Implementation of Unsupervised Learning Method in Rule Learning from Construction Schedules

B.Y. Ryoo[a] and M. Ashtab[b]

[a] Department of Construction Science, Texas A&M University, United States
[b] College of Architecture, Texas A&M University, United States
E-mail: bryoo@tamu.edu sokratis10@tamu.edu

Abstract –

The construction industry has insufficient utilization of standard work and workload. Generally, scheduling for construction projects follows the common sense of the industry. The sequencing of activity and its duration estimation is highly dependent on the experience of the experts who are assigning them to the project, and it is a considerable barrier for automating scheduling process.

To overcome this challenge, the FP-Growth algorithm, which is an automated unsupervised learning tool, applied to create a platform for the acquisition of knowledge from actual construction schedules which are the outcome of experienced experts. The main advantage of this method in comparison to supervised learning models is the fact that it can generate contractor-specific rules from a given schedule and also identify a variety of potential path when it is applied for multiple projects which are similar to each other. The main contribution of FP-Growth Algorithm to this research is in finding association rules between sets of activities and identifying recurrent patterns in the sequence of activities, their duration, logical relationship (FF, SS, SF, FS) and specifications in different sections of construction projects.

The model applied on schedules of two case studies with different occupational function and structural material. The model substantiated to be capable of learning and identifying various rules including activity durations, predecessor activity and logical relationship and lead times that can happen in between two related activities.

Keywords –

Unsupervised Learning; Construction Scheduling; Rule Learning; Machine Learning; Association rules; FP-Growth Algorithm

1 Introduction

Construction project schedules are heavily replying to schedulers' expertise and corporate scheduling approaches.

Also the construction industry always faces delays and change orders.[1] Poor planning, lack of flexibility during change orders and inconsistent resource allocation are the leading causes of delay, which are instigated directly by deficiencies in scheduling practice. The main goal in scheduling is to set a baseline for activity relationships and their duration while considering optimized tradeoffs between time, cost and resources. The scheduling process mostly based on personal experience of planners and lacks standardized sets of work items. The complex nature of the scheduling process with the presence of logical and resource-based constraints makes it difficult for project planners to generate the optimum schedule persistently [2]. Hence, construction management practice needs more dynamic and integrated schedules.

Previous works on the issue mainly focus on small and partial sets of activity for projects and try to find a mathematical solution to optimize time while meeting all the resource and cost restraints of projects [3]. The common shortcoming for all of these studies is that they applied in large scale and real-life projects. The main approach to overcome this gap and simplify the scheduling problems could be binding sets of activities together and assign single values of constraint to them [4]. Another way is generating association rules based on the sequence of activities and use them as constants while generating a schedule based on mathematical models. While the majority of the solutions for automating schedules focus on supervised learning which has to establish probabilistic hypothesis to generate a single solution for similar scheduling, this research implements the Frequent Pattern Algorithm(FP-Growth algorithm) to absorb different possibilities and sequences that can be performed to create a schedule. FP-Growth Algorithm identifies association rules among activities which can be different from project to project and instead of having a model with given premises, the model learns contractor specific rules. The main benefit of using FP-Growth over other pattern finding methods is the fact that it is compatible with confined groups of datasets which

would be the case for current research [5]. After shaping the constants for the schedule genetic algorithm would be implemented to optimize time for the rest of the project based on its tradeoff with resource and cost.

2 Related Researches

Time overrun is one of the significant barriers to project success. There are a plethora of factors contributing to the intensification of this problem. Some studies investigated the factors which are instigating delays and their importance to tackle the negative effects of them on schedules. According to Zidane and Andersen [1] who implemented a broad literature review on the issue, poor planning and scheduling, resource shortage and poor decision making in change orders are the most repeated factors in altering construction project time globally. Different strategies had been applied to improve each problem.

In addressing resource-constrained scheduling, many scholarly works focused basically on using mathematical optimization tools. Adeli [6] defines an optimization function for minimizing direct cost based on the maximum acceptable time for a transportation project and applies artificial neural networks to generate a solution for the target function. Toklu [7] uses sets of concrete-related activities for applying Genetic Algorithm as the time minimization tool which sets on the stage for other papers to implement a genetic algorithm for scheduling problems. Dawood and Sriprasert [8] had continued the path adding time-space confliction alongside the resource constraint as a variable to the mathematical model of optimizing time according to cost. In another significant approach to the problem, Birjandi and Mousavi [4] differentiated between a set of activities that can be bind together and other activities in the schedule in modelling and formulating their target function for optimization. There has been plenty of similar researches conducted based on optimizing cost time relation through the application of some alterations using Genetic Algorithm and stochastics [9-16].

Koo, et al. [17] concentrate on rescheduling practice by developing an anthological hierarchy for categorizing critical path activities based on the level of effect they got from resource constraint during the project and also their effect on their successor activities. Building information modelling also applied as a tool to automate data generating in some of the recent works where tasks which are driven from BIM models are subject to optimization to reach the time-cost-efficient solution.

Faghihi, et al. [3] focused on sequencing activities for a limited set of activities in the steel structure model while Chen, et al. [18] developed a comprehensive module to generate a full schedule of the project through linking it with resource and cost data.

Reviewed works are containing valuable solutions to address scheduling problems, but they all share a common gap. They generate a single solution for a dynamic problem of creating schedules and using stochastic models for considering uncertainties to run their model. This is due to the complexity of calculations for mathematical models when they engage with a plethora of activities and their intensified logical relations. Alongside that, they put aside the current practice by trying to generate schedules with any kind of input from historical data from contractors. Hence, the schedules generated in this manner are not expandable to complex projects in commercial, residential sections of the construction industry. Figure1 summarizes the problems and gaps that have been identified in this research.

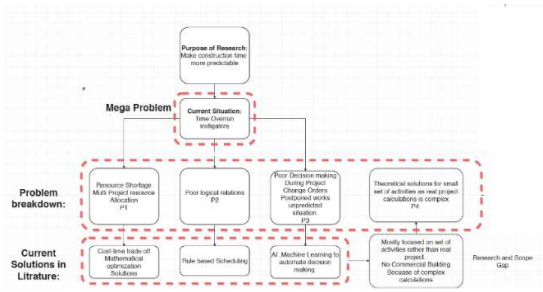

Figure1.

The construction scheduling automation literature

FP-Growth algorithm has the potential to ease both problems. By extracting rules from current schedules and apply them as constants, uncertainty and thus, the complexity of mathematical models can decrease. Furthermore, the FP-Growth algorithm has the potential to mine project-specific rules schedule by schedule and integrate the knowledge of experts with generated results.

The current study focuses on generating knowledge from real-life project schedules. Identifying patterns in a sequence of activities as association rules inside a project and assessing the significance of them could be the missing step in generating automated schedules.

3 Research methodology and Data preparation

As discussed, the scheduling practice in the construction industry lacks consistency and flexibility, and both problems could be alleviated through automation. Automation can be considered as sequencing activities or allocating resources. The current article aims

to reduce the complexity of automation in generating schedules, by suggesting rules which are driven through investigating contractor schedules and reduce variables and give more certainty to optimizing models. Unsupervised learning has the ability to capture underlying knowledge), which is specific to each schedule. To be more specific FP-Growth algorithm as an unsupervised learning tool is capable of mining durations and relationships of activities in a project-specific manner.

FP-Growth's main contribution to the research was it is ability in Finding frequency among features and attribute relationships which were both needed in finding a chain of activities that would be repeated in each floor or work sections.

The research uses two construction schedules as the case studies to examine the possibility of extracting association rules from them which draw a certain and meaningful sequence of activities. The first project was a multi-story hotel building with a concrete structure (CASE 1), and the second one was a 3-story commercial building with steel structure (CASE 2).

For extracting association rules and their significance, there are multiple options in the unsupervised machine learning realm. The FP-Growth Algorithm selected as the data available were confined. While the FP-Growth Algorithm works with nominal variables, it fits with the type of data that can be extracted from schedules.

3.1 Preparing data for the FP-Growth algorithm

The next step was transforming the available schedule into a decent input for FP-Growth Algorithm. The schedules included activities identification number (ID), description, start date, finish date, predecessor activity or activities and duration. Furthermore, logical relationships and sequences like Finish-to-Start (FS), Start-to-Start (SS), Start-to-Finish (SF), Finish-to-Finish (FF) gathered to be mined and provide more practical rules.

While the FP-Growth Algorithm works with nominal variables, it fits with most of the attributes except duration, which is a continuous value. By rounding all times into an integer, this problem solved as well. The main challenge here was to identify the type of activities from the description section.

3.1.1 Attribute extraction from descriptions

Each project had a specific routine in the description section. For CASE 1 schedule, the description part had a systemic approach. The operational part separated into three main parts. The building had 14 stories above ground and description was based on the level of the building, the activity, and the section of work on that level which can be extracted in excel and put into three different attributes. The predecessor for each activity also located and expanded in the same row. Table1 shows attributes extracted from CASE 1 and their range.

For CASE 2 dataset available had an attribute as activity types like contracts, procurement, structure, foundations, interior, paving and etc. In each group of activities with the same type, the extraction of attributes of section, level and type from the description part was the challenging part due to inconsistency in naming activities. So, as an instance in the foundation to extract attributes from activities, the tags created based on possible foundation types. The first step was to know what is the foundation type and how it is named description section. Tags such as Drill, Drilled, Pier, Shaft, Caisson, Mat, Grade, Grade Beam, Slab on grade, footing, strip, spread checked on data set and the results showed that the building uses drilled piers and grade beams for the foundation. Here due to lack of enough datasets, general knowledge of construction work played a role in guessing tags. By having more datasets, text mining can come to help in not only in generating tags but also in creating trees of activities which are connected based on tags. Table2 shows attribute extracted from activities with foundation type and their variety and range in CASE 2.

Table 1. Attributes extracted from CASE 1

Attribute	Range
Activity	
Duration	Integer (weeks)
Activity/ PRE-Requisite	
Type	MB(Mobilization), EX (Excavation), C.R. (Crane), SG (Slab on grade), CS(Concrete) R.B.(Rebar), FR (Framing)
Section	1,2,3, ALL
Level	B2-L14
Spec	PR (pouring), FT (footing) FR (framing) SC (stress cable)
Logic	Finish-to-Start (FS) Start-to-Start (SS) Start-to-Finish (SF) Finish-to-Finish (FF)

PRE-Requisite	
Level Difference(D)	Integer
Lag	Integer (hours)

Table 2. Attributes extracted from CASE 1

Attribute	Range
Activity	
Duration	Integer (weeks)
Activity/ PRE-Requisite	
Element/area of Work	Piers Grade beams Elevator Slab on Grade Under Slab
Section	1,2,3, ALL
Level	N/A for foundation
Spec	Pour Cure Forms(edge) Forms (Carton) Waterproof Electrical Plumbing Reinforcing Strip/lift Excavation Backfill
Logic	Finish-to-Start (FS) Start-to-Start (SS) Start-to-Finish (SF) Finish-to-Finish (FF)
PRE-Requisite	
Level Difference(D)	Integer
Lag	Integer(hours)

3.1.2 Attribute extraction from sequences

For each activity, there might be a predecessor or successor activities in a given schedule. So, extracting attributes from predecessor or successor has already done except the possible connection between predecessor or successor and the activity itself. The logic of connection (i.e., Finish-to-Start, Start-to-Start, Finish-to-Finish and Start-to-Finish) and lags were ones which already provided in the schedules. Furthermore, there are some locational dependencies which can happen between activities. So, to consider that, Level (i.e., floor) difference (i.e., Level D) attribute defined as subtraction of level in which activity takes place, and the level predecessor needs to be done. To monitor sections, section-relation attribute defined. As there are only three of them in each CASE, all six possible connections considered as a different value of the attribute.

Hence, for a given activity, attributes extracted for itself and also for its predecessor shape a row of feature for it. If activities have multiple predecessors, a new row of feature would be considered for the attributes of the same activity and the other predecessor. So, for example for and activity with four predecessors there would be four rows of features in the dataset which is subject to be analyzed by the FP-Growth algorithm. It is evident that while covering activities and their predecessors, its precisely the same job in the reverse direction if the study would focus on the activities and their successors. Hence, to eliminate the unnecessary data and keep dataset consistent only the predecessor relations considered as the basis.

3.2 Implementation of FP-Growth Algorithm

In the previous section, the activities and each of their predecessor analyzed and expanded into a row of features. FP-Growth algorithm implemented to figure out if there is a significant pattern among the rows of features, While all the attributes could get finite values (either it is integer or string), an operator used to transfer or nominal variables into binominal. Dummy encoding used to separate columns for each value of a single attribute to make it more flexible to use in the FP-Growth model. Furthermore, the model applied to sets of activities with the same attribute of type in both CASE 1 and CASE 2 to give more realistic results. The process of data preparation and modelling has done in RapidMiner Studio®. The overall process has shown in Figure 2.

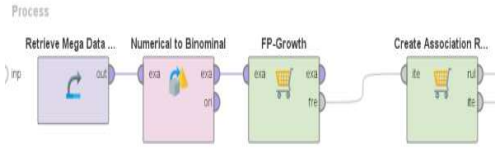

Figure2.

Process of implementing FP-Growth in Rapid mining

The association rules which are driven from the FP-Growth Algorithm are containing premises and conclusions. Premises are the constant parts that we would always have in our datasets. Conclusions are that might vary when the data changes. In this case activity, spec and detail are the constant part, and the parts that are needed to get defined as conclusions are the duration of activity and its predecessor logic, type, Level D. So, the rule generation followed the same logic by separating different activity types.

This process repeated for all the sections of both projects and CASE 1 concrete pouring and framing generated some rules and in CASE 2 foundation and structure came out with some rules that will be discussed in section 4.

4 Data Analysis

The first group of activities that generated meaningful association rules were framings in CASE 1. The main reason for that was the fact that the framing activities repeated in 14 levels without any sectional consideration. The first step was applying FP-Growth to all row of features with framing (i.e., FR) as their activity type. Here the main output is not all the created rules but just identification of most repetitive values in each attribute. The most repetitive values were as below:

As it is discussed in section 3 section, specification and type of activity (i.e., Activity Section, Activity Spec, Activity Type) and also the type, section and specification of predecessor and its logical relationship (i.e., SS/FS/FF/SF PRE Requisite Spec, SS/FS/FF/SF PRE Requisite Section, SS/FS/FF/SF PRE Requisite Type) alongside the Lag (i.e., SS/FS/FF/SF PRE Requisite lag) would be components of one rows of attributes in the dataset generated from the schedule.

Activity Section = ALL,
Activity Spec = IN,
Activity Type = FR,
SS PRE Requisite Lag = 0,
SS PRE Requisite Level D = -5,
SS PRE Requisite Section = 2,
SS PRE Requisite Spec = PR
SS PRE Requisite Type = CS
FS PRE-Requisite Lag = 14 days,
FS PRE-Requisite Level D = 1,
FS PRE-Requisite Section = ALL,
FS PRE-Requisite Spec = IN,
FS PRE-Requisite Type = F.R.
Duration = 3(weeks)

Before reporting rules resulted from the implementation of the FP-Growth algorithm, it is necessary to mention that the association rules are frequent if-then patterns which can be found through a data set and significance of them in identifying recurrent patterns are related to *support* and *confidence criteria.* Support is an indication of how frequently the items appear in the data. Confidence indicates the number of times the if-then statements are found true.

As the first endeavour for finding association rules, the search was for the attributes that could be bind together in the framing section. To make that happen the attributes connections with FR (framing) attribute evaluated. Initial interpretation for the outputs was the fact that the framing had taken place without any further section separation inside each level. It had happened at the same time for a single level. (Activity sec=All).

Premise:
Activity Type = FR
Conclusion:
Activity Section = ALL, Activity Spec = IN
(Confidence:1, Support:1)

So, for any given activity with FR (framing) type, its section and spec would be as of ALL and IN (interior). While doing more detailed rule mining, these three attributes could be considered as one for framing related rows of features and bind together as the general premise in the next steps.

Further looking there are two frequent predecessors for Framing activities in CASE 1. The first one has a Start-to-Start logic and is among pouring activities from the floors above. Second, framing activities from lower floors which has a Finish-to-Start logical sequence with the desired activity. While the framing activities take place in similar square feet in most levels, the duration could also be mined as three weeks (Duration=3) which is not a rule but a valuable taking for indicating the rate crews would work on that.

Premise:
Activity type= FR,
Activity Section = ALL,
Activity Spec = IN
Conclusion:
Duration=3weeks
(Confidence:1, Support:0.923)

After figuring out which attributes are the most frequent in framing rows of feature, a more specific FP-Growth algorithm had performed separately on the row of features in framing with FS predecessor and also SS predecessor. Given rows with FR. as the type and SS as

the logic of relation and putting them in the premise section of creating association rules, the first significant rule was as below:

Premise:
Activity type= FR,
Activity Section = ALL,
Activity Spec = IN
Conclusion:
FS PRE-Requisite Lag = 14 days,
FS PRE-Requisite Level D = 1,
FS PRE-Requisite Type = F.R.
(Confidence:.0.8, Support:0.923)

Given rows with FR as the type and FS as the logic of relation and putting them in premise section of creating association rules the first significant rule was as below:

Premise:
Activity type= FR,
Activity Section = ALL,
Activity Spec = IN
Conclusion:
SS PRE Requisite Lag = 0,
SS PRE Requisite Level D = -5,
SS PRE Requisite Section = 2,
SS PRE-Requisite Type = CS
(Confidence 0.9, support:0.583)

Based on the analysis on this section for any given framing activity that would be inside and will not be done on sections in each level. Furthermore, for nearly most of the FR activity, there is the FR predecessor with the Finish-to-Start relationship in one level lower. There would be a 2-week lag between predecessor and successor in this case

Finally, for nearly half of framing activities (support=0.583) there is CS (concrete for structure) predecessor with Start-to-Start logic which happens in Section two and six-level higher than the level that activity takes place, without any lags.

The second part of CASE 1, which generated meaningful rules was pouring concrete, and the schedule was gathered all the related tasks (forming, reinforcing and pouring) into one single item.The process which has applied for the framing section utilized here as well. First, the most repetitive attributes identified from applying the FP-Growth algorithm to all rows of features in CS (concrete of structure) section.

Activity Spec = PR
Activity Type = CS
FS PRE-Requisite Lag = 0
SS PRE Requisite Level D = 0
Duration = 2weeks
FS PRE-Requisite Level D = 1
FS PRE-Requisite Spec = PR
FS PRE-Requisite Type = CS
SS PRE-Requisite Lag = 0
SS PRE-Requisite Level = NA
SS PRE-Requisite Logic = NA
SS PRE-Requisite Section = 0
SS PRE-Requisite Spec = NA
SS PRE-Requisite Type = NA

Before starting any further analysis, the frequent attributes show that there is not any Start-to-Start predecessor for pouring activities while all the extracted frequent values for that are showing NA (not applicable). Hence, the main focus here remains with Finish-to-Start predecessors.

The first part was searching attributes that could be bind with activity type (CS). Implementing FP-Growth here gave us the following results.

Premise:
Activity Type = CS
Conclusion:
Activity Spec = PR
(Confidence:1, Support:1)

The duration of activities proved to be minable in the CS section while the pouring parts were in equal square feet and setup.

Premise:
Activity Type = CS
Activity Spec = PR
Conclusion:
Duration = 2
(Confidence:0.8, support:0.972)

The level difference and Lag and also type predecessor and the support rate of them also evaluated by putting them in the conclusion section of FP-Growth algorithm while considering Activity type (CS) and specs (PR) as the premises.

Premise:
Activity Type = CS
Activity Spec = PR
Conclusion:
FS PRE-Requisite Level D = 1
FS PRE-Requisite Type = CS
SS PRE-Requisite Lag = 0
(Confidence:0.9, support:0.889)

This indicates that for nearly most of CS activity, there is a CS predecessor with the Finish-to-Start relationship in one level lower. There would be no lag between predecessor and successor in this case. All the CS activities have PR (pouring) specification, which directly relates with the setup that schedule uses in gathering all forming, reinforcing and pouring as one

activity.

CASE 2 undergone the same process FP-Growth Algorithm and for foundation section, main rules were as below:

Premise:
Element= Piers,
Type= Pour,
Logic = FS,
Conclusion:
Element P=Pre1-Cage
(Confidence: 1.000, support: 0.375)

Here our model identified for any pier pouring activity there would be predecessor containing pier cage with Finish-to-Start relation to that. The reason for the decrease in support ratio is the fact that in CASE2, the four kinds of logic between activities put in separated rows of features. So, the initial ratio divided into each logic. For example, the support rate for the appearance of Element=Piers and Logic = FS is 0.625.

For the grade beams, the model came up with association rules as below:

Premise
Element=Grade Beams
Conclusion
Element P=Pre1-Grade Beams
(Confidence:1.000, support: 0.890)

Premise:
Element=Grade Beams,
Element P=Pre1-Pour
Conclusion:
Logic = FS
(Confidence: 0.917, support: 0.890)

5 Findings and conclusion

Implementation of the FP-Growth algorithm for CASE1 and CASE2 had a variety of notable findings. In CASE1 due to abbreviations in the work description and consistent way of naming activities make data mining relatively easy comparing CASE2, which had more descriptive titles as its activities' names.

Furthermore, unlike the successful experience of extracting duration for activities in CASE1, the model experienced lower confidence rates in extracting durations in CASE2. The main reason for that was more repetitive and typical activities in CASE1 with the sequence of activities remaining the same in each floor. For buildings with a lower number of floors, the only possible way is gathering group of similar projects and create full rows of features for them to mine duration.

In learning the logical relationship between activities and their Lag, the model performed well in both cases and generated meaningful rules. Start-to-Start and Finish-to-Start relationships between activities were identified as the most informative rules because they were addressing the connection between varying types and specifications in most cases which could be the cornerstone in shaping a masterplan. For example, FP-Growth algorithm learned SS relationship between framing and pouring five floors away from each other in the CASE1, which could be a significant barrier if it has not been put into consideration in master and detailed planning.

The model was performing better over CASE2 in identifying the sequence of activities ending in a specific element. The main reason for that could be the detailed description of each activity in their name column. The rules were found showed the pier cage as the predecessor of pouring for the pier. Also, pouring the pier was a predecessor for grade beam related activities with excavation specification. The reason for that was the broad details provided in naming each activity. The rules generated from CASE2 could be used to complete in between any two milestones which are identified for masterplans with activities.

To conclude, the research shows the possibility of generating rules in micro and macro scale from historical data and real-life schedules. Generated rules could minimize the uncertainty of mathematical models for scheduling as they can function as constant features in them. Also, by having hundreds of similar schedules, the standardization of construction work can happen through generated rules from more descriptive schedules. The results also have the capability to be used in generating alternative schedules Further uses of the results of current research could be in reducing uncertainties in scheduling process which can help to optimize the projects' cost-time tradeoffs which are highly dependent on the mathematical model. By having more learned rules that their frequency and certainty had been evaluated, the initial schedule could be set up and the allocation of resource with the goal of optimizing time-cost function happen for remaining activities.

References

[1] Y. J. T. Zidane and B. Andersen, "The top 10 universal delay factors in construction projects," *International Journal of Managing Projects in Business,* vol. 11, no. 3, pp. 650-672, 2018.

[2] J.-B. Yang and P.-R. Wei, "Causes of delay in the planning and design phases for construction projects," *Journal of Architectural Engineering,* vol. 16, no. 2, pp. 80-83, 2010.

[3] V. Faghihi, A. Nejat, K. F. Reinschmidt, and J. H. Kang, "Automation in construction scheduling: a review of the literature," (in

English), *Int J Adv Manuf Tech,* vol. 81, no. 9-12, pp. 1845-1856, Dec 2015.

[4] A. Birjandi and S. M. Mousavi, "Fuzzy resource-constrained project scheduling with multiple routes: A heuristic solution," *Automation in Construction,* vol. 100, pp. 84-102, 2019.

[5] M. Kavitha and S. T. Selvi, "Comparative Study on Apriori Algorithm and Fp Growth Algorithm with Pros and Cons," *International Journal of Computer Science Trends and Technology (I JCS T)–Volume,* vol. 4, 2016.

[6] H. Adeli, & Karim, A., *Construction scheduling, cost optimization and management*, 1st ed. London: CRC Press, 2001.

[7] Y. C. Toklu, "Application of genetic algorithms to construction scheduling with or without resource constraints," (in English), *Can J Civil Eng,* vol. 29, no. 3, pp. 421-429, Jun 2002.

[8] N. Dawood and E. Sriprasert, "Construction scheduling using multi‐constraint and genetic algorithms approach," *Construction Management and Economics,* vol. 24, no. 1, pp. 19-30, 2006.

[9] S. Chand, Q. Huynh, H. Singh, T. Ray, and M. Wagner, "On the use of genetic programming to evolve priority rules for resource constrained project scheduling problems," *Information Sciences,* vol. 432, pp. 146-163, 2018.

[10] M. Đumić, D. Šišejković, R. Čorić, and D. Jakobović, "Evolving priority rules for resource-constrained project scheduling problem with genetic programming," *Future Generation Computer Systems,* vol. 86, pp. 211-221, 2018.

[11] H. Seidgar, M. Kiani, and H. Fazlollahtabar, "Genetic and artificial bee colony algorithms for scheduling of multi-skilled manpower in combined manpower-vehicle routing problem," (in English), *Prod Manuf Res,* vol. 4, no. 1, pp. 133-151, Aug 25 2016.

[12] G. Campos Ciro, F. Dugardin, F. Yalaoui, and R. Kelly, "Open shop scheduling problem with a multi-skills resource constraint: a genetic algorithm and an ant colony optimization approach," *International Journal of Production Research,* vol. 54, no. 16, pp. 4854-4881, 2015.

[13] Z. Irani and M. M. Kamal, "Intelligent Systems Research in the Construction Industry," (in English), *Expert Syst Appl,* vol. 41, no. 4, pp. 934-950, Mar 2014.

[14] V. Faghihi, K. F. Reinschmidt, and J. H. Kang, "Construction scheduling using Genetic Algorithm based on Building Information Model," *Expert Syst Appl,* vol. 41, no. 16, pp. 7565-7578, 2014.

[15] M. Chen, S. Yan, S.-S. Wang, and C.-L. Liu, "A generalized network flow model for the multi-mode resource-constrained project scheduling problem with discounted cash flows," *Engineering Optimization,* vol. 47, no. 2, pp. 165-183, 2014.

[16] R. H. A. El Razek, A. M. Diab, S. M. Hafez, and R. F. Aziz, "Time-cost-quality tradeoff software by using simplified genetic algorithm for typical repetitive construction projects," *World academy of science, engineering and technology,* vol. 37, pp. 312-320, 2010.

[17] B. Koo, M. Fischer, and J. Kunz, "Formalization of construction Sequencing rationale and classification mechanism to support rapid generation of Sequencing alternatives," (in English), *Journal of Computing in Civil Engineering,* vol. 21, no. 6, pp. 423-433, Nov-Dec 2007.

[18] S. M. Chen, F. H. Griffis, P. H. Chen, and L. M. Chang, "Simulation and analytical techniques for construction resource planning and scheduling," (in English), *Automation in Construction,* vol. 21, pp. 99-113, Jan 2012.