Cost optimization of multiunit construction projects using linear programming and metaheuristic-based simulated annealing algorithm. *Journal of Civil Engineering and Management*, *25*(8), 848–857. https://doi.org/10.3846/jcem.2019.11308

Rogalska, M., & Hejducki, Z. (2007). Time buffers in construction process scheduling. *Journal of Civil Engineering and Management*, *13*(2), 143–148. https://doi.org/10.1080/13923730.2007.9636430

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

Tang, Y. J., Sun, Q., Liu, R., & Wang, F. (2018a). 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

Tang, Y. J., Liu, R. K., Wang, F., Sun, Q., & Kandil, A. A. (2018b). Scheduling optimization of linear schedule with constraint programming. *Computer-Aided Civil and Infrastructure Engineering*, *33*(2), 124–151. https://doi.org/10.1111/mice.12277

Tran, D. H., Chou, J. S., & Luong, D. L. (2019). Multi-objective symbiotic organisms optimization for making time-cost tradeoffs in repetitive project scheduling problem. *Journal of Civil Engineering and Management*, *25*(4), 322–339. https://doi.org/10.3846/jcem.2019.9681

Zhang, L., Tang, Y., & Qi, J. (2017). Resource leveling based on backward controlling activity in line of balance. *Mathematical Problems in Engineering*, Article ID 7545980. https://doi.org/10.1155/2017/7545980

Zou, X., Zhang, L. H., & Zhang, Q. (2018). A biobjective optimization model for deadline satisfaction in line-of-balance scheduling with work interruptions consideration. *Mathematical Problems in Engineering*, Article ID 6534021. https://doi.org/10.1155/2018/6534021

Zou, X., Wu, G., & Zhang, Q. (2021). Work continuity constraints in repetitive project scheduling considering soft logic. *Engineering, Construction and Architectural Management*, *28*(6), 1713–1738. https://doi.org/10.1108/ECAM-11-2019-0595

Zou, X., Zhang, L., & Zhang, Q. (2022), Time-cost optimization in repetitive project scheduling with limited resources. *Engineering, Construction and Architectural Management*, *29*(2), 669–701. https://doi.org/10.1108/ECAM-10-2020-0843