



Improved Adaptive Harmony Search algorithm for the Resource Leveling Problem with minimal lags



J.L. Ponz-Tienda ^a, A. Salcedo-Bernal ^a, E. Pellicer ^{b,*}, J. Benlloch-Marco ^c

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

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

^c Department of Architectural Construction, Universitat Politècnica de València, Valencia, Spain

ARTICLE INFO

Article history:

Received 28 June 2016

Received in revised form 17 January 2017

Accepted 20 January 2017

Available online xxxx

Keywords:

Algorithm

Benchmarking

Construction project

Harmony Search

Resource Leveling Problem

ABSTRACT

The Resource Leveling Problem (RLP) aims to provide the most efficient resource consumption as well as minimize the resource fluctuations without increasing the prescribed makespan of the construction project. Resource fluctuations are impractical, inefficient and costly when they happen on construction sites. Therefore, previous research has tried to find an efficient way to solve this problem. Metaheuristics using Harmony Search seem to be faster and more efficient than others, but present the same problem of premature convergence closing around local optimums. In order to diminish this issue, this study introduces an innovative Improved and Adaptive Harmony Search (IAHS) algorithm to improve the solution of the RLP with multiple resources. This IAHS algorithm has been tested with the standard Project Scheduling Problem Library for four metrics that provide different leveled profiles from rectangular to bell shapes. The results have been compared with the benchmarks available in the literature presenting a complete discussion of results. Additionally, a case study of 71 construction activities contemplating the widest possible set of conditions including continuity and discontinuity of flow relationships has been solved as example of application for real life construction projects. Finally, a visualizer tool has been developed to compare the effects of applying different metrics with an app for Excel. The IAHS algorithm is faster with better overall results than other metaheuristics. Results also show that the IAHS algorithm is especially fitted for the Sum of Squares Optimization metric. The proposed IAHS algorithm for the RLP is a starting point in order to develop user-friendly and practical computer applications to provide realistic, fast and good solutions for construction project managers.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Project scheduling problems (PSP) are NP-Hard optimization problems that comprise resource-constrained problems (RCPSP) and Resource Leveling Problems (RLP) among others [10,27,32]. In the former, resources are considered a constraint, and in the latter, the problem is the efficiency in resource consumption. Both problems are similar but of different natures. The RCPSP is a regular problem with the objective of minimizing the project makespan without exceeding the resource availability. The RLP is a non-regular problem with the objective of providing the most efficient resource consumption and reducing the resource fluctuations without increasing the prescribed makespan of the project.

Resource fluctuations are impractical, inefficient and costly when they happen on construction sites [13,24]; therefore, increasing the efficiency of the project sequence is one key factor to achieve the project

goals [9]. In order to measure the efficiency of the project sequence, different metrics for the objective function have been proposed along the literature [8]. The first and most common objective function is the so called minimum squares optimization method or Sum of Squares Optimization (SSQR) method, which aims to provide an ideal uniform shape, minimizing the sample variance of resources consumption.

In construction projects a bell-shaped resource profile would be better from a practical point of view [37], so other metrics have been proposed by El-Rayes & Jun ([13,14]. The first authors proposed the Resource Idle Days and Maximum Daily Resource Demand Method to provide nearly gauss shape, eliminating the resources' idle periods. Florez, Castro-Lacouture, & Medaglia [14] proposed maximizing labor stability, aiming to increase the extent of use of workers and job continuity, and minimizing the maximal fluctuation of workers and the sum of the fluctuations.

Exact algorithms based upon enumeration, integer programming or mixed integer programming have been proposed to offer optimal solutions, but this kind of problems have a phenomenon of "combinatorial explosion" or rapid non-polynomial increase in the number of possible solutions, especially for large strong problems [32]. Although these

* Corresponding author.

E-mail address: pellicer@upv.es (E. Pellicer).

algorithms produce the absolute optimum to a given problem, they are only functional from a practical point of view for small problems, as they require a vast computational capability and complex parallel processing of the network graph [31].

Alternative heuristic algorithms have been proposed to find near optimal solutions in an acceptable computational effort as the Burgess & Killebrew [4] algorithm, or the Minimum Moment (MOM) and Packing Method (PACK) proposed by Harris in 1978 and 1990 respectively. When the complexity of the problem increases, heuristic approaches fail to produce near optimum solutions, arising nature inspired algorithms, known as metaheuristics that apply smart searching strategies over a population of solutions (population-based) or evaluating only one potential solution (neighborhood-based) [35].

Numerous metaheuristics have been developed in the past years to solve complex optimization models. A new family of population-based metaheuristic inspired on the composition and improvisation process of jazz musicians, known as Harmony Search (HS) [17], has acquired special relevance. Harmony is synonymous to proper rhythm, the opposite of the dissonance and the anarchy; in other words, harmony seeks to provide the sequence of sounds that best fit with an ideal of sonorous beauty, or its equivalent in optimization problems, an objective function represented by a metric of efficiency.

Harmony Search (HS) seems to be faster and more efficient than other metaheuristics [28] avoiding the premature convergence and relapse into local optimums, but proposals with the application of HS algorithms in construction project scheduling are still scarce and limited to small projects. Aiming to prove the goodness of Harmony Search (HS) algorithms in construction project scheduling problems, this research adapts the Improved Harmony Search algorithm proposed by Chakraborty et al. [6], taking into consideration the adaptive adjustment of parameters proposed by Mahdavi et al. [25] in an Improved and Adaptive Harmony Search (IAHS) algorithm. This IAHS algorithm is tested solving the j30, j60 and j120 instances of the Project Scheduling Problem Library (PSPLIB) [23] for four different metrics, comparing the obtained results with benchmarks available in the literature. Additionally, as case study of a real life construction projects, a building project of 15 floors [30,31], is used to illustrate the versatility and adaptability of the proposed IHSA for the four objective functions analyzed in this research.

2. Literature review of the RLP

The first contribution to solve the RLP to optimality was proposed by Petrovic [29]. It was later improved by Bandelloni et al. [2], applying dynamic programming with precedence constraints. After that, Ahuja [1] exposed an enumeration method to compute all the combinations of the activities' starting times for the minimum lags problem. Applying mixed binary-integer programming, Easa [11] proposed the first know formulation based on the Pritsker et al. [33] model for the RCPSP in which the solution is presented by binary variables that represent the finishing period of the tasks. Rieck et al. [34] introduced the domain-reducing pre-processing technique for the mixed-integer formulation. Hariga and El-Sayegh [20] improved the classical mixed binary-integer model allowing the activity splitting minimizing its associated costs. Ponz-Tienda et al. [32] developed two different binary optimization models, based on decision variables that establishes the period in which the activities are finished and executed respectively. More recently, the same authors proposed a Parallel Branch and Bound (B&B) [31] algorithm, solving to optimality 50 instances of the PSPLIB (with complexity from 10^8 to 10^{18}) for the RLP with minimal lags for the first time using an acceptable computational effort. Proposals to reduce the set of feasible solutions, branching the nodes in order to solve the problem approximately, were suggested by Neumann and Zimmermann [27], Mutlu [26] and Gather et al. [16].

The RLP, as an NP-Hard problem, has a phenomenon of "combinatorial explosion" [32] and exact algorithms are only efficient for small

projects. To avoid this problem, different heuristic procedures have been proposed along the literature to provide local optimal against global optimal solutions. The first heuristic procedure for the RLP was proposed by Burgess and Killebrew [4], establishing the Sum of Squares Optimization metric as the performance measure. The Burgess and Killebrew algorithm presents some inefficiencies that were solved by Burman [5] using the free float as the limit for the activity shifting. Other sound proposals are the Minimum Moment (MOM) algorithm [21] and the packing method (PACK) [22].

As alternative to heuristic procedures, metaheuristic algorithms are higher-level procedures designed to find sufficiently good solutions to an optimization problem with limited computation capacity. Metaheuristic algorithms are grounded in physical, biological and animal behaviour, such as Greedy Randomized Adaptive Search Procedure (GRASP), evolutionary algorithms (EA), genetic algorithms (GA), tabu search (TS), simulated annealing (SA), ant colony optimization (ACO), particle swarm optimization (PSO), shuffled frog-leaping (SFL), the grenade explosion (GE) method, or more recently the harmony search (HS), a population-based metaheuristic algorithm, proposed by Geem et al. [17].

3. Research justification, goal and process

Construction activities need to be sequenced in a way that minimizes resource variability whereas optimize the project schedule sequences. This way, the Resource Leveling Problems (RLP) aims to minimize resource fluctuations. Previous authors [28] indicate that HS algorithm are faster and more efficient than other metaheuristics avoiding the premature convergence and relapse into local optimums. However, applications of the HS algorithm to scheduling problems are still scarce in the literature.

Therefore, the goal of this study is to prove the goodness of the HS algorithms in project scheduling problems, by means of adapting the HS algorithm proposed by Chakraborty et al. [6] with a variation of the adaptive adjustment of parameters proposed by Mahdavi et al. [25] in an Improved and Adaptive Harmony Search (IAHS) algorithm. The Improved and Adaptive Harmony Search will be tested solving the j30, j60 and j120 instances of the PSPLIB [23] for four different metrics, comparing the obtained results with benchmarks available in the literature. Additionally, the results of the computational experimentation, with 5760 solved instances from 30 to 120 activities, is generalized with a real case study of 71 construction activities contemplating the widest possible set of conditions including continuity and discontinuity of flow relationships also known as point-to-point relationships. In order to fulfill this goal, this study follows the following process:

1. Literature review on the RLP (summarized in the previous section).
2. After stating the Resource Leveling Problem with minimal time lags in the next section, the proposal of an algorithm for the Improved and Adaptive Harmony Search (IAHS henceforth) for the RLP with multiple resources is introduced in the sub-sequent section.
3. The rationale of the IAHS is confirmed through computational experimentation as well as a benchmarking test.
4. Finally, a real case study (with 71 construction activities) is used to show the applicability of the IAHS proposal.

4. Problem statement of the RLP

For a complete comprehension of the remainder of the paper, some elements and the general formulation of the RLP based on activity-on-node networks with minimal finish-to-start relationships should take into consideration:

1. The set N of activities (being n the total number of activities that must be executed with constant intensity and without interruption, and

$\{j_0, j_{n+1}\}$ two dummy activities (zero duration) that represent the starting and finishing time of the project):

$$N = \{j_0, j_1, \dots, j_n, j_{n+1}\} \quad (1)$$

2. The set D of durations (being n the total number of activities):

$$D = \{d_1, \dots, d_n\} \quad (2)$$

3. The set T of times, not periods (Fig. 1), in which these activities have to be distributed (being $\bar{T}$ the deadline of the project):

$$T = \{0, 1, \dots, t\} | t \leq \bar{T} \quad (3)$$

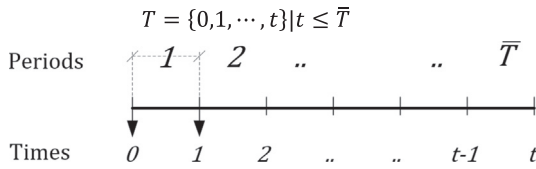


Fig. 1. Times vs. periods of the project.

4. The set R of resources (being k the total number of resources):

$$R = \{r_1, \dots, r_k\} \quad (4)$$

5. The set RQ of resources requirements for each activity (being k the total number of resources and n the total number of activities):

$$RQ = \{\{rq_{11}, \dots, rq_{k1}\}, \dots, \{rq_{1n}, \dots, rq_{kn}\}\} \quad (5)$$

6. The set U of resources requirements for each period:

$$U = \{u_1, \dots, u_t\} \quad (6)$$

7. The set C of cost associated to each resource (being k the total number of resources):

$$C = \{c_1, \dots, c_k\} \quad (7)$$

8. The set S_j of a sequence of scheduled starting times of each activity along the elements of the set T , in the following way:

$$S_j = \{ss_1, \dots, ss_k\} | es_j \leq ss_j \leq ls_j. \quad (8)$$

Table 1

Algorithmic complexity of RLP objective functions.

Objective function	Algorithm complexity
SSQR	$k \cdot n \cdot \bar{T}$
SDCDR	$k \cdot n \cdot \bar{T}$
SSDCDR	$k \cdot n \cdot \bar{T}$
RRH – MRD	$k \cdot (n \cdot \bar{T} + \bar{T})$
RID – MRD	$k \cdot (n \cdot \bar{T}^2 + \bar{T})$

Being es_j and ls_j the early and latest starting time, and k the total float of the activity j .

Once the elements that compose the problem are set, a general formulation of the objective function for the optimization problem could be a function $c_i \cdot f[r_i(S, t)]$, which computes the consumption of the resource r_i (during the period of time t that corresponds to a time t) for a feasible schedule S , for all the k resources of the project multiplied by its associated cost (c_i):

$$\begin{aligned} \text{Minimize } z &= \sum_{i=1}^k c_i \cdot f[u_i(S, t)] \\ \text{Subject to :} \\ ss_{n+1} &\leq \bar{T} \\ ss_j + d_i + \gamma_{ij} &\leq ss_j, \forall i \text{ predecessor of } j \\ \gamma_{ij} &\text{ being the lead/lag between } i \text{ and } j. \end{aligned} \quad (9)$$

The conceptual model exposed in Eq. (9) minimizes an objective function (z) subject to the following restrictions: a) the scheduled start of the finish dummy activity (j_{n+1}) must be equal or less than the prescribed makespan of the project ($ss_{n+1} \leq \bar{T}$); b) the scheduled start of a successor activity (ss_j) must be greater or equal than the scheduled start of a predecessor activity (ss_i) plus its duration (d_i) and an additional led/lag (γ_{ij}) between them ($ss_i + d_i + \gamma_{ij} \leq ss_j$).

The function $f[r_i(S, t)]$ provides different ways of dealing with the RLP. The most usual criterion focuses on getting the resource consumption as leveled as possible by minimizing the sample variance or mean square error over an ideal reference. Consequently, a suitable formulation for Eq. (10), known as the Minimum Squares Optimization method or Sum of Squares Optimization (SSQR) method, is written this way:

$$\min \sum_{i=1}^k c_i \cdot \sum_{t=1}^{\bar{T}} u_{it}^2. \quad (10)$$

And considering that $c_i = 1$; for all $1 \leq i \leq k$, the Eq. (10) can be simplified as shown in Eq. (11):

$$\min \sum_{t=1}^{\bar{T}} u_{it}^2. \quad (11)$$

Table 2

Parameters of the computational test.

Parameter	Value
Makespan	RUPSP
SES	5
HMS	$2 \cdot N $
MI	$1,000 \cdot \log(O(n))$
HMCR	0.90
PARmin	0.20
PARmax	0.75
Pitch evolution	Exponentially decreasing
FWmin	1
FWmax	4
Fetch evolution	Exponentially decreasing
SC	MI
W1	1
W2	1

Table 3
Mean improvements for different objective functions.

	j30	j60	j120
SSQR	18.656%	24.731%	30.200%
SDCDR	38.978%	44.557%	48.347%
SSDCDR	56.425%	61.728%	69.416%
RID – MRD	56.477%	61.939%	67.028%

The objective function of the Sum of Squares Optimization (SSQR) method exposed in Eq. (10) for the Resource Leveling Problem force the mathematical models and heuristic procedures to yield a flat resource utilization histogram where sampling variance is minimized, but in construction projects a bell-shaped resource profile would be better from a practical point of view [37].

Other formulations for the objective function have been proposed by several authors. The Resource Idle Days (RID) metric was proposed by El-Rayes & Jun [13] to quantify the total number of idle and nonproductive resource days caused by undesirable resource fluctuations (Eq. (12)). The Resource Idle Days metric is complemented with an additional value to control the Maximum Resource Demand (MRD) (Eq. (13)) in such a way that an adjustable multi-objective function Resource Idle Days and Maximum Resource Demand (RID-MRD) is obtained (Eq. (14)):

$$RID = \sum_{i=1}^k c_i \cdot \sum_{t=1}^{\bar{T}} [\min\{Max(u_1, u_2, \dots, u_t), \max((u_t, \dots, u_{\bar{T}-1}, u_{\bar{T}}))\} - u_t] \quad (12)$$

$$MRD = \sum_{i=1}^k c_i \cdot \max(u_1, u_2, \dots, u_t, \dots, u_{\bar{T}-1}, u_{\bar{T}}) \quad (13)$$

$$RID-MRD = W_1 \cdot RID + W_2 \cdot MRD \quad (14)$$

Being W_1 and W_2 the planner defined weight (or relative importance) for Resource Idle Days (RID) and Maximum Resource Demand (MRD) respectively.

The use of Resource Idle Days (RID) objective function yields resource utilization profiles from rectangular-shapes to hill-shapes with high peak demand, which are controlled through the inclusion of Maximum Resource Demand (MRD) metric, guarantying that the obtained resource profile be more like a bell shape rather than a hill shape.

The Release and Re-Hire (RRH) metric was proposed as alternative metric besides the Resource Idle Days (RID) metric by El-Rayes & Jun [13] to quantify the total amount of resources that need to be temporarily released during low demand periods and rehired at a later stage during high demand periods (Eq. (15)). As in the Resource Idle Days (RID) metric, the single objective function is complemented with Maximum

Table 4
Compared improvements of IAHS vs Ponz-Tienda, et al. [32].

	J30	J60	J120
IAHS	18.656%	24.731%	30.200%
AGA [32]	18.165%	23.099%	27.858%
Difference	0.491%	1.633%	2.342%
Improved instances	394	477	479
Matched instances	82.083%	99.375%	99.792%
Worsened instances	77	0	0
	16.042%	0.00%	0.00%
	9	3	1
	1.875%	0.625%	0.208%

Resource Demand (MRD) to provide an adjustable multi-objective metric (Eq. (16)).

$$RRH = \left[\frac{1}{2} \cdot \left(u_1 + u_T - \sum_{t=1}^{\bar{T}-1} |u_t - u_{t+1}| \right) - MRD \right] \quad (15)$$

$$RRH-MRD = W_1 \cdot RRH + W_2 \cdot MRD. \quad (16)$$

The Sum of Differences of Consecutive Daily Resources, (SDCDR) was proposed by Florez, Castro-Lacouture & Medaglia [14] and defined by the authors as a metric of labor stability which quantifies the project's capability of maintaining a stable crew workforce. The Sum of Differences of Consecutive Daily Resources metric aims to minimize the sum of fluctuations or absolute variation of resources along the planning horizon (Eq. (17))

$$SDCDR = \sum_{i=1}^k c_i \left(u_1 + \sum_{t=1}^{\bar{T}-1} |u_t - u_{t+1}| + u_{\bar{T}} \right). \quad (17)$$

The Sum of Squares of Differences of Consecutive Daily Resources, (SSDCDR) is an evolution of the Sum of Differences of Consecutive Daily Resources metric squaring the differences to provide a more flat shape (Eq. (18)):

$$SSDCDR = \sum_{i=1}^k c_i \cdot \left(u_1^2 + \sum_{t=1}^{\bar{T}-1} (u_t - u_{t+1})^2 + u_{\bar{T}}^2 \right). \quad (18)$$

The previous objective function for the RLP provides different shapes to provide alternative metrics and resource utilization histogram depending on the preferences of the decision maker and the needs of the project. There is not a performance metric better than another, but some metrics are more efficient algorithmically than others (Table 1), in such a way that the Resource Idle Days and Maximum Resource Demand (RID-MRD) metric requires $k \cdot (n \cdot \bar{T}^2 + \bar{T})$ calculations, being the less efficient from a computational point of view.

Table 5
Compared improvements of IAHS vs Yeniocak [37].

	29 problems from J30
IAHS	51.116%
Yeniocak [37]	52.391%
Difference	1.294%
Improved instances	1
Matched instances	12
Improved/matched instances	44.828%
Average distance	3.597%

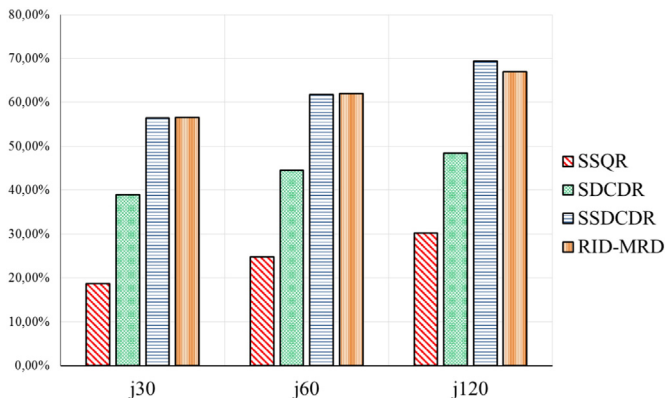


Fig. 2. Mean improvements for different objective functions.

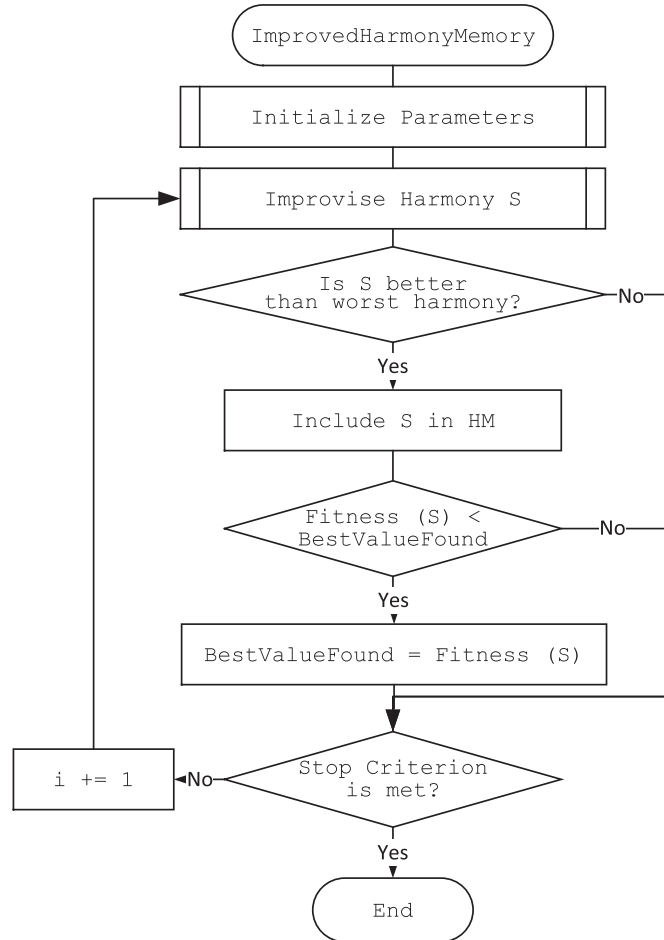
Table 6
Comparison of IAHS vs Ponz-Tienda et al. [31].

	50 problems from J30
IAHS	11.694%
Ponz-Tienda et al. [31]	11.895%
Difference	0.201%
Matched instances to optimal	45/50
	90.000%
Average distance to optimality	0.235%

5. The proposed IAHS Algorithm for the RLP

The development of Improved and Adaptive Harmony Search (IAHS henceforth) algorithm is structured in three steps: Step 1 initializes the parameters of the IAHS algorithm, Step 2 initializes the Memory of Harmonies (MH) and, Step 3 improvises new harmonies from the memory of harmonies (MH). The main structure of the IAHS algorithm is exposed in Algorithm 1 and Pseudocode 1 (see the Supplementary Data). Note that each time that the memory of harmonies is modified (lines 5 and 13), a dummy vector “Order()” is actualized, such that if $f(S(i)) \geq f(S(j))$ then $Order(i) \geq Order(j)$.

Algorithm 1. Structure of the IAHS.



5.1. Step 1. Initialize the parameters of the Improved and Adaptive Harmony Search algorithm

Once the structure of the IAHS algorithm is stated, its parameters should be set. These parameters are not static values, and must be

adjusted to the nature of the problem to be solved and the especial characteristic of the instance. These parameters are:

- Number of Notes of the Harmonies (n): It is the number of tasks of the problem to be considered in the sequences. This parameter is the cardinality of the set of non-critical tasks.
- Number of Sessions to be played by the Musicians (SES): It is the number of times that a problem is completely solved.
- Harmony Memory (HM) and Harmony Memory Size (HMS): The musician's Harmony Memory is the quantity (Harmony Memory Size, HMS) of randomly generated solution vectors (initial solutions) simultaneously handled by the algorithm. The Harmony Memory (HM) can be represented as a matrix $[HMS, n + 1]$ (Eq. (19)) with HMS rows and n columns with an additional column for the objective function $f(S_i) = f_i(S, t)$.

$$HM = \begin{bmatrix} S_1 & f(S_1) \\ S_2 & f(S_2) \\ \vdots & \vdots \\ S_i & f(S_i) \\ \vdots & \vdots \\ S_{HMS} & f(S_{HMS}) \end{bmatrix} = \begin{bmatrix} SS_{1,1} & SS_{1,2} & \dots & SS_{1,n} & f(S_1) \\ SS_{2,1} & SS_{2,2} & \dots & SS_{2,n} & f(S_2) \\ \dots & \dots & \dots & \dots & \dots \\ SS_{HMS,1} & SS_{HMS,2} & \dots & SS_{HMS,n} & f(S_{HMS}) \end{bmatrix} \quad (19)$$

- Maximum of Improvisations (MI): It is the number of iterations (improvisations) that the musician (algorithm) improvises for a new harmony (sequence) from the Harmony Memory.
- Harmony Memory Considering Rate ($HMCR$): It is the probability to pick a note from the Harmony Memory (HM), in such a way that with a $HMCR$ probability, a value is pitched from HM and with $(1 - HMCR)$ probability is selected from the set between the minimal and maximal feasible values. In the case of scheduling problems, it is the set of values between the earliest and latest starting times ($ss_{i,j} \in \{es_j, \dots, ls_j\}$).
- Pitch Adjusting Rate (PAR): It is the rate where the algorithm tweaks (change) the value which was originally picked from the Harmony Memory enabling to choose a neighboring feature. Consequently, $(1 - PAR)$ is the rate where the algorithm keeps the original picked value.
- Minimum Pitch Adjusting Rate (PAR_{min}): It is the minimal value for the Pitch Adjusting Rate (PAR) in the improvisation.
- Maximum Pitch Adjusting Rate (PAR_{max}): It is the maximal value for the (Pitch Adjusting Rate) PAR in the improvisation.
- Evolution of the Pitch Adjusting Rate ($PARE$): It is the criterion used to dynamically update the Pitch Adjusting Rate (PAR) value along the improvisation to avoid the drawbacks associated with fixed values of the Pitch Adjusting Rate (PAR).
- Fret Width (FW): It is the value that constrains the dissimilarity allowed (the shift around the neighborhood) produced by the Pitch Adjusting Rate (PAR).
- Minimum Fret Width (FW_{min}): It is the minimal value for Fret Width (FW) in the improvisation.
- Maximum Fret Width (FW_{max}): It is the maximal value for Fret Width (FW) in the improvisation.
- Evolution Fret Width (FWE): It is the criterion used to dynamically update the Fret Width (FW) value along the improvisation to avoid the drawbacks associated with fixed values of the Fret Width (FW).
- Stopping Criterion (SC): It is the rule used for finishing the improvisation, in such a way that the improvement process of the Harmony Memory (HM) continues until a stopping criterion (or the most restrictive of several Stopping Criterion) has been satisfied. The Stopping Criterion can be a given maximum number of iterations, probabilistic stopping rules [32], a target value found, an improvement in the best

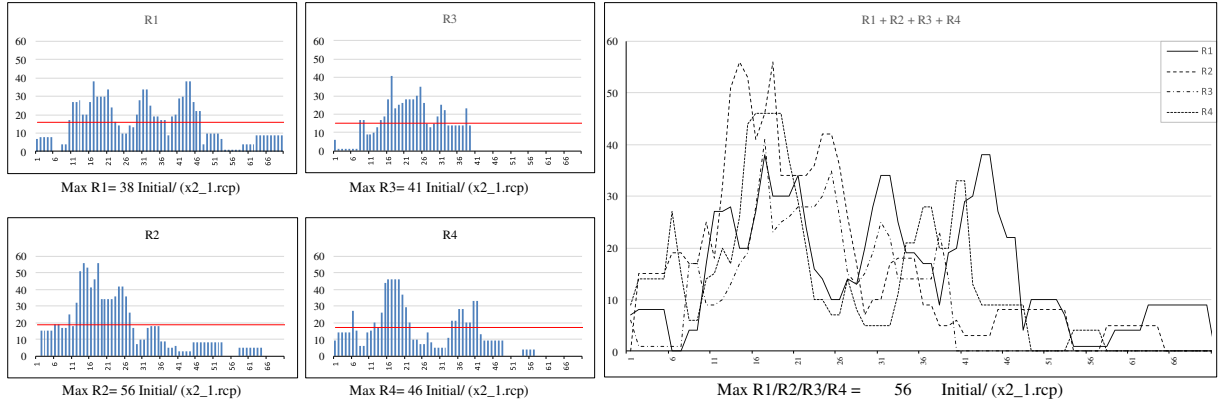


Fig. 3. Initial profile of resources (x2_1.rcp).

known solution value, or a given maximum number of consecutive iterations without improvement.

The key control parameters Harmony Memory Considering Rate (*HMCR*), Fret Width (*FW*), and Pitch Adjusting Rate (*PAR*) affects the explorative power of the IAHS algorithm. It is necessary to maintain a balance between exploration (extensive diversification) and exploitation (intensification) in the local fine-tuning at different stages of the search. The IAHS algorithm requires an effective diversification, but also intensification in the search process, opposing forces that need to be balanced [7,12]. The most commonly accepted strategy is to start with an extensive diversification along the universe of feasible values, finishing with intensification on the best found solutions. In that line, high values (*FWmax* and *PARmax*) are chosen at the beginning of the improvisation to increase the global search ability, and gradually being reduced to minimal values (*FWmin* and *PARmin*), guaranteeing the intensification.

Other proposals to avoid the use of constant values for Fret Width (*FW*) and Pitch Adjusting Rate (*PAR*) redefine dynamically this values based on the current variance [15], or adjust them randomly (does not require initial parameters) between the minimum and maximum values found in the Harmony Memory (*HM*) [36]. Mahdavi et al. [25] proposed the increasing linear function presented in Eq. (20) for adjusting the Pitch Adjusting Rate (*PAR*) and the decreasing exponential function presented in Eq. (21) for adjusting the Fret Width (*FW*).

$$PAR(i) = PARmin + i \cdot \frac{PARmax - PARmin}{MI} \quad (20)$$

$$FW(i) = FWmax \cdot e^{-i \cdot \frac{\ln \left(\frac{FWmin}{FWmax} \right)}{MI}} \quad (21)$$

For a decreasing linear evolution of Pitch Adjusting Rate (*PAR*), Eq. (22) can be applied, and for a quicker decreasing evolution, the exponential function presented in Eq. (23) should be used.

$$PAR(i) = PARmax - i \cdot \frac{PARmax - PARmin}{MI} \quad (22)$$

$$PAR(i) = PARmax \cdot e^{-i \cdot \frac{\ln \left(\frac{PARmin}{PARmax} \right)}{MI}} \quad (23)$$

In a similar way, for a decreasing and quicker evolution of Fret Width (*FW*) Eq. (24) can be applied.

$$FW(i) = FWmax - i \cdot \frac{FWmax - FWmin}{MI} \quad (24)$$

5.2. Step 2. Initialize Memory of Harmonies (*HM*)

The initial Memory of Harmonies (*HM*) is randomly generated (Algorithm 2 is shown below and Pseudocode 2 can be found in the Supplementary Data) from a uniform distribution by two heuristic procedures, forward and backward, to preserve the maximum diversification of the Memory of Harmonies (*HM*).

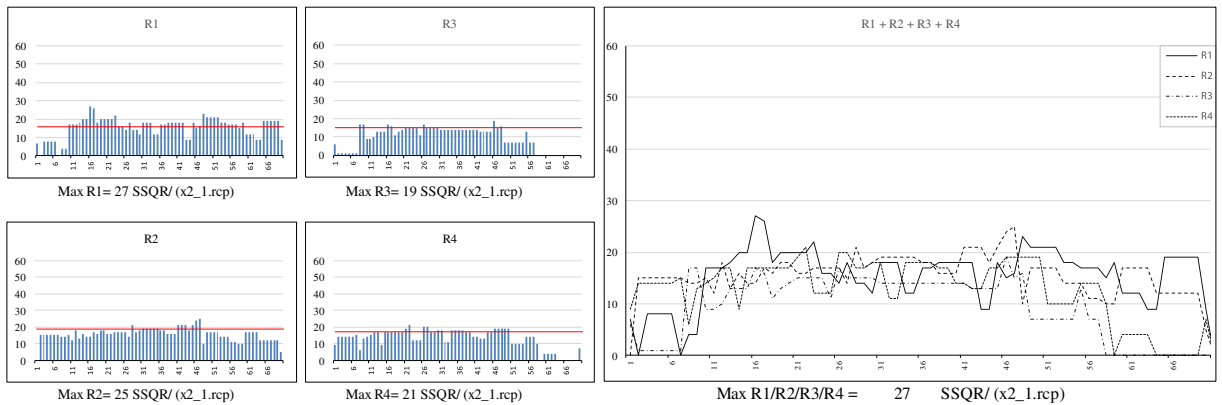


Fig. 4. Levelled profile with SSQR (x2_1.rcp).

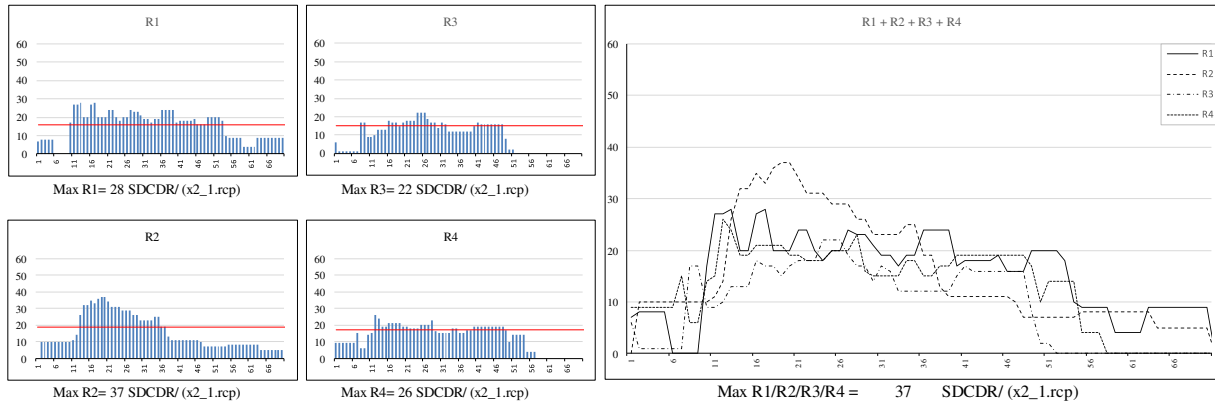
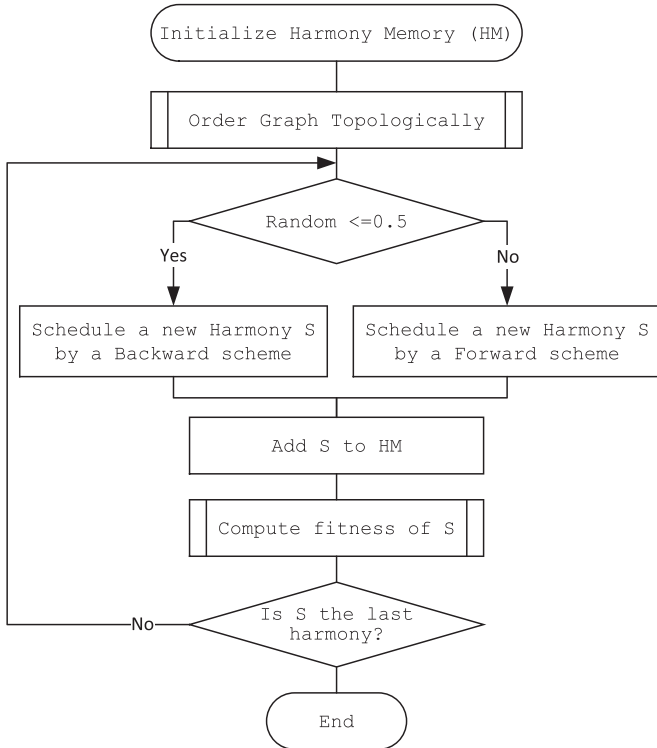


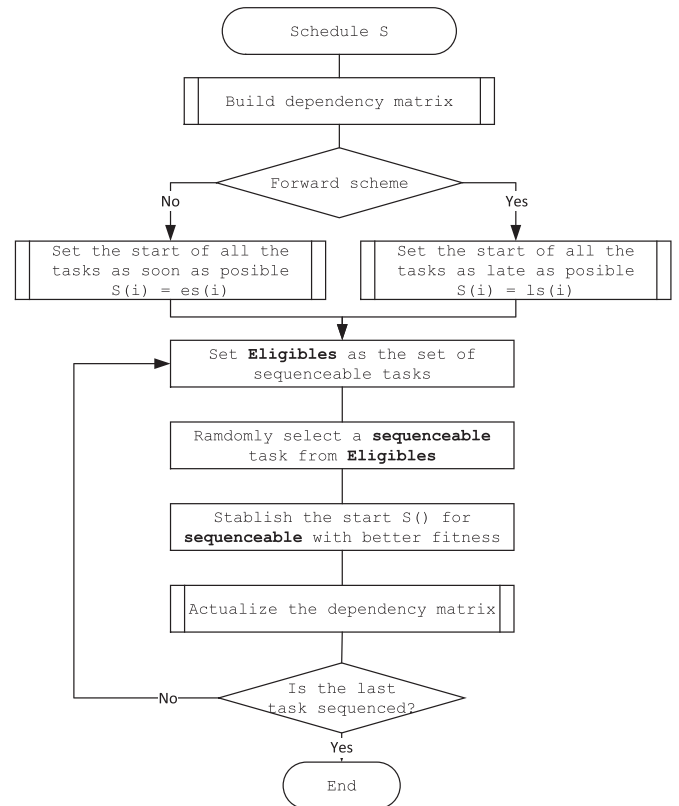
Fig. 5. Levelled profile with SDCDR (x2_1.rcp).

Algorithm 2. Initialize the Harmony Memory (*HM*).



For the RLP, all the sequences that meet the deadline $\bar{T}$ of the project are feasible sequences, because the RLP is not a resource constrained problem. For this reason, the initial Harmony Memory (*HM*) is generated applying improvement algorithms with a serial forward/backward scheduling scheme and random selection criterion. In the backward-serial scheduling scheme (Algorithm 3 is shown below and Pseudocode 3.1 can be found in the Supplementary Data), the activities are selected from a Matrix of Dependencies (*DP*) following a descending topological order. In the backward-serial scheduling scheme, an activity (*i*) belongs to the set of eligible activities to be sequenced, if all the elements of the set (*j*) of successor activities are sequenced, or in other words, all the values for *j* in $DP(i,j)$ are equal to zero ($DP(i,j) = 0$; line 10). Once the activity *i* is randomly selected (line 13) and sequenced (line 14), all the values *j* in the row *selected* are set to one ($DP(selected,j) = 1$), and the rows *i* in column *selected* are set to zero ($DP(i,selected) = 0$; lines 16 to 21).

Algorithm 3. To create a new harmony serial backward / forward with random criterion.



In the forward-serial scheduling scheme (Algorithm 3 is shown above and Pseudocode 3.2 can be found in the Supplementary Data), the activities are selected following an ascending topological order. In the forward-serial scheduling scheme, an activity (*j*) belongs to the set of eligible activities to be sequenced, if all the elements of the set (*i*) of predecessor activities are sequenced, or in other words, all the values for *i* in $DP(i,j)$ are equal to zero ($DP(i,j) = 0$; line 12). Once the activity *j* is randomly selected (line 15) and sequenced (line 16), all the values *i* in the column *selected* are set to one ($DP(i,selected) = 1$), and the columns *j* in row *selected* are set to zero ($DP(selected,j) = 0$; lines 18 to 23). Note that it is necessary to schedule all the activities with its latest starting times before applying the forward-serial scheduling scheme (lines 5 to 7).

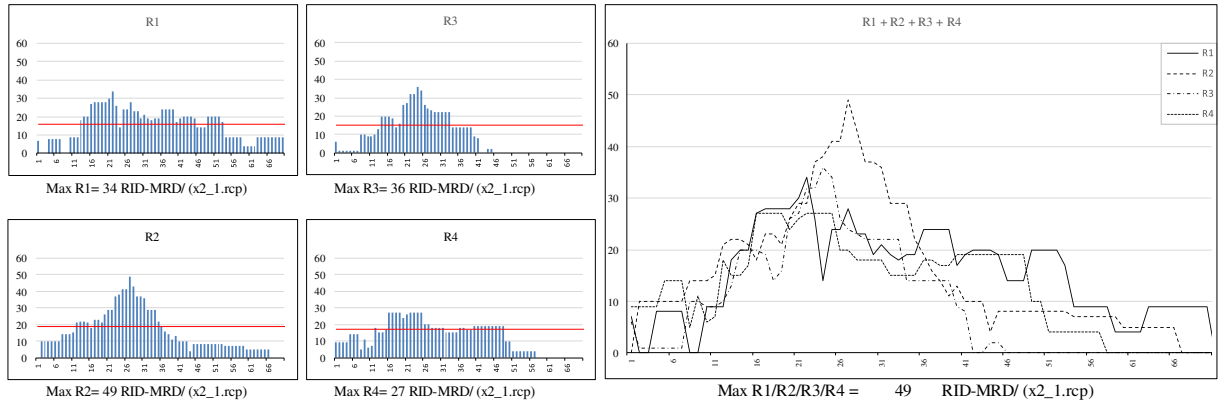
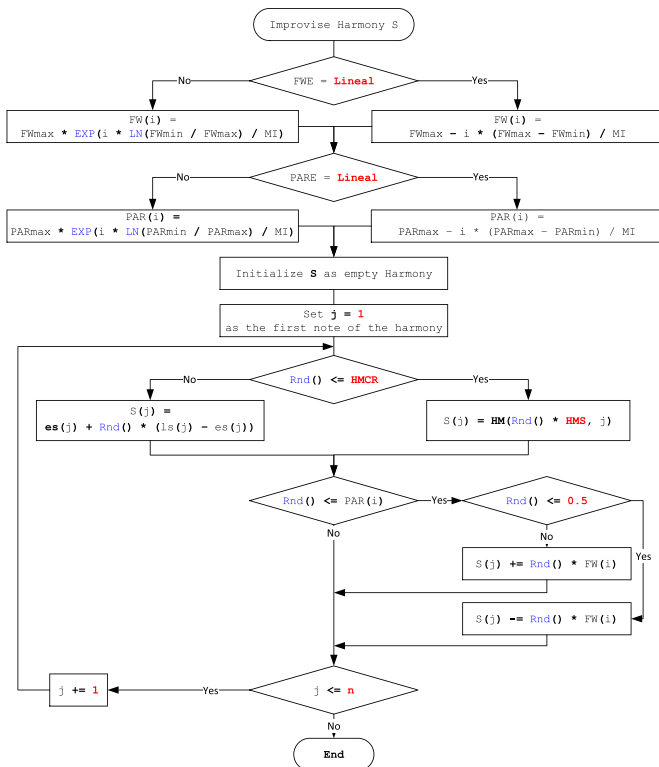


Fig. 6. Levelled profile with RID-MRD (x2_1.rcp).

5.3. Step 3. Improvise a new harmony from the Harmony Memory (HM)

Once the Harmony Memory (HM) is build, new harmonies are improvised from Harmony Memory (HM) as stated in Algorithm 4 below and Pseudocode 4 (see the Supplementary Data). First of all, the Fret Width (FW) and Pitch Adjusting Rate (PAR) values are adjusted according to the iteration (lines 3 to 13 in Pseudocode 4) following a decreasing evolution criterion for both parameters. The improvisation procedure starts pitching, with Harmony Memory Considering Rate (HMCR) probability, a note from Harmony Memory (HM), and with $1 - HMCR$ from the set between the minimal and maximal feasible values $\{es_j, \dots, ls_j\}$. Once a new note is pitched, the algorithm tweaks it with $1 - PAR$ probability enabling to choose a neighboring feature, and then the Fret Width (FW) value constrains the shift applying a uniform distribution between 0 and FW.

Algorithm 4. To improvise a new harmony S from Harmony Memory HM.



6. Computational experimentation and quantitative results

The IAHS algorithm has been completely implemented in an app developed in C# language (this app is available at <https://goo.gl/7Dcirl>) and tested with the standard sets j30, j60 and j120 of the PSPLIB library [23]. The PSPLIB library is a set of instances represented as activity-on-node networks generated by the standard project generator ProGen [23] for various types of project-scheduling problems, used for the evaluation of algorithms and solution procedures. The j30, j60 and j120 sets was designed originally for the Resource-Constrained Project Scheduling Problem with minimal lags and four removable resources, and selected to guarantee a not biased benchmarking because it provides the widest collection of solutions for the RLP.

The values of the IAHS parameters used to compute the instances are shown in Table 2. The instances are solved with five sessions, Harmony Memory Size (HMS) as twice of the cardinality of the set of activities ($2 \cdot |N|$), Maximum of Improvisations (MI) as one thousand times the logarithm of the complexity, and the prescribed makespan as that obtained for the Resource Unconstrained Problem (RUPSP).

The objective functions considered in the test are: a) the Sum of Squares Optimization (SSQR), b) Sum of Differences of Consecutive Daily Resources (SDCDR), c) Sum of Squares of Differences of Consecutive Daily Resources (SSDCDR), and d) Resource Idle Days and Maximum Resource Demand (RID-MRD), obtaining the mean improvements against the sequence with the earliest starting times (initial solution). The mean improvements for these objective functions are summarized in Table 3 and Fig. 2. Note that the improvement achieve by the IAHS algorithm for the metrics Sum of Squares of Differences of Consecutive Daily Resources (SSDCDR) and Resource Idle Days and Maximum Resource Demand (RID-MRD) are considerable greater than the obtained using the Sum of Squares Optimization (SSQR) and Sum of Differences of Consecutive Daily Resources (SDCDR), suggesting that objective functions with high hill bell shapes have a greater improvement. The PSPLIB and the complete benchmarking test for all the metrics applying the IAHS can be downloaded from <https://goo.gl/7Dcirl>.

7. Validation and benchmarking of the computational experimentation

To prove the goodness of the IAHS, the obtained results are compared with the values available in the literature performed by Ponz-Tienda et al. [32] with the Adaptive Genetic Algorithm (AGA), with the fifty instances solved to optimality by Ponz-Tienda, et al. [31] with the Parallel Branch & Bound, and by Yeniocak [37] with the efficient Branch and Bound. The first two benchmarking were solved for the Sum of Squares Optimization (SSQR) metric and the third for the Resource Idle Days and Maximum Resource Demand (RID-MRD) objective

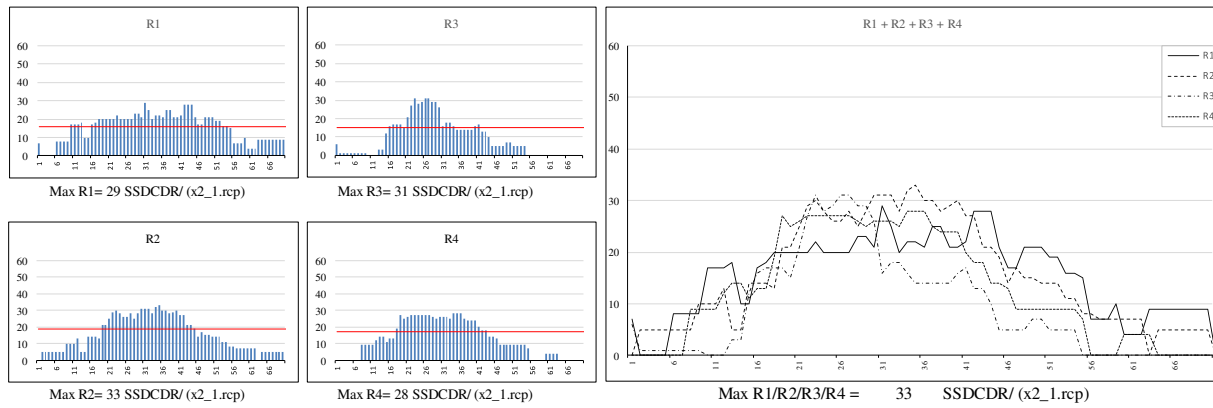


Fig. 7. Levelled profile with SSDCDR (x2_1.rcp).

function. The compared results are displayed in Tables 4, 5 and 6 respectively.

8. Discussion of results

The IAHS is faster with better overall results than other metaheuristics providing 0.491%, 1.632% and 2.342% better average values, improving 394, 477, 479 instances and matching 77, 0, 0 instances from the j30, j90 and j120 sets, respectively. Additionally, from the 480 instances of each of the sets, only nine instances from j30, three instances from j60 and one instance from j120 offered worse values than previous benchmarking with Adaptive Genetic Algorithm (AGA) [32]. The efficiency of the IAHS algorithm is especially relevant for the j60 and j120 sets, presenting an absolute difference of 1.633% and 2.342% points respectively. The absolute difference of 0.491% of the IAHS compared to the AGA for the j30 set seems to be small, but the 82.083% of improved instances and the 90.000% of matches to optimality, proves the goodness of the IAHS in order to avoid the premature convergence and the relapse into local optimums.

The efficiency of the IAHS is especially outstanding for the Sum of Squares Optimization (SSQR) metric, matching 45 instances equivalent to 90.000% and an average distance to optimality of 0.235% for the 50 instances solved by Ponz-Tienda, et al. (2016). For the Sum of Differences of Consecutive Daily Resources (SDCDR) and the Sum of Squares of Differences of Consecutive Daily Resources (SSDCDR) conclusions cannot be draw because there are no previous values with these metrics.

The Resource Idle Days and Maximum Resource Demand (RID-MRD) is less efficient than other metrics from a computational point of view. Nevertheless, IAHS is able to enhance one instance from the Yeniocak's benchmark [37] developed by applying an efficient Branch and Bound algorithm with three hours of time limit for j30 instances vs an average of five seconds required by the IAHS.

The shapes and maximum resource requirements differ depending on the applied metric, in such a way that the Sum of Squares Optimization (SSQR) metric provides a near rectangle shape with the overall less resource demand along the project. The other metrics present near bell shapes with different high peak demands. The Resource Idle Days and Maximum Resource Demand (RID-MRD) metric present a high hill bell shape with the maximum resource demand. Between the two extreme metrics, Sum of Squares Optimization (SSQR) and Resource Idle Days and Maximum Resource Demand (RID-MRD), the Sum of Differences of Consecutive Daily Resources (SDCDR) can be used for a bell shape with lowest resource demand, or the Sum of Squares of Differences of Consecutive Daily Resources (SSDCDR) can alternatively be used for a less resource demand more near to planar.

Fig. 3 displays the initial profile for instance “x2_1.rcp” of the j120 library; Figs. 4, 5, 6 and 7 show the leveled shapes for the analyzed metrics. For long-term construction projects, the Resource Idle Days and Maximum Resource Demand (RID-MRD) metric seem to be more desirable than other metrics, which are more suitable for short-term projects and discretionary by decision makers. Readers can compare the metrics for all the instances from the J120 library, because it offers the most glaring differences, with the app for Excel “J120 Metrics Visualizer” that can be downloaded from <https://goo.gl/7Dcirl>.

Table 7

Example of application; durations, relationships and construction constraints ().

Activity code	Activity description	# of activities	Duration in weeks	Weekly resource demand	Continuity condition	Precedence Relationships	
						#1	#2
1	Previous works	1	1	5	Yes	–	–
2	Excavations 0.0/– 1.0	1	2	3	Yes	FS ₁₋₂ (0)	–
3	Diaphragm-wall	1	8	5	Yes	FS ₂₋₃ (0)	–
4	Excavations	1	6	5	Yes	FS ₃₋₄ (0)	–
5	Rebars for foundation works	1	3	10	Yes	FS ₄₋₅ (0)	–
6	Concrete foundation	1	1	5	Yes	FS ₄₋₅ (0)	–
7	Structures 1 to 15	15	2	15	No	FS ₆₋₇ (0)	–
8	Masonry works 1 to 12	12	1	5	No	FI ₇₋₈ (5,1,3)	–
9	Facades 1 to 12	12	2	10	No	FI ₈₋₉ (6,1,2)	–
10	Paving works 1 to 12	12	1	5	No	FI ₈₋₉ (1,1,1)	–
11	Office works 1 to 12	12	2	10	No	FI ₉₋₁₁ (3,1,0)	FI ₁₀₋₁₁ (1,1,0)
12	Reworks and finishing	1	1	10	Yes	FS ₁₁₋₁₂ (0)	–
13	Delivery/reception	1	1	0	Yes	FS ₁₂₋₁₃ (0)	–

FS_{i-j}(z) Finish to start relationship with z lag units from activity i to j.

FI_{i-j}(p_i,p_j,z) Flow relationship with z lag units from subactivity p_i to p_j.

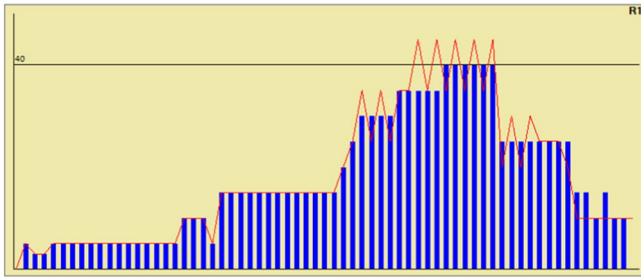


Fig. 8. Resource profile of the example of application for SSQR metric.

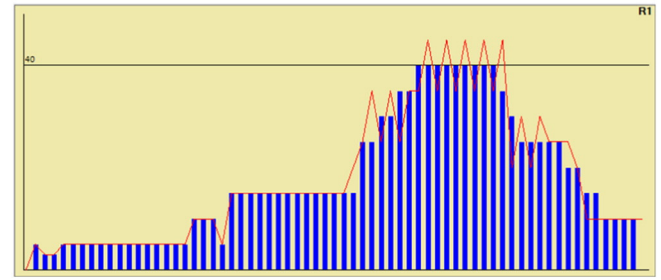


Fig. 10. Resource profile of the example of application for SDCDR metric.

9. Case study

As a case study of implementation in the construction industry, a building project of 15 floors, three underground and 12 aboveground [30,31], is used to illustrate the versatility and adaptability of the proposed IAHS. The problem has been solved, using the app exposed in Section 6, for the four objective functions considered in this research: SSQR, RID-MRD, SDCDR and SSDCDR.

The building project consist on 71 construction activities contemplating the widest possible set of conditions including continuity and discontinuity of flow relationships also known as point-to-point [3,18,19]. The structure, masonry, facades and basements are overlapped processes with an additional lag of three, two and one weeks respectively (the first lag guarantees the concrete hardening for a proper formwork removal). The durations, relationships, weekly resource demand and continuity conditions of each task and process are shown in Table 7.

The values of the IAHS parameters used to solve the example of application are the same used for the computational experimentation and shown in Table 2 with only one session. The construction resource profiles of the building project, before (red line) and after (blue bars) the leveling process, for the four metrics are presented in Figs. 8 to 11, and the values for the metrics are exposed in Table 8. It is important to note that the leveled value for the SSQR metric (30,068), provided by the IAHS algorithm, is the same as the optimal value obtained by Ponz-Tienda et al. [31], proving the goodness of the IAHS.

10. Conclusions, limitations and recommendations for future research

Resource efficiency is determinant to accomplish the objectives of a construction project. It reduces the resource variability while improving cost savings optimizing the project schedule sequences. Considering this difficulty, particularly important in the construction industry, the Resource Leveling Problems (RLP) intends to optimize the construction project activities, minimizing the resource fluctuations. These fluctuations are impractical, inefficient and costly to construction sites; therefore, increasing the efficiency of the project sequence is one key factor to achieve the goals of the project.

In order to measure the efficiency of the project sequence, different metrics has been proposed as the Sum of Squares Optimization (SSQR) method, which provides an ideal uniform shape. In construction projects, a bell-shaped resource profile would be better from a practical point of view. Other metrics have also been developed as the Resource Idle Days (RID) metric, the Release and Re-Hire (RRH) metric or the Sum of Differences of Consecutive Daily Resources (SDCDR). Exact algorithms have been proposed to offer optimal solutions, but this kind of problems is non-regular NP-Hard; consequently, heuristic and metaheuristic algorithms are developed to find local optimal solutions with an acceptable computational effort.

Therefore, the main contribution of this study to the body of knowledge in construction project management is the proposal of an innovative IAHS algorithm to deal with the RLP with multiple resources aiming to minimize resource fluctuations. This IAHS algorithm has been tested with the standard Project Scheduling Problem Library (PSPLIB) for four metrics that provide different leveled profiles from rectangular to bell shapes. The results have been compared with the benchmarks available in the literature presenting a complete discussion of results. Finally, a visualizer tool has been developed to compare the effects of applying different metrics on the J120 library with the app for Excel “J120 Metrics Visualizer”.

The IAHS algorithm is faster with better overall results than other metaheuristics. Additionally, from the 480 instances of each of the sets, only nine instances from j30, three instances from j60 and one instance from j120 offered worse values than previous benchmarking. The efficiency of the IAHS algorithm is especially relevant for the j60 and j120 sets, presenting an absolute difference of 1.633% and 2.342% points respectively.

These results show that the IAHS algorithm is especially fitted for the Sum of Squares Optimization (SSQR) metric, matching 90.000% with an average distance to optimality of 0.265% for the instances solved by Ponz-Tienda et al. [31]. The Resource Idle Days and Maximum Resource Demand (RID-MRD) is less efficient than other metrics from a computational point of view, and can be switched by the Sum of Squares of Differences of Consecutive Daily Resources (SSDCDR), which produces similar shapes requiring less computational effort.

The results of the computational experimentation, with 5760 solved instances from 30 to 120 activities, has been generalized with an

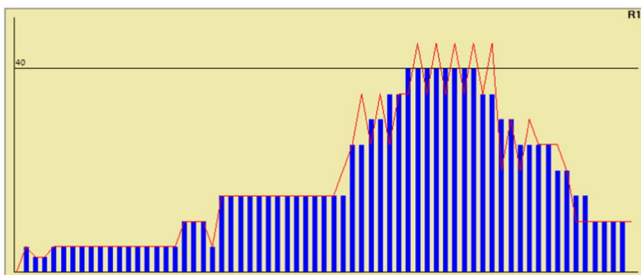


Fig. 9. Resource profile of the example of application for RID-MRD metric.

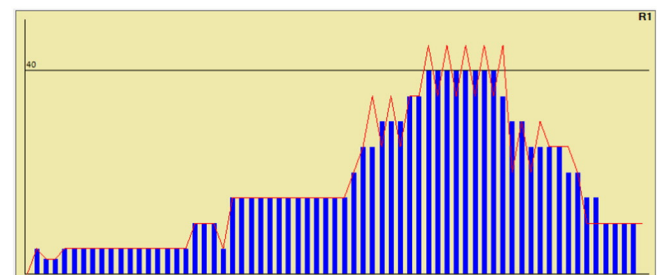


Fig. 11. Resource profile of the example of application for SSDCDR metric.

Table 8
Mean improvements for the different objective functions.

	Before	After
SSQR	31,418	30,068
RID – MRD	134	49
SDCDR	264	94
SSDCDR	2808	558

example of application of 71 construction activities contemplating the widest possible set of conditions including continuity and discontinuity of flow relationships also known as point-to-point relationships. The leveled value for the SSQR metric (30,068) is the same than the optimal value obtained by Ponz-Tienda, et al. [31], proving the goodness and applicability of the IAHS in real construction projects finding the optimal in less than one second.

The proposed IAHS algorithm for the RLP has been completely explained and implemented as a starting point in order to develop user-friendly and practical computer applications to provide realistic, fast and good solutions for construction project management. This way, in the future, the IAHS algorithm can be implemented in commercial applications including simultaneous calendars, generalized precedence relationships and discretional splitting of activities, helping practitioners to improve the planning and scheduling of complex construction projects. The proposed IAHS algorithm should be improved applying parallel computing to solve in “real time” optimization problems in commercial software for construction planning and scheduling.

Acknowledgements

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).

Appendix A. Supplementary data

Supplementary data to this article can be found online at <http://dx.doi.org/10.1016/j.autcon.2017.01.018>.

References

- [1] H. Ahuja, *Construction Performance Control by Networks*, John Wiley and Sons, New York, 1976.
- [2] M. Bandelloni, M. Tucci, R. Rinaldi, Optimal Resource Leveling using non-serial dynamic programming, *Eur. J. Oper. Res.* 78 (1994) 162–178, [http://dx.doi.org/10.1016/0377-2217\(94\)90380-8](http://dx.doi.org/10.1016/0377-2217(94)90380-8).
- [3] O. Bokor, M. Hajdu, Investigation of critical activities in a network with point-to-point relations, *Procedia Eng.* 123 (2015) 198–207, <http://dx.doi.org/10.1016/j.proeng.2015.10.078>.
- [4] A. Burgess, J. Killebrew, Variation in activity level on a cyclic arrow diagram, *J. Ind. Eng.* 13 (2) (1962) 76–83.
- [5] P.J. Burman, *Precedence Networks for Project Planning and Control*, McGraw-Hill, Incorporated, New York, NY 10036 USA, 1973.
- [6] P. Chakraborty, G.G. Roy, S. Das, D. Jain, A. Abraham, An improved harmony search algorithm with differential mutation operator, *Fundam. Informaticae* 95 (4) (2009) 401–426, <http://dx.doi.org/10.3233/FI-2009-157>.
- [7] J. Chen, Q. Yang, J. Ni, Y. Xie, S. Cheng, An improved fireworks algorithm with landscape information for balancing exploration and exploitation, *Evolutionary Computation (CEC)*, 2015 IEEE Congress on, IEEE 2015, pp. 1272–1279, <http://dx.doi.org/10.1109/CEC.2015.7257035>.
- [8] A. Damci, G. Polat, Impacts of different objective functions on resource leveling in construction projects: a case study, *J. Civ. Eng. Manag.* 20 (4) (2014) 537–547, <http://dx.doi.org/10.3846/13923730.2013.801909>.
- [9] A. Damci, D. Arditi, G. Polat, Resource leveling in line-of-balance scheduling, *Comput. Aided Civ. Infrastruct. Eng.* 28 (2013) 679–692, <http://dx.doi.org/10.1111/mice.12038>.
- [10] E. Demeulemeester, Minimizing resource availability costs in time-limited project networks, *Manag. Sci.* 41 (10) (1995) 1590–1598, <http://dx.doi.org/10.1287/mnsc.41.10.1590>.
- [11] S. Easa, Resource leveling in construction by optimization, *J. Constr. Eng. Manag.* 115 (2) (1989) 302–316, [http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(1989\)115:2\(302\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(1989)115:2(302)).
- [12] A.E. Eiben, C.A. Schippers, On evolutionary exploration and exploitation, *Fundamenta Informaticae* 35 (1–4) (1998) 35–50, <http://dx.doi.org/10.3233/FI-1998-35123403>.
- [13] K. El-Rayes, D.H. Jun, Optimizing resource leveling in construction projects, *J. Constr. Eng. Manag.* 135 (11) (2009) 1172–1180, [http://dx.doi.org/10.1061/\(ASCE\)CO.1943-7862.0000097](http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0000097).
- [14] L. Florez, D. Castro-Lacouture, A.L. Medaglia, Sustainable workforce scheduling in construction program management, *J. Oper. Res. Soc.* 64 (8) (2012) 1169–1181, <http://dx.doi.org/10.1057/jors.2012.164>.
- [15] J. Fourie, R. Green, Z.W. Geem, Generalised adaptive harmony search: a comparative analysis of modern harmony search, *J. Appl. Math.* (2013) 2013, <http://dx.doi.org/10.1155/2013/380985> (13 pp.).
- [16] T. Gather, J. Zimmermann, B. Jan-Hendrik, Exact methods for the resource levelling problem, *J. Sched.* 14 (6) (2011) 557–569, <http://dx.doi.org/10.1007/s10951-010-0207-8>.
- [17] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, *Simulation* 76 (2) (2001) 60–68, <http://dx.doi.org/10.1177/003754970107600201>.
- [18] M. Hajdu, Continuous precedence relations for better modelling overlapping activities, *Procedia Eng.* 123 (2015) 216–223, <http://dx.doi.org/10.1016/j.proeng.2015.10.080>.
- [19] M. Hajdu, Point-to-point versus traditional precedence relations for modeling activity overlapping, *Procedia Eng.* 123 (2015) 208–215, <http://dx.doi.org/10.1016/j.proeng.2015.10.079>.
- [20] M. Harija, S.M. El-Sayegh, Cost optimization model for the multiresource leveling problem with allowed activity splitting, *J. Constr. Eng. Manag.* 137 (1) (2010) 56–64, [http://dx.doi.org/10.1061/\(ASCE\)CO.1943-7862.0000251](http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0000251).
- [21] R. Harris, *Precedence and Arrow Networking Techniques for Construction*, Wiley, 1978.
- [22] R. Harris, Packing method for resource leveling (PACK), *J. Constr. Eng. Manag.* 116 (2) (1990) 331–350.
- [23] R. Kolisch, A. Sprecher, PSPLIB – a project scheduling problem library, *Eur. J. Oper. Res.* 96 (1) (1996) 205–216.
- [24] G.K. Koulinas, K.P. Anagnostopoulos, A new tabu search-based hyper-heuristic algorithm for solving construction leveling problems with limited resource availabilities, *Autom. Constr.* 31 (2013) 169–175, <http://dx.doi.org/10.1016/j.autcon.2012.11.002>.
- [25] M. Mahdavi, M. Fesanghary, E. Damangir, An improved harmony search algorithm for solving optimization problems, *Appl. Math. Comput.* 188 (2) (2007) 1567–1579, <http://dx.doi.org/10.1016/j.amc.2006.11.033>.
- [26] M.Ç. Mutlu, A branch and bound algorithm for resource leveling problem (doctoral dissertation), Middle East Technical University, 2010.
- [27] K. Neumann, J. Zimmermann, Procedures for resource leveling and net present value problems in project scheduling with general temporal and resource constraints, *Eur. J. Oper. Res.* 127 (2) (2000) 425–443.
- [28] C. Peraza, F. Valdez, O. Castillo, A harmony search algorithm comparison with genetic algorithms, in: C. Peraza, O. Castillo (Eds.), *Fuzzy Logic Augmentation of Nature-Inspired Optimization Metaheuristics*, Springer International Publishing, CH-6330 Cham (ZG), Switzerland 2015, pp. 105–123, http://dx.doi.org/10.1007/978-3-319-10960-2_7.
- [29] R. Petrovic, On optimization of resource leveling in project plans, *Project Planning by Network Analysis*, Lombaers HJ, North-Holland, Amsterdam 1969, pp. 268–273.
- [30] J.L. Ponz-Tienda, E. Pellicer, J. Benlloch-Marco, C. Andrés-Romano, The fuzzy project scheduling problem with minimal generalized precedence relations, *Comput. Aided Civ. Infrastruct. Eng.* 30 (11) (2015) 872–891, <http://dx.doi.org/10.1111/mice.12166>.
- [31] J.L. Ponz-Tienda, A. Salcedo-Bernal, E. Pellicer, A parallel branch and bound algorithm for the resource levelling problem with minimal lags, *Comput. Aided Civ. Infrastruct. Eng.* (2016) <http://dx.doi.org/10.1111/mice.12233>.
- [32] J.L. Ponz-Tienda, V. Yepes, E. Pellicer, J. Moreno-Flores, The resource leveling problem with multiple resources, *Autom. Constr.* 29 (2013) 161–172, <http://dx.doi.org/10.1016/j.autcon.2012.10.003>.
- [33] A.A. Pritsker, L.J. Walters, P.M. Wolfe, Multiproject scheduling with limited resources: a zero-one programming approach, *Manag. Sci.* 16 (1) (1969) 93–108, <http://dx.doi.org/10.1287/mnsc.16.1.93>.
- [34] J. Rieck, J. Zimmermann, T. Gather, Mixed-integer linear programming for resource leveling problems, *Eur. J. Oper. Res.* 221 (2012) 27–37, <http://dx.doi.org/10.1016/j.ejor.2012.03.003>.
- [35] N. Siddique, H. Adeli, Nature inspired computing: an overview and some future directions, *Cogn. Comput.* 7 (6) (2015) 706–714, <http://dx.doi.org/10.1007/s12559-015-9370-8>.
- [36] C.-M. Wang, Y.-F. Huang, Self-adaptive harmony search algorithm for optimization, *Expert Syst. Appl.* 37 (4) (2010) 2826–2837, <http://dx.doi.org/10.1016/j.eswa.2009.09.008>.
- [37] H. Yeniocak, An Efficient Branch and Bound Algorithm for the Resource Leveling Problem, Middle East Technical University, Ankara, Turkey, 2013 (Master of Science Dissertation).