Journal of Civil Engineering and Management
ISSN 1392-3730 / eISSN 1822-3605
2023 Volume 29 Issue 1: 15–34
https://doi.org/10.3846/jcem.2023.17927

BLOCKCHAIN OF OPTIMAL MULTIPLE CONSTRUCTION PROJECTS PLANNING UNDER PROBABILISTIC ARRIVAL AND STOCHASTIC DURATIONS

Abbas AL-REFAIE[1], Ahmad AL-HAWADI[1], Natalija LEPKOVA[2*], Ghaleb ABBASI[1]

[1]*Department of Industrial Engineering, The University of Jordan, Amman, Jordan*
[2]*Department of Construction Management and Real Estate, Vilnius Gediminas Technical University, Vilnius, Lithuania*

Received 11 February 2022; accepted 13 July 2022

Abstract. With the rapid development of projects, firms are facing challenges in planning and controlling complex multiple construction projects. This research, therefore, aims at developing blockchain of optimal scheduling and sequencing of multiple construction projects under probabilistic arrival and stochastic durations. Each project task was considered as a block. Then, a framework for electronic project recording (EPR) system was developed. The EPRs are records for project tasks that make information available directly and securely to authorized users. In this framework, two optimization models were developed for scheduling and sequencing project blocks. The scheduling model aims to assign project tasks to available resources at minimal total cost and maximal the number of assigned project tasks. On the other hand, the sequencing model seeks to determine the start time of block execution while minimizing delay costs and minimizing the sum of task's start times. The project arrival date and the task's execution duration were assumed probabilistic and stochastic (normally distributed), respectively. The developed EPR system was implemented on a real case study of five projects with total of 121 tasks. Further, the system was developed when the task's execution duration follows the Program Evaluation and Review Technique (PERT) model with four replications. The project costs (idle time and overtime costs) at optimal plan were then compared between the task's execution duration normally distributed and PERT modelled. The results revealed negligible differences between project costs and slight changes in the sequence of project activities. Consequently, both distributions can be used interchangeably to model the task's execution duration. Furthermore, the project costs were also compared between four solution replications and were found very close, which indicates the robustness of model solutions to random generation of task's execution duration at both models. In conclusion, the developed EPR framework including the optimization models provided an effective planning and monitoring of construction projects that can be used to make decisions through project progress and efficient sharing of project resources at minimal idle and overtime costs. Future research considers developing a Blockchain of optimal maintenance planning.

Keywords: blockchain, sequencing, scheduling, optimization, project management.

Introduction

Construction management deals with economical consumption of the resources available in the least possible time for successful completion of construction project. Rapid development in construction industry led to an increasingly competitive market incentives and required a shorter project life cycle, higher completion percentages, lower incurred costs, and resources. Hence, many firms are facing challenges in planning, executing, controlling, and communicating multiple construction projects simultaneously while sharing different resource types. In fact, more than 33 percent of unsuccessful projects failed because of lack planning of project tasks and resources (Project Management Institute [PMI], 2021). Managers of complex construction projects are, therefore, required to have the ability to allocate time and resources efficiently to manage costs and keep those projects on their tracks, as well as to stay well-organized and communicate effectively with project members.

Effective planning and scheduling of multiple construction projects simultaneously by optimally link and

*Corresponding author. E-mail: *natalija.lepkova@vilniustech.lt*

chain project associated tasks are highly desirable but challenging at the same time, particularly when using limited resources and costs. Fortunately, the application of blockchain technology for planning multiple construction projects might have a profound impact on project planning and management. Typically, a construction project consists of number tasks and activities that should be linked or chained logically to be executed as shown in Figure 1, in which each task might be considered as a block on distributed database without chains (Alladi et al., 2019).

Blockchain might be defined as the decentralized technology that is used to register, confirm, allocate, and transfer all manner of data (Swan, 2015). Blockchain provides the opportunity to traceability aid, record management, automation for the supply chain, and other business applications (Christodoulou et al., 2018; Issaoui et al., 2019). In technology circles, the blockchain is used as a permanent distributed directory to record all value activities. Each node participating in the activity would have access to the ledger from multiple devices. Participants initially review all activities connected to the blockchain. Blockchain technology, sometimes known as distributed ledger technology, is a relatively new form of a database for activity information stored in a decentralized and transparent manner (Kawaguchi, 2019; Lee et al., 2019). The database is run by a network of computers called nodes, so there is no single-point-of-failure, and information can be accessed in real-time. On any distributed database, each element is called block. The structure of the block is mainly centered on several components (Singh, 2020; Ullah & Al-Turjman, 2021). The first component (Data) is the application data maintained in the distributed database blocks, where the block has the capacity to hold numerous data units. The second component (Current hash) is functions and values that meet the demands needed to solve for the current block computations. The third component (Previous hash) is values that are used to chain the blocks. The fourth component (Timestamp) describes when the block was added to the chain. Finally, the fifth component contains (Other information) about software used and the complexity level. The blocks are chained together to ensure synchronization between blocks and avoid separating any individual block from the other blocks as also presented in Figure 1.

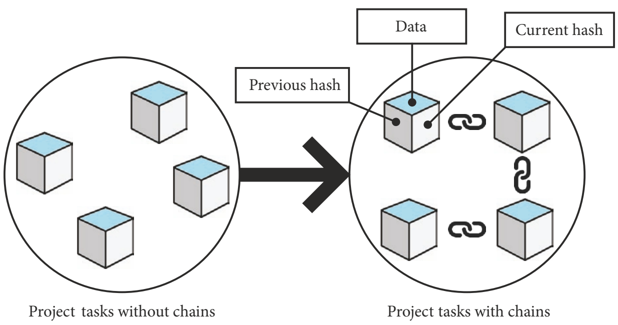

Figure 1. Blockchain of project tasks

For managing multiple construction projects, the move towards EPR systems has been accelerating at high-rate side by side with Internet of Things (IoT) and smart devices due to the significant anticipated advantages to projects managers and decision makers. The EPRs are records for the related tasks in projects that make information available directly and securely to authorized users. This suggested system will be built to go beyond standard approaches for data collection in a project management office and can be inclusive of a wider view for project management. In these regards, this paper proposes an EPR system including two optimization models for the scheduling and sequencing of multiple construction projects under probabilistic project arrival dates and stochastic task durations. The EPR system contains projects' related information, tasks, due dates, durations, resources, and costs, allows access to digital tools that can be used to make decisions through project progress, and streamlines and automate project workflow. This system may support project managers in increasing resources utilization, reducing costs, and providing effective planning and management of project activities. The remaining of this paper is organized as follows. Section one review relevant literature on project scheduling and sequencing. Section two developed the blockchain design and optimization models formulation. Section three implements the developed models on a real case study and discusses research results. Last section summarizes research conclusions.

1. Literature review

Recently, developing optimization models for planning projects has significantly received research attention.

1.1. Resource constrained project planning

Almeida et al. (2016) investigated the multi-skill resource constrained project scheduling problem and proposed the use of a parallel scheduling scheme. Two new concepts were developed: resource weight and activity grouping. A series of computational tests were performed using a large number of instances and the results showed that the proposed heuristic was very effective in finding high quality solutions. Ning et al. (2017) investigated the multi-mode cash flow balanced project scheduling problem with stochastic duration of activities. The objective was to generate a robust baseline schedule to minimize the contractor's maximal cumulative gap between cash outflows and cash inflows. Two metaheuristics, simulated annealing and tabu search were developed to solve the problem. Delgoshaei et al. (2017) developed a new method for modifying over-allocated schedules in multi-mode resource constrained project scheduling problems with positive cash flows. The aim was maximizing net present value or logically minimizing negative cash flows. The proposed method was designed to consider all types of activity precedence. In this research progress payment method and preemptive resources were considered. The proposed approach maximizes negative cash flows by scheduling activities through the resource calendar respecting to the available level of pre-emptive resources and activity numbers. The results showed that the proposed algorithm can provide modified schedules with no over-allocated days for experiment with 1000 activities and 100 preemptive resources in a few seconds. Shu et al. (2018) proposed a mathematical optimization model to solve the resource-constrained multi-project scheduling problem. The genetic algorithm was then designed which used the coding method based on priority of activities, combined with the storage adjacency matrix. Besides, this paper applies this model to the medical resource scheduling. The objective function was to minimize the total durations. The proposed model was applied on a case study in the medical field where the results showed feasible solutions. Quoc et al. (2019) presented an optimization model to solve multi-skill resource constrained scheduling problem. The objective function was built to minimize the time span for projects. They also proposed an evolutionary algorithm based on Cuckoo Search algorithm. The proposed algorithm showed fast convergence and avoided program getting trapped in local extremum. de Melo and de Queiroz (2021) introduced an integer linear programming model to study the resource constrained project planning problem while considering multi skill, many nodes, and time delays constraints. The objective function was to minimize the make span between tasks. The formulations are solved with the default branch-and-cut algorithm of the solver Gurobi Optimizer. Kadri and Boctor (2018) addressed the resource-constrained project scheduling problem with transfer times where pre-emption was not allowed, and precedence relations were zero-lag finish-to-start relations. Also, they assumed that the durations and resource transfer times of activities were known and deterministic. The objective was to choose a start time for each activity of the project so that the project duration was minimized, while satisfying precedence relations, resource availabilities, and resource-transfer time constraints. They proposed a new genetic algorithm using a two-point crossover operator. Delgoshaei et al. (2019) developed a multi-objective nonlinear mixed integer programming model where resource availability was not deterministic and expressed by triangular probability function. In addition, a multi-objective weighting genetic algorithm was proposed. To verify the performance of the proposed method, a number of experiments were solved. The results indicated that while resource uncertainty increases, higher complexity in schedules was observed. Tirkolaee et al. (2019) addressed the multi-objective multi-mode resource-constrained project scheduling problem with payment planning where the activities can be done through one of the possible modes and the objectives were to maximize the net present value and minimize the completion time concurrently. A nonlinear programming model was proposed to formulate the problem based on the suggested assumptions. Two metaheuristics of non-dominated sorting genetic algorithm II and multi-objective simulated annealing algorithm were developed to solve the problem. Finally, the performances of the proposed solution techniques were evaluated using some well-known efficient criteria. Tian et al. (2022) proposed a new extension of the multi-skill resource-constrained project scheduling problem with skill switches. A mixed-integer programming model, which aimed to minimize project completion time and total cost, was developed. Then, a new flexible solution representation scheme with reduced search space and a novel greedy-like schedule builder scheme that reorders tasks to reduce skill switches was proposed.

1.2. Stochastic resource-constrained project scheduling

Stochastic resource-constrained project scheduling has received significant research attention. For example, Ballestin and Leus (2009) developed a GRASP-heuristic to produce high-quality solutions, using so-called descriptive sampling, for resource-constrained project scheduling with stochastic activity durations. Various objective functions related to timely project completion were examined, as well as the correlation between these objectives. Deblaere et al. (2011) proposed a stochastic methodology that was used to determine criteria for project execution policies and predict the project tasks starting times and minimize the incurred costs. In a computational experiment, the results showed that the proposed procedure greatly outperformed existing algorithms. Abello and Michalewicz (2014) investigated the dynamic resource-constrained project scheduling problem where the number of tasks varies in time. the significance of the sub algorithms of mapping of task IDs for centroid-based approach with random immigrants (McBAR) was investigated. The McBAR was compared to a technique called, Estimation Distribution Algorithm. Creemers (2015) developed an exact procedure to optimally solve the stochastic resource-constrained scheduling problem for activity durations of a moderate-to-high level of variability. The procedure aimed to minimizing the expected makespan of a project. The study considered exponential activity durations. Bruni et al. (2015) overviewed the models and methods for the resource-constrained project scheduling under uncertainty. The case of known deterministic renewable resource requirements and random activity durations with a known probability distribution function was studied. Joint probabilistic constraints were used to generate a feasible baseline schedule with minimum makespan. The results obtained with the proposed heuristics were discussed and compared with two well-known heuristics taken from the literature on a set of randomly generated project instances. The heuristics were applied on a real construction project for students' apartments at the University of Calabria, Italy. Ortiz-Pimiento and Diaz-Serna (2018) presented an exhaustive literature review on the project scheduling problem with non-deterministic activities duration. the study aimed at identifying the existing models where the activities duration was taken as uncertain or random parameters. Huang et al. (2020) formulated a stochastic mixed integer programming to optimize tasks schedules that maximizes the net present value of the project and minimizes the related risks. A case study on gold mining was implemented to demonstrate the application of the model. Ulusoy and Hazır (2021) presented the basic scheduling models under uncertainty, considered the variability of activity durations, and addressed how to represent the durations using probability distributions. The characteristics of the Beta distribution were introduced. The Program Evaluation and Review Technique (PERT) was used to minimize the expected project duration. Finally, the PERT-Costing method was considered to minimizing the expected project cost. Monte-Carlo simulation was applied to uncertainty both in activity duration and activity cost. Al-Refaie et al. (2021) proposed optimization models for task scheduling and sequencing in work-intensive multiple projects under normal and unexpected events. The objective functions model was minimizing the total costs and maximizing satisfaction values on tasks due dates and processing standard times. Illustrations of the proposed scheduling and sequencing optimization models were provided where the results showed effective scheduling and sequencing of project tasks at minimal costs and achieved the desired satisfaction levels on tasks and projects. Ansari et al. (2022) proposed a two-stage stochastic and multi-objective programming model to determine the size of the time buffers in engineering and construction projects. The method was compared to the classic and extended critical chain management approaches.

1.3. Blockchain

Blockchain has received significant research attention. For example, Guo and Liang (2016) considered blockchain application and outlook in the banking industry. Muzylyov and Shramenko (2019) applied blockchain technology in transportation as a part of the efficiency in Industry 4.0 strategy. Zhong et al. (2019) studied the smart contract systems by introducing new system to protect the privacy of blockchain and considered the encryption of all data within controllable time horizon. The results showed that the proposed system could effectively protect data and make smart contracts more secure. Zhao et al. (2019) examined blockchain technology in agri-food value chain management. Tanwar et al. (2020) studied blockchain-based electronic healthcare record system for healthcare 4.0 applications. Lohmer and Lasch (2020) identified potential and barriers of blockchain in operations management and manufacturing. Furthermore, Su et al. (2020) proposed new solutions for secure data sharing using blockchain technology. They proposed data sharing model and data sharing protocol. Their model set access control strategies with blockchain platform usage and databases were distributed to be stored together. The analysis revealed the correctness and validation of the proposed scheme. Olawumi et al. (2021) examined the potentiality of integrating blockchain and digital tools and technologies in modular integrated construction projects. Lee et al. (2021) developed and tested an integrated digital twin and blockchain framework for traceable data communication in construction projects. Marinho et al. (2021) performed an analysis of the determining factors in the adoption of relational contracting in combination with Building Information Modelling (BIM) to reduce asymmetric information phenomenon in construction projects in Portugal.

This research contributes to ongoing research by: (a) developing a framework for a blockchain system that can be used for effective chaining the tasks of multiple construction projects, (b) proposing two multi-objective optimization models for scheduling and sequencing of project tasks, and (c) considering stochastic task durations; modeled by a normal and PERT distributions, and probabilistic arrival project dates.

2. EPR system development and optimization modelling

2.1. EPR system development

The EPR lifecycle is depicted in Figure 2 and detailed as follows:

Step 1: Smart contracts are programs that will be activated when certain and predetermined conditions stored in blockchain are met. Smart contracts simply form the relationship between projects' owner and project manager. They are essential for project management through the implementation of blockchain technology. Therefore, multiple construction projects are received via the smart contract, where each project has a specific due date.

Step 2: After signing the smart contract, the time horizon for project execution will be determined. For practical implementation for blockchain technology, probabilistic arrival project dates will be generated to adapt the blockchain to real case scenarios. Typically, the execution of a project should be performed after the arrival date of the project.

Step 3: Project tasks are next broken down and classified based on their dependencies. Each project is composed of a known number of tasks which require specific resources and skill levels. Each firm resource has a known availability and specific skills on project tasks. In this research, each task is treated as a block.

Step 4: Given the block data, scheduling optimization model are formulated and then solved to assign project tasks to the available resources at minimum incurred costs. Then, the assigned schedule will be shared with project managers by cloud for validation.

Step 5: After approval of the optimal scheduling of project tasks, optimization model is developed to sequence project tasks at minimal total of delay costs for project and start execution times. The results are then stored in each block and shared with project members via cloud. Each block is assigned a hash to avoid access to the block information by unauthorized persons.

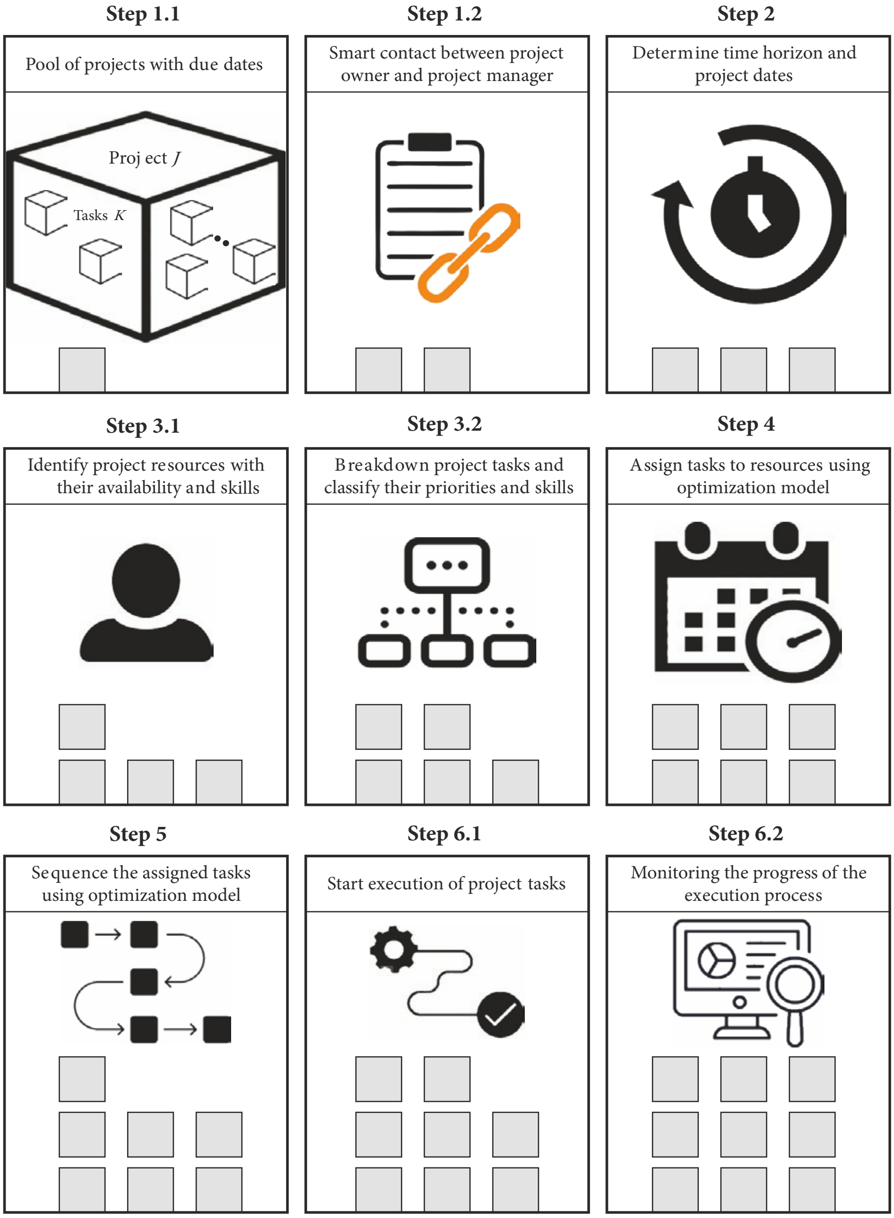

Figure 2. Blockchain implementation in project management

Step 6: Once the blockchain of project tasks (blocks) has been established, the task execution takes place over the predetermined time horizon. The data of completed blocks are then transferred between project blocks to update the time schedule. With the implementation of IoT, the project manager will be able to monitor the progress of the execution of project blocks and take any needed actions. Finally, the completed project will be communicated with finance department and project owner to finalize financial issues.

2.2. Application of ERP system

Steps 1 and 2: Smart contracts and project arrivals

In Steps 1 and 2, assume multiple construction projects have arrived via the smart contract. For more practical and fully reflecting the real project planning conditions, project arrival dates will be treated as probabilistic parameters (Pham et al., 2021). Hence discrete possible outcomes will be associated with different probabilities for project arrival dates. Suppose that there are t possible project arrival dates, where the project arrival has a specific probability in each day as illustrated in Figure 3.

Steps 3 and 4: Project breakdown structure and optimal blocks scheduling

In Step 3, the signed construction project j (j = 1, ..., J) consists of several tasks. Let k be the index for any task; k = 1, ..., K. Initially, it is assumed that the task durations are normally distributions (Hong et al., 2019). Further, the information about available resources and skill levels are known in advance. Each project is then broken down into its main tasks.

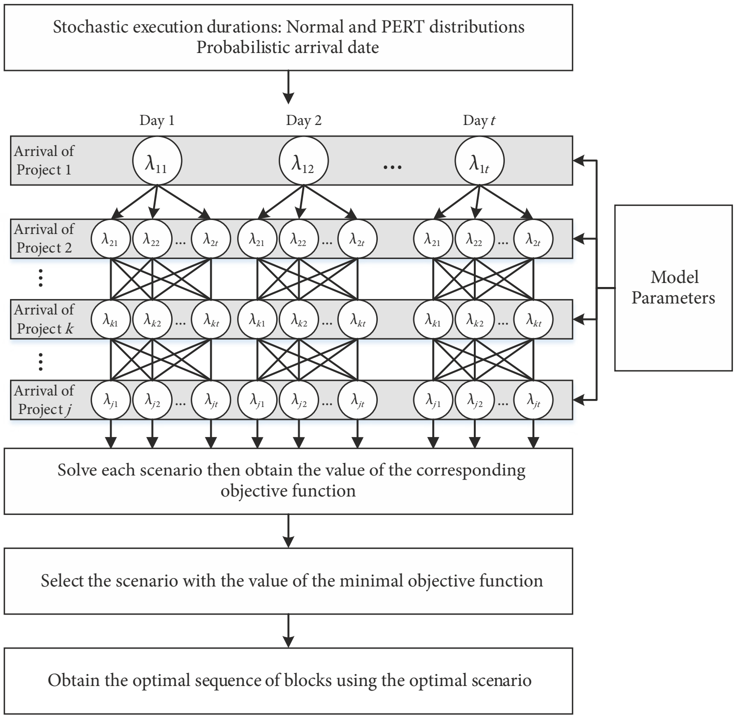

Figure 3. Lingo scenarios for optimal project plan

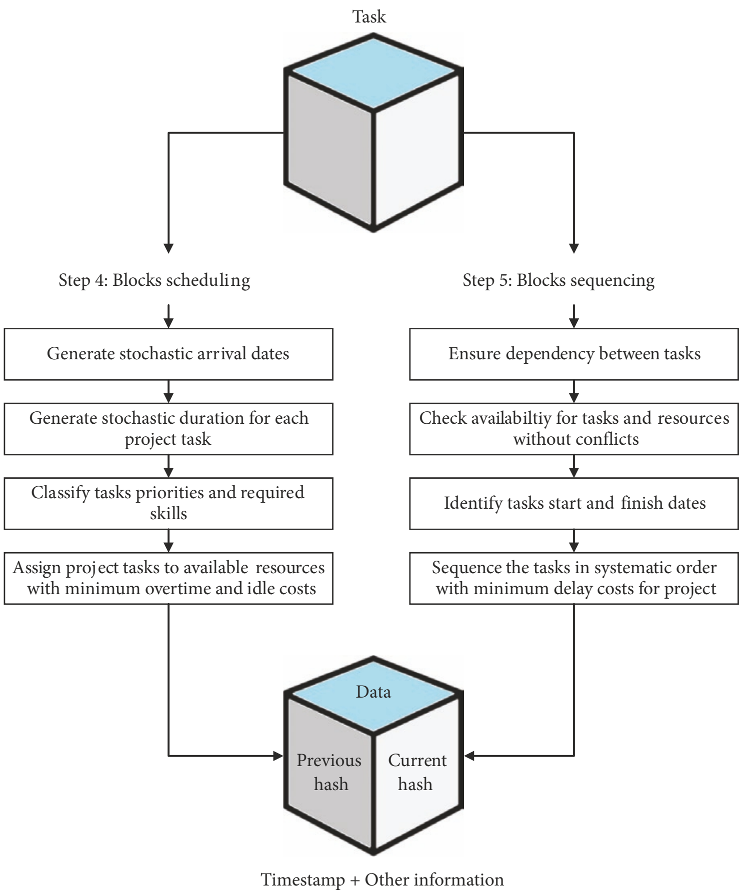

Figure 4. The procedure for scheduling and sequencing of project blocks

Given available block parameters (task duration, the required skill level, priority, and dependency), optimization models for scheduling and sequencing project blocks will be developed as shown in Figure 4.

Optimization model's notations

The notations for the assigning and sequencing optimization models are listed in Table 1.

Table 1. Model's notations

Notation	Description
$TIDT$	The total idle times in hours
$TOVT$	The total overtime in hours
$TDYT$	The total delay times in intervals
CI	The idle time unit cost per idle hour
CO	The overtime unit cost per overtime hour
CD	The delay time unit cost per interval
$IDTC$	The total idle time costs
$OVTC$	The total overtime costs
$DYTC$	The total delay time costs
z_{ij}	The possible dates for project j arrival in interval i
λ_{ij}	The associated probability for project j arrival in interval i
ARR_{ji}	The arrival of project j in interval i
DUE_j	The due date of project j
STD_{jk}	The standard time required to execute task k from project j
ACT_{jk}	The actual time required to execute task k from project j
DUR_{mi}	The spent duration in executing project tasks by member m in interval i
β_m	Member's effectiveness
AVB_{mi}	Binary parameter determines the availability of member m in interval i
REG	Regular working hours
ID_{mi}	The idle time incurred by member m in interval i
OV_{mi}	The overtime incurred by member m in interval i
$MXID$	The maximum allowable idle time
$MXOV$	The maximum allowable overtime
SK_{ml}	Binary parameter determines if member m has skill l
RSK_{jkl}	Binary parameter determines if task k from project j requires skill l
B_{jkmi}	Binary decision variable determines if task k from project j scheduled to member m in interval i
TES_{jkmi}	The execution start time for task k from project j by member m in interval i
TEF_{jkmi}	the execution finish time for task k from project j by member m in interval i
PFT_j	The finish time for project j
$\chi_{(jk),(jk')}$	Binary parameter which determines whether or not task k depends on test k' from project j
Q	Very large number

In step 4, the optimization model for scheduling project blocks will be developed. The objective functions and constraints of the develop scheduling model are formulated as follows.

(a) Objective function: Minimizing incurred costs (idle time and overtime)

Let $TIDT$ denotes the total idle times from all members in all intervals and let CI denotes the cost per idle hour. Then, $IDTC$ is mathematically expressed in Eqn (1):

$$IDTC = TIDT \cdot CI. \quad (1)$$

Let $TOVT$ denotes the total overtime in hours from all members in all intervals and let CO denotes the cost per overtime hour. Then, $OVTC$ is calculated as stated in Eqn (2):

$$OVTC = TOVT \cdot CO. \quad (2)$$

Combining Eqns (1) and (2), the first objective function is to minimize the total costs of idle time and overtime, as shown in Eqn (3):

$$\text{Min } IDTC + OVTC. \quad (3)$$

(b) Objective function: Maximizing the number of assigned project tasks

Let B_{jkmi} be a binary decision variable that determines whether task k from project j is assigned to member m in interval i; where a value of one indicates that task k is assigned and zero otherwise. Then, the second objective function is to maximize the number of assigned project tasks, as stated in Eqn (4):

$$\text{Max} \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{m=1}^{M} \sum_{i=1}^{I} B_{jkmi}. \quad (4)$$

(c) The objective functions of the scheduling model are subjected to the following constraints:

The arrival of each project, ARR_{ji}, has different possible dates of specific corresponding probabilities. Let the arrival dates, $z_{ji} \in \{z_{j1}, ..., z_{ji}\}$ of project j has associated probability, $\lambda_{ji} \in \{\lambda_{j1}, ..., \lambda_{ji}\}$. That is:

$$ARR_{ji} \equiv z_{ji}, \quad \forall j,i; \quad (5)$$

$$\Pr\left(ARR_{ji} = z_{ji}\right) = \lambda_{ji}, \quad \forall j,i. \quad (6)$$

Let STD_{jk} denotes the standard time required to execute task k from project j. The STD_{jk} is generated from normal distribution with mean, μ, and standard deviation, σ, as expressed in Eqn (7). Moreover, STD_{jk} cannot be negative or zero value as stated in inequality (8):

$$STD_{jk} \sim \left(\mu, \sigma\right), \quad \forall j,k; \quad (7)$$

$$STD_{jk} > 0, \quad \forall j,k. \quad (8)$$

Let ACT_{jk} denotes the actual time consumed to execute task k from project j. The ACT_{jk} depends on members effectiveness, β_m, which is different from one member to another. Hence, when the actual time, ACT_{jk}, should equal to STD_{jk}, then β_m is 100%, but effectiveness is less than 100%, when ACT_{jk}, is larger than STD_{jk}, then β_m is smaller than 100%. Mathematically:

$$ACT_{jk} = \sum_{m=1}^{M}\sum_{i=1}^{I}\frac{STD_{jk}\cdot B_{jkmi}}{\beta_m},\ \ \forall j,k. \tag{9}$$

Let DUR_{mi} denotes the total of task execution durations assigned to member m in interval i. Then, DUR_{mi} is calculated as presented in Eqn (10):

$$DUR_{mi} = \sum_{j=1}^{J}\sum_{k=1}^{K} ACT_{jk}\cdot B_{jkmi},\ \ \forall m,i. \tag{10}$$

Let ID_{mi} denotes the idle time incurred by member m in interval i. Also, let AVB_{mi} be a binary variable that indicates the availability of member m in interval i; where the value of one means that member m is available in interval i and zero otherwise. Then, the ID_{mi} is calculated by DUR_{mi} from the regular working hours, REG, times AVB_{mi} as presented in Eqn (11):

$$ID_{mi} \geq \left(REG\cdot AVB_m\right) - DUR_{mi},\ \ \forall m,i. \tag{11}$$

The ID_{mi}, should be greater than or equals zero and less than or equals the maximum allowable idle time, $MXID$, as presented in inequality (12):

$$0 \leq ID_{mi} \leq MXID,\ \ \forall m,i. \tag{12}$$

The total idle time in hours, $TIDT$, from all members in all intervals is calculated as shown in Eqn (13):

$$TIDT = \sum\nolimits_{m=1}^{M}\sum\nolimits_{i=1}^{I} ID_{mi}. \tag{13}$$

Let OV_{mi} denotes the overtime incurred by member m in interval I, which is calculated by subtracting REG from DUR_{mi}; or mathematically:

$$OV_{mi} \geq DUR_{mi} - \left(REG\cdot AVB_m\right),\ \ \forall m,i. \tag{14}$$

The OV_{mi} should be greater than or equals zero and less than or equals the maximum allowable overtime, $MXOV$, as presented in inequality (15):

$$0 \leq OV_{mi} \leq MXOV,\ \ \forall m,i. \tag{15}$$

The total overtime in hours, $TIDT$, from all members in all intervals is calculated as shown in Eqn (16):

$$TOVT = \sum_{m=1}^{M}\sum_{i=1}^{I} OV_{mi}. \tag{16}$$

Let SK_{ml} be a binary parameter that determines whether member m has skill l, where the value of one indicates that member m has skill l and zero otherwise. Also, let RSK_{jkl}, be a binary parameter that determines whether task k from project j requires skill l; where the value of one indicates it requires and zero otherwise. Then, task k which requires L skills should be assigned to member m who has at least all the same required L skills as in Eqn (17):

$$B_{jkmi} \leq \sum_{l=1}^{L}\left(SK_{ml}\cdot RSK_{jml}\right),\ \ \forall j,k,m,i. \tag{17}$$

Project task k shall be assigned to member m at most once in any interval i; as in Eqn (18):

$$\sum_{m=1}^{M}\sum_{i=1}^{I} B_{jkmi} = 1,\ \ \forall j,k\ . \tag{18}$$

The assigning of project task k to member m should be implemented after the arrival date of the related project j as stated in Eqns (19) and (20):

$$B_{jkmi} \leq 1,\ \ \forall j,k,m,i \,|\, i \geq ARR_{ji}; \tag{19}$$

$$B_{jkmi} = 0,\ \ \forall j,k,m,i \,|\, i < ARR_{ji}. \tag{20}$$

Let $\chi(jk)$, (jk') be a binary variable which determines whether task k from project j depends on test k' from the same project j; where $\chi(jk)$, (jk') equals one when the task k depends on task k' and zero otherwise. Conversely, let $\chi(jk')$, (jk) be a binary variable which determines whether task k' from project j depends on test k from the same project j; where $\chi(jk')$, (jk) equals one when the task k' depends on task k and zero otherwise. Suppose that there are two tasks k and k' from the same project j. Assume that task k' depends on task k (task k should be assigned and executed first; $\chi(jk')$, (jk) = 1), if task k has not been assigned, then task k' cannot be assigned to be executed as stated in Eqn (21). On the other hand, when task k is be executed after task k' ($\chi(jk)$, (jk') = 1) and task k' has not been executed yet; then task k cannot be assigned as shown in Eqn (22). While, if task k depends on task k' ($\chi(jk)$, (jk') = 1) and task k' has not been assigned; then task k might be assigned to be executed as presented in Eqn (23). Similarly, if task k' depends on task k ($\chi(jk')$, (jk) = 1) and task k has been assigned to be executed, then task k' might be assigned to be executed as well; mathematically, as expressed in Eqn (24).

$$\sum_{m=1}^{M}\sum_{i=1}^{I} B_{jk'mi} = 0,$$
$$\forall j,k,k' \,|\, \chi_{(jk'),(jk)} = 1 \,\&\, B_{jkmi} = 0 \,\&\, k \neq k'; \tag{21}$$

$$\sum_{m=1}^{M}\sum_{i=1}^{I} B_{jkmi} = 0,$$
$$\forall j,k,k' \,|\, \chi_{(jk),(jk')} = 1 \,\&\, B_{jk'mi} = 0 \,\&\, k \neq k'; \tag{22}$$

$$\sum_{m=1}^{M}\sum_{i=1}^{I} B_{jkmi} \leq 1,$$
$$\forall j,k,k' \,|\, \chi_{(jk),(jk')} = 1 \,\&\, B_{jk'mi} = 1 \,\&\, k \neq k'; \tag{23}$$

$$\sum_{m=1}^{M}\sum_{i=1}^{I} B_{jk'mi} \leq 1,$$
$$\forall j,k,k' \,|\, \chi_{(jk'),(jk)} = 1 \,\&\, B_{jkmi} = 1 \,\&\, k \neq k'. \tag{24}$$

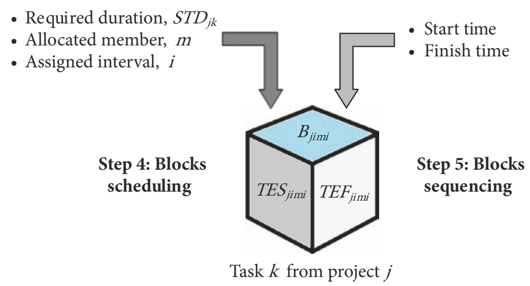

Figure 5. The results of optimal blocks scheduling from Step 4

The variable B_{pkmi} is binary variable, as stated in Eqn (25):

$$B_{jkmi} \in \{0,1\}, \ \ \forall j,k,m,i. \tag{25}$$

Solving the scheduling model, the optimal values are inserted in project blocks as depicted in Figure 5.

Step 5: Block sequencing

In particular, the sequencing process aims to complete the scheduled tasks in systematic order. Consequently, the sequencing optimization model will be developed to determine the start and finish times of each project block. The objective functions and constraints of the sequencing model are formulated as follows.

(i) Minimizing incurred costs (delay time)

The delay costs are incurred because of a delay in the execution process. Let $TDYT$ denotes the total delay times from all members in all intervals and let CD denotes the delay time unit cost per interval. Thus, the delay time costs, $DYTC$, is mathematically expressed in Eqn (26):

$$DYTC = TDYT \cdot CD. \tag{26}$$

Thus, the objective function is to minimize the total delay costs, as shown in Eqn (27):

$$\text{Min}\, DYTC. \tag{27}$$

(ii) Minimizing the task's start times

Let TES_{jkmi} denotes the execution start times of task k from project j by member m in interval i. In this regard, the execution of project tasks at the earliest possible start guarantees lower overtimes. Then, the following objective function is to minimize the execution start times, as stated in Eqn (28):

$$\text{Min} \sum\nolimits_{j=1}^{J} \sum\nolimits_{k=1}^{K} \sum\nolimits_{m=1}^{M} \sum\nolimits_{i=1}^{I} TES_{jkmi}. \tag{28}$$

(iii) The main constraints in the sequencing model are:

1. The execution of any task k from project j should start after the arrival of project j, as expressed in Eqn (29):

$$TES_{jkmi} \cdot B_{jkmi} \geq ARR_{ji}, \ \ \forall j,k,m,i\ . \tag{29}$$

2. Let TEF_{jkmi} denotes the execution finish time of task k from project j by member m in interval i. Then, the execution finish time of task k from project j by member m in interval i, TEF_{jkmi}, is calculated by adding the actual time required, ACT_{jk}, to the execution start time, TES_{jkmi}, as presented in Eqn (30):

$$TEF_{jkmi} = \left(TES_{jkmi} + ACT_{jk}\right) \cdot B_{jkmi}, \ \forall j,k,m,i. \tag{30}$$

3. Let PFT_j denotes the finish time for project j. Then, PFT_j equals the finish time for the latest task k from project j and executed by member m in interval i, TEF_{jkmi}, as presented in inequality (31):

$$PFT_j = Max \sum_{k=1}^{K} \sum_{m=1}^{M} \sum_{i=1}^{I} TEF_{jkmi} \cdot B_{jkmi}, \ \forall j. \tag{31}$$

4. Let $DUEj$ denotes the due date for project j. Consequently, the delay times incurred by all members in all intervals, $TDYT$, can be calculated by determining the difference between the project finish time, PFT_j, and the project due date, $DUEj$. Further, the difference between the project finish time, PFT_j, and the project due date, $DUEj$, should be greater or equal to zero as presented in Eqns (32) and (33):

$$\left(PFT_j - DUE_j\right) \geq 0, \ \ \forall j\ ; \tag{32}$$

$$TDYT = \sum_{j=1}^{J} \left(PFT_j - DUE_j\right), \ \ \forall j\ . \tag{33}$$

5. Member m cannot execute more than one task k during the execution of another task k' at the same interval i. For example, suppose that task k from project j and task k' from project j', were assigned to be executed by the same member m at the same interval i. Then, the execution of tasks k and k' should not start at the same time as expressed in Eqn (34):

$$\begin{aligned} &TEF_{j'k'mi} > TES_{jkmi} - \left(Q \times \left(2 - \sum_{j=1}^{J} \sum_{k=1}^{K} B_{jkmi} - \sum_{j'=1}^{J'} \sum_{k'=1}^{K'} B_{j'k'mi}\right)\right), \\ &TEF_{j'k'mi} < TES_{jkmi} + \left(Q \times \left(2 - \sum_{j=1}^{J} \sum_{k=1}^{K} B_{jkmi} - \sum_{j'=1}^{J'} \sum_{k'=1}^{K'} B_{j'k'mi}\right)\right), \\ &\forall j,j',k,k',m,i \mid j = j' \,\&\, k \neq k', \\ &\forall j,j',k,k',m,i \mid j \neq j'. \end{aligned} \tag{34}$$

6. In project j, the label number of tasks should have the priority to determine which task k to be executed first, unless that task k depends on another task k'. For example, suppose that project j has three tasks (k = 1, 2, 3). Then, the execution start time for task 1 should be before tasks 2 and 3. In the same manner, task 2 should be executed before task 3. Mathematically, as stated in Eqn (35):

$$TES_{jkmi} < TES_{jk'm'i} + \left(Q \times \left(2 - \sum_{m=1}^{M} \sum_{k=1}^{K} B_{jkmi} - \sum_{m'=1}^{M'} \sum_{k'=1}^{K'} B_{jk'm'i}\right)\right), \forall j,k,k',m,m',i \mid k < k'. \tag{35}$$

7. The dependency should be respected between assigned tasks for the execution in the sequencing process. For example, suppose that the execution of task k' from project j depends on the execution of task k from the same project j; ($\chi_{(jk'),(jk)} = 1$). Also, suppose the executions of tasks k and k' from project j have been assigned to members m and m' in interval i, ($B_{jk'm'i}$ and $B_{jkmi} = 1$), respectively. Then, the execution start time for task k', should begin after finishing the execution of task k, as presented in Eqn (36):

$$TES_{jkmi} < TES_{jk'm'i} + \left(Q \times \left(2 - \sum_{m=1}^{M} \sum_{k=1}^{K} B_{jkmi} - \sum_{m'=1}^{M'} \sum_{k'=1}^{K'} B_{jk'm'i} \right) \right), \forall j,k,k',m,m',i \mid k < k'. \quad (36)$$

Solving the sequencing model, the additional information will be added for each block as depicted in Figure 6. After storing the new component into the blocks, each block will carry the data about the assigned member and in which interval to be executed with start and finish times. the link between blocks in the same interval can be created from the start and finish times. Also, any block has start time at time 0, then it will be the first block to be executed. For example, block 1 has start time at time 0 and finish time at time 3 in interval 6. Block 2 has start time at time 3 and finish time at 4 in the same interval 6. Then, logically block 2 should be executed right after block 1 has been finished because there were at the same interval. In addition, the finish time of block 1 the same as the start time of block 2. Suppose there are 3 different blocks (1, 2, 3) from project j that were assigned to member m at interval i. Block 1 has start time at time 0 and finish at time ω, block 2 has start time at time ω and finish at time α while block 3 has start time at time α and finish at time ξ. Because block 1 has start time at time 0, then it will be the first block to be executed. Block 2 has the same start time as the finish time of block 1, thus, it will be the second block to be executed. If block 3 has start time the same as the finish time of block 2 then it will be the third block to be executed. In general, block 2 has start time the same as the previous block, and the finish time of block 2 will be the same as the start time of the following block as depicted in Figure 6.

Step 6: Block chaining and project executing

The optimal results of the scheduling and sequencing models will be shared with the relevant departments through computer networks. Once a block execution has been started, it is communicated in the blockchain. Project team can make necessary amendments to update the block data. Project manager can monitor the execution progress of project blocks. Figure 7 presents the chaining of project blocks.

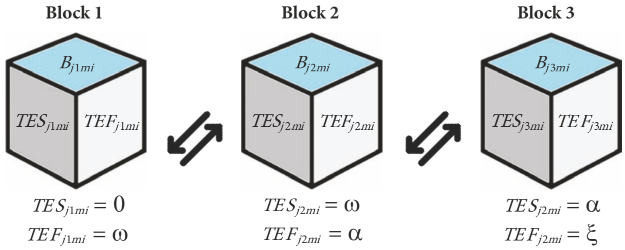

Figure 6. The sequencing block information from Step 5

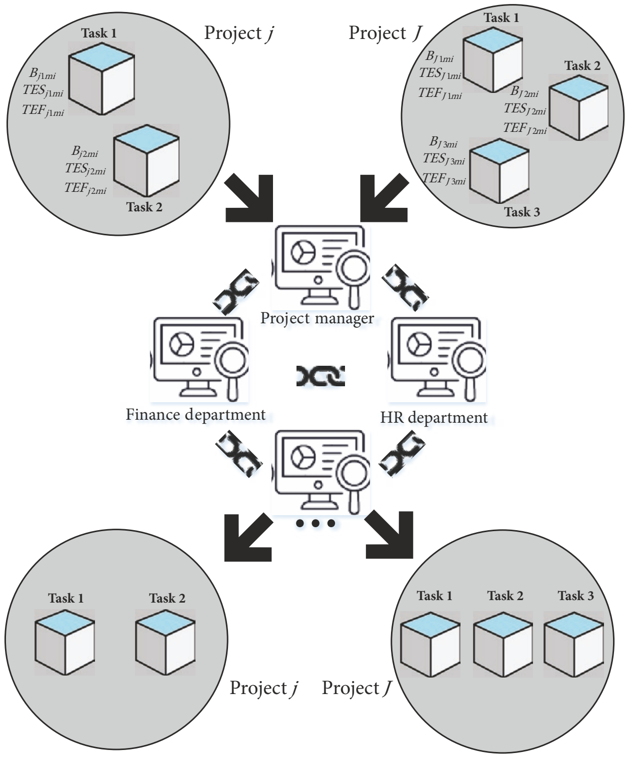

Figure 7. Chaining project blocks

3. Application of EPR system and research results

The EPR system was implemented on a real case study and is presented as follows.

Steps 1 and 2: Smart contacts and project arrivals

In this case study, a private company received five construction projects via smart contracts technology for specific time horizon (10 time-intervals). After approving the project contacts, five electronic project records were created in the EPR system with predetermined due dates. Assume that a project j has several possible arrival dates, $z_{ji} \in \{z_{j1}, ..., z_{ji}\}$, where each arrival has specific probability elements, $\lambda_{ji} \in \{\lambda_{j1}, ..., \lambda_{ji}\}$, as shown in Table 2. The outcomes for arrival dates of each project with their due dates are summarized in Table 3.

Step 3: Project breakdown structure

The five construction projects were broken down into tasks. A total of 121 tasks were reported. Table 4 summarizes the number of project tasks and resources. The tasks' execution durations were generated from normal distributions with four reptations as presented in Table 5. Further, the dependencies between project tasks were considered as presented in Table 6. The members' effectiveness and skills are listed in Table 7. Further, the availability matrix of members is presented in Table 8.

Table 2. The probabilistic arrival dates with their associated probabilities

Probabilistic sets	Outcomes		
Possible arrival dates, z_{ji}	z_{j1}	z_{j2}	z_{j3}
$z_{ji} \in \{1, 2, 3\}$	1	2	3
Associated probability, λ_{ji}	λ_{j1}	λ_{j2}	λ_{j3}
λ_{ij}	36%	41%	23%

Table 3. The confirmed projects arrival dates

Project	1	2	3	4	5
Arrival date (Interval)	2	2	1	3	1
Due date (Interval)	10	9	9	10	9

Table 4. The project tasks and main resources

Parameter	Value	Parameter	Value
Number of projects	5	Number of members	5 members
Number of tasks of project 1	15	Number of intervals	10 intervals
Number of tasks of project 2	23	Cost for overtime	10 $/hour
Number of tasks of project 3	28	Cost for idle time	15 $/ hour
Number of tasks of project 4	25	Maximum allowable overtime	4 hours per interval
Number of tasks of project 5	30	Maximum allowable idle time	3 hours per interval

Table 5. The generated replicates of tasks' durations using normal distribution

Project	Task	(μ, σ)	Task duration replicates			
	k		1	2	3	4
1	1	(2.5, 0.52)	2.54	2.41	2.39	2.33
	2		2.47	2.56	2.77	1.31
	3		2.37	2.52	2.42	2.35
	4		2.41	2.49	3.02	4.24
	5		2.59	2.41	2.75	2.14
	6	(4, 0.5)	4.52	3.67	3.36	6.07
	7		4.66	4.48	3.96	1.11
	8		3.91	4.05	4.42	2.33
	9		3.94	3.92	4.13	5.24
	10		4.52	3.87	3.38	3.47
	11	(5, 0.65)	5.80	5.83	5.27	4.13
	12		5.56	5.67	5.07	2.38
	13		5.26	5.03	5.52	3.99
	14		4.70	4.97	6.29	2.54
	15		5.33	5.41	4.23	4.80

Continue of Table 5

Project	Task	(μ, σ)	Task duration replicates			
	k		1	2	3	4
2	1	(3, 0.23)	3.19	3.31	3.07	4.05
	2		2.89	3.16	3.02	3.30
	3		2.84	3.57	2.89	2.47
	4		3.07	3.20	3.14	4.00
	5	(4.75, 0.5)	4.82	3.92	4.86	3.74
	6		3.47	4.90	4.97	3.59
	7		4.11	4.84	4.36	2.77
	8		3.68	4.50	4.99	4.55
	9		5.23	4.24	4.85	3.56
	10		4.95	5.19	5.78	4.96
	11		5.23	5.24	3.65	3.33
	12	(3.5, 0.3)	3.50	3.89	3.73	2.02
	13		4.20	3.40	3.68	3.62
	14		3.56	3.57	3.42	4.08
	15		3.12	3.34	3.55	4.27
	16		3.19	3.98	3.38	2.09
	17		4.06	3.83	3.41	5.04
	18		3.82	3.66	3.34	2.00
	19	(5.5, 0.65)	4.66	5.85	5.44	2.06
	20		5.21	5.27	4.89	5.53
	21		5.13	5.27	5.29	5.52
	22		5.79	4.91	4.84	4.64
	23		5.94	5.68	6.21	3.71
3	1	(2, 0.12)	2.07	1.73	2.02	2.33
	2		1.71	2.26	2.00	3.51
	3		1.94	2.05	2.09	2.26
	4		2.04	2.07	1.83	2.52
	5		2.20	2.09	2.09	2.09
	6		2.12	1.95	1.97	3.58
	7		2.04	2.00	1.83	3.92
	8		2.06	1.76	1.98	3.61
	9		2.30	1.75	1.99	2.78
	10		1.88	2.14	1.91	3.80
	11	(3.35, 0.17)	3.44	3.32	3.32	5.06
	12		3.33	3.42	3.33	5.18
	13		3.34	3.56	3.20	2.54
	14		3.34	3.50	3.42	2.00
	15		3.31	3.33	3.20	3.81
	16		3.44	3.26	3.36	1.22
	17		3.28	3.21	3.05	1.99
	18		3.46	3.35	3.60	3.41
	19		4.82	1.68	6.06	3.45
	20		3.38	3.57	3.53	1.07
	21		2.99	3.58	3.11	3.59
	22	(4.2, 0.22)	4.21	4.44	4.32	4.63
	23		4.24	4.25	4.25	3.34
	24		4.40	4.14	3.94	5.63
	25		4.24	4.17	4.43	4.63
	26		4.30	4.53	4.12	5.11
	27		4.04	4.60	4.07	4.12
	28		3.97	4.53	4.24	3.98

End of Table 5

Project	Task	(μ, σ)	Task duration replicates			
	k		1	2	3	4
4	1	(2.35, 0.75)	3.90	1.87	3.51	2.75
	2		2.68	2.31	2.05	1.09
	3		2.18	1.66	2.13	1.02
	4		2.23	2.17	2.36	1.76
	5	(2.75, 0.19)	2.98	2.82	2.98	2.18
	6		2.87	2.72	2.57	1.46
	7		2.61	2.68	2.66	3.54
	8		2.86	2.78	2.71	2.16
	9		2.42	3.07	2.72	3.72
	10		2.65	2.70	2.67	2.34
	11		2.49	2.57	2.86	2.36
	12		2.65	2.62	2.80	2.72
	13	(4.1, 0.25)	3.95	4.29	4.05	2.28
	14		2.86	1.71	2.46	2.45
	15		3.90	3.94	4.22	3.06
	16		4.46	3.95	4.08	3.28
	17		3.63	4.69	4.67	3.57
	18	(1.5, 0.1)	1.68	1.49	1.45	4.13
	19		1.64	1.64	1.42	1.99
	20		1.49	1.58	1.60	1.24
	21		1.44	1.38	1.57	1.27
	22		1.50	1.54	1.39	1.98
	23		1.46	1.55	1.39	1.72
	24		1.59	1.42	1.53	1.98
	25		1.46	1.61	1.50	1.54
5	1	(3.1, 0.45)	3.11	2.57	3.18	5.55
	2		3.28	2.85	3.66	1.46
	3		3.24	3.74	2.63	3.90
	4		3.19	2.70	2.81	3.65
	5		2.86	3.10	3.29	2.52
	6	(6.1, 0.66)	5.49	5.92	6.69	3.94
	7		3.21	3.57	3.21	4.00
	8		6.15	6.95	6.09	5.52
	9		6.05	5.82	5.61	4.73
	10		5.56	5.60	6.74	4.64
	11		5.07	6.34	6.94	4.43
	12	(5.57, 0.3)	5.67	5.37	5.52	3.92
	13		5.09	5.74	5.26	3.10
	14		5.70	5.42	5.46	5.14
	15		5.65	5.14	5.87	4.67
	16		5.54	5.88	5.91	3.72
	17		5.86	5.45	5.21	4.65
	18	(2.9, 0.1)	2.90	2.81	2.92	3.50
	19		2.76	2.91	2.79	2.11
	20		2.99	2.87	2.87	3.48
	21		3.02	3.12	2.94	2.84
	22		2.76	2.95	3.14	6.11
	23		2.90	2.87	2.95	1.27
	24		2.78	2.87	3.04	5.35
	25		2.96	2.93	2.82	5.06
	26		3.09	2.92	2.75	1.54
	27		2.96	2.85	2.99	4.51
	28		2.82	2.72	2.95	2.89
	29		3.00	2.92	2.82	3.68
	30		2.91	3.03	3.01	2.59

Table 6. Dependency between Task pairs

Project, j	Task, k	Task, k'	$\chi(jk), (jk')$	$\chi(jk'), (jk)$
1	1	7	1	0
	10	4	1	0
	13	9	1	0
2	5	18	1	0
	15	4	1	0
	3	22	1	0
	23	2	1	0
3	11	14	1	0
	12	10	1	0
	21	27	1	0
	25	23	1	0
4	24	22	1	0
	15	13	1	0
	20	9	1	0
	17	8	1	0
5	19	15	1	0
	24	13	1	0
	28	6	1	0
	12	2	1	0

Table 7. The matrix of members' effectiveness and skills

Members	Effectiveness	Skills						
		1	2	3	4	5	6	7
1	90%	1	–	1	1	–	1	1
2	99%	1	1	1	–	1	1	1
3	99%	1	1	1	1	1	–	–
4	87%	1	1	–	1	1	1	1
5	92%	1	1	1	1	1	1	1

Table 8. The availability matrix of members

Members	Intervals									
	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	0	1	1	1	1	1
2	0	1	1	0	1	1	1	1	1	1
3	1	0	1	0	1	1	1	1	1	1
4	1	1	1	1	1	0	1	1	1	1
5	1	1	0	1	1	1	1	1	1	1

Step 4: Block scheduling

This step assigns project tasks (blocks) to skilled members to be executed over ten predetermined intervals. The optimization models were coded using LINGO 18.0 using PC with an Intel Core i7- 7700 CPU of 3.6 GHz. The optimal scheduling of 121 blocks from the confirmed five projects over ten intervals is obtained, and the results are then shown in Table 9. The information on the optimal block assignment, such as the allocated member, required duration, and the assigned interval, are then attached to blocks. For example, block 10 from project 1 of duration 3.85 hours is assigned to member 1 in interval 3 as shown in Figure 8.

Table 9. The optimal assignment of project blocks to members over ten-time intervals

Project, j	Member, m	Interval, i									
		1	2	3	4	5	6	7	8	9	10
1	1	–	–	10	–	–	–	–	–	–	–
	2	–	4	–	–	–	–	5	–	–	–
	3	–	–	8	–	–	6	11	–	–	1
	4	–	15	14	9, 13	–	–	2	–	–	–
	5	–	12	–	–	–	–	–	3, 7	–	–
2	1	–	18	–	19	–	7	1, 22	6	–	14
	2	–	–	4, 15	–	–	8	3	–	–	–
	3	–	–	21	–	20	16	9	–	–	11
	4	–	–	–	–	12	–	13	–	–	2, 23
	5	–	5	–	–	–	17	–	–	–	10
3	1	4, 5	–	–	–	–	27	–	24	26	–
	2	–	6	9	–	3	21	–	10, 12	22	–
	3	15	–	–	–	–	–	–	–	–	–
	4	19	1, 7	20	–	8	–	17, 28	14	–	–
	5	18	16	–	13	23, 25	–	–	11	2	–
4	1	–	–	22, 24	16	–	11	5	–	14	21
	2	–	–	–	–	–	–	–	1	18	–
	3	–	–	4	–	23	–	–	9, 20	–	2
	4	–	–	–	–	13, 15	–	–	25	8, 17	10
	5	–	–	–	3	6	–	19	–	12	7
5	1	15, 19	13, 24, 26	–	23	–	–	–	–	10	16
	2	–	–	–	–	3, 17	–	–	–	–	14, 22
	3	11	–	–	–	–	–	–	5	27	9
	4	30	–	6, 28	7	–	–	–	21, 29	1	–
	5	20	18	–	25	8	4	2, 12	–	–	–

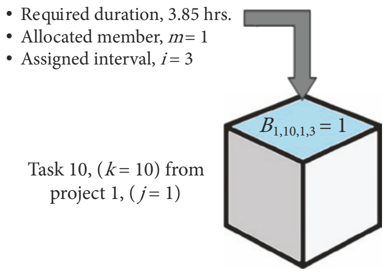

Figure 8. Example of optimal assignment information to blocks

Step 5: Block sequencing

The main parameters of the block sequencing model project include the tasks assignment matrix, actual durations, and the dependency between tasks. The optimal sequence of the assigned blocks is then solved by minimizing the total delay costs and execution start times. The obtained optimal results of blocks sequencing are listed in Table 10.

For example, block 17 (duration = 4.65 hrs in replicate 4 as stated in Table 5) in project 5 is executed in interval 5 by member 2 (effectiveness = 99%). Then, the actual execution time by member 2 will be 4.7 hrs (= 4.65 / 99%). Therefore, the execution process starts at 6.22 and finishes at 10.92. Next, the optimal sequence information is attached to blocks as shown in Figure 9. For illustration, the sequence of blocks 13, 24, and 26 in project 5 is shown in Figure 10. Note that the finish time of a block is the start time of the subsequent block. This information can be utilized in hash development block hash and control the changes in block information.

Step 6: Block chaining and execution

The blocks' optimal information was saved of all project tasks and then communicated with chained network stations to form construction projects' blockchain. For example, the blockchain for optimal scheduling and sequencing of project 1 is shown Figure 11.

Table 10. The optimal blocks sequence of each project

Interval, i	Member, m	Task, k	TES_{jkmi}	TEF_{jkmi}	Interval, i	Member, m	Task, k	TES_{jkmi}	TEF_{jkmi}
Project 1									
3	1	10	0	3.85	3	4	14	0	2.92
2	2	4	0	4.28	4		9	0	6.02
7		5	0	2.16			13	6.02	10.61
3	3	8	0	2.35	7		2	0	1.51
6		6	0	6.13	2	5	12	0	2.59
7		11	0	4.17	8		3	0	2.55
10		1	0	2.35			7	2.55	3.76
2	4	15	0	5.52	–				
Project 2									
2	1	18	0	2.22	5	3	20	0	5.59
4		19	0	2.29	6		16	6.13	8.24
6		7	0	3.08	7		9	4.17	7.77
7		1	0	4.5	10		11	3.25	6.61
		22	4.5	9.65	5	4	12	0	2.32
8		6	0	3.99	7		13	1.51	5.67
10		14	0	4.53	10		2	0	3.79
3	2	4	0	4.04			23	3.79	8.05
		15	4.04	8.35	2	5	5	2.59	6.65
6		8	0	4.6	6		17	0	5.48
7		3	6.16	8.65	10		10	0	5.39
3	3	21	2.35	7.93	–				
Project 3									
1	1	4	0	2.8	2	4	1	5.52	8.2
		5	2.8	5.12			7	8.2	12.71
6		27	3.08	7.66	3		20	2.92	4.15
8		24	3.99	10.25	5		8	2.32	6.47
9		26	0	5.68	7		17	5.67	7.96
2	2	6	4.28	7.9			28	7.96	12.53
3		9	8.35	11.16	8		14	0	2.31
5		3	0	2.28	1	5	18	0	3.71
6		21	4.6	8.23	2		16	4.56	5.89
8		10	0	3.84	4		13	0	2.76
		12	3.84	9.07	5		23	0	3.63
9		22	0	4.68			25	3.63	8.67
1	3	15	0	3.85	8		11	3.76	9.26
	4	19	0	3.97	9		2	0	3.82
Project 4									
3	1	22	3.63	5.83	10	3	2	6.61	7.71
		24	5.83	8.03	5	4	13	6.47	9.09
4		16	2.29	5.93			15	9.09	12.61
6		11	4.66	7.28	8		25	2.31	4.08
7		5	5.46	7.88	9		8	0	2.48
9		14	5.68	8.41			17	2.48	6.58
10		21	4.53	5.94	10		10	8.05	10.74
8	2	1	9.07	11.85	4	5	3	2.76	3.87
9		18	4.68	8.85	5		6	8.67	10.26
3	3	4	3.93	5.71	7		19	0	2.16
5		23	5.59	7.33	9		12	3.82	6.78
8		9	0	3.76	10		7	5.39	9.24
		20	3.76	5.01	–				

End of Table 10

Interval, i	Member, m	Task, k	TES_{jkmi}	TEF_{jkmi}	Interval, i	Member, m	Task, k	TES_{jkmi}	TEF_{jkmi}
Project 5									
1	1	15	5.12	10.33	10	3	9	7.71	12.49
		19	10.33	12.67	1	4	30	6.97	9.95
2		13	2.22	5.64	3		6	4.15	8.68
		24	5.64	11.58			28	8.68	12
		26	11.58	13.29	4		7	6.61	11.21
4		23	5.93	7.34	8		21	6.08	9.34
9		10	7.41	12.3			29	9.34	13.58
10		16	4.94	9.07	9		1	6.58	12.96
5	2	3	2.28	6.22	1	5	20	6.71	10.49
		17	6.22	10.92	2		18	5.89	9.69
10		14	0	5.19	4		25	3.87	9.37
		22	5.19	11.36	5		8	6.26	12.26
1	3	11	3.85	8.32	6		4	5.48	9.45
8		5	5.01	7.56	7		2	2.16	3.75
9		27	0	4.56			12	3.75	8.01

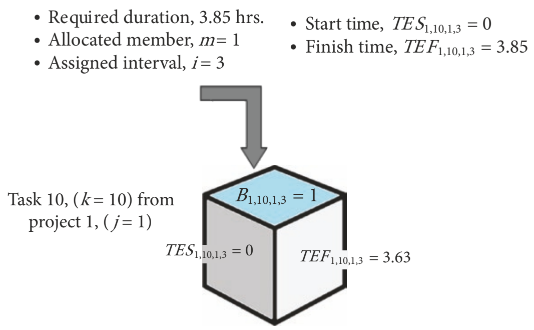

Figure 9. Example of optimal sequencing information to blocks

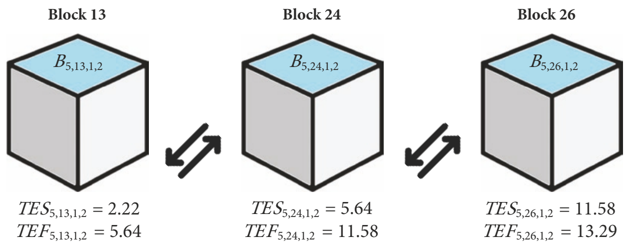

Figure 10. Example of blocks sequence in project 5

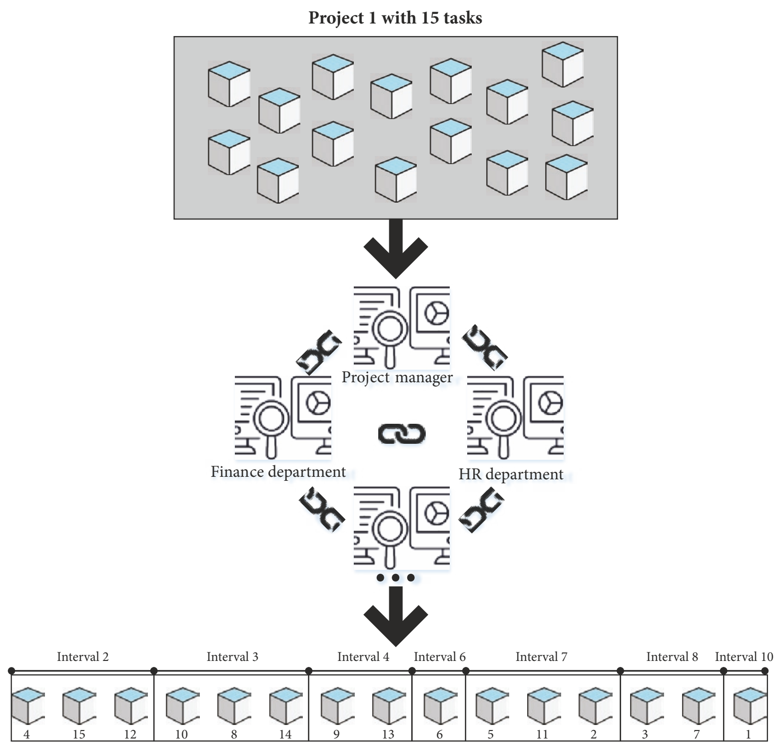

Figure 11. Chaining of optimal blocks project 1 using normally distributed durations

4. Research results and sensitivity analysis

4.1. Research results

This research developed an optimal blockchain for 121 tasks of five construction projects. The objective function values at the optimal blocks scheduling and sequencing using normal distribution are presented in Table 11, where the averages of the total overtime cost and idle time costs are $ 685.97 and 117.19, respectively. Figures 12 and 13 depict the numbers of assigned tasks to each member in all intervals and assigned tasks in each interval for all members, respectively. In Figure 12, the largest numbers of executed tasks correspond to member 1 (= 28 blocks) and member 4 (30 blocks). Further in Figure 13, the number of executed blocks in each interval ranges between 9 and 15. Figure 14 shows the calculated efficiency of each member in all intervals, where it is found that the smallest efficiency is 95%.

The members' idle times in all intervals are shown in Table 12, where the total idle time is found to be 8.53 hours. The maximum average idle time is 4.75 hours, which corresponds to member 1, while no idle time is incurred by member 4 in all intervals. In addition, no idle time is incurred in interval 5 by all members. Finally for intervals, the largest total idle time is incurred in interval 7 (= 2.91 hours).

Table 11. The values of objective functions

Output	Replicate				Average
	1	2	3	4	
Objective function	794.79	807.46	805.38	789.62	799.31
Total overtime (hrs.)	69.27	70.99	67.22	66.91	68.60
Total idle time (hrs.)	7.68	7.95	8.09	7.53	7.81
Overtime costs ($)	692.72	709.91	672.19	669.07	685.97
Idle time costs ($)	115.2	119.25	121.35	112.95	117.19

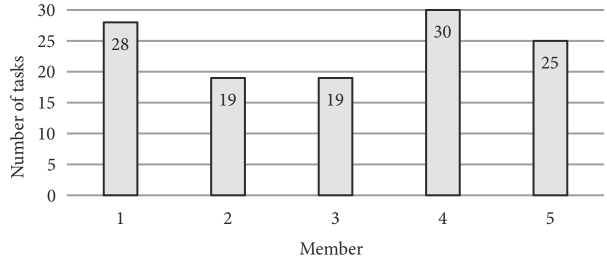

Figure 12. Number of assigned tasks for each member

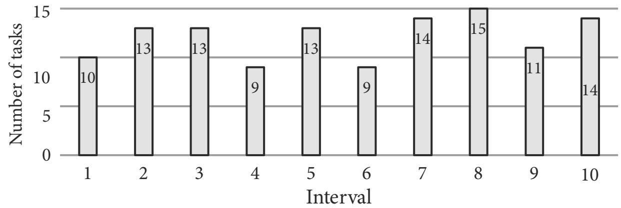

Figure 13. Number of assigned tasks in each interval

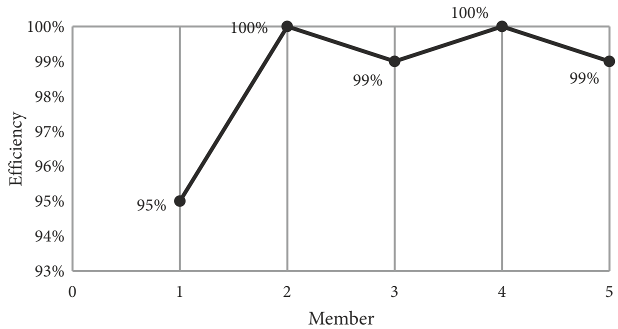

Figure 14. The efficiency of each member in all intervals

Table 12. The estimated members' idle times in each interval

Members	Interval										Idle time (hrs.)
	1	2	3	4	5	6	7	8	9	10	
1	0	0	0	1.39	–	1.45	1.91	0	0	0	4.75
2	–	0.48	0	–	0	0	0	0	0	0	0.48
3	0	–	0	–	0	0	0.31	1.52	0.78	0	2.61
4	0	0	0	0	0	–	0	0	0	0	0
5	0	0	–	0	0	0	0.69	0	0	0	0.69
Total (hrs.)	0	0.48	0	1.39	0	1.45	2.91	1.52	0.78	0	8.53

Table 13. The estimated members' overtimes in each interval

Members	Interval										Overtime (hrs.)
	1	2	3	4	5	6	7	8	9	10	
1	3.41	3.97	1.03	0	–	0	0	1.22	3.07	0.17	12.87
2	–	0	3.05	–	2.37	0.84	1.04	3.73	0.76	3.25	15.04
3	1.24	–	0.62	–	2.22	2.16	0	0	0	1.15	7.39
4	0.51	3.05	2.44	1.05	2.97	0	2.9	3.82	3.28	1.35	21.37
5	1.65	0.91	–	2.62	3.28	0.69	0	0.52	0.07	0.5	10.24
Total (hrs.)	6.81	7.93	7.14	3.67	10.84	3.69	3.94	9.29	7.18	6.42	66.91

Table 14. The estimated project delay costs

Project, j	ARR_{ji} (interval)	DUE_j (interval)	PFT_j	Delay times (interval)	Delay costs ($)
1	2	10	10	0	0
2	2	9	10	1	350
3	1	9	9	0	0
4	3	10	10	0	0
5	1	9	10	1	350
				Total	
				$TDYT = 2$	$ 700

On the other hand, the overtime hours of all members are listed in Table 13, where the total overtime is 66.91 hours. The largest total overtime of 21.37 and 10.84 hours corresponds to member 4 and interval 5, respectively.

Table 14 shows the associated incurred delay costs for each project, where the total delay cost is $ 700, which is incurred due to the delays in only project 2 (= $ 350) and project 5 (= $ 350). However, no delay costs are incurred in the remaining projects.

Based on the above analyses, it is concluded that the optimization models are found effective in scheduling and sequencing blocks in multiple construction projects by minimizing project costs and efficient utilization of project teams. Moreover, the blockchain technology can provide valuable information to project managers in analysing teams performance, in real-time monitoring of project progress, continual reduction of project costs, and saving project resources.

4.2. Sensitivity analysis

(1) Effect of duration replication

In this research, four replicates of tasks duration are generated from a normal distribution. The comparison of overtime and idle time in hours and costs between replicates is shown in Figure 15. In this figure, the smallest idle time (overtime) corresponds to replicate 4 of 66.91 (7.52) hours, whereas the largest idle time (overtime) corresponds to replicate 2 of 70.99 (7.95) hours. Apparently, slight differences (less than 6%) between the idle times (overtimes) of 4.08 (0.43) hours are observed which indicates the robustness of the model to replications. Similar conclusion can be obtained when comparing the idle time and overtime costs between replicates. The duration replication can provide project planners valuable information on the impact of the uncertainty of task duration on project plan and progress and enables them to set contingency plans to deal with the negative effects on project completion.

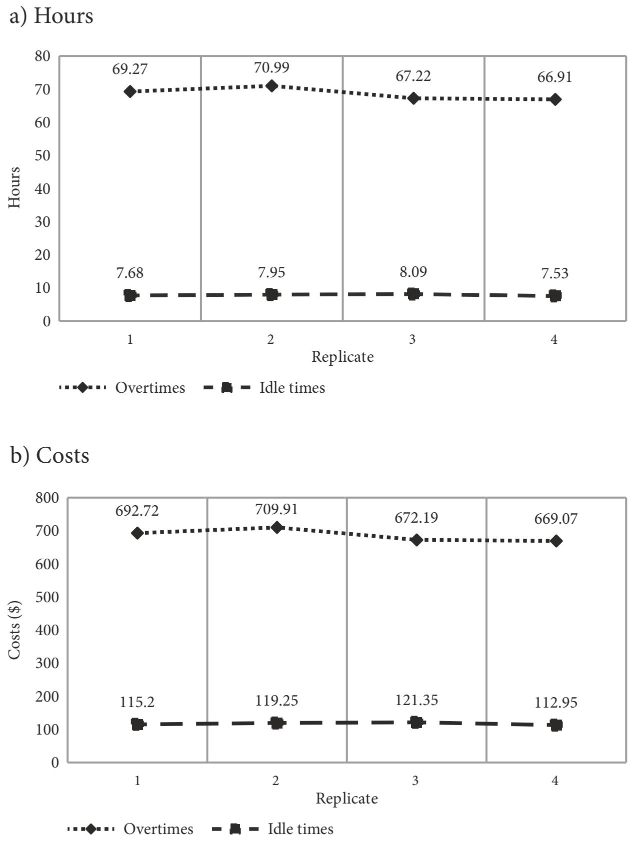

Figure 15. The comparison of overtime and idle time between replicates

(2) PERT distributed task durations

Program Evaluation and Review Technique (PERT) model predicts the mean and variance of activity duration utilizing higher, lower, and neutral time estimates provided by experiments and historical data as the distribution of activity duration. To examine the effect of changing the duration distribution on the optimal planning of project blocks, the tasks' execution durations are generated from PERT distribution with four reptations. Table 15 displays the PERT generated duration four replicates of each of the 15 tasks in project 1. The estimated idle time and overtime hours and costs are displayed in Table 16 for all replicates. It is obvious that there are negligible differences between objective function values between replicates due to uncertainty.

Further, the optimal sequence of the blocks for project 1 when the tasks durations are generated from PERT distribution is depicted in Figure 16. Comparing this sequence with the optimal blocks sequence for project 1 using normally and PERT distributed task durations, slight difference in blocks sequence in time intervals is observed. To examine the impact of such difference on the estimated model outputs, a comparison of estimated objective functions at between normal and PERT distributed task durations is conducted and also presented in Table 16. The results revealed insignificant differences (less than 5%) in the estimated objective functions between normal and PERT distributions for all replicated. Consequently, the normal or PERT distributions can be employed to generate task durations because they provide almost similar scheduling and sequencing plans at very close estimated objective functions.

Table 15. The PERT generated four replicates of task durations for project 1

Project *j*	Task *k*	Parameters	Task duration replicates			
			1	2	3	4
1	1	Low = 2.1 Neutral = 2.5 High = 3.02 α = 2.74 β = 3.26	2.36	2.19	2.32	2.33
	2		2.26	2.20	2.38	2.43
	3		2.28	2.36	2.16	2.27
	4		2.36	2.28	2.33	2.42
	5		2.42	2.40	2.29	2.16
	6	Low = 3.15 Neutral = 4 High = 5 α = 2.84 β = 3.16	3.63	3.69	3.49	3.60
	7		3.58	3.36	3.31	3.78
	8		3.49	3.54	3.42	3.42
	9		3.49	3.54	3.70	3.49
	10		3.48	3.39	3.52	3.53
	11	Low = 4 Neutral = 5 High = 6.5 α = 2.60 β = 3.40	4.41	4.19	4.38	4.61
	12		4.27	4.15	4.38	4.47
	13		4.51	4.11	4.29	4.19
	14		4.19	4.38	4.55	4.33
	15		4.49	4.32	4.55	4.16

Table 16. The comparison of objective function values between normal and PERT distributed block durations

Model output	Replicate				Average
	1	2	3	4	
Objective function using Normal distribution	794.79	807.46	805.38	789.62	799.31
Objective function using PERT distribution	826.72	799.85	821.76	802.59	812.73
% Difference	4.02%	0.94%	2.03%	1.64%	
Total overtime using Normal distribution (hrs.)	69.27	70.99	67.22	66.91	68.60
Total overtime using PERT distribution (hrs.)	72.05	70.32	68.59	68.01	69.74
% Difference	4.01%	0.94%	2.03%	1.64%	
Total idle time using Normal distribution (hrs.)	7.68	7.95	8.09	7.53	7.81
Total idle time using PERT distribution (hrs.)	8.01	7.86	8.25	7.65	7.94
% Difference	4.3%	1.13%	1.98%	1.59%	

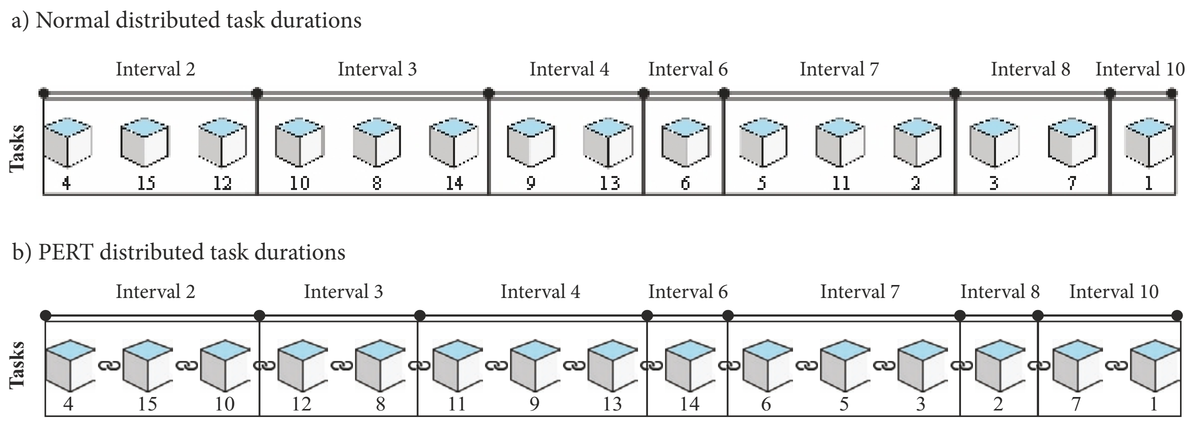

Figure 16. Optimal blocks sequence for project 1

Conclusions

In this research, an EPR system in blockchain was developed to communicate optimal scheduling and sequencing of multiple construction projects' tasks. A real case study of five construction projects with total of 121 task was provided for illustration. In developed EPR, two optimization models were formulated and then implemented to plan multiple projects over ten-time intervals. The objective functions mainly minimize the incurred costs (idle time costs and overtime costs) in the scheduling model, whereas minimizing the total delay costs in the sequencing process, and the summation of the start execution times. The developed optimization model considers probabilistic project arrival dates. Further, the model was formulated to consider stochastic task durations. The EPR and its optimization models were applied on a real case study to plan 5 projects of 121 tasks. The models assigned and sequencing all project tasks. The calculated idle time costs and overtime costs were $ 117.19 and $ 685.97, respectively. In addition, the total delay cost of $ 700 was incurred. Sensitivity analyses were performed by using PERT instead of normally distributed task durations with four replicates. The results showed that both distributions provide almost similar sequencing plans and very close estimates of objective functions in all replicates. In conclusion, the developed EPRs provides great assistance to project managers, as well as to relevant departments, through providing complete information on, effective monitoring of, and efficient communication for project blocks. Future research considers developing electronic record system for planning maintenance, port, and health activities.

Author contributions

Abbas Al-Refaie, Ahmad Al-Hawadi and Ghaleb Abbasi conceived the study and were responsible for the development of the framework for electronic project recording system, blockchain development to communicate scheduled and sequenced projects, data collection, analysis, and results interpretation. Natalija Lepkova was responsible for the review of the presented literature and data analysis.

Disclosure statement

Authors do not have any competing financial, professional, or personal interests from other parties. The authors declare no conflict of interest.

References

Abello, M. B., & Michalewicz, Z. (2014). Multiobjective resource-constrained project scheduling with a time-varying number of tasks. *The Scientific World Journal*, *2014*, 420101. https://doi.org/10.1155/2014/420101

Alladi, T., Chamola, V., Parizi, R. M., & Choo, K. K. R. (2019). Blockchain applications for industry 4.0 and industrial IoT: A review. *IEEE Access*, *7*, 176935–176951. https://doi.org/10.1109/ACCESS.2019.2956748

Almeida, B. F., Correia, I, & Saldanha-da-Gama, F. (2016). Priority-based heuristics for the multi-skill resource constrained project scheduling problem. *Expert Systems with Applications*, *57*, 91–103. https://doi.org/10.1016/j.eswa.2016.03.017

Al-Refaie, A., Qapaja, A., & Al-Hawadi, A. (2021). Optimal fuzzy scheduling and sequencing of work-intensive multiple projects under normal and unexpected events. *International Journal of Information Technology Project Management*, *12*(3), 64–89. https://doi.org/10.4018/IJITPM.2021070105

Ansari, R., Khalilzadeh, M., & Hosseini, M. R. (2022). A multi-objective dynamic optimization approach to project schedule management: A case study of a gas field construction. *KSCE Journal of Civil Engineering*, *26*(3), 1005–1013. https://doi.org/10.1007/s12205-021-0410-5

Ballestin, F., & Leus, R. (2009). Resource-constrained project scheduling for timely project completion with stochastic activity durations. *Production and Operations Management*, *18*(4), 459–474. https://doi.org/10.1111/j.1937-5956.2009.01023.x

Bruni, M. E., Beraldi, P., & Guerriero, F. (2015). The stochastic resource-constrained project scheduling problem. In C. Schwindt, & J. Zimmermann (Eds.), *Handbook on project management and scheduling: Vol. 2. International handbooks on information systems* (pp. 811–835). Springer. https://doi.org/10.1007/978-3-319-05915-0_7

Christodoulou, P., Christodoulou, K., & Andreou, A. (2018). A decentralised application for logistics: Using blockchain in real-world applications. *The Cyprus Review*, *30*(2), 181–193.

Creemers, S. (2015). Minimizing the expected makespan of a project with stochastic activity durations under resource constraints. *Journal of Scheduling*, *18*(3), 263–273. https://doi.org/10.1007/s10951-015-0421-5

de Melo, L. V., & de Queiroz, T. A. (2021). Integer linear programming formulations for the RCPSP considering multi-skill, multi-mode, and minimum and maximum time lags. *IEEE Latin America Transactions*, *19*(1), 5–16. https://doi.org/10.1109/TLA.2021.9423821

Deblaere, F., Demeulemeester, E., & Herroelen, W. (2011). Proactive policies for the stochastic resource-constrained project scheduling problem. *European Journal of Operational Research*, *214*(2), 308–316. https://doi.org/10.1016/j.ejor.2011.04.019

Delgoshaei, A., Rabczuk, T., Ali, A., & Ariffin, M. K. A. (2017). An applicable method for modifying over-allocated multi-mode resource constraint schedules in the presence of preemptive resources. *Annals of Operations Research*, *259*(1), 85–117. https://doi.org/10.1007/s10479-016-2336-8

Delgoshaei, A., Aram, A., Mantegh, V., Hanjani, S., Nasiri, A. H., & Shirmohamadi, F. (2019). A multi-objectives weighting genetic algorithm for scheduling resource-constraint project problem in the presence of resource uncertainty. *International Journal of Supply and Operations Management*, *6*(3), 213–230. https://doi.org/10.22034/2019.3.3

Guo, Y., & Liang, C. (2016). Blockchain application and outlook in the banking industry. *Financial Innovation*, *2*(1), 24. https://doi.org/10.1186/s40854-016-0034-9

Hong, Y., Choi, B., & Kim, Y. (2019). Two-stage stochastic programming based on particle swarm optimization for aircraft sequencing and scheduling. *IEEE Transactions on Intelligent Transportation Systems*, *20*(4), 1365–1377. https://doi.org/10.1109/TITS.2018.2850000

Huang, S., Li, G., Ben-Awuah, E., Afum, B. O., & Hu, N. (2020). A stochastic mixed integer programming framework for underground mining production scheduling optimization considering grade uncertainty. *IEEE Access*, *8*, 24495–24505. https://doi.org/10.1109/ACCESS.2020.2970480

Issaoui, Y., Khiat, A., Bahnasse, A., & Ouajji, H. (2019). Smart logistics: Study of the application of blockchain technology. *Procedia Computer Science*, *160*, 266–271. https://doi.org/10.1016/j.procs.2019.09.467

Kadri, R. L., & Boctor, F. F. (2018). An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case. *European Journal of Operational Research*, *265*(2), 454–462. https://doi.org/10.1016/j.ejor.2017.07.027

Kawaguchi, N. (2019). Application of blockchain to supply chain: Flexible blockchain technology. *Procedia Computer Science*, *164*, 143–148. https://doi.org/10.1016/j.procs.2019.12.166

Lee, J., Azamfar, M., & Singh, J. (2019). A blockchain enabled Cyber-Physical System architecture for Industry 4.0 manufacturing systems. *Manufacturing Letters*, *20*, 34–39. https://doi.org/10.1016/j.mfglet.2019.05.003

Lee, D., Lee, S. H., Masoud, N., Krishnan, M. S., & Li, V. C. (2021). Integrated digital twin and blockchain framework to support accountable information sharing in construction projects. *Automation in Construction*, *127*, 103688. https://doi.org/10.1016/j.autcon.2021.103688

Lohmer, J., & Lasch, R. (2020). Blockchain in operations management and manufacturing: Potential and barriers. *Computers & Industrial Engineering*, *149*, 106789. https://doi.org/10.1016/j.cie.2020.106789

Marinho, A., Couto, J. P., & Teixeira, J. M. C. (2021). Relational contracting and its combination with the BIM methodology in mitigating asymmetric information problems in construction projects. *Journal of Civil Engineering and Management*, *27*(4), 217–229. https://doi.org/10.3846/jcem.2021.14742

Muzylyov, D., & Shramenko, N. (2019). Blockchain technology in transportation as a part of the efficiency in Industry 4.0 strategy. In *Lecture notes in mechanical engineering. Advanced manufacturing processes. InterPartner 2019* (pp. 216–225). Springer. https://doi.org/10.1007/978-3-030-40724-7_22

Ning, M., He, Z., Jia, T., & Wang, N. (2017). Metaheuristics for multi-mode cash flow balanced project scheduling with stochastic duration of activities. *Automation in Construction*, *81*, 224–233. https://doi.org/10.1016/j.autcon.2017.06.011

Olawumi, T. O., Chan, D. W., Ojo, S., & Yam, M. C. (2021). Automating the modular construction process: A review of digital technologies and future directions with blockchain technology. *Journal of Building Engineering*, *46*, 103720. https://doi.org/10.1016/j.jobe.2021.103720

Ortiz-Pimiento, N., & Diaz-Serna, F. (2018). The project scheduling problem with non-deterministic activities duration: A literature review. *Journal of Industrial Engineering and Management*, *11*(1), 116–134. https://doi.org/10.3926/jiem.2492

Pham, A., Bui, A., Nguyen, T., Nguyen, T., Pashchenko, F., & Pashchenko, F. (2021). Optimization of model parameters by complex probabilistic criteria. In *2021 International Siberian Conference on Control and Communications* (*SIBCON*), Kazan, Russia. IEEE. https://doi.org/10.1109/SIBCON50419.2021.9438856

Project Management Institute. (2021). *Beyond agility. PMI's pulse of the profession report*. https://www.pmi.org/learning/library/beyond-agility-gymnastic-enterprises-12973

Quoc, H., The, L., Doan, C., & Thanh, T. (2019). Solving resource constrained project scheduling problem by a discrete version of Cuckoo search algorithm. In *NAFOSTED Conference on Information and Computer Science* (pp. 73–76), Hanoi, Vietnam. IEEE. https://doi.org/10.1109/NICS48868.2019.9023867

Shu, X., Su, Q., Wang, Q., & Wang, Q. (2018). Optimization of resource-constrained multi-project scheduling problem based on the genetic algorithm. In *2018 15th International Conference on Service Systems and Service Management* (*ICSSSM*), Hangzhou, China. IEEE. https://doi.org/10.1109/ICSSSM.2018.8465086

Singh, M. (2020). Blockchain technology for data management in Industry 4.0. In R. Rosa Righi, A. Alberti, & M. Singh (Eds.), *Blockchain technology for industry 4.0* (pp. 59–72). Springer, Singapore. https://doi.org/10.1007/978-981-15-1137-0_3

Swan, M. (2015). *Blockchain blueprint for a new economy*. O'Reilly Media Inc.

Su, Z., Wang, H., Wang, H., & Shi, X. (2020). A financial data security sharing solution based on blockchain technology and proxy re-encryption technology. In *2020 IEEE 3rd International Conference of Safe Production and Informatization* (pp. 462–465), Chongqing City, China. https://doi.org/10.1109/IICSPI51290.2020.9332363

Tanwar, S., Parekh, K., & Evans, R. (2020). Blockchain-based electronic healthcare record system for healthcare 4.0 applications. *Journal of Information Security and Applications*, *50*, 102407. https://doi.org/10.1016/j.jisa.2019.102407

Tian, Y., Xiong, T., Liu, Z., Mei, Y., & Wan, L. (2022). Multi-objective multi-skill resource-constrained project scheduling problem with skill switches: Model and evolutionary approaches. *Computers & Industrial Engineering*, *167*, 107897. https://doi.org/10.1016/j.cie.2021.107897

Tirkolaee, E. B., Goli, A., Hematian, M., Sangaiah, A. K., & Han, T. (2019). Multi-objective multi-mode resource constrained project scheduling problem using Pareto-based algorithms. *Computing*, *101*(6), 547–570. https://doi.org/10.1007/s00607-018-00693-1

Ullah, F., & Al-Turjman, F. (2021). A conceptual framework for blockchain smart contract adoption to manage real estate deals in smart cities. *Neural Computing and Applications*. https://doi.org/10.1007/s00521-021-05800-6

Ulusoy, G., & Hazır, Ö. (2021). Stochastic project scheduling with no resource constraints. In *An introduction to project modeling and planning. Springer texts in business and economics* (pp. 167–198). Springer, Cham. https://doi.org/10.1007/978-3-030-61423-2_6

Zhao, G., Liu, S., Lopez, C., Lu, H., Elgueta, S., Chen, H., & Boshkoska, B. M. (2019). Blockchain technology in agri-food value chain management: A synthesis of applications, challenges, and future research directions. *Computers in Industry*, *109*, 83–99. https://doi.org/10.1016/j.compind.2019.04.002

Zhong, P., Zhong, Q., Mi, H., Zhang, S., & Xiang, Y. (2019). Privacy-protected blockchain system. In *2019 20th IEEE International Conference on Mobile Data Management* (pp. 457–461), Hong Kong, China. IEEE. https://doi.org/10.1109/MDM.2019.000-2