AUTOMATED AND OPTIMIZED PROJECT SCHEDULING USING BIM

A Dissertation

by

VAHID FAGHIHI

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Kenneth F. Reinschmidt
Co-Chair of Committee,	Julian H. Kang
Committee Members,	Stuart D. Anderson
	John Walewski
Interim Department Head,	Robin Autenrieth

May 2014

Major Subject: Civil Engineering

ABSTRACT

Construction project scheduling is one of the most important tools for project managers in the Architecture, Engineering, and Construction (AEC) industry. The Construction schedules allow project managers to track and manage the time, cost, and quality (i.e. Project Management Triangle) of projects. Developing project schedules is almost always troublesome, since it is heavily dependent on project planners' knowledge of work packages, on-the-job-experience, planning capability, and oversight. Having a thorough understanding of the project geometries and their internal interacting stability relations plays a significant role in generating practical construction sequencing. On the other hand, the new concept of embedding all the project information into a three-dimensional (3D) representation of a project (a.k.a. Building Information Model or BIM) has recently drawn the attention of the construction industry.

In this dissertation, the author demonstrates how to develop and extend the usage of the Genetic Algorithm (GA) not only to generate construction schedules, but to optimize the outcome for different objectives (i.e. cost, time, and job-site movements). The basis for the GA calculations is the embedded data available in BIM of the project that should be provided as an input to the algorithm. By reading through the geometry information in the 3D model and receiving more specific information about the project and its resources from the user, the algorithm generates different construction schedules. The output Pareto Frontier graphs, 4D animations, and schedule wellness scores will help the user to find the most suitable construction schedule for the given project.

DEDICATION

In the name of GOD, the most compassionate, the most merciful

This work is dedicated to:

The owner of the age and the time,

Hojjah son of Al-Hasan son of Ali son of Mohammad

son of Ali son of Musa son of Jafar son of Mohammad son of Ali son

of Al-Husain son of Ali, father of Al-Hasan, husband of Fatima Al-Zahra, and

brother of Prophet Mohammad, peace be upon them all until the day of Judgment.

ACKNOWLEDGEMENTS

I would like to thank my both committee chairs, Dr. Reinschmidt and Dr. Kang, for their guidance and support throughout the course of this research. I also thank my committee members, Dr. Anderson and Dr. Walewski, for helping me with their comments and recommendations.

Thanks also go to my father and my mother for their encouragement and support.

And finally, the special thanks go to my wife for her great companionship, patience, and love.

NOMENCLATURE

3D	Three Dimensional
4D	Four Dimensional
BIM	Building Information Model
CAD	Computer Aided Design
CPM	Critical Path Method
DSM	Design Structure Matrix
GA	Genetic Algorithm
IFC	Industry Foundation Classes
MoCC	Matrix of Constructability Constraints
MoG	Matrix of Genome
NoG	Number of Genes
PERT	Program Evaluation and Review Technique
SD	System Dynamic
WBS	Work Breakdown Structure

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

LIST OF EQUATIONS

Page

CHAPTER I

INTRODUCTION

The Project Schedule

In the area of project management, a schedule is the list of activities, milestones, and deliverables of the project. Project schedules control project time, cost, and resources, as an essential part of construction engineering and management. More than 140 software packages have been produced to generate, develop, and manage project schedules (Comparison of Project Management Software, 2012). These software packages help schedulers and managers to manage multiple projects (project portfolio).

Problem Description

Project schedules are the main tool for project managers and engineers to control time and cost; however, they are still heavily dependent on the scheduler's experience and still largely an interactive manual process of trial-and-error (Hendrickson, Zozaya-Gorostiza, Rehak, Baracco-Miller, & Lim, 1986). The project scheduler schedules the activities (sets the start and finish date) and the computer application computes the time and cost associated with each trial schedule. The construction planning processes usually have the following characteristics (Zozaya-Gorostiza, Hendrickson, & Rehak, 1989):

- manually formulated
- performed intuitively and unstructured
- based on previous projects, construction experience, and engineering judgment

Project plans are seldom optimal toward time, cost, or other objectives due to dependence on planners' randomly accumulated experience (Firat C. E., Kiiras, Kähkönen, & Huovinen, 2007) and limited time to generate alternate schedule solutions. In addition, poor schedules or estimates can easily cause delays or cost increases in large projects (Zozaya-Gorostiza C. , Hendrickson, Rehak, & Lim, 1988). The manual scheduling process can cause multiple drawbacks, which can be listed as the following (Mikulakova E. , König, Tauscher, & Karl, 2010):

- planning process errors (logical/precedence)
- difficulties in performing tasks on the job-site (workability)
- not easily updatable based on project changes
- developed often by subcontractors without adequate overall project coordination
- time consuming updates, and
- depending on schedulers personal project experiences

Knowing the potential drawbacks in the manual process of project scheduling, a tentative solution can be anticipated, automatically generated project schedules that contain all the project components. The solution should be automatic so the process of project scheduling can be performed on demand rapidly while it eliminates the potential problems of the manual process. This solution should also cover all the project elements in order to be considered as a complete scheduling tool for the construction purpose of the project. To solve some of the mentioned scheduling problems and reach to the tentative solution, the following list of information about the project elements (components) is needed:

- Complete list of elements and their types
- Size, volume, or dimension data
- Spatial or location information
- Dependencies of the elements

The above list of information can be summarized as geometric data and physical information of the project elements. This list of needed information can be found in a well-developed three-dimensional (3D) model that could include all the elements as well as their geometric data.

The recently developed Building Information Modeling (BIM) concept is a 3D representation of construction buildings with the inherent data of all the project elements. This database of project element information is increasing rapidly in Architecture, Engineering, and Construction (AEC). Knowing that the geometric and topological information required to build the project is embedded in the BIM, this model can be counted as an essential input source for the project schedule.

Focusing Question

The author of this study listed potential drawbacks of the manual scheduling process as well as the severe impact of a poor schedule on the project time and cost. The proposed tentative solution needs complete geometric information for all the project elements. The usage of the project BIM makes the project components' geometric data available. Thus, if an algorithm can be developed to take the project BIM as the input data, all the spatial data should be accessible. Considering the author's hypothesis that, by knowing the geometric information of project elements, a construction schedule can be generated, the focusing question of this research will be “How can a computer-based algorithm generate project schedules using the spatial information of project components?”

Research Purpose

Computer-based algorithms are faster than humans in calculating math and solving defined mathematical equations and problems. In solving project scheduling problems, the computer-based algorithm can generate and develop multiple project schedule alternatives due to their high calculation speed. Having multiple valid and feasible schedules for a given project, professional project schedulers can select the one that suits the project based on their overall insight and accumulated experience. In addition, inexperienced schedulers can use the high calculation speed of computer-cased algorithms to observe the impacts of different factors on the project schedules. These schedulers can modify input factors to the algorithm and immediately see its impact on the resulting schedules.

Research Nature

The project scheduling problem has a unique nature of its own. Rational project schedules are those with less duration and less cost, while they are more constructible in the job-site. Defining and measuring more objectives can be helpful to determine the better project schedule among many other schedules. This need of multiple objectives to show how well-suited a project schedule is, regarding satisfying defined objectives, makes the problem a *multi-objective* problem. Meanwhile, optimizing project schedules toward minimum time and cost are some of the best identified opportunities for the application of optimization in the commercial world (Ward Systems Group, Inc., 2007).

Another characteristic of the scheduling problem is its search space. Any changes to task durations, start times, or lags and relationships between tasks will result in a new project schedule. These characteristics can create numerous different project schedules for a single project and all together construct the search space for the scheduling problem of the given project. Since this multidimensional search space contains a large number of solutions, an algorithm should be used that avoids an exhaustive search through all the available solutions most efficiently.

In addition, finding the best project schedule for a project could be too computationally intensive. Also, in most of the scheduling cases, the schedulers or project managers are not looking for the theoretically best project schedule since they know the dynamically-changing nature of projects. Therefore, an algorithm that finds near optimum and feasible solutions can be used.

Research Objectives

The two main objectives of this research are as follow:

- Develop an algorithm that automatically generates optimized project schedules using a 3D model.
- Develop an application to test if the proposed algorithm is capable of generating feasible construction schedules.

Beside the main objectives, another achievement is anticipated from this research work, which is described more in the last chapter. This minor objective is to support the use of the interactive 3D BIM interfaces in classrooms and computer laboratories, so that students can learn and practice construction planning and scheduling using the computer models.

Dissertation Outline

Chapter II contains the related literature for finding a well-suited automation and optimization tool for project scheduling. Chapter III describes the algorithm for detecting project elements dependencies and relations from the three-dimensional representation of the project. Chapter IV shows how the chosen optimization tool can generate structurally stable and constructible project schedules for the 3D model. Chapter V extends the usage of the optimization tool to generate better project schedules regarding defined metrics. Chapter VI illustrates other usages of the proposed algorithm in construction industry. The last chapter suggests possible further extensions for the algorithm and potential benefits that the proposed algorithm can offer for professionals in project scheduling.

CHAPTER II

AUTOMATION IN CONSTRUCTION SCHEDULING: A REVIEW OF THE LITERATURE*

Introduction

The project schedule, specifically in construction projects, is a tool that helps project managers and project management teams handle several critical aspects of management. Through construction schedules they manage time, cost, resources, etc. Having the ability to ensure enough information is available to the management team makes the construction schedule one of the most, if not the most, vital gears for managing the projects. Knowing these facts about the importance of the project schedules, their development should be done very carefully. Based on the nature of construction works, as a common way to initiate and develop a construction schedule, the developer's background knowledge and experience plays a very critical role. In case the scheduler does not have enough information from or correct understanding of the project and its scope, that helpful construction schedule turns into a time and cost consuming tool, which also misleads the project workers. To solve information lacking problems, researchers have been focusing on automating the process of schedule generating.

---

* This chapter is submitted to "Journal of Computing in Civil Engineering" as an individual paper and is under review (Faghihi, Reinschmidt, & Kang, Automation in Construction Scheduling: a Review of the Literature, 2014).

The research interest on automatically generating and optimizing the construction schedule has been around for almost four decades starting from early 1960s (Newell & Simon, 1972). They have been trying to find better ways to use computer-based algorithms and applications to ease the process of scheduling the projects. Some of the researchers focused directly on the cumulative past knowledge of the construction works as a database and scheduled new projects accordingly. Others have used project information models, as their input, to reach the desired outcome. When expert systems were common as a research tool, some researchers tried to use its advantages to generate schedules. Introducing neural networks opened another door for researchers in this field to mimic the way human brains work regarding the project scheduling. As a well suited optimization tool, many researchers showed their interest in optimizing project resource allocation and leveling using the Genetic Algorithm (GA). Different methods such as predicting project's future and using the System Dynamics for the program design process are other computer aided ways to have better project schedules.

The previous researches are divided into the following sections: Case-Based Reasoning and knowledge-based, model-based, Genetic Algorithm, Expert Systems, Neural Networks, and a few other ways of solving the scheduling problem. Each section starts with the definition of the section that follows by related research works to the section.

Case-based Reasoning & Knowledge-based Approaches

Definitions

Case-based reasoning (CBR), as an essentially different tool from the other major artificial intelligent tools, is able to exploit the specific knowledge of formerly practiced situations (Aamodt & and Plaza, 1994). The CBR method remembers an earlier situation comparable to the present problem or situation and uses that earlier data to solve and explain the new one. This method can adapt and use older situations (cases) to explain, critique, or cause new situations (Kolodner, 1992). The main features of CBR can be summarized as below (Watson & Marir, 1994):

- It does not need a specific domain model
- Its application is reduced to “identifying significant features that describe a case”
- It uses databases to handle huge amount of information
- It learns by receiving new knowledge in the form of new cases

Research Works

In the late 1980s, Navinchandra et al. described their GHOST network generator. The GHOST was able to take activities as the input to the system and a develop precedency network for those activities as the output, considering knowledge about construction rules, basic physics, etc. (Navinchandra, Sriram, & Logcher, 1988). Benjamin et al. (1990) proposed a knowledge-based prototype for the purpose of planning and scheduling construction projects. Their prototype was aimed at increasing the productivity of inexperienced schedulers and also to generate schedules using the system. Their system helped the schedulers in identifying precedence relationships and work breakdown structures (WBS) by mimicking the process of an expert's decision making. Another group of researchers worked on a knowledge-based planning system (a.k.a. KNOW PLAN) in which the artificial intelligence (AI) and computer aided design (CAD) are integrated for generating and then simulating construction schedules (Morad & Beliveau, 1991). Echeverry et al. (1991) listed four basic factors effecting the sequencing construction activities: physical relationships, construction trades interactions, interference-free paths of the objects, and code regulations. Then, they proposed their own developed knowledge-based prototype system that used some of the mentioned factors to publish project sequencing plan. Schirmer (2000) integrated heuristic and case-based reasoning approaches for resource-constrained project scheduling problems. In his paper, he verified the proposed algorithm and described how to develop such a CBR system.

Muñoz-Avila and his team started working on developing a CBR solution for generating construction schedules in 2001. As their first step, they introduced their novel case-based planning algorithm, named SiN. SiN was able to generate project plans using previously provided cases while an incomplete domain theory is given (Muñoz-Avila, et al., 2001). Then, they focused on how to acquire proper cases from a project automatically or with minimum end user efforts (Mukkamalla & Muñoz-Avila, 2002). Then, they used their integrated plan retrieval model (CBR) to help the project planners create WBS more efficiently (Knowledge-Based Project Planning, 2002). Later in 2003, they described how to use justification truth-maintenance system (JTMS) technology for further development on the algorithm (Xu & Muñoz-Avila, ICCBR 2003, 2003). By the use of this technology along with CBR module they were able to create an interactive environment in which a user can either edit the project schedule or retrieve a case from the database to be reused in the scheduling process. They also presented their CBM-Gen+ algorithm that revised and edited the available cases in the database when there was a new solution (Xu & Muñoz-Avila, ICCBR 2003, 2003). With this revision on the existing case, the chance of inconsistency between the cases was reduced. Ultimately, they proposed their CBR solution called CaBMA (Case-Based Project Management Assistant), which was developed as an add-in extension for Microsoft Project (Xu & Muñoz-Avila, IAAI 2004, 2004). This software was able to properly identify the cases from the project plans, reuse the previously captured cases to generate a new plan and preserve the consistency of the entire project schedule (Xu & Muñoz-Avila, CaBMA: a Case-based Reasoning System for Capturing, Refining, and Reusing Project Plans, 2008). They also worked on another software called DInCAD (Domain-Independent System for Case-Based Task Decomposition) that consisted of all the four main steps of CBR in addition to the idea of re-using globalized cases to suite the new problems (Xu & Muñoz-Avila, AAAI 2005, 2005). This last research was also published in details as a PhD dissertation (Xu, Case-Based Task Decomposition with Incomplete Domain Descriptions, 2006).

König and his team also were interested in this field of research and began their research in 2006. They presented a way to generate various task ordering alternatives for a construction plan along with evaluation on each alternative (König, Beucke, & Tauscher, 2006). Their algorithm was able to automatically generate project schedules at any time and took the advantage of using Feature Logic theory to associate existing constraints (Tauscher, Mikulakova, König, & Beucke, 2007). Later on, they used 3D model data in the form of Industry Foundation Classes (IFC) along with the cases from previous projects. When a new scheduling for the given 3D model needed to be generated, their algorithm used Feature Logic to identify the cases in the 3D model and CBR retrieved the most similar case(s) from the database, using the proposed evaluation method. The solution to the scheduling problem would then be presented to the project manager and the final approval from him/her will be added to the database as a new case (Mikulakova E. , König, Tauscher, & K., 2008). Therefore they have used Building Information Model (BIM) for identifying the subjects to be scheduled and by retrieving, reusing, revising, and retaining the learned experiences as cases from CBR (Mikulakova E. , König, Tauscher, & Karl, 2010).

Model-based

Fischer and his team have shown their interest in using project models as the input for their algorithm for developing construction schedules from 1994. Based on their work in the Center for Integrated Facility Engineering (CIFE) at Stanford University, they extended the idea of automatic project schedules by adding models of construction methods. Their system, known as MOCA, used formalized construction method models to perform the scheduling based on product models (Fischer, Aalami, & O'Brien Evans, Model-Based Constructibility Analysis: The MOCA System, 1994). They defined five characteristics for each method as the following: constituting activities, domain, constituting objects, resource requirements, and activity sequencing.

These methods were describing higher-level activities of the schedule into lower-level ones to ease the linking of the schedules with diverse level of details (Fischer & Aalami, Scheduling with Computer-Interpretable Construction Method Models, 1996). Then, they presented their constructability knowledge approach tested for reinforced concrete structures. This approach was divided into the following five items: layout knowledge, application heuristics, dimensioning knowledge, exogenous knowledge, and detailing knowledge (Fischer & Tatum, Characteristics of Design-Relevant Constructability Knowledge, 1997). In their next step they approached to use component-based CAD models as their source of data. They discussed about shortcoming of common 4D (3D+time) models and showed the planning support as a requirement for CAD tools. Also, they proposed their own solution to the scheduling problem by generating 4D+x models for showing construction processes more accurately (McKinneya & Fischer, 1998). After few years, they addressed the Critical Path Method (CPM) limitations in rescheduling by defining a “constraint ontology” and “classification mechanism”. They implemented their method as a prototype that can quickly find out which tasks should be postponed to accelerate bottleneck tasks or critical milestone (Koo, Fischer, & Kunz, 2007).

Firat was another researcher interested in automated solutions for the scheduling problem. He proposed the Building Construction Information Model (BCIM) including three models: building product model (BPM), building construction resource, and cost model (BPRCM), and building construction process model (BCPM) (Firat C. E., Kiiras, Kähkönen, & Huovinen, 2007). These three sub-models were focusing on design objectives, resource objectives, and activity objectives respectively (Firat, Kiiras, & Huovinen, AEC2008, 2008). Then, he used the location-based Advanced Line of Balance (ALoB) as the output of his proposed methodology to show and solve the scheduling problem (Firat, Kiiras, & Huovinen, 2008). This model consisted of two steps; the first one generated a master schedule with the help of mentioned sub-models. On the second step the project manager input detailed information to come up with an extended schedule based on the mater schedule from previous step (Firat C. , Arditi, Hämäläinen, & Kiiras, 2009). Finally he extended his model to be able to perform quantity take-off in residential construction projects using his BCIM sub-model along with ALoB method (Firat C. E., Arditi, Hamalainen, Stenstrand, & Kiiras, 2010).

Vriesa and Harink (2007) presented their algorithm that extracted the construction sequence from a 3D model of the building. They detected the adjacency inferences and used the approach of displacing objects downward to find intersecting components. Tulke et al. addressed common object splitting problems in using BIM for scheduling and proposed their algorithm for BIM objects boundary representations as defined in IFC (Tulke, Nour, & Beucke, 2008). Karaoka (2008) described his new way for automating construction simulations with the help of "construction method templates" stored as knowledge base. König's works described earlier could also mentioned here as model-based approaches since his team explicitly used IFC as 3D model input to their algorithm (Tauscher, Mikulakova, Beucke, & König, 2009). Büchmann-Slorup and Andersson (2010) reviewed the constrcution scheduling process taking into acount the BIM-based approaches and the way to implement BIM in scheduling. In 2012, Wledu and Knapp developed a “rule-based spatial reasoning” method that used the BIM component topological relationships and automatically generated meaningful schdules for constructing the given 3D model (Weldu & Knapp, 2012).

Genetic Algorithm

Definitions

The Genetic Algorithm is an optimization tool that uses a heuristic search mimicking the natural evolutionary process (Mitchell, 1996). Using a well-defined fitness function, as the objective function or the core metric, the initial randomly-generated genomes can evolve into optimized solution(s) for a given problem. This optimization is based on the objective(s) that are mathematically defined by the fitness function. The GA is known as a popular meta-heuristic optimization method that is mainly suitable for solving multi-objective problems (Konaka, Coit, & Smith, 2006), such as construction scheduling.

Research Works

Davis introduced the use of the Genetic Algorithm for optimization of job shop scheduling in the 1980’s (Davis, 1985). Few years later, Wall used this algorithm for resource constrained scheduling as his dissertation topic (Wall, 1996). He optimized the sequencing of job shop tasks by feeding the GA with more than 1,000 different types of scheduling problems ranging from small job shop to project scheduling (10-300 activities, 3-10 resource types). Chan et al. presented their work as scheduling of resource-constrained construction projects using GA (Chan, Chua, & Kannan, 1996). In their paper, they showed how their proposed GA-Scheduler can optimize the resource usage and do the resource leveling to come up with better project schedules compared to the heuristic methods in regard to resource allocation. Gonçalves et al. continued this work later by tackling resource-constrained multi-project scheduling (Gonçalves, Mendesb, & Resendec, 2008). Murata et al. (1996) introduced their multi-objective GA to reach Pareto fronts of flowshop scheduling and described how their GA was developed.

In 2002, Toklu (2002) used Genetic Algorithm for construction project scheduling for both having and not having resource constraints. He used a model for defining the relationships between the network activities (Start-to-Start or SS, Start-to-Finish or SF, Finish-to-Start or FS, and Finish-to-Finish or FF). Toklu simplified the relationships by defining basic mathematical equations; for instance he defined the Start-to-Start relation between task i and task j as $T_i + L_{ssij} \leq T_j$, where L_{ssij} is the start-to-start time lag between task i and task j. As seen in the above-defined mathematical relation, there can be different relationship types between task i and task j, but the defined mathematical relation still remains valid. For instance, if the in-between lag, L_{ssij}, is bigger than the duration of task i, a Finish-to Start relation between task i and taks j with smaller lag, $T_i + L_{fsij} \leq T_j \; while \; L_{fsij} = L_{ssij} + D_i$, can be valid too. Jaśkowski and Sobotka (2006) introduced an Evolutionary Algorithm (also called GA) to minimize the duration of project construction given structural relationships, available resources, and resource requirements associated with each project task (Jaśkowski & Sobotka, 2006).

For multi-objective optimization of construction schedules, the GA has been used successfully amongst the researchers solving engineering problems (Feng, Liu, & Burns, 1997). In 1997, Feng et al. (1997) introduced a GA methodology for optimizing time-cost relationship in construction projects. They also produced a computer application based on their methodology that could run the algorithm. Zeng et al. 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 (Zheng, Ng, & Kumaraswamy, 2002). They also presented their own multi-objective GA using adaptive weight approach, which was able to point out an optimal total project cost and duration (Zheng, Ng, & Kumaraswamy, Applying a Genetic Algorithm-Based Multiobjective Approach for Time-Cost Optimization, 2004). On their next step, they showed that using niche formation, Pareto ranking, and adaptive weighting approach in multi-objective GA could result in more robust time-cost optimization results (Zheng, Ng, & Kumaraswamy, 2005).

In 2005, Azaron et al. (2005) 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 (Dawood

& Sriprasert, 2006). 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 (Senouci & Al-Derham, 2008). Hooshyar et al. presented their GA time-cost tradeoff problem solver with higher calculation speed than Siemens algorithm (Hooshyar, Tahmani, & Shenasa, 2008). Similar research on this topic has been conducted by Senouci and Al-Derham, focusing of multi-objective GA-based optimization. They implemented their algorithm for scheduling linear construction projects (Senouci & Al-Derham, 2008).

Abd El Razek et al. developed an algorithm that used Line of Balance 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 (Abd El Razek, Diab, Hafez, & Aziz, 2010). Late in 2011, Mohammadi introduced his MOGA (Multi-Objective Genetic Algorithm) that generated Pareto front in its approach toward solving the TCO problem in industrial environment (Mohammadi, 2011). In 2012, Lin et al. (2012) designed and introduced their multi-section GA model for scheduling problems. They showed the combination of that model with their proposed network modeling technique can do the automatic scheduling in the manufacturing system.

Expert Systems

Definitions

An expert system, as one of the artificial intelligence subsets, is defined as a computer-based algorithm that imitates the human's decision-making skill (Jackson, 1999). Expert systems are generated for resolving complex and difficult problems by reasoning about knowledge. These systems are designed mainly using IF-THEN structures instead of regular practical codes (McGartland & Hendrickson, 1985). The initial development of the expert systems occurred in the 1970s and then became more mature in the 1980s (Durkin, 2002).

Research Works

Hendrickson and his team started their work on using expert system method for construction scheduling problem in mid-80s. In their first attempt, they evaluated how an expert system can be used for controlling a project in two sets of aspects by defining sample if-then structures: cost and time control and purchasing and inventory control (McGartland & Hendrickson, 1985). Then, they further developed their idea, as a prototype expert system, to estimate duration for masonry construction projects, called MASON (Hendrickson, Martinelli, & Rehak, Hierarchical Rule-Based Activity Duration Estimation, 1987). In 1987, Hendrickson et al. (1987) described their "prototypical knowledge-intensive expert system" named as CONSTRUCTION PLANEX, written on top of PLANEX, which can perform construction planning (Zozaya-Gorostiza, Hendrickson, & Rehak, 1989). They focused on construction planning tasks that developed project activity networks, cost estimating, and scheduling (Hendrickson, Zozaya-Gorostiza, Rehak, Baracco-Miller, & Lim, 1986). They used the proposed method to schedule a modular structural system of a high-rise building including activities such as excavation and foundation (Zozaya-Gorostiza C. , Hendrickson, Rehak, & Lim, 1988). They also developed a software package named "Economic Optimization Module (EOM)" particularly aiming at minimizing the total cost of a concrete pour activity considering time-delay fines and material cost (Phelan, Radjy, Haas, & and Hendrickson, 1990). Also they presented their prototype system, IBDE (Integrated Building Design Environment), to explore the communication and integration related issues in the construction industry. The addressed issues were data organization, implementation, intercommunication, knowledge representation, and control (Fenves, Flemming, Hendrickson, Maher, & Schmitt, 1990).

Levitt et al., attempted the use of AI for construction planning in 1988. They pointed out the limitations of the planning tools and demonstrated the strength of AI for scheduling construction projects in their first step (Levitt, Kartam, & Kunz, 1988). Then, they introduced their "System for Interactive Planning and Execution (SIPE)" that was able to do generate correct activity network for multistory office building projects (Kartam & Levitt, Intelligent Planning of Construction Projects, 1990). The extension to the software (SIPE-2), was also able to develop hierarchical schedules for building a single-family house (Kartam, Levitt, & Wilkins, Extending Artificial Intelligence Techniques for Hierarchical Planning, 1991).

In 1990, Mohan listed 37 different expert system tools that were developed, focusing at construction and management field of research and predicted the construction industry will use expert systems more in next few years (Mohan, 1990). Moslehi and Nicholas (1990) described their work as an integrated hybrid expert system that was produced using a relational database, traditional network analyzing software and an interface written in FORTRAN programming language. Their system was able to consider different productivity levels based on labor reassignment, site congestion, learning curve, and overtime. Shaked and Warszawski (1992) presented their CONSCHED system that was able to work quantity estimation, activity generation, activity time and resource allocation, and schedule determination. Then, they extended their knowledge-based expert system to take an object-oriented model of a high-rise building along with the production functions, rules, and routines for developing construction schedule. Then, they used algorithms to optimize resource allocation for managerial efficiency, least cost, or shortest duration (Shaked & Warszawski, Knowledge-Based System for Construction Planning of High-Rise Buildings, 1995). Wang (2001) developed an expert system with knowledge-based programming technique, called ESSCAD, specifically for construction scheduling using information in CAD drawings. The outcome of the system was a primary construction project schedule and as a test, construction schedule of a reinforced concrete frame structure was generated from its AutoCAD drawings.

Neural Networks

Definitions

Artificial Neural Networks (ANN), as computational models, are initially inspired by the brain of animals that are able to do pattern recognition using the "all-or-none" (similar to mathematical binary language, 0 and 1) rule of the nerves.

McCulloch and Pitts (1943) were stimulated by "all-or-none" characteristics of the nervous functions and generated the first computational model defining neural networks using algorithms and mathematics. Then, Hebb (1949) described a neural based learning theory known today as "Hebbian theory" or "Hebb's rule". In 1954, Farley and Clark (1954) simulated a Hebbian network using so called calculators as a computational machines in that time.

Research Works

Sabuncuoglu (1998) showed in his extensive literature review that although using ANN has been a tool for diverse scheduling problems (e.g. job-shop scheduling, single machine scheduling, timetable schduling, etc.), it was not used for construction sequencing and scheduling.

Adeli and Karim started their work on using ANN in construction field of research. They introduced their mathematical formulation of construction scheduling and used their own developed ANN to optimize construction cost and ultimately cost-duration trade-off by varying the project duration (Adeli & Karim, 1997). Then, they extended their work and developed an object-oriented model (Karim & Adeli, OO Information Model for Construction Project Management, 1999). Later, they implemented their work as a software named CONSCOM aiming to solve construction scheduling, change order management and cost optimization problems (Karim & Adeli, CONSCOM: An OO Construction Scheduling and Change Management System, 1999).

Hashemi Golpayegani (2007) desinged an ANN framework that could generate WBS of a given project. The entire solution consisted of five different ANN modules in three main categories as functional WBS, project control WBS, and relational WBS each of which participated in developing the master WBS for their own section. Then, he extended the proposed system to the level that the generated WBS could have simple finish-to-start relations, leading to have a project schdule at the end (Hashemi Golpayegani & Parvaresh, 2011). While he was working on this extenstion, another group of researchers showed their interest in developing WBS of the projects using ANN (Bai, Zhao, Chen, & Chen, 2009). They used four successively arranged Neural Networks rather than Hashemi's parralel structure. In 2008, Rondon et al. (2008) indtruduced a Neural Network desinged to schedule a single machine considering variables such as the operation, deadline time, setup time, processing time, duedate time, etc.

Other Methods

Kim et al. (2012) designed a System Dynamic (SD) model to find optimum program-level scheduling of sustainability projects in a university campus. The SD model was able to rearrange the projects in the given program to come up with the better sequencing of the projects in regard to saving more money at a given time. Damnjanovic et al. (2013) developed a model of predicting project future and its milestones using prediction market with Hanson calculation method. By the use of that proposed tool, a project manager and his/her team can have better insight of the project, helping them in wiser rescheduling of the project plan.

Conclusion and Suggestions

This chapter showed how researchers tried to present their solutions for one of the most important problems in the construction and management field. These researchers have tested several ways to tackle the problem some more effective than the others. On the other hand, the recent introduction of BIM in the AEC industry opened a new aspect of integrating project information and data with its 3D model view. This new BIM can cooperate with any scheduling techniques and models that need project data, as a good source of project information to play the role of input for them. For the knowledge-based and case-based approaches, the BIM of the previous projects can be considered as the past knowledge and cases to be retrieved and reused. This chapter described and listed some interests that have been raise to use BIM in the model-based approach section as a rich source of project data. Decompiling the embedded data from the 3D model can produce the relationship network for the project to be used as the basis for GA fitness function.

This enriched source of data of the project elements and members increases a need to revisit some of the previous approaches while having the BIM in mind to find possibly more robust solution for construction scheduling problem. For instance, using the embedded information in BIM can facilitate accessing precise geometry information of all the project elements. Also having a digital 3D model of the entire project helps the scheduler to get the visual understanding of the project as the plastic 3D model was used before the introduction of digital 3D models, and lately with embedded information as BIM.

CHAPTER III

MATHEMATICAL FRAMEWORK FOR SPATIAL RELATIONSHIPS OF THE PROJECT 3D ELEMENTS[†]

Introduction

A new approach of combining and integrating Architecture, Engineering, and Construction (AEC) data has been adopted in the AEC industry during the last several years. This approach embeds all the needed information about a project in the three-dimensional (3D) representation of the project or links those 3D objects to the database of information and is called Building Information Modeling (BIM). Nowadays, the usage of BIM is widely spread throughout all of the AEC industry and is solving problems and serving companies in many ways. BIM is providing faster and more effective processes, better designs, and better production quality (Azhar, 2011). Above all, the most important benefit of a BIM is its precise geometrical representation of the elements of a building in an integrated data environment (CRC Construction Innovation, 2007).

On the other hand, project schedules are key tools for managing the time, cost, and resources in construction projects. The project schedules are highly dependent on the planners' experience and effects the entire project. The AEC industry attempts to have better project schedules that are more cost effective, have a shorter construction

---

[†] This chapter is submitted to "Automation in Construction" as an individual paper and is under review (Faghihi, Reinschmidt, & Kang, Mathematical Framework for Spatial Relationships of the Project 3D Elements, 2014).

duration, and are more practical and constructible in the real world job-site situations. This chapter describes a new method for developing a construction project schedule by generating a matrix of relationships using the spatial information of the project and its elements.

The BIM of the projects are known as a great source of well-established and well-defined geometrical information of the project components. Knowing the inherent information embedded in BIM project models, the BIM database can be very useful in the process of generating project schedules. Having this concept in mind, by the help of computer programs the built sequence of the entire project can be derived from its 3D representation, which has most of the project characteristics and attributes. To generate the project schedule, the relationships and dependencies of elements in the construction project must be known to completely understand the "structural stability relations" of the project elements. Satisfying these element relationships requirements can guarantee the structural stability of the entire project in every step of its construction process. These dependencies and relationships between elements can be detected by understanding the spatial and geometrical information of all the elements existing in the BIM database. These element geometrical information are defined in IFC format and can be analyzed to find the stability relations. Calculated information on BIM elements' stability, can be decoded from the 3D model in a way that can be understood mathematically by other application programs or users.

In this chapter, the author has represented a mathematical stability representation of a project BIM in the form of a matrix (as a Design Structure Matrix). This type of matrix is selected for this research because the Design Structure Matrix (DSM) is a logical matrix, consisting of 1 (true) and 0 (false), and due to this characteristic it is very computer-friendly for calculation purposes. The DSM is defined and structured to show and define the relationships between components as its rows and columns. The above matrix makes it much easier for computer-based programs to read through it and understand the relationships between rows and columns. This matrix will be the basis for the tool to generate construction project sequences later on using the Genetic Algorithm that basically is based on a string of zeros and ones for its genomes. The concept of generating a construction operation relation matrix was initially inspired by the framework of DSM to show the relationship between elements of any kind. Thus, the proposed algorithm of this research takes the mentioned advantage of this well-established DSM. The algorithm uses the DSM structure to develop a matrix defining constructability constraints between each two 3D elements existing in the construction project 3D model.

The “structural stability relation” is defined as the rational relationship between two elements that results in the combination of the two being stable and ultimately constructible. By stable and constructible, the author means the status of the constructed combination of the elements in which the elements are not intended to fall apart during or after the completion of the construction process due to the lack of sufficient supporting elements. A very simple example is that a beam can be installed only if the both end supports (columns or other beams) are constructed earlier, or in the cantilever beam case it can be supported only from one end. The relationship between this beam and its supporting elements is called a “structural stability relation” or “constructability constraint” herein. Also in this chapter, the matrix that holds all of this information regarding the “constructability constraints” of the elements is named “Matrix of Constructability Constraints” or MoCC. The value of the respective cell in the MoCC would be 1 if an element is constrained by another and 0 if it is not. More description on how to generate the MoCC will come later in this chapter.

The “Model Use” section of this chapter shows how the project scheduling problem can be solved by an optimization tool using the proposed MoCC. The chosen optimization method to reach a well-optimized project schedule is the Genetic Algorithm, considering matrices representing stability and project schedule. This optimization method has been proven useful in the literature for optimizing the construction project schedules having different objectives. These objectives are listed as resource leveling, resource allocation, schedule duration, and construction cost (Jaśkowski & Sobotka, 2006; Toklu, 2002; Wall, 1996), which are described more in the “Model Use” section. The proposed matrix of stability relations, based on its mentioned characteristics, is well-fitted as a metric in the GA fitness function. This function rates and scores the tentative construction schedules toward their constructability and stability status. The more constructible and structurally stable a project schedule is, the higher its constructability score will be.

Definitions

In this section, the basic definitions of networks and matrices are presented. These definitions will lead to the introduction of the proposed matrix in this chapter and will show how a matrix can be used for different purposes.

In construction project management, Program Evaluation and Review Technique (PERT) and Critical Path Method (CPM) networks are the most common ways of showing the dependencies or relationships between tasks in *project networks*. The project network itself is a type of *dependency network* customized for project management. The dependency network is a *graph*-based representation of an *adjacency matrix* or specifically as a *design structure matrix*, which is used more in engineering and project management fields. The detailed definitions of the highlighted phrases are as follows.

Project Network (Network Chart)

A project network is a flow chart (graph) presenting the completion order or sequence of a project's elements by showing project elements and their dependencies. One of the main differences between typical dependency networks and project networks is that project networks will almost never contain loops (See Figure 1). In commonly used networks such as PERT and CPM, loops are not allowed as a constraint of the critical path solution algorithm; if there could be loop(s) in those networks, the critical path would have an infinite duration. However, loops may be used in other types of solution algorithms (i.e. Graphical Evaluation and Review Technique - GERT).

There are three kinds of dependencies commonly considered in construction projects:

- Causal (logical), an example could be:
    - It is illogical to do foundation excavation after pouring its concrete.
- Resource constraints, as an example:
    - It is logically feasible to paint all walls of a room concurrently, but there is only one painter.
- Discretionary (preferential), such as:
    - One may want to paint the dining room before the living room, although it could be done the other way around, too.

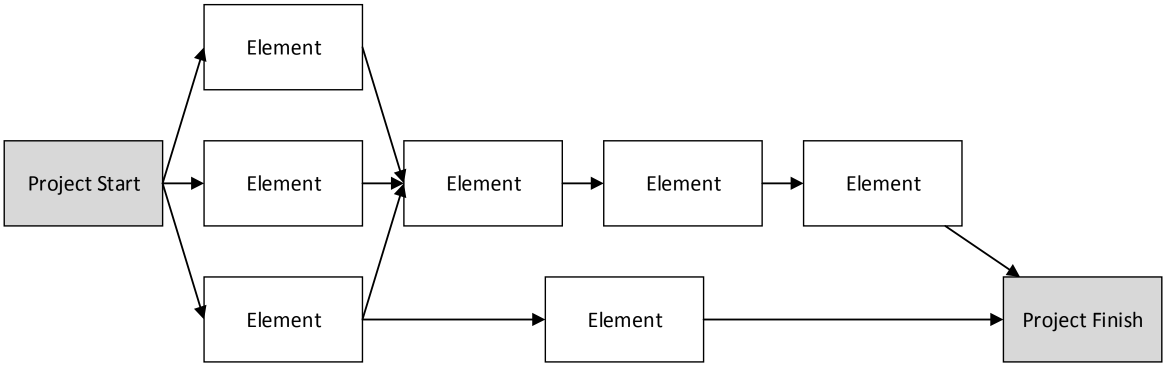


Figure 1- Sample Project Network with 9 Nodes and 11 Relations

Graph

A "graph", as a visual representation of the matrices, is a set of items (called nodes or vertices) connected by edges. In other words, a graph is a set of nodes and a binary relation between nodes, adjacency (Black & Tanenbaum, Graph, 2012). An undirected graph has edges that are unordered pairs of nodes and each edge connects two nodes (Black, Undirected Graph, 2007), or a directed graph with edges that are ordered pairs of nodes and each edge can be followed from one node to another (Black, Directed Graph, 2008). See Figure 2 for a directed graph example.

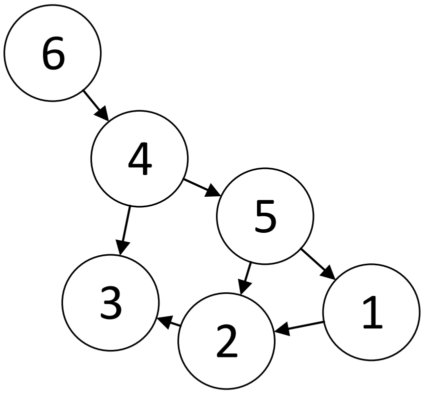


Figure 2- A Drawing of a Directed Graph

Dependency Network

One type of graph that is used in engineering and management is called the Dependency Network. The dependency network provides a system level of activity analysis and directed network topology. By analyzing the network structure, the dependency network approach extracts underlying topological relations between the nodes of the network. Although this methodology was originally introduced for the study of financial data (Kenett, et al., 2010), it has been extended and applied to much broader domain, such as the immune system (Madi, et al., 2011). A sample dependency network is shown in Figure 3.

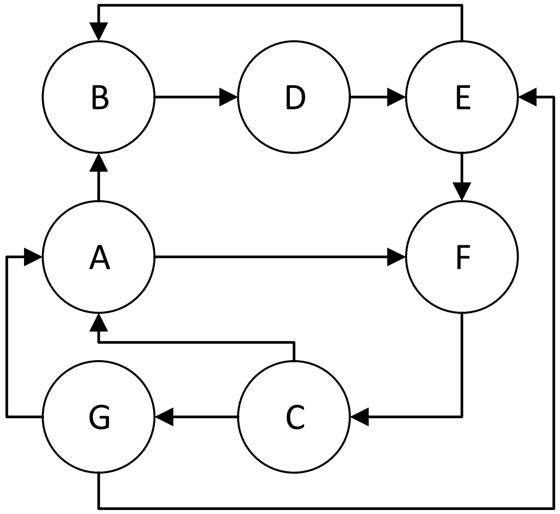
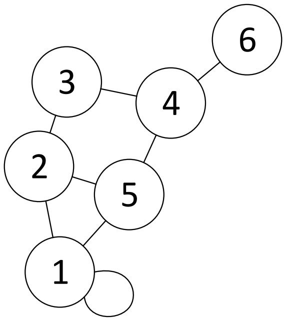


Figure 3- Dependency Network Example

Adjacency Matrix

In computer science and mathematics, an adjacency matrix is a tool for representing which nodes of a graph are related (adjacent) to which other nodes in a compact and abstract way. The adjacency matrix is directed when the adjacency of two nodes is only one way. This means when node A is related to node B, the reverse relation is not true. The adjacency matrix counts as undirected when the vertex between two nodes means bidirectional adjacency (See Figure 4 for an example of undirected adjacency matrix and its graph representation). A directed graph with n nodes using an

$n \times n$ matrix can be represented as follows: the value at (i,j) is 1 if there is a connection (edge) from node i to node j; otherwise the value is 0 (Shukla, 2009).



↔

$$\begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 2 & 1 & 0 & 1 & 0 & 1 & 0 \\ 3 & 0 & 1 & 0 & 1 & 0 & 0 \\ 4 & 0 & 0 & 1 & 0 & 1 & 1 \\ 5 & 1 & 1 & 0 & 1 & 0 & 0 \\ 6 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}$$

(a) (b)

Figure 4- Labeled Graph (a) and Its Equivalent Adjacency Matrix (b)

Design Structure Matrix

Design structure matrix (DSM) is a flexible and straightforward modeling method that enables users to analyze and show system elements dependencies for functionality improvement. It can also be used for designing, developing, and managing complex systems (Kuqi, Eveleigh, Holzer, & Sarkani, 2012). DSM is especially well suited for applications in the development of complex engineering systems and has mainly been used in the engineering management area (Eppinger & Browning, 2012). Donald Steward began using the term “Design Structure Matrix” in 1981 (Lindemann, 2009), but the method has been in use since the 1960s (Steward, 1962).

A sample DSM is shown as Figure 5, where the off-diagonal mark in cell BD indicates that design activity D must be completed on or before design activity B is executed (logical relation of B precedes D). The complete logical relations defined in this example matrix are plotted in Figure 3 as a dependency network.

	A	B	C	D	E	F	G
Element A	A	1				1	
Element B		B		1			1
Element C	1		C				
Element D				D	1		
Element E		1			E	1	
Element F			1			F	
Element G	1				1		G

Figure 5- A sample DSM with 7 Elements and 11 Dependency Marks.

Methodology

The first step toward generating a project network from the 3D model is to detect and list all the elements that the given 3D BIM contains. This detection of the 3D elements of the model will be performed by reading the data embedded in the standard IFC (Industry Foundation Classes) file format. Having the list of all elements, the algorithm reads the geometry information for each of those elements and calculates their spatial information. Then, the MoCC can be generated using the set of predefined stability rules. This newly generated matrix, as mentioned earlier, has the main characteristics of DSM, therefore it represents the mathematical or matrix-based form of the project network by itself. To enrich this project network further by including timed sequencing of construction operations and installations, an optimization tool for solving the scheduling problem is used.

Industry Foundation Classes (IFC)

Industry Foundation Classes (IFC), as an open and neutral specification, is an object-based file format. The data model of this neutral file format is developed by buildingSMART and the main goal for its development is to facilitate interoperability between the AEC companies, as a commonly used file format for BIM (buildingSMART, 2013). The IFC model specification is listed as an official International Standard ISO 16739:2013 (International Organization for Standardization, 2013).

Detecting Project Elements

The spatial and geometric data of an element from the IFC file can be obtained either using available IFC reader DLLs (Dynamic Link Library, Microsoft's method for the shared library implementation in the Microsoft® Windows® operating systems) or even a well-established text reader since IFC files are text based. Here, the author has developed his own web-based IFC reader and extracted the required geometry information from the IFC files. This data was then used for the next step, which is detecting the project element dependencies and their stability requirements.

Retrieving Spatial Data of IFC Elements

The geometry information of the elements in the IFC file format is stored based on a standard published by buildingSMART (buildingSMART, 2013). Although the latest version of the IFC is now IFC4, the version used in this methodology is IFC2x3

TC1 (Online documentation available in: buildingSMART, 2011). The two main entities from the IFC file format that are used in this part of the research are IfcColumn and IfcBeam, representing the column and beam elements respectively. These two 3D elements are basically an extrusion of a profile in a direction, which all are defined precisely in their attributes. For instance, as shown in Figure 6, the first line (#143) is the line for defining a column in the IFC file format and the second line (#375) is the same thing for a beam. These two IFC lines represent two structural elements of the 3D model shown in Figure 8.

$$\#143$$
$$= IFCCOLUMN('2ed437SVDCI8RyHuUeaHBs', \#42,' W - Wide\ Flange$$
$$- Column{:}\, W10X49{:}\, W10X49{:}\, 161188', \$, 'W10X49', \#142, \#139, '161188');$$
$$\#375 = IFCBEAM('2ed437SVDCI8RyHuUeaH5r', \#42, 'W$$
$$- Wide\ Flange{:}\, W12X26{:}\, W12X26{:}\, 161319', \$, 'W$$
$$- Wide\ Flange{:}\, W12X26{:}\, 116116', \#297, \#374, '161319');$$

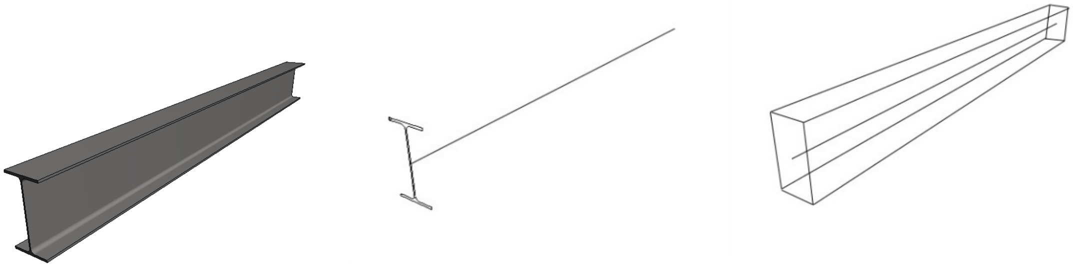
Figure 6- Sample IFC Definitions for Beams and Columns

Using the IFC standard (buildingSMART, 2011), the approach to decode and use each of these fields is described. Having this standard in mind, the author simplified the beam and column definitions by considering their profile shapes as rectangles. Based on the extrusion direction and the global starting point XYZ data from the IFC file, the end point of the element is calculated. This set of information lets the algorithm understand the geometric shape and position of the detected element.

Since the bounding box helps define the joining and cutback of structural framing elements (Autodesk, 2013), the author used this concept to find the connecting elements. In this proposed method, as an initial step, all the existing elements of the 3D model (as mentioned earlier only beams and columns are considered in this chapter) are detected, mathematically understood, and stored in a database. Then, for each individual element all the connecting and related elements should be retrieved from that database. Although the connecting elements data (IfcRelConnectsElements entity subtype of IfcRelConnects and IfcRelationship) is an inherent feature of the IFC file format, since the software vendors do not publish this type of information to the exported IFC file, this valuable data is lacking. To retrieve this information mathematically from the calculated geometric data, an offset distance will be considered to find the connecting elements.

To correctly calculate the connections between elements in the model (beams and columns in this case), different values for the offset have been set for different connections existing in the model (column-to-column, beam-to-beam, and column-to-beam). This offset defined a virtual boundary from the center line of the extrusion line of the element profile that creates a cube, creating a slightly bigger volume that the actual element in size. For instance, 12-inch offset is set for the 3D model in this research (shown in Figure 10) for W-Wide Flange 10X49 column elements. Figure 7 shows how the boundary box is calculated for a beam. In this figure, (a) shows the exact

3D element, (b) is the extrusion line with the profile and (c) shows the extraction line in addition to the calculated boundary box.



(a) (b) (c)

Figure 7- 3D Element Simplification

Each of the detected elements that have intersection in their offset boundary (crossing each other) are considered connected. Thus, they are dependent and related to each other for construction purposes based on the predefined structural stability rules described in the next section.

Predefined Structural Stability Rules

There are several predefined and rational rules for decoding the inherent relationships between the elements in a 3D model into a textual or mathematical form. In construction projects, there are inevitable rules regarding construction of the projects. These rules are typically for satisfying the stability of the construction processes. Below is the list of those constructability and stability common knowledge rules that are considered in this research:

- Upper level columns should be installed after the lower level columns

- Beams should be installed after supports at both ends
    - Cantilever beams will have one-end support
- Walls should be installed after adjacent columns of the same level and adjacent same-level beams and the lower level beams (all four sides)
- Doors should be installed after the walls that are including them
- Windows should be installed after the walls that are including them
- Slabs should be installed after all the beams in the slab region from the lower level
- Roofs should be installed after all the beams of the lower level

The author could not find any example violating any of the above-mentioned facts in ordinary construction schedules. Obviously, these rules do not completely cover all construction activities and relationship requirements. The set of constructability rules can be extended by the users later, based on their needs. In addition, the "before and after" relationships stated in the above rules are considered to be the minimum possible interval duration between two activities. By defining and knowing these rules, a 3D model of a project can be decoded and the Matrix of Constructability Constraints (MoCC), which holds all the parent-child relations between elements, can be generated.

The MoCC is only based on the spatial or location information of the object, its structural type, and the above mentioned common knowledge rules. Therefore, if a 3D model is built logically (obeying the logical placements of the building components), the resulting MoCC will be logical. In case an incorrect building element placement in the 3D model exists, the MoCC still will be generated completely; however, it will not represent a logical project network of element dependencies. If this incorrect MoCC is used later on for construction schedule development the algorithm will produce 100% constructible schedule, but since its metric for calculating the score was an incorrect one, the schedules will not be constructible in the real world.

Creating Directed DSM

In the MoCC (shown in Equation 1), A_i represents elements as well as activities associated to each element indicating installation of that activity. Values of $s_{i,j}$ could be either one or zero, indicating immediate prerequisite installations or no relationship respectively. For instance $s_{2,6}=1$ means that 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.

Equation 1- Matrix of Constructability Constraints

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

Where:

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

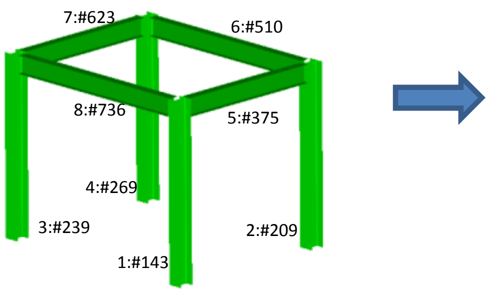
$S_{j,i}$: dependencies between elements (that could be either 0 or 1 showing not dependent or dependent respectively)

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.

Generating Project Network

Based on previous descriptions of graphs and matrices and by considering the characteristics of the MoCC, this proposed MoCC is a Directed Adjacency Matrix or basically a directed DSM. This Matrix of Constructability Constraints (MoCC), similar to all DSMs, represents a relation network that in this research is the project network of the given 3D model. The conversion from 3D BIM to the MoCC and then to the project network is shown in Figure 8. In this figure, the MoCC is generated based on the spatial relations of the geometry elements in the 3D model considering all the previously mentioned stability rules or common knowledge. By having this matrix, which is a numerical representation of the structural stability relations of a BIM, the next step can be generating a constructible and stable project schedule for that specific 3D model.

The Matrix of Constructability Constraints for the Figure 8 is generated as the following description. The construction of elements number 1 to 4, which are all columns, is not constrained by any of the model elements. This means the erection operation of them is not dependent on installation of any other model elements. Therefore, all the values of the first four rows of the MoCC (i.e. 1:#143, 2:#209, 3:#239, and 4:#269) are zeros, indicating independence of the four columns. Element number 3, which is beam #375, is supported by two end columns labeled as 1:#143 and 2:#209. This dependency for structural stability of the beam 5:#375 is shown as two values of one for the matrix column 1 (3D column 1:#143) and matrix column 2 (3D column 2:#209). With similar logic, the structural relation of the other beams to columns are detected through their spatial information embedded in IFC file of the 3D model and the values of 1 (true) are added to the corresponding cell in the Matrix of Constructability Constraints.



$$MoCC = \begin{array}{r} \\ 1:\#143 \\ 2:\#209 \\ 3:\#239 \\ 4:\#269 \\ 5:\#375 \\ 6:\#510 \\ 7:\#623 \\ 8:\#736 \end{array} \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

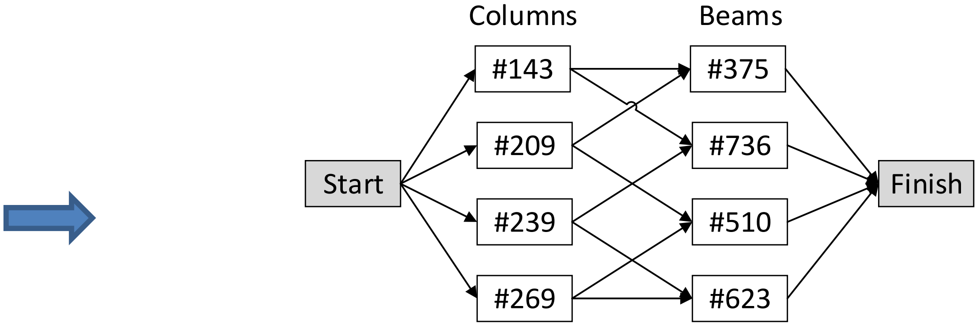


Figure 8- BIM to Project Network

Model Use

At this point, all the constructability constraints and structural stability requirements are retrieved from the 3D model of the project and are embedded in a matrix (MoCC) as the project network. Now, the problem of generating construction schedules can be solved as an optimization problem. The goal or objective of the optimization is sequencing installments of the elements in a way that the constructability or structural stability of the entire project is maximized. This means the ultimate outcome of this optimization should be a set of completely constructible and structurally stable construction sequences for the given 3D model of the project. Maximizing the overall feasibility of the project construction (or in other words, minimizing the infeasibility of the construction) has always an upper bound limit of 100%. The 100% score means that the entire project (100% of its elements) is scheduled to be installed in a structurally stable sequence.

To accomplish that outcome, one of the methods that has already been proven to be useful in management as an optimization tool (described in Chapter II) is the GA. Using this optimization tool, it is possible to maximize or minimize the predefined objectives. For instance, by maximizing the constructability objective the resulting project schedules tend to be more and more constructible and stable. Also minimizing the project duration, cost, or other objectives will result in more desirable project schedules. In addition, since the proposed matrix consists of 0s and 1s, representing stability relations, it is well suited for the GA genome representation. This binary alphabet {0,1} is typically used to represent the genes of the binary GA genomes (Haupt & Haupt, 2004).

In Chapter II, the author mentioned the researchers who were focusing on project schedules and GA for optimization. It was shown that how they have demonstrated the usefulness of the GA for various forms of construction schedule optimization. Although they were mainly focusing on resource leveling and assignment, this tool can be used here to produce the construction schedule from scratch. To do so, a well-established fitness function is required. This requirement can be satisfied by using the derived stability matrix from the BIM of the project. The defined objective in this optimization is the constructability of the project that means the GA would try to find constructible project schedules for the given project 3D model. The constructability of a project schedule herein is defined as having all the elements stable during and after the installation process. For instance, installation of a beam is considered constructible and stable only if the two end structural supports (columns or beams) have been installed before.

All in One

The entire methodology process that has been described above is summarized in the following schematic view (Figure 9).

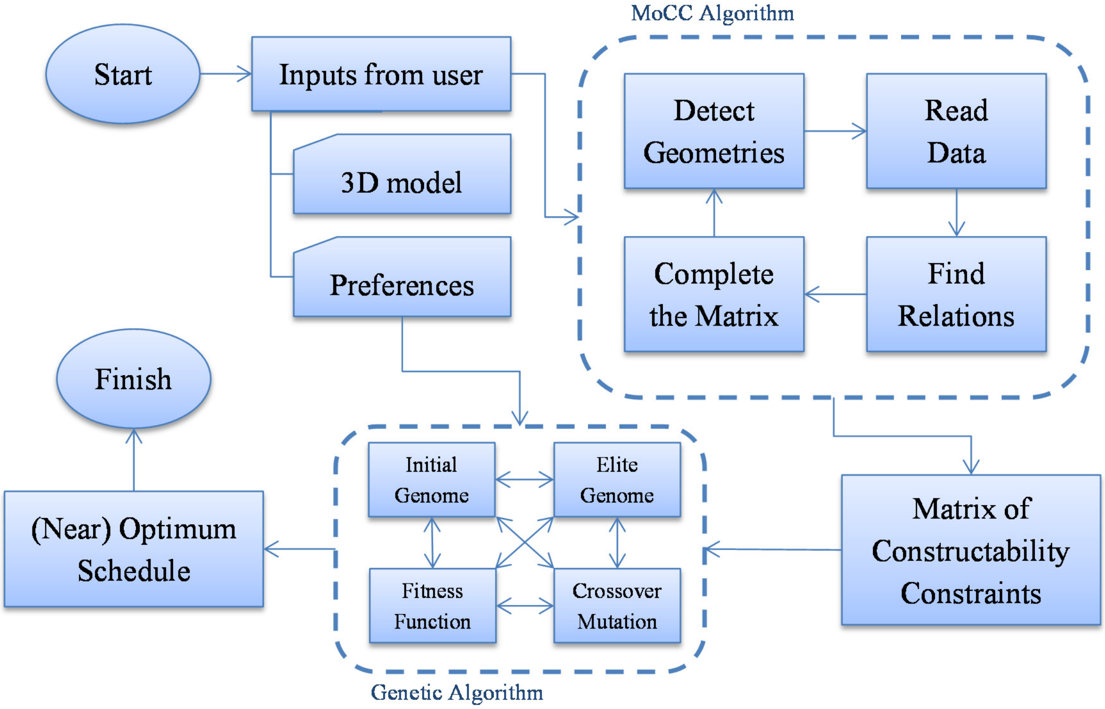


Figure 9- Schematic View of the Methodology

Testing the Model Usage

To take advantage of the proposed methodology, a software application package has been developed based on the described characteristics. This software package can receive project BIM as well as other user inputs such as resource limitations. The MoCC algorithm of the software will then calculate the MoCC from the 3D BIM and export it for later use in the fitness function of the GA engine. Random construction schedules would be fed into the GA as its first population. Then, by considering MoCC as the metric in the fitness function and after hundreds of cycles of evolutions, several different constructible project schedules for the given BIM will be generated.

To test this research method, a real-size generic 3D model of the steel structure from a turbine machine building, as shown in Figure 10, is created and used as the input for the proposed algorithm. This model consists of 274 structural elements, of which 172 are beams and the rest are columns.

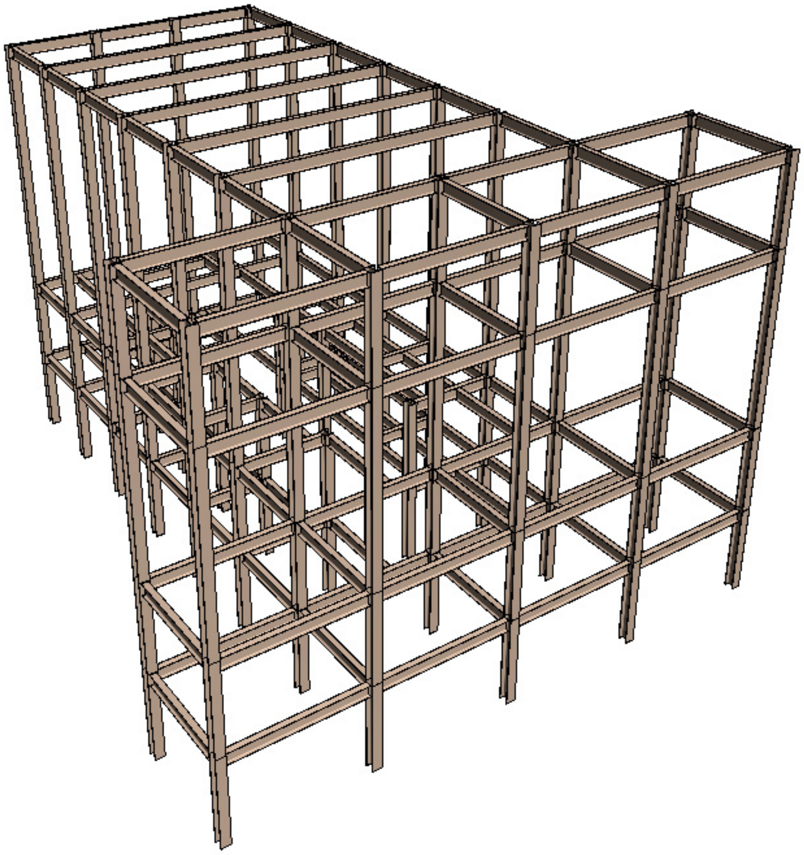


Figure 10- Steel Structure BIM of a Turbine Machine Building

The algorithm explained herein could detect all the stability and constructability requirements and constraints of the 3D model. Then, it successfully and properly generated the MoCC for the given relatively complex 3D model. In the next step, this calculated MoCC was used as the core for the fitness function of the GA, which was given a randomly generated project schedule as its initial population. Then, the GA, using its inherent functions such as fitness function, mutation, crossover, and selection, tried to maximize the constructability and structural stability of the project schedules for the 3D model, using the MoCC as the controlling criterion for stability. Finally and after several hundreds or thousands of iterations of calculations through GA populations, multiple complete construction project schedules were generated that were 100% stable and constructible. These generated construction schedules can also be optimized toward other objectives such as shorter duration or less cost later on using the same tool with different and more advanced fitness function.

Conclusion and Future Work

This chapter showed how the inherent spatial information in Building Information Models could be used to generate the valid construction sequences of the project. This new way of mathematical representation of the stability model of a project as matrix of 1s and 0s, is shown to be helpful for developing project network and then construction sequence of the project. For the future studies, a larger number of examples will be generated and tested to show how this methodology can be used in education and project control. These different examples will include: different 3D models with different complexity (number of elements and connectivity types), more different 3D element types, diverse project durations, different GA parameters, etc. In addition, defining more objectives such as shorter duration and/or less associated cost of the project could be added to the fitness function of the optimization tool. By adding these new objectives, the construction schedules generated by this methodology would be more feasible to be accepted for the real projects in the education of construction managers and for construction industry.

As another outcome from the future extensions of this research, by integrating BIM with project scheduling, students could learn through hands-on interaction with the system, how to generate more effective project networks and schedules. The use of the interactive 3D interface provides a superior learning environment for graduate students (or even undergraduates) in Construction Management. They can practice construction scheduling in a classroom situation in which they are provided with immediate feedback on their trial solutions for construction schedules. Also, project schedulers and construction managers can take advantage of using the outcome of this research for developing initial project schedule templates for the specific project they have. They can change the input parameters to the algorithm (such as available resource, work policies, and so on) to manipulate the generated project schedules and work more precisely on those automatically generated schedules to come up with a rich one.

In addition, application developers can use this mathematical model of the building for better understanding of the project elements relationships and dependencies. Considering this helpful feature of this mathematical model and as an example, an enhancement can be implemented to the automatic project progress detection from photos taken on the job-site. Recently Golparvar-Fard has shown methods for detecting progress of the construction projects by taking still images of the construction job-site

(Golparvar-Fard M. , Peña-Mora, Arboleda, & Lee, 2009; Golparvar-Fard, Peña-Mora, & Savarese, 2009). The proposed mathematical model of elements' stability relations can help this automation progress detection to check the element dependencies and hopefully reduce the time and increase the accuracy of the calculations. For instance, by using the MoCC matrix defined in this chapter, Golparvar-Fard's method can understand which elements should have been installed prior to the detected installed element from the photos, even if those elements are covered and not visible in the photos taken.

CHAPTER IV

CONSTRUCTION SCHEDULING USING GENETIC ALGORITHM BASED ON BIM[‡]

Introduction

The development of project schedules is a critical part of all types of projects including engineering, manufacturing, construction, and others. However, engineering education, whether at the graduate or undergraduate level, typically provides little instruction on how to develop good construction or fabrication schedules. Construction engineers and managers on projects learn on the job how to visualize the sequence of activities that will lead to good, feasible schedules, without formal training. By integrating project scheduling with virtual three-dimensional geometric modeling, students could learn through hands-on interaction with the system how to generate more effective project networks and schedules.

The main purpose of this research is to create an environment for construction planners to have a visually interactive communication between the planning process and 3D models of the project at each increment of time. This environment would use an algorithm that simulates the natural evolutionary process in a rule-based approach to reach a feasible project schedule. The natural evolutionary process in this research considers the relationships and dependencies of the project elements from the Matrix of

---

[‡] This chapter is submitted to "Expert Systems with Applications" as an individual paper and is under review (Faghihi, Reinschmidt, & Kang, Construction Scheduling Using Genetic Algorithm Based on BIM, 2014).

Constructability Constraints (MoCC), presented in Chapter III and shown in Equation 1, and uses previous knowledge gained through experience from similar works. The determination and calculation of the relationships and dependencies of the project elements would be handled through a well-developed mathematical algorithm reading all the geometric information on the project elements from the 3D project file. The necessary common knowledge and previous experiences would be helping the initial phase of developing the algorithm by defining sets of rules to express element dependences. Using this algorithm, a project planner can change the work strategy using predefined parameters and the 3D model, as a visual representation of the entire project, to see the effects of different strategies on the schedule.

The proposed algorithm can potentially extend to have a two-way interactive environment between project BIM and its schedule. This environment can bring new dimensions to the management team of the project in which changing the design of the 3D model directly and immediately results in a new and updated project schedule. Also, when the project schedule is manipulated, the extended algorithm can detect which parts or elements of the project may not be constructible regarding the updates in the project schedule by highlighting them.

Considering Graph Theory and definitions of the matrices, the MoCC is a Directed Adjacency Matrix or a Directed Design Structure Matrix, representing the project network (Kanda, 2011). The conversion from 3D BIM to the MoCC and then to the project network is shown in Chapter III as Figure 8. In the mentioned figure, the matrix in the middle, MoCC, is generated based on the spatial relations of the geometry elements in the 3D model considering the previously mentioned common knowledge of stability. By having this matrix, as a numerical representation of the stability relations in a BIM, the next step will be generating a constructible and stable project schedule for that specific 3D model. To reach a set of fully stable project schedules, the stability score can be assigned to each possible solution. This stability score will be the percentage of the project elements that are scheduled for installation in a stable order (i.e. obeying the constructability constrains calculated in MoCC). Then, the target score will be 100% of the stability that should be reached. This approach brings the environment of optimization methods where the stability score should be maximized. To accomplish that outcome, one of the best methods is the Genetic Algorithm (GA), which has already been proven to be useful in project management as an optimization tool. In addition, since the proposed matrix consists of zeros and ones, representing stability relations, it is very well suited for the GA fitness function.

Defining GA Functions

Below are the general descriptions of core Genetic Algorithm functions and their definitions in this research.

Genome Creation

In this approach the genomes consist of lists of elements to be installed in each time-unit (e.g. day, week, or month) throughout the total project duration. By this definition, a genome can be shown in either of the following two ways. The Matrix of Genome (MoG), as shown in Equation 2, consists of *n* rows, each of which represents a single element from the 3D model, and *k* columns, indicating total installation duration.

The non-zero value of $g_{i,j}$ shows the installation time j, for the element, i. For example, if $g_{5,3}=1$, it means that element number 5 is scheduled to be installed in the third time-unit (which could be either hour, day, or week based on user definition). If all the rows of this matrix are put in a single row in a way that the first column of a row gets placed after the last column of the previous row, then a single string of matrix values is generated (as shown in Equation 3) and is ready to be used in a GA population.

Equation 2- Matrix of Genome (MoG)

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

OR

Equation 3- Genome

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

Where:

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)

The length of the genome or the number of genes, $g_{i,j}$, in the genome is calculated as follows:

Equation 4- Length of the Genome

$$Length\ of\ the\ genome = number\ of\ genes = n \times k$$

In a random genome generation, the total project duration would be chosen based on initial data from the user. Then, a string of zeros and ones is generated, with the length calculated by multiplying the total project duration and the number of elements (tasks) retrieved from the 3D BIM file. Therefore the only limitation for this random genome would be as shown in Equation 5:

Equation 5- Genome Creation Logical Requirements

$$g_{i,j} = 1\ \&\ g_{i,j+1} = 0 \rightarrow \forall\, m > j + 1 : g_{i,m} = 0$$

The above condition simply means that if an element has been installed before, it cannot be reinstalled. The easiest way to create this genome is to spread out *n* number of ones in string of $n \times k$ zeros, where *n* is the number of elements and *k* is the total number of time-units. With this condition, an element could be installed in one or more time-units if and only if all the time-units are sequential. To simplify the genome more, each element would be installed in only one time-unit and no more. Thorugh this simplification, in each row of the MoG there would only be a single 1 and all other values would be 0.

The algorithm is programmed in a way that it schedules the model elements to be installed using cumulative normal distribution, simulating the S-curve work load in real project completion phase.

Elite Members

In the GA, it is desirable that the fitness function score does not decrease from one population to the next when maximizing the objective. Thus, when generating a new population, some of the better genomes are allowed to move from the current generation to the next generation, unchanged. This method is known as elitist selection and those selected genomes are called elite members of the old population.

As an example, looking at Table 9 and Table 10 from the Appendix, it is noticeable that the first genome, which has the highest score, is moved to the next generation intact. In that example, the elite rate is set to 20%, one genome out of entire 5 member population.

Fitness Function

The fitness function for the GA could have multiple variables to measure and in this case it is considered as a multi-objective GA. However, in this step of the research only one objective is defined: constructability of the project sequences.

The constructability objective is the most important objective for the construction sequences and should be closely tracked and measured. Above all, a project schedule should be completely constructible covering all the project components scheduled to be installed. The constructability score, in the form of a percentage, is calculated based on the number of the elements that are obeying the constraints defined in MoCC, divided by total number of the elements, as described below.

To determine the constructability score, the Matrix of Constructability Constraints (MoCC) is developed as the key factor, which has been shown in Equation 1. In this matrix, all the rules and constraints related to the specific 3D BIM are defined element-by-element using the rationales behind the geometry and their dependencies to each other. Having the MoG, mentioned in Equation 2, a function can easily be defined to read the sequence of elements installed from the genome and determines the elements that are scheduled to be installed in each time-unit. Then, for each of those elements, all the constructability constraints (prerequisite elements) would be retrieved from MoCC. Then again, each element from this list of constraints would be checked against the MoG to see if it has been scheduled for installation before or not. In case all the prerequisite elements (constructability constraints) are satisfied, the element would be considered as constructible.

By dividing the total number of constructible elements by the total number of elements and multiplying by 100, the constructability percentage of the genome is calculated.

Selection Method

Selection functions that are traditionally used in GA are categorized in the following three groups: (Sivaraj & Ravichandran, 2011)

- **Proportionate Selection:** in this method (better described as Roulette Wheel selection), each genome will get a score assigned, f_i, using the fitness function. Then, the cumulative fitness of the entire population, Pf_i, will be calculated. After that, the probability of selection for each genome is calculated as $psel_i = \frac{f_i}{Pf_i}$. This fitness score is then used to assign the probability of being selected to each individual genome.
- **Ranking Selection:** In linear ranking selection (Baker, 1987), first the individual genomes are ranked based on their fitness values. Those genomes that have higher fitness values will be ranked higher and those with lower fitness values will have lower ranks. Then, the genomes are selected based on a probability that is linearly relative to the rank of the genomes in the population.
- **Tournament selection:** it consists of running several "tournaments" between a few individual genomes selected randomly from the population. The one with the best fitness (the winner of each tournament) is then chosen for crossover.

For this research, the Fitness Proportionate Selection (a.k.a. roulette wheel method) introduced by Holland (1992) is chosen to be the selection function to pick parent genomes for crossover function.

Crossover

In the GA, two genomes are selected as parents from the previous generation, which has just been created, and paired for breeding two new genomes as their children for populating the new generation. As described before, the parent selection would be handled using the specified *Selection Function*. In this section, the process of how to breed two new child genomes from the parent genomes is described.

To do the crossover for this GA, after selection of the two parents, a random duration will be selected from one of the parents and both parents would be split from that random point in time, in equal proportion. Mathematically the cutting point for both parents would be calculated as follows:

Equation 6- Calculating the Parents Cutting Points (Crossover)

$$\{Cutting\ point\ of\ parent\ 1\} = random(1, \{duration\ of\ parent\ 1\} - 1)$$

$$\{Cutting\ point\ of\ parent\ 2\} = integer\left(\frac{\{Cutting\ point\ of\ parent\ 1\}}{\{duration\ of\ parent\ 1\}} \times \{duration\ of\ parent\ 2\}\right)$$

The reason for this proportional cutting point is that, since both parents are schedule representations of a single project, they would naturally have a similar pattern. This way of cutting creates better children because each part of the two parents may carry similar construction schedule information and pairing them up in this way results in more constructible children. The efficiency of this approach has been tested and verified during this research, as the growth in constructability score is sped up by doing the crossover in the mentioned way.

The concern here is that all the elements are supposed to be scheduled for installation in one and only one time-unit. This assumption simply means that there should not be an element such that all the values in its row in the MoG are zero and no element that has multiple values of 1. In other words, there should be no row whose sum is not equal to 1, meaning that every element is installed at one time.

The above conditions can be represented by the following:

Equation 7- Validation Logics

$$\forall i \in \{1,2,\dots,n\} \ \exists\ j \ \{1,2,\dots,k\} \colon g_{i,j} = 1$$

AND

$$\forall i \in \{1,2,\dots,n\} \ \&\ j \ \{1,2,\dots,k\} \ \&\ 0 < c < k \colon g_{i,j} = 1 \ \&\ g_{i,j+c} = g_{i,j-c} = 0$$

OR

$$\forall i \in \{1,2,\dots,n\} : \sum_{j=1}^{k} g_{ij} = 1$$

Where:

g_{ij}: the value of element i in time-unit j in the genome

i: the element number

j: time-unit step number

n: total number of elements

k: total number of time-unit steps

c: *any random number between zero and k*

The following steps show how this crossover function is performed. Here two different genomes have been defined. These two genomes represent two different installation sequences (in this case) for a three-element structure with durations equal to 4 and 3 time-unit respectively for genome 1 and genome 2.

$$Genome\ 1 = \begin{vmatrix} 0 & 0 & 1 & 0 & \vdots & 0 & 1 & 0 & 0 & \vdots & 0 & 1 & 0 & 0 \end{vmatrix}$$

$$Genome\ 2 = \begin{vmatrix} 0 & 0 & 1 & \vdots & 0 & 1 & 0 & \vdots & 1 & 0 & 0 \end{vmatrix}$$

Figure 11- Sample Genomes

As mentioned earlier, these genomes can be shown as the following two matrices shown in Figure 12. The first genome (matrix) schedules no installation on the first time-unit. The first element is sequenced for the third time-unit and the second element for the fourth time-unit and the last element for the second time-unit. In the second genome (or matrix), the third element is scheduled for the first time-unit and the second element for the second time-unit and the first element for the last time-unit.

To perform the crossover function on these two genomes (matrices), as described above, a random time-unit will be selected from the range defined in Equation 6. The duration range from the MoG 1 for random number pick up is 1 to 3 time-unit. Assumption in this example is that the randomly picked number is 3. Therefore the MoG 1 will have a cutting point between its third and fourth time-unit. Using the second part of the same mentioned equation, the cutting point of the MoG 2 will be cutting between the second and the third time-unit. These cutting points are illustrated as red dotted lines in Figure 12.

$$MoG\ 1 = \begin{vmatrix} 0 & 0 & 1 & \vdots & 0 \\ 0 & 0 & 0 & \vdots & 1 \\ 0 & 1 & 0 & \vdots & 0 \end{vmatrix}$$

$$MoG\ 2 = \begin{vmatrix} 0 & 0 & \vdots & 1 \\ 0 & 1 & \vdots & 0 \\ 1 & 0 & \vdots & 0 \end{vmatrix}$$

Figure 12- Sample MoGs, Showing Cutting Points as Red Dotted Lines

The crossover function will attach the first part of the first MoG to the second part of the second one and do the same for the remaining parts to generate to new matrices. The early matrices are called parent matrices and the later ones are the child matrices. The result of the crossover function on these two MoGs considering the calculated cutting points will be as the following.

$$New\ MoG\ 1 = \begin{vmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{vmatrix}$$

$$New\ \ MoG\ 2 = \begin{vmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{vmatrix}$$

Figure 13- Child Matrices after the Crossover Function on Their Parents

As seen in Figure 13, both genomes are invalid regarding the placement schedule of the elements. This invalidity is due to both not scheduling an element for installation (element number 2 in new MoG 1 and element number 1 in new MoG 2) and double installation time for another element (element number 1 in new MoG 1 and element number 2 in new MoG 2). There is another function named "validation function" that is responsible to make necessary changes in genomes to make them valid. This function is described further in this chapter.

Mutation

As a part of the GA, there is a mutation function changing the genomes in some point randomly. The random mutation helps the optimization to avoid being trapped in local minima. To do this mutation, a random gene needs to be picked and inverted its value to mimic the mutation. After doing this mutation, the genome needs to be validated also. The mentioned random mutation could be mathematically shown as below:

Equation 8- Mutation Logics

$$randE = random(1, number\ of\ elements)$$

$$randT = random(1, number\ of\ time - units\ )$$

$$selected\ gene = g_{randE,randT}$$

$$g_{randE,randT} = 0 \rightarrow g_{randE,randT} = 1 \parallel g_{randE,randT} = 1 \rightarrow g_{randE,randT} = 0$$

However, a better mutation function could be described in another way. Each schedule genome has multiple project elements that are not scheduled for installation obeying the MoCC, if and only if the constructability score is less than 100%. To define a better mutation function reaching for the 100% score faster one way is to find out those MoCC violating elements and randomly mutate their installation time to somewhere later than the current time. Since changing the installation time of a violating element to a later time is not violating any predefined rules, genome validation is not required anymore and the calculation speeds up.

A simple example of how the mutation function effects the genomes is shown below. For this purpose, imagine that the matrix shown in Figure 14 is for a very simple structure consisting from two columns supporting a beam. The columns are considered as the first and the second elements defined in the following matrix and the beam is the third one.

$$MoG = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{vmatrix}$$

Figure 14- MoG before Mutation Function

As shown in the Figure 14, one of the columns is scheduled to be installed in the first time-unit and the other in the third. Since the beam is supported by the two columns, as assumed in this example, it should be installed after all the structural supports are installed. Based on the MoG shown in Figure 14, the beam is scheduled to be installed in the second time-unit, before installation of the second column. Therefore, the constructability score of this genome is 66% (two elements out of three are scheduled correctly regarding the structural stability of the model).

In case this genome is selected to be mutated in the GA process, the third element (the beam) will be chosen for mutation. In the first step of the mutation process, the initial installation time value will be set to zero. Then, the function will find the latest installation time of the structurally supporting element, which in this example is the latest time for two columns, calculated as the third time-unit. Therefore, the mutation process determines the range from the next time-unit, 4, and the total duration of the schedule, which is again 4, and will pick a random number from that range. In this simple example the random time-unit selected from the range will be 4. Thus, the gene that should be mutated is the 4th time-unit of the 3rd element. The mutated MoG is shown in Figure 15. Since the mutation function is not violating the structural stability rules, the validation of the genome is not needed.

$$MoG = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Figure 15- MoG after Mutation Function

Genome Validation

As mentioned earlier in "Crossover" section of this chapter, it is very likely that the newly generated genomes after the crossover function are not valid project schedules. This invalidity can be due to not scheduling at least one element for installation at all or schedule it more than one time to be installed. Considering the Figure 13, both child MoGs are invalid, considering the two above mentioned criteria. To resolve this problem, the "Validation Function" is added to the methodology.

The validation function will take a look at the MoG and finds out which elements are not defined to be installed or they have multiple installation times. For the first case it randomly schedules the elements within the project duration and for the later it maintains the first installment and removes the rest. Taking the new MoG 1 of the Figure 13 as an example, the validated version of it will look like Figure 16. In this example, the second installation for the first element is removed and randomly the second element is scheduled to be installed in the third time-unit.

$$New\ MoG\ 1 = \begin{vmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{vmatrix}$$

Figure 16- Validated MoG

A simple example of how this methodology can generate project schedules, using all of the above mentioned functions, is available in the Appendix. The example is the project scheduling development for the model shown in Figure 8 using the proposed methodology of Chapter III.

Research Validation

For validating this proposed model, the “Experimental Validation and Design” is chosen and it fits well enough for this research, considering different types of validations (Landry, Malouin, & Oral, 1983).

Experimental Design

In an experiment, one or more process variables (or factors) should be changed intentionally so that the effect of the changes on one or more response variables could be monitored. The design of experiments (DOE) is an effective way in order to analyze the obtained data and produce valid and objective conclusions (NIST ITL, 2012) with a minimum number of experiments.

The variables selected to be changed in this research are as follows:

- Changing the Complexity of the input 3D BIM:
    - Number of elements
    - Connection types
- Changing Genetic Algorithm parameters:
    - Elite member percentage
    - Mutation rate
    - Number of genomes per population (population size)
    - Construction duration range

For the 3D model inputs, three different models have been created to represent simple, moderate, and complex BIMs. It should also be taken into account that in this chapter, the author is only focusing on structural models and architectural elements. This means that elements such as piping, equipment, and HVAC, are not included in these models. The level of complexity of the models is detected based on number of the structural elements, size of the model, and connection types between structural elements. The screen shots from three different BIM inputs to the method are shown in Figure 17.

The model (a) in Figure 17 is a simple structural model with 42 elements, 18 columns and 24 beams. The second model, (b), is a more complex model with 42 columns and 58 beams, summing up to 100 elements. The last model is a generic turbine building structural model with 274 elements that consists of 102 columns and 172 beams (146 girders and 26 joists). The last model is extracted from models of the typical turbine buildings used in the power-plant industry.

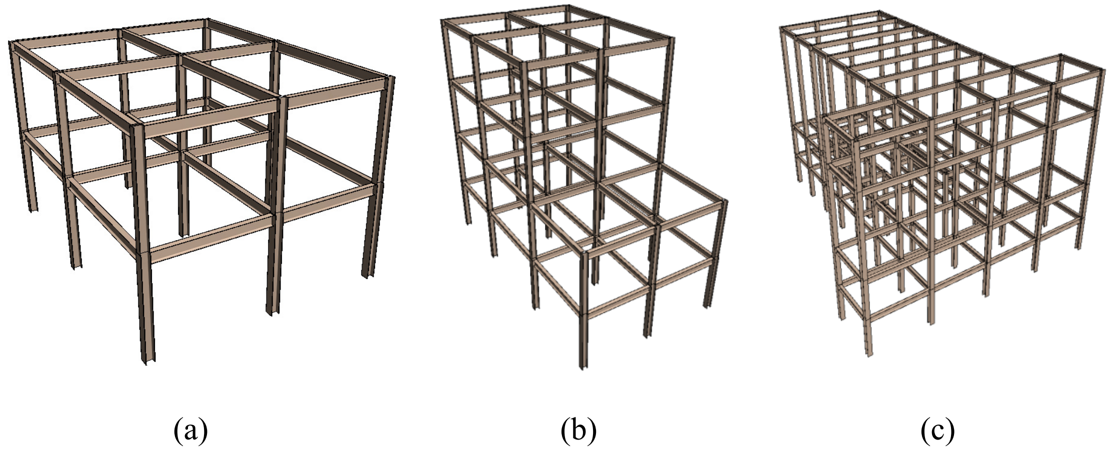

(a) (b) (c)

Figure 17- Different 3D BIM Input Models

For the complexity changes, the GA parameters would be set as following, not changing while switching between different complex models. Also the GA parameters changes are shown in Table 1.

- Elite number: 20% of the population
- Mutation rate: 10% of the population
- Population size: 30 genomes
- Duration range: 15±20% time-unit (time-unit is a generic duration unit that can be specified by the user as hour, day, or week at the initial step of the method)

Table 1- The Genetic Algorithm Parameter Change Sets

GA Parameter Sets:	1st	2nd	3rd	4th	5th	6th	7th
Population size	30	20	50	30	20	50	100
Elite number	20%	10%	30%	20%	10%	30%	5%
Mutation rate	10%	5%	15%	5%	15%	10%	20%
Duration range	15±20%	10±20%	20±20%	10±10%	20±10%	15±10%	25±10%

To design the experiment to prove the benefits of the proposed methodology for developing project schedules, the author executed 21 different combinations of parameter changes. These 21 different cases used in this experiment consist of 3 model complexity changes as defined in Figure 17 for each 7 GA parameter change sets shown in Table 1. All of the above changes in parameters (both GA parameters and model changes) would result in 21 different runs of the entire algorithm to validate its usefulness in almost any case.

The expected outcome for all of these different runs is to achieve multiple (the same number as the population size) complete construction sequences that satisfy the constructability and stability constraints of the model. These constructability constraints are calculated before in MoCC, as mentioned earlier. If it can be shown that all the different designed experiments are satisfying the objective of this research, it can be justified that the proposed algorithm is applicable for automatic development of stable construction project scheduling.

Results

As defined in the last section, different sets of inputs have been created and imported to the proposed algorithm to see how they could generate completely constructible project schedules. As described earlier, the constructability objective is defined as, having the elements of the project scheduled for the construction in a way that the local and global stability of the project (model) is preserved. This stability of the elements and the model is controlled by the MoCC that is calculated earlier. More details on how these different runs reach the objective of this research is shown in Table 2 and Figure 18. The execution of this methodology was performed on regular personal computer (CPU: Intel® Core™ 2 Due @ 316GHz, RAM: 8GB, OS: Windows® 7 Enterprise 64-bit).

Table 2- Completion Results for Different Runs for the Experimental Design

Run Name ↓	1^{st} 100% score occurred @	Ended @	Calculation duration per generation (second)
Simple, 30G, 15T	82	100	0.05
Simple, 20G, 10T	535	559	0.04
Simple, 50G, 20T	75	99	0.10
Simple, 30G, 10T	98	121	0.04
Simple, 20G, 20T	276	306	0.04
Simple, 50G, 15T	67	97	0.08
Simple, 100G, 25T	36	64	0.26
Moderate, 30G, 15T	7,785	7,817	0.13
Moderate, 20G, 10T	49,932	50,050	0.07
Moderate, 50G, 20T	4,916	4,954	0.24
Moderate, 30G, 10T	8,217	8,281	0.10
Moderate, 20G, 20T	4,325	4,364	0.11
Moderate, 50G, 15T	4,888	4,915	0.23
Moderate, 100G, 25T	3,979	4,013	0.66
Complex, 30G, 15T	81,183	81,210	0.68
Complex, 20G, 10T	244,200	244,709	0.42
Complex, 50G, 20T	4,146	4,204	1.26
Complex, 30G, 10T	52,078	52,165	0.57
Complex, 20G, 20T	105,523	105,657	0.54
Complex, 50G, 15T	17,185	17,215	1.05
Complex, 100G, 25T	13,896	14,512	3.43

The first label for the “Run Name” field in the Table 2 is as, Simple, Moderate, or Complex that are defined earlier as input model complexity shown in Figure 17 representing models (a), (b), and (c) respectively. In that field, the label G is showing the number of genomes in each population for that specific run and the label T represents the mean of the initial duration range as defined in Table 1.

Figure 18 shows the trend in which each of the designed experiments followed for maximizing the objective. The goal of this research, as mentioned earlier, is to have stable and constructible project schedules for any given 3D model. By maximizing the defined objective, constructability score, the GA tends to incline to the highest score, 100%, in each population generation step. As seen in the Figure 18, some of the designed experiments reach the complete score much faster than the others. This difference in the pace of completing the calculations is due to several parameters. The two most important effecting parameters are: input 3D model complexity and number of genomes in each population. The more complex the input 3D model is, the harder for the algorithm to schedule the entire 3D model elements. Since the infinite number of model elements is not possible, the calculation will always merge to the defined objective. However, the calculation time increases by having more elements in the model. Also by increasing the number of genomes in each generation, there will be better chances to have better crossovers and mutations to reach the goal of the experiment. To show the trade-off between the different input variables and the calculation duration of the proposed algorithm, a new metric is defined in this research. This new metric is named NoG (the Number of Genes in each population) and is calculated as shown in Equation 9.

Equation 9- Calculation of NoG (Number of Genes in Population)

$$NoG = Average\ genome\ length \times Population\ size$$

Where based on Equation 4:

$$\begin{aligned} &Average\ genome\ length \\ &\qquad = Number\ of\ 3D\ elements \times Mean\ of\ initial\ duartion\ range \end{aligned}$$

The correlation coefficient between the NoG of the 21 runs and their calculation duration per generation, as shown in Table 2, is equal to +0.9, showing a high positive correlation between these two variables. This correlation score clearly shows that by increasing the size of NoG (either by increasing the number of elements, initial duration range, or population size), the completion of calculation will take a longer time for each population. More correlation coefficient calculations are shown in Table 3.

Table 3- Correlation Coefficient Table

	Calculation Rounds	Calculation Duration per Generation	Total Calculation Duration
NoG	-0.05	0.98	0.30
Number of Elements	0.58	0.62	0.76
Population size	-0.32	0.55	-0.11
Mean of the Initial Duration Range	-0.31	0.43	-0.11

Table 3 shows how the generation parameters are interacting with the calculation process. As seen in that table, NoG has an extreme (>90%) positive impact on the generation calculation time and minor (<40%) impact on the entire calculation of the runs. On the other hand, the number of elements of the input 3D model has moderate (>40% & <90%) positive impact on the calculation rounds and durations. Population size and average initial duration of the runs have very similar impact on the calculation process. These two both have minor negative impact on calculation rounds as well as total calculation time, beside slightly higher but positive impact on the each generation calculation duration.

Furthermore, the calculation duration for each generation is increased almost on the same scale of increase in NoG. The calculation duartion is also increased around half the scale of increase in either number of 3D model elements, population size, or average duration of the initial population (as the user inputs). On the other hand, the number of total generation rounds to complete the schedule generation process increased in half the scale of increase in number of 3D model elements. Also it is decreased in the third scale of increase in either population size or average initial duration. As can be seen in the Table 3, the increase or decrease in NoG would not affect the total number of calculation rounds, but its increase would increase the total calculation duration by the scale of 30%. The total project duration is increased by the factor of 76% of the scale of increase in number of 3D model elements. In the meantime, it decreased by 10% of the increase scale of the population size and average initial duration.

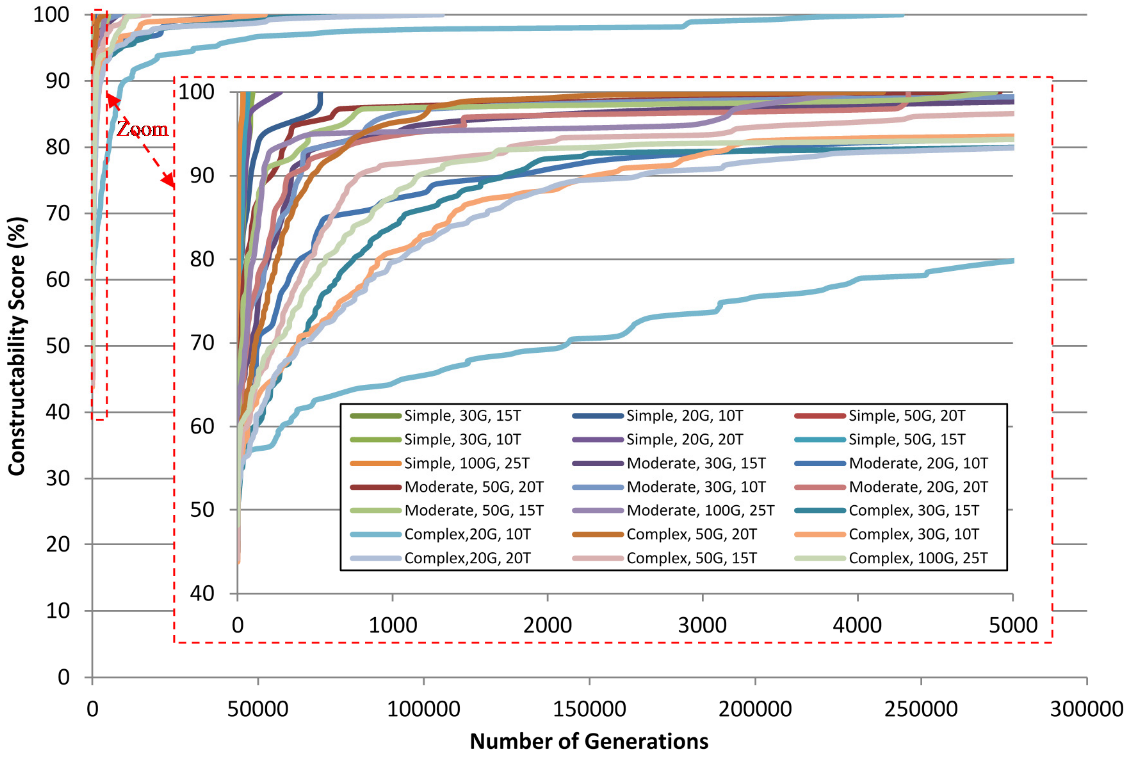


Figure 18- Experimental Design Results

Figure 18 shows the outcome plot of all the 21 different experiments. As can be seen in both regular and zoomed views, the experiments merged to the score of 100% as the objective of this research. Some of these runs reach the final score much faster than the others, as described earlier in this chapter (see Table 3 and its descriptions). Generally, the similar pattern in all the runs shows initial sharper increase in the constructability score that become steeper as the elements are getting scheduled.

The developed tool for this methodology not only generates stable construction schedules for the project, it also shows the 4-dimensional representation of the construction sequence, illustrating the completion of the 3D model in the time spans. This type of outcome, besides showing how the project is supposed to be built, can be used to evaluate the project schedule for stability and constructability too. Figure 19 shows eight screen-shots of the generated 4D construction sequence animation for one of the designed experiments (Moderate, 50G, 15T).

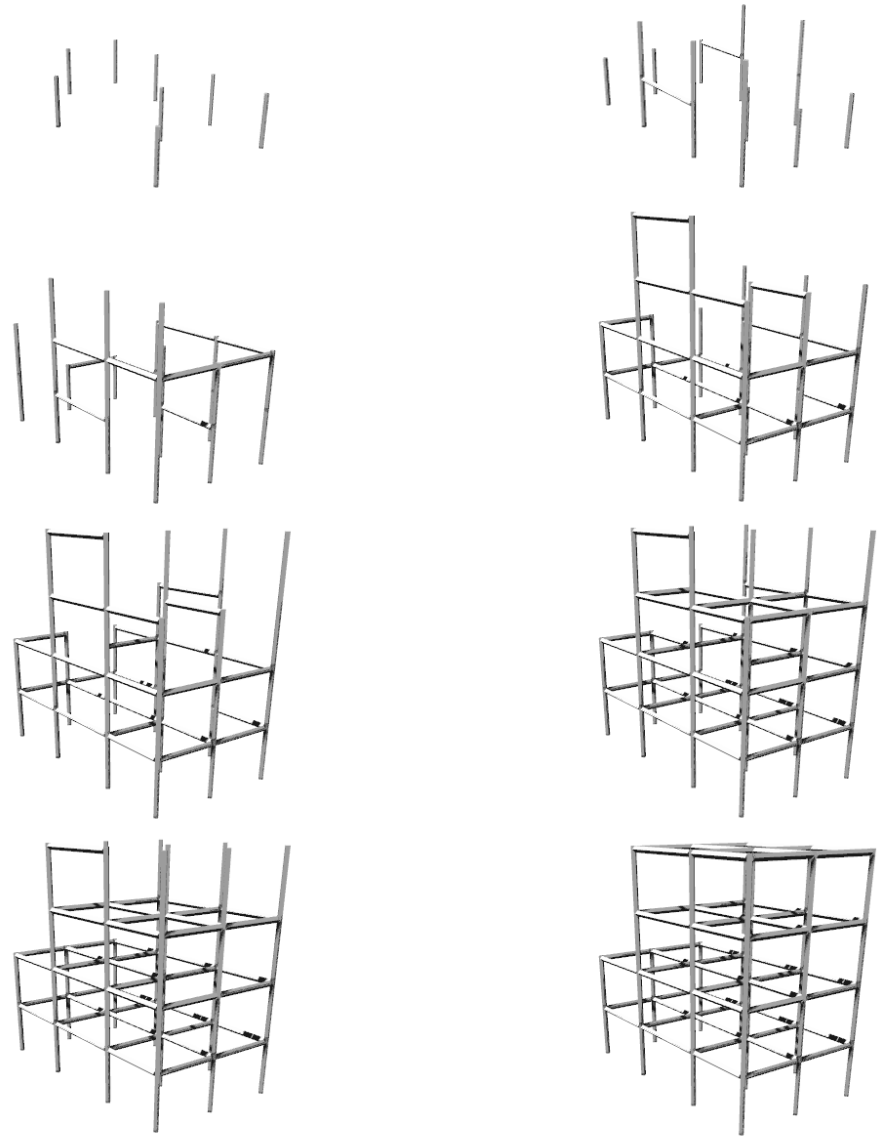


Figure 19 -Screen-shot of a Generated Construction Sequence (Moderate, 50G, 15T)

Conclusion and Future Work

As it has been shown in the result section, all the different inputs to the proposed algorithm merged up to reach completely constructible project schedules. Although the runs needed different calculation times based on their inputs, the performed experimental design proved that this method can produce 100% constructible schedules and is working. As mentioned earlier in this chapter, the main objective was to retrieve construction project schedules from inherent geometry information embedded in BIM of the project. This objective has been proved to be achievable through this methodology.

To make this methodology more useful for future projects and industrial and educational use, different element types (such as walls, doors, windows, HVAC, pipeline, pumps, etc.) should be added to the MoCC creation approach. Adding these new types of elements to the MoCC creating algorithm will enable the application to handle oil and gas projects, which have inherent difficulties to develop project network and schedule. In addition to extending the detection of more element types, different objectives should be added to the constructability objective mentioned in this chapter. Other objectives could lead the outcome of the methodology (project schedules) to be more optimized in cost, time, and even workability in the real job-site. Adding all of these improvements to the proposed algorithm, enables it to produce semi-perfect project schedules in terms of reducing construction duration, minimizing labor cost, and site mobilization and workability of the construction processes.

After future extensions of the current work, the outcome will can tested to see the benefits of this methodology of automatic project scheduling from the project BIM in an educational environment. To do this evaluation, the previously proven tool of predicting future of the projects in educational environment, called Project Management Prediction Market (Damnjanovic, Faghihi, Scott, McTigue, & Reinschmidt, 2013), can be used alongside the extended version of this proposed methodology.

CHAPTER V

EXTENDED GENETIC ALGORITHM FOR OPTIMIZED BIM-BASED CONSTRUCTION SCHEDULING[§]

Introduction

The determination of project schedules is a critical part of all types of projects, including engineering operations, manufacturing, construction, maintenance, and others. However, engineering education, whether at the graduate or undergraduate level, typically provides little instruction on how to develop good construction or fabrication schedules. Construction engineers and managers working on the projects learn on the job how to visualize the sequence of activities that will lead to good and feasible schedules without formal training. By integrating project scheduling with virtual three-dimensional geometric modeling, the author believes that students would learn how to generate more effective project networks and schedules through hands-on interaction and the use of the system.

The main purpose of this chapter is to extend the algorithm described in Chapter IV to support more construction elements from the 3D model and also to extend the objectives in the Genetic Algorithm. These additional extensions will result in more workable project schedules, with regard to supporting and calculating the typical construction project components. Also, these changes will produce optimized output

---

[§] This chapter is submitted to "Computer-Aided Civil and Infrastructure Engineering" as an individual paper and is under review (Faghihi, Reinschmidt, & Kang, Extended Genetic Algorithm for Optimized BIM-based Construction Scheduling, 2014).

construction sequences toward construction duration, project cost, and job-site movement distances of machinery and crew. The job-site movement distance mentioned in this chapter is the total distance between each set of installations of the project elements and will be described in more detail later on.

To conduct these extensions, the previously developed processes of determining the MoCC in Chapter III were chosen and used along with GA to generate optimized construction project sequences. In the process of this proposed algorithm, in addition to beams and columns covered in the previous chapter, other common building components are detected from the 3D model, and the proper relationships and constraints are calculated and considered in generating the MoCC. These other building components are as follows: slabs (floors), roofs, walls, doors, and windows. This chapter is also adding three more optimization objectives to the algorithm introduced in the earlier chapters. These objectives will help generating more workable and reasonable project schedules.

Reading the Geometry

In this chapter, the same file format (IFC) for 3D model input to the algorithm is used. More definitions on this file format and how the reading and extraction algorithm is working will be described as follows.

Detecting More 3D Elements

The previously developed algorithm to prove the usefulness of the methodology in Chapter IV was supporting only the structure, columns (IfcColumn) and beams (IfcBeam), from the IFC file format of the BIM of a project. This chapter extends the algorithm in Chapter III by adding the support of more element types of the project. For the purposes of this chapter, the authors view element types as detecting, reading, and calculating geometry information of slabs (IfcSlab), roofs (IfcRoof), walls (IfcWall), doors (IfcDoor) and windows (IfcWindow). For the simplification of the calculations, the first two element types that were covered in the earlier chapter (beam and column) were assumed as lines with a boundary box around them. In this chapter, also for simplification, these new element types are assumed as plain square surfaces with boundary boxes to calculate connections.

By going through the IFC standard (buildingSMART, 2011), defining the dimensions of all of these elements can be calculated and simplified to just start point and end point of a plate. For instance, three variables are available in IFC file for standard definition of a door, the placement point, width, and height. Using the last two variables along with the placement point, as the starting point, the end point of the square surface (door element) can be calculated as shown in Equation 10. Similar calculation is possible for other elements considered in this chapter.

Equation 10- Calculating the End Point Having Placement Point, Width, and Height

$$\begin{aligned} &Placement\ Direction \\ &\quad = (Width\ in\ X\ Direction\ ,\quad Width\ in\ Y\ Direction,\quad Height) \\ &\quad = (PD_x\ ,\quad PD_y\ ,\quad PD_z) \end{aligned}$$

$$First\ Point\ Coordinates = Placement\ Point = (FP_x\ ,\quad FP_y\ ,\quad FP_z)$$

$$End\ Point\ Coordinates = (FP_x + PD_x\ ,\quad FP_y + PD_y\ ,\quad FP_z + PD_z)$$

The logic behind finding and calculating the connections and relations between elements is similar to the previous chapter. Whenever two elements are intersecting within their boundary box regions, they are assumed as physically connected. Knowing these connections from the calculation of the data retrieved from IFC file and considering the following stability common knowledge rules, the MoCC can be generated as described in Chapter III. These stability common knowledge rules can also be summarized as Table 4.

- Upper level columns should be installed after the lower level columns,
- Beams should be installed after supports at both ends,
    - Cantilever beams will have one-end support,
- Walls should be installed after adjacent columns of the same level and adjacent the same level beams and the lower level beams (all four sides),
- Doors should be installed after the walls including them,
- Windows should be installed after the walls including them,

- Slabs should be installed after all the beams in the slab region from the lower level,
- Roofs should be installed after all the beams of the lower level.

Table 4- Stability Prerequisites Common Knowledge

	Lower Level	Same Level	Upper 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	-

Having the MoCC generated and ready, the next step would be developing and extending the GA in a way that it can optimize the new objectives of time, cost, and movement, which will be elaborated in more detail later on this chapter.

The defined objectives in this optimization are minimizing the total project duration, minimizing the total project cost by minimizing the labor cost and minimizing the needed job-site movements for the machinery and workers for more workability in the project site. These objectives are described in detail in Fitness Function section. The primary objective that is very critical to be fully met is project constructability. This means the GA would try to find constructible project schedules for the given project 3D model. The constructability of a project schedule herein is defined as having all the elements stable during and after the installation process. For example, the installation of a beam is considered constructible and stable only if the two end structural supports (columns or beams) have been installed earlier.

Defining GA Functions

Below are the general descriptions of core Genetic Algorithm functions and their definitions in this chapter.

Genome Creation

The genomes in this chapter have the exact same structure and definition as the previous chapter and consist of lists of elements to be installed in each time-unit (e.g. day, week, or month) throughout the total project duration. By this definition, a genome can be shown in either of the following two ways, the Matrix of Genome (as shown in Equation 2) and genome as a string (as shown in Equation 3), which is ready to be used in a GA population.

The difference between genomes in this chapter and the earlier one is in the way they are generated. The previous chapter focused on random genome generation that converged to the fully constructible ones. In the current approach, the genomes are generated based on MoCC and they will be fully constructible (the constructability score would be 100%) from the beginning. This approach for generating a genome is due to the critical importance of constructability of the construction schedules that should be met throughout the entire process.

The steps toward generating fully constructible project schedules based on MoCC would be as follows:

1. List all the elements with no prerequisites for installation (the elements that do not have 1 in their associated row in the MoCC).
2. Pick a random number from the elements in the list. This number can be from zero to the total number of the elements in the list.
3. Schedule these selected elements for the first (if it is the first round) or next time-unit in the construction schedule.
4. Remove the recently scheduled elements from the list of elements that are available for installation.
5. Find the list of new elements that can be installed since their prerequisites are installed already.
6. Add these elements to the list of available elements for installation.
7. Repeat from step 2 to 6 until there are no more elements left unscheduled.

The above steps can be summarized in Figure 20.

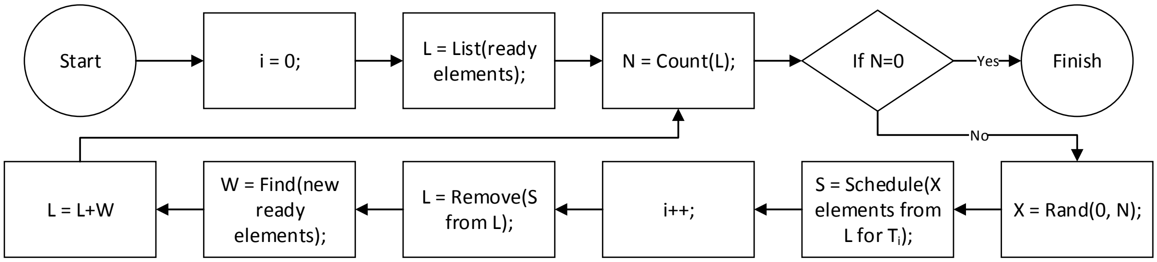


Figure 20- Genome Generation Steps

To simplify the genome, each element would be installed in only one time-unit and no more. By this simplification, in each row of the MoG there would only be a single 1 and all other values would be 0.

Elite Members

Similar to previous chapter, when generating a new population, some of the better genomes are allowed to move from the current generation to the next generation, unchanged. This method is known as elitist selection and those selected genomes are called elite members of the old population.

Fitness Function

The fitness function for the GA has multiple variables to measure and in this case it is considered as a multi-objective GA. As mentioned before, other than constructability, there are three objectives for optimizations that will be considered in this chapter. These objectives are project duration, project (labor) cost, and job-site movements.

Project Duration

As shown in Equation 4, 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.

Direct Project Labor Cost

Since the material take-off calculation is not in the scope of this research, the calculation of the project cost does not reflect the cost for material. The author's approach to calculate the labor cost in this chapter is based on the resource information entered by the user, which are the resource availabilities and their associated costs data. For instance, a user may enter the following inputs to the algorithm:

- Maximum number of columns per day: 4 Cost for this installation: $400/day
- Maximum number of doors per day: 5 Cost for this installation: $200/day

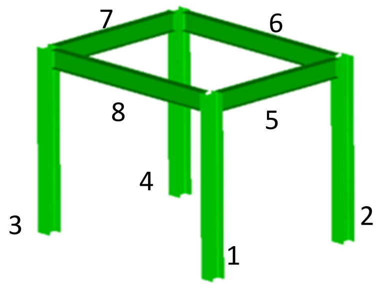
Considering the above example, the calculation of the cost objective will be $400 for any project day that has 1 to 4 column installations schedules and/or $200 for any day with 1 to 5 door installations. If there is a day in which there is no column scheduled to be installed, there would be no calculated 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 multiple the cost of the extra column installations by 1.5, simulating the cost for overtime work. Based on the Fair Labor Standards Act 1938 of the United States (U.S. Department of Labor, 2009), which has guaranteed “time-and-a-half” for overtime in certain jobs, the multiplier of the regular work cost, “1.5”, is used in this chapter. The logic behind this way of calculating overtime work is that the algorithm will assume installations of 4 columns (as user input) will take one full working day and any extra installations needs to be installed as overtime works. For the above example, if there is a day with six columns scheduled to be installed, the associated cost for that day for column installations would be:

$$Cost = \$400 + \frac{\$400}{4} \times 1.5 \times 2 = \$700$$

Job-site Movements

Reducing the required movements of the crew and machinery in the job-site for installing the project elements can increase the workability of the construction processes. 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 the installations in a single time-unit and also between time-units can be calculated. In this chapter, the author defined the following method to calculate total distance between installations of the elements. 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 these central positioning points of each time-units 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.

As an example of how to calculate the total movement, a simple structure as shown in Figure 21 is used. As shown in the figure, a sample MoG (Figure 21, b) for construction sequence of the 3D model (Figure 21, a) is demonstrated. In the given genome, which has 5 unit-time as total construction duration, the associated installation time-unit for each elements of the model are indicated as one. For instance, elements number 1 and 3 are scheduled to be installed in the first time-unit and element number 8 is scheduled for the second. For the calculation purposes the author assumes that the distances between column 1 and 3 is 10 feet, the height of the structure is also 10 feet and it is symmetric in all directions.



(a)

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

(b)

Figure 21- Sample Structure (a), a Sample Construction Sequence (b)

To visualize the installation distances associated to the MoG and 3D model shown in Figure 21, a schematic top-view of the distances with consideration of the sequencing is drawn in Figure 22. As shown in Figure 22 (a), which is top-view for installation distances of the columns, it is shown that column 1 and 3 are installed together and in the first time-unit, the column number 4 in the second time-unit, and number 2 in the third. In the first time-unit, the installation distance is equal to the distance between column 1 and 3, which is 10 feet. In the second time-unit, there is only one column installed, therefore the installation distance would be equal to the distance of column 4 to the central positioning point of the previous installations (i.e. columns 1 and 3). For the third time-unit, the installation distance is simply calculated as the distance between column 2 and the last installation, column 4. With the similar concept and calculations, the total installation distances for beams shown in Figure 22 (b) can be calculated. Calculated lengths of installations for the given example are shown in

Equation 11 (a) and (b). 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 Equation 11 (c).

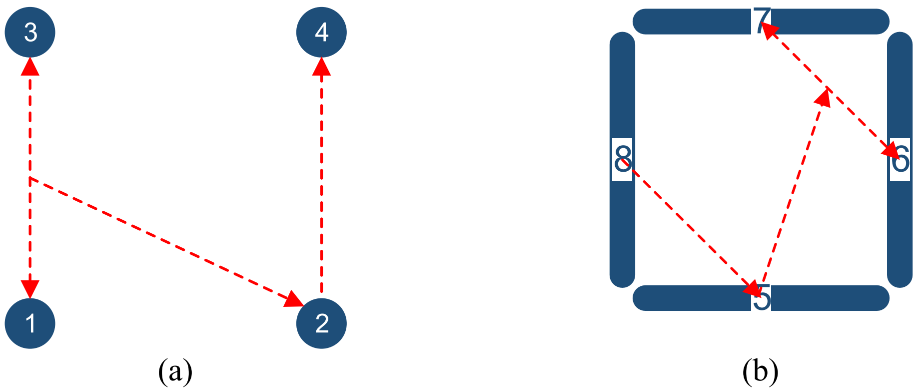


Figure 22- Top-view of the Installation Distances for Sample MoG, (a) for the Columns and (b) for the Beams

Equation 11- Total Installation Distances of the Example

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

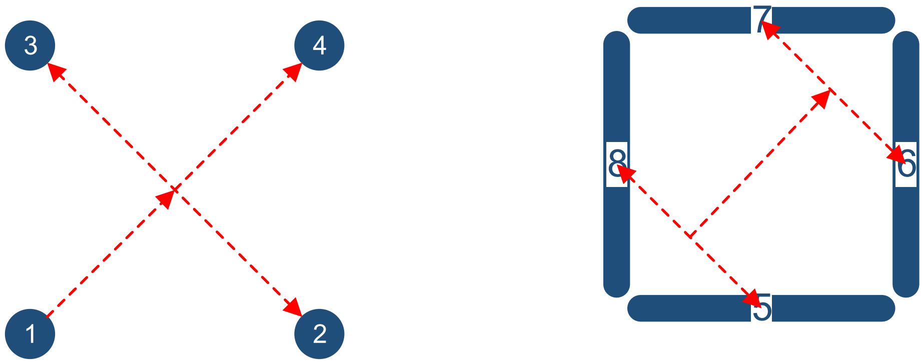
$$b)\ Total\ installation\ distances\ for\ beams = 7.07 + 7.91 + 7.07 = 22.05^{ft}$$

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

In case the given genome for the above 3D model was as Figure 23, the top-view distances and calculation would be as shown in Figure 24 and Equation 12 respectively. By changing the order and number of element installations, the total installation distance as well as the construction duration have been reduced, while the sequence looks more logical considering the construction directions and paths.

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

Figure 23- New MoG for the Example



(a) (b)

Figure 24- Top-view of Installation Distances for the New MoG

Equation 12- Total Installation Distances of the New MoG

$$a)\ Total\ installation\ distances\ for\ columns = 7.07 + 14.14 + 7.04 = 28.28^{ft}$$

$$b)\ Total\ installation\ distances\ for\ beams = 7.07 + 7.07 + 7.07 = 21.21^{ft}$$

$$(c)\ Total\ installation\ distances\ = 28.28 + 21.21 = 49.49^{ft}$$

This example can briefly show how the reduction in the movement objective score, as described by the author here, can result in more workable and logical installation sequencing. By defining this method in the fitness function of the GA even in more complex models, schedules with less job-site movement distance can be found in each generation cycles.

Objective Summation

Among different approaches to calculate the fitness function score in multi-objective GA (Konaka, Coit, & Smith, 2006), 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, all the values for the same objective are summed up and divided by the number of generation population to normalize the scores. Then, the user defined weights would be multiplied to the respective normalized scores and then all three scores are summed up to form the final score. The minimizing function is shown in Equation 13.

Equation 13- Weighted Sum Approach

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

This weighting feature can later be used to tweak the entire algorithm to respond properly on restrictions such as time-driven or cost-driven requirements of the user. This tweaking can be done through the user input dialog of the developed algorithm, where the user sets objective weights for a specific run.

Selection Method

Similar to the earlier chapter, the Fitness Proportionate Selection (a.k.a. roulette wheel method) introduced by Holland (1992) is chosen to be the selection function to pick parent genomes for crossover function.

Crossover

In the GA, two genomes are selected as parents from a generation, and paired them for breeding two new genomes as their children for populating the new generation. As described before, the parent selection would be handled using the specified *Selection Function*. The crossover function in this chapter is almost the same as the previous development in the earlier chapter, as shown in Equation 6, with the same reasoning and concept, and mathematically described in Equation 14.

Equation 14- Calculating the Parents' Cutting Points (Crossover)

$$\{Cutting\ point\ of\ parent\ 1\} = random(1, \{duration\ of\ parent\ 1\} - 1)$$

$$\{Cutting\ point\ of\ parent\ 2\} = integer(\frac{\{Cutting\ point\ of\ parent\ 1\}}{\{duration\ of\ parent\ 1\}} \times \{duration\ of\ parent\ 2\})$$

The only difference between the crossover function in this chapter with the previous one is that, if either or both children of the crossover parents do not satisfy this requirement they cannot be accepted to be members of the next generation. That is because all the genomes of a population should be fully constructible. Therefore, the unsatisfactory child will be ignored and the crossover function will be repeated until two valid children are generated for the given parent genomes. This loop for crossover function is limited to 50 times, and if there is not two valid children from the parent by then, randomly generated genomes will fill the remaining (the parents adopt a child).

Mutation

Since the genomes (schedules) are very sensitive to the changes and they easily become invalid regarding the constructability of the sequence, this function is disabled in this phase of research for simplicity. As an extension to this research, adding a well-developed mutation function can increase the productivity of this research and may also speed up the objective minimization process times.

Genome Validation

Unlike the earlier research, since all the genomes have full constructability sequences guaranteed at the creation time, as described in Genome Creation section, there is no need to revalidate them. Therefore, this function is not taking any action in this chapter.

Research Validation

In Chapter IV the author validated the proposed method using the "Experimental Validation and Design" (Landry, Malouin, & Oral, 1983), considering different types of validations. In this research, several extensions are applied to the developed and validated model and its capability toward generating acceptable results is tested as follows.

Test Inputs

To test this extending research, a sample 3D model of a building is selected, as shown in Figure 25. 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 input along with the other input variables (shown in Table 5) used to run GA for testing the proposed algorithm with new extensions.

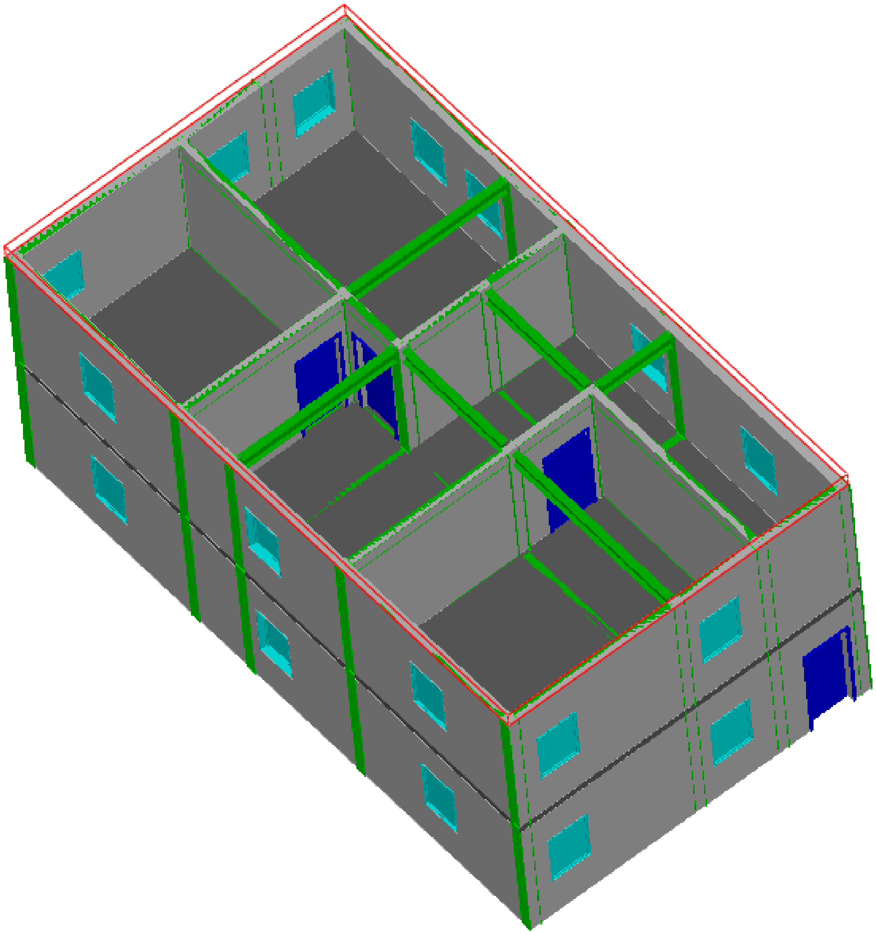


Figure 25- 3D Model Input

Table 5- Variable Inputs to the GA Calculations

Title	Value
Population per generation	10
Elite rate	20%
Installment (the 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

Results

The GA was allowed to take the action and run 20,000 rounds of generating populations, creating 200,000 genomes (schedules) in total. Out of all the generated schedules, the best schedule from the first generation and the 20,000th generation are compared in Table 6.

Table 6- Comparing the Best Schedules from the 1st and the Last Generations

	the 1st generation	the 20,000th generation	improvements	
Duration	31 d	23 d	8 d	26%
(Labor) Cost	$35,310	$34,940	$370	1%
Movement	2,314 ft	1,954 ft	358 ft	15%

The results show improvement in the values of all the objectives, while all the produced schedules still can be a valid construction sequence for the given 3D model. Figure 26 through Figure 28 show the trends of how the defined objectives were getting minimized throughout the GA calculations and population generations. This is evident by the graphs that the minimization process has taken place by GA.

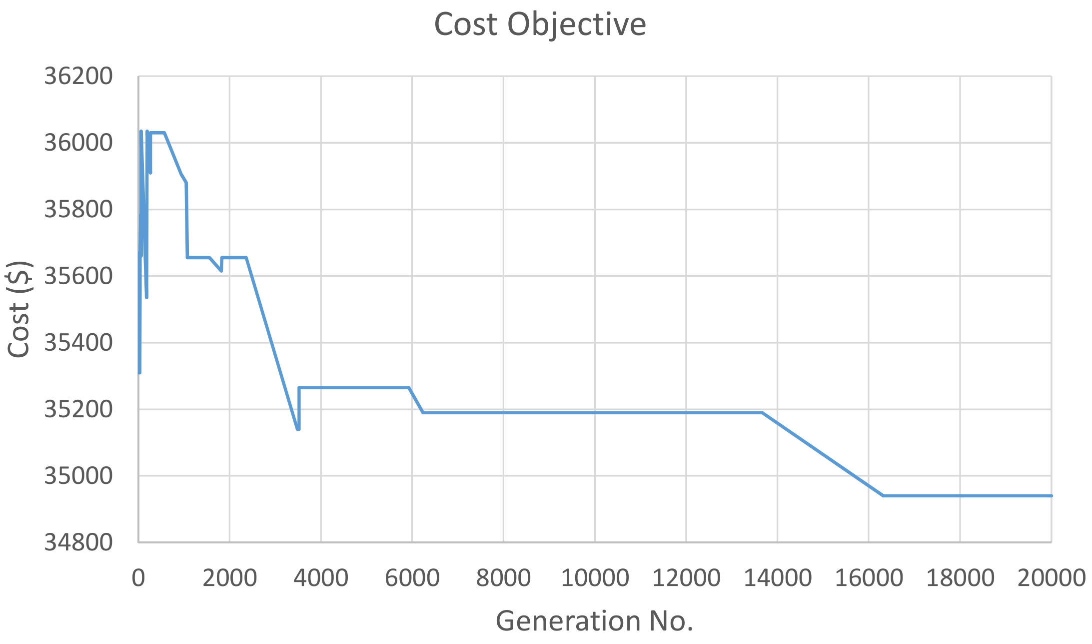


Figure 26- Cost Objective Minimizing in Generations

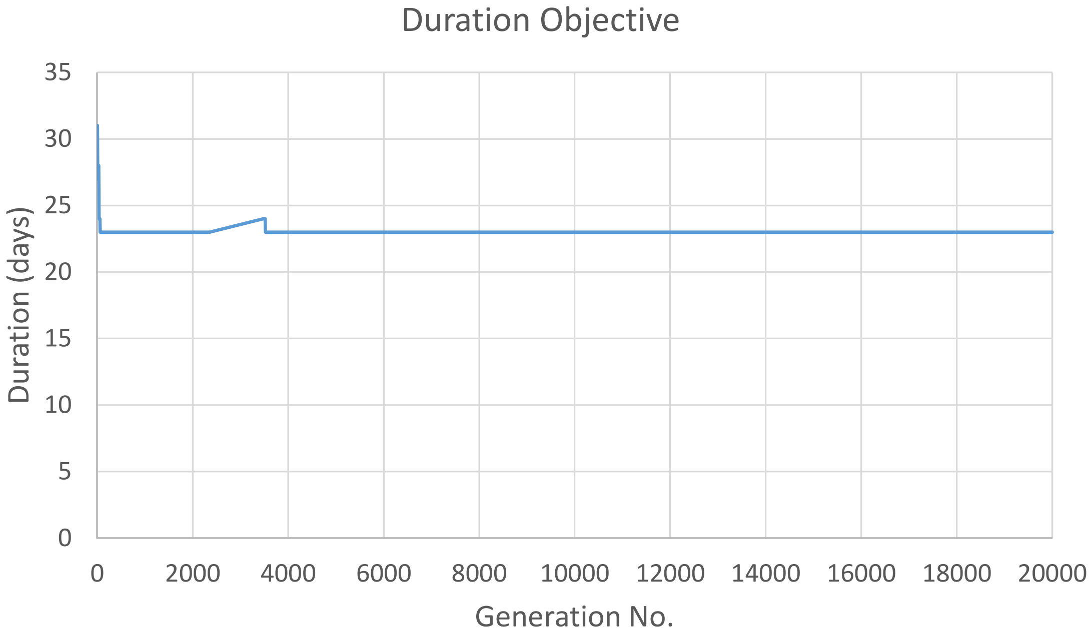


Figure 27- Duration Objective Minimizing in Generations

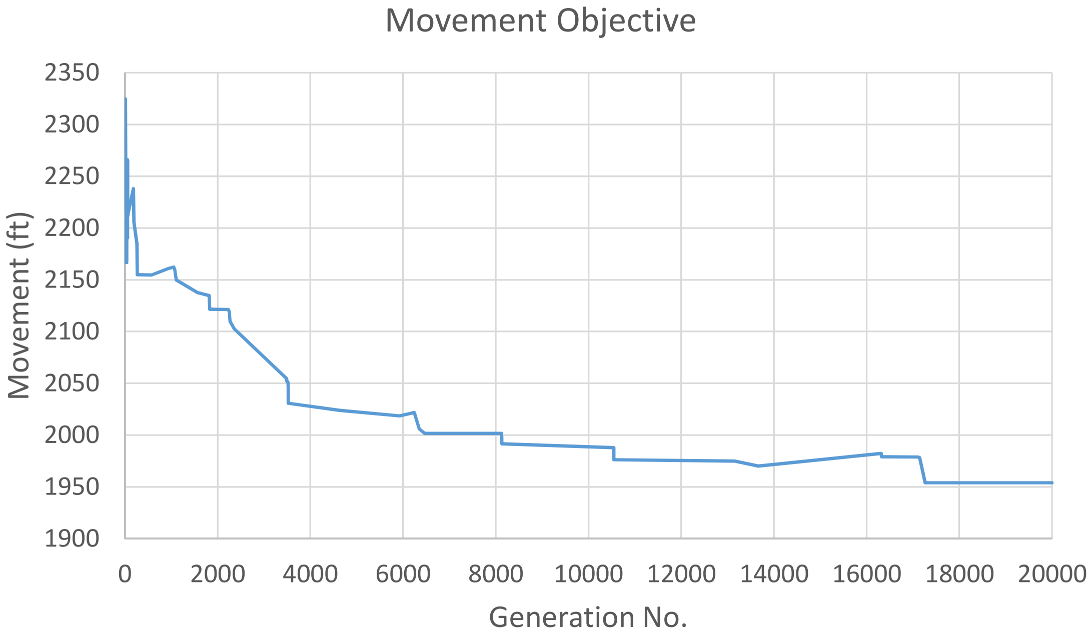


Figure 28- Movement Objective Minimizing in Generations

The small increase in the time and cost objective is due to significant improvement (shown as a huge drop) in satisfying the other objective (i.e. the movement objective), in that specific generation. The GA designed in this chapter considers all the objective at the same time, as described earlier. With a huge drop in one objective while the other two may increase a little, the overall objective score can still be reduced from the previous generation.

Conclusion and Future Work

As described in previous sections, this extension to the earlier chapter is shown to be successful. The extension includes supporting more element types in the detect, read, calculate, store, and draw modes of the algorithm. The other enhancement to the algorithm was calculation and consideration of three more objectives that have brought more logic to the outcomes of the entire process. This logic includes shorter durations, less costs, and less required job-site movements for the installation process of the project. Adding these objectives, the author believes a successful and tangible step toward reaching more workable and optimized construction project schedules has been taken.

Unlike other methods that optimize construction duration or focus on resource allocation and leveling, the proposed method offers many opportunities to modify the algorithm and improve performance through subsequent researches. Despite of all the efforts on this chapter, there are still places for improvements as the future works. A list of improvements to this algorithm can be as follows.

- Find out the relationships between the defined objectives using Pareto Frontier graphs.
- Adding support of more element types to cover more variety of the projects (i.e. pipes, stairs, devices, etc.).
- Defining a validation function that could find the constructability violating elements and with respect to their float times in the genome (schedule), reschedule them. This violation could happen in both crossover and mutation functions.
- Revise the crossover function to take benefits from the previous suggestion on validation function in a way that with each single crossover, two valid children are generated. With this enhancement the repeating process and killing invalid genomes will not be needed.
- Investigate other multi-objective GA approaches to reach the solutions faster, while still having the ability to find Pareto Frontier graphs.

CHAPTER VI

OBJECTIVE-DRIVEN AND PARETO FRONT ANALYSIS: OPTIMIZING TIME, COST, AND JOB-SITE MOVEMENTS[**]

Introduction

Extending or shortening the construction project duration clearly affects the total construction cost. The most important aspect is how project time and cost are related and with a single unit change in either of them, how much the other one would be changed. This means the in-between relationship needs to be formulated and shown graphically in order to bring a better understanding the effects. Several successful attempts have been conducted to show this relationship, which will be described in more detail later in this chapter. Different optimization tools have been applied to find time-cost relationship of the projects. In most cases, the 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.

The main purpose of this chapter is using the outputs of the previously developed algorithm to find the relationship between the defined objectives. These objectives are

---

[**] This chapter is submitted to "Journal of Construction Engineering and Management" as an individual paper and is under review (Faghihi, Reinschmidt, & Kang, Objective-driven and Pareto Front Analysis: Optimizing Time, Cost, and Job-site Movements, 2014).

“cost”, “time”, and “job-site movements”, described in detail in Chapter V. As a first step toward conducting this research of finding objective relationships, the author developed a matrix of constructability relationships between all the elements from the 3D model shown in Chapter III. This 3D model is the Building Information Model (BIM) of the project, and should be the main input to the algorithm. The mentioned matrix is called Matrix of Constructability Constraints (MoCC) and is defined as Equation 1.

Using the GA and the MoCC as the primary calculation basis for the GA fitness function, the author developed a method that was able to generate valid construction sequencing of the building structure for the given 3D model, as shown in Chapter IV. By “a valid construction sequence”, the author implies that all the project elements are scheduled for installation in a way that the structural stability requirements for the building are preserved throughout the construction process. To make the algorithm more mature and complete, the author defined a new objective as job-site movements first. Then, he implemented this new objective along with cost and time in the GA optimization process. In addition to these efforts, he extended the support of more project element types described in Chapter V. By developing this three-objective GA, the entire proposed method is able to generate constructible and optimized construction schedules only from the BIM of a project.

2D Pareto Fronts

Running the developed GA for 20,000 rounds of population generation with 10 genomes in each, produced 200,000 valid construction schedules. Each of these generated construction schedules has three fitness function scores for the three objectives defined earlier. Plotting each two objectives on a single 2D coordinate system shapes the following outputs.

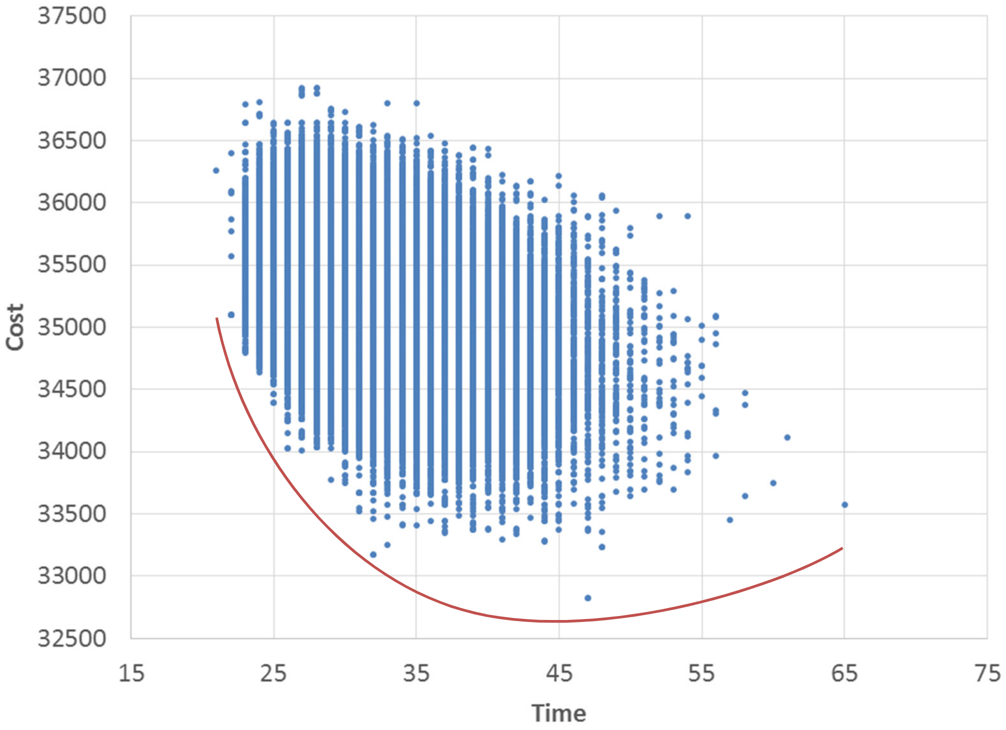


Figure 29- Cost-Time Pareto Front

Figure 29 shows by increasing the project duration, the labor cost for the project will be reduced and then starts to slightly increase. The reduction is due to decreasing the overtime works for the project. The later increase of the labor cost is caused by the growth in the number of working days.

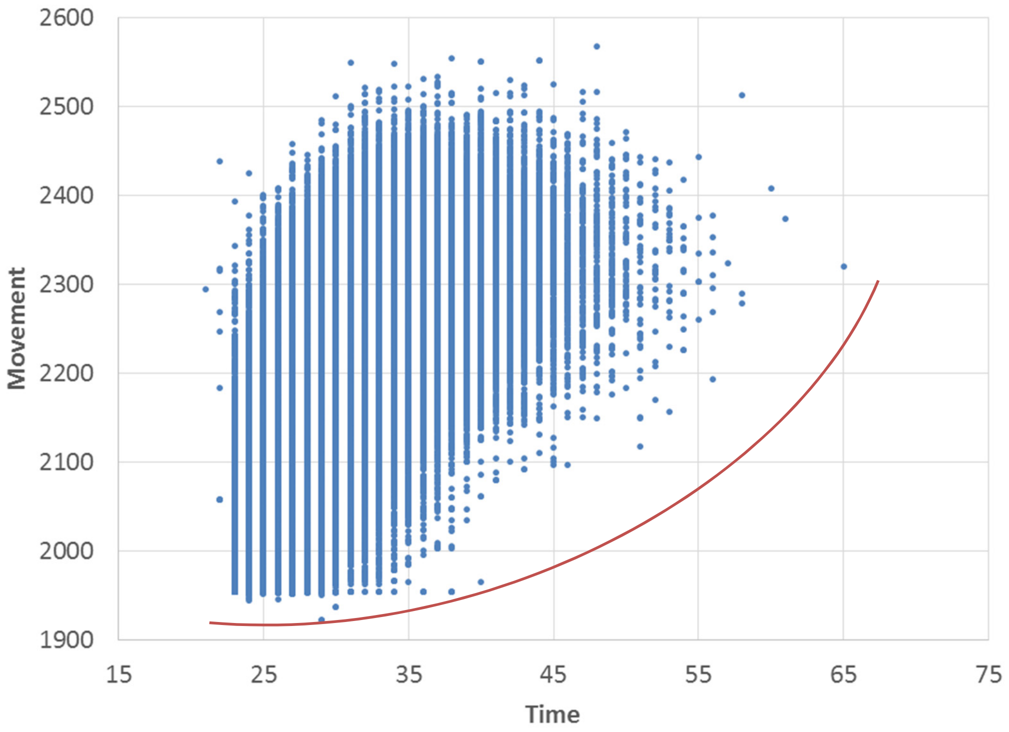


Figure 30- Time-Movement Pareto Front

The Time-Movement Pareto Front graph as presented in Figure 30 shows the 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 sets of installation in next the time step.

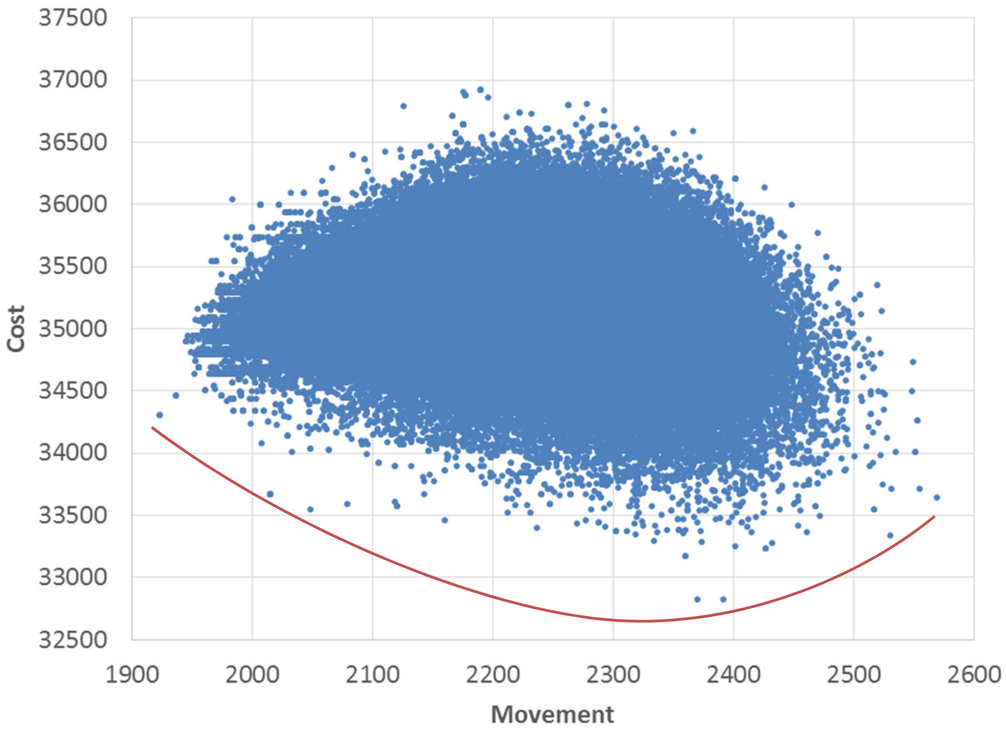


Figure 31- Movement-Cost Pareto Front

As shown in Figure 31, 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 and that is due to overtime installations of the elements. When elements are installed in sets of groups in each time step (e.g. 6 columns in a day) the total distances between these elements and also the distance between the mid-point of this installation group and the next one will be reduced based on formulation and calculations shown in Equation 11. 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 4 columns per day), the surplus elements (e.g. 2 columns) have the labor cost 1.5 times more than the regular installation. These two facts together make the labor cost higher when the movement score gets lower.

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

Solutions Cloud Point

As described earlier in this chapter, each of the generated project schedules has three objective scores for its time, cost, and movement objectives. Figure 29 through Figure 31 showed how all the 200,000 solutions can be represented in the 2D coordinate systems. Since there are three objectives defined in this research, it is possible to show all the solution points in a single 3D scattered plot, as seen in Figure 32. Further development of this analysis can generate a 3D Pareto Front surface showing the optimum relationship between the defined three objectives of this research.

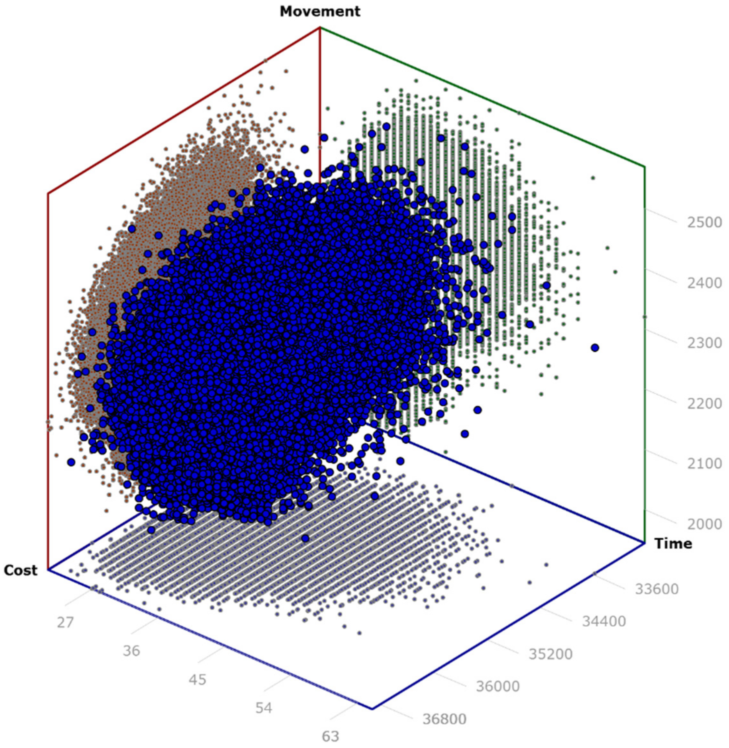


Figure 32- Solutions 3D Cloud Point

Objective Driven Analysis

Other than generating three shown Pareto Fronts and the solutions 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 reflect being cost-driven 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 equal. This equality of the objectives 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 elite member or for crossover function of the GA.

In this chapter, the author showed how the solutions cloud 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 the weight as 100 while the other two had been set to one. By inputting objective weights in this manner, in each of the new calculations, one of the objectives will be considered hundred 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 set the cost objective weight 100 times more that duration and movement objective weights, the algorithm will understand that the cost object is much more important than the other two. 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.

The Figure 29 through Figure 31 were showing 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.

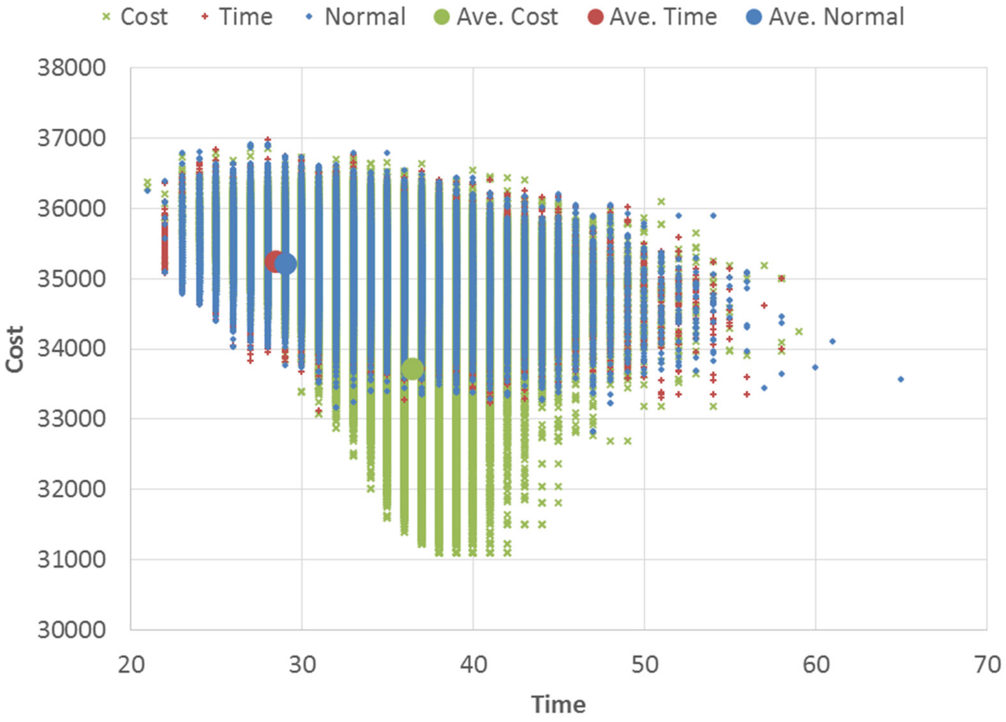


Figure 33- Time-Cost Cloud Point Comparing Objective-driven Calculations

As described in this chapter, 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 remained at one in that algorithm run, for each run 200,000 new project schedules have been generated. Figure 33 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 hundred time of the other two objectives) the entire cloud point (red pluses in the figure) are shifted 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 weights of the objectives were equal. The calculated average of the cloud points are shown as big red, green, and blue dots indicating average values for time-driven, cost-driven, and normal calculations respectively.

It is shown in the figure 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 is shifted to the left for shorter construction durations (or shifted down for less construction labor cost in cost-driven run). When the user intents to run the algorithm to be time-driven, the algorithm produces more construction schedules with shorter duration. Imagine this example project has a constraint of being constructed 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 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 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.

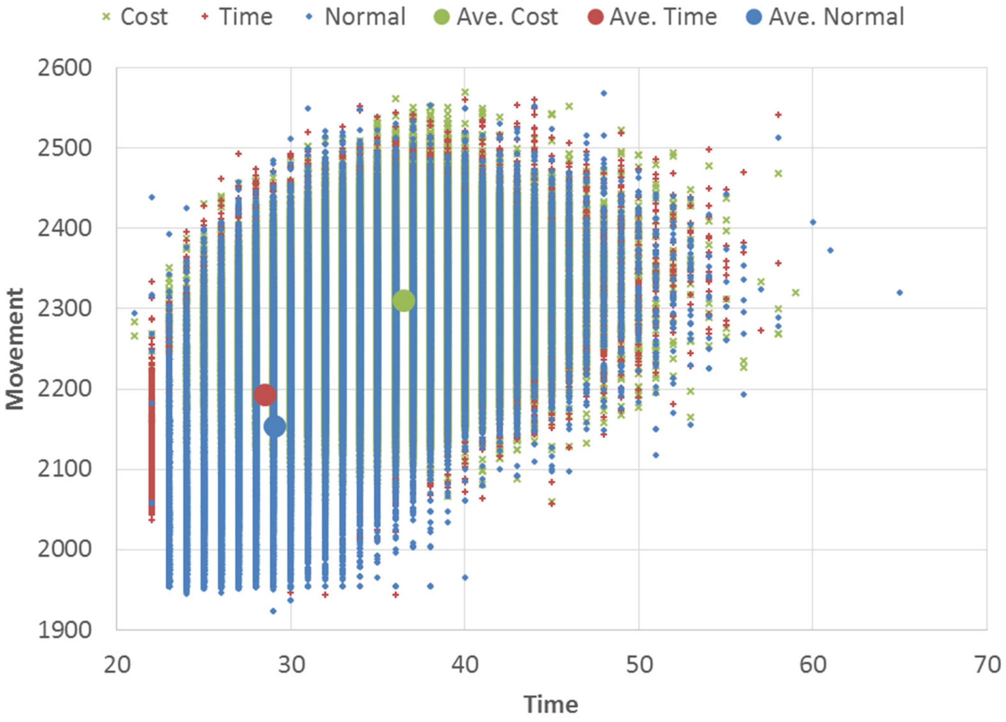


Figure 34- Time-Movement Cloud Point Comparing Objective-driven Calculations

Similar to Figure 33, Figure 34 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 the calculations by the user. Figure 35 shows the same behaviors in the Movement-Cost graph.

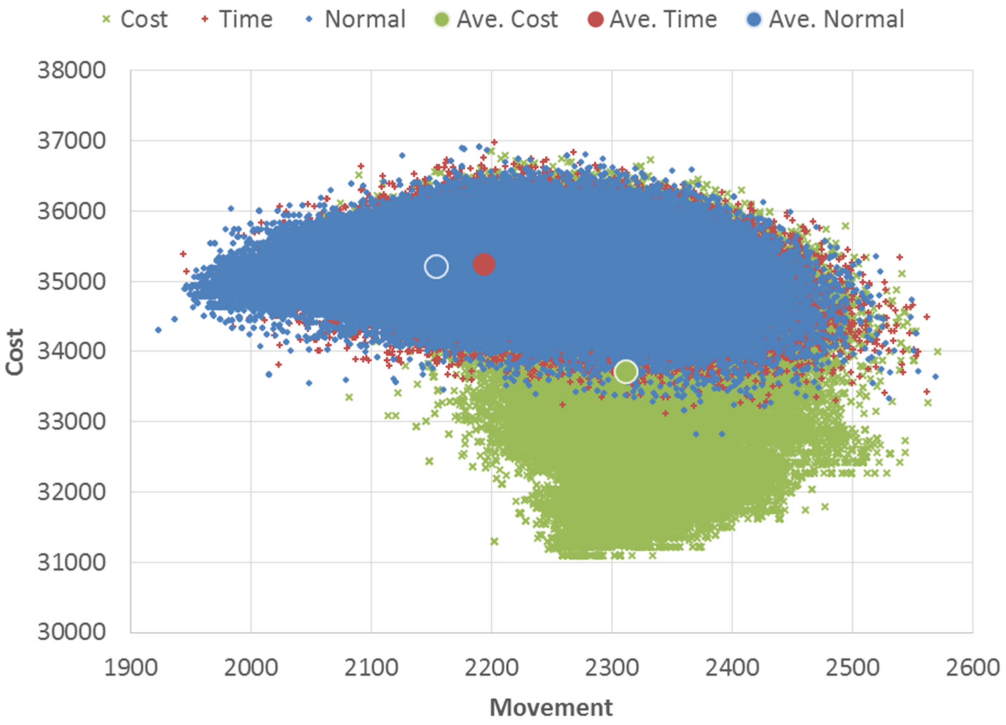


Figure 35- Movement-Cost Cloud Point Comparing Objective-driven Calculations

The differences in the average values of the cloud points in all three runs are shown in Table 7. As described before and as visible in Table 7, the cost objective score is increased in time-driven 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.

Table 7- 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	2,154	2,311.4	2,192.8

Conclusion and Future Works

This chapter showed some other useful benefits of the proposed construction scheduling algorithm. The algorithm, as described in earlier chapters, 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 defined in Chapter III. 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 it can show the 4D construction animation as shown in Chapter IV. The generated populations that contain construction schedules are ordered and handled based on their objective scores and those with better scores will have a higher chance to reach the next generations, which was described in Chapter V. As mentioned in this chapter, 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, solutions cloud point, and 3D Pareto Front surface (with later extensions), and reflect the objective driven nature of the project in the provided solutions for construction project scheduling problems.

CHAPTER VII

CONCLUSION AND FUTURE WORK

Conclusion

The present research shows how the usefulness of the BIM of a project can be extended to generate the construction schedule. Using the BIM of the project as the main input to the algorithm, several valid and optimized construction schedules can be developed that can be used in different ways. The analysis of objective scores of generated schedules can give an oversight of how the objectives are interacting with each other. Also, based on the total fitness function of each schedule, the user can select the one that best fits to the project. Therefore, the main contribution of this research is the automatic and optimized development of construction schedules from the BIM of a given project. Although the created project schedules are in their primary or initial version, more research is needed to demonstrate the algorithm can provide the best solutions. In addition to the main contribution mentioned, several other contributions are identified in this research in programming, mathematics, and construction fields.

Contributions

Web-based and Open-source IFC Reader

The developed web-based application for testing the algorithm is capable of detecting the predefined 3D elements in the Industry Foundation Classes (IFC) file of the project BIM and calculating the geometric information. Also, the application extracts certain information and form them in a useful table of data in the database. This reader application is written in PHP language and is open-source.

Web-based and Open-source 4D Animation Generator

When the 3D model in the format of IFC is imported to the algorithm, a 3D view of the model can be seen for controlling purposes. Also, when the GA part of the algorithm generates project schedules, the viewer module of the algorithm can display the associated 4D construction animation for each generated schedule. This viewer module is developed as a web-based application in PHP language, in line with other parts of the module, and is open-source.

Matrix of Constructability Constraints

The Matrix of Constructability Constraints (MoCC), as described in detail in Chapter III, is an automatically generated matrix from the BIM of the project containing the structural relationships between project elements. The MoCC can be used in any application that need mathematical understanding of the structural stability of a building. One of the usages of the MoCC is shown in this research as the development of constructible and valid project schedules.

When corroborated by other investigators from distinct industries, the MoCC would be a valuable innovation that can be quite different from other methods. As mentioned in the Chapter III, project progress control from image processing can benefit from this matrix. Another application might include manufacturing assembly. For instance, if a data-embedded 3D model of a car is given to the algorithm along with common knowledge of element type dependencies, the algorithm can generate the MoCC for the given model. Then, the rest of the algorithm can propose assembly sequence of the car.

Developing Structurally Stable Schedules with the Genetic Algorithm

Using the MoCC developed by the algorithm, the Genetic Algorithm (GA) part of this research approaches to generate 100% structurally stable construction schedules from random and invalid installation sequences. Using the GA to generate valid construction schedules is a new GA usage that has not been investigate before based on the literature review conducted in Chapter II.

Introducing Movement Objective

A mathematical function was needed to let the computer-based algorithm determine how the project elements should be scheduled to have a reasonable installation pattern. This function calculates distances between element installations and by minimizing that distance in the GA calculation cycles tries to install elements that are closer to each other. This function results in installation sequences that start from one side of the building and continues to the other side. Chapter V describes more on how this function is developed and calculates the objective score.

The Extended GA for Optimized Scheduling

Acceptable project schedules should not only be structurally stable and valid, but also they need to be optimum for some specific objectives. For this reason, three objectives were defined in this research to show how this algorithm can minimize multiple objectives at the same time. The three objectives were time, cost, and movement. The algorithm uses the multi-objective Genetic Algorithm method and develops optimized project schedule templates. These generated and optimized schedules can be used by the project manager or the management team in the project scheduling process.

Objective Driven Scheduling

The proposed algorithm in this research, unlike the other algorithm, can reflect the importance of an objective while generating schedules. When the user assigns more weight on a certain objective, the algorithm considers the user input to generate project schedules with less score for that specific objective. This reaction results in better project schedules regarding the more important objective (e.g., shorter schedules when duration objective is more important).

Cloud Point of Project Schedules

Plotting project schedules on a 2D or 3D coordinate system of the objective scores demonstrates how the project schedules are scattered in the graph regarding their objective scores. These 2D or 3D objective plots illustrate what the common objective score ranges are for the given project. These plots can also demonstrate the density of the project schedules in a 2D or 3D space, displaying the region the actual project schedule is most likely to be.

2D and 3D Pareto Fronts from BIM

Indicating the normalized total objective score by gradient colors for each project schedule on the graph shows where the best schedules are in the cloud point. Drawing a curved line on the boundary with minimum total scores indicates the near optimum relationship between the objectives. This line can be helpful to the project managers to understand how the objectives (e.g., time and cost) will effect each other when one is changed, specifically for the given project. A 3D Pareto Surface can be generated in the 3D cloud points, representing the optimum relationships of the three objectives of the project.

Currently Available Features

The features listed in this section are already available through the use of the current status of the proposed algorithm.

Scheduling Learning Tool

The inexperienced schedulers have so many things to learn while their schedules are compared to existing solutions that are not optimum solutions. To learn from experience, one must have a good teacher; to learn from existing schedules, one must have good solutions for comparison, or one learns the wrong things. The ultimate schedule obtained from the proposed algorithm starting schedule should be better than any schedule generated from any other method.

On the other hand, using the high calculation speed of computer-based programs, the proposed algorithm can generate project schedules rapidly. This high computational speed of the algorithm can help the novice project scheduler in learning the impacts of different changes to the project schedules. For instance, the project scheduler can change the placement of the project elements in the 3D model of the project and see how it will affect the construction sequence. Also, changing the importance of objectives could result in different policies for completion of the project. In addition, the 4D animation of construction sequence for all the developed schedules helps the scheduler to visually see the schedule and understand the changes deeper.

Customized Schedule Template

As a common practice in AEC industry for developing a project schedule for a new project, project schedulers usually use a project schedule from past projects as a starting template, which is similar to the current one. This adoption of an old project schedule may cause problems such as missing some parts of the new project, perpetuating similar past mistakes, etc. The algorithm proposed in this research not only is capable of generating template project schedules exactly for the current project, but also it can optimize them toward different user-defined objectives.

The algorithm is capable of finding numerous valid project schedules from the entire search space. The defined fitness function of the Genetic Algorithm assigns a wellness score to each schedules. A professional project scheduler can sort all the generated schedules by their scores and select the ones fitting better to the project. Then, the professional scheduler can start using his or her accumulated experiences from the previous works to tweak the automatically developed project schedules. This final fine tuning the schedules makes them more compatible with the real situations of the project and more acceptable to be the construction schedule.

3D Pareto Front

All the points from the GA calculations that have been plotted on 2D graphs can be combined and plotted in a single 3D coordinate system. The 3D plotting can bring in new information for the managers or schedulers to understand the overall objective relationships. In addition, if the normalized total fitness function scores are indicated on each score points using a color gradient, the (near) optimum surface, in which all the objectives are in their minimum values, will be visible. Further work on the 3D plot can show an actual surface for the Pareto Front.

Future Work

The introduced automatic and optimized algorithm for project scheduling in this research demonstrated its usefulness for the academic research purposes as well as the AEC industry. Those benefits can be listed as automatic project schedule development, useable as scheduling learning tool, customized scheduling template generation, and 3D Pareto Front analysis. However; applications are not confined to the AEC industry; assembly process in manufacturing is another potential application as briefly described in this chapter. In addition to all the mentioned benefits of the algorithm, there are still several potential profits in using the algorithm that need more extension to the existing development. A list of the potential useful extensions to the algorithm that would open new research fields for academic researchers and would help the AEC in project scheduling is described in the following section.

Extension Needing Features

Workaround Solutions

In many cases in construction job-sites, some of the needed materials for installation processes may not arrived on time. In these cases, the project manager or the project superintendent needs to have a workaround strategy to continue the work, but would that be the most optimum one? The algorithm developed in this research can also solve this type of problems. Using the MoCC defined in Chapter III, the algorithm can detect which project elements are depending on the lacking part. Then, it will schedule those elements that have no connecting with the lacking one.

In the further development of the algorithm, a feature can be added so that the scheduler identifies the element or elements that have not arrived to the job-site on time. Then, the algorithm considers these new limitations and starts to generate new sets of schedules.

Material Ordering and Dumping

The project schedules developed by the proposed algorithm contain all the needed information for the installation time of each project element. Knowing the exact time for the element installations allows the managers to order the required materials right before they are needed. Although the material provider cannot handle instant ordering, the precise knowledge of the date materials are needed will help managers to better plan the resource ordering process.

On the other hand, when a manager decides to order bulk materials at a same time and off-load them in the job-site, the proposed algorithm can recommend places for off-loading and storing the materials. Since the algorithm has all the BIM components linked to their schedule activities, the nearest location to all the dumping elements and materials can be calculated for any given time and any set of elements. This dumping location feature can help superintendents and project managers to better plan their job-site regarding the optimum material storage locations.

Built Sequence Policies

The introduced movement objective in this research calculates the entire distance between groups of elements installations. This calculation, which is described in detail in Chapter V, computes distances in X, Y, and Z directions first. Then, the computed distances are combined to reach the total distance. Since this calculation have distances in each direction (X, Y, and Z), further extension can prioritize the importance of minimizing the distance in any given direction. Therefore, by increasing the importance of the movement objective calculation in one direction, the algorithm focuses more on placing the elements in that direction. The following example can illustrate the described calculation more.

Imagine a scheduler prioritize the movement objective directions as X, then Z, and finally -Y for the beam elements of the project. For this example X axis is assumed from West to East, Y axis from North to South, and Z axis from the bottom of the building to the top. With these assumptions, the algorithm tries to install beams with the installation location priority from West to East, then bottom to top, and at the end South to North. In other words, the algorithm ultimately schedules the beams from South-West point of the building. The building will be completed in Z direction (bottom to top) faster than -Y direction (South to North) and slower than X direction (West to East).

Installation Starting Point

The proposed algorithm links all the spatial and geometrical information of the elements to their installation schedule. Additional extension to the algorithm can take the preferred starting installation point from the user and generate the construction schedule accordingly. In the extended version of the algorithm, if the scheduling user pinpoints a specific location from the 3D coordinate system, the algorithm detects the selected location and considers that as the starting point for installations. The movement objective, by minimizing the distances, will force the algorithm to start the installations from the closest elements to the selected point.

A more advanced version of the algorithm can take installation starting points for each element types of the building separately (i.e. beam, column, wall, etc.). For instance, the starting point for installing the beams may be different than the windows of the building. Also, adding the feature of having multiple starting points can be helpful. As an example, identifying crane locations for installing precast concrete elements or steel structure can be done with the multi-staring point feature.

Extended 3D Element Support

The current algorithm can read and calculate the geometric information of columns, beams, walls, slabs, roofs, doors, and windows and generate the MoCC based on their information. The algorithm also draws these elements in a 3D environment and connects this installation schedule automatically to generate 4D construction animation. Another extension to this algorithm can be adding support for more 3D element types (piles, pipes, ducts, stairs, ramps, etc.) to the algorithm element reader and viewer. By defining the common construction knowledge for the new element types, the algorithm can generate the MoCC and the rest of the calculations can still generate construction schedules.

REFERENCES

Aamodt, A., & and Plaza, E. (1994, March). Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications, 7*(1), 39-59. Retrieved from http://dl.acm.org/citation.cfm?id=196108.196115

Abd El Razek, R. H., Diab, A. M., Hafez, S. M., & Aziz, R. F. (2010). Time-Cost-Quality Trade-off Software by Using Simplified Genetic Algorithm for Typical-repetitive Construction Projects. *World Academy of Science, Engineering & Technology*, *37*, pp. 312-321.

Adeli, H., & Karim, A. (1997, December). Scheduling/Cost Optimization and Neural Dynamics Model for Construction. *Journal of Construction Engineering and Management, 123*(4), 450-458. doi:10.1061/(ASCE)0733-9364(1997)123:4(450)

Autodesk. (2013). *Bounding Boxes.* Retrieved August 24, 2013, from Autodesk Wikihelp: http://wikihelp.autodesk.com/Revit/enu/2013/Help/00001-Revit_He0/0328-Build_th328/0735-Structur735/0843-Joins_an843/0844-Bounding844

Avila Rondon, R. L., Carvalho, A. S., & Hernandez, G. I. (2008). Neural Network Modelling And Simulation Of The Scheduling. *Eighth IFIP International Conference on Information Technology for Balanced Automation Systems* (pp. 231-238). Porto, Portugal: Springer US. doi:10.1007/978-0-387-09492-2_25

Azaron, A., Perkgoz, C., & Sakawa, M. (2005, September 15). A Genetic Algorithm Approach for the Time-cost Trade-off in PERT Networks. *Applied Mathematics and Computation, 168*(2), 1317-1339. doi:10.1016/j.amc.2004.10.021

Azhar, S. (2011, June 15). Building Information Modeling (BIM): Trends, Benefits, Risks, and Challenges for the AEC Industry. *Leadership and Management in Engineering, 11*(3), 241-252. doi:10.1061/(ASCE)LM.1943-5630.0000127

Bai, Y., Zhao, Y., Chen, Y., & Chen, L. (2009). Designing Domain Work Breakdown Structure (DWBS) Using Neural Networks. *6th International Symposium on Neural Networks* (pp. 1146-1153). Wuhan, China: Springer Berlin Heidelberg. doi:10.1007/978-3-642-01513-7_127

Baker, J. E. (1987). Reducing Bias and Inefficiency in the Selection Algorithm. *Proceedings of the Second International Conference on Genetic Algorithms and their Application* (pp. 14-21). Hillsdale, New Jersey: L. Erlbaum Associates. Retrieved 06 16, 2013, from http://dl.acm.org/citation.cfm?id=42515

Benjamin, C., Babcock, D., Yunus, N., & Kincaid, J. (1990, April). Knowledge-Based Prototype for Improving Scheduling Productivity. *Journal of Computing in Civil Engineering, 4*(2), 124-137. doi:10.1061/(ASCE)0887-3801(1990)4:2(124)

Black, P. E. (2007, October 18). *Undirected Graph.* (P. E. Black, Ed.) Retrieved July 17, 2013, from Dictionary of Algorithms and Data Structures: http://www.nist.gov/dads/HTML/undirectedGraph.html

Black, P. E. (2008, November 20). *Directed Graph.* (P. E. Black, Ed.) Retrieved July 17, 2013, from Dictionary of Algorithms and Data Structures: http://www.nist.gov/dads/HTML/directedGraph.html

Black, P. E., & Tanenbaum, P. J. (2012, December 26). *Graph.* (P. E. Black, Ed.) Retrieved July 17, 2013, from Dictionary of Algorithms and Data Structures: http://www.nist.gov/dads/HTML/graph.html

Büchmann-Slorup, R., & Andersson, N. (2010). BIM-Based Scheduling of Construction: A Comparative Analysis of Prevailing and BIM-Based Scheduling Processes. *Applications of IT in the AEC Industry: Proceeding of the 27th International Conference - CIB W78* (p. 113). Cairo, Egypt: Virginia Tech. Retrieved from http://orbit.dtu.dk/en/publications/bimbased-scheduling-of-construction(80dc0b89-6638-4649-bafa-513bdd256991).html

buildingSMART. (2011, November 17). *IFC2x3 Final Documnetation.* Retrieved August 20, 2013, from buildingSMART, International Home of openBIM: http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/index.htm

buildingSMART. (2013). *buildingSMART, International Home of openBIM*. Retrieved June 22, 2013, from http://www.buildingsmart-tech.org/

Chan, W.-T., Chua, D. K., & Kannan, G. (1996). Construction Resource Scheduling with Genetic Algorithms. *Journal of Construction Engineering and Management, 122*(2), 125-132. doi:10.1061/(ASCE)0733-9364(1996)122:2(125)

Comparison of Project Management Software. (2012, April 5). Retrieved from Wikipedia, the Free Encyclopedia: http://en.wikipedia.org/wiki/List_of_project_management_software

CRC Construction Innovation. (2007). *Adopting BIM for Facilities Management, Solutions for Managing the Sydney Opera House.* Brisbane, Australia: Cooperative Research Center. Retrieved May 16, 2013, from http://www.construction-innovation.info/images/CRC_Dig_Model_Book_20070402_v2.pdf

Damnjanovic, I., Faghihi, V., Scott, C., McTigue, E., & Reinschmidt, K. (2013, April). Educational Prediction Markets: A Construction Project Management Case Study. *Journal of Professional Issues in Engineering Education and Practice, 139*(2), 134-138. doi:10.1061/(ASCE)EI.1943-5541.0000127

Davis, L. (1985). Job Shop Scheduling with Genetic Algorithms. In J. J. Grefenstette (Ed.), *International Conference on Genetic Algorithms and their Applications* (pp. 136-140). Pittsburgh: Lawrence Erlbaum Associates.

Dawood, N., & Sriprasert, E. (2006). Construction Scheduling Using Multi-constraint and Genetic Algorithms Approach. *Construction Management and Economics, 24*(1), 19-30. doi:10.1080/01446190500310486

Durkin, J. (2002). History and applications. In C. T. Leondes, & C. T. Leondes (Ed.), *Expert Systems: The Technology of Knowledge Management and Decision Making for the 21st Century* (Vol. I, pp. 1-22). USA: Academic Press. doi:10.1016/B978-012443880-4/50045-4

Echeverry, D., Ibbs, C., & Kim, S. (1991, March). Sequencing Knowledge for Construction Scheduling. *Journal of Construction Engineering and Management, 117*(1), 118-130. doi:10.1061/(ASCE)0733-9364(1991)117:1(118)

Eppinger, S. D., & Browning, T. R. (2012). *Design Structure Matrix Methods and Applications.* MIT Press. Retrieved July 18, 2013, from http://mitpress.mit.edu/books/design-structure-matrix-methods-and-applications

Faghihi, V., Reinschmidt, K. F., & Kang, J. (2014). Automation in Construction Scheduling: a Review of the Literature. *Journal of Computing in Civil Engineering, Under-review*.

Faghihi, V., Reinschmidt, K. F., & Kang, J. (2014). Construction Scheduling Using Genetic Algorithm Based on BIM. *Expert Systems with Applications, Under-review*.

Faghihi, V., Reinschmidt, K. F., & Kang, J. (2014). Extended Genetic Algorithm for Optimized BIM-based Construction Scheduling. *Computer Aided Civil and Infrastructure Engineering, Under-review*, Wiley.

Faghihi, V., Reinschmidt, K. F., & Kang, J. (2014). Mathematical Framework for Spatial Relationships of the Project 3D Elements. *Automation in Construction, Under-review*.

Faghihi, V., Reinschmidt, K. F., & Kang, J. (2014). Objective-driven and Pareto Front Analysis: Optimizing Time, Cost, and Job-site Movements. *Journal of Construction Engineering and Management, Under-review*.

Farley, B., & Clark, W. A. (1954, September). Simulation of Self-organizing Systems by Digital Computer. *Information Theory, Transactions of the IRE Professional Group on, 4*(4), 76-84. doi:10.1109/TIT.1954.1057468

Feng, C., Liu, L., & Burns, S. (1997, July). Using Genetic Algorithms to Solve Construction Time-Cost Trade-Off Problems. *Journal of Computing in Civil Engineering, 11*(3), 184-189. doi:10.1061/(ASCE)0887-3801(1997)11:3(184)

Fenves, S. J., Flemming, U., Hendrickson, C., Maher, M. L., & Schmitt, G. (1990, January). Integrated Software Environment for Building Design and Construction. *Computer-Aided Design, 22*(1), 27-36. doi:10.1016/0010-4485(90)90026-9

Firat, C. E., Arditi, D., Hamalainen, J.-P., Stenstrand, J., & Kiiras, J. (2010). Quantity Take-Off In Model-Based Systems. *Proceedings of the CIB W78 2010: 27th International Conference.* Cairo, Egypt. Retrieved from http://itc.scix.net/cgi-bin/works/Show?w78-2010-112

Firat, C. E., Kiiras, J., & Huovinen, P. (2008). A Building Construction Information Model for Managing Projects Virtually. *The 5th International Conference on Innovation in Architecture, Engineering and Construction.* Antalya, Turkey.

Firat, C. E., Kiiras, J., & Huovinen, P. (2008). Solving Fundamental Problems in Model Based, Semi-Automated Building Project Scheduling. *The 5th International Conference on Cybernetics and Information Technologies, Systems and Applications.* Orlando, Florida, USA.

Firat, C. E., Kiiras, J., Kähkönen, K., & Huovinen, P. (2007). Model Based Scheduling in Building Projects – Is It Oxymoron? *24th CIB W78 Conference* (pp. 337-343). Maribor, Slovenia: CIB.

Firat, C., Arditi, D., Hämäläinen, J., & Kiiras, J. (2009). Extended Model-based Master Scheduling for Building Projects Using Advanced Line of Balance. *26th CIB W78 Conference - Managing IT in Construction.* Istanbul ,Turkey: CIB. Retrieved from http://itc.scix.net/cgi-bin/works/Show?w78-2009-1-29

Fischer, M., & Aalami, F. (1996, December 1). Scheduling with Computer-Interpretable Construction Method Models. *Journal of Construction Engineering and Management, 122*(4), 337-347. doi:10.1061/(ASCE)0733-9364(1996)122:4(337)

Fischer, M., & Tatum, C. (1997, September 1). Characteristics of Design-Relevant Constructability Knowledge. *Journal of Construction Engineering and Management, 123*(3), 253-260. doi:10.1061/(ASCE)0733-9364(1997)123:3(253)

Fischer, M., Aalami, F., & O'Brien Evans, M. (1994). Model-Based Constructibility Analysis: The MOCA System. *CIB W78 Workshop on Computer Integrated Construction.* Helsinki, Finland: CIB.

Golparvar-Fard, M., Peña-Mora, F., & Savarese, S. (2009, June). D4AR- A 4-Dimensional Augmented Reality Model for Automating Construction Progress Data Collection, Processing and Communication. *Journal of Information Technology in Construction (ITcon), 14*(Special Issue Next Generation Construction IT: Technology Foresight, Future Studies, Roadmapping, and

Scenario Planning), 129-153. Retrieved 06 16, 2013, from http://www.itcon.org/2009/13

Golparvar-Fard, M., Peña-Mora, F., Arboleda, C. A., & Lee, S. H. (2009, Nov./Dec.). Visualization of Construction Progress Monitoring with 4D Simulation Model Overlaid on Time-lapsed Photographs. *Journal of Computing in Civil Engineering, 23*(6), 391-404. doi:10.1061/(ASCE)0887-3801(2009)23:6(391)

Gonçalves, J., Mendesb, J., & Resendec, M. (2008, September 16). A Genetic Algorithm for the Resource Constrained Multi-project Scheduling Problem. *European Journal of Operational Research, 189*(3), 1171-1190. doi:10.1016/j.ejor.2006.06.074

Hashemi Golpayegani, S. A., & Emamizadeh, B. (2007, November). Designing Work Breakdown Structures Using Modular Neural Networks. *Decision Support Systems, 44*(1), 202-222. doi:10.1016/j.dss.2007.03.013

Hashemi Golpayegani, S. A., & Parvaresh, F. (2011, August). The Logical Precedence Network Planning of Projects, Considering the Finish-to-Start (FS) Relations, Using Neural Networks. *The International Journal of Advanced Manufacturing Technology, 55*(9-12), 1123-1133. doi:10.1007/s00170-010-3125-1

Haupt, R. L., & Haupt, S. E. (2004). The Binary Genetic Algorithm. In *Practical Genetic Algorithms* (2nd ed., pp. 27-50). John Wiley & Sons, Inc. Retrieved June 6, 2013, from http://www.ie.itcr.ac.cr/rpereira/mat_ant/Genetic%20Algorithms/ch2.pdf

Hebb, D. O. (1949). *The Organization of Behavior.* New York: Wiley & Sons.

Hendrickson, C., Martinelli, D., & Rehak, D. (1987, June). Hierarchical Rule-Based Activity Duration Estimation. *Journal of Construction Engineering and Management, 113*(2), 288-301. doi:10.1061/(ASCE)0733-9364(1987)113:2(288)

Hendrickson, C., Zozaya-Gorostiza, C., Rehak, D., Baracco-Miller, E., & Lim, P. (1986). *An Expert System Architecture for Construction Planning.* Carnegie Mellon University, Department of Civil and Environmental Engineering . Pittsburgh, PA: CMU Engineering Design Research Center. Retrieved from http://repository.cmu.edu/cgi/viewcontent.cgi?article=1033&context=cee

Hendrickson, C., Zozaya-Gorostiza, C., Rehak, D., Baracco-Miller, E., & Lim, P. (1987, October). An Expert System for Construction Planning. *Journal of Computing in Civil Engineering, 1*(4), 253-269. doi:10.1061/(ASCE)0887-3801(1987)1:4(253)

Holland, J. H. (1992). *Adaptation In Natural And Artificial Systems.* (2nd, Ed.) MIT Press. Retrieved June 23, 2013, from http://mitpress.mit.edu/books/adaptation-natural-and-artificial-systems

Hooshyar, B., Tahmani, A., & Shenasa, M. (2008). 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). Hong Kong: IEEE. doi:10.1109/CEC.2008.4631214

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

Jackson, P. (1999). *Introduction To Expert Systems* (3rd ed.). USA: Addison-Wesley.

Jaśkowski, P., & Sobotka, A. (2006). Scheduling Construction Projects Using Evolutionary Algorithm. *Journal of Construction Engineering and Management, 132*(8), 861-870. doi:10.1061/(ASCE)0733-9364(2006)132:8(861)

Kanda, A. (2011). Matrix Multiplication Procedures for Adjacency Matrices. In A. Kanda, *Project Management - A Life Cycle Approach* (pp. 75-77). PHI Learning Pvt. Ltd. Retrieved June 23, 2013

Karim, A., & Adeli, H. (1999, September). CONSCOM: An OO Construction Scheduling and Change Management System. *Journal of Construction Engineering and Management, 125*(5), 368-376. doi:10.1061/(ASCE)0733-9364(1999)125:5(368)

Karim, A., & Adeli, H. (1999, September). OO Information Model for Construction Project Management. *Journal of Construction Engineering and Management, 125*(5), 361-367. doi:10.1061/(ASCE)0733-9364(1999)125:5(361)

Kartam, N., & Levitt, R. (1990, April). Intelligent Planning of Construction Projects. *Journal of Computing in Civil Engineering, 4*(2), 155-176. doi:10.1061/(ASCE)0887-3801(1990)4:2(155)

Kartam, N., Levitt, R., & Wilkins, D. (1991). Extending Artificial Intelligence Techniques for Hierarchical Planning. *Journal of Computing in Civil Engineering, 5*(4), 464-477. doi:10.1061/(ASCE)0887-3801(1991)5:4(464)

Kataoka, M. (2008, August). Automated Generation of Construction Plans from Primitive Geometries. *Journal of Construction Engineering and Management, 134*(8), 592-600. doi:10.1061/(ASCE)0733-9364(2008)134:8(592)

Kenett, D. Y., Tumminello, M., Madi, A., Gur-Gershgoren, G., Mantegna, R. N., & Ben-Jacob, E. (2010). Dominating Clasp of the Financial Sector Revealed by Partial Correlation Analysis of the Stock Market. *PLoS ONE, 5*(12). doi:10.1371/journal.pone.0015032

Kim, A., Hessami, A., Faghihi, V., & Ford, D. (2012). Designing Perpetual Sustainability Improvement Programs for Built Infrastructures. *The 30th International Conference of the System Dynamics Society.* St. Gallen, Switzerland: System Dynamics Society. Retrieved from http://www.systemdynamics.org/conferences/2012/proceed/papers/P1278.pdf

Knowledge-Based Project Planning. (2002). In R. Dieng-Kuntz, & N. Matta (Eds.), *Knowledge Management and Organizational Memories* (1st ed., pp. 125-134). Springer US. doi:10.1007/978-1-4615-0947-9_11

Kolodner, J. L. (1992, March 1). An Introduction to Case-Based Reasoning. *Artificial Intelligence Review, 6*(1), 3-34. doi:10.1007/BF00155578

Konaka, A., Coit, D. W., & Smith, A. E. (2006, September). Multi-objective Optimization Using Genetic Algorithms: A Tutorial. *Reliability Engineering and System Safety, 91*(9), 992-1007. doi:10.1016/j.ress.2005.11.018

König, M., Beucke, K., & Tauscher, E. (2006). Management and Evaluation of Alternative Construction Tasks. In L. Soibelman, & B. Akinci (Ed.), *The 11th*

International Conference on Computing in Civil and Building Engineering (pp. 2247-254). Montreal, Canada: ASCE.

Koo, B., Fischer, M., & Kunz, J. (2007, November). Formalization of Construction Sequencing Rationale and Classification Mechanism to Support Rapid Generation of Sequencing Alternatives. *Journal of Computing in Civil Engineering, 21*(6), 423-433. doi:10.1061/(ASCE)0887-3801(2007)21:6(423)

Kuqi, K., Eveleigh, T., Holzer, T., & Sarkani, S. (2012). Using Design Structure Matrix for Improving Electronic Medical Record Usability. *Systems Conference (SysCon)* (pp. 1-6). Vancouver, British Columbia, Canada: IEEE International. doi:10.1109/SysCon.2012.6189484

Landry, M., Malouin, J.-L., & Oral, M. (1983, November). Model Validation in Operations Research. *European Journal of Operational Research, 14*(3), 207-220. doi:10.1016/0377-2217(83)90257-6

Levitt, R. E., Kartam, N. A., & Kunz, J. C. (1988, March). Artificial Intelligence Techniques for Generating Construction Project Plans. *Journal of Construction Engineering and Management, 114*(3), 329-343. doi:10.1061/(ASCE)0733-9364(1988)114:3(329)

Lin, L., Hao, X.-C., Gen, M., & Jo, J.-B. (2012, December). Network Modeling and Evolutionary Optimization for Scheduling in Manufacturing. *Journal of Intelligent Manufacturing, 23*(6), 2237-2253. doi:10.1007/s10845-011-0569-6

Lindemann, U. (2009). *Design Structure Matrix (DSM).* Retrieved July 18, 2013, from DSMweb: http://www.dsmweb.org/en/understand-dsm/tutorials-overview/descripton-design-structre.html

Madi, A., Kenett, D. Y., Bransburg-Zabary, S., Merbl, Y., Quintana, F. J., Boccaletti, S., . . . Ben-Jacob, E. (2011, March 29). Analyses of Antigen Dependency Networks Unveil Immune System Reorganization between Birth and Adulthood. *Focus Issue: Mesoscales In Complex Networks, 21*(1). doi:10.1063/1.3543800

McCulloch, W. S., & Pitts, W. (1943, December). A Logical Calculus of the Ideas Immanent in Nervous Activity. *The Bulletin of Mathematical Biophysics, 5*(4), 115-133. doi:10.1007/BF02478259

McGartland, M. R., & Hendrickson, C. T. (1985, September). Expert Systems for Construction Project Monitoring. *Journal of Construction Engineering and Management, 111*(3), 293-307. doi:10.1061/(ASCE)0733-9364(1985)111:3(293)

McKinneya, K., & Fischer, M. (1998, September). Generating, Evaluating and Visualizing Construction Schedules with CAD Tools. *Automation in Construction, 7*(6), 433-447. doi:10.1016/S0926-5805(98)00053-3

Mikulakova, E., König, M., Tauscher, E., & K., B. (2008). Case-based Reasoning for Construction Tasks. In A. Ren, Z. Ma, & L. X. (Ed.), *The 12th International Conference on Computing in Civil and Building Engineering*, (p. 199). Beijing, China.

Mikulakova, E., König, M., Tauscher, E., & Karl, B. (2010, November). Knowledge-based Schedule Generation and Evaluation. (W. Huhnt, Ed.) *Advanced Engineering Informatics, 24*(4), 389-403. doi:10.1016/j.aei.2010.06.010

Mitchell, M. (1996). *An Introduction to Genetic Algorithms.* Cambridge, MA: MIT Press.

Mohammadi, G. (2011, October). Using Genetic Algorithms to Solve Industrial Time-cost Trade-off Problems. *Indian Journal of Science and Technology, 4*(10), 1273-1278.

Mohan, S. (1990, March). Expert Systems Applications in Construction Management and Engineering. *Journal of Construction Engineering and Management, 116*(1), 87-99. doi:10.1061/(ASCE)0733-9364(1990)116:1(87)

Morad, A., & Beliveau, Y. (1991, March). Knowledge-Based Planning System. *Journal of Construction Engineering and Management, 117*(1), 1-12. doi:10.1061/(ASCE)0733-9364(1991)117:1(1)

Moselhi, O., & Nicholas, M. (1990, June). Hybrid Expert System for Construction Planning and Scheduling. *Journal of Construction Engineering and Management, 116*(2), 221-238. doi:10.1061/(ASCE)0733-9364(1990)116:2(221)

Mukkamalla, S., & Muñoz-Avila, H. (2002). Case Acquisition in a Project Planning Environment. *6th European Conference. 2416*, pp. 264-277. Aberdeen, Scotland, UK: Springer Berlin Heidelberg. doi:10.1007/3-540-46119-1_20

Muñoz-Avila, H., Aha, D. W., Nau, D. S., Breslow, L. A., Weber, R., & Yamal, F. (2001). SiN: Integrating Case-based Reasoning with Task Decomposition.

Seventeenth International Joint Conference on Artificial Intelligence. Seattle, Washington, USA.

Murata, T., Ishibuchi, H., & Tanaka, H. (1996, September). Multi-objective Genetic Algorithm and Its Applications to Flowshop Scheduling. *Computers & Industrial Engineering, 30*(4), 957-968. doi:10.1016/0360-8352(96)00045-9

Navinchandra, D., Sriram, D., & Logcher, R. (1988, July). GHOST: Project Network Generator. *Journal of Computing in Civil Engineering, 2*(3), 239-254. doi:10.1061/(ASCE)0887-3801(1988)2:3(239)

Newell, A., & Simon, H. A. (1972). *Human Problem Solving.* Prentice-Hall.

NIST ITL. (2012). 5.1.1. What is experimental design? In *NIST/SEMATECH e-Handbook of Statistical Methods.* National Institute of Standards and Technology, NIST. Retrieved 06 01, 2013, from http://www.itl.nist.gov/div898/handbook/pri/section1/pri11.htm

Phelan, R., Radjy, F., Haas, C., & and Hendrickson, C. (1990, March). Computer-Aided Concrete-Placement Optimization. *Journal of Construction Engineering and Management, 116*(1), 172-187. doi:10.1061/(ASCE)0733-9364(1990)116:1(172)

Sabuncuoglu, I. (1998). Scheduling with Neural Networks: A Review of the Literature and New Research Directtions. *Production Planning & Control: The Management of Operations, 9*(1), 2-12. doi:10.1080/095372898234460

Schirmer, A. (2000, April). Case-based Reasoning and Improved Adaptive Search for Project Scheduling. *Naval Research Logistics (NRL), 47*(3), 201-222. doi:10.1002/(SICI)1520-6750(200004)47:3<201::AID-NAV2>3.0.CO;2-L

Senouci, A., & Al-Derham, H. R. (2008, December). Genetic Algorithm-based Multi-objective Model for Scheduling of Linear Construction Projects. *Advances in Engineering Software, 39*(12), 1023-1028. doi:10.1016/j.advengsoft.2007.08.002

Shaked, O., & Warszawski, A. (1992, September). CONSCHED: Expert System for Scheduling of Modular Construction Projects. *Journal of Construction Engineering and Management, 188*(3), 488-506. doi:10.1061/(ASCE)0733-9364(1992)118:3(488)

Shaked, O., & Warszawski, A. (1995, June). Knowledge-Based System for Construction Planning of High-Rise Buildings. *Journal of Construction Engineering and Management, 121*(2), 172-182. doi:10.1061/(ASCE)0733-9364(1995)121:2(172)

Shukla, S. K. (2009, August 3). *Adjacency-matrix Representation.* (P. E. Black, Ed.) Retrieved July 17, 2013, from Dictionary of Algorithms and Data Structures: http://www.nist.gov/dads/HTML/adjacencyMatrixRep.html

Sivaraj, R., & Ravichandran, T. (2011). A Review of Selection Methods in Genetic Algorithm. *International Journal of Engineering Science and Technology (IJEST), 3*(5), 3792-3797. Retrieved June 23, 2013, from http://www.ijest.info/docs/IJEST11-03-05-190.pdf

Steward, D. V. (1962, October). On an Approach to Techniques for the Analysis of the Structure of Large Systems of Equations. *SIAM Review, 4*(4), 321-342. Retrieved July 19, 2013, from http://www.jstor.org/stable/2027903

Tauscher, E., Mikulakova, E., Beucke, K., & König, M. (2009). Automated Generation of Construction Schedules Based on the IFC Object Model. *International*

Workshop on Computing in Civil Engineering (pp. 666-675). Austin, Texas, United States: ASCE. doi:10.1061/41052(346)66

Tauscher, E., Mikulakova, E., König, M., & Beucke, K. (2007). Generating Construction Schedules with Case-Based Reasoning Support. In L. Soibelman, & B. Akinci (Ed.), *Proceedings of the International Workshop on Computing in Civil Engineering* (pp. 119-126). Pittsburgh, Pennsylvania : ASCE. doi:10.1061/40937(261)15

Toklu, Y. C. (2002). Application of Genetic Algorithms to Construction Scheduling With or Without Resource Constraints. *Canadian Journal of Civil Engineering, 29*(3), 421-429.

Tulke, J., Nour, M., & Beucke, K. (2008). Decomposition of BIM Objects for Scheduling and 4D Simulation. In R. Scherer, & A. Zarli (Eds.), *eWork and eBusiness in Architecture, Engineering and Construction* (pp. 653-660). London, UK: CRC Press. doi:10.1201/9780203883327.ch73

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

Vriesa, B. d., & Harink, J. M. (2007, January). Generation of a Construction Planning from a 3D CAD Model. *Automation in Construction, 16*(1), 13-18. doi:10.1016/j.autcon.2005.10.010

Wall, M. B. (1996). *A Genetic Algorithm for Resource-Constrained Scheduling.* MIT.

Wang, S. Q. (2001). ESSCAD: Expert System Integrating Construction Scheduling with CAD. *CIB-W78 International Conference. IT in Construction in Africa 2001.* Mpumalanga, South Africa.: CIB.

Ward Systems Group, Inc. (2007, June 29). *Genetic Algorithms and What They Can Do For You*. Retrieved from Advanced Neural Network and Genetic Algorithm Software: http://www.wardsystems.com/ga101.asp

Watson, I., & Marir, F. (1994). Case-Based Reasoning: A Review. *The Knowledge Engineering Review, 9*(4), 327-354. doi:10.1017/S0269888900007098

Weldu, Y. W., & Knapp, G. M. (2012). Automated Generation of 4D Building Information Models through Spatial Reasoning. *Construction Research Congress* (pp. 612-621). West Lafayette, Indiana, United States: ASCE. doi:10.1061/9780784412329.062

Xu, K. (2006). *Case-Based Task Decomposition with Incomplete Domain Descriptions.* Lehigh University, Department of Computer Science & Engineering . Bethlehem, PA: Lehigh University. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.138.3038&rep=rep1&type=pdf

Xu, K., & Muñoz-Avila, H. (2003). CBM-Gen+: An Algorithm for Reducing Case Base Inconsistencies in Hierarchical and Incomplete Domains. In K. D. Ashley, & D. Bridge (Ed.), *5th International Conference on Case-Based Reasoning* (pp. 665-678). Trondheim, Norway: Springer Berlin Heidelberg. doi:10.1007/3-540-45006-8_50

Xu, K., & Muñoz-Avila, H. (2003). Maintaining Consistency in Project Planning Reuse. In K. D. Ashley, & D. Bridge (Ed.), *5th International Conference on Case-Based Reasoning* (pp. 679-690). Trondheim, Norway: Springer Berlin Heidelberg. doi:10.1007/3-540-45006-8_51

Xu, K., & Muñoz-Avila, H. (2004). CaBMA: Case-Based Project Management Assistant. *The Sixteenth Innovative Applications of Artificial Intelligence Conference* (pp. 931-936). San Jose, California: The AAAI Press. Retrieved from http://www.aaai.org/Library/IAAI/iaai04contents.php

Xu, K., & Muñoz-Avila, H. (2005). A Domain-Independent System for Case-Based Task Decomposition without Domain Theories. *The 20th National Conference on Artificial Intelligence* (pp. 234-240). Pittsburgh, Pennsylvania: The AAAI Press. Retrieved from http://www.aaai.org/Library/AAAI/aaai05contents.php

Xu, K., & Muñoz-Avila, H. (2008, May). CaBMA: a Case-based Reasoning System for Capturing, Refining, and Reusing Project Plans. *Knowledge and Information Systems, 15*(2), 215-232. doi:10.1007/s10115-007-0077-3

Zheng, D. X., Ng, S. T., & Kumaraswamy, M. M. (2002). Applying Genetic Algorithm Techniques for Time-cost Optimization. In D. Greenwood (Ed.), *18th Annual ARCOM Conference. 2*, pp. 801-810. University of Northumbria, Association of Researchers in Construction Management.

Zheng, D., Ng, S., & Kumaraswamy, M. (2004, April). Applying a Genetic Algorithm-Based Multiobjective Approach for Time-Cost Optimization. *Journal of*

Construction Engineering and Management, 130(2), 168-176. doi:10.1061/(ASCE)0733-9364(2004)130:2(168)

Zheng, D., Ng, S., & Kumaraswamy, M. (2005, January). Applying Pareto Ranking and Niche Formation to Genetic Algorithm-Based Multiobjective Time–Cost Optimization. *Journal of Construction Engineering and Management, 131*(1), 81-91. doi:http://dx.doi.org/10.1061/(ASCE)0733-9364(2005)131:1(81)

Zozaya-Gorostiza, C., Hendrickson, C., & Rehak, D. R. (1989). Knowledge-Based Construction Project Planning. In R. J. Bard (Ed.), *Excellence in the Constructed Project* (pp. 217-222). San Francisco, California: ASCE. Retrieved from http://cedb.asce.org/cgi/WWWdisplay.cgi?63892

Zozaya-Gorostiza, C., Hendrickson, C., Rehak, D. R., & Lim, P. (1988). Construction Planex : A Knowledge Intensive Planner for Construction Projects. *The 5th ISARC* (pp. 511-520). Tokyo, Japan: International Association for Automation and Robotics in Construction. Retrieved from http://www.iaarc.org/publications/fulltext/Construction_planex_A_knowledge_intensive_planner_for_construction_projects.PDF

APPENDIX

Simple Step-by-Step Example

Considering the example in Figure 8, with five genomes per population, 20% of population as elite members, 10% chance of mutation, and the tentative duration of 5±20% time-unit (i.e. either 4, 5, or 6 time-units), the first population generated by the described GA is as shown in Table 8. This table consists of two main columns: the left column containing the genome number, constructability score, and schedule duration, and the right column containing the genomes. Since there are eight elements in the model shown in Figure 8, the constructability scores (in form of percentage of the number of elements that have been scheduled based on MoCC) are either 0%, 12.5%, 25%, 37.5%, 50%, 62.5%, 75%, 87.5%, or 100%.

The genomes in the right column of the Table 8 are basically the Matrix of Genome or MoG (defined in Equation 2) that are shown in the form of Equation 3, but the rows of the matrix are put consecutively one after the other. The dotted lines are the dividers between the rows of the MoG. Because there are eight elements in the 3D model, all of the genomes have eight sections that are divided from each other by the dotted line, as seen in Table 8. Then, each of these eight sections in the genomes contains the same number of zeros and ones as the duration of the genome. For instance, the first four digits of the genome number 0 in the first population, as seen in Table 8 (i.e. 0100) is the installation schedule for the first element of the 3D model, column 1:#143, as shown in Figure 8. These four digits define the installation of the element number 1:#143 to in the second time-unit as its second digit is 1 and the rest are 0s.

Table 8- Population Number 1

0:62.5%:4	0100 0100 0100 0001 0001 0010 0000 0000
1:50%:6	010000 000001 000010 000100 000000 000000 000000 000000
2:37.5%:5	00100 00010 01000 00000 00000 00000 00000 00000
3:12.5%:5	00001 00000 00000 00000 00000 00000 00000 00000
4:0%:4	0000 0000 0000 0000 0000 0000 0000 0000

As mentioned before, the genomes can be represented as the MoG too. The MoG for the first genome of the first population is shown in Figure 36.

$$MoG\ for\ Genome\ \#\ 0 = \begin{array}{r c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{r} 1{:}\ \#143 \\ 2{:}\ \#209 \\ 3{:}\ \#239 \\ 4{:}\ \#264 \\ 5{:}\ \#375 \\ 6{:}\ \#510 \\ 7{:}\ \#623 \\ 8{:}\ \#736 \end{array} & \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Figure 36- MoG for Genome # 0

Given the genome in the form of the matrix, it can be easily interpreted and the scheduling of the installments are much more understandable. As illustrated by Figure 36, elements number 1, 2, and 3 are scheduled to be installed in the second time-unit while the elements number 4 and 5 are planned to be done in the fourth time-unit. Similarly, the element number 6 is scheduled for the third time-unit, but the remaining two elements, 7 and 8, are not planned to be installed in this sequencing order defined by this genome.

Clearly, the schedule in Figure 36 has four time-units as represented by four columns in its matrix. The calculated constructability score for this genome (construction schedule of the 3D model) as shown in Table 8 is 62.5%. This score means that 62.5% (5 elements out of 8) are scheduled for installation correctly based on the constructability constraints detected from the model as MoCC (see Figure 8). These five elements are columns 1:#143, 2:#209, 3:#239, and 4:#26 and the beam 5:#375. The first four columns do not have any installation constraints detected in the MoCC, and the only beam has columns 1 and 2 as its supporting constraints being scheduled for installation as of fourth time-unit (columns are scheduled in the second time-unit and the beam is scheduled in the fourth time-unit).

All five genomes in the first population were generated based on the random genome generation function described in Chapter IV, section “Genome Creation”. Performing all the inherent Genetic Algorithm functions (i.e. Elite selection, Crossover function, Mutation function, and Selection function), the next populations will be generated as shown in Table 9, Table 10, Table 11, Table 12, and Table 13, showing from second population to the sixth respectively.

Table 9- Population Number 2

0:87.5%:5	01000	01000	01000	01000	00100	10000	00100	00001
1:75%:4	1000	1000	0100	0001	0001	0010	1000	0010
2:75%:4	0010	0100	0100	0001	0001	0001	0100	0001
3:62.5%:4	0100	0100	0100	0001	0001	0010	0000	0000
4:50%:5	01000	01000	01000	00100	10000	00100	01000	10000

Table 10- Population Number 3

0:87.5%:5	01000	01000	01000	01000	00100	10000	00100	00001
1:75%:4	1000	1000	0100	0001	0001	0001	1000	0001
2:75%:5	01000	01000	01000	01000	00001	10000	00100	10000
3:62.5%:4	0100	0001	0100	0001	0001	0010	1000	0010
4:62.5%:4	0010	1000	0010	0001	0100	0100	0100	0001

Table 11- Population Number 4

0:87.5%:5	01000	01000	01000	01000	00100	10000	00100	00001
1:87.5%:5	01000	01000	01000	01000	00100	00001	00100	10000
2:75%:5	01000	01000	01000	01000	00001	10000	00100	10000
3:75%:5	01000	01000	01000	01000	00001	10000	00100	01000
4:50%:5	10000	00010	00100	01000	00010	00100	01000	01000

Table 12- Population Number 5

0:87.5%:5	01000	01000	01000	01000	00100	10000	00100	00001
1:87.5%:5	01000	01000	01000	01000	00100	00001	00100	10000
2:50%:4	0010	0100	0010	0100	0010	0100	0100	1000

3:50%:4	1000	0010	0100	0001	0100	0010	0010	0100
4:50%:5	10000	00010	00100	01000	00010	00100	01000	01000

Table 13- Population Number 6

0:100%:5	10000	01000	01000	01000	00100	00001	00100	00100
1:87.5%:5	01000	01000	01000	01000	00100	10000	00100	00001
2:75%:6	001000	000100	010000	001000	001000	100000	000100	000010
3:62.5%:5	010000	001000	000100	000100	001000	000100	000010	100000
4:50%:5	00100	00010	00100	01000	00010	00100	01000	01000

As seen in Table 13, the first genome with the complete score of 100% for its constructability is generated. After this achievement, the proposed algorithm will continue until all the genomes in the population reach the complete 100% score for their constructability. The final population can be seen in Table 14, as the 20th population in this simple run that all the genomes have reach the 100% score. The algorithm has now ended.

Table 14- Population Number 20

0:100%:6	001000	010000	010000	010000	000100	000010	000100	000100
1:100%:6	100000	010000	010000	010000	000100	000010	000100	000100
2:100%:6	10000	01000	01000	01000	00100	00001	00100	00100
3:100%:5	10000	01000	10000	01000	00100	00001	00100	00100
4:100%:6	100000	010000	010000	010000	000001	000010	001000	001000