Automation in Construction 105 (2019) 102799

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Multipurpose linear programming optimization model for repetitive activities scheduling in construction projects

Juan Diego García-Nieves[1], José Luis Ponz-Tienda*, Angélica Ospina-Alvarado, Mateo Bonilla-Palacios

Universidad de los Andes, Carrera 1 Este No. 19A-40, Bogotá, Colombia

ARTICLE INFO

Keywords:
Resources allocation
Construction project scheduling
RCPSP
TCTP
Repetitive activities
Linear programming

ABSTRACT

Repetitiveness in project's activities has gained an important role in the construction industry. Multiple linear scheduling methods have been proposed in order to fully take advantage of the spatial and temporal information these type of project can provide to practitioners. Besides the advances in the optimization models in these fields, to the extent of the authors knowledge, there is still pending a complete and flexible mathematical linear programming formulation that allow practitioners to easily and jointly solve the Resource allocation, Resource-Constrained Project Scheduling and Time-Cost Tradeoff problem, taking into account as many scheduling properties, benefits and challenges that linear scheduling of repetitive activities imply. This paper shows a complete guide and computational experimentation, of a novel mathematical model that can be easily used by practitioners to optimize construction schedules considering to the largest extent the time and space conditions repetitive projects offer. Particularly, it contributes to the repetitive activities scheduling body of knowledge by successfully implementing a robust linear programing optimization model in a real construction project, while considering as much linear scheduling characteristics as possible. It proves that relationships in the sub-activity level, continuity conditions, multiple modes of execution, controlled acceleration routines and execution mode shifts, and multiple crews can be easily and jointly integrated to a linear optimization model by adding simple linear restrictions to the model.

1. Introduction

The improvement of the scheduling techniques has always been a very important field of study in construction not only at the scholar level, but at industrial level. Researchers working collaboratively with industry practitioners have developed numerous mechanisms in order to deliver projects in a more efficient way. Specifically, time and resource consumption efficiency are one of the most important challenges that schedulers must overcome when a new project is conceived [1–3].

As the nature of the construction project differ, traditional scheduling methods has been extensively criticized, forcing specialized scheduling tools to emerge [4–7]. Extensive literature research has demonstrated that these traditional time-driven techniques do not display spatial and resource consumption information that allows practitioners to build schedules based on spatial resource consumption continuity and distance constraints [5,8–11]. Particularly, these methods fail to incorporate resource continuity and distance constraints, while offering poor or none information about crew execution times and location [11]. Additionally, the uninterrupted placement of resources in construction units is not a problem addressed by them, nor by its resource-oriented extensions [5]. To overcome this limitation, authors have proposed modification to networks techniques that can handle continuous and non-continuous connection of similar sub activities by using FS0 and maxFS0 relationships. In this line Hajdu [12,13], proposes PtP (point-to-point) and maxFS relationships to overcome precedence diagram method impossibility to simultaneously include location and time lags, non-linear activities and activity overlapping, assuming that activities are non-interruptable, besides OR and bi-directional relationships to deal with multimode execution modes and sub-activities.

Despite all advances, all the previously mentioned limitations make traditional methods and techniques unsuitable for scheduling and controlling activities in which work is repeated in unit by unit throughout the length of the project. Over the years, the need for

* Corresponding author at: Universidad de los Andes, Bogotá, Colombia.
E-mail addresses: jd.garcia12@uniandes.edu.co (J.D. García-Nieves), jl.ponz@uniandes.edu.co (J.L. Ponz-Tienda), am.ospina@uniandes.edu.co (A. Ospina-Alvarado), m.bonilla867@uniandes.edu.co (M. Bonilla-Palacios).
[1] First Author: Juan Diego García-Nieves.

https://doi.org/10.1016/j.autcon.2019.03.020
Received 12 August 2018; Received in revised form 4 March 2019; Accepted 23 March 2019
Available online 09 May 2019

repetitive activity (RA henceforth) scheduling methodologies has been identified and valued by several authors, resulting in the development of various linear scheduling methods that seek to integrate time and space [10,14]. Mainly, these methods can be classified in to two groups: Line of Balance (LOB) [15,16] and Linear Scheduling Method (LSM) [14,17]. The first one, provides additional information on the activities execution times and termination rate, while the second, provides an insight on activities productivity rates [3,6,8,9,11]. Both methods, provide an extensive approach on new space-time interactions that practitioners can use to create more sophisticated schedules. Space and time buffers and relationships between activities and sub-activities, crew availability, work interruption and construction work units are some of new schedule characteristics that can be observed using these methods.

Besides the nature of the activities analyzed and the scheduling method used, scheduling problems have been classified depending on the project time, cost or resource consumption needs or limitations. Literature has defined these core areas of study in 4 distinctive problems: Resource allocation problem, Time cost trade-off problem (TCTP), Resource leveling problem (RLP) and resource-constrained project scheduling problem (RCPSP). Resource allocation or resource management deals with activity scheduling and resource consumption identification. Further, TCTP deals with the construction and/or optimization of the project's cost profile that arise from alternative project schedules, providing practitioners information of how the project's duration impact the project cost. Lastly, RLP and RCSPS deal with resource consumption smoothing or constraints. The former, takes advantage of non-critical activity float times to improve and optimize resource consumption without modifying the project duration. The latter deals with project scheduling under limited resource availability. As practitioners might observe, the inability to obtain the necessary resources to carry out scheduled activities may produce a variation of the project's duration.

In contrast to the case studied, logical and precedence relationships between activities and sub-activities must be guaranteed and considered in the scheduling process, in order to determine the scheduled times and sub-activity execution modes that optimize a specific goal (objective function).

On the one hand, resource allocation and TCTP aim to determine the optimal schedule that either minimize the projects makespan or minimize the project cost without exceeding a given deadline [18]. On the other hand, resource-based problems (RLP and RCPSP) emerged from the practitioners' need to handle specific resource consumption characteristics. Schedulers need to reduce the inefficiency embodied in unbalanced resource consumption profiles (RLP) or need to struggle with resource availability that might limit the implementation of ideal non-restrictive schedules (RCPSP) [3,11,19–22].

To the extent of authors' knowledge, there is still a need of a complete and realistic mathematical model to jointly solve the RCPSP and TCTP, taking into account as many scheduling properties, benefits and challenges that the scheduling of repetitive activities under realistic environments imply. To partially fulfil this need, this research proposes a flexible linear mathematical model, specially designed for repetitive activities in construction projects. The proposed model takes into account realistic conditions as: 1) the four traditional relationships (Finish-to-start, Start-to-start, Finish-to-finish and Start-to-finish) between activities on sub-activity level, 2) discretional continuity between sub-activities established by the scheduler, 3) multiple execution modes for the activities, 4) acceleration and deaccelerating routines inside each activity, 5) controlled maximum shifts in the execution of the activities independently of the total execution modes available for the activities, and finally 6) the possibility of establishing multiple crews for the execution of an activity.

The novelty of the proposed mathematical model and its contribution to knowledge resides in: a) the simultaneous implementation of multiple execution with controlled execution shifts, accelerating or deaccelerating the sub-activities respect the previous sub-activity, in such way that an activity can be executed in 6 different execution modes, but with only two shifts (changes) in such way that will be executed in three modes, b) the possibility of consider multiple crews for the execution of the activities, fully integrated with repetitive activity scheduling, resource-constrained scenarios, time-cost analysis and discretional continuity restrictions. Additionally, and perhaps more importantly, this research provides a detailed step-by-step guideline that will allow practitioners to build optimization models for repetitive activity schedules based on the specific project demands.

Accordingly, Section 2 presents a literature review and the state-of-art related to the distinct solutions authors have proposed to attend the repetitive activity scheduling problems. Section 3 details the proposed model for repetitive activities scheduling in construction projects. In Section 4, an example of implementation is shown. Finally, conclusions, limitations, and further research are presented in Section 5.

2. Literature review

Due to feasible solutions combinatorial explosion (typical in NP-hard problems) [3,21], the use of exact optimization models to solve construction scheduling problems strictly rely on hardware processing capacities. Therefore, a great number of heuristic and metaheuristic mathematical models have been proposed to rapidly find approximate acceptable schedules solutions. As models continuously evolve, authors have progressively incorporated repetitive activity characteristics such as: multiple crews, multiple modes of activity and sub-activity execution, sub-activity relationships, acceleration routines, and continuity constraints.

Regarding to resource allocation problems in repetitive activities, Selinger [10] initially formulated a multimode one state variable model based on continuity constraints. Further, Russel & Caselton [17] advanced on the previous model by considering two-state variables and interruption vectors that allow discontinuous activity execution. Adeli & Karim [18] CONSCOM neural dynamic optimization model, successfully integrated repetitive and non-repetitive activities while considering job conditions variation, multiples crew, and discontinuous activities. El-Rayes & Moselhi [4] delivered a dynamic programming formulation considering optimum crew formation and activity interruptions. Hyari & El-Rayes [23] proposed a multi-objective genetic algorithm (GA henceforth) to search for construction schedules that minimize project makespan while maximizing crew continuity.

Hegazy & Kamarah [24] developed a high-rise construction scheduling model that optimally determined the work interruptions, construction methods and the number of crews necessary to minimize project duration. The model considered productivity factors, resource constraints, and distance relationships. Liu Wang [25] constraint programming formulation, continued advancing on previous models' conception by allowing multiple modes of execution within the sub-activities of a particular activity. Zhang & Zou [11] GA, successfully integrated sub-activity relationships, activity fragmentation, and multimode activity execution.

Likewise, multiple authors have studied resource-based scheduling problems in the context of repetitive construction projects. Particularly, in the case of RCPSP in repetitive activities projects (RCPSP-RA henceforth), Leu & Hwang [26] proposed a GA that successfully integrated resource constraints and resource sharing. Hsie et al. [27] introduced an effective evolutionary algorithm that addressed work continuity while adopting lead-time and lead distance between activities. Contrarily from previous proposals, the model considered time-based sub-activities units instead of length based sub-activity units. At the same time, multiple modes of execution per time unit are allowed. Zhang & Zou [11] formulated a GA based on activity fragmentation, sub-activity relationships and multiple modes of sub-activity execution. Further, Biruk & Jaskowski [28] built a mixed integer linear programming model based on work continuity and optimal crew formation. Finally, Garcia-Nieves et al. [3] successfully introduced an integer linear programming model in order to solve the RCPSP-RA, taking into account continuity restrictions, the coexistence of multiple modes of execution within the same activity and acceleration routines.

Similarly, TCTP in repetitive activities projects (TCTP-RA henceforth) was initially studied by Reda [7], introducing a linear programming mathematical model to schedule repetitive activities minimizing project cost. The model considered finish-to-start activity relationships, continuity constraints, and constant production rates. Skibniewski & Armijos [29] proposed an optimization model for the minimization of the total cost of the project performance, under two approaches, the fist by making use of functional relationships between the itemized costs of performance of a specific activity and the time required to perform such work, and the second by making use of historical data concerning the costs os a given activity. These authors solved the nonlinearity in the nonlinear function, were necessary, approximating them to piecewise linear functions. In other line, Moselhi & El-Rayes [30] further utilized a dynamic programming model that incorporated cost as an important decision variable in the optimization problem. Senouci & Eldin [31] proposed a dynamic programming formulation for scheduling of non-serial linear projects that successfully incorporated time-cost tradeoff analysis. The model handled linear, non-linear, and discrete activity time cost functions while considering activities with multiple crew formation, productions rates and lag times. Hegazy & Wassef [32] delivered a GA to optimally schedule non-serial repetitive activities considering multiple constructions methods, number of crews and sub-activities with non-continuous execution.

El-Rayes & Kandil [33] adopted a multi-objective GA to execute a newly conceived time-cost-quality trade-off analysis. This model successfully integrated repetitive and non-repetitive activities scheduling, while optimizing resource consumption schedules that minimize construction time and maximize quality. Hyari et al. [34] delivered a bio-objective optimization model that provided a set of Pareto near optimal TCTP solutions. The formulation was built on repetitive activities and considered multiple crews and work interruptions. Terry & Lucko [35] further expanded the TCTP scope by exploring the use of singularity functions within the cost optimization process.

Finally, Zhang & Zou [11], developed a non-linear mixed integer formulation to solve the multimode discrete TCTP. The authors' proposal allowed practitioners to implement multiple modes of execution in each sub-activity on a particular activity through the use of a GA that can handle soft logic relationships between activities. More recently, Zou et al. [36] proposed a mixed integer linear programming formulation using linear optimization to determine the number of crews needed per activity to meet an established deadline (deadline satisfaction problem). The initial model considered continuous activities, constant production rates, and cost analysis. Further, Zou et al. [37] expanded the scope of the previous model presenting a bio-objective optimization that allowed discontinuous activity execution.

Despite the continuous evolution of repetitive activities scheduling tools, none of these models provide neither robust nor exact linear programming mathematical formulation for RA that successfully integrates as many linear scheduling characteristics as possible.

3. The proposed model

The proposed model considers a set of P activities in which an identical set of N_q repetitive units are completed using M_q different executions modes. In order to develop a flexible model that allows schedulers to analyze as much possible variants of the scheduling of repetitive activities problem, the use of M_q binary variable sets containing variables X_{qjtm} is necessary. Based on Pritsker, Waiters, & Wolfe [38] formulation, variables $X_{qjtm} = 1$ if a sub-activity j in an activity q is finished in the period t under the execution mode m, and $X_{qjtm} = 0$ otherwise.

Hereunder, the main objective of this section is to illustrate how to strengthen the model formulation, in order to handle most of the scheduling challenges that commonly can arise in construction projects. The mathematical model proposed considers the following initial parameters:

P Number of activities of the project

N_q Number of sub − activities in each activity q

M_q Number of execution modes for activity q

$$\lambda_q \in \{0,1\} \begin{cases} =1 & \text{if an activity } q \text{ must be executed in a continuous way} \\ =0 & \text{otherwise} \end{cases}$$

d_{qm} duration of the sub − activities in activity q executed in mode m

$\gamma_{p_i q_j}$ Lag between two subactivities i and j, of activities p and q

K Number of renovable resources of the project

a_{qkm} Consumption of resource k by activity q executed in mode m

u_k Availability of resource k per period t

UB Known project *Upper Bound makespan*

Moreover, the model decision variables are:

$$X_{qjtm} \begin{cases} =1 & \text{if a sub − activity } j \text{ of an activity } q \text{ finishes in period } t \\ & \text{in mode } m \\ =0 & \text{otherwise} \end{cases}$$

3.1. Objective function for resource allocation problem

Finding the optimal activity times in order to minimize the project makespan is the core problem of activity scheduling. Based on the proposed formulation, the resource allocation problem can be solved using the objective function presented by Eq. (1).

$$\min f(x) = \sum_{m=1}^{M_P} \sum_{t=1}^{UB} X_{PN_Ptm} \bullet t \tag{1}$$

If the latest project activity is unknown prior to the scheduling procedure, a dummy activity $P + 1$ must be added to the model. This activity must have a duration of zero units and must be the successor of all possible schedule routes.

3.2. Preserving the relationship between sub-activities of an activity

Precedence relationships between sub-activities of the same activity are easily kept by adding relationship restrictions to the model. These restrictions can be easily modified to deal with lag time and all nature of sub-activities relationships (start-to-finish, start-to-start, start-to-finish, and finish-to-finish). For example, to ensure a sub-activity j + 1 in an activity q cannot start until the sub-activity j in the same activity is completely finished, the model restriction can be written as follows.

$$\sum_{m=1}^{M_q} \left(\sum_{t=1}^{UB} X_{qjtm} \bullet t \right) \le \sum_{m=1}^{M_q} \left(\sum_{t=1}^{UB} (X_{q,j+1,tm} \bullet t) - d_{qm} \sum_{t=1}^{UB} X_{q,j+1,tm} \right) \quad \forall\, q \in \{1,\ldots,P\}, \forall\, j \in \{1,\ldots,N\} \tag{2}$$

Also, to guarantee all starting times of the first sub-activity of all activities are non-negative integers, the following restriction must be added. If sub-activities within an activity have no precedence relationship, this restriction should be applied to all the starting time of the complete set of sub-activities.

$$0 \leq \sum_{m=1}^{M_q} \left(\sum_{t=1}^{UB} (X_{q,j,tm} \bullet t) - d_{qm} \sum_{t=1}^{UB} X_{q,j,tm} \right) \quad \forall\, q \in \{1,\ldots,P\},\, \forall\, j \in \{1\} \tag{3}$$

3.3. Establishing relationships between activities in the sub-activity level

Activity precedence based on sub-activity precedence of interdependent activities is fundamental to build more realistic construction projects. In contrast with the previous section, were finish-to-start (FS) relationships are the most common relationships, in this section, it is very important to exemplify how to build all four kinds of natural precedence relationship between activities in the sub-activity level. The following restriction illustrates how a sub-activity j in an activity q cannot start until a sub-activity i located in a predecessor activity p is completely finished. A lag time $\gamma_{p_i q_j}$ between sub-activities is also included [3].

$$\sum_{m=1}^{M_p} \left(\sum_{t=1}^{UB} X_{pitm} \bullet t \right) + \gamma_{p_i q_j} \leq \sum_{m=1}^{M_q} \left(\sum_{t=1}^{UB} (X_{qjtm} \bullet t) \quad \forall\, i \textit{ predecessor of } j \right. \\ \left. - d_{qm} \bullet \sum_{t=1}^{UB} X_{qjtm} \right) \tag{4}$$

Eqs. (4a), (4b), (4c) are reformulations of these restriction based on start-to-start (SS), start-to-finish (SF), and finish-to-finish (FF) relationships.

$$\sum_{m=1}^{M_p} \left(\sum_{t=1}^{UB} (X_{pitm} \bullet t) - d_{pm} \bullet \sum_{t=1}^{UB} X_{pitm} \right) + \gamma_{p_i q_j} \\ \leq \sum_{m=1}^{M_q} \left(\sum_{t=1}^{UB} (X_{qjtm} \bullet t) - d_{qm} \bullet \sum_{t=1}^{UB} X_{qjtm} \right) \forall\, i \textit{ predecesor of } j \tag{4a}$$

$$\sum_{m=1}^{M_p} \left(\sum_{t=1}^{UB} (X_{pitm} \bullet t) - d_{pm} \bullet \sum_{t=1}^{UB} X_{pitm} \right) + \gamma_{p_i q_j} \quad \forall\, i \textit{ predecesor of } j \\ \leq \sum_{m=1}^{M_q} \left(\sum_{t=1}^{UB} X_{qjtm} \bullet t \right) \tag{4b}$$

$$\sum_{m=1}^{M_p} \left(\sum_{t=1}^{UB} X_{pitm} \bullet t \right) + \gamma_{p_i q_j} \leq \sum_{m=1}^{M_q} \left(\sum_{t=1}^{UB} X_{qjtm} \bullet t \right) \quad \forall\, i \textit{ predecesor of } j \tag{4c}$$

3.4. Guarantying continuous execution

Depending on the nature of the activity analyzed, practitioners might differ whether an activity must be executed in a continuous way. To ensure model flexibility related to this issue, a new restriction containing a dummy variable per activity λ_q is proposed. In Eq. (5), if an activity q must be executed in a continuous way, $\lambda_q = 1$, and otherwise, $\lambda_q = 0$. Furthermore, to guarantee restriction compliance, a large negative constant value δ is included (i.e., $\delta = -1e4$).

$$\sum_{m=1}^{M_q} \left(\sum_{t=1}^{UB} X_{q1tm} \bullet t - d_{qm} \bullet \sum_{t=1}^{UB} X_{qitm} \right) + \delta \bullet \lambda_q \qquad \forall\, q \in \{1,\ldots,P\} \\ \geq \sum_{m=1}^{M_q} \left(\sum_{t=1}^{UB} X_{qN_q tm} \bullet t \right) - \sum_{m=1}^{M_q} \sum_{j=1}^{N} \left(\sum_{t=1}^{UB} (X_{qjtm}) \bullet d_{qm} \right) + \delta \tag{5}$$

3.5. Coexistence of multiple execution modes

Multimode execution allows practitioners to choose an optimal execution mode between a list of different execution velocities in order to provide a better scheduling solution. To ensure a realistic model, only one mode of execution can be attributed to each sub-activity. Therefore, the following restriction must be added to the model.

$$\sum_{m=1}^{M_q} \sum_{t=1}^{UB} X_{qjtm} = 1 \ \forall\ q \in \{1,\ldots,P\},\, \forall\, j \in \{1,\ldots,N\} \tag{6}$$

3.6. Allowing acceleration routines

Constant variation of the execution modes within the sub-activities of the same activity might be undesired and inefficient [3]. The need for scheduling techniques that allow accelerations and decelerations on the rate of activity execution were proposed by Arditi [8] and solve to optimality by Garcia-Nieves et al. [3] under the concept of controlled acceleration routines. Controlled acceleration routines are defined as changes in the execution modes between successive sub-activities, that correspond to exclusively forward or exclusively backward shifts on crew size changes during the complete activity execution. The implementation of this behavior requires the use of the restriction represented by Eq. (7).

$$j * \sum_{z=m}^{M_q} \sum_{t=1}^{UB} X_{qjtz} - \sum_{x=1}^{j} \sum_{t=1}^{UB} X_{qxtm} \geq 0 \ \forall\ q \in \{1,\ldots,P\},\, \forall\, j \in \{1,\ldots,N\},\, \forall\, m \\ \in \{2,\ldots,M\} \tag{7}$$

On the one hand, if forward accelerations are desired in a particular activity, the list of execution modes of that activity must be built from lowest to highest execution velocity. On the other hand, if decelerations are desired, the list must be built from highest to lowest execution velocity.

3.7. Controlling execution mode shifts

Practitioners may require under some scenarios that, for a particular activity q, only a maximum number of execution modes (M_{q-max}) from a tentative list of execution velocities can be used. The restrictions presented by Eqs. (8) and (9), allow the optimization model to consider this potential schedule option. In order to build these mathematical restrictions, new binary decision variables per activity mode must be added (λ_{qm}). In addition, a large positive constant value m and nearly zero positive constant value e are included to guarantee restriction compliance (i.e., $m = 1e4 - e = 1e\text{–}2$).

$$\sum_{j=1}^{N} \sum_{t=1}^{UB} X_{qjtm} - (m - e) \bullet \lambda_{qm} \leq e \ \forall\ q \in \{1,\ldots,P\},\, \forall\, m \in \{1,\ldots,M\} \tag{8}$$

$$\sum_{m=1}^{M_q} \lambda_{qm} = M_{q-max} \ \forall\ q \in \{1,\ldots,P\} \tag{9}$$

3.8. Adapting to resource consumption problems

Resource k consumption per period (ε_{tk}) can be calculated using Eq. (10).

$$\varepsilon_{tk} = \sum_{q=1}^{P} \sum_{m=1}^{M_q} \left(a_{qkm} \bullet \sum_{x=t}^{x+d_{qm}-1} \sum_{j=1}^{N} X_{qjxm} \right) \tag{10}$$

In this way, the model can be rapidly adjusted to solve resource constrained (RCPSP) scheduling problems just by adding a new restriction. In the case of RCPSP, the model modification simply implicates the addition of the restriction presented in Eq. (11).

$$\varepsilon_{tk} \leq u_k \ \forall\ t \in \{1,\ldots,UB\},\, \forall\, k \in \{1,\ldots,K\} \tag{11}$$

From a practical point of view, if specialized teams, or subcontractors, are used for executing different sub-activities, then, is not needed to consider these resources when the objective function is the project makespan.

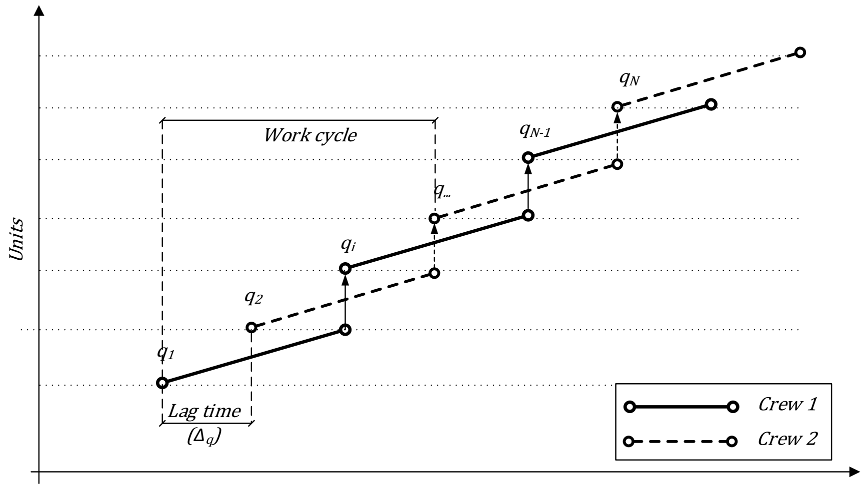

Fig. 1. Enabling multiple crew implementation.

3.9. Implementing time-cost trade-off (TCTP)

Practitioners' priority resides in finding the optimal schedule that provides the minimum project cost. In order to adapt the model to this specific need, the objective function can be changed to the one proposed by Eq. (12). In this equation, *DC* stands for direct cost, *IC* for indirect cost and BM for the monetary expenses or income related to contractor's penalties or incentives.

$$\min f(x) = DC + IC + BM \tag{12}$$

Project direct and indirect costs are easily obtained using Eqs. (13) and (14). Nonetheless, the cost trade-off implementation strictly depends on previously defined indirect costs per period (*ICR*) and the labor (L_{qm}), equipment (E_{qm}), material (M_{qm}) and idle day (IDR_{qm}) costs for each execution mode for each activity. Also, idle days per activity (ID_q) can be obtained using Eq. (15).

$$DC = \sum_{q=1}^{P}\left(\sum_{j=1}^{N}\left(\sum_{m=1}^{M_q}\sum_{t=1}^{UB}(d_{jqm}\bullet(L_{qm}+E_{qm})+M_{qm})\bullet X_{qjtm}\right)+ID_q\bullet IDR_{qm}\right) \tag{13}$$

$$IC = \left(\sum_{m=1}^{MP}\sum_{t=1}^{UB}X_{PNtm}\bullet t\right)\bullet ICR \tag{14}$$

$$ID_q = \sum_{m=1}^{M_q}\left(\sum_{t=1}^{UB}X_{qNtm}\bullet t\right) - \sum_{m=1}^{M_q}\sum_{j=2}^{N}\left(\sum_{t=1}^{UB}(X_{qjtm})\bullet d_{qm}\right) - \sum_{m=1}^{M_q}\left(\sum_{t=1}^{UB}X_{q1tm}\bullet t\right) \tag{15}$$

With regards to Penalties and Incentives allocated to or by each contractor, they can be computed using Eq. (16). A new decision binary variable φ_q is used to detect when penalties and incentives occur. Therefore, $\varphi_q = 1$ when an activity q contractor overpass certain period benchmark (BM_q), otherwise $\varphi_q = 0$.

$$BM = \varphi_q\bullet Pen_q + (\varphi_q - 1)\bullet Inc_q \tag{16}$$

To guarantee the decision variable φ_q is assigned accordingly to the obtained schedule, the following restriction must be added. A large positive constant value m and nearly cero positive constant value e is included (i.e., $m = 1e4 - e = 1e\text{–}2$).

$$BM_q - (m-e)\bullet\varphi_q \geq e + \sum_{m=1}^{MP}\sum_{t=1}^{UB}X_{qNtm}\bullet t \tag{17}$$

It is worth highlighting the formulation for penalization and incentive can be easily transformed into a more complex discontinuous function by adding more benchmarks, or by calculating penalizations or incentives depending on the number of mismatch days with reference to the activity's benchmark. Equation (18) illustrates a penalization and incentive equation which value varies linearly with the mismatch days (function slope ≠ 0.

$$BM = \varphi_q\bullet Pen_q\left(BM_q - \sum_{m=1}^{MP}\sum_{t=1}^{UB}X_{qNtm}\bullet t\right) + (\varphi_q - 1)\bullet Inc_q\left(BM_q - \sum_{m=1}^{MP}\sum_{t=1}^{UB}X_{qNtm}\bullet t\right) \tag{18}$$

One may note this consideration affects the linearity of the proposed restriction. Hence, non-linear optimization software must be used.

3.10. Enabling multiple crew implementation

Under some highly pressure circumstances, schedulers might need to use multiple crews to simultaneously execute sub-activities of a particular activity. Evidence of this need is exposed by the deadline satisfaction problem, which seeks practitioners to determine the number of crews needed in each specific activity of the project to meet the goals. To solve this problem, several methodologies and mixed integer linear programming (MILP) formulations have been proposed by different authors [36,37,39,40].

Nonetheless, due to the nature of the proposed binary formulation, which forces activity times to be integers, the model cannot be modified with minor changes to include the deadline satisfaction problem while always preserving linearity conditions. However, the model can rapidly be modified in order to consider the use of a deterministic number of crews in the scheduling process.

If a C_q number of crews are desired to execute an activity q, restriction 19 must be added to the model. A new variable (Δ_q) is used to specify the lag time between successive crews from a specific work cycle (Fig. 1) in an activity q.

$$\sum_{m=1}^{M_q}\sum_{t=1}^{UB}X_{qNtm}\cdot t - \sum_{m=1}^{M_q}\sum_{t=1}^{UB}X_{q1tm}\cdot t + \delta\cdot\lambda_q \geq \Delta_q\cdot\left(N_q - \frac{N_q - e}{C_q} - 1\right) + \left(\sum_{m=1}^{M_q}\left(d_{qm}\cdot\sum_{t=1}^{UB}X_{q1tm}\right) - \Delta_q\cdot C_q - \Delta_q\right)\cdot\frac{N_q - e}{C_q} + \delta$$

$$\forall\ q \in \{1,\dots,P\} \tag{19}$$

Similarly, as in the continuity restriction proposed before, if an activity q must be executed in a continuous way, $\lambda_q = 1$, and otherwise, $\lambda_q = 0$. Likewise, to guarantee restriction compliance, a large positive constant value δ and e near cero positive constant value e is included (i.e., $\delta = 1e4$, $e = 1e\text{–}2$).

For a more realistic projects condition, it is required that, while using multiple crews, the same execution mode is utilized during the whole activity execution for all the crews. In this way, only one execution mode should be considered, or restriction 20 should be used with $M_{q-max} = 1$. Also, Eqs. (20), (21), (22) must be added to ensure a maximum of C_q crews working simultaneously and precedence relationships between sub-activities of the same activity.

$$\sum_{m=1}^{M_q}\sum_{t=1}^{UB} X_{qjtm}\cdot t - \sum_{m=1}^{M_q}\sum_{t=1}^{UB} X_{q1tm}\cdot t \ge \Delta_q\cdot\left(j - \frac{j-e}{C_q} - 1\right) + \left(\sum_{m=1}^{M_q}\left(d_{qm}\cdot\sum_{t=1}^{UB} X_{q1tm}\right) - \Delta_q\cdot C_q - \Delta\right)\cdot\frac{j-e}{C_q}$$

$$\forall\, q \in \{1,\dots,P\},\ \forall\, j \in \{2,\dots,N\} \tag{20}$$

$$\sum_{m=1}^{M_q}\left(\sum_{t=1}^{UB} X_{q,j-C_q,tm}\bullet t\right) \le \sum_{m=1}^{M_q}\left(\sum_{t=1}^{UB}(X_{qjtm}\bullet t) - d_{qm}\sum_{t=1}^{UB} X_{qjtm}\right) \quad \forall\, q \in \{1,\dots,P\},\ \forall\, j \in \{C_q+1,\dots,N\} \tag{21}$$

$$0 \le \sum_{m=1}^{M_q}\left(\sum_{t=1}^{UB}(X_{qjtm}\bullet t) - d_{qm}\sum_{t=1}^{UB} X_{qjtm}\right) \quad \forall\, q \in \{1,\dots,P\},\ \forall\, j \in \{1,\dots,C_q\} \tag{22}$$

4. Example of implementation

In order to illustrate the flexibility of the proposed model, the construction project example proposed by Garcia-Nieves et al. [3] is analyzed. The project consists of the construction of a twelve-story building, the seven main activities analyzed are 1. Structure, 2. Facilities, 3. Masonries, 4. Carpentry & Painting, 5. Equipment, 6. Finishing, and 7. Delivery of the apartments. Each activity includes 12 identical sub-activities, totaling 84 sub-activities for the whole project. The example contemplates two different construction modes (different duration and normalized monetary resource consumption per period) for each sub-activity belonging to each activity.

As detailed by Garcia-Nieves et al. [3], the precedence relationships and lag between activities, the normalized monetary periodical resource consumption and the duration per mode of execution of each activity are presented in Table 1. Thus, q_j stands for the activity assigned number, p_i stands the predecessor activity assigned number and p_iq_j refer to the number of the sub-activity j in the predecessor activity p, that must be completed before sub-activity j in activity q could be started. It is important to mention that only finish-to-start relationships between and within activities in the sub-activity level are considered.

To demonstrate the formulation flexibility, ten different models containing different combinations of restrictions where built. It is important to highlight that in all models the same precedence between activities in the sub-activity level is guaranteed. On one hand, in models considering only one crew per activity, sub-activities of the same activity should be at least executed one after the other (Eqs. (2) and (3)). On the other hand, when multiple crews were used, relationships between activities of the same activities must satisfy restrictions given by Eqs. (20), (21) and (22).

Table 1
Conceptual data model representation for the example of implementation [3]

							Mode #1	
Activity Id	Activity	# of subactivities	q_j	p_i	p_iq_j	$\gamma_{pi\ qj}$	d_{q1}	a_{qk1}
1	Structure	12	1			0	12	8
2	Facilities	12	2	1	5	0	6	5
3	Masonries	12	3	2	3	0	5	8
4	Carp-paint	12	4	3	3	0	6	3
5	Equipment	12	5	4	3	0	5	5
6	Finishing	12	6	5	2	0	5	10
7	Deliveries	12	7	6	1	0	1	3

Structure modes 1 & 2 durations were changed in order to achieve a better representation of the model capabilities.

The models were executed considering an upper bound of 150 labor days. As a result, 21,000 binary decision variables were accounted for in the optimization process. All models were implemented via Excel 2016 interface and linearly optimized using Open Solver Optimizer V.2.9.0 and Gurobi Optimization Software V.7.0.2 (Academic License). Computing optimization time did not exceed 754 s. A HP Z640 desktop (Intel Xeon E5-2637v4 3.5 2400 processor and a 32GB DDR4-2400 (4x8GB) 2CPU RegRAM) was used to run all the models.

It is common that when optimizing the makespan, multiple optimal schedule solutions are obtained with the same project duration (more commonly when fragmentation is allowed). In order to obtain optimal schedules with the earliest possible activities starting time, a simple variation of the objective function must be considered. This modification requires that the sum of the starting times of all the activities, also known as Tardiness, must be included in the objective function. Eq. (23) shows how to build the modified objective function based on the proposed binary formulation. It is imperative to highlight that an impact reduction factor ρ is added to guarantee that the optimization process is guided mainly by the makespan minimization and not the tardiness minimization. A ε value of 0.0001 was used in all the examples of implementation models.

$$\min f(x) = \sum_{m=1}^{M_P}\sum_{t=1}^{UB} X_{PNtm}\bullet t + \varepsilon\bullet\sum_{m=1}^{M_q}\sum_{q=1}^{P}\sum_{j=1}^{N}\left(\sum_{t=1}^{UB} X_{qjtm}\bullet t - d_{qm}\bullet\sum_{t=1}^{UB} X_{qjtm}\right) \tag{23}$$

Table 2, presents the conditions considered for each model, and the resultant project makespan. It is important to mention that if continuity is considered, all activities are being executed in a continuous way. In addition, since the goal of this paper is to show the versatility of the model, multiple crews were only implemented in the Structure activity (two crews assigned and a 3 days lag time between crews). Further, RCPSP scenario consisted of 40 monetary units per period availability. The models' Excel files which include the optimization model, LSM graphs and resource consumption histograms, can be downloaded from https://goo.gl/4J3LDd.

Table 2
Model's execution conditions and resultant project's makespan.

Model	Continuity	Acceleration routines	Makespan	RCPSP availability	Multiple crews
1	Yes	Yes	121	Unrestricted	No
2	No	No	121	Unrestricted	No
3	Yes	Yes	121	Unrestricted	No
4	No	Yes	121	Unrestricted	No
5	Yes	Yes	128	u = 40	No
6	No	Yes	126	u = 40	No
7	Yes	No	88	Unrestricted	Yes
8	No	No	88	Unrestricted	Yes
9	Yes	No	113	u = 40	Yes
10	No	No	103	u = 40	Yes

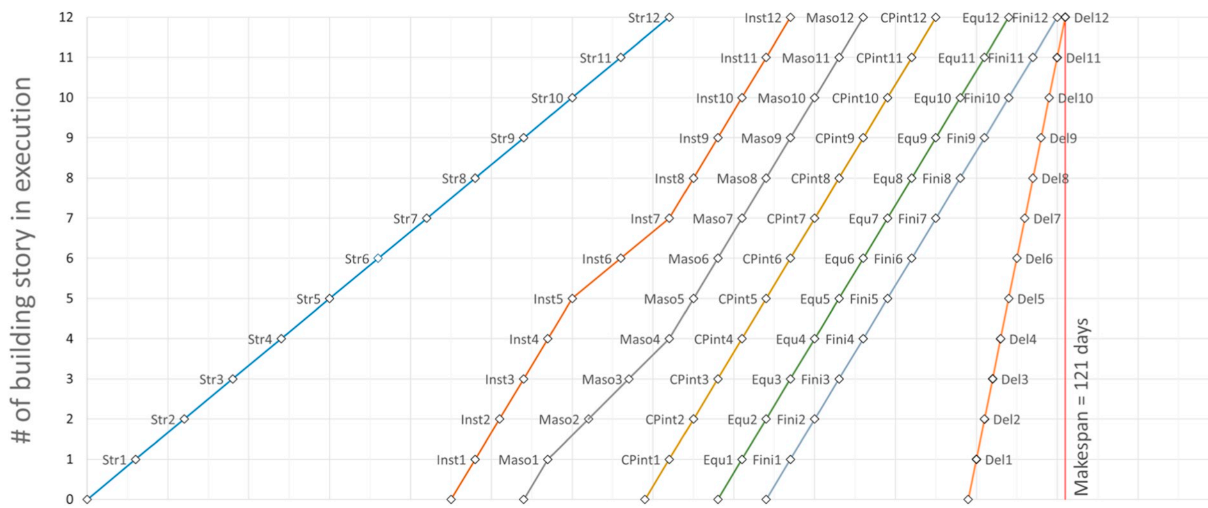

Fig. 2. Model 1 with one crew, continuity condition and acceleration routine.

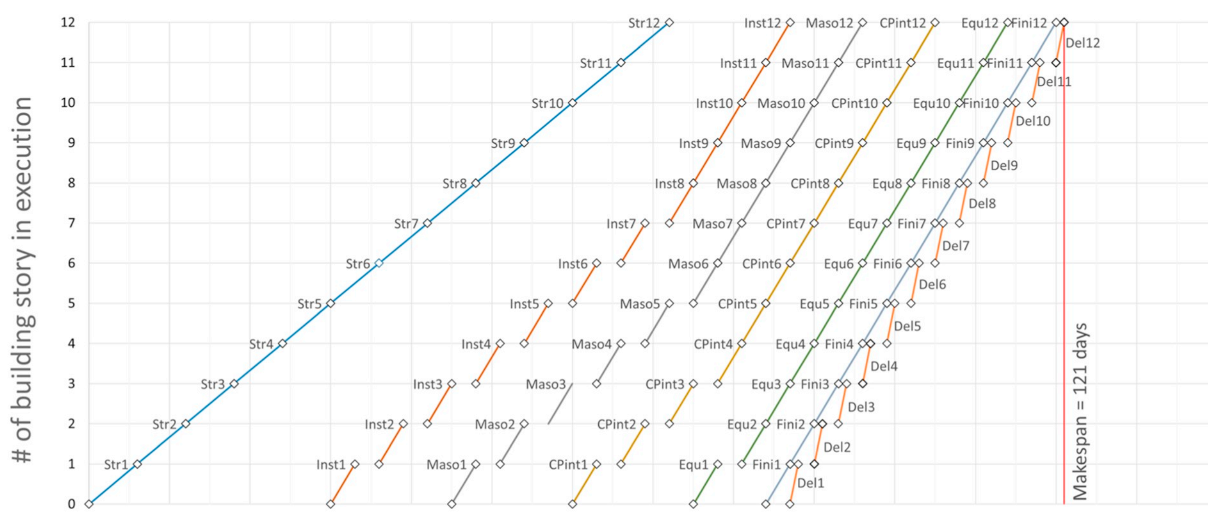

Fig. 3. Model 2 with one crew, non-continuity condition, and non-acceleration routine.

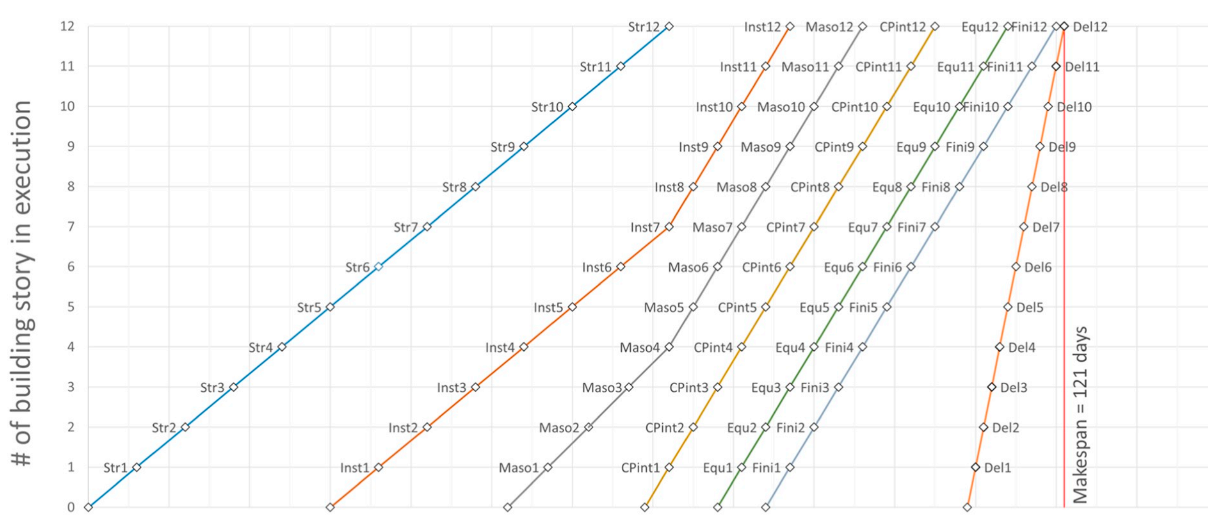

Fig. 4. Model 3 with one crew, continuity condition, and acceleration routine.

In order to graphically observe the implications of the different model's execution conditions in the project schedule, Figs. 2 to 11 provide the LSM graphic schedules of the ten analyzed models, Figs. 2 to 7 considering only one crew in activity execution and Figs. 8 to 11 considering multiple crew execution only in the structure activity. As one can further notice, the models with one crew differ from models with multiple crews, mainly in the sub-activity overlapping that occurs in the first activity. As mentioned before, this phenomenon is allowed

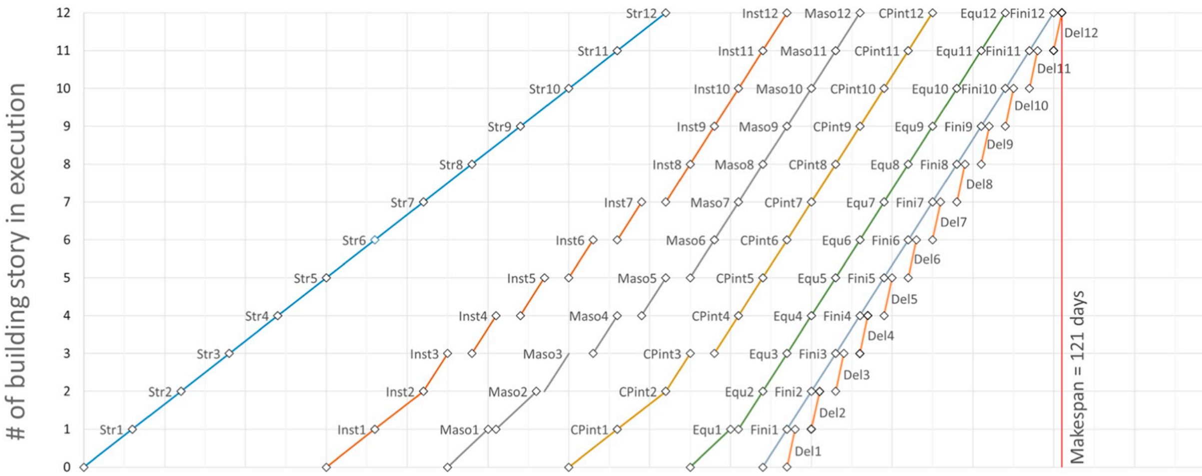

Fig. 5. Model 4 with one crew, non-continuity condition and acceleration routine.

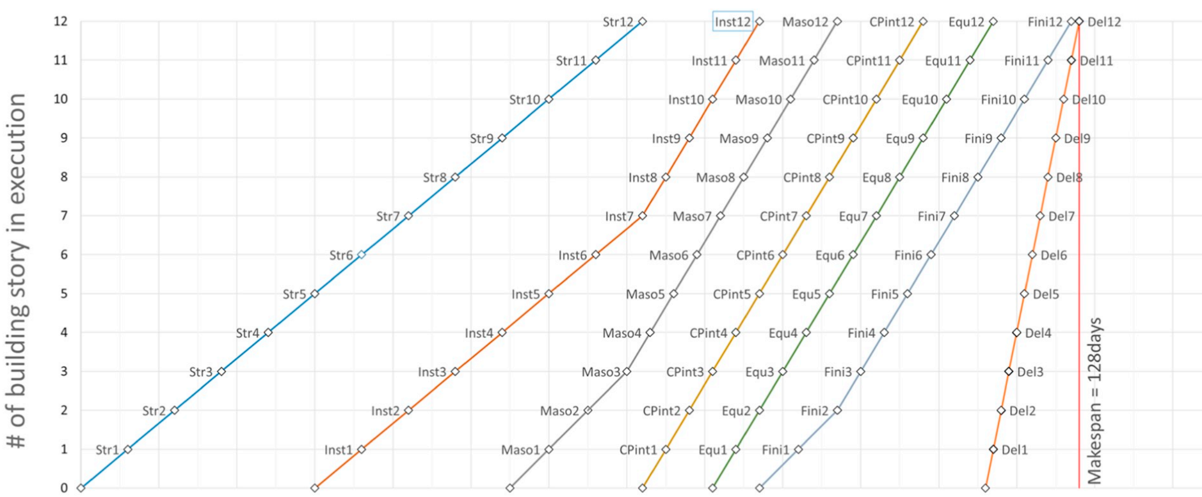

Fig. 6. Model 5 with one crew, continuity condition, acceleration routine, and resource consumption restriction.

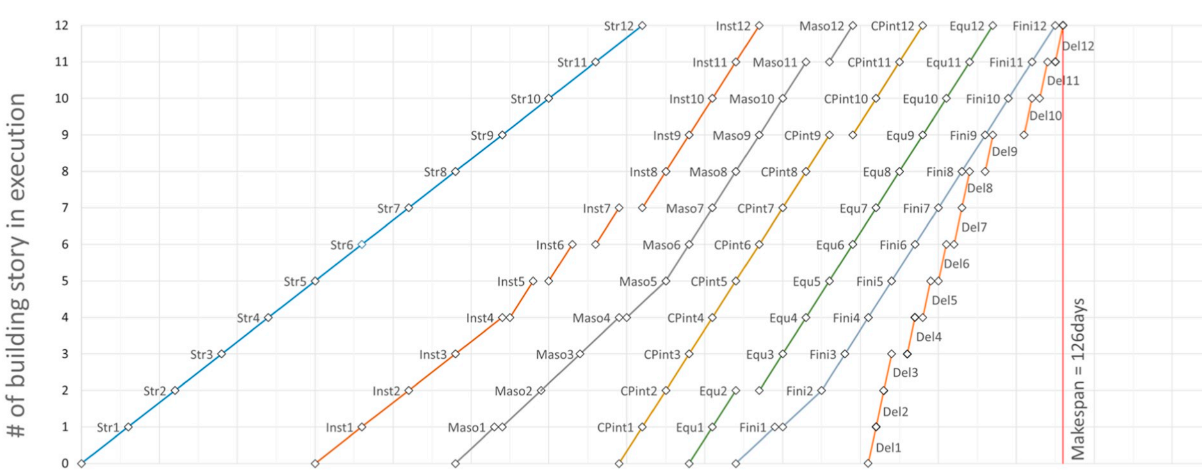

Fig. 7. Model 6 with one crew, non-continuity condition, acceleration routine, and resource consumption restriction.

by the implementation of novel linear optimization restrictions.

At first glance, if only the first four models are analyzed, it is clear that all the models provide the same makespan. Nonetheless, if the LSM graphs of these models are observed (Figs. 2 to 7), it can be identified that the different conditions considered in these four simulations contribute to different optimal schedules. On the one hand, models 1 and 3 provide a schedule where all activities are executed continuously. On the other, models 3 and 4 provide a project schedule with a controlled variation of the execution modes within the same activity. Depending on the analyzed project and under this particular situation, a practitioner may consider a continuous execution and a controlled acceleration routine may contribute to enhance the project execution efficiency, choosing model 4 as the best model.

Furthermore, analyzing Table 2 results, it is possible to infer from

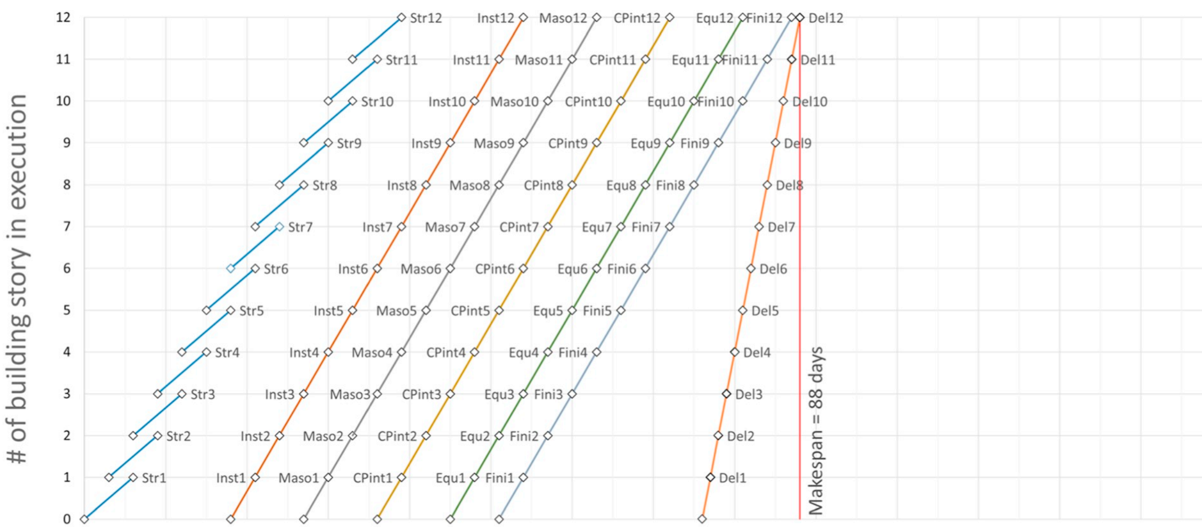

Fig. 8. Model 7 with multiple crews, non-accelerated routine, and continuity condition.

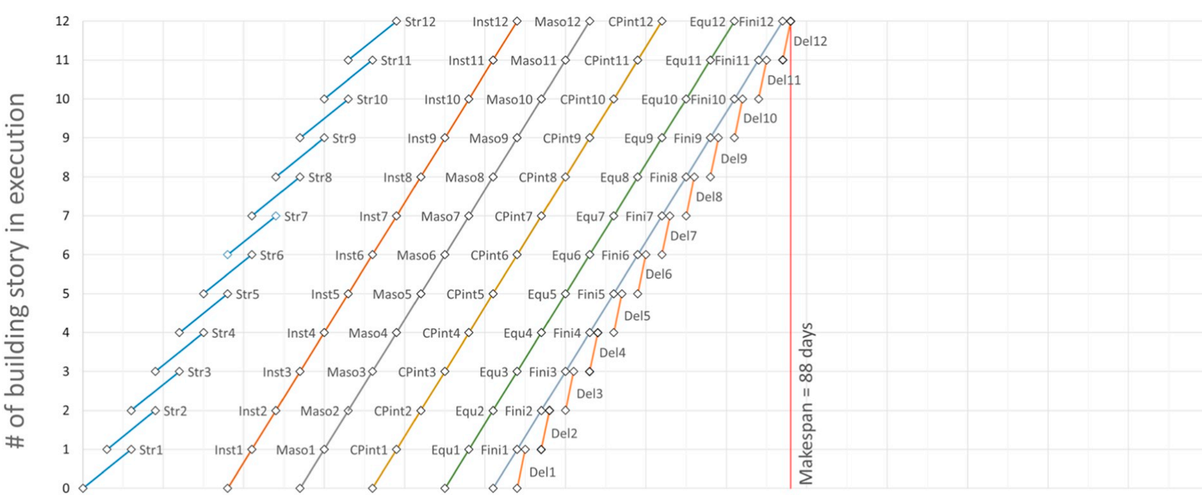

Fig. 9. Model 8 with multiple crews, non-accelerated routine, and non-continuity condition.

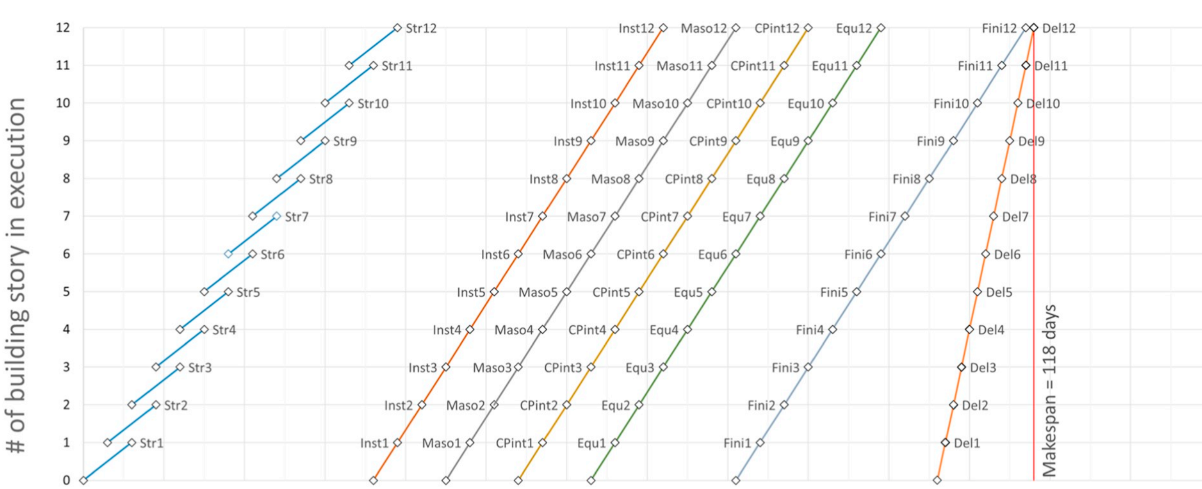

Fig. 10. Model 9 with multiple crews, continuity condition, non-accelerated routine, and resource consumption restriction.

models 5 and 6, the benefits of allowing activity fragmentation. In this particular experimentation, allowing discontinuous activity execution permitted the project schedule to gain two labor days under a resource-constrained scenario. Accordingly, models 7 to 10 clearly expose the benefit of working with multiple crews (Fig. 4). In one hand, models 7, 8 and 9 clearly show how project makespan is considerably reduced by allowing 2 simultaneous crews to work in the structure activity (sub-activity overlapping). On the other, even though model 10 reinforces this observation, it also exposes the rare schedule conditions that might emerge when limiting the resource consumption and allowing crew

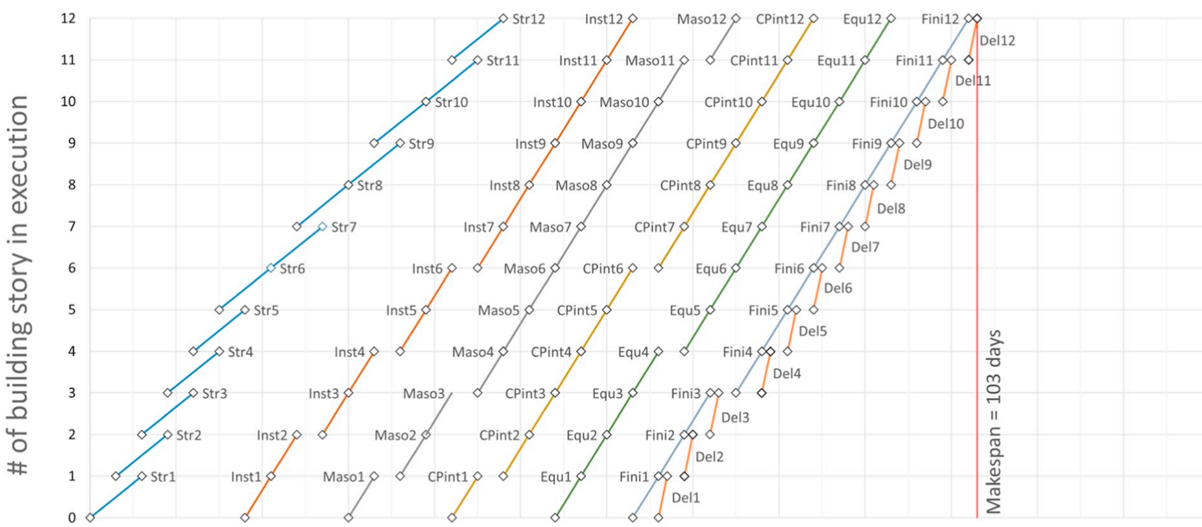

Fig. 11. Model 10 with multiple crews, non-continuity condition, non-accelerated routine, and resource consumption restriction.

fragmentation in a multiple crew condition. As one can observe, RCPSP conditions might challenge the working efficiency embodied in allowing two crews to work together, revealing that in some activity unit lengths multiple crews are not justified. In this type of scenarios, practitioners are forced to use their scheduling criterion to make the best execution decisions for the project. In models 7–10, all relationships within sub-activities of the same activity were build using multiple crew's restriction equations. Therefore, only the fastest execution modes of each activity were considered in the optimization process.

5. Conclusions, limitations, and recommendations for future research

Projects with a repetitive activity nature have continuously demanded the emergence of new models that fully take advantage of all the time and space information linear scheduling techniques can offer. The progressive implementation of relationships and lag times in the sub-activity level, continuity restrictions, multiple modes of sub-activity execution, and execution of activities with multiple crews, clearly depict practitioners' desire to progressively build more robust and realistic optimization tools.

Besides the nature of all the previously proposed models, authors consider there is still a need for a flexible mathematical linear programming formulation that allow practitioners to easily and jointly solve for the case of Repetitive Activities (RA) the Multimode Resource Constrained Problem Scheduling Problem (RA-MRCPSP) and the Multimode Time-Cost Tradeoff Problem Scheduling Problem (RA-MRCPSP), considering as many scheduling properties, benefits and challenges that linear scheduling of repetitive activities imply.

In order to fulfill this gap, this paper successfully presents an exact linear programming binary formulation specially designed to handle repetitive activities in construction projects. This research contributes to the repetitive activities scheduling body of knowledge in the following ways:

- It provides a novel mathematical formulation to optimally solve the resource allocation problem, MRCPSP and MTCTPSP with repetitive activities in construction projects considering multiple modes of execution, acceleration and deceleration controlled routines and multiple crews for each activity in a discretional way.
- It serves as a guideline for practitioners willing to implement linear programming in their scheduling process.
- It provides an extremely flexible mathematical model that can be easily be modified depending on the schedulers needs.
- It successfully implements a robust mathematical model in a real construction project, while considering as much linear scheduling characteristics as possible. It proves that relationships in the sub-activity level, continuity conditions, multiple modes of execution, controlled acceleration routines and execution mode shifts, and multiple crews are easily integrated by adding simple restrictions to the model.
- Schedules generated by the proposed mathematical model are robust and closer to real construction project conditions, especially useful for scheduling pipelines, highways, and high-rise buildings.

Lastly, authors have identified the following model's limitations: 1) the model does not handle several calendars, 2) only linear and non-interruptible sub-activities are considered, 3) the number of crews is an input variable and cannot be a result of the optimization process due the linearity conditions, 4) only traditional SS, SF, FS, FF relationships are considered, 5) if a very large number of sub-activities and mode of execution are considered, computational capabilities cannot be sustainable and might limit model implementation. As mentioned by Garcia-Nieves et al. [3] ‘metaheuristic models based on the proposed model can be developed in order to overcome computational limitation’.

Acknowledgments

The authors would like to thank the research group of Construction Engineering and Management (INgeco) of Los Andes University.

References

[1] P. Chakraborty, G.G. Roy, S. Das, D. Jain, A. Abraham, An improved harmony search algorithm with differential mutation operator, Fundamenta Informaticae 95 (4) (2009) 401–426, https://doi.org/10.3233/FI-2009-157.

[2] L. Florez, D. Castro-Lacouture, A.L. Medaglia, Sustainable workforce scheduling in construction program management, J. Oper. Res. Soc. 64 (8) (2012) 1169–1181, https://doi.org/10.1057/jors.2012.164.

[3] J.D. Garcia-Nieves, J. Ponz-Tienda, A. Salcedo-Bernal, E. Pellicer, The multimode resource-constrained project scheduling problem for repetitive activities in construction projects, Computer-Aided Civil and Infrastructure Engineering (2018) 1–17, https://doi.org/10.1111/mice.12356.

[4] K. El-Rayes, O. Moselhi, Optimazing resource utilization for repetitive construction projects, ASCE, Ed. J. Constr. Eng. Manag. (2001) 18–27, https://doi.org/10.1061/(ASCE)0733-9364(2001)127:1(18) January/Febraury.

[5] R. Harris, P. Ioannou, Scheduling projects with repeating activities, J. Constr. Eng. Manag. 124 (1998) 269–278, https://doi.org/10.1061/(ASCE)0733-9364(1998)124:4(269).

[6] K. Mattila, A. ASCE, A. Park, Comparison of linear scheduling model and repetitive scheduling method, J. Constr. Eng. Manag. 129 (1) (2003) 56–64, https://doi.org/10.1061/(ASCE)0733-9364(2003)129:1(56.

[7] R. Reda, RPM: repetitive project modeling, J. Constr. Eng. Manag. 116 (2) (1990) 316–330, https://doi.org/10.1061/(ASCE)0733-9364(1990)116:2(316).
[8] D. Arditi, O. Tokdemir, K. Suh, Challenges in line-of-balance scheduling, ASCE, Ed. J. Constr. Eng. Manag. (2002) 545–556, https://doi.org/10.1061/(ASCE)0733-9364(2002)128:6(545).
[9] K. El-Rayed, O. Moselhi, Resource-driven scheduling of repetitive activities, Constr. Manag. Econ. 16 (4) (2010) 433–446, https://doi.org/10.1080/014461998372213.
[10] Selinger, S. (1980). Construction Planning for Linear Projects. (ASCE, Ed.) J. Constr. Div., 106(2), 195–205. Retrieved from https://cedb.asce.org/CEDBsearch/record.jsp?dockey=0009573
[11] L.-H. Zhang, X. Zou, Repetitive Project Scheduling: Theory and Methods, Elsevier, 2015, https://doi.org/10.1016/C2013-0-18950-7.
[12] M. Hajdu, Continuous precedence relations for better modelling overlapping activities, Procedia Engineering 123 (2015) 216–223, https://doi.org/10.1016/j.proeng.2015.10.080.
[13] M. Hajdu, Survey of precedence relationships: classification and algorithms, Autom. Constr. 95 (2018) 245–259, https://doi.org/10.1016/j.autcon.2018.08.012.
[14] D. Johnston, Linear scheduling method for highway construction, J. Constr. Div. 107 (2) (1981) 247–261 Retrieved from https://cedb.asce.org/CEDBsearch/record.jsp?dockey=0010240.
[15] R. Carr, W. Meyer, Planning construction of repetitive building units, J. Constr. Div. 3 (1974) 403–412 Retrieved from https://cedb.asce.org/CEDBsearch/record.jsp?dockey=0021593.
[16] D. Halpin, R. Woodhead, Design of Construction and Process Operation, John Wiley & Sons Inc, New York, NY, USA, 1976 Retrieved from https://dl.acm.org/citation.cfm?id=578699.
[17] A. Russel, W. Caselton, Extensions to linear scheduling optimization, ASCE, Ed. J. Constr. Eng. Manag. 114 (1) (1988) 36–52, https://doi.org/10.1061/(ASCE)0733-9364(1988)114:1(36).
[18] H. Adeli, A. Karim, Scheduling/cost optimization and neural dynamics model for construction, J. Constr. Eng. Manag. 123 (1997) 450–458, https://doi.org/10.1061/(ASCE)0733-9364(1997)123:4(450).
[19] J.L. Ponz-Tienda, A. Salcedo-Bernal, E. Pellicer, A parallel branch and bound algorithm for the resource levelling problem with minimal lags, Computer-Aided Civil and Infrastructure Engineering (2016), https://doi.org/10.1111/mice.12233.
[20] J.L. Ponz-Tienda, V. Yepes, E. Pellicer, J. Moreno-Flores, The resource leveling problem with multiple resources using an adaptive genetic algorithm, Autom. Constr. 29 (2013) 161–172, https://doi.org/10.1016/j.autcon.2012.10.003.
[21] J. Ponz-Tienda, A. Salcedo-Bernal, E. Pellicer, J. Benlloch-Marco, Improved adaptive harmony search algorithm for the resource leveling problem with minimal lags, Autom. Constr. 77 (2017) 82–92, https://doi.org/10.1016/j.autcon.2017.01.018.
[22] J.L. Ponz-Tienda, E. Pellicer, J. Benlloch-Marco, C. Andrés-Romano, The fuzzy project scheduling problem with minimal generalized precedence relations, Computer-Aided Civil and Infrastructure Engineering 30 (11) (2015) 872–891, https://doi.org/10.1111/mice.12166.
[23] K. Hyari, K. El-Rayes, Optimal planning and scheduling for repetitive construction projects, J. Manag. Eng. 22 (2006) 11–19, https://doi.org/10.1061/(ASCE)0742-597X(2006)22:1(11).
[24] T. Hegazy, E. Kamarah, Efficient repetitive scheduling for high-rise construction, J. Constr. Eng. Manag. 134 (4) (2008) 253–264, https://doi.org/10.1061/(ASCE)0733-9364(2008)134:4(253).
[25] S.-S. Liu, C.-C. Wang, Optimizing linear project scheduling with multi-skilled crews, Autom. Constr. 24 (2012) 16–23, https://doi.org/10.1016/j.autcon.2011.12.009.
[26] S. Leu, S. Hwang, Optimal repetitive scheduling model with shareable resource constraint, J. Constr. Eng. Manag. 127 (4) (2001) 270–280, https://doi.org/10.1061/(ASCE)0733-9364(2001)127:4(270).
[27] M. Hsie, C.-J. Chang, I.-T. Yang, C.-Y. Huang, Resource-constrained scheduling for continuous repetitive projects with time-based production units, Autom. Constr. 18 (2009) 942–949, https://doi.org/10.1016/j.autcon.2009.04.006.
[28] S. Biruk, P. Jaskowski, Scheduling linear construction projects with constraints on resource availability, Arch. Civ. Eng. 63 (1) (2017), https://doi.org/10.1515/ace-2017-0001.
[29] M. Skibniewski, A. Armijos, Linear programming approach to construction equipment and labour assignments, Civ. Eng. Syst. 7 (1) (1990) 44–50, https://doi.org/10.1080/02630259008970569.
[30] O. Moselhi, K. El-Rayes, Scheduling of repetitive projects with cost optimization, J. Constr. Eng. Manag. 119 (1993) 681–697, https://doi.org/10.1061/(ASCE)0733-9364(1993)119:4(681).
[31] A. Senouci, N. Eldin, Dynamic programing approach to scheduling nonserial linear projects, Journal in Computing in Civil Engeneering 10 (1996) 106–114, https://doi.org/10.1061/(ASCE)0887-3801(1996)10:2(106).
[32] T. Hegazy, N. Wassef, Cost optimization in projects with repetitive nonserial activities, J. Constr. Eng. Manag. 127 (2001) 183–191, https://doi.org/10.1061/(ASCE)0733-9364(2001)127:3(183).
[33] K. El-Rayes, A. Kandil, Time-cost-quality trade-off analysis for highway construction, J. Constr. Eng. Manag. 131 (2005) 477–486, https://doi.org/10.1061/(ASCE)0733-9364(2005)131:4(477).
[34] K. Hyari, K. El-Rayes, M. El-Mashaleh, Automated trade-off between time and cost in planning repetitive construction projects, Constr. Manag. Econ. 27 (2009) 749–761, https://doi.org/10.1080/01446190903117793.
[35] S. Terry, G. Lucko, Algorithm for time-cost tradeoff analysis in construction projects by aggregating activity-level singularity functions, Construction Research Congress, 2012, https://doi.org/10.1061/9780784412329.024.
[36] X. Zou, Q. Zhang, L. Zhang, Modeling and solving the deadline satisfaction problem in line-of-balance scheduling, J. Manag. Eng. 34 (1) (2018), https://doi.org/10.1061/(ASCE)ME.1943-5479.0000565.
[37] X. Zou, L. Zhang, Q. Zhang, A Biobjective optimization model for deadline satisfaction in line-of-balance scheduling with work interruptions consideration, Math. Probl. Eng. (2018), https://doi.org/10.1155/2018/6534021.
[38] A.A. Pritsker, L.J. Waiters, P.M. Wolfe, Multiproject scheduling with limited resources: a zero-one programming approach, Manag. Sci. 16 (1) (1969) 93–108, https://doi.org/10.1287/mnsc.16.1.93.
[39] M. Ammar, LOB and CPM integrated method for scheduling repetitive projects, J. Constr. Eng. Manag. 30 (2013) 44–50, https://doi.org/10.1061/(ASCE)CO.1943-7862.0000569.
[40] H. Dolabi, A. Afshar, R. Abbasnia, CPM/LOB scheduling method for project deadline constraint satisfaction, Autom. Constr. 48 (2014) 107–118, https://doi.org/10.1016/j.autcon.2014.09.003.