Journal of Engineering and Sustainable Development
www.jeasd.org
Vol. 22, No.03, May 2018
ISSN 2520-0917

SOFTWARE FOR LINE OF BALANCE IN PROJECTS OF HIGHWAYS

*Dr. Mohammed Khachi Hatem[1], Dr. Ahmed Mancy Mosa[2], Dr. Mohammed Hussein Al-Dahlaki[1]

1) Asst. Prof., Faculty of Engineering, Mustansiriyah University, Baghdad, Iraq.
2) Lec.,Civil Engineering Department, AL Mansour University College, Baghdad, Iraq.

Abstract: Highways play an important role in countries' economy improvement as they are the major transportation facilities. Highway construction project needs effective planning method to overcome its complicity. Line of Balance is the most effective technique for scheduling such projects. Computerization of this technique to cover highway construction is very powerful to obtain the full advantages of this technique as computer-based program can simplify the mission of end users. The software was build using Visual Basic to ensure flexible code and interactive user-interface. The developed software was verified and validated through extensive testing during and after its building. The result of test showed its correctness and effectiveness to attain objectives.

Keywords: *Line Of Balance, Projects Of Highways, Repetitive Works, Software, Construction*

برنامج حاسوب لاعداد خط التوازن في مشاريع الطرق

الخلاصة: تلعب الطرق دورا مهما في نمو الاقتصاد حيث انها تمثل الوسيلة الرئيسية للنقل. و تحتاج مشاريع انشاء الطرق الى طريقة تخطيط فعالة للتغلب على تعقيداتها. تعد طريقة خط التوازن واحدة من الطرق الفعالة في جدولة تلك المشاريع. حوسبة هذه الطريقة لادارة مشاريع انشاء الطرق فعال جدا للافادة منها. ان برامج القواعد المحوسبة تبسط عمل المستخدم (مدير المشروع). تم بناء البرنامج المحوسب في هذا البحث باستخدام لغة فيجوال بيسك لضمان مرونة الشفرات المحوسبة و ضمان اعداد واجهة تطبيق تفاعلية مع المستخدم. تم التحقق من صحة و فعالية البرنامج من خلال الفحص الشامل اثناء و بعد بناء البرنامج. و قد اثبت نتائج الفحص صحة و فعالية البرنامج لتحقيق اهدافه المنشودة.

1. Introduction

Construction of a highway system as a part of public infrastructure is a significant way for any country to improve its economy [1, 2, 3, 4]. Transportation in developing countries typically depends on a road network rather than on other modalities. Therefore, there is high demand for construction of new roads to connect city centres with the new expanding districts and other locations[5]. However, little researches on highway constructionare implemented [6], and research on this topic is in great demand [7].

*Corresponding Author mohammedkhachi@yahoo.com

Generally, construction projects are complex and require high management efforts[8].Planning which is the core of the effective project management[9, 10, 11]is a challenge in construction projects[8].Planning of highway construction can be considered as an essential process [12, 13]to ensure efficient construction management that lead to attain project objectives in lowest time and cost. In addition, proper planning in highway projects eliminates work stoppage and prevents technical, economical, and legal problems. In contrast, improper planning in highway construction industry may lead to costly problems due to the size of such projects[3].

Line of Balance (LOB) can be considered as the most effective technique for presentation of linear construction projects with receptive activities such as highway works[14]. In contrast with network methods, planning based on Line of Balance (LOB) technique can be considered as the most appropriate technique for highway construction projects as they involve repetitive environment[14-16]. In spite of the advantages of LOB technique on other scheduling methods such as bar charts and critical path method (CPM)[17-19], its use is still limited[20] in comparison with the use of computerized packages of CPM due to their availability and simplicity [10].

Mostly, participants in highway construction management such as planners, construction managers, and contractor representatives deal with LOBmanually. This means that they must remember all parameters during planning which is impractical[11],costly and time consuming[8]. They may avoid computerized techniques due to their complexity; whereas, the aim of computerization is simplification of the planning process not the opposite. A number of commercial software packages such as Asta, Vico and DYNAProject are available in this domain. However, the available computerized packages are suitable for experts in the domain of LOB technique[11, 21]. In addition, these packages deal with general projects but not, specifically, with highway projects. Therefore, there is a clear demand for simple and user friendly software to be suitable for novice users specialized in highway projects. Based on these facts, this study aims to build software in this domain to be used by novice users to obtain LOB diagrams and tables in highway construction projects simply and accurately. The proposed software can simplify the novices' mission through providing them with predesigned lists involved typical highway construction activities. In addition, the proposed software can be used by users with intermediate experience as well as experts to obtain rapid and accurate results in the study domain. Furthermore, it can suggest a number of recommendations to solve disputes that interrupts the construction process.

Visual Basic (VB) programming language was used to build the proposed software due it its advantages such as visual capabilities that help users to deals with programs easily. VB, also, is capable to provide accurate calculations. The present software involves easy and user-friendly input-output user interface to encourage the users and simplify their mission. The software can be update easily and can be expanded to involve other types of projects that exhibit repetitive activities such as railways, tunnels, and pipelines.

2. Literature Review

LOB technique assumes stable production rate for all activities in a project. Compared to other scheduling techniques, LOB involves several advantages such as: (1) it is an effective tool for progress monitoring; (2) It presentsprogram bottleneck that lead to schedule slippage; (3) It minimizes conflicts among units and provides smooth procession of crews. The advantages of LOB lead to an expansion in its use among the workers in construction industry. Therefore, researchers paid efforts to study, improve, adopt, and apply this technique in different domains. This review deals with researches in construction domain. Table 1 covers 21 researches in planning of projects with repetitive nature. The parameters related to these researches are abstracted in Table 1. The review shows that two researches were developed in highway engineering domain. However, these two researches based on traditional manners and not computerized in software. On the other hand, 5 researches adopted software to treat the problems. One of the five was specialized with housing projects and four can deals with general construction projects. Among the four, two researches deal with repetitive scheduling, one deals with linear scheduling, and one deals with general scheduling; however, no one is specialized in LOB specifically. Based on this review, a number of software packages were developed for general projects. However, no research was implemented to build software for LOB technique in highway construction specifically. Therefore, this study covers the gap in this domain.

3. Software Building

In order to build the software in this study, the domain was covered with an extensive literature review to ensure the effectiveness of the study and to build useful software which is the main objective of this study. The steps of software building are presented in the following sub-articles

3.1. Selection of Software Building Environment

Several programming environments are used to build software such as conventional programming languages (Pascal, C, and Visual Basic (VB)) and general purpose programming languages (Prolog, Lisp, Loops, and OPS5). Conventional programming languages are designed and optimized for procedural manipulation of data such as numbers and arrays.

In building the proposed software, the present research chose Visual Basic, one of the most widely used computer programming languages, in building the software. It not only creates Windows programs, but also takes full advantage of the graphical way that Windows works by letting programmers develop their systems by using a computer mouse. Visual Basic is truly revolutionary and gives programmers a much more capable, efficient, and flexible way to write computer software programs [22]. This programming language offers ease of use, which is a requirement of the user interface [23]and writes codes in a simple syntax in a natural mathematic language.

Table 1 a Review of Researches in Planning of Projects with Repetitive Nature

[1] Reference	[2] Technique	[3] Specialization	[4]	[5]	[6]	[7]	[8]
Carr and Meyer [24]	Line of Balance	Buildings	×	✓	×	✓	✓
[25]O'Brien James	Visual Project Management	Buildings	×	✓	×	✓	✓
Selinger[26]	General Construction Planning	General	×	✓	✓	✓	✓
[Johnston David [21]	Linear Scheduling Method	Highway	×	✓	✓	×	✓
[28Arditi and Albulak [28]	Line of Balance	Pavement	×	✓	✓	×	✓
[29]Chrzanowski E.N and Johnston	Linear Scheduling Method	General	×	✓	×	✓	✓
[Reda [30]	Repetitive Project Modeling	General	×	✓	✓	✓	✓
[31]El-Rayes and Moselhi	Resource Driven Scheduling	General	×	✓	×	✓	×
[32]Harmelink and Rowings	Linear Scheduling Model	General	×	✓	✓	×	✓
[Harris and Ioannou [33]	Repetitive Scheduling Method	General	×	✓	×	✓	✓
[Arditi, Tokdemir et al. [16]	Line of Balance	General	×	✓	✓	×	✓
[Hegazy and Wassef [34]	Repetitive non-serial activity scheduling	General	✓	✓	×	×	✓
Yang and Ioannou[35]	Repetitive Scheduling Method	General	✓	✓	×	✓	✓
Mahdi [10]	Linear Scheduling Method	Housing	✓	✓	×	✓	✓
[Yang and Chang [15]	Repetitive Project Modeling	Housing	×	✓	×	✓	✓
[Huang and Sun [36]	Repetitive Scheduling Method	General	×	×	✓	✓	✓
[17]Agrama	Linear Scheduling Method	General	✓	✓	✓	✓	✓
[37]Anuradha and Bhavani	General Construction Scheduling	General	✓	✓	×	✓	✓
Pai, Verguese et al.[38]	Line of Balance	Housing	×	✓	×	✓	✓
[Maheswari, Charlesraj et al. [20]	Linear Scheduling Method	Buildings	×	✓	×	×	×
[Su and Lucko [14]	Line of Balance/ Linear Scheduling	General	×	✓	×	✓	✓

[4] Computerized [5] Unit Based [6] Untypical repetitive activities [7] Work resources continuity [8] Fixed sequence

Visual Basic is an event-driven language in which code is executed in response to events. These events come from either Windows itself or the users. Visual Basic uses forms designed to interact with the user running the system. The Windows dialogue box guides the user when operating the system.

Visual Basic makes programming enjoyable and reduces the effort required of the programmer. While it simplifies programming and makes it as easy as dragging graphic

objects onto the screen by using a mouse, the programming language forms the background of everything that occurs in a running Visual Basic program. However, the language is a secondary consideration to user interface. A Windows program offers a high degree of user interaction by using graphical elements that form the objects on the window which the user sees.

Visual Basic provides the developer with the power and flexibility required to support a complex application. In addition, VB can provide the software with computing speed and flexibility and offers a large set of tools which allow the developer to build and design customized software suitable for the intended application. This process requires long development time and skill of experienced programmers to develop sophisticated and appropriate software. The developed software is a Windows-based application created in the Visual Basic development environment which contains all resources to build powerful Windows-based programs efficiently and quickly. The main advantage of Windows is the use of graphic packages, conventional programming languages, and popular data files to interface with the user, thus making human computer interaction easier and more natural [22, 23].

3.2. Software Coding

Objects of the present software are non-functional unless acode in event procedures is written to enable a response to system actions. Therefore, the developer wrote a code that controls every object in the software. In addition, the code was written to drive logical and mathematical operations. Moreover, a specific code was written to connect controls and forms in the present software. Another code was written for a variety of purposes, such as code for capturing incorrect inputs (message box code) and remarks' code. Remarks do not execute any function in the software. However, they are kept in the code window to help the developer and updater during the programming process by defining the components' names and variables, describing the code procedure, and separating the procedures.

Visual Basic assigns a default name to each object, which is used during coding and operation. The developer changed the default names with clearer, more expressive names related to the functions of each command. Visual Basic established a framework of event-driven procedures to enable the developer to add codes. The format for each of these subroutines (all object procedures in Visual Basic are subroutines) is as follows:

```
Sub Object Name_Event (Optional Arguments)
Code: invoke event 1
Code: invoke event 2
…….
End Sub
```

Visual Basic provides the Sub line with its arguments (if any) and End Sub statement. The developer provides the required code, and Visual Basic automatically builds a framework of all event procedures. The developer adds code to event procedures to obtain a response from the required application. Code is a number of statements which work together to perform events in the present software. The simplest statement used in the present software is the assignment statement. It consists of a variable name followed by the assignment operator (=) and followed by some sort of

expression. Assignment statement stores information. Comment statements begin with the keyword (Rem) or a single quote ('). Expressions in the present software work with three types of operators, namely, arithmetic operators, comparison operators, and logical operators. The arithmetic operators carry out arithmetic operations, such as exponentiation (^), multiplication (*), division (\), addition (+), and subtraction (-). Six comparison operators were used in coding of the present software, namely, greater than (>), less than (<), greater than or equal to (>=), less than or equal to (<=), equal to (=), and not equal to (<>). The result of a comparison operation is a Boolean value (true or false). Two logical operators were used in the present software code, namely, AND and OR.

Developing software that will perform without errors at the first attempt is nearly impossible. The present software encountered a few errors after it was completed and has undergone numerous modifications until all functional requirements worked properly. Errors may occur during program coding under Visual Basic or any other programming language. These errors must be captured and corrected to ensure that the product is error-free. To capture and correct errors in the present software, the developer performed several single tests during the coding process. Unit testing and integrated testing were performed. Unit testing involves testing the units one by one in separate testing activities. This testing was continually performed during all stages of the present software development to verify that each unit in the system performs its intended function. The developer checked the internal structure of the software by covering all possible combinations of constants, variables, relationships, and source code paths. Errors were diagnosed during the testing process and corrected while the software was still under construction and before transforming it into an executable version. The developer performed integration testing to verify that all units operate together as expected [39]. To run the system, the user shall provide the required inputs. Input data are processed in the software to provide the user with outputs (LOB table and LOB diagram). The developer tested the interaction between the user and the present software. Similar to unit testing, integration testing was periodically performed during the present software building to verify its capability to execute the intended functions. Three types of errors were captured in the present software testing, namely, syntax errors, runtime errors, and logic errors. Syntax error (or compiler error) is a mistake (such as a misspelled property or keyword) that violates the programming rules of Visual Basic. Visual Basic points out several types of syntax errors when entering a statement in the present software, which prevents it from running. The developer fixed all these syntax errors. A runtime error causes the present software to stop unexpectedly during execution. Runtime errors occur when an outside event or an undiscovered syntax error forces the present software to stop in the middle of an operation. For instance, it occurs when the file name or object name (control name) was misspelled or when the intended event does not exist. In such cases, the software stopped running and Visual Basic displayed a runtime error message which alerted the programmer to a bug in the code. Visual Basic provides a debug button to help programmers capture errors. When the developer pressed “Debug,” Visual Basic highlighted the error. Logic error is a human error that causes the program code to produce wrong results. The developer performed extensive debugging to track down and eliminate logic errors.

The source code of the developed software has multiple forms connected in one structure. The source code (with the extension .vbp) was designed for the use of the developer who is responsible for developing and updating the software. To run the source code, the Visual Basic programming language package needs to be installed in a computer. The developer ensured that this version is verified and free of bugs. An executable version (with the extension .exe) was created for the end-user. This version is protected and uneditable.

3.3. Design of User Interface

User interface is the interface between user and software. The user communicates with the software via this interface. Generally, user interfaces obtain information needed to obtain the objective by asking the user to answer questions or prompting the user to provide relevant information in a preformatted manner. User interfaces should be designed in such a way that they effectively collect information and structurally present the whole software without restricting the overall performance. A good graphic user interface also reduces the required learning time and the number of mistakes made by first-time users. Regardless of how well the internal code is organized or how effective the software's performance is, the quality of user interface design alone may determine the fate of the software.

The developer started building the system by creating a new project under the Visual Basic environment. Visual Basic creates at least one form with each project. Each form includes the interface window (screen) and code window. Fig. 1 and Fig. 2 illustrate examples of input and output interface windows respectively.

The developer designed the interface windows to be clear, user-friendly, and interactive. The interface window includes a number of components or controls (each represents a tool) such as frames, labels, text boxes, images, option buttons (radio buttons), checkboxes, command button, combo boxes, and message boxes.

Each control was designed to perform a specific function in the software. Each interface window includes the title bar and the "minimize", "maximize", and "close" buttons. Forms and controls are called objects. The developer utilized the Visual Basic to design clear interface windows by using a variety of colours for background and texts. Changes in colours, fonts, and other visual effects were used to draw the user's attention to some points.

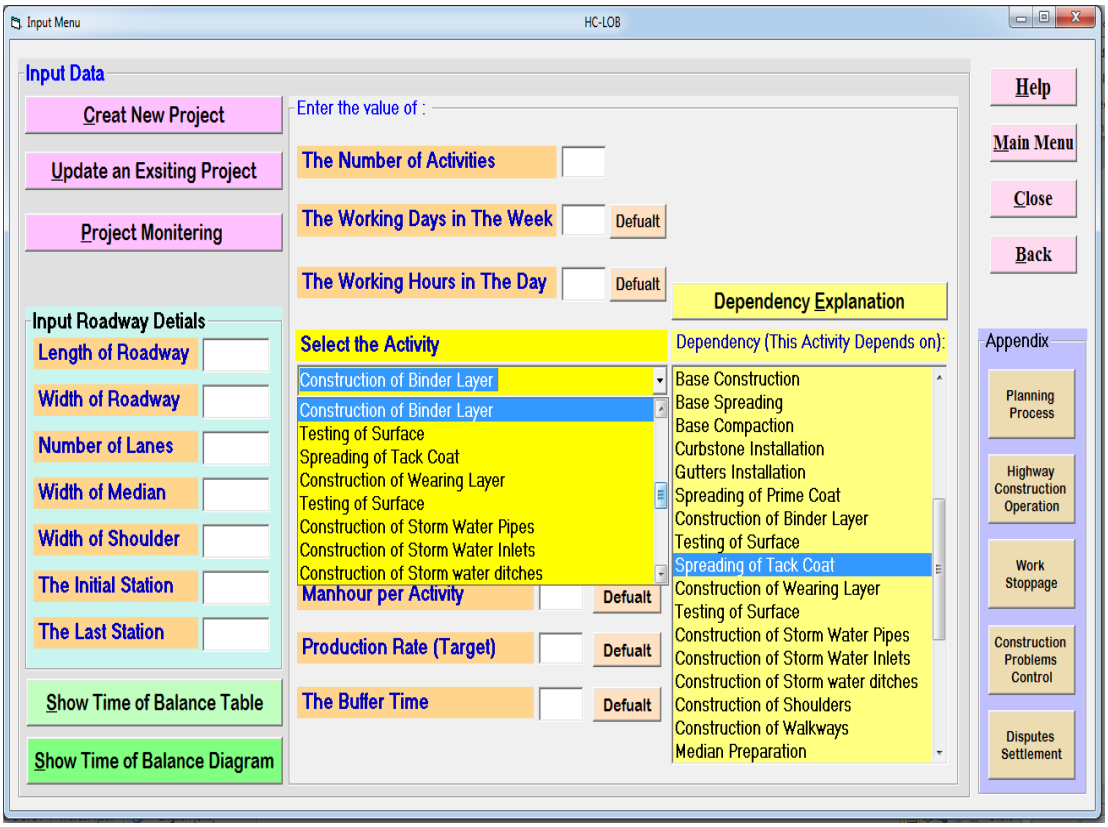

Figure 1: Input user interface window (an example)

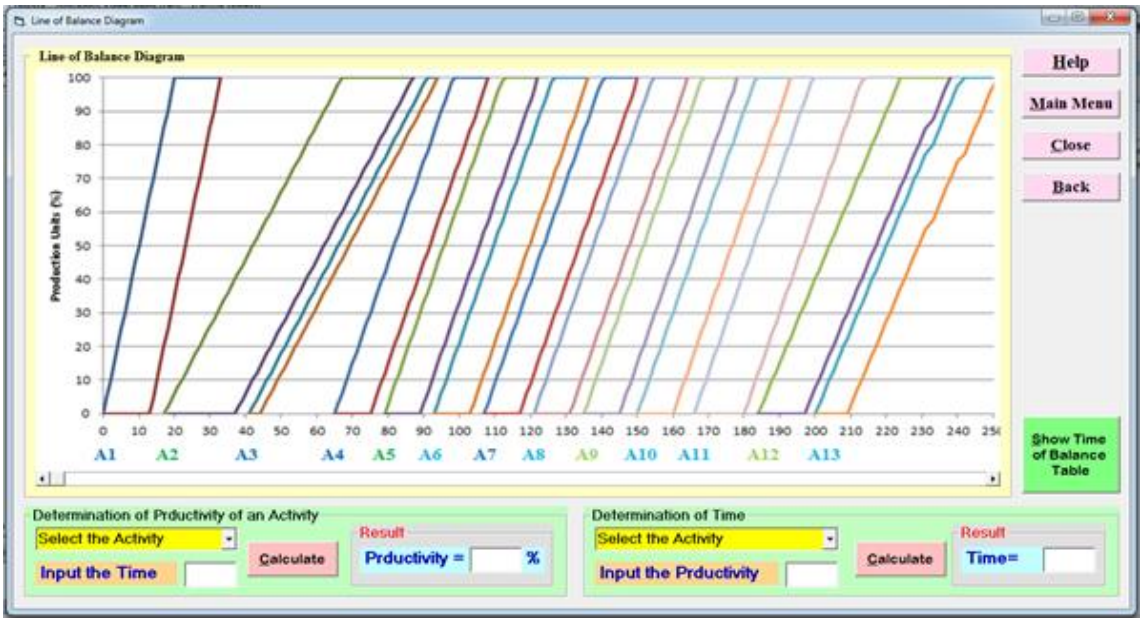

Figure 2: Output user interface window (an example).

4. Software Verification

Software testing is the first step of verification. Two testing types, namely, informal and formal testing, are conducted. The developer conducts informal tests to measure the development progress and to find errors or exceptions. For this type of testing, wide input data ranges are used to assess whether the test data are reasonable for the software. Formal testing, on the other hand, is requirement driven and is performed to verify that the software is ready for its intended use and to

demonstrate that the software meets requirements. Verifications are performed prior to the completion of the system. During the development process, the uncompleted system is subjected to frequent informal testing. Whenever a new object (function, class, database, and so on) is developed, the system developer must perform informal tests by using wide test data ranges which aim to capture any error or exception from the newly developed component. Informal testing (unit testing and integrated testing) was extensively performed on the developed software as previously described.

Formal testing was performed by design of a questionnaire survey based on a 5-point Likert scale to test users' satisfaction with the software. Four groups of users were selected to participate in this survey. The first group includes 4 computer specialists; one with PhD in information technology and three with Master in software engineering. The second group includes two professionals in project management (both have PhD in civil engineering). The third group includes 2 professionals in highway construction. The fourth group consists of 8 novice engineers.

The software was used by the participants to evaluate it depending on the questionnaire presented in Table 2. The result of evaluation reflects the satisfaction of the users through their high mean ratings (more than 3). Satisfaction of the users in all groups (computer professionals, experts and novice engineers) indicates that the user interface is friendly. Questionnaire reliability was statistically tested by calculating the Cronbach's alpha, which was 0.946, indicating high reliability.

Table 2. Results of software testing

	Questions	Mean Value	Standard Deviation
1	The software is easy to use	4.3	0.68
2	The software runs quickly	5.00	0.00
3	The user interface is user friendly	4.56	0.61
4	Obtaining an explanation from the software is easy	3.8	0.39
5	The explanations are useful	3.8	0.39
6	Help facilities are effective	3.68	0.46
7	The questions are helpful	3.75	0.43
8	The questions are clear	3.75	0.43
9	The terms are clear	3.75	0.43
10	Presentation of results is clear	4.13	0.70
11	Presentation of results is complete	4.30	0.60
12	ES-CCPRHP is helpful to provide solutions	4.25	0.66
13	Generally, I am satisfied with the software	4.25	0.66

5. Validation

Validation is performed to ensure that the softwareworks accurately [39] andgives correct results to obtain the objectives. The satisfaction of the four professionalsin Table 2 can be considered as an evidence for the software's validation as they are satisfied with the input-output system, software speed, help facilities, and other components. In addition, several projects were performed using the software and were compared with manual calculations and graphing. The results of the software were in line with manual ones. In addition, the software results were more expressive and accurate.

6. Software Updating

The developed software can be updated easily by the developer or by any competent Visual Basic user under the supervision of a civil engineer. The source code of software includes remarks to simplify the update operation, especially when the updater is not the developer.

7. Conclusions

Line of Balance is an effective technique in highway construction planning. This paper presents the development of software for scheduling of highway construction projects using Line of Balance technique to help the engineers in this domain. The software was build using Microsoft Visual Basic. The software was built with a flexible code and interactive graphical user-interface to simplify the using and updating. The softwarewas verified and validated by extensive testing. The results of testing proved that the system is built correctly and can efficiently provide the accurate outputs in a clear form; generally, the system is acceptable with 4.25 on a scale of 5.The software also has a flexible and friendly user interface based on evaluation of the users with different backgrounds. The developed software can be used by novices to obtain accurate outputs. In addition, professionals can use the developed software to obtain rapid outputs. Due to flexible coding of the developed software, it can be expanded to include other types of projects in transportation such as railways and tunnels and other types of project with repetitive nature such as pipe lines and channels. Moreover, it can be expanded to include other linear construction scheduling techniques.

8. References

1. Chou J S and Tseng H C (2011). *Establishing expert system for prediction based on the project-oriented data warehouse*, Expert Systems with Applications Vol. 38, pp. 640-651.
2. Santos G, Behrendt H, Maconi L, Shirvani T and Teytelboym A (2010).*Part I: Externalities and economic policies in road transport*, Research in Transportation Economics Vol. 28, pp. 2-45.
3. Kaliba C, Muya M and Mumba K (2009). *Cost escalation and schedule delays in road construction projects in Zambia*, International Journal of Project Management Vol. 27, pp. 522-531.
4. Mosa A M, Taha M R, Ismail A and Rahmat R A O K (2013). *A diagnostic expert system to overcome construction problems in rigid highway pavement* Journal of Civil Engineering and Management Vol. 19, pp. 846-861.
5. Mosa A M, Atiq R, Raihantaha M and Ismail A (2011). A *knowledge base system to control construction problems in rigid highway pavements* Australian, Journal of Basic and Applied Sciences Vol. 5, pp. 1126-1136.
6. Miller S, Huerne H and Dorée A (2007).*"The asphalt paving process: plans for action research"*. In: Proceedings of 4th Nordic Conference on Construction Economics and Organisation Development Processes in Construction Management,

ed B Atkin and J Borgbrant (Luleå, Sweden: Luleå University of Technology Department of Civil and Environmental Engineering) pp. 37-46.

7. Sivilevičius H, Zavadskas E K and Turskis Z (2008).*Quality attributes and complex assessment methodology of the asphalt mixing plant*, Baltic Journal of Road and Bridge Engineering Vol. 3, pp. 161-166.
8. Monghasemi S, Nikoo M R, Khaksar Fasaee M A and Adamowski J (2015).*A novel multi criteria decision making model for optimizing time–cost–quality trade-off problems in construction projects,*Expert Systems with Applications Vol. 42, pp. 3089-3104.
9. Krzemiński M and Wypysiak A (2014).*Scheduling Complete Review Application for Road Works*, Procedia Engineering Vol. 91, pp. 400-405.
10. Mahdi I M (2004).*A new LSM approach for planning repetitive housing projects,*International Journal of Project Management Vol.22, pp. 339-346.
11. Chau K W, Anson M and Zhang J P (2003).*Implementation of visualization as planning and scheduling tool in construction,* Building and Environment Vol.38, pp. 713-719.
12. [Mosa A M, Rahmat R A O K, Ismail A and Taha M R (2013).*Expert System to Control Construction Problems in Flexible Pavements,*Computer-Aided Civil and Infrastructure Engineering Vol.28, pp. 307-23.
13. [Mosa A M, Atiq R, Raihantaha M and Ismail A (2011).*Classification of construction problems in rigid highway pavements*, Australian Journal of Basic and Applied Sciences Vol.5, pp. 378-395.
14. Su Y and Lucko G (2015).*Comparison and Renaissance of Classic Line-of-balance and Linear Schedule Concepts for Construction*, Industry Procedia Engineering Vol.123, pp. 546-556.
15. Yang I T and Chang C Y (2005). S*tochastic resource-constrained scheduling for repetitive construction projects with uncertain supply of resources and funding,*International Journal of Project Management Vol. 23, pp. 546-553.
16. Arditi D, Tokdemir O B and Suh K (2001).*Effect of learning on line-of-balance scheduling*, International Journal of Project Management Vol.19, pp. 265-277.
17. Agrama F A E M (2011).*Linear projects scheduling using spreadsheets features,*Alexandria Engineering Journal Vol. 50, pp. 179-185.
18. Agrama F A (2012).*Multi-objective genetic optimization of linear construction projects*, HBRC Journal Vol.8, pp. 144-151.
19. Huang R Y and Sun K S (2005).*System development for non-unit based repetitive project scheduling Automation in Construction,*Vol.14, pp. 650-665.
20. Maheswari J U, Charlesraj V P C, Goyal A and Mujumdar P (2015).*Application of Relationship Diagramming Method (RDM) for Resource-constrained Scheduling of Linear Construction Projects*, Procedia Engineering Vol.123, pp. 308-315
21. Cooke B and Williams P (2004).*"Construction planning*", Programming and Control: Oxford: Blackwell Publishing.
22. Olugu E U and Wong K Y (2012). A*n expert fuzzy rule-based system for closed-loop supply chain performance assessment in the automotive industry*, Expert Systems With Applications Vol.39, pp. 375-384.

23. Mosa A M, Taha M R, Ismail A and Rahmat R A O K (2013). A*n Educational Knowledge-based System For Civil Engineering Students in Cement Concrete Construction Problems,* Procedia - Social and Behavioral Sciences Vol.102, pp. 311-319.
24. Carr R I and Meyer W L (1974).*Planning construction of repetitive building units,*Journal of the Construction Division Vol.100, pp. 403-412.
25. O'Brien James J (1975). VPM *scheduling for high-rise buildings* American Society of Civil Engineers, Journal of the Construction Division Vol.101, pp.895-905.
26. Selinger S (1980).*Construction planning for linear projects,*American Society of Civil Engineers, Journal of the Construction Division Vol.106 , pp.195-205.
27. Johnston David W (1981). *Linear scheduling method for highway construction,*Journal of the Construction Division Vol.107, pp. 247-261.
28. Arditi D and Albulak M Z (1986).*Line-of-Balance scheduling in pavement construction,*Journal of Construction Engineering and Management Vol.112, pp. 411-424.
29. Chrzanowski E.N, Jr. and Johnston D W (1986).*Application of linear scheduling,* Journal of Construction Engineering and Management Vol.112, pp. 476-491.
30. Reda R M (1990).*RPM: Repetitive Project Modeling,* Journal of Construction Engineering and Management Vol.116, pp. 316-330.
31. El-Rayes K and Moselhi O (1998).*Resource-driven scheduling of repetitive activities,* Construction Management and Economics Vol.16, pp. 433-46
32. Harmelink D J and Rowings J E (1998).*Linear scheduling model: Development of controlling activity path*, Journal of Construction Engineering and Management Vol.124, pp. 263-268.
33. Harris R B and Ioannou P G (1998).*Scheduling projects with repeating activities,*Journal of Construction Engineering and Management Vol.124 269-278.
34. Hegazy T and Wassef N (2001).*Cost optimization in projects with repetitive nonserial activities,* Journal of Construction Engineering and Management Vol.127, pp. 183-191.
35. Yang I-T and Ioannou P G (2001).*Resource-driven scheduling for repetitive projects: A pull-system approach,* In: The 9th Conference for International Group of Lean Construction (IGLC-IX), Singapore: National Singapore University.
36. Huang R Y and Sun K S (2006). N*on-Unit-Based Planning and Scheduling of Repetitive Construction Projects,* Journal of Construction Engineering and Management Vol.132, pp. 585-597.
37. Anuradha D and Bhavani S (2012).*Unit based Scheduling in Project Management: A Programming Approach,* International Journal of Computer Applications Vol.59, pp. 49-55.
38. Pai S K, Verguese M P and Rai M S (2013).*Application of Line of Balance Scheduling Technique (LOBST) for a Real estate sector,* International Journal of Science, Engineering and Technology ResearchVol. 2 pp: 82-95.
39. Aguilar R M, Muñoz V, Noda M, Bruno A and Moreno L (2008). V*erification and validation of an intelligent tutorial system,* Expert Systems with Applications Vol.35, pp. 677-685.