Computer-Aided Civil and Infrastructure Engineering **00** (2018) 1–17

The Multimode Resource-Constrained Project Scheduling Problem for Repetitive Activities in Construction Projects

J. D. García-Nieves, J. L. Ponz-Tienda & A. Salcedo-Bernal

Department of Civil and Environmental Engineering, Universidad de Los Andes, Bogotá, Colombia

&

E. Pellicer*

School of Civil Engineering, Universitat Politècnica de València, Valencia, Spain

Abstract: *In construction projects, resource availability might limit the implementation of ideal schedules. Especially, when repetitive activities are involved, traditional resource-constrained project scheduling problem (RCPSP) models fail to allocate the resource consumption in an efficient manner. Besides, actual models only provide local optimal solutions and do not incorporate activity acceleration routines. To fulfill this gap, partially, a mathematical optimization model, the multimode RCPSP for repetitive activities in construction projects, is proposed and solved to optimality; it takes into account acceleration routines under real construction scenarios using spreadsheets. The article shows a complete computational experimentation over a real construction project, considering several scenarios of resource availabilities and continuity conditions. The model allows analyzing the resources efficiency indexes comparing them to resource consumptions, continuity of activities, and objective functions that reveal that fragmented activities do not provide better resource efficiency outcomes.*

1 INTRODUCTION

Increasing construction project efficiency in terms of time and cost is one of the most important factors that schedulers have in mind each time a new project starts (Chakraborty et al., 2009). This implies that practitioners and construction schedulers must take into account resource availability and its efficient consumption, besides the precedence relationships and location conditions. The inefficiency represented by resources consumption fluctuations is expensive and interrupts the workflow (El-Rayes and Jun, 2009; Koulinas and Anagnostopoulos, 2013). On the other hand, resource availability might limit the implementation of ideal and nonrestrictive schedules; in particular of construction processes, physical resources such as labor force and materials are restricted and might limit the scheduling of activities and construction processes (Benjaoran et al., 2015; Hinze, 2002).

In response to these challenges, literature has classified these scheduling problems as resource-constrained project scheduling problem (RCPSP) and resource leveling problem (RLP) (Harris, 1990; Icmeli, 1993; Özdamar and Ulusoy, 1995; Herroelen et al., 1998; Neumann and Zimmermann, 1999; Kolish and Padman, 2001; Demeulemeester and Herroelen, 2002; Hinze, 2002; Kim et al., 2005; Anagnostopoulos and Koulinas, 2010; Benjaoran et al., 2015; Ponz-Tienda et al., 2016; Herroelen and Leus, 2005; Ponz-Tienda et al., 2015). First, the RCPSP aims to minimize the project makespan (or the tardiness in terms of the sum of the finishing time of the activities), preserving the precedence constraints between activities and the resource

*To whom correspondence should be addressed. E-mail: *pellicer@upv.es*.

DOI: 10.1111/mice.12356

availability. Second, RLP focuses in project's activity rescheduling to minimize the resource consumption fluctuation under a nonrestrictive resource context without increasing a previously prescribed project deadline.

RCPSP and RLP belong to the set of NP-hard (nonpolynomic) problems, and consequently are affected by the "combinatorial explosion" phenomenon (Ponz-Tienda et al., 2016; Neumann et al., 2003; Rieck and Zimmermann, 2015). This implies that the universe of solutions arises exponentially as the considered number of activities is increased. When a large number of activities are involved, the computational capacity needed to solve these kinds of problems was simply unavailable with the current technology. Especially for RLP, exact algorithms are only efficient for small projects (Ponz-Tienda et al., 2016).

Moreover, the problem of scheduling repetitive activities in construction projects has been deeply studied since 1950s, when organization methods for assembly lines were introduced (Nezval, 1958). In practice, there are two main models that allow practitioners to understand the nature of repetitive activities: Line of Balance (LoB) and Linear Scheduling Method (LSM). On the one hand, LoB is a mixture between traditional and linear scheduling derived from activity-on-row networks, where practitioners can observe the delivery rate of finishing of repetitive activities (Su and Lucko, 2015). This method has been deeply studied since the 1970s (Lumsdem, 1968; O'Brien, 1969). On the other hand, LSM is a linear scheduling technique that allows practitioners to study production rates through the use of work quantity versus time graphs (Su and Lucko, 2015).

When LoB and LSM diagrams are used by construction schedulers to analyze the results obtained from traditional RCPSP models, where only finish-to-start relationships between activities were considered, a poor perception of the resources allocation over the space-time, even though resource availability is constrained, is perceived.

If resource restrictions per period are considered, RCPSP should not only minimize the project makespan, but also must help to analyze the efficient resource consumption under different scenarios. Nonetheless, the finish-to-start relationships considered by the traditional models do not allow practitioners to optimize projects taking into account repetitive activities relationships, such as activities overlapping, continuity between subactivities, and the linear dependences of subactivities with other activities.

To construct a feasible RCPSP approach for repetitive activities in construction projects, several heuristic and metaheuristic algorithms have been proposed to find local optimal solutions. However, to the best of the authors' knowledge, mathematical models considering overlapping, multimode activities execution, and linear scheduling for repetitive activities still have pending solutions in global optimality. Furthermore, existent models have not fully integrated the line of balance challenges proposed by Arditi et al. (2002), impeding them to achieve effectiveness. These challenges include: interdependencies among activities, dealing with constraints, nonlinear or discrete activities, learning curve effect, optimum crew size, acceleration routine, and cost optimization.

The acceleration routine challenge proposed by Arditi et al. (2002), a common and real problem that engineers face, and multimode RCPSP (MRCPSP) mathematical models that solve the problem to optimality are still waiting to be solved by the researchers' community in construction scheduling. To partially fulfill these gaps, the authors propose a new mathematical model for construction projects that specifically design to schedule repetitive activities. The formulation of the proposed model for the MRCPSP for repetitive activities (MRCPSP-RA) is based on the following principles: (1) MRCPSP solved to optimality, (2) implementation of realistic construction acceleration routines, (3) relationship between activities based on subactivities relationships and continuity restrictions, (4) discretional activity fragmentation, and (5) serve as an easy criterion to implement scheduling problems in interface such as Excel to generate global optimum solutions.

For the nature of this article, acceleration routines are conceived as controlled variations of the execution modes within an activity. Mainly, they are implemented to speed up the construction process. The authors' intention is to provide a model that allows different execution modes in each subactivity of a particular activity without incurring any inefficient hiring and firing conditions. Furthermore, global optimization of the MRCPSP-RA is achieved using Gurobi software engine, which uses linear programming algorithms, such as simplex and interior point-method algorithms, to explore all the feasible solutions of the linear optimization problem. Following this train of thought, the authors propose a mathematical model that consists of linear objective function and restrictions, which are solved through linear programming optimization.

To expose this proposal properly, this research is organized as follows: First, a literature review of the state-of-knowledge of the resource allocation and resource-constrained problems for repetitive activities in construction projects is shown. Section 3 details the proposed model for MRCPSP-RA. In Section 4, a real residential building construction project is used as an example of implementation to evaluate the

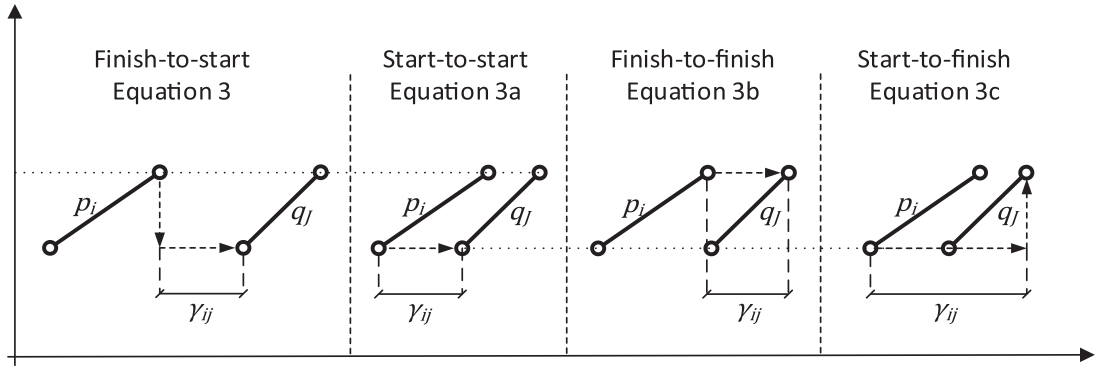

Fig. 1. Graphical representation of Equations (3)–(3c).

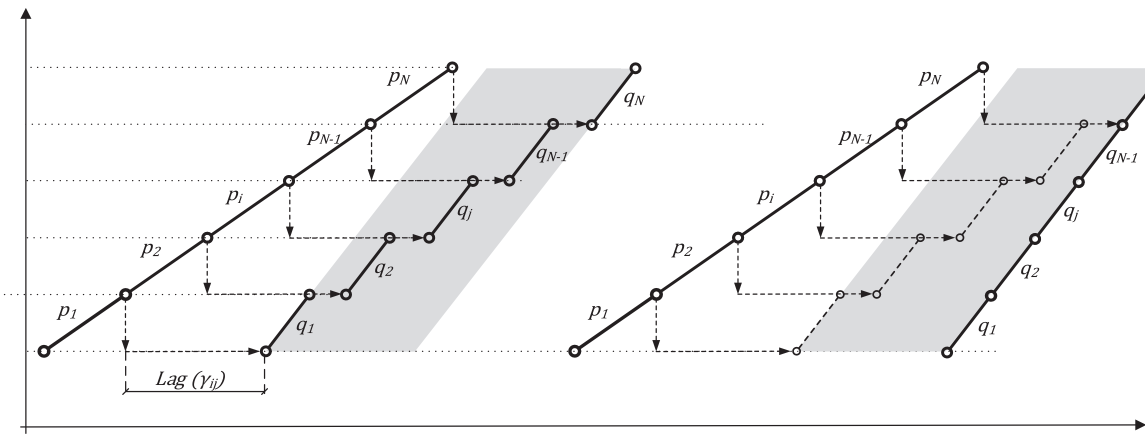

Fig. 2. LSM relationships with lag and continuity implications.

functionality of the model. Four scenarios, with a total of 60 instances, are considered and solved in the computational experimentation for different resource availability and continuity conditions. Finally, conclusions, limitations, and further research are drawn.

2 LITERATURE REVIEW

Several studies have been proposed to solve the problems surrounding the resource allocation in repetitive construction projects. These models were built to find the optimum starting time for all activities and subactivities to minimize a specific objective function. It is important to note that the authors understand subactivities as the fundamental unit of a repetitive activity; for example, a 12-story building, that has 12 identical floors, has 12 identical subactivities that compose the complete painting activity. Selinger (1980) formulated a one-state variable multimode model for repetitive construction based on continuous activity execution. The word "multimode" means the model conceived various possible modes of executions per activity. Each mode of execution has different crew size and subsequently, different execution durations and cost. Russel and Caselton (1988) improved the Selinger proposal considering a two-state variable model allowing activity fragmentation through the implementation of predetermined interruption vectors. Lately, to face the inefficiency produced by interruptions, El-Rayes and Moselhi (2001) proposed a dynamic multimode algorithm for scheduling of repetitive activities that considers, as part of the optimization process, the automatization of interruption options during the activity execution.

Adeli and Karim (1997) proposed a neural dynamics optimization nonlinear model (CONSCOM) to successfully schedule repetitive and nonrepetitive activities in construction projects. The resource allocation problem contemplates subactivity fragmentation, continuity condition, multiple crew-strategies, and effects of varying job conditions. Further, Senouci and Adeli

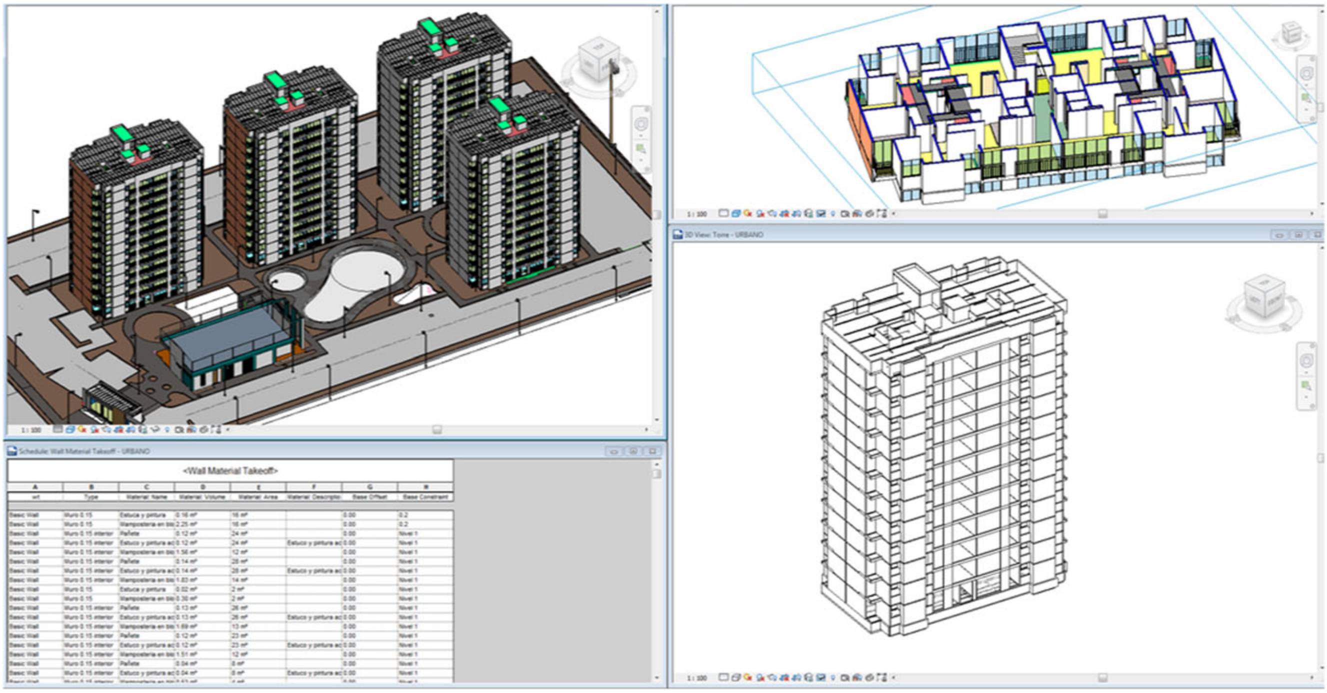

Fig. 3. Building information modeling example of implementation.

Table 1
Conceptual data model representation for the example of implementation

						Mode #1		Mode #2	
Activity	q_j	p_i	$p_i q_j$	γ_{piqj}	λ_{pq}	d_{q1}	a_{qk1}	d_{q2}	a_{qk2}
Structure	1			0	1	7	8	4	14
F. installations[1]	2	1	5	0	1	6	5	3	10
Masonries	3	2	3	0	1	5	8	3	13
Carpentry	4	3	3	0	1	6	3	3	6
Equipment	5	4	3	0	1	5	5	3	8
Finishing	6	5	2	0	1	5	10	3	17
R.S. delivery[2]	7	6	1	0	1	1	3	1	3

[1]Fixings installations.
[2]Real state delivery.

(2001) proposed a neural dynamic model for resource scheduling considering precedence relationships, multiple crew-strategies, and time cost trade-off. The model is mainly focused on the total project cost minimization that provides nearly optimal solutions applying artificial intelligence techniques. It considers, in a simultaneous way, cost and project leveling besides resource constraints. The novelty of the Senouci and Adeli (2001) proposal consists of the inclusion of the additional features such as total cost optimization, resource-constrained scheduling, and resource leveling, which outperform the Critical Path Method. Hegazy (2006) and Hegazy and Kamarah (2008) developed scheduling computerized models to solve the problem of repetitive activities optimum scheduling in infrastructure maintenance/repair programs and high-rise construction. In particular, efficient repetitive scheduling for high-rise construction was achieved through the implementation of a cost-based genetic algorithm, which successfully incorporates logical floor relationships, work continuity and crew synchronization, productivity factors, work interruptions, and resource constraints.

Liu and Wang (2012) improved existent models enabling multiple execution modes for subactivities and their discretional fragmentation. Zhang and Zou (2015) restructured the Liu and Wang (2012) resource allocation model to consider simultaneously generalized relationships for the subactivities, activity fragmentation, and one execution mode per activity. Due to the great number of decision variables that can be involved in the application of these models in a real construction context, heuristic and metaheuristic algorithms have been simultaneously proposed by Liu and Wang (2012) and Zhang and Zou (2015) to find local optimum solutions instead of global optimal solutions. More recently, Su and Lucko (2016) proposed the use of singularity functions to enhance linear scheduling resource allocation by suggesting a new approach that can unify the LSM and the LoB technique.

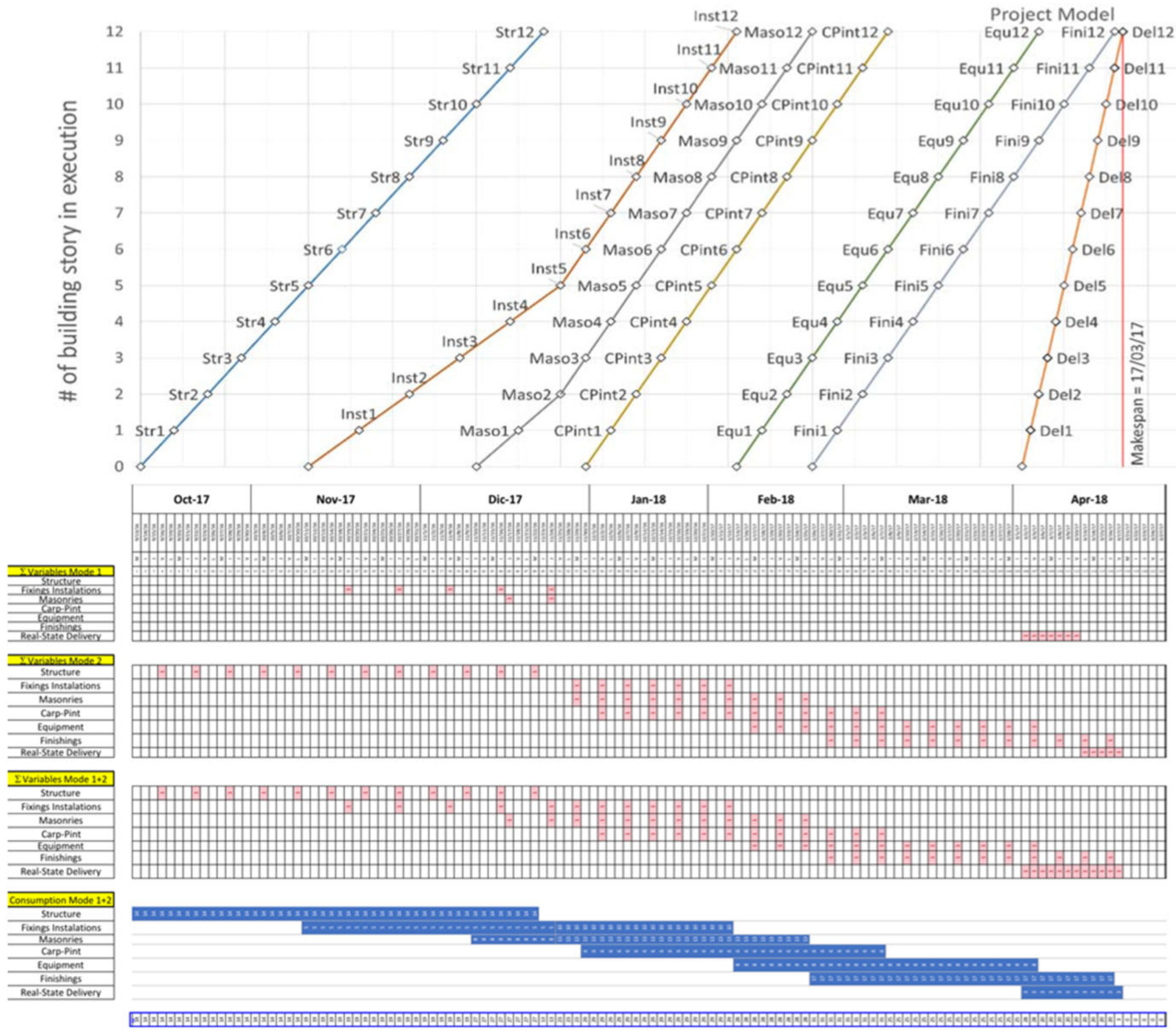

Fig. 4. Example of LSM, decision variables, and temporal diagram.

RCPSP and RLP have also been studied by several different authors in the context of repetitive construction projects with real variables (Leu and Hwang, 2001; Hsie et al., 2009; Zhang and Zou, 2015) and with integer variables (Ponz-Tienda et al., 2013; Ponz-Tienda et al., 2017). Metaheuristic models, built to face multiple intensity execution modes for the activities under resource availability restrictions (MRCPSP), have partially incorporated the previously exposed Arditi et al.'s (2002) challenges. Leu and Hwang (2001) proposed a metaheuristic algorithm to establish an optimal resource-constrained sequence based on activity continuity and only one optimum crew size per activity. Hsie et al. (2009) proposed an evolutionary algorithm that contemplates continuity restrictions, coexistence of multiple modes of execution in the same activity, and finish-to-start precedence relationships. In the Hsie et al. (2009) proposal, the execution mode shifting is not allowed at the finishing of each LoB segment but at the end of the time periods. Zhang and Zou (2015) incorporated the conditioned activity fragmentation, generalized precedence relationship constraints, and the coexistence of multiple execution modes; these authors returned to the original conception for shifting mode, in such way that they should be allowed only when LoB

Table 2
Minimization of objective function: makespan with allowed activity fragmentation

Resource availability	*Makespan*	*Tardiness*	*SSQR*	*SCDCR*	*SSDCDR*	*RIC*	*RID*	*RSCR*	*RCAR*
31	110	5,559	69,125	114	1,051	1,071	72	28%	78%
32	108	5,407	70,658	227	1,940	1,075	160	30%	77%
33	108	5,407	70,658	227	1,940	1,075	160	34%	75%
34	107	5,319	71,831	244	1,772	1,083	215	37%	73%
35	106	5,241	72,997	241	1,963	1,090	197	39%	72%
36	105	5,232	73,767	211	1,556	1,091	183	42%	70%
37	105	5,232	73,767	211	1,556	1,091	183	46%	69%
38	103	5,069	77,041	272	3,507	1,118	249	47%	68%
39	102	5,035	77,461	220	1,675	1,113	199	49%	67%
40	102	5,035	77,461	220	1,675	1,113	199	53%	65%
41	102	5,018	78,376	220	2,314	1,126	218	57%	64%
42	102	5,018	78,376	220	2,314	1,126	218	61%	62%
43	102	5,018	78,376	220	2,314	1,126	218	65%	61%
44	102	5,018	78,376	220	2,314	1,126	218	68%	59%
45	100	4,953	81,839	281	3,317	1,153	348	69%	59%

Table 3
Minimization of objective function: makespan with activities continuously executed

Resource availability	*Makespan*	*Tardiness*	*SSQR*	*SCDCR*	*SSDCDR*	*RIC*	*RID*	*RSCR*	*RCAR*
31	117	5,925	65,027	85	852	1,072	102	36%	73%
32	115	5,775	66,244	91	783	1,073	205	38%	72%
33	115	5,767	66,207	92	832	1,073	230	42%	70%
34	115	5,814	66,766	102	947	1,082	223	47%	68%
35	111	5,611	69,415	118	1,138	1,086	176	46%	69%
36	111	5,611	69,415	118	1,138	1,086	176	50%	67%
37	111	5,611	69,415	118	1,138	1,086	176	56%	65%
38	104	5,229	76,751	97	994	1,125	103	48%	67%
39	104	5,229	76,751	97	994	1,125	103	52%	66%
40	104	5,241	76,760	83	816	1,125	96	56%	64%
41	104	5,241	76,760	83	816	1,125	96	60%	62%
42	104	5,241	76,760	83	816	1,125	96	64%	61%
43	104	5,241	76,760	83	816	1,125	96	68%	60%
44	104	5,179	77,273	103	1,078	1,132	73	72%	58%
45	100	5,103	81,934	109	1,162	1,154	161	69%	59%

length segments (viewed as subactivities) are finished. Biruk and Jaskowski (2017) proposed a mixed linear mathematical model for the MRSCSP in repetitive construction projects that guarantee work continuity and optimal crew formation. Finally, Tang et al. (2018) proposed a constraint programming model based on logical relationships to tackle the schedule optimization of linear projects. The model shows great adaptability and flexibility to different optimization scenarios, including resource allocation problems, RLP, and MRCPSP.

In spite of all the advances to face the challenges proposed by Arditi et al. (2002), none of these models provide a holistic and global optimal solution for the MRCPSP in repetitive scheduling construction projects (MRCPSP-RA), considering activity fragmentation, activity acceleration routines, controlled coexistence of multiple execution modes in a same activity and subactivity interdependence with all nature of relationships for real and complex construction projects.

Table 4
Minimization of objective function: tardiness with allowed activity fragmentation

Resource availability	*Makespan*	*Tardiness*	*SSQR*	*SCDCR*	*SSDCDR*	*RIC*	*RID*	*RSCR*	*RCAR*
31	110	5,458	69,097	151	1,012	1,071	93	28%	78%
32	108	5,294	70,992	221	1,758	1,080	142	30%	77%
33	108	5,290	71,006	229	1,784	1,081	159	34%	75%
34	108	5,258	71,361	247	1,784	1,086	208	38%	73%
35	107	5,204	72,563	277	2,256	1,094	200	41%	71%
36	105	5,185	73,881	253	1,896	1,093	175	42%	70%
37	105	5,185	73,881	253	1,896	1,093	175	46%	69%
38	103	4,987	76,996	250	1,942	1,117	204	47%	68%
39	102	4,981	77,800	272	2,187	1,118	226	49%	67%
40	102	4,979	77,820	282	2,235	1,118	232	53%	65%
41	102	5,018	78,376	220	2,314	1,118	218	57%	64%
42	102	4,978	77,864	288	2,363	1,119	243	61%	62%
43	102	4,978	77,864	288	2,363	1,119	243	65%	61%
44	102	4,974	78,475	362	3,173	1,128	382	68%	59%
45	100	4,906	81,070	340	2,889	1,142	331	69%	59%

Table 5
Minimization of objective function: makespan with activities continuously executed

Resource availability	*Makespan*	*Tardiness*	*SSQR*	*SCDCR*	*SSDCDR*	*RIC*	*RID*	*RSCR*	*RCAR*
31	117	5,925	65,027	85	852	1,072	102	36%	73%
32	115	5,667	66,767	85	863	1,082	121	38%	72%
33	115	5,667	66,767	85	863	1,082	121	42%	70%
34	115	5,667	66,767	85	863	1,082	121	47%	68%
35	111	5,605	69,650	129	1,351	1,089	192	46%	69%
36	111	5,605	69,650	129	1,351	1,089	192	50%	67%
37	111	5,605	69,650	129	1,351	1,089	192	54%	65%
38	104	5,229	76,751	97	994	1,125	103	48%	67%
39	104	5,229	76,751	97	994	1,125	103	52%	66%
40	104	5,229	76,751	97	994	1,125	103	56%	64%
41	104	5,226	76,949	109	1,144	1,128	115	60%	62%
42	104	5,226	76,949	109	1,144	1,128	115	64%	61%
43	104	5,226	76,949	109	1,144	1,128	115	68%	60%
44	104	5,157	77,778	109	1,162	1,140	63	72%	58%
45	100	5,067	82,056	109	1,162	1,156	156	69%	59%

3 THE PROPOSED MRCPSP FOR REPETITIVE ACTIVITIES SCHEDULING IN CONSTRUCTION PROJECTS

The proposed model considers P number of activities in which identical amounts of N repetitive subactivities are executed in M_q different execution modes (different crew size). In this manner, the optimization model is developed to be applicable in repetitive constructive projects such as residential building projects, where activities like structure, facilities, architectural finishes, and mechanics fixing must be executed repetitively in each story.

This proposal implies the use of binary variables denoted as X_{qjt}, based on Pritsker et al. (1969) formulation, in such way that $X_{qjtm} = 1$ if a subactivity j executed in mode m of an activity q is finished in the period t, and $X_{qjtm} = 0$ otherwise. Additionally, dummy binary variables denoted as λ_q are used to indicate whether the execution of all the subactivities, which derive from an activity q, must be executed in a continuous way. Even though the use of integer variables is more natural than with binary ones, the use of binary decision variables is justified by the time-resource trade-off nature of the RCPSP problems.

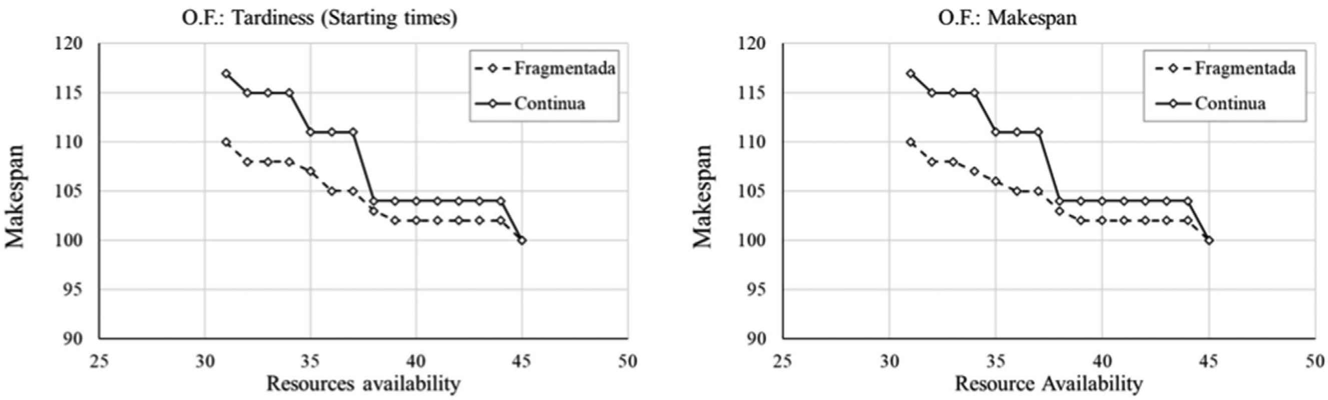

Fig. 5. Project makespan versus resource consumption under activity continuity and fragmented conditions.

The general formulation of the mathematical model considers the following parameters:

P	Number of activities of the project
N	Number of subactivities in each activity
M_q	Number of execution modes in each 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 subactivities 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 renewable resources of the project
a_{qkm}	Consumption of resource k in the 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:

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

Two different objective functions are taken into account in this model to optimize the construction schedule: the project makespan, computed as the finishing period of the latest subactivity in the latest activity; and the project tardiness, as the sum of the starting times of all the subactivities of the project (Equation (1)). Both objective functions represent the project duration. On one hand, the project makespan directly expresses the project duration. On the other hand, the project tardiness indirectly expresses the project duration in such way that, a larger tardiness indicates that some subactivities starting times have been postponed. This implicitly suggests that project duration or makespan should stay the same or increase (this relationship between the makespan and the tardiness depend on the activities and subactivities relationships):

$$\min \begin{cases} \text{makespan} & \sum_{m=1}^{M_P} \sum_{t=1}^{UB} X_{PNtm} \cdot t \\ \text{Tardiness} & \sum_{m=1}^{M_q} \sum_{q=1}^{P} \sum_{j=1}^{N} \left(\sum_{t=1}^{UB} X_{qjtm} \cdot t - d_{qm} \cdot \sum_{t=1}^{UB} X_{qjtm} \right) \end{cases} \tag{1}$$

Moreover, six restrictions must be added to the model to consider real construction conditions. First, subactivities precedence relationships in the same activity must be kept. In this manner, Equation (2) limits that a subactivity $j + 1$ in an activity q cannot start until the subactivity j in the same activity is completely finished (Figure 1):

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

Third, the activity precedence between subactivities of two interdependent activities is stated in Equation (3) in such way that a subactivity j in an activity q cannot start until a subactivity i located in a predecessor activity p is completely finished. Additionally, a lag time between these two related subactivities, described as $\gamma_{p_i q_j}$, is included (Figure 1):

$$\sum_{m=1}^{M_p} \left(\sum_{t=1}^{UB} X_{pitm} \cdot t \right) + \gamma_{p_i q_j} \leq \sum_{m=1}^{M_q} \left(\sum_{t=1}^{UB} (X_{qjtm} \cdot t) - d_{qm} \cdot \sum_{t=1}^{UB} X_{qjtm} \right) \forall i \ predecesor\ of j \tag{3}$$

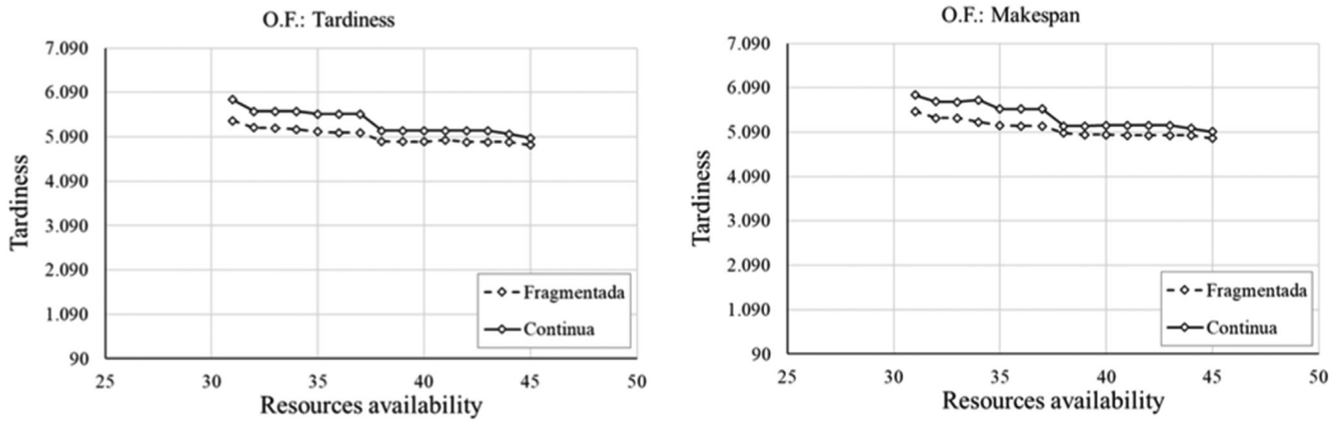

Fig. 6. Project tardiness versus resource consumption under activity continuity and fragmented conditions.

Equation (3) can be adjusted to deal with all nature of relationships as start-to-start, start-to-finish, and finish-to-finish. These modifications are exposed in Equations (3a)–(3c), respectively (Figure 1):

$$\sum_{m=1}^{M_p}\left(\sum_{t=1}^{UB}(X_{pitm}\cdot t)-d_{pm}\cdot\sum_{t=1}^{UB}X_{pitm}\right)+\gamma_{p_iq_j}\leq \sum_{m=1}^{M_q}\left(\sum_{t=1}^{UB}(X_{qjtm}\cdot t)-d_{qm}\cdot\sum_{t=1}^{UB}X_{qjtm}\right) \quad \forall\, i \text{ predecesor of } j \tag{3a}$$

$$\sum_{m=1}^{M_p}\left(\sum_{t=1}^{UB}(X_{pitm}\cdot t)-d_{pm}\cdot\sum_{t=1}^{UB}X_{pitm}\right)+\gamma_{p_iq_j}\leq \sum_{m=1}^{M_q}\left(\sum_{t=1}^{UB}X_{qjtm}\cdot t\right)\forall\, i \text{ predecesor of } j \tag{3b}$$

$$\sum_{m=1}^{M_p}\left(\sum_{t=1}^{UB}X_{pitm}\cdot t\right)+\gamma_{p_iq_j}\leq\sum_{m=1}^{M_q}\left(\sum_{t=1}^{UB}X_{qjtm}\cdot t\right) \quad \forall\, i\ predecesor\ of\ j \tag{3c}$$

To ensure that activities can be executed discretionally in a fragmented way or must be executed in a continuous way, one dummy variable per activity (λ_q) is used. If an activity q must be executed in a continuous way, $\lambda_q = 1$, and otherwise, $\lambda_q = 0$. In Equation (4), a discretional constant value of δ is included to guarantee that the restriction is always true (i.e., $\delta = -1e6$). Figure 2 illustrates the activity relationship constraints without continuity restriction (left-hand side) and under the continuity restriction on the activity execution (right-hand side). This restriction forces the subactivities to be executed one after the other (continuous activity execution) or not (fragmented activity execution).

$$\sum_{m=1}^{M_q}\left(\sum_{t=1}^{UB}X_{q1tm}\cdot t-d_{qm}\cdot\sum_{t=1}^{UB}X_{qitm}\right)+\delta\cdot\lambda_q\geq \sum_{m=1}^{M_q}\left(\sum_{t=1}^{UB}X_{qNtm}\cdot t\right)-\sum_{m=1}^{M_q}\sum_{j=1}^{N}\left(\sum_{t=1}^{UB}(X_{qjtm})\cdot d_{qm}\right)+\delta \quad \forall\, q\in\{1,\ldots,P\} \tag{4}$$

Equation (5) guarantees that each subactivity is only executed in one construction mode.

$$\sum_{m=1}^{M_q}\sum_{t=1}^{UB}X_{qjtm}=1 \quad \forall\, q\in\{1,\ldots,P\},\quad \forall\, j\in\{1,\ldots,N\} \tag{5}$$

Then, as denoted by Equation (6), the resource consumption of the whole project in one period cannot exceed the resource availability per period (u_k):

$$\sum_{q=1}^{P}\sum_{m=1}^{M_q}\left(a_{qkm}\cdot\sum_{x=t}^{x+d_{qm}-1}\sum_{j=1}^{N}X_{qjxm}\right)\leq u_k \quad \forall\, t\in\{1,\ldots,UB\},\forall\, k\in\{1,\ldots,k\} \tag{6}$$

Finally, to enhance the realistic and practical approach of the model, Equation (7) ensures that controlled acceleration routines are implemented. Controlled acceleration routine is understood as execution mode changes between subactivities that correspond to forward shifts in crew size. This concept emerges to eliminate random mode of execution implementation in each subactivity of an activity, which can result in inefficient hiring and firing conditions. It is important to note that the accelerations conceived by the proposed model are only due to crew size increase. The inclusion of acceleration due to learning is still a challenge that should be considered in further research:

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

The complete formulation for the mathematical model is stated ~~as follows:~~

$$\min f(x) = \begin{cases} \text{makespan} & \sum_{m=1}^{M_P} \sum_{t=1}^{UB} X_{PNtm} \cdot t \\ \text{Tardiness} & \sum_{m=1}^{M_q} \sum_{q=1}^{P} \sum_{j=1}^{N} \left(\sum_{t=1}^{UB} X_{qjtm} \cdot t - d_{qm} \cdot \sum_{t=1}^{UB} X_{qjtm} \right) \end{cases}$$

Subject to :

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

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

$$\sum_{m=1}^{M_q} \left(\sum_{t=1}^{UB} X_{q1tm} \cdot t - d_{qm} \cdot \sum_{t=1}^{UB} X_{qitm} \right) + \delta \cdot \lambda_q \geq \sum_{m=1}^{M_q} \left(\sum_{t=1}^{UB} X_{q\mathrm{N}tm} \cdot t \right) - \sum_{m=1}^{M_q} \sum_{j=1}^{N} \left(\sum_{t=1}^{UB} (X_{qjtm}) \cdot d_{qm} \right) + \delta \quad (8)$$
$$\forall q \in \{1, \ldots P\}$$

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

$$\sum_{q=1}^{P} \sum_{m=1}^{M_q} \left(a_{qkm} \cdot \sum_{x=t}^{x+d_{qm}-1} \sum_{j=1}^{N} X_{qjxm} \right) \leq u_k \quad \forall t \in \{1, \ldots, UB\}, \forall k \in \{1, \ldots, k\}$$

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

Note that objective function min $f(x)$ is adjusted for finish-to-start relationships, being necessary to include a dummy finishing activity P to include start-to-finish and start-to-start relationships. This activity has a duration of zero and it has predecessor activities as all the activities of the project. In the following case study, the real state delivery serves as a dummy variable even though its duration is not zero.

4 EXAMPLE OF IMPLEMENTATION

A real construction project was analyzed as an example of implementation to illustrate the versatility and applicability of the proposed MRCPSP algorithm. The project consists of four identical residential buildings. Each building has 12 stories; it is scheduled considering seven activities composed of 12 identical subactivities for a total of 84 work units. The activities considered are structure, facilities, masonries, carpentry and painting, equipment, finishing, and commissioning. For simplicity, the article only focuses on one of the building's construction schedules. The building project is shown in Figure 3.

The model requires 42,000 binary decision variables, considering an upper bound of 250 labor days. It has been implemented in Excel 2016, using Open Solver Optimizer V.2.8.6 and Gurobi Optimization Software V.7.0.2 (Academic License), requiring a computing time between 2 and 5,443 seconds. The computer used to

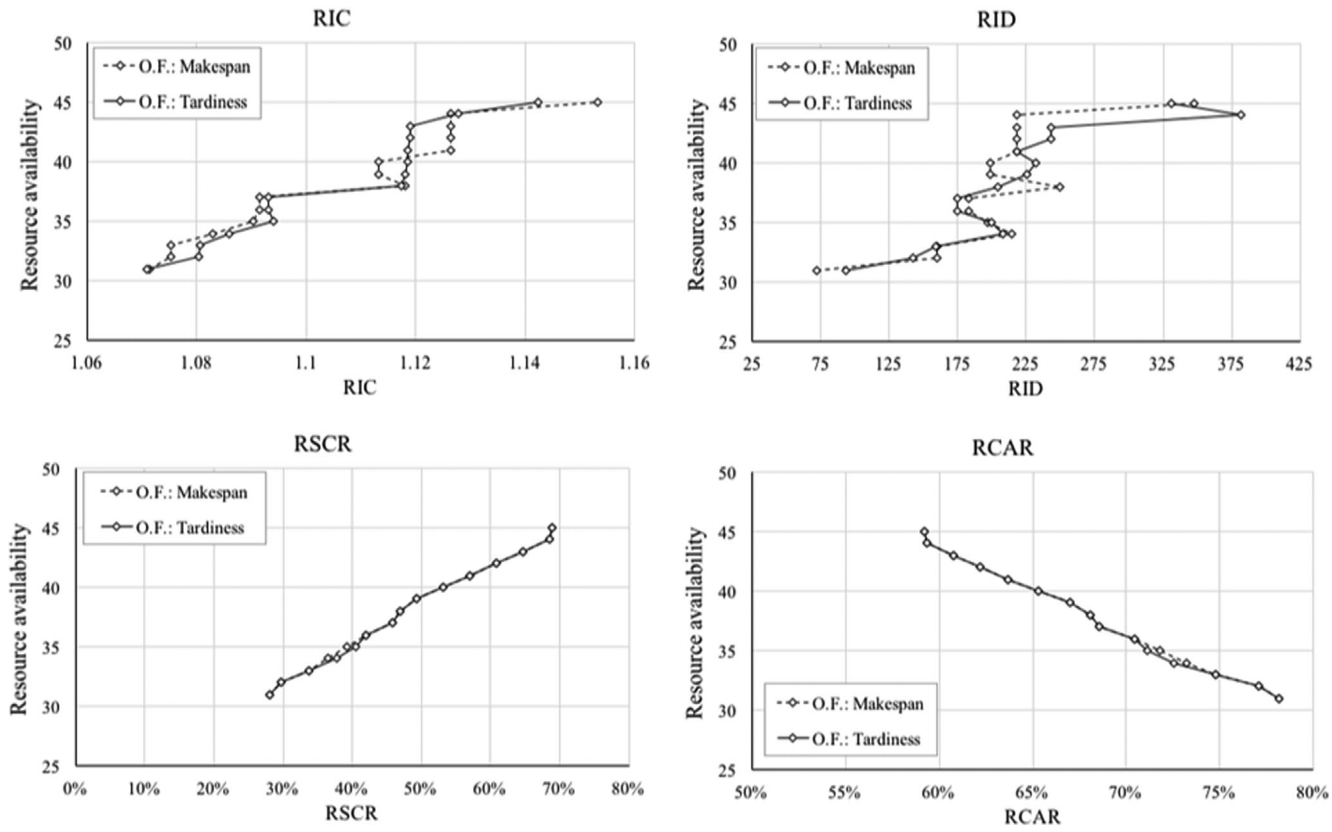

Fig. 7. Resource consumption efficiency indexes under different optimization objective functions (tardiness, makespan) for continuous activities.

solve the optimization model was a HP Z640 desktop, with an Intel Xeon E5-2637v4 3.5 2400 processor and a 32 GB DDR4-2400 (4 × 8 GB) 2CPU RegRAM.

All the relationships between subactivities of the same activities or dependent activities are finish-to-start relationships. Two construction modes (velocities) are proposed for subactivities belonging to each activity and only one construction acceleration routine per activity is accepted. Each mode will include the renewable resource consumption per period, expressed in normalized monetary units, and its duration. Also, activities can be restricted to be executed continuously or fragmentally (as considered by the different scenarios analyzed). Furthermore, the project scheduling calendar and LSM are developed in terms of labor days based on the Colombian labor calendar.

The precedence relationships and lag between activities, periodical resource consumption and duration per mode of execution, and the continuity conditions are defined by the practitioner depending on the desired scenarios. Table 1 displays the precedence relationships and resource consumption conditions of the project. As an example, continuous execution in all activities of the project is also proposed. As mentioned in the model description, each activity has a name, an assigned number q_j, and a predecessor p_i. Column $p_i q_j$ stands for the number of the subactivity i, in the predecessor activity p, that must be executed before subactivity j in activity q starts.

In Figure 4, the obtained result for an instance of the optimization problem for the example of implementation is displayed considering a resource availability of 37 monetary units per period, continuous activity execution, and the makespan as objective function. On the upper side of the figure, the optimized LSM project schedule combining different execution modes and acceleration routines is presented. In the middle, a summary table presenting the values of X_{qjtm} is displayed; in the lower side a Gantt diagram synthesizing daily resource consumption per activity is shown. The complete computational experimentation and quantitative results in Excel files can be downloaded from https://goo.gl/isjKMc.

To test the MRCPSP model proposed by the authors, four scenarios for the building construction scheduling are evaluated. The scenarios considered are: (1) minimization of project makespan with activity continuity restriction, (2) minimization of project makespan

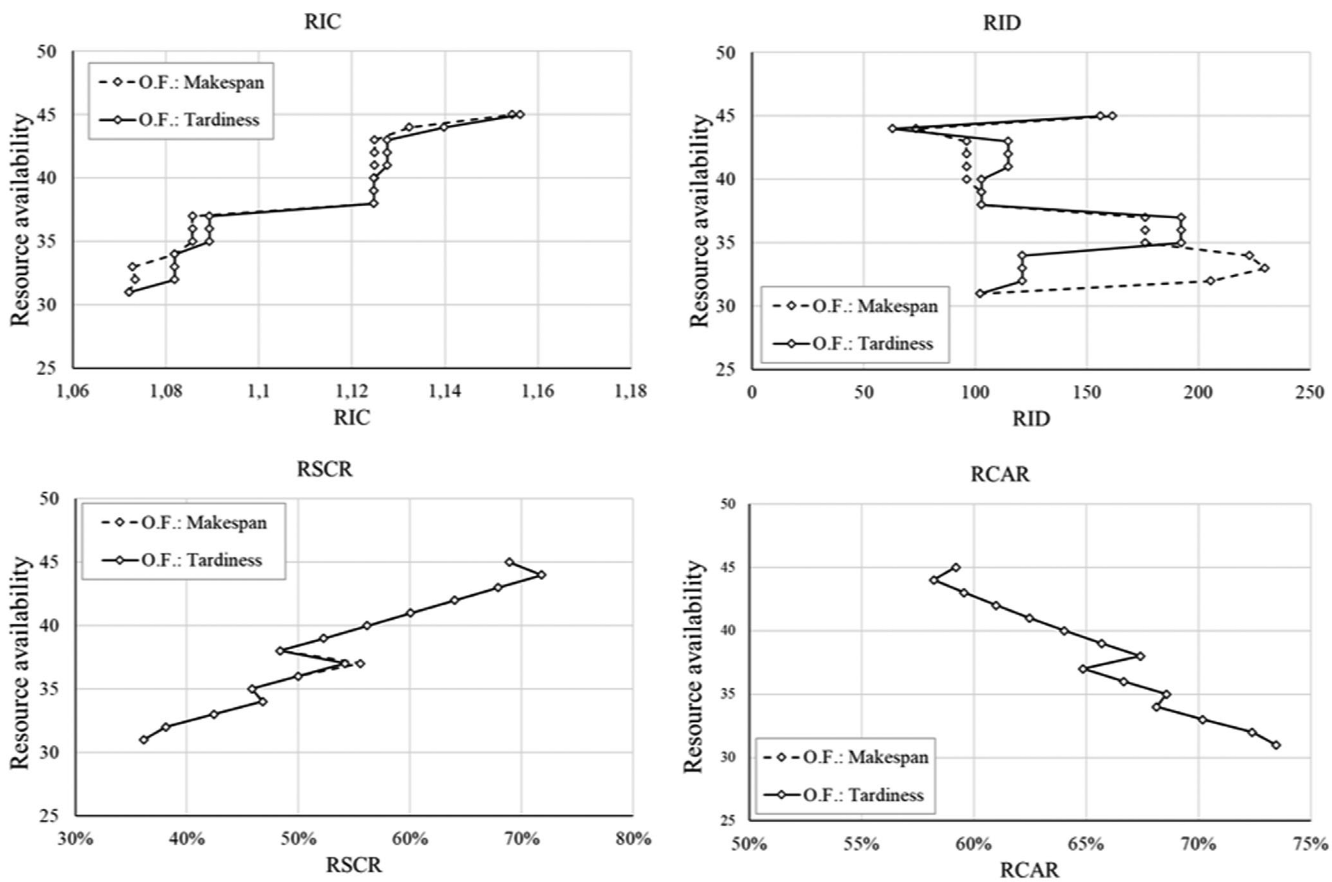

Fig. 8. Resource availability versus resource consumption efficiency indexes under different optimization objective functions (tardiness, makespan) for noncontinuous activities.

with allowed activity fragmentation, (3) minimization of project tardiness with continuity restriction, and (4) minimization of project tardiness with allowed activity fragmentation. Each scenario is evaluated under 15 different resource availability instances, from 31 to 45 resource units per period, yielding 60 solved instances.

Additionally, the resource efficiency consumption of the obtained schedules is measured throughout seven indicators: the sum of squares (SSQR) (Burguess and Killebrew, 1962), sum of differences of consecutive daily resources (SCDCR) (Florez et al., 2012), the sum of squares of differences of consecutive daily resources (SSDCDR) (Ponz-Tienda et al., 2017), the resource improvement coefficient (RIC) (Harris, 1978), the resource idle days (RID) (El-Rayes and Jun, 2009), and two new efficiency indexes developed by the authors: the resource surplus-consumption ratio (RSCR) and the resource consumption-availability ratio (RCAR).

Graphical representation and analysis of the efficiency indexes SSQR, SCDCR, and SSCDCR are not developed because these indexes are biased by the project makespan. In this order, they may provide erroneous information to practitioners when comparing two or more project schedules. (Generally, these indexes are only useful when the comparable schedules have the same project duration.) In contrast, indexes RIC, RID, RSCR, and RCAR truly allow authors to compare the efficiency associated with resource leveling and consumption in different projects.

The formulas to compute the previously mentioned efficiency indexes are shown in Equations (9)–(15), respectively. Note that in these equations, variable ε_{tk} refers to the project consumption of the resource k in a particular period of time t. The computation of ε_{tk} based on the binary formulation used by the proposed model is exposed in Equation (16):

$$\mathrm{SSQR}_k = \sum_{t=1}^{\mathrm{makespan}} {\varepsilon_{tk}}^2 \tag{9}$$

$$\mathrm{SCDCR}_k = \varepsilon_{tk} + \sum_{t=1}^{\mathrm{makespan}-1} \left|\varepsilon_{tk} - \varepsilon_{t+1,k}\right| + \varepsilon_{\mathrm{makespan},k} \tag{10}$$

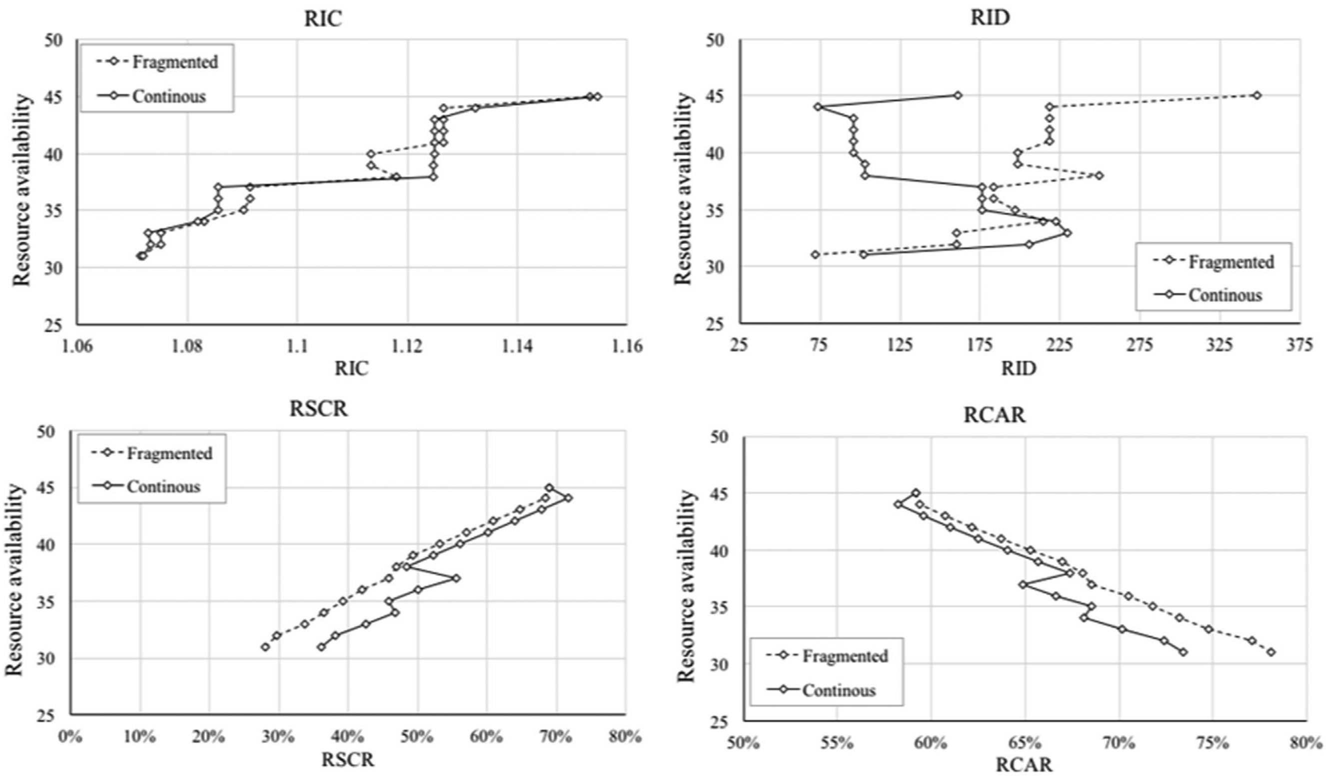

Fig. 9. Resource availability versus resource consumption efficiency indexes under two activity execution conditions (fragmented, continuous) for makespan objective function.

$$\mathrm{SSDCR}_k = \varepsilon_{tk}{}^2 + \sum_{t=1}^{\mathrm{makespan}-1} \left(\varepsilon_{tk} - \varepsilon_{t+1,k}\right)^2 + \varepsilon_{\mathrm{makespan},k}{}^2 \tag{11}$$

$$\mathrm{RIC}_k = \frac{\mathrm{Makespan} * \sum_{t=1}^{\mathrm{makespan}} \varepsilon_{tk}{}^2}{\left(\sum_{t=1}^{\mathrm{makespan}} \varepsilon_{tk}\right)^2} \tag{12}$$

$$\mathrm{RID}_k = \sum_{t=1}^{\mathrm{mak}} \min\left(\max\left(\varepsilon_{1k}, .., \varepsilon_{tk}\right) \times \max\left(\varepsilon_{tk}, ., \varepsilon_{\mathrm{makespan},k}\right)\right) - \varepsilon_{tk} \tag{13}$$

$$\mathrm{RSCR}_k = \frac{\text{Total resource } k \text{ surplus}}{\text{Total resource } k \text{ consumption}} = \frac{\sum_{t=1}^{\mathrm{makespan}} u_k - \varepsilon_{tk}}{\sum_{t=1}^{\mathrm{makespan}} \varepsilon_{tk}} \tag{14}$$

$$\mathrm{RCAR}_k = \frac{\text{Total resource } k \text{ consumption}}{\text{Total resource } k \text{ availability}} = \frac{\sum_{t=1}^{\mathrm{makespan}} \varepsilon_{tk}}{\mathrm{Makespan} * u_k} \tag{15}$$

$$\varepsilon_{tk} = \sum_{m=1}^{M} \sum_{q=1}^{N} \left(a_{qkm} \cdot \sum_{t=t}^{t+d_q-1} \sum_{j=1}^{N} X_{qjtm} \right) \tag{16}$$

The information regarding the project schedule tardiness, makespan in labor days, and resource efficiency consumption is registered for each instance and scenario of the computational experimentation. The scheduling results obtained after running the mathematical model under the four scenarios are shown in Tables 2–5. Table 2 shows the values for the makespan minimization as objective function with allowed activity fragmentation. The third column corresponds to the tardiness for the obtained makespan. Additionally, columns four to nine display the efficiency indexes for each one of the instances. Table 3 is similar to the second with the difference that all the activities are executed in a continuous way. Tables 4 and 5 show the same information as Tables 2 and 3 with the difference that tardiness is selected as the objective function.

Figure 5 shows that activity fragmentation, under all the studied scenarios, provides schedules with shorter makespan than the obtained when all activities are executed continuously. Also, under some resource availability conditions, the project makespan obtained

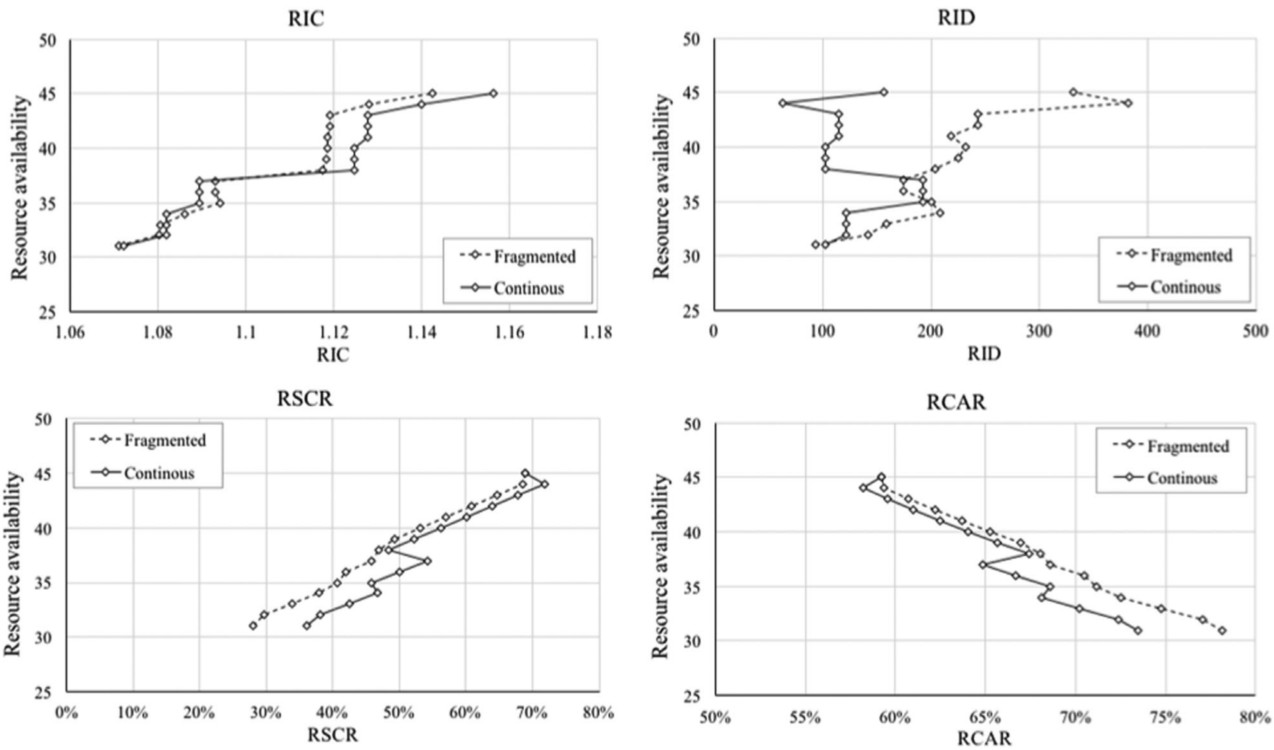

Fig. 10. Resource availability versus resource consumption efficiency indexes under two activity execution conditions (fragmented, continuous) for tardiness objective function.

after minimizing both objective functions under the same conditions are not equal. Makespan as objective function has shown that the optimized schedules obtained have equal or shorter makespan in comparison with the optimized schedules generated when tardiness is minimized.

Similarly, Figure 6 shows that activity fragmentation, under all the scenarios studied, also provides schedules with shorter tardiness. In addition, the project tardiness obtained, after minimizing both objective functions, is not equal under continuous and discontinuous conditions. The tardiness objective function has shown that the schedules obtained have equal or shorter tardiness in comparison with the schedules generated when the makespan is minimized. Figures 5 and 6 indicate that minimizing the project makespan is not equal to minimizing the tardiness as the sum of the activities starting times.

With the aim to analyze the correlation between the use of a specific objective function and the project's resource consumption efficiency, the authors have displayed valuable information in Figures 7–10. Figure 7 shows that, when activity fragmentation is allowed, there is no correlation that can state that the use of a specific objective function provides a better resource-consumption efficiency.

Similarly, Figure 8 shows that, when activity continuity execution is guaranteed, there is also no correlation that can state that the use of a specific objective function provides better efficiency in the resource consumption.

Additionally, Figure 9 (makespan as objective function) and Figure 10 (tardiness as objective function), indexes RSCR and RCAR, suggest that less monetary resource surplus is left behind when activities are allowed to fragmentize. Additionally, RIC index graphs propose there is no evidence that can state that fragmented activity conditions provide a better resource leveling shape in terms of uniform resource consumption. Finally, RID index suggests that fragmentation conditions result in better resource leveling conditions in terms of idle days, when soft resource availability restrictions are imposed. As resource availability is hardened, both continuity activity conditions tend to provide a very similar RID index. In general, as it can be observed in Figures 9 and 10, the index behavior in relation to the resource availability is very similar using makespan or tardiness as objective function.

As partial conclusion, Figures 7–10 suggest that a decrease in resource availability provides project schedules with a more efficient consumption of resources.

Regarding the acceleration routine implementation, the authors also registered the number of activities that

Table 6
Number of activities that applied acceleration routines in each scenario test

Resource availability	*Makespan*		*Tardiness*	
	Fragmented	*Continuous*	*Fragmented*	*Continuous*
31	2	3	2	3
32	2	4	2	3
33	2	4	2	3
34	3	4	3	3
35	4	5	4	5
36	4	5	4	5
37	4	5	4	5
38	2	3	2	3
39	3	3	2	3
40	3	3	2	3
41	4	3	4	4
42	4	3	2	4
43	4	3	2	4
44	4	4	2	3
45	1	2	1	2
Correlation coefficient	0.25	−0.49	−0.27	−0.12

applied an acceleration routine during the activity execution in each scenario. Table 6 shows that in all scenarios, the scheduling optimization has applied acceleration routines to activities execution to minimize the project makespan or tardiness.

Despite having achieved correlation coefficients up to −0.49 between resource availability and the number of activities with acceleration routines, there is not a clear linear correlation between these variables. It seems that the number of accelerated activities vary depending on the particular needs of a specific project, the schedule conditions, and the optimization needs.

5 CONCLUSIONS, LIMITATIONS, AND RECOMMENDATIONS FOR FUTURE RESEARCH

Creating efficient construction project schedules is one of the main priorities that every construction has during its planning stage. On many occasions, practitioners have to build efficient schedules to battle inefficient resource consumption fluctuations or restrictive resource availability.

In the case of construction projects that involve repetitive activities, practitioners and researchers have noticed that traditional RCPSP models failed to allocate in an efficient manner the resource consumption considering activity overlapping, fragmentation, multiple activity execution modes, and subactivities precedence relationships. To fulfill this gap, several authors have developed heuristic and metaheuristic models, especially shaped to overcome the challenges that emerged with repetitive activity scheduling. However, the literature review has exposed that the incorporation of activity acceleration routines, a common and real problem that engineers face, seems to be missing.

Currently, advanced MRCPSP models for repetitive activities in construction have achieved makespan minimization considering the optimum execution of crews and starting times for all subactivities. Nonetheless, wild fluctuations on crew sizes within a same activity may result in inefficient, problematic, or unrealistic construction schedules.

For this reason, the authors consider that the best way to simultaneously implement multiple execution modes in a same activity is through the use of unidirectional changes in optimum crew size throughout an activity execution, which are denoted as acceleration routines. Therefore, with the aim to build a more realistic MRCPSP model for repetitive activity scheduling, this article presents a holistic approach to the MRCPSPRA. This model allows practitioners to obtain optimal schedules based on project duration objective functions, which simultaneously provides a realistic approach of the MRCPSP through the implementation of acceleration routines.

The proposal computes the time and execution modes for all subactivities involved in the project based on continuity restrictions, precedence relationships, optimum crew size, and activities acceleration routines. This article contributes to the body of knowledge of construction project scheduling in several ways:

(1) It provides a more realistic MRCPSP model for repetitive activities that better fits to real construction projects.
(2) MRCPSP for repetitive activities solved to optimality.
(3) Controlled activity acceleration routines are incorporated to the MRCPSP-RA model.
(4) It introduces an easy criterion to implement scheduling problems in spreadsheets to generate global optimum schedule solutions, proposing a compact representation based on matrix that requires less information for the input variables than traditional models.
(5) It shows a complete computational experimentation over a real construction project, considering several scenarios of resource availabilities and continuity conditions.
(6) It analyzes the resource efficiency indexes comparing them to resource consumptions,

continuity of activities, and objective functions that reveal that fragmented activities do not provide better resource efficiency profiles.

The model can benefit current industrial construction processes such as bridge and high-rise building construction among others. The case of study is a real example of how the mathematical model proposed can be used to schedule activities in high-rise housing complexes in low socioeconomic districts. The scope of this type of project and the industrial process used to build the infrastructure largely promote the occurrence of repetitive activities that can be scheduled with the proposed mathematical model.

Finally, the authors identify eight model limitations that open space for future research. First, the mathematical model proposed does not consider soft logic scheduling optimization. Second, even though the model can be adjusted to contemplate nonrepetitive activities, it does not consider them. Third, only identical subactivity units are considered from a practical point of view, but this issue can be included easily. Fourth, negative acceleration routines (deceleration) are not allowed by the mathematical model restrictions. Fifth, subactivity segments are assumed to be linear and noninterruptible. Sixth, only minimal start-to-start, start-to-finish, finish-to-start, finish-to-finish relations are allowed. Seventh, calendar effect on the scheduling process is not allowed. Finally, a larger number of activities and execution modes can result in a high computational effort that may limit the model application. This limitation can be partially solved by applying parallel computing to improve the computational cost. In addition, metaheuristic models based on the proposed model can be developed to overcome computational limitations.

ACKNOWLEDGMENTS

This research was partially supported by the FAPA program of Universidad de Los Andes, Colombia (code P14.246922.005/01). The authors would also like to thank the research group of Construction Engineering and Management (INgeco) at Universidad de los Andes.

REFERENCES

Adeli, H. & Karim, A. (1997), Scheduling/cost optimization and neural dynamics model for construction, *Journal of Construction Engineering and Management*, **123**, 450–58.

Anagnostopoulos, K. P. & Koulinas, G. K. (2010), A simulated annealing hyperheuristic for construction resource levelling, *Construction Management and Economics*, **28**(2), 163–75. https://doi.org/10.1080/01446190903369907.

Arditi, D., Tokdemir, O. & Suh, K. (2002), Challenges in line-of-balance scheduling, *Journal of Construction Engineering and Management*, **128**(6), 545–56.

Benjaoran, V., Tabyang, W. & Sooksil, N. (2015), Precedence relationship options for the resource levelling problem using a genetic algorithm, *Construction Management and Economics*, **33**(9), 711–23. https://doi.org/10.1080/01446193.2015.1100317.

Biruk, S. & Jaskowski, P. (2017), Scheduling linear construction projects with constraints on resource availability, *Archives of Civil Engineering*, **63**(1), 3–15.

Burguess, A. & Killebrew, J. (1962), Variation in activity level on a cyclic arrow diagram, *Journal of Industrial Engineering*, **13**(2), 76–83.

Chakraborty, P., Roy, G. G., Das, S., Jain, D. & Abraham, A. (2009), An improved harmony search algorithm with differential mutation operator, *Fundamenta Informaticae*, **95**(4), 401–26. https://doi.org/10.3233/FI-2009-157.

Demeulemeester, E. & Herroelen, W. (2002). *Project Scheduling: A Research Handbook*, Kluwer Academic Publishers, Dordrecht, The Netherlands.

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

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

Florez, L., Castro-Lacouture, D. & Medaglia, A. L. (2012), Sustainable workforce scheduling in construction program management, *Journal of the Operational Research Society*, **64**(8), 1169–81. https://doi.org/10.1057/jors.2012.164.

Harris, R. (1978), *Precedence and Arrow Networking Techniques for Construction*, Wiley, New York.

Harris, R. (1990), Packing method for resource leveling, *Journal of Construction Engineering and Management*, **116**(2), 331–50.

Hegazy, T. (2006), Computerized system for efficient delivery of infrastructure maintenance/repair programs, *Journal of Construction Engineering and Management*, **132**(1), 26–34.

Hegazy, T. & Kamarah, E. (2008), Efficient repetitive scheduling for high-rise construction, *Journal of Construction Engineering and Management*, **134**(4), 253–64.

Herroelen, W. & Leus, R. (2005), Project scheduling under uncertainty: survey and research potentials, *European Journal of Operational Research*, **165**, 289–306.

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

Hinze, J. (2002), *Construction Planning and Scheduling*, 4th edn., Prentice Hall, Upper Saddle River, NJ.

Hsie, M., Chang, C.-J., Yang, I.-T. & Huang, C.-Y. (2009), Resource-constrained scheduling for continuous repetitive projects with time-based production units, *Automation in Construction*, **18**, 942–49.

Icmeli, O. E. (1993), Project scheduling problems: a survey, *International Journal of Operations & Production Management*, **13**, 80–91.

Kim, J., Kim, K. & Jee, N. (2005), Enhanced resource leveling technique for project scheduling, *Journal of Asian Architecture and Building Engineering*, **4**(2), 461–66.

Kolish, R. & Padman, R. (2001), An integrated survey of deterministic project scheduling, *Omega The International Journal of Management Science*, **29**(3), 249–72.

Koulinas, G. & Anagnostopoulos, K. (2013), A new tabusearch-based hyper-heuristic algorithm for solving construction leveling problems with limited resource availabilities, *Automation in Construction*, **31**, 169–75.

Leu, S. & Hwang, S. (2001), Optimal repetitive scheduling model with shareable resource constraint, *Journal of Construction Engineering and Management*, **127**(4), 270–80.

Liu, S.-S. & Wang, C.-C. (2012), Optimizing linear project scheduling with multi-skilled crews, *Automation in Construction*, **24**, 16–23.

Lumsdem, P. (1968), *The Line-of-Balance Method*, Pergamon Press Ltd., Oxford/New York.

Neumann, K. & Zimmermann, J. (1999), Resource levelling for projects with schedule-dependent time windows, *European Journal of Operational Research*, **117**, 591–605.

Neumann, K., Schwindt, C. & Zimmermann, J. (2003), *Project Scheduling with Time Windows and Scarce Resources*, Springer, Berlin.

Nezval, J. (1958), *Foundations of Flow Production in Construction*, VEB Verlag für Bauwesen, Berlin.

O'Brien, J. (1969), *Scheduling Handbook*, McGraw-Hill Inc., New York.

Özdamar, L. & Ulusoy, G. (1995), A survey on the resource-constrained project scheduling problem, *IIE Transactions*, **27**, 574–86.

Ponz-Tienda, J. L., Pellicer, E., Benlloch-Marco, J. & Andrés-Romano, C. (2015), The fuzzy project scheduling problem with minimal generalized precedence relations, *Computer-Aided Civil and Infrastructure Engineering*, **30**(11), 872–91. https://doi.org/10.1111/mice.12166.

Ponz-Tienda, J. L., Salcedo-Bernal, A. & Pellicer, E. (2016), A parallel branch and bound algorithm for the resource levelling problem with minimal lags, *Computer-Aided Civil and Infrastructure Engineering*, **32**(6), 474–98. https://doi.org/10.1111/mice.12233.

Ponz-Tienda, J. L., Salcedo-Bernal, A., Pellicer, E. & Benlloch-Marco, J. (2017), Improved adaptive harmony search algorithm for the resource leveling problem with minimal lags, *Automation in Construction*, **77**, 82–92. https://doi.org/10.1016/j.autcon.2017.01.018.

Ponz-Tienda, J. L., Yepes, V., Pellicer, E. & Moreno-Flores, J. (2013), The resource leveling problem with multiple resources, *Automation in Construction*, **29**, 161–72. https://doi.org/10.1016/j.autcon.2012.10.003.

Pritsker, A. A., Waiters, L. J. & Wolfe, P. M. (1969), Multi-project scheduling with limited resources: a zero-one programming approach, *Management Science*, **16**(1), 93–108. https://doi.org/10.1287/mnsc.16.1.93.

Rieck, J. & Zimmermann, J. (2015), Exact methods for resource leveling problems, in C. Schwindt and J. Zimmermann (eds.), *Handbook on Project Management and Scheduling*, vol. 1, Springer International Publishing, Switzerland, pp. 361–87.

Russel, A. & Caselton, W. (1988), Extensions to linear scheduling optimization, *Journal of Construction Engineering and Management, ASCE*, **114**(1), 36–52.

Selinger, S. (1980), Construction planning for linear projects, *Journal of Construction Division*, *ASCE*, **106**(2), 195–205.

Senouci, A. & Adeli, H. (2001), Resource scheduling using neural dynamics model, *Journal of Construction Engineering and Management*, **127**, 28–34.

Su, Y. & Lucko, G. (2015), Comparison and renaissance of classic line-of-balance and linear schedule concepts for construction industry, *Procedia Engineering*, **123**, 546–56.

Su, Y. & Lucko, G. (2016), Linear scheduling with multiple crews based on line-of-balance and productivity scheduling method with singularity functions, *Automation in Construction*, **70**, 38–50.

Tang, Y., Liu, R. & Wang, F. (2018), Scheduling optimization of linear schedule with constraint programming, *Computer-Aided Civil and Infrastructure Engineering*, **33**(2), 124–51.

Zhang, L.-H. & Zou, X. (2015), *Repetitive Project Scheduling: Theory and Methods*, Elsevier, Oxford.