Automation in Construction 69 (2016) 79–88

Contents lists available at ScienceDirect

Automation in Construction

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

Objective-driven and Pareto Front analysis: Optimizing time, cost, and job-site movements

Vahid Faghihi [a,*], Kenneth F. Reinschmidt [a], Julian H. Kang [b]

[a] Zachry Dept. of Civil Engineering, Texas A&M Univ., College Station, TX 77843-3136, United States
[b] Dept. of Construction Science, Texas A&M Univ., College Station, TX 77843-3137, United States

ARTICLE INFO

Article history:
Received 26 May 2015
Received in revised form 30 May 2016
Accepted 3 June 2016
Available online xxxx

Keywords:
Pareto Front
Optimization
Genetic algorithm
Construction project scheduling
Building information model

ABSTRACT

Finding the optimized trade-off relationship between cost and time, two important objectives of construction projects, helps project managers and their teams select a more suitable schedule for a given project. This trade-off relationship can roughly be estimated using past and cumulative knowledge, but since the early 1970s, researchers have been working on a systematic and mathematical solution to define this relationship more accurately. These researchers have used different optimization techniques such as the genetic algorithm (GA), ant colony, and fuzzy logic to further explore the relationship.

In the present paper, the authors have used their previously introduced construction schedule generator algorithm to present graphical relationships between pre-defined objectives of schedule optimizations. The process starts with developing construction schedules from the project's Building Information Model (BIM) as part of the input along with resource data. Then the process continues with optimization of all developed construction schedules according to the two mentioned objectives along with the introduced job-site movement objective, which mathematically helps the sequence of installation be more logical and practical. Finally generation of a 3D space for all the created and calculated construction schedules in the form of a 3D solution cloud point. These 3D construction schedules show solution cloud points and three Pareto Fronts for the given project.

1. Introduction

Extending or shortening a construction project's duration clearly affects the total construction cost. The most important aspect is how project time and cost are related, and how much a single change in either of them, can effect and change the other one. This means the in-between relationship needs to be formulated and shown graphically in order to bring a better understanding of the effects. Several successful attempts have been conducted to show this relationship. Different optimization tools have been applied to find the time–cost relationship of projects [10]. In most cases, optimization tools that can produce numerous outputs while optimizing the solutions (e.g., genetic algorithm) are selected for this type of research. This feature of having numerous outputs can result in a Pareto Front graph representing the relationship between the defined objectives. Therefore, for each optimization output (project schedule in this context), multiple objective scores are needed. A common problem is whether the original project schedules are comprehensive enough to cover all project elements and needed tasks. It is important to make sure that the initial project schedule represents the project well so that the optimization makes sense. The Building Information Model (BIM), on the other hand, contains all the project information in a 3D representation view. This source of project data can be and possibly should be used for the mentioned optimization purposes and to generate project specific time–cost optimization (TCO) graphs and reports.

The contribution of the current research is the ability to use the inherent data of a construction project from its BIM to generate the project schedule initially and then find and calculate the relationship between predefined objectives for the given project. Thus, the main purpose of this paper is to use the outputs of the previously developed algorithm to find the relationship between the defined objectives. These objectives are "cost," "time," and "job-site movements." As the first step toward conducting this research of finding objective relationships, the authors extracted and calculated a matrix of constructability relationships between all the elements directly from the BIM of the project and called our calculations a matrix of constructability constraints (MoCC).

Using the GA and the MoCC as the primary calculation basis for the GA fitness function, the authors developed a method that was able to generate valid construction sequencing of the building structure for the given 3D model [11]. By "a valid construction sequence," the authors imply that all the project elements are scheduled for installation in a

* Corresponding author.
E-mail addresses: svfaghihi@gmail.com (V. Faghihi), kreinschmidt@civil.tamu.edu (K.F. Reinschmidt), juliankang@tamu.edu (J.H. Kang).

http://dx.doi.org/10.1016/j.autcon.2016.06.003
0926-5805/

way that the structural stability requirements of the building are preserved throughout the construction process. To make the algorithm more mature and complete, the authors defined a new objective as “job-site movements.” This new objective is supposed to make element installation patterns more logical and acceptable by minimizing the distance between installation groups of each type of element. By minimizing the distances, the installation patterns of the elements tend to get more logical and doable in the construction process. The authors implemented this new objective along with cost and time in the GA optimization process. By developing this three-objective GA (cost, time, and minimal distance), the entire proposed method is able to generate constructible and optimized construction schedules only from the BIM of a project.

This section is followed by a comprehensive literature review on multi-objective genetic algorithm as the chosen methodology in this research. Then, Section 4 will provide a descriptive section on how the 3D model input is handled along with the process of generating MoCC. Next, Section 5 gives a definition of the genetic algorithm and lists the objectives. After that, input parameters of the entire algorithm are elaborated upon in Section 6, followed by Section 7 with tests and their results. Finally, the conclusion section and future works are summarized in Section 8.

2. Literature review

This research is mainly focused on using the genetic algorithm for the calculation and optimization of defined objectives. The genetic algorithm is selected as the optimization tool for this research, not because it is the only or the best optimization tool for multi-objective optimization problems, but because of the nature of the extracted data from the BIM of the project and how easily data can be translated to GA genomes. Obviously, other optimization methods, as long as they can use existing data from BIM, can replace GA in further research for evaluation and comparison purposes. The other optimization methods that have already been used for the time–cost optimization problem in the construction industry and are known by their ant colony optimization (e.g. [18,22,31]), particle swarm optimization (e.g. [20,32,33]), linear and integer programming (e.g. [21]), artificial intelligent system (e.g. [6]), and mathematical modeling (e.g. [23]). These methods can be evaluated later and compared to GA when using BIM data to do time–cost optimization, which can also be referred to as ant colony optimization. In current research, a 3D model along with other GA variables are imported into the genetic algorithm, after which, the algorithm yields numerous logical and constructible schedules for the given 3D model. These results from GA are used to generate 3D cloud point and Pareto front graphs. Therefore, the literature review for this research is limited to multi-objective genetic algorithms used for the optimization of construction projects.

For multi-objective optimization of construction schedules, the GA has been used successfully among researchers solving engineering problems [12]. In 1997, Feng et al. [12]) introduced a GA methodology for optimizing time–cost relationships in construction projects. They also produced a computer application based on their methodology, which could run the algorithm. Zheng et al. [35]) also showed their interests in using GA for time–cost trade-off optimization problems in construction projects. By comparing GA with other techniques, they showed that GA is capable of generating the most optimum results for the time–cost optimization (TCO) problems in large construction projects. They also presented their own multi-objective GA using the adaptive weight approach, which was able to point out an optimal total project cost and duration [36]. In their next step, they showed that using niche formation, Pareto ranking, and the adaptive weighting approach in multi-objective GA could result in more robust time–cost optimization results [37].

In 2005, Azaron et al. [4]) introduced their multi-objective GA for solving time–cost relationship problems, specifically in PERT networks. In their research they defined four objectives as minimizing project direct cost, minimizing mean of project duration, minimizing variance of project duration, and maximizing probability of reaching project duration limit. Another group of researchers developed their own multi-objective GA to reach set of project schedules with near optimum duration, cost, and resource allocation and embedded their algorithm as a MS Project macro [9]. In 2008, a multi-objective GA was introduced for scheduling linear construction projects and focused on optimizing both project cost and time as its objectives [26]. Hooshyar et al. [13] presented their GA time–cost tradeoff problem solver with higher calculation speed than made possible in the highly efficient Siemens' algorithm [27].

Abd El Razek et al. [1]) developed an algorithm that used the line-of-balance technique and critical path method concepts in a multi-objective GA. This proposed algorithm was designed to help project planners in optimizing resource usage. This resource usage optimization was conducted by minimizing cost and time while maximizing the project quality by increasing the resource usage efficiency. Late in 2011, Mohammadi [25]) introduced his MOGA (multi-objective genetic algorithm) that generated Pareto front in its approach toward solving the time cost optimization (TCO) problem in industrial environment. In 2012, Lin et al. [19] designed and introduced their multi-section GA model for scheduling problems. They combined that model with their proposed network modeling technique to perform automatic scheduling in the manufacturing system.

In recent years also, researchers have shown interest in new ways to solve the time–cost optimization problem. Amiri et al. [3] added quality to the TCO problem and used the Non-dominated Sorting Genetic Algorithm-II (NSGA-II) for time–cost–quality trade-off project scheduling problems (GPDTCQTP). Ke [15]) considered the indeterminacy of the environment in the proposed model and used the genetic algorithm to solve uncertain time–cost trade-off problems. Using Line of Balance (LoB) technique, Agrama [2]) used her MOGA for scheduling multistory buildings, in which project duration, total crews, and total interruptions were defined as conflicting objectives. Cheng and Tran [7]) proposed a novel approach by introducing their two-phase differential evolution (DE) model which was able to successfully reflect both time–cost effects and resource constraints. They [8] included opposition technique to their multi-objective DE, introducing the Opposition-based Multiple Objective Differential Evolution (OMODE), to solve the time–cost-utilization work shift tradeoff (TCUT) problem. They continued their work [28] on the TCO problem by proposing a new hybrid multiple objective evolutionary algorithm that is based on the hybridization of an artificial bee colony and DE (MOABCDE-TCQT). Later on Tran et al. [29]) showed the benefits of using their novel approach named “Multiple Objective Symbiotic Organisms Search” (MOSOS) to solve multiple work shift problems in the context of TCO problems by adding labor utilization. Lee et al. [17]) used the existing data from the project schedules for each individual task to find optimal set of parameters for GA as an advanced stochastic time–cost tradeoff (ASTCT) method to solve the TCO problem. Zhang et al. [34]) applied genetic algorithm in repetitive construction projects, such as bridges, to solve discrete time/cost trade–off problem (DTCTP) adding soft logic to make it more complex.

All of these researchers successfully tackled the time–cost trade-off problem in construction schedules, but a research information gap exists due to the lack of a way to ensure the complete and automated coverage of all the project elements in the calculations and scheduling. The techniques mentioned herein were able to calculate and retrieve enough data from the existing project schedules to solve TCO problem. However, the project schedules used had some inherent scheduling problems such as incompleteness in not covering the entire scope of the project and a lack of logic in not satisfying proper relations. It is obvious that problematic input will result in wrong and useless output in this type of optimization problem. Therefore, enriching the existing approach with automated project scheduling technique is a needed step toward eliminating the above mentioned problems.

In the current research, since the key input to the algorithm is the BIM of the project and the algorithm uses all the inherent geometry data from this input, full coverage of the project elements as well as well-defined processor relationships between elements is guaranteed. Considering all the project elements in both schedule development and later analysis, the end result of the new algorithm is more realistic and has proven accurate.

3. Methodology

The entire methodology is summarized in Fig. 1. As shown in this figure, the user inputs the BIM along with resource-related information on the project. Then, the matrix of constructability constraint (MoCC) algorithm detects the geometric information of the 3D elements and creates the matrix. This matrix will be the basis for the genetic algorithm to ensure all the generated construction schedules are structurally stable. Using the resource-related data, initially provided by the user, GA optimizes the resulting schedules toward multiple project objectives. This calculation can be repeated whenever there is any change in designs or other project parameters to obtain updated project schedules. In the proposed MoCC algorithm, in addition to beams and columns [11], other common building components are included. These other building components are slabs (floors), roofs, walls, doors, and windows. Adding this support for more element types makes it possible to input a more realistic 3D model into the algorithm. In addition, three more objective functions were defined to help generate more workable and reasonable project schedules by optimizing time, cost, and required job-site movements.

4. Reading the geometry

In this research, the 3D model input to the algorithm is in a standard data format, called Industry Foundation Classes (IFC). The IFC file format, as an open and neutral specification, is an object-based file format. The data model of this neutral file format is developed by buildingSMART with the main goal of facilitating interoperability between the AEC companies, as a commonly used file format for BIM [5]. The IFC model specification is listed as an official International Standard ISO 16739:2013 [14].

4.1. Detecting more 3D elements

The previously developed algorithm, which served to prove the usefulness of the methodology [11] supported the structural elements, columns (IfcColumn) and beams (IfcBeam). The extended research supports more general 3D elements from the IFC file format of the BIM of a project. These general 3D elements are geometric information of basic element types, such as slabs (IfcSlab), roofs (IfcRoof), walls (IfcWall), doors (IfcDoor) and windows (IfcWindow) to be detected, read, and calculated. For simplification of the geometric calculations, these element types were assumed as either lines or plain squares with a boundary box around them. The boundary boxes around elements are calculated with tolerances, different for each type, that make them slightly larger than their actual size. These tolerances of the boundary boxes ensure that when two elements are connected, they will be close enough to have intersecting boundary boxes that are interpreted by the algorithm as the connection of the two elements. An example of how the boundary boxes of beams or columns are assumed is shown in Fig. 2.

In this research, whenever two elements are close enough to have intersecting boundary box regions, they are assumed to be physically connected. After locating these connections by performing geometric calculation on the data retrieved from IFC files and applying stability rules as shown in Table 1, the MoCC can be generated. The information presented in this table simply shows the construction prerequisites that must be constructed. For instance, to install a wall, the lower level beams in the same region must be installed prior to the wall as well as any columns and/or beams that the wall is covering.

4.2. Generating MoCC

In the MoCC (shown in Eq. (1)), A_i represents elements as well as activities associated with each element indicating installation of that activity. Values of $s_{i,j}$ could be either one or zero, indicating immediate prerequisite installation or no relationship, respectively. For instance, $s_{2,6} = 1$ means that the element number 6, A_6, should be installed prior to installation of element 2, A_2. This matrix can be used in the genetic algorithm fitness function for checking the constructability of genomes in each generation. The following gives the MoCC calculation as

Eq. (1) — matrix of constructability constraints.

$$\text{Matrix of constructability constraints (MoCC)} = \begin{array}{c} \\ A_1 \\ A_2 \\ \vdots \\ A_n \end{array} \begin{array}{c} \begin{array}{cccc} A_1 & A_2 & \ldots & A_n \end{array} \\ \begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,n} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n,1} & s_{n,2} & \cdots & s_{n,n} \end{bmatrix} \end{array}$$

where:

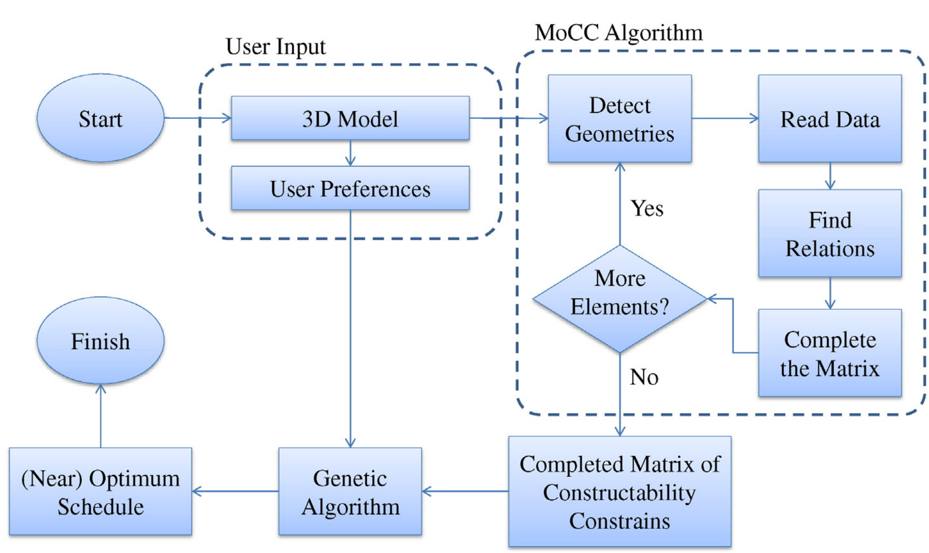


Fig. 1. Schematic view of the methodology.

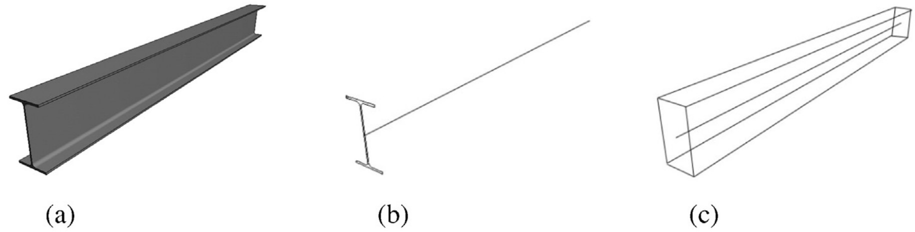


Fig. 2. 3D Element simplification, a) actual element, b) element section and extrusion line, c) element boundary box.

A_i: project tasks (geometric elements in the 3D model or the activities to be scheduled)

$s_{j,i}$ represents dependencies between elements (that could be either 0 or 1 showing not dependent or dependent respectively).

These relationships represent parent–child dependencies in the model. This means if $s_{2,6} = 1$ then element A_6 has a child relation to element A_2. For example, element 6 is a door and element 2 is a wall with that door installed in it.

Having the MoCC generated and ready, the next step would be defining the GA that can generate optimized construction schedules and produce 2D and 3D Pareto Fronts. The GA calculations are restricted by the constructability rules and constraints detected and defined in MoCC as well as resource limitations defined by the user. The resource limitations are representing the availability of materials and project crew to perform project tasks.

5. Genetic algorithm

The genetic algorithm is an optimization tool that uses a heuristic search and mimics the natural evolutionary process [24]. Using a well-defined fitness function as the objective function or core metric randomly generates initial genomes that can evolve into optimized solution(s) for a given problem, considering objectives that are mathematically defined by the fitness function. The GA has become a practical optimization tool in construction-related fields of research due to its inherent features and characteristics. In this paper, each GA genome matrix (MoG), to be used as a construction schedule in this case, is defined as shown in Eq. (2) to be used in the GA.

Eq. (2) — genome matrix (MoG).

$$\text{Genome matrix (MoG)} = \begin{array}{c} \\ A_1 \\ A_2 \\ \vdots \\ A_n \end{array} \begin{array}{c} \begin{array}{cccc} D_1 & D_2 & \ldots & D_k \end{array} \\ \begin{bmatrix} g_{1,1} & g_{1,2} & \cdots & g_{1,k} \\ g_{2,1} & g_{2,2} & \cdots & g_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n,1} & g_{n,2} & \cdots & g_{n,k} \end{bmatrix} \end{array}$$

or

$$\text{genome} = \{g_{1,1}, \ldots, g_{1,k}, g_{2,1}, \ldots, g_{2,k}, \ldots, g_{n,1}, \ldots, g_{n,k}\}$$

where:

A_i: project tasks i (geometric elements in the 3D model, also interpreted as the activities to be scheduled)

D_j: duration (time span) number j

$g_{i,j}$: indicates if the installation of element i is occurring in time span j or not, with 1 or 0, respectively.

n: number of project tasks (geometric elements in the 3D model or the number of activities to be scheduled)

k: total project time-unit (e.g., days, weeks, or months)

For instance, if $g_{2,6} = 1$, it means that element number 2 is scheduled to be installed in the 6th timespan.

Table 1
Stability prerequisites.

	Lower level	Same level
Column	Column	–
Beam	–	Supporting columns or beams
Wall	Beams	Adjacent columns and beams
Slab	Regional beams	–
Roof	Regional beams	–
Door	–	Container wall
Window	–	Container wall

5.1. Objective definition

The following three fitness function scores, which are calculated for each member (genome) of GA populations, should be summed up to a single value to treat the multi-objective GA as a single objective GA. The summation approach is described later in this section.

5.1.1. Project duration

As shown in Eq. (2), the length of a genome is equal to the number of project elements multiplied by the number of time units (duration). For any given genome, the duration can be calculated by dividing the length of the genome by the number of 3D elements. With the same concept the total construction duration associated with a MoG is equal to the number of matrix columns.

5.1.2. Project labor cost

As described earlier in this paper, the current research aims to introduce a framework that is capable of detecting the relationship between major construction schedule objectives, which are cost, time, and job-site movements. To build this framework, a few simplifications and assumptions are needed. For instance, a precise material take-off calculation is not in the scope of this research; therefore, the calculation of the project cost does not reflect the cost for material. The authors have calculated the project labor cost as follows: Initially, the algorithm collects resource-related data from the user. These user inputs contain the maximum number of each element type that could be installed in each day as well as the associated labor cost for those installations. The algorithm assigns the full labor cost for the days that the element installations do not exceed the maximum defined by the user. In case there is a day where the installation number exceeds the maximum limit, the extra installations will be assigned a labor cost 1.5 times that of the regular installation, considering them as overtime work. This increase in cost is based on the U.S. Fair Labor Standards Act of 1938 [30] that guarantees "time-and-a-half" for overtime in certain jobs.

5.1.3. Job-site movements

Workability of construction processes can be increased by reducing the required movements of the crew and machinery installing the project elements. Having the location information for all the pieces of the project from its BIM in addition to the installation sequence of them from a given schedule (genome), the distances between all installations in a single time-unit and also between time-units can be calculated. A short mathematical description of the calculation is that in each time-unit, the distances of all the scheduled installations for that time-unit to the central positioning point of those elements are calculated, and then the distances between the central positioning points of each time-unit are added to the sum. Minimizing the sum of total distances between installations of elements of each type will be another objective for the fitness function.

5.1.4. Calculation example

As an example of how to calculate the defined objectives, a simple structure and its represented MoG (Fig. 3b) for the construction sequence of the 3D model (Fig. 3a) is demonstrated.

5.1.4.1. Calculating duration objective. As mentioned earlier in the objective definition section, the construction duration for the given genome is equal to the number of the matrix columns, which is 5 unit-times. One unit-time will be described and defined by the user.

5.1.4.2. Calculating cost objective. If a user entered the following inputs to the algorithm:

Maximum number of columns per day: 4	Cost for this installation: $400/day
Maximum number of beams per day: 5	Cost for this installation: $300/day

Then, the calculation of the cost objective will be $400 for any project day that has 1 to 4 column installations and/or $300 for any day with 1 to 5 beam installations. If there is a day in which no column is scheduled to be installed, there would be no calculated cost associated to the column costs. On the other hand, if there are more than 4 columns (e.g. 6 columns) scheduled for a single day to be installed, the algorithm will multiply the cost of the extra column installations by 1.5, simulating the cost for overtime work. Therefore, if there is a day with six columns scheduled to be installed, the associated cost for that day for column installations would be:

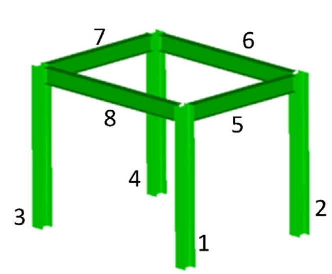
$$\mathrm{Cost} = 400 + \frac{400}{4\times1.5\times2 = 700}$$

The total cost for the given genome shown in Fig. 3b will be equal to a three-day cost of column installations ($400) and a three-day cost of beam installations ($300):

$$\mathrm{Cost} = 400 \times 3 + 300 \times 3 = 2100$$

5.1.4.3. Calculating movement objective. In the given genome, which has five unit-times as the total construction duration, the associated installation duration for each element of the model is indicated as one time-unit. For instance, element numbers 1 and 3 are scheduled to be installed in the first time-unit, and element number 8 is scheduled for the second. For calculation purposes, the authors assume that the distances between columns 1 and 3 is 10 ft, the height of the structure is also 10 ft, and it is symmetric in all directions.

To visualize the installation distances associated with the MoG and 3D model shown in Fig. 3, a schematic top-view of the distances with consideration of the sequencing is drawn in Fig. 4. In this figure when multiple elements are installed in a single time-span, the distances are shown with a dotted line with an arrow head on both ends, connecting

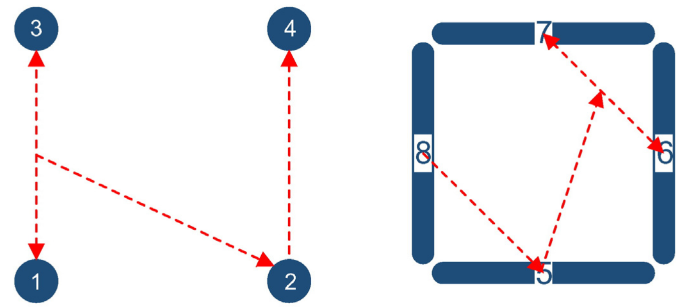


(a)

$$MoG = \begin{array}{c} \\ 1\\2\\3\\4\\5\\6\\7\\8 \end{array} \overset{\begin{array}{ccccc}1&2&3&4&5\end{array}}{\begin{bmatrix} 1&0&0&0&0\\ 0&0&1&0&0\\ 1&0&0&0&0\\ 0&0&0&1&0\\ 0&0&0&1&0\\ 0&0&0&0&1\\ 0&0&0&0&1\\ 0&1&0&0&0 \end{bmatrix}}$$

(b)

Fig. 3. (a) Sample structure and (b) a sample construction sequence.



(a) (b)

Fig. 4. Top-view of the installation distances for sample MoG: (a) for the columns and (b) for the beams.

all elements to be installed in that time span. The distance between installed elements in two time spans following each other is shown with a dotted line with arrow head on one end showing the direction of the steps, which connect the central positioning point of both sets of installing elements.

Fig. 4a shows a top-view for installation distances of the columns whereby columns 1 and 3 are installed together in the first time unit, column number 2 is in the third time unit, and column number 4 is in the fourth time unit. In the first time unit, the installation distance is equal to the distance between columns 1 and 3, which is 10 ft. There are no columns scheduled for the second time unit, which is accordingly skipped. In the third time unit, there is only one column installed; therefore, the installation distance would be equal to the distance from column 2 to the central positioning point of the previous installations (i.e., columns 1 and 3). For the fourth time-unit, the installation distance is simply calculated as the distance between column 4 and the last installation of the previous time span, column 2. With a similar concept and calculations, the total installation distances for beams shown in Fig. 4b can be calculated. Calculated lengths of installations for the given example are shown in Eq. (3a) and (3b). Notice that the total installation distances (or job-site movements required for installations) is equal to the sum of all the element types. In this example, the movement objective score is equal to the sum of total installation distances of columns and beams as shown in Eq. (3c).

Eq. (3) — the total installation distances in the sample presented as Fig. 4.

(a) Total installation distances for columns $= 10 + 11.18 + 10 = 31.18\,\mathrm{ft}$

(b) Total installation distances for beams $= 7.07 + 7.91 + 7.07 = 22.05\,\mathrm{ft}$

(c) Total installation distances $= 31.18 + 22.05 = 53.23\,\mathrm{ft}$

By defining this method in the fitness function of the GA in even more complex models, schedules with less job-site movement distance can be found in each generation cycle. By implementing this objective into the fitness function of the GA, the resulting construction schedules will have a more reasonable pattern of installation sequences as briefly shown in the above example.

5.1.5. Objective summation

Among the different approaches to calculate the fitness function score in multi-objective GAs [16], the classic "weighted sum approach" is adopted as a very computationally efficient approach. In this approach, the user needs to simply assign weights for each of the objectives defined in the fitness function. For each objective, the score for each genome is divided by the average score of the entire population for the same score to normalize the scores as shown in Eq. (4). This normalization of objective scores forbids any individual objective to dominate the final score. The user-defined weights would be multiplied by the respective normalized scores, and then all three scores are summed up to form the final score. The minimizing function is shown in Eq. (5), where the weight factors (w_d, w_e, and w_m) are inputs received from user defining any desired priority for objectives. The following calculation shows how to obtain the normalized score for a genome objective

Eq. (4) — normalization of an objective score of a genome.

$$\text{Normalized score for an objective of a genome} = \frac{\text{Objective score to the genome}}{\text{Objective score average from entire populations}}$$

Eq. (5) — weighted sum approach.

$$\min Z = w_d \times F_{\text{duration}}(x) + w_c \times F_{\text{cost}}(x) + w_m \times F_{\text{movement}}(x)$$

6. Inputs and variables

The model shown in Fig. 5a is a generic turbine building structural model with 42 columns and 58 beams, adding up to 100 elements. This generic model of a turbine building was previously used for proving the usefulness of this algorithm in developing structurally stable construction schedules extracted from a real-world and complex structural 3D model [11]. To test the extended research, a typical 3D model of a two-story residential building was selected, as shown in Fig. 5b. For the purpose of better internal viewing, the roof element is made invisible in the figure. This 3D model consists of 38 columns, 56 beams, 1 roof, 1 floor, 18 walls, 24 windows, and 7 doors, summing up to 145 elements. The 3D model shown in Fig. 5b is used in this research for elaboration of the results and Pareto Fronts.

The 3D model input along with the other input variables (shown in Table 2) was used to run the GA for testing the proposed algorithm with new extensions.

7. Results

Running the developed GA for 20,000 rounds of population generation with 10 genomes in each, for the 3D model, as shown in Fig. 5b, produced 200,000 valid construction schedules. Each of these generated construction schedules has three fitness function scores for the three objectives defined earlier.

7.1. 2D Pareto Fronts

Plotting each of two objectives, calculated in the results on a single 2D coordinate system, shapes the following outputs. Fig. 6 shows that by increasing the project duration, the labor cost for the project will be reduced before it starts to slightly increase. The reduction is due to decreasing the overtime work for the project. The latter increase of labor cost is caused by the growth in the number of working days. This interpretation matches the correct relationship between time and cost in real projects.

The time–movement Pareto Front graph as presented in Fig. 7 shows a constant and exponential increase in the movements when the project duration increases. This constant increase is a result of spreading out the element installations over the construction duration. Thus the distances between elements are not minimized by being installed as a group in a single day to have a shorter mid-point distance to the other set or setting up installation at the next time step.

As shown in Fig. 8, when the movement score is at its minimum (meaning that elements are installed in groups in each time step), the associated labor cost is higher, which is due to overtime installations of the elements. When elements are installed in sets of groups in each time step (e.g., six columns in a day), the total distance between each of these elements and also the distance between the mid-point of this installation group and the next one will be reduced based on the similar formulation and calculations shown in Eq. (3). On the other hand, when there are more elements to be installed in a single day than the user defined maximum limit (e.g., maximum of four columns per day), the surplus elements (e.g., two columns) have a labor cost 1.5 times higher than the regular installation. These two facts together make labor cost higher when the movement score lowers.

When the movement score exceeds a certain number and continues to increase, it indicates that the project elements installed are becoming much more scattered each day. Based on the authors' definition of labor cost, any number of installations per day can be equal to or less than the user-defined maximum and will have the user-defined associated labor cost for the day. Putting these two factors together, we know that the more the element installations are spread out in each day, the higher the labor cost will be.

Demonstrations of Pareto Fronts in all three graphs mentioned in this section show that this methodology shows the correct behavior of the three objectives defined in this research. The time–cost–movement relations described in this section match the same inter-relationships

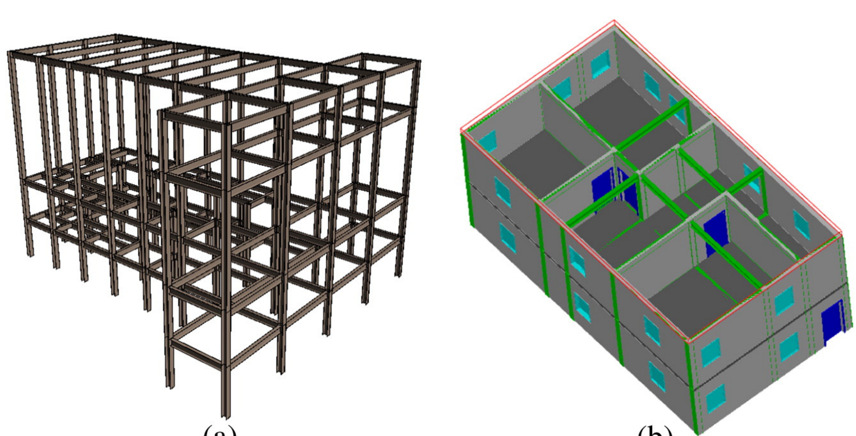


Fig. 5. 3D model input, (a) generic turbine building structural model, (b) typical 2-story residential building.

Table 2
Variable inputs to the GA calculations.

Title	Value
Population per generation	10
Elite rate	20%
Installment (maximum number of installations in each time-unit)	–
- Columns	2
- Beams	
- Walls	1
- Slabs	1
- Roofs	1
- Doors	2
- Windows	4
Cost/wage (dollars per each time-unit)	–
- Columns	500
- Beams	600
- Walls	300
- Slabs	400
- Roofs	450
- Doors	200
- Windows	320
Objective weights	–
- Duration	1
- Cost	1
- Movement	1

between these objectives as known in the construction field. These demonstrations and descriptions validate the use of this methodology.

7.2. Solutions cloud point

As described earlier in this paper, each of the generated project schedules has three objective scores for time, cost, and movement objectives. Fig. 6 through Fig. 8 showed how all 200,000 solutions can be represented in the 2D coordinate systems. Since there are three objectives in this paper, it is possible to show all the solution points in a single 3D scattered plot, as shown in Fig. 9. Further development of this analysis can generate a 3D Pareto Front surface showing the optimum relationship between the defined three objectives of this research.

7.3. Objective-driven analysis

Other than generating three previously shown Pareto Fronts and the solution cloud points, the proposed algorithm is capable of responding to the different project characteristics as being objective driven. For instance, the solutions from this algorithm can properly demonstrate the nature of cost- or time-driven schedules. To receive these types of reactions, the user needs to input the desired behavior for the project schedules when defining the objective weights right before running the GA calculations. In the previously shown 200,000 results from the algorithm, the defined weights for the objectives were set as equal. This equality of objective weights means that the GA calculations considered all three objectives with the same importance when the scores were summing up. Therefore, the changes in each objective had the same impact on the overall score for the genomes and thus on the chance for being selected as an elite member or for the crossover function of the GA.

In this paper, the authors show how the solution cloud points will be changed reflecting different objective weights. For this reason, three different runs with the same input 3D model and data have been conducted. In each of these three runs, one of the objectives received a weight of 100 while the other two were set to one. By inputting objective weights in this manner, in each of the new calculations, one of the objectives will be considered 100 times more important than the other two. This assumption will reflect the expectation of the user to have scheduling solutions driven toward a specific objective. For example, if the user sets the cost objective weight at 100 times more than the duration and movement objective weights, the algorithm will understand that the cost object is much more important than time and distance. In other words, the cost-driven solutions are requested by the user. Then, the algorithm will use that input to produce the construction schedules for the project.

Fig. 6 through Fig. 8 shows results from the calculation with the same objective weights for all three objectives. The following figures show how the cost-driven and time-driven calculations can differ.

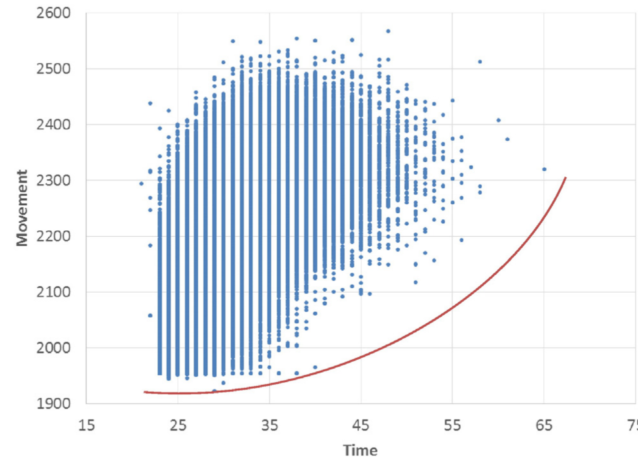


Fig. 7. Time–movement Pareto Front.

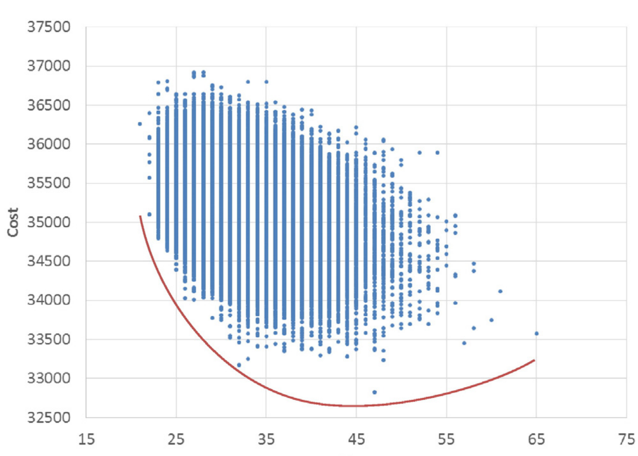


Fig. 6. Cost–time Pareto Front.

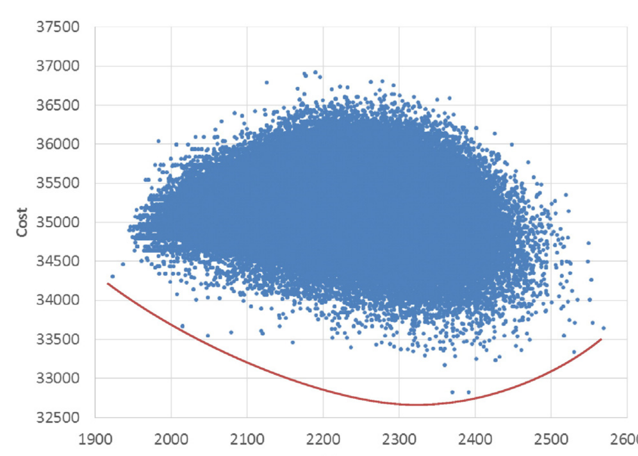


Fig. 8. Movement–cost Pareto Front.

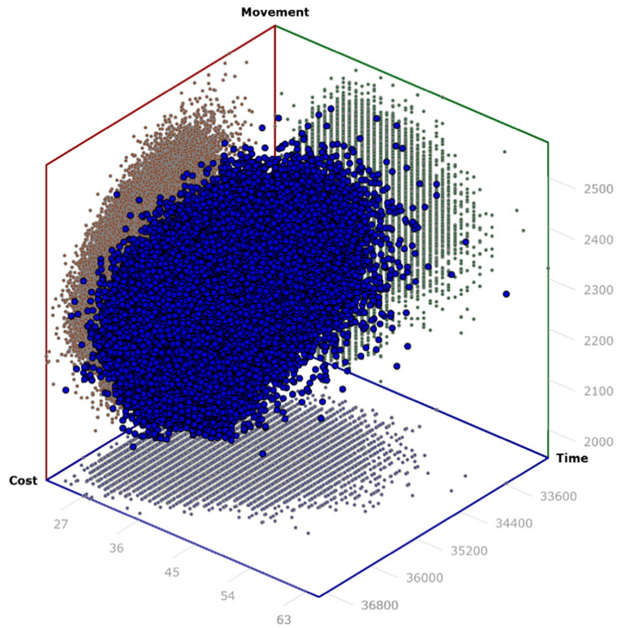


Fig. 9. Solutions 3D cloud point.

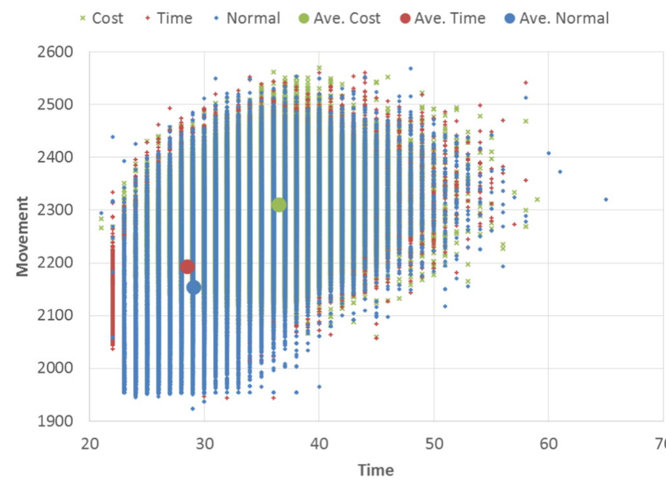


Fig. 11. Time–movement cloud point comparing objective-driven calculations.

As described earlier in this paper, by changing the weighting parameter for time and cost objectives in two different sets of runs from 1 to 100, while the other objective weights remain at one in that algorithm run, 200,000 new project schedules are generated for each run. Fig. 10 shows how the cloud points for the two new sets of calculations and construction schedules are different from the earlier run. As shown in the figure, when the calculation is set to be time-driven (time objective has a weight 100 times greater than the other two objectives), the entire cloud point (red pluses in the figure) shifts to the left of the graph. This left-shift means that the entire solution cloud point has construction schedules shorter than the normal calculation (blue dots) in which the weighted objectives are equal. The calculated average of the cloud points is shown as big red, green, and blue dots indicating average values for time-driven, cost-driven, and normal calculations, respectively.

Fig. 10 also shows that when the calculation is set to be time-driven (or cost-driven) the entire cloud point as well as the average point of the cloud shifts to the left for shorter construction durations (or shifts down for less construction labor costs in cost-driven runs). When the user intends to run the algorithm as time-driven, the algorithm produces more construction schedules with shorter duration. Imagine this example project has a constraint of a construction plan for a building to be built in less than 23 days. To satisfy this constraint the user needs to put more weight on the time objective in the calculation (e.g., 100 for time and 1 for the other objectives). By running the algorithm with this setting, the normal run generated 11 construction schedules with a duration of less than 23 days while the time-driven run generated more than 38,000 different construction schedules satisfying the constraint. Similar results can be discussed with the cost-driven algorithm calculations.

Also it is visible that in cost-driven calculation results, since the time objective had less weight (importance) set by the user, the average of the cloud point has been shifted to the right. This means that for cost-driven construction sequences, while the average cost has been reduced, the average time has been increased due to less importance of the time objective. Similar descriptions can be explained for other objectives and calculations.

Similar to Fig. 10, Fig. 11 shows the objective-driven calculations versus the normal ones, which had equal weights assigned to all the objectives. As seen in this figure, in both cost and time driven calculations, the average value for the movement objective has been increased. As similarly described before, this behavior is due to less importance (objective weight) assigned to the movement objective for calculations by the user. Fig. 12 shows the same behavior in its movement–cost graph.

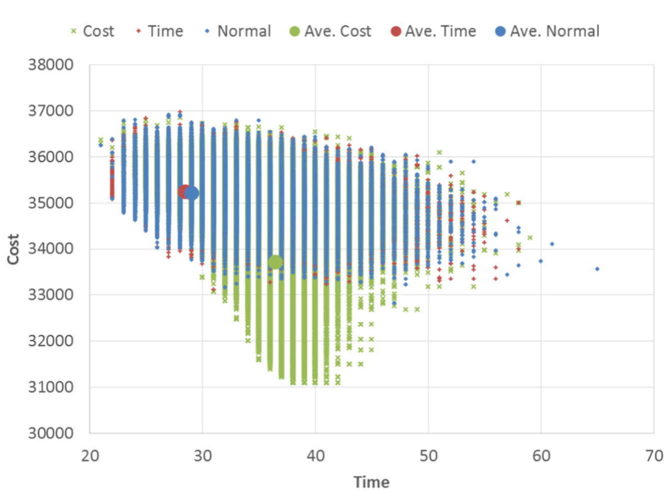


Fig. 10. Time–cost cloud points comparing objective-driven calculations. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

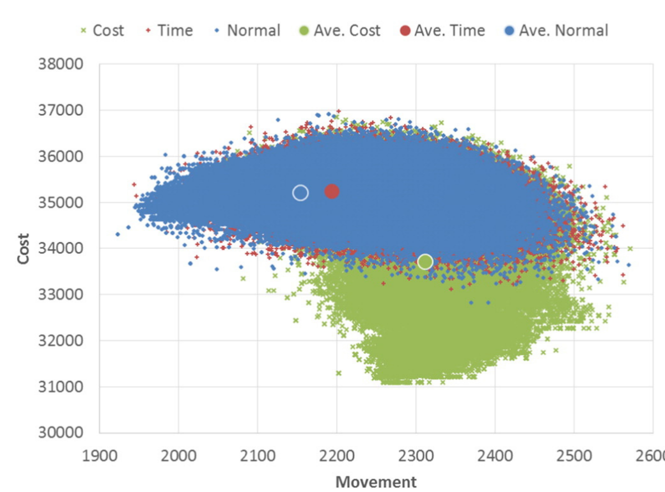


Fig. 12. Movement–cost cloud point comparing objective-driven calculations.

Table 3
Average score difference for normal vs. objective driven runs.

	Normal	Cost-driven	Time-driven
Cost	35,214.6	33,718.4	35,255.3
Time	29.1	36.5	28.5
Movements	2154	2311.4	2192.8

The differences in the average values of the cloud points in all three runs are shown in Table 3. As described before and as visible in Table 3, the cost objective score is increased in time-driven runs and decreased in cost-driven runs as expected. Likewise, the time score was reduced when calculations were time-driven and enlarged when cost-driven. The movement objective score was increased in both runs since the user-defined objective weight of this objective was set to the minimum.

8. Conclusion and future work

This article shows useful benefits of the proposed construction scheduling algorithm through the Pareto front relationship between optimization objectives of a construction project. The entire algorithm, as described in earlier papers, is able to read through the 3D model input in the form of IFC, detect all the structural stability dependencies and relations, and form a structural stability matrix called MoCC. Then, it uses that matrix as the basis for the GA fitness function to validate the structural stability wellness of the populations and produces project schedules while showing 4D construction animation [11]. The generated populations that contain construction schedules are ordered and handled based on their objective scores. Those construction schedules with better scores will have a better chance of reaching the next generations.

As mentioned in this paper, in addition to automatic construction schedule development, the proposed algorithm can also provide several managerial tools to help project managers and project management teams in their scheduling of projects. These managerial tools can provide two-by-two Pareto Front graphs for objectives, solution cloud points and a 3D Pareto Front surface (with later extensions). The algorithm's tools can also be used to reflect the objective driven nature of the project in the provided solutions for construction project scheduling problems.

Besides all of the mentioned benefits and advantages of this method, it still has some limitations, one of which is that it takes some time to produce the required number of rounds for GA calculations. When the project gets bigger, and its BIM model gets relatively more complex, the number of elements will increase and the time needed to prepare will also increase if acceptable results are to increase exponentially. Another limitation to the current algorithm is its support for a building's irregular shaped elements. If, for example, a building contains curved walls (similar to the Walt Disney Concert Hall) the existing algorithm cannot address that type of dimensions and connecting points for those walls and elements. Therefore, it cannot do the required calculations to generate a graph as shown in this paper.

For further extension on this research, other multi-objective optimization methods can be evaluated and investigated to see if better representation of the objectives' relationships can be determined. We also need to find a way to support more 3D element types so the input model can generate more realistic results.

References

[1] R.H. Abd El Razek, A.M. Diab, S.M. Hafez, R.F. Aziz, Time–cost–quality trade-off software by using simplified genetic algorithm for typical-repetitive construction projects, World Acad. Sci. Eng. Technol. 37 (2010) 312–321.
[2] F.A. Agrama, Multi-objective genetic optimization for scheduling a multi-storey building, Autom. Constr. 44 (2014, August) 119–128.
[3] M. Amiri, A.-R. Abtahi, K. Khalili-Damghani, Solving a generalised precedence multi-objective multi-mode time–cost–quality trade-off project scheduling problem using a modified NSGA-II algorithm, International Journal of Services and Operations Management 14 (3) (2013).
[4] A. Azaron, C. Perkgoz, M. Sakawa, A genetic algorithm approach for the time–cost trade-off in PERT networks, Appl. Math. Comput. 168 (2) (2005, September 15) 1317–1339.
[5] buildingSMART. (2013). buildingSMART, International home of openBIM. Retrieved June 22, 2013, from http://www.buildingsmart-tech.org/
[6] S.-M. Chen, F. Griffis, P.-H. Chen, L.-M. Chang, Simulation and analytical techniques for construction resource planning and scheduling, Autom. Constr. 21 (2012, January) 99–113.
[7] M.-Y. Cheng, D.-H. Tran, Two-phase differential evolution for the multiobjective optimization of time–cost tradeoffs in resource-constrained construction projects, IEEE Transactions on Engineer 61 (3) (2014, August) 450–461.
[8] M.-Y. Cheng, D.-H. Tran, Opposition-based Multiple Objective Differential Evolution (OMODE) for optimizing work shift schedules, Autom. Constr. 55 (2015 July)) 1–14.
[9] N. Dawood, E. Sriprasert, Construction scheduling using multi-constraint and genetic algorithms approach, Construction Management and Economics 24 (1) (2006) 19–30.
[10] V. Faghihi, A. Nejat, K.F. Reinschmidt, J. Kang, Automation in construction scheduling: a review of the literature, Int. J. Adv. Manuf. Technol. 81 (9) (2015, December) 1845–1856.
[11] V. Faghihi, K.F. Reinschmidt, J. Kang, Construction scheduling using genetic algorithm based on BIM, Expert Syst. Appl. 41 (16) (2014) 7565–7578.
[12] C. Feng, L. Liu, S. Burns, Using genetic algorithms to solve construction time–cost trade-off problems, J. Comput. Civ. Eng. 11 (3) (1997, July) 184–189.
[13] B. Hooshyar, A. Tahmani, M. Shenasa, A Genetic Algorithm to Time–Cost Trade off in project scheduling. Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on (pp. 3081–3086), IEEE, Hong Kong, 2008.
[14] International Organization for Standardization. (2013, March 21). Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries. (1). International Organization for Standardization. Retrieved June 22, 2013, from http://www.iso.org
[15] H. Ke, A genetic algorithm-based optimizing approach for project time–cost trade-off with uncertain measure, Journal of Uncertainty Analysis and Applications 2 (8) (2014).
[16] A. Konaka, D.W. Coitb, A.E. Smith, Multi-objective optimization using genetic algorithms: a tutorial, Reliab. Eng. Syst. Saf. 91 (9) (2006, Spetember) 992–1007.
[17] H.-G. Lee, C.-Y. Yi, D.-E. Lee, D. Arditi, An advanced stochastic time–cost tradeoff analysis based on a CPM-guided genetic algorithm, Computer-Aided Civil and Infrastructure Engineering 30 (2015) 824–842.
[18] H. Li, H. Zhang, Ant colony optimization-based multi-mode scheduling under renewable and nonrenewable resource constraints, Autom. Constr. 35 (2013, November) 431–438.
[19] L. Lin, X.-C. Hao, M. Gen, J.-B. Jo, Network modeling and evolutionary optimization for scheduling in manufacturing, J. Intell. Manuf. 23 (6) (2012, December) 2237–2253.
[20] H. Liu, M. Al-Hussein, M. Lu, BIM-based integrated approach for detailed construction scheduling under resource constraints, Autom. Constr. 53 (2015) 29–43.
[21] L. Liu, S. Burns, C. Feng, Construction time–cost trade-off analysis using LP/IP hybrid method, J. Constr. Eng. Manag. 121 (4) (1995, December) 446–454.
[22] X. Liu, Z. Ni, X. Qiu, Application of ant colony optimization algorithm in integrated process planning and scheduling, Int. J. Adv. Manuf. Technol. (2015) 1–12.
[23] G. Lucko, H.M. Said, A. Bouferguene, Construction spatial modeling and scheduling with three-dimensional singularity functions, Autom. Constr. 43 (2014, July) 132–143.
[24] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, Cambridge, MA, 1996.
[25] G. Mohammadi, Using genetic algorithms to solve industrial time–cost trade-off problems, Indian Journal of Science and Technology 4 (10) (2011, October) 1273–1278.
[26] A. Senouci, H.R. Al-Derham, Genetic algorithm-based multi-objective model for scheduling of linear construction projects, Adv. Eng. Softw. 39 (12) (2008, December) 1023–1028.
[27] N. Siemens, A simple CPM time–cost tradeoff algorithm, Manag. Sci. 17 (6) (1971, February 1) B-354–B-363.
[28] D.-H. Tran, M.-Y. Cheng, M.-T. Cao, Hybrid multiple objective artificial bee colony with differential evolution for the time–cost–quality tradeoff problem, Knowl.-Based Syst. 74 (2015, January) 176–186.
[29] D.-H. Tran, M.-Y. Cheng, D. Prayogo, A novel Multiple Objective Symbiotic Organisms Search (MOSOS) for time–cost–labor utilization tradeoff problem, Knowl.-Based Syst. 94 (2016, February 15) 132–145.
[30] U.S. Department of Labor. (2009, September). Wages and Hours Worked: Minimum Wage and Overtime Pay. Retrieved March 11, 2013, from United States Department of Labor: http://www.dol.gov/compliance/guide/minwage.htm
[31] Y. Xiong, Y. Kuang, Applying an ant colony optimization algorithm-based multiobjective approach for time–cost trade-off, J. Constr. Eng. Manag. 134 (2) (2008, February) 153–156.
[32] I. Yang, Using elitist particle swarm optimization to facilitate bicriterion time–cost trade-off analysis, J. Constr. Eng. Manag. 133 (7) (2007, July) 498–505.
[33] H. Zhang, X. Li, H. Li, F. Huang, Particle swarm optimization-based schemes for resource-constrained project scheduling, Autom. Constr. 14 (3) (2005, June) 393–404.