

Expert System for Construction Scheduling Decision Support Based on Travelling Salesman Problem

Tulio Sulbaran, Ph.D

The University of Southern Mississippi
Hattiesburg, Mississippi

Feroz Ahmed

The University of Southern Mississippi
Hattiesburg, Mississippi

Construction of a project includes the physical activities requiring resources (such as labor, materials and equipment), but more importantly it also includes the entire scope of activities from conception to realization to successfully execute the project. Successful execution of the project involves the effective planning, scheduling and management of resources. Planning and scheduling of activities and resources is one of the most important tasks of a construction project. A properly prepared construction schedule provides an opportunity to efficiently manage the project and to convey to stakeholders how the construction company is planning to meet its obligations. However, preparing good schedules is a time-consuming process. New innovative technologies might be the key to reduce the time needed to prepare a good schedule. The purpose of this paper is to present an innovative idea to automatically prepare draft construction schedules from Building Information Models that will allow the scheduler (and/or Project Manager) to focus on the construction aspects at much higher and/or deeper level. The research conducted for this paper follows a one-shot experimental case study methodology. An expert system was developed in Jess incorporating the Travelling Salesman Problem (often called TSP) algorithm. This expert system was provided with information obtained from Building Information Model of few simple test case projects. The expert system was able to schedule the activities in each of the test case projects. A subject-matter expert reviewed the schedules and found only one error. This is a very important first step to support the decisions that are made by construction scheduler and perhaps has the potential to reduce the time needed to prepare schedules and/or improve the quality of final construction schedules. It also shows that additional research in this area might be very beneficial to construction.

Keywords: Construction scheduling, decision making/Support, building information modeling, travelling salesman problem

Introduction

The research presented in this paper is grounded mainly in four areas: 1- Traveling Salesman Problem, 2- Project Scheduling, 3- Building information model, and 4- Jess Expert System (as shown in Figure 1). Thus, a short synopsis of each area is presented herein prior to providing the explanation of the expert system developed during this research for construction scheduling decision support.

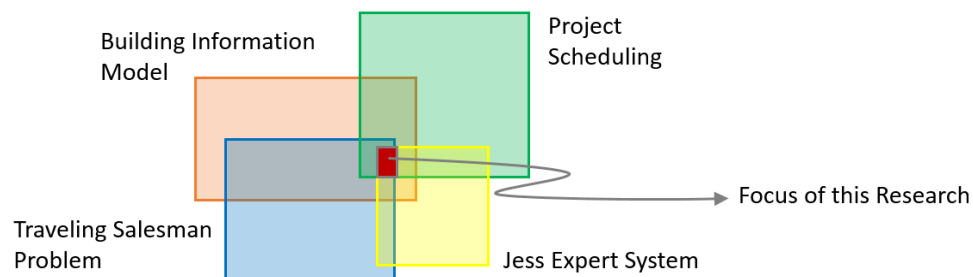


Figure 1. Research Foundation

Travelling Salesman Problem (TSP)

Travelling Salesman Problem (TSP) is a mathematical problem that involves a salesman who must make a tour to a number of cities using the shortest path available and visit each city exactly once and return to the original starting city. It is a classic algorithmic problem in the field of computer science that focuses on optimization. It is also considered as one of the most intensively studied problems in computational mathematics (Weisstein., 2016). The TSP is typical of a large class of "hard" optimization problems that have intrigued mathematicians and computer scientists for years. There are two types of a TSP problem: a) The symmetric TSP, where the distance between two cities is the same in each opposite direction, b) the asymmetric TSP, where the paths may not exist in both directions and the distances might be different (Hahsler , Hornik, 2006).

This optimization algorithm has several applications such as planning, logistics, the manufacture of microchips and DNA sequencing, vehicle routing, order-picking from warehouses (Matai, R., Singh, and Mittal, 2010) are just a few to mention. An efficient solution to TSP reduces production costs for the manufacturer by increasing the efficiency and reducing the delays in the projects (Klansek, 2011). TSP is also used in astronomy, as astronomers may want to minimize the time spent moving the telescope between the sources that are under observation.

Although, some research has been done in applications of TSP in Construction management such as the one shown in Table 1, to the knowledge of the researchers no in-depth experimental research has been done in implementing TSP for construction scheduling.

Table 1 Sample Applications of TSP in Construction Management

Application	Description
Vehicle routing	The optimal solution of the TSP is used to determine for a fleet of vehicles which customers should be visited by which vehicles, and in which order each vehicle should visit its customers. (Christofides, 1985)
Handling Operations With Tower Cranes	Optimal scheduling of crane operations not only has direct cost savings, but also results in indirect cost saving by minimizing the idle time of equipment and crew on the job site as well as the downstream delays in the job process. (Zavichi, Madan, Xanthopoulos, and Oloufa, 2013)
Construction of roads	TSP solution can be of tremendous help while laying out a path to start the construction of roads connecting multiple cities. It provides the most efficient route to follow the construction and cover each and every city with minimal wastage of time and resources. (Klansek, 2011).

Project Scheduling

Project schedule is a listing of a project's milestones, activities, and deliverables, usually with intended start and finish dates (Hendrickson, C., 1998). All activities require labor, material and equipment to be executed. Therefore, in some instances a resource loaded schedule is prepared intending to match the resources of equipment, materials and labor with each of the project work tasks. In construction, the development and maintenance of the project schedule is the responsibility of a scheduler, project manager, assistant project

manager and/or in some instances a team of people, depending on the size of the project. The person responsible for creating the schedule takes the advantage of the information that is already available in the project. This information could come from cost estimation, drawings, specifications and/or building information models to prepare the best possible schedule. Poor scheduling can result in considerable waste as laborers and equipment wait for the availability of needed resources or the completion of preceding tasks. According to the Association of Construction and Development, there are several methods for construction scheduling techniques (ACA 2012). The four basic construction scheduling techniques are summarized in Table 2 (ACA 2012).

Table 2. Basic Construction Scheduling Techniques (ACA 2012)

Technique	Description/Summary
Bar chart	It graphically represents the list of activities that must be accomplished. Designates start and completion dates and also states the length of time each activity will require. It is straightforward and easy to understand which is critical when communicating with site foremen and sub-contractors
Critical path method	It is a lot more detailed than the bar graph. It shows how they are interrelated. It spells out which activities must be completed and/or started in order for other activities to be started and/or completed;
Line of balance technique	It is ideal for construction work that is repetitive in nature such as house building. This type of scheduling focuses more on allocation of resources for each activity and less on time.
Q scheduling	It is a form of construction management that addresses the sequence of activities and the relationship of those activities to each other as well as the cost of finishing the job. It takes the overall construction site into account and prevents two competing activities from occurring at the same time in the same location.

Construction scheduling methods mainly help in the process of decision making. The construction schedule also communicates means and methods, as well as planned sequences and timing for a project. The key processes that are common in all scheduling techniques include planning, controlling, and managing. Planning is the process of developing ‘what needs to be done’. Controlling is the process of keeping the project on course and making sure it is heading in the right direction. Managing is the process of communicating the progress to all parties that are involved in the project. The scheduling process provides the contractor with a more thorough and structured planning process while they review the plans and figure out the sequence for building the project. (Tang, Ahmad, Ahmed, & Lu, 2004)

Building Information Model

Building Information Modeling (BIM) is an intelligent 3D model-based process that equips architecture, engineering, and construction professionals with the insight and tools to more efficiently plan, design, construct, and manage buildings and infrastructure. It is a process involving the generation and management of digital representations of physical and functional characteristics of buildings. BIM is a digital representation of physical and functional characteristics of a facility/project. It is a shared knowledge resource for information about a facility/project forming a reliable basis for decisions during its life-cycle

(Corke, 2016). BIMs can also be seen as files which can be extracted, exchanged or networked to support decision-making regarding a building or other built asset. BIM models not only contain architectural data, but the full depth of the building information including data related to the different engineering disciplines such as the load-bearing structures, all the ducts and pipes of the different building systems and even sustainability information. All of this allow easy simulation of the project well in advance of the construction process. (Graphisoft, 2016).

Jess Expert System

Jess is a rule engine and scripting environment written entirely in Oracle's Java language (Friedman-Hill, 2003). Jess uses an enhanced version of the Rete algorithm to process rules (Bejar, 2010). Jess uses an enhanced version of the Rete algorithm to process rules. Rete algorithm is a very efficient mechanism for solving the difficult many-to-many matching problem. A normal jess program includes facts, rules, functions and template (as shown in Figure 2). The following are the definitions/description of each of them:

- Rules: A Jess *rule* is something like an if... then statement in a procedural language.
- Facts: a collection of knowledge nuggets used by the *rules*
- Template: A template describes the characteristics of the facts. Every *fact* has a *template*. A template has a name and a set of slots.
- Functions: are a combination of operations and comments executed together. *Functions* can be defined in the Jess rule language using the '*deffunction*' construct.

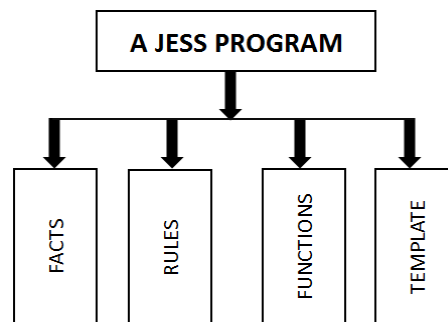


Figure 2. Structure of Jess Expert System program

Development of the Scheduling Decision Support Expert System in JESS

The Scheduling Decision Support Expert System was developed based on the TSP. The TSP was approximated in Jess language implementing functions and rules using the hill-climbing algorithm. It is important to remember that the TSP problem consists of finding a path that passes through a set of cities, in such a way that it must begin and end in the same city and should not visit a city more than once. The activities represented by 3D models in the BIM will take the place of the cities and the schedule will take the place of the path.

The templates '*deftemplate solution*' and '*deftemplate mat-dist*' are used to define the distance matrix and the solution. These templates are multi-slot templates so they can hold a list of values. The '*deftemplate solution*' has three slots, the solution path, the cost, and a slot that will be used to control the search. Figure 3 shows the Jess code for the two templates.

```

; Structure of the the Variable for Solution Path
(deftemplate solution
  (multislot path)
  (slot cost (type INTEGER))
  (slot desc)
)
;-----
; Structure of the Variable that contains the Distance between Cities
(deftemplate mat-dist
  (multislot dist)
)

```

Figure 3. Templates for Solution Path and Distance between Cities

The function '*deffunction calc-dist*' is used to assign the number of cities and the distance between the cities. Since the number of nodes and distance between the nodes might not be constant, the values are given in the form on a matrix. *Bind* function is used to allot the initial values to the variables. '*?loopmax*' is used so the system can evaluate the distances between every possible city. This function transforms the indices to get the correct position in the list representing the distance matrix. Figure 4 shows the Jess code for this function.

```

(deffunction calc-dist (?i ?j ?nciu ?m)
  (if (< ?i ?j)
    then (bind ?pm ?i) (bind ?pn ?j)
    else (bind ?pm ?j) (bind ?pn ?i)
  )

  (bind ?off 0)
  (bind ?i 1)
  (bind ?loopmax (- ?pm 1))

  (while (<= ?i ?loopmax)
    (bind ?off (+ ?off (- ?nciu ?i)))
    (bind ?i (+ ?i 1))
  )
)

```

Figure 4. Function to Calculate Distance between Cities

Figure 5 illustrates the function '*deffunction calc-cost*', which is used to calculate the accumulating distance from one city to another. This function goes through the list representing the path accumulating the distances.

```

(deffunction calc-cost (?s ?m)
  (bind ?c 0)
  (bind ?i 1)
  (bind ?loopmax (- (length$ ?s) 1))
  (while (<= ?i ?loopmax)
    (bind ?c (+ ?c (calc-dist (nth$ ?i ?s) (nth$ (+ ?i 1) ?s) (length$ ?s) ?m)))
    (bind ?i (+ ?i 1))
  )

  (bind ?c (+ ?c (calc-dist (nth$ 1 ?s) (nth$ (length$ ?s) ?s) (length$ ?s) ?m)))
)

```

Figure 5. Function to Calculate Accumulated Distance between Cities

'*(assert (num-cities))*' is used to provide the expert system the total number of cities that should be considered. The number of cities is then used by the expert system to expect the proper number of distances

between the cities. `(bind ?m (create$))` creates the distance matrix between the cities as shown in Figure 6.

```
(assert (num-cities 6))

(defrule init
  (num-cities ?x)
  =>
  ; Distance matrix between cities is created here
  (bind ?m (create$ 12.6 21.5 12.3 12.2 12.0 12.12))
)
```

Figure 6. Number of Cities and Distances

For every iteration, all possible city interchanges are generated and the best one is kept. To determine all possible exchanges the Jess inference engine was used to perform the calculations itself. Only an auxiliary fact was needed to control the generation of combinations. This fact was called `'pos'` and there are as many facts as positions and each one has one of the possible values. To generate all pairs it was only needed a condition like `'(pos ?i) (pos ?j)'`. The constraint `'(pos ?i) (pos ?j&:(> ?j ?i))'` was added so that the pairs are not duplicated. The algorithm will stop when from the current solution no better solution can be generated as shown in Figure 7. This algorithm works based on the following three steps:

1. Initializing the search by creating an initial solution.
2. Generating all the best descendants using swapping the cities.
3. Choosing the best successor city and updating the solution or stopping if there is no new solution.

```
(assert (mat-dist (dist ?m)))
; Control fact for the swaps
(bind ?loopmax ?x)
(bind ?i 1)
(while (<= ?i ?loopmax)
  (assert (pos ?i))
  (bind ?i (+ ?i 1))
)
; Initial solution (Cities in sequential order)
(bind ?s (create$))
(bind ?loopmax ?x)
(bind ?i 1)
(while (<= ?i ?loopmax)
  (bind ?s (insert$ ?s ?i ?i))
  (bind ?i (+ ?i 1))
)
```

Figure 7. Distance Value Insert and Swap in the Expert System

The rule in the Figure 8 performs the generation of each solution iteration. It instantiates the rule to create a solution that will later be compared against the best solution already obtained.

```
(defrule HC-step1
  (pos ?i)
  (pos ?j&:(> ?j ?i))
  (solution (path $?s) (cost ?c) (desc n))
  (mat-dist (dist $?m))
)
```

Figure 8. Expert System Solution Finder

After finding a solution, it must be compared against the best solution already obtained. So, the rule showed in Figure 9 compares the new found solution against the best solution that the expert system has. This comparing rule has a '*(declare (salience -10))*' to give it a lower priority than the rule that finds the solutions (shown in Figure 8). This tells the expert system to execute the solution comparison after each solution is found.

```
(defrule HC-step2
  (declare (salience -10))
  (solution (path $?s) (cost ?c) (desc s))
  (solution (cost ?oc) (desc s))
  (test (<= ?c ?oc))
  ?solact <- (solution (cost ?cs&(< ?c ?cs)) (desc n))
  =>
  (modify ?solact (path ?s) (cost ?c))
)
```

Figure 9. Rule comparing the best known solution to the new solution

The main goal is to find a solution that has the minimum cost and that has a cost better than the current solution. After all possible solutions have been determined; the most optimal solution (shortest path) is presented by the rule shown in Figure 10. This rule has a very low salience number which make it wait until all other rules of the expert system have been applied and completed.

```
(defrule HC-is-end
  (declare (salience -20))
  (solution (path $?s) (cost ?c) (desc n))
  =>
  (printout t "Final ->" ?s ": " ?c crlf)
)
```

Figure 10. Rule that shows most optimal solution

Experiment & Results

After developing the phase 1, of the scheduling decision support expert system based on TSP, an experiment was conducted to test the expert system. The experiment consisted of three test cases. The three test cases were modeled in a Building Information Modeling software (Revit). Test case 1 consisted of a simple 4 wall room; test case 2 was a room with a middle wall and test case 3 was two rooms with one having a middle wall (as shown in Figure 11). The walls in all the test cases were denoted using cardinal directions. So, the wall facing west was denoted as West Wall (WW), the wall facing north was denoted as North Wall (NW), etc. The middle wall was denoted as Middle Wall (MW) and in the cases where multiple walls facing in one direction a sequential number was added (i.e: SW1 and SW2 in case 3)

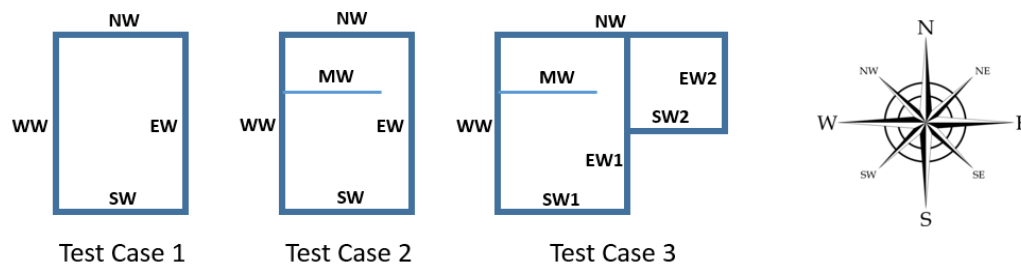


Figure 11. Test Cases for Experiment

The three test cases were shown to a subject-matter expert who provided possible scheduling sequences to erect the walls. The three test cases were also run through the experiment steps. The experiment consisted of a total of six (6) steps as shown in Figure 12 and described below.

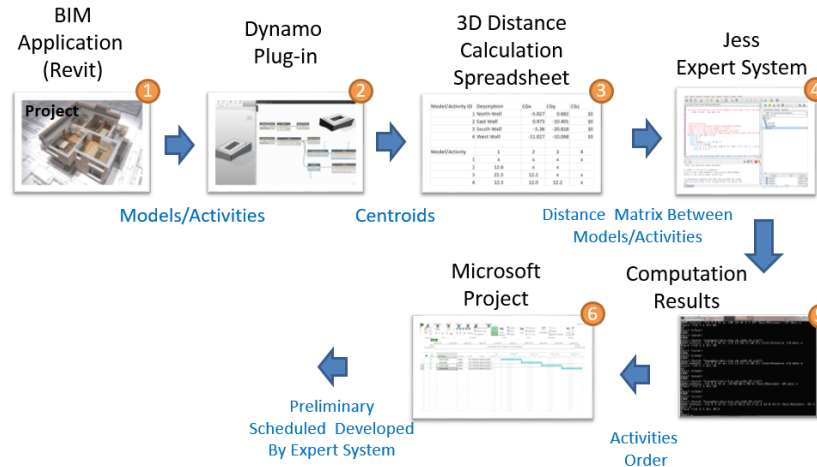


Figure 12. Overview/Steps of the Experiment

Step 1- BIM Revit

The experiment was conducted using the Building Information Modeling software called ‘Revit’. Revit Architecture is a robust architectural design and documentation software application created by Autodesk for architects and building professionals (AUTODESK, 2016). The tools and features that make up Revit Architecture are specifically designed to support building information modeling (BIM) workflows. BIM is a digital representation of physical and functional characteristics of a facility/project. (Corke, B., 2016). Figure 13 shows the test case 1.

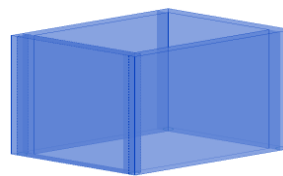


Figure 13. 3D mode of the Test Case 1.

Step 2- Dynamo Plug-In

Dynamo is a plug in for Revit that can be used to determine the properties of each model in the project. In this experiment, it was used to determine the co-ordinates (X, Y and Z) of the centroid for all the models in the project.

The Dynamo plug-in can be accessed in Revit through the ‘Manage’ tab. When Dynamo is started from Revit, it connects automatically to the active project. In Dynamo, the code consists of inter connected nodes. The code developed for the experiment included: 1- Model Element, 2- Element Geometry and 3- Solid Centroid as shown in Figure 14

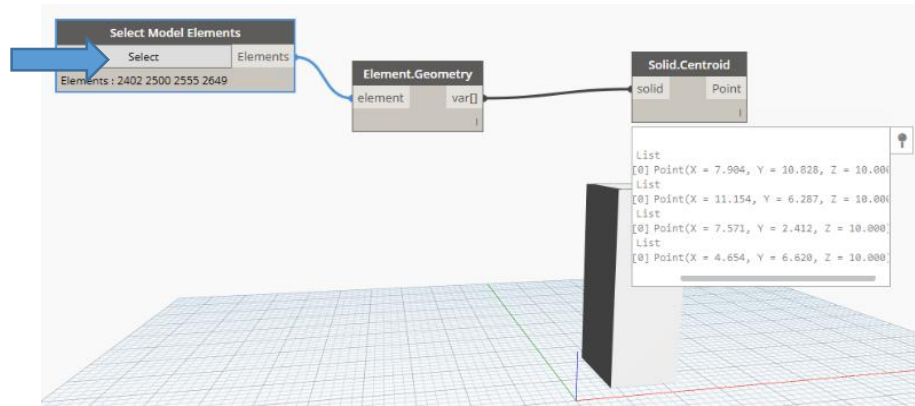


Figure 14. Dynamo code to find each model centroid

Step 3: Distance Calculations

The centroids from each model (wall) are used to find the distance between the centroids of each wall. The distances are calculated using the formula in Figure 15a through a spreadsheet developed as part of the research project to create the distance matrix shown in Figure 15b

$$d = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2}$$

Model/Activity ID	Description	CGx	CGy	CGz
1	North Wall	-5.027	0.682	10
2	East Wall	0.973	-10.401	10
3	South Wall	-5.36	-20.818	10
4	West Wall	-11.027	-10.068	10
Model/Activity	1	2	3	4
1	x	x	x	x
2	12.6	x	x	x
3	21.5	12.2	x	x
4	12.3	12.0	12.2	x

Figure 15b. Centroids & distance Matrix

Figure 15a. Distance Formulation

Step 4: Jess Expert System

The calculated matrix distances between the centroids are given as the input in the form of list of distances to the Jess Expert System to apply the rules and functions explained in the previous sections.

Step 5: Computation Results

The Jess Expert system produces a sequence of activities currently solely based on the TSP implementation.

Step 6: Microsoft Project

The resulting sequence from the expert systems is entered in a scheduling software for further processing.

Results and Lesson Learned

The results from the experiment were compared against the multiple solutions provided by the subject-matter expert. Table 3 summarizes one of the proposed solutions from the subject-matter expert for each

test case and the expert system proposed solution. It can be observed that the expert system did not have any incorrect sequencing in the first two cases. The expert system incorrectly scheduled the Middle Wall (MW) in the third case.

Table 3. Expert System Solutions

Test Case	Description	Subject-Matter Expert Possible Solution	Expert System Proposed Solution	Incorrect Sequence
1	One Room Four Wall	NW → EW → SW → WW	NW → EW → SW → WW	None
2	Two Rooms Five Walls	WW → SW → EW → NW → MW	WW → SW → EW → NW → MW	None
3	Three Rooms Seven Walls	SW1 → EW1 → SW2 → EW2 → NW → WW → MW	SW1 → EW1 → SW2 → EW2 → NW → MW → WW	One

It was also learned that to increase the complexity of the test cases more automation through programming will be needed because the centroid distances grow exponentially with the increase in number of models. For example, the third test case that was composed of seven walls had a distance matrix composed of 21 distances. However, if the number of models would double to 14 then the matrix will be composed of 91 distance and for 28 models the matrix distance will be composed of 378 distance (as shown in Figure 16)

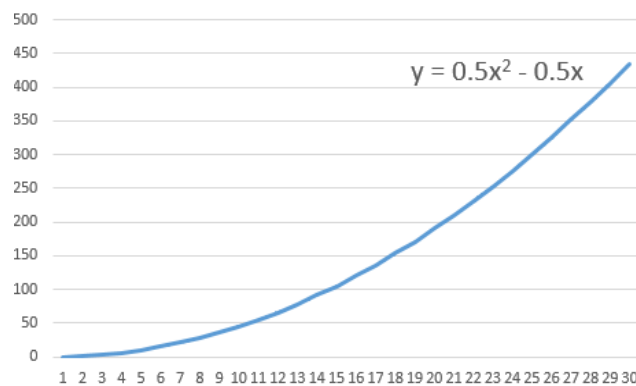


Figure 16. Exponential Growth of the Distance Matrix for the Expert System

Summary

Preparing good schedules is a time-consuming process that requires a deep understanding of the construction process. It is not ambioned to capture such deep construction expertise in an expert system. But, an expert system as shown in this paper could be the first step to support the decisions that are made by construction scheduler to reduce the time needed to prepare schedules and/or improve the quality of final construction schedules. The expert system presented in this paper showed a considerable potential for applications in construction scheduling. The rule based system of the Jess allows for future enhancement of the scheduling process.

References

- ACA (2012). Common Techniques for Construction Scheduling : Construction & Development Article by Association of Construction and Development. Retrieved October 28, 2016, from http://www.associationofconstructionanddevelopment.org/articles/view.php?article_id=10821
- AUTODESK. (2016). Revit Family | BIM Software | Autodesk. Retrieved October 31, 2016, from <http://www.autodesk.com/products/revit-family/overview>.
- Bejar, J. (2010). CLIPS - Code Snippets Version 0.8 (0.8). California 94305, US: Creative Commons. Retrieved from <http://www.cs.upc.edu/~bejar/ia/material/laboratorio/clips/CLIPS-snippets-eng.pdf>
- Corke, Bruce H. (2016) Building Information Modeling (BIM) – Expert Article on Construction Disputes. Retrieved October 31, 2016, from <http://www.robsonforensic.com/experts/http://www.robsonforensic.com/articles/building-information-modeling-BIM-expert-witness>
- Friedman-Hill, E. (2003). *Jess in Action: Java Rule-Based Systems (In Action series)*: Ernest Friedman-Hill: 9781930110892: Amazon.com: Books. Manning Publications.
- Graphisoft. (2016). Open BIM. Retrieved October 31, 2016, from http://www.graphisoft.com/archicad/open_bim/about_bim/
- Hendrickson, C (1998) Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA. “*Project Management for Construction, Fundamental Concepts for Owners, Engineers, Architects and Builders*”. ISBN 0-13-731266-0.
- Klansek, U (2011). “*Using the TSP Solution for Optimal Route Scheduling in Construction Management*” DOI 10.5592/otmcj.2011.1.3 Research paper.
- Matai, R., Singh, S P and Mittal M L (2010). Traveling Salesman Problem: An Overview of Applications, Formulations, and Solution Approaches, Traveling Salesman Problem, Theory and Applications, Prof. Donald Davendra (Ed.), ISBN: 978-953-307-426-9,
- Michael Hahsler, Kurt Hornik. (2006). “*TSP – Infrastructure for the Traveling Salesperson Problem*” Retrieved October 31, 2016.
- Rana, S; Sharma, A. (2014) “*Travelling Salesman Problem Solution with Adaptive Resonance Neural Networks*” International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 5, May 2014. ISSN: 2277 128X

- Tang, S., Ahmad, I., Ahmed, S., & Lu, M. (2004). PLANNING AND SCHEDULING DECISIONS. In Quantitative Techniques for Decision Making in Construction (pp. 163-180). Hong Kong University Press. Retrieved from <http://www.jstor.org/stable/j.ctt2jc6xz.15>
- Waddell, K. (2012). "Construction Scheduling Basics" PDHonline. Retrieved from <http://www.pdhonline.com/courses/p150/p150content.pdf>
- Wazirali, R; Alzughairi, A; and Chaczko Z (2014) "Adaptation of Evolutionary Algorithms for Decision Making on Building Construction Engineering (TSP Problem)" INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2014, VOL. 60, NO. 1, PP. 125–128.
- Weisstein, Eric W (2016). "Traveling Salesman Problem." From MathWorld--A Wolfram Web Resource. <http://mathworld.wolfram.com/TravelingSalesmanProblem.html>
- Zavichi, A; Madan, K; Xanthopoulos, P and Oloufa, (2013) A "Tsp-Based Model for On-Site Material Handling Operations with Tower Cranes".