Alexandria Engineering Journal (2013) 53, 81–93

FACULTY OF ENGINEERING ALEXANDRIA UNIVERSITY

Alexandria University

Alexandria Engineering Journal

www.elsevier.com/locate/aej
www.sciencedirect.com

ORIGINAL ARTICLE

RPERT: Repetitive-Projects Evaluation and Review Technique

Remon Fayek Aziz *

Structural Engineering Department, Faculty of Engineering, Alexandria University, Egypt

Received 16 March 2013; revised 7 July 2013; accepted 13 August 2013
Available online 12 September 2013

KEYWORDS

Repetitive Construction Projects;
Program Evaluation and Review Technique;
PERT Activity;
Optimistic Duration;
Pessimistic Duration;
Most likely Duration;
Expected Duration;
Critical Path;
Software

Abstract Estimating expected completion probability of any repetitive construction project with a specified/certain duration including repetitive identical activities by using program evaluation and review technique is the most essential part in construction areas since the activities were had optimistic, most likely and pessimistic durations. This paper focuses on the calculation of expected completion probability of any repetitive construction project within a specified/certain duration (contract duration) by using Line Of Balance technique (LOB) in case of single or multiple number of crews integrated with Program Evaluation and Review Technique (PERT). Repetitive-Projects Evaluation and Review Technique (RPERT), which is a simplified software, will generate the expected project completion probability of a specified/certain duration (contract duration). RPERT software is designed by java programming code system to provide a number of new and unique capabilities, including: (1) Viewing the expected project completion probability according to a set of specified durations per each identical activity (optimistic time, most likely time, and pessimistic time) in the analyzed project; (2) Providing seamless integration with available project time calculations. In order to provide the aforementioned capabilities of RPERT, the system is implemented

Abbreviations: ACI, Activity Critical Index; ADM, Arrow Diagram Method; CPM, Critical Path Method; DSS, Decision Support Systems; EXPERT, EXemplary Program Evaluation and Review Technique; HSS, Hybrid Scatter Search; LOB, Line Of Balance; MC, Monte Carlo; PACE, Programme Analysis, Control and Evaluation; PCI, Path Critical Index; PDM, Precedence Diagram Method; PERT, Program Evaluation and Review Technique; RCPSP, Resource-Constrained Project Scheduling Problem; RPERT, Repetitive-Projects Evaluation and Review Technique; VPM, Vertical Production Method

* Tel.: +20 12 2381 3937.
E-mail addresses: Remon_fayek@hotmail.com, remon.aziz@alexu.edu.eg.

Peer review under responsibility of Faculty of Engineering, Alexandria University.

ELSEVIER Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.aej.2013.08.003

and developed in four main modules: (1) A user interface module; (2) A database module; (3) A running module; and (4) A processing module. At the end, an illustrative example will be presented to demonstrate and verify the applications of proposed software (RPERT), by using probabilistic calculations for repetitive construction projects.

1. Introduction

In 1957, a new managerial planning and control technique was developed, Program Evaluation and Review Technique (PERT) by the U.S. Navy Special Projects Office on the Polaris missile system to support the nuclear submarine projects. It is a technique for estimating and planning a large project. One of its most powerful concepts is the management of *probabilities*. PERT makes use of simple statistical mathematics in order to come up with a probability distribution for the completion dates of the project milestones, it was designed to provide: (1) Management information on actual and impending problems in completing a project; (2) A continuous status report on active projects for achieving established objectives, completion dates, and the probability of reaching both; and (3) Notation of most and least critical component activities within each project. PERT presented a comprehensive illustration of all major project activities and their interdependencies. In fact, it even provided time requirements needed for completing each component activity. It focused managerial attention on those business activities most vital in meeting the project completion date and identified which resources could be used more effectively if transferred to other phases of a project. Finally, PERT provided a scheme of the project as it occurred, thereby illustrating the effects of managerial changes in the entire project. The ability of PERT to predict future performance and potential future problems through frequency reporting, marked its major departure from previous planning and control techniques which relied heavily on historical data. The estimation of time for the completion of each activity is important in the network analysis; this can be done using three possible assumptions: (1) Most optimistic time (a): This time assumes that everything will go according to minimum amount of difficulties and such situation may occur approximately one percent of time; (2) Most pessimistic time (b): This time assumes that everything will not go according to plan and that the maximum potential difficulties will develop and may occur approximately one percent of time; and (3) Most likely or normal time (m): This is the time that would most often occur, should this effort be reported over again. The estimated time of an activity completion is given by using three estimates that are combined into an *expected duration* and a *standard deviation*. The expected completion duration (μ) is assumed to be $(1 \times a + 4 \times m + 1 \times b) \div 6$, and the standard deviation (s) is assumed to be $(b–a) \div 6$.

2. Terminology

According to Davidson [1] and Ronald [2], they defined some expressions of PERT as follows: (1) PERT event: is a point that marks the start or completion of one or more tasks. It consumes no time and uses no resources. It marks the completion of one or more tasks and is not reached until all activities leading to that event have been completed; (2) Predecessor event: an event (or events) that immediately precede some other events without any other intervening events. It may be the consequence of more than one activity; (3) Successor event: an event (or events) that immediately follow some other events without any other intervening events. It may be the consequence of more than one activity; (4) PERT activity: is the actual performance of a task. It consumes time, it requires resources (such as labor, materials, space, machinery), and it can be understood as representing the time, effort, and resources required to move from one event to another. A PERT activity cannot be completed until the event preceding has occurred; (5) Optimistic time: the minimum possible time required to accomplish a task, assuming everything proceeds better than what is normally expected; (6) Pessimistic time: the maximum possible time required to accomplish a task, assuming everything goes wrong but excluding major catastrophes; (7) Most likely time: the best estimate of time required to accomplish a task, assuming everything proceeds as normal; (8) Expected time: the best estimate of time required to accomplish a task, assuming everything proceeds as normal (the implication being that the expected time is the average time that the task would require if the task were repeated on a number of occasions over an extended period of time); (9) Float or Slack: the amount of time that a task in a project network can be delayed without causing a delay in the Subsequent tasks (free float) or project completion (total float); (10) Critical Path: the longest possible continuous pathway taken from the initial event to the terminal event. It determines the total calendar time required for the project; and, therefore, any time delays along the critical path will delay the reaching of the terminal event by, at least, the same amount; (11) Critical Activity: An activity that has total float equals to zero. Activity with zero float does not mean it is on the critical path; (12) Lead time: the time by which a predecessor event must be completed in order to allow sufficient time for the activities that must elapse before a specific PERT event reaches completion; (13) Lag time: the earliest time by which a successor event can follow a specific PERT event; (14) Slack: the slack of an event is a measure of the excess time and resources available in achieving this event, positive slack would indicate ahead of schedule; negative slack would indicate behind schedule; and zero slack would indicate on schedule; (15) Fast tracking: performing more critical activities in parallel; and (16) Crashing critical path: Shortening duration of critical activities, [1,2].

3. Research background

PERT is a management tool for defining and integrating events with coordinating moves for completing a project's objectives on time; a process which must be accomplished in time to assure completing project objectives on schedule. Its use is not restricted to the business world. It can be applied to any endeavor which requires planned, controlled, and integrated work patterns; PERT is valuable to manage where multiple tasks are going simultaneously to reduce the redundancy. The technique used for project scheduling will vary depending upon project's size, complexity, duration, personnel, and owner requirements as shown in Fig. 1. It divided the construction projects into two main groups. The first one is projects with non-repetitive activities. The second group has multiple numbers of stages. Projects with non-repetitive activities are divided into two main groups. The first one is the Gantt chart, and being one of the oldest methods used in construction planning and developed by Harvey Gantt during the World War I, it is a visual management system, on which future time is plotted horizontally and work to be completed is indicated in a vertical line; the second one is "network-based method" which are two widely known network based techniques, namely, Critical Path Method (CPM) and Program Evaluation and Review Technique (PERT). Both methods were developed simultaneously and independently during the first use in 1957. PERT borrowed some CPM applications. PERT proved to be an ideal technique for one-of-a-kind projects, using a time network analysis to manage personnel, material resources, and financial requirements. The growth of PERT paralleled the rapid expansion in the defense industry and meteoric developments in the space race. After 1960, all defense contractors adopted PERT to manage the massive one-time projects associated with the industry. In the last 30 years, PERT has spread, as has CPM, as a major technique of integrated project management. As with all aspects of business, the Internet has become a powerful tool with respect to PERT. Managers can now locate PERT applications on the World Wide Web and apply them directly to the appropriate manufacturing project. In most instances, PERT diagrams are available to eliminate the estimating process and make PERT a more useful and convenient tool. Clearly, PERT is a manufacturing-based project planning and scheduling network. In many instances, managers have attempted to apply PERT principles to other types of projects. It divided the construction projects with repetitive activities into two categories: linear and non-linear projects; in linear projects, the projects are composed of a number of typical stages with identical activities of the same duration repeated consecutively from one unit to the next. Several techniques were developed for projects with discrete units, such as floors, houses, and offices, and were developed for repetitive projects, such as Line Of Balance (LOB) and Vertical Production Method (VPM). The Line Of Balance (LOB) management control technique collected, measured, and analyzed data to show the progress, status, and timing of production projects. It was introduced at Goodyear Tire and Rubber Company in 1941 and fully utilized during World War II in the defense industry. In the next sections, the proposed Repetitive-Projects Evaluation and Review Technique software (RPERT), which calculates the completion probability of the repetitive construction projects with a specified/certain duration of identical activity (contract duration) by using Line Of Balance technique (LOB) in case of single or multiple number of crews integrated with Program Evaluation and Review Technique (PERT) and is applied on a small illustrative example to facilitate the use of the proposed software and demonstrate its capabilities. The performance of the proposed software is compared with "manual solution" of the same illustrative example. Conclusion is summarized at the end of the paper.

Tools like CPM and PERT for project planning had attracted the attention of both the practitioners and researchers. Determination of the job completion time in PERT networks is important for planning and bidding purposes. The complexity involved in accurately determining job completion time has led to development of many approximating procedures. Most of them ignore the dependence between paths in the network. Mehtotra et al. [3] proposed an approximation to determine job completion time which explicitly recognizes this dependency. Dependency in networks arises due to commonality of activities among various paths in the network. They developed an approximation which is simple in use and makes use of readily available tables. Also, the approximation employed the traditional concept of the critical path which is easy to understand and to operationalize. The activities on the critical path are divided into independent and dependent portion; the dependent portion comprises common to various critical path. Order statistics were used in computing the time for the

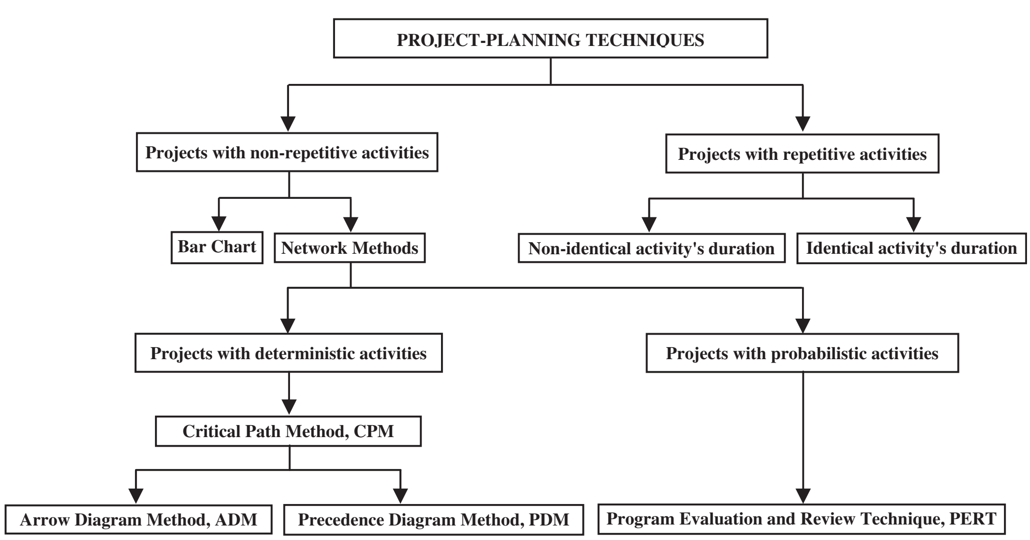

Figure 1 Taxonomy of the existing planning techniques for construction projects.

dependent portion of the critical path. Mehtotra et al. [3] presented the theoretical underpinnings of their approach and illustrated its application via an example for only normal and exponentially distributed activity durations; the approach can be extended to any underlying activity distribution. In absence of other measures, they used simulation results as a proxy for job completion time and as a benchmark to measure the accuracy of their approximation. They showed that the distributions of the job completion time are better approximated by a mixture of distributions and their approximation yields estimates for the job completion time which are closer to the simulation results. More than half a century after the debut of CPM and PERT, planners still lack a project scheduling system with calibrated and validated distributions and without requiring complex user input. Modern Decision Support Systems (DSS) for project management are more sophisticated and comprehensive than PERT/CPM. Nonetheless, in terms of stochastic analysis, they show insufficient progress. Trietsch and Kenneth [4] offered PERT 21 as a radically different stochastic analysis for projects, based on relevant and validated theory. It is designed to enhance existing DSS, and thus, it can be implemented without sacrificing the investment already made in project management systems. Finally, regarding the important sequencing and crashing, models were developed under CPM, PERT 21, permitting their adaptation to stochastic reality. Mouhoub et al. [5] presented a new approach for generating PERT networks with minimum number of dummy arcs. Indeed, by applying the seven rules in sequential order, the algorithm reduces certain number of dummy arcs until the last rule which provides the minimal PERT network with the total respect of the constraints in schedule table and by using some results of line graphs. The algorithm was programed in c^{++} Builder. This new approach is very simple to be applied and gives minimal PERT in a number of dummy arcs within very short time. Another major benefit that is worth noting is the fact that the algorithm works without any problem in the presence of transitive arcs. Haga and Marold [6] explored the suitability of a new method to crash PERT networks. A simulation program was written in the c^{++} programming language and run on three hypothetical data sets. Three representative PERT networks were used. For all data sets, the new method significantly reduced the mean total (crashed plus overrun) cost for the project. The saving varied from 60.6% to 56.2% over the traditional PERT method. Pontrandolfo [7] analyzed the problem of scheduling projects with activities characterized by probabilistic duration. A particular perspective had been adopted focusing on the ways in which a given project may evolve, namely dates and sequences of events (PERT-paths) occurring from project start-up to completion. Based on the PERT-path technique, an analytical approach has been proposed to compute the project duration of stochastic networks. The required input data are duration (in terms of mean and variance) and occurrence probability of every path. Therefore, the equations to compute mean and variance of the project duration from the path data are the key issue of the methodology. Such equations have been derived and applied on two case examples. The technique accuracy has been varied against the PERT conventional approach and simulation. Pillai and Tiwari [8] developed a new technique using the finalized PERT as its point of departure called Programme Analysis, Control and Evaluation (PACE); this enhanced PERT is very effective where PERT becomes inadequate in dealing with ill structured, ambiguous problems. PACE is complementary to PERT and had been proved to be an effective project-management tool in the Indian projects. Participative technique such as PACE is essential in projects in which: (1) There is uncertainty about the goals; and (2) There is uncertainty about the method of achieving the goals. The PACE meetings can be used to monitor and control the definition of these milestones, the specification of the deliverables to be achieved at these points, and the work to be undertaken to achieve these milestones. Williams [9] used simulation to illustrate the nature and extent of PERT approximation errors in simple examples from two excellent texts. The examples raise serious questions about the utility of PERT project duration estimates and suggest opportunities for improvement in introductory PERT instruction. Simulation is clearly the most appropriate method for assessing project duration, most introductory discussions touch on these issues and move quickly to standard approximations, implying that PERT offers useful, if only approximate, project duration estimates. Baradaran et al. [10] presented a metaheuristic algorithm for Resource-Constrained Project Scheduling Problem (RCPSP) in PERT networks. A PERT-type project, where activities require resources of various types with random duration, is considered. The resource project scheduling model is an NP-hard, therefore to obtain a precise solution; a metaheuristic algorithm is suggested namely Hybrid Scatter Search (HSS). Results from project completion time were provided from Monte Carlo (MC) simulation runs. In order to validate the performance of new hybrid metaheuristic algorithm, solutions are compared with "optimal solution" for small networks. Also, the efficiency of the proposed algorithm, for real world problems, in terms of solution quality, is compared with well-reported benchmark test problems. The computational results reveal that the proposed algorithm had appropriate results for small networks and real world problems. Chretienne and Sourd [11] found that the problem of scheduling dependent activities with no resource limitations and arbitrary convex cost functions has been considered. A generic algorithm had been designed for the general case. An efficient polynomial algorithm had been designed for the special case. Velasco et al. [12] mentioned the difficulties in interpreting the parameters of beta distribution and let researchers suggest the Program Evaluation and Review Technique (PERT). They provided an alternate for the PERT variance expression addressing a concern raised by Hahn [13] regarding the constant PERT variance assumption given the range for an activity's duration, while retaining the original PERT mean expression. Moreover, this approach ensures that an activity elicited most likely value aligns with the beta distribution mode. While this was the original intent of researchers, their method of selecting beta parameters via the PERT mean and variance is not consistent in this manner. Kumar et al. [14] developed a new approach that was simple and key to keep the printing firms competitive within competitive environment by supporting better decisions in production serving in time to the market. Estimating the stochastic job completion time by PERT uses approximate calculation of standard deviation σ and the activity expected time T_{exp}. The project scheduled time T_{sch} is greatly influenced by standard deviation. The most popular N-method of standard deviation adopted in calculating the T_{sch} in two conditions where $T_{\min}$ and $T_{\max}$ are closer to T_{mean} and in extreme time estimate condition to $T_{\min}$ and

T_{max}. Permutation and combinations of the T_{min}, T_{mean}, and T_{max} are used to calculate T_{sch} of the project by software developed for such purpose. The T_{sch} obtained in these three methods studied with the actual projects completion time over a period of time [14]. Azaron et al. [15] developed a multi-objective model for resource allocation problem in PERT networks with exponentially or Erlang distributed activity durations, where the mean duration of each activity is a non-increasing function and the direct cost of each activity is a non-decreasing function of resource amount allocated to it. The decision variables of the model are the allocated resource quantities. The problem is formulated as a multi-objective optimal control problem that involves four conflicting objective functions. The objective functions are the following: the total direct costs of the project (to be minimized), the mean of project completion time (min), the variance of project completion time (min), and the probability that the project completion time does not exceed a certain threshold (max). The surrogate worth trade-off method is used to solve a discrete-time approximation of the original problem. Yaghoubi et al. [16] proposed a multi-objective model to optimally control the resources allocated to the service stations in a dynamic PERT network with finite capacity. In model, the total direct costs of service stations per period was considered as the first objective and the mean project completion time in the steady-state as the second one, both to be minimized. Moreover, the probability that the system is empty in the steady-state was considered as the third objective function to be minimized as well. The dynamic PERT network was represented as a network. It was also assumed that the number of servers in each service station to be one, while the service times (activity durations) are independent random variables with exponential distributions. Finally, they used goal attainment technique to solve a discrete-time approximation of the original continuous-time problem. Castro et al. [17] defined a new rule for the resolution of slack allocation problem in a PERT network. This problem of allocating existing extra time exists in some paths among the activities belonging to those paths. The allocation rule that they proposed assigns extra time to the activities proportionally to their durations in such a way that no path duration exceeds the completion time of the whole project. This time allocation enabled them to make a schedule for the PERT project under study. They gave two characterizations of the rule and compared it with others. Azaron and Moghaddam [18] developed a multi-objective model for the time–cost trade-off problem in a dynamic PERT network using an interactive approach. The activity durations exponentially distributed random variables, and the new projects are generated according to a renewal process and share the same facilities. Thus, these projects cannot be analyzed independently. This dynamic PERT network is represented as a network of queues, where service times represent the durations of corresponding activities and the arrival stream to each node follows a renewal process. At the first stage, they transformed the dynamic PERT network into a proper stochastic network and then computed the project completion time distribution by constructing a continuous-time Markov chain. At the second stage, the time–cost trade-off problem is formulated as a multi-objective optimal control problem that involves four conflicting objective functions. Then, any method is used to solve a discrete-time approximation of the original problem. Finally, the proposed methodology is extended to the generalized Erlang activity durations. Ghaleb et al. [19] developed a method for investigating the application of mathematical programming to the concept of crashing in Program Evaluation and Review Technique (PERT). The main objective is the minimization of the pessimistic time estimate in PERT networks by investing additional amounts of money in the activities on the critical path. The constructed mathematical model, which is built in terms of additional amounts of money that must be invested, shows that minimizing the pessimistic time decreases project duration and, at the same time, reduces its variance. The result of applying the model showed that the probability of realizing the terminal node is increased. Castro et al. [20] defined the weighted serial cost sharing rule for the cost allocation problem. They applied this new rule to the problem of sharing delay costs in a PERT network. Furthermore, they presented a characterization of this rule and a polynomial algorithm for its calculation. Wenying and Xiaojun [21] aimed at the deficiency of the traditional PERT method in the risk assessment for engineering project progress; they improved PERT method, so as to make it more suitable for engineering project risk assessment of spliced network. Firstly, calculated project activity duration is more approximate to the realistic one. Meanwhile, they took advantage of equivalent-weight probability method to revise the main path, so that the calculated project duration is more suitable for the realistic situation. Finally, they demonstrated the feasibility of the method by an engineering project case study. PERT is a widely utilized framework for project management. However, as a result of underlying assumptions about activity times, the PERT formulas prescribe a light-tailed distribution with a constant variance conditional on the range. Hahn [13] provided a distribution which permits varying amounts of dispersion and greater likelihoods of more extreme tail-area events that is straightforward to implement with expert judgments. Moreover, the distribution can be integrated into the PERT framework such that the classic PERT results represent an important special case of the presented method. Xiangxing et al. [22] investigated Programming Evaluation and Review Technique networks with independently and generally distributed activity durations. For any path in this network, they selected all the activities related to this path such that the completion time of the sub-network (only consisting of all the related activities) is equal to the completion time of this path. They used the elapsed time as the supplementary variables and modeled this sub-network as a Markov skeleton process; the state space is related to the sub-network structure. Then, they also used the backward equation to compute the distribution of the sub-network's completion time, which is an important role in project scheduling. Banerjee and Paul [23] said most studies of project time estimation assume that (a) activity times are mutually independent random variables; many also assume that (b) path completion times are mutually independent. They subjected the impact of both such assumptions to close scrutiny. Using tools from multivariate analysis, they made a theoretical study of error direction in the classical PERT method of estimating mean project completion time when correlation is ignored. They also investigated the effect of activity dependence on the normality of path length via simulation. Seal [24] developed the implementation of the traditional PERT/CPM algorithm for finding the critical path in a project network as a spreadsheet. The problem is of importance due to the recent shift of attention to using the spreadsheet environment as a vehicle for delivering techniques to end-users. Azaron and Ghomi [25] applied the stochastic dynamic programming to obtain a lower bound for the mean project completion time in a PERT network, where the activity durations exponentially distributed random variables. Moreover, these random variables are non-static as the distributions themselves vary according to some randomness in society like strike or inflation. This social randomness was modeled as a function of a separate continuous-time Markov process. The results were verified by simulation. Lu and AbouRizk [26] proposed simplified CPM/PERT simulation model; it was validated through the comparison with classic CPM/PERT analysis and was proved that both are simple and robust. This new solution to CPM network analysis can provide project management with a convenient tool to assess alternative scenarios based on computer simulation and risk analysis. Cottrell [27] developed and tested a simplified version of the Program Evaluation and Review Technique (PERT) for project planning. The simplification was to reduce the number of estimates required for activity durations from three, as in conventional PERT, to two. This is accomplished by applying the normal distribution, rather than the beta, to activity duration. The two required duration estimates are the "most likely" and the "pessimistic." These modifications reduce the level of effort needed to apply PERT. By analyzing 12 project networks, the simplified PERT produced values similar to those of conventional PERT for durations, variances, and probabilities. Haga and Tim [28] created a computer simulation model to determine the order in which activities should be crashed as well as the optimal crashing strategy for a PERT network to minimize the expected value of the total (crash + overrun) cost, given a specified penalty function for late project completion. This work was initially explored by Haga [29] as part of his unpublished Doctoral dissertation. Premachandra [30] proposed new approximations for the mean and the variance of activity based on "pessimistic," "optimistic," and "most likely" time estimates. By numerical comparison with actual values, the proposal was shown as more accurate than the original PERT estimates and its modifications. Another advantage of the proposed approximation is that it requires no assumptions about the parameters of the beta distribution. Ghomi and Teimouri [31] presented a new exact formula to compute the Path Critical Index (PCI) and Activity Critical Index (ACI) for the PERT network with any structure. It is assumed that duration time for each activity is a discrete random variable. Numerical examples presented to prove that the method is highly efficient and had excellent results. Khan [32] addressed an EXemplary Program Evaluation and Review Technique (EXPERT) using evaluation of outcomes and implementation of projects quality. Zhong and Zhang [33] presented a new path float concept different from that of traditional CPM, which calculates the non-critical path float in the PERT, to copy with the uncertainties within the network implementation and to reduce the misleading information. An example network was analyzed with the new method; the results showed the consistent path float under required completion probability and required duration. The new path float concept brought useful planning information to managers and planners in the construction. Shankar and Sireesha [34] proposed an improvement to Ginzberg [35] approximation for the mean and variance of a PERT activity time, and improvement was proposed by means of reasonable assumption. By generalizing the assumption on parameters in original PERT, an approximation for the mean and variance of a PERT activity duration was proposed, and by comparison with numerical case, it was shown that the mean and variance of PERT activity duration in this proposed method and original PERT were approximately equal.

4. Employed techniques

According to this research, the following techniques are used for formulating the present model: (1) Precedence Diagram Method, "PDM" is used to represent each stage of the project; (2) For each activity (k), (where $k = 1, 2, \ldots, K$) in the typical-repetitive network, Line Of Balance, "LOB" is used to represent the activity schedule per all stages in project time plan; (3) Transformation from the traditional LOB to modified CPM must be done in the calculations; (4) Program Evaluation and Review Technique (PERT) is used to find expected completion probability of any repetitive construction project with a specified/certain duration of identical activity; (5) Each activity (k) (where $k = 1, 2, \ldots, K$) has a time buffer ($TB_{k,kk}$), per each stage (s) (where $s = 1, 2, \ldots, S$), between the completion time of the activity (k) and the start time of each following activity (kk) in the network; and (6) Any two sequential activities may have a stage buffer ($SB_{k,kk}$), of a specific number of stages at any time to meet practical and/or technological purposes, this stage buffer has to be identified by the planner for these activities.

5. Considered assumptions

As shown in the following discussion which assumed that: (1) No idle time is allowed for employed crews; thus, once a crew starts working on an activity at the first stage, it will continue working with the same production rate until finishing the work at the last stage; (2) The estimation of each activity duration for the completion is important in the network analysis; this can be done using three possible assumptions: (I) Most optimistic duration (D_{min}); (II) Most pessimistic duration (D_{max}); and (III) Most likely or normal duration (D_{nor}); (3) A constant average duration is set for the same activity per all stages to maintain a constant production rate. If an activity duration needs to be changed to meet a particular feasible project duration, then an equal change must be made to the activity duration per all stages; (4) Single or multiple number of crews is used in employed technique; (5) The learning phenomenon, whereby the actual duration of an activity is reduced as repetition increases, is neglected; (6) The work on each activity is conducted by one unit at a time per all stages; and (7) The project under study is not disturbed by incidents during constructing.

6. Mathematical formulations

First phase, the needed formulations, which are used for calculating total activity duration that takes into consideration activity quantity with related maximum, minimum, and normal production rates of the studied activity into studied construction project per stage, is shown in the following equations:

$$D_{min} = BQ \div PR_{max} \tag{1}$$

$$D_{max} = BQ \div PR_{min} \quad (2)$$

$$D_{nor} = BQ \div PR_{nor} \quad (3)$$

$$T_e = (D_{min} + D_{max} + 4 \times D_{nor}) \div 6 \quad (4)$$

$$\sigma = (D_{max} - D_{min}) \div 6 \quad (5)$$

where D_{min}, D_{max}, and D_{nor} are the optimistic, pessimistic, and most likely durations by (days) for each activity into studied construction project; BQ is budget quantity by (units) of studied activity; PR_{max}, PR_{min}, and PR_{nor} are the optimistic, pessimistic, and most likely production rates by (units/day) of studied activity into studied construction project; T_e is equivalent duration by (days) for each activity into studied construction project; σ is standard deviation for each activity into studied construction project; FS $_{(1\&2)}$ is the time lag by (days) between finish of predecessor activity to start of successor activity into any path of studied construction project network; $SS_{(1\&2)}$ is the time lag by (days) between start of predecessor activity to start of successor activity into any path of studied construction project network; FF $_{(1\&2)}$ is the time lag by (days) between finish of predecessor activity to finish of successor activity into any path of studied construction project network; and SF $_{(1\&2)}$ is the time lag by (days) between start of predecessor activity to finish of successor activity into any path of studied construction project network.

Eqs. (1)–(3) are used to estimate the optimistic, pessimistic, and most likely durations for each activity per stage, Eq. (4) is used to find the equivalent activity duration per stage, while Eq. (5) is used to find the standard deviation for each activity per stage and Fig. 2 shows that.

Second phase, the needed formulations, which are used for calculating total activity duration that takes into consideration the activities repetition of the studied construction project, is shown in Eqs. (6) and (7) as follows:

$$T_e^* = T_e \times \{1 + [(N - 1) \div (NC)]\} \quad (6)$$

$$\sigma^* = N \times \sigma \quad (7)$$

where T_e^* is the modified equivalent duration (in days) for each activity per all stages as one unit; N is the number of repetitive stages into studied construction project; NC is the number of crews; and σ^* is the modified standard deviation of studied activity per all stages as one unit.

Third phase, the needed formulations, which are used for calculating total project duration that takes into consideration the activities repetition of the studied construction project and are shown in Eqs. (8)–(16) as follows:

$$TB_{(1\&2)} \geq FS_{(1\&2)} \quad (8)$$

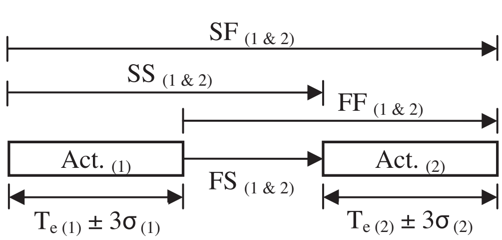

Figure 2 Sample of network per single stage.

$$TB_{(1\&2)} \geq SS_{(1\&2)} - T_{e(1)} \quad (9)$$

$$TB_{(1\&2)} \geq FF_{(1\&2)} - T_{e(2)} \quad (10)$$

$$TB_{(1\&2)} \geq SF_{(1\&2)} - T_{e(1)} - T_{e(2)} \quad (11)$$

$$FS^*_{(1\&2)} \geq TB_{(1\&2)} + T_{e(1)} - T^*_{e(1)} \quad (12)$$

$$FS^*_{(1\&2)} \geq TB_{(1\&2)} + T_{e(2)} - T^*_{e(2)} \quad (13)$$

$$FS^*_{(1\&2)} \geq T_{e(1)} \times SB_{(1\&2)} - T^*_{e(1)} \quad (14)$$

$$FS^*_{(1\&2)} \geq T_{e(2)} + T_{e(2)} \times SB_{(1\&2)} - T^*_{e(2)} - T_{e(1)} \quad (15)$$

$$P_{D\ mean} \geq \sum_{k=1}^{k=k} \left\{ T^*_{e(k)} + FS^*_{(k\&k+1)} \right\} \quad (16)$$

where $TB_{(1\&2)}$ is the time buffer by (days) between two sequential activities from finish of predecessor activity to start of successor activity into any path of studied construction project per stage; $SB_{(1\&2)}$ is the stage buffer by (stages) between the starts of two sequential activities into any path of studied construction project per all stages; $FS^*_{(1\&2)}$ is the modified finish to start between two sequential activities from finish of predecessor activity to start of successor activity into any path of studied construction project per all stages as a single unit; and $P_{D\ mean}$ is the mean project duration by (days) per all stages for every network path (from first activity in the path $k = 1$ to last activity in the path $k = K$), and Fig. 3 shows that.

Last phase, the needed formulations, which are used for calculating the expected completion probability of the studied construction project at specified/certain duration that takes into consideration the activities repetition, is shown in Eqs. (17)–(21) as follows:

$$P_{SD} \geq \left\{ \sum_{k=1}^{k=k} (\sigma^*_k)^{\wedge 2} \right\}^{\wedge 0.5} \quad (17)$$

$$P_{D\ min} = P_{D\ mean} - (3 \times P_S) \quad (18)$$

$$P_{D\ max} = P_{D\ mean} + (3 \times P_S) \quad (19)$$

$$P_{D\ max} \geq P_{D\ required} \geq P_{D\ min} \quad (20)$$

$$Z = \left(P_{D\ required} - P_{D\ mean}\right) \div P_{SD} \quad (21)$$

where P_{SD} is the standard deviation of studied construction project per all stages for every network path (from first activity in the path $k = 1$ to last activity in the path $k = K$); $P_{D\ min}$ is the minimum duration of studied construction project per all stages as a single unit; $P_{D\ max}$ is the maximum duration of studied construction project per all stages as a single unit; $P_{D\ required}$ is the required duration of studied construction

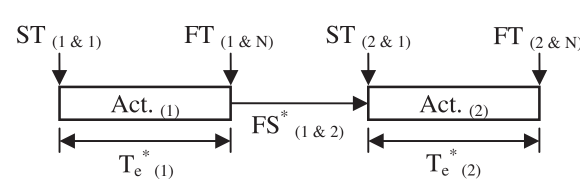

Figure 3 Sample of network per all stages.

project per all stages as a single unit (Contract duration); and Z is the difference length between required and mean duration of studied construction project per all stages as a sinlgle unit by using standard deviation of studied construction project steps per all stages as a single unit.

Finally, by using both of Z "Eq. (21)" and area tables under normal distribution curve, the planner can get the expected completion probability of any repetitive construction project within a specified/certain duration or by using both of the expected completion probability of any repetitive construction project within a specified/certain duration and area tables under normal distribution curve, the planner can get Z "Eq. (21)" and required contract duration that satisfy this completion probability by using Line Of Balance technique (LOB) in case of single or multiple number of crew integrated with Program Evaluation and Review Technique (PERT), which is called Repetitive-Projects Evaluation and Review Technique (RPERT).

7. Software implementation

As shown in Fig. 4, the proposed software (RPERT) is designed by using java programming code to provide a number of new and unique capabilities and is implemented in four major modules including: (1) A user interface module to facilitate inserting the input of project data and visualizing the output data. The present user interface module is designed to implement the necessary interface functions in two main phases: (a) An input phase that facilitates the input of project data details, project activities, activities relations, and contract duration; and (b) An output phase that allows the user to view the expected project completion probability within a specified/certain duration. (2) A database module to facilitate data storage. The main purpose of this module is to develop a relational database capable of storing necessary input data (e.g., project data details, project activities, activities relations,

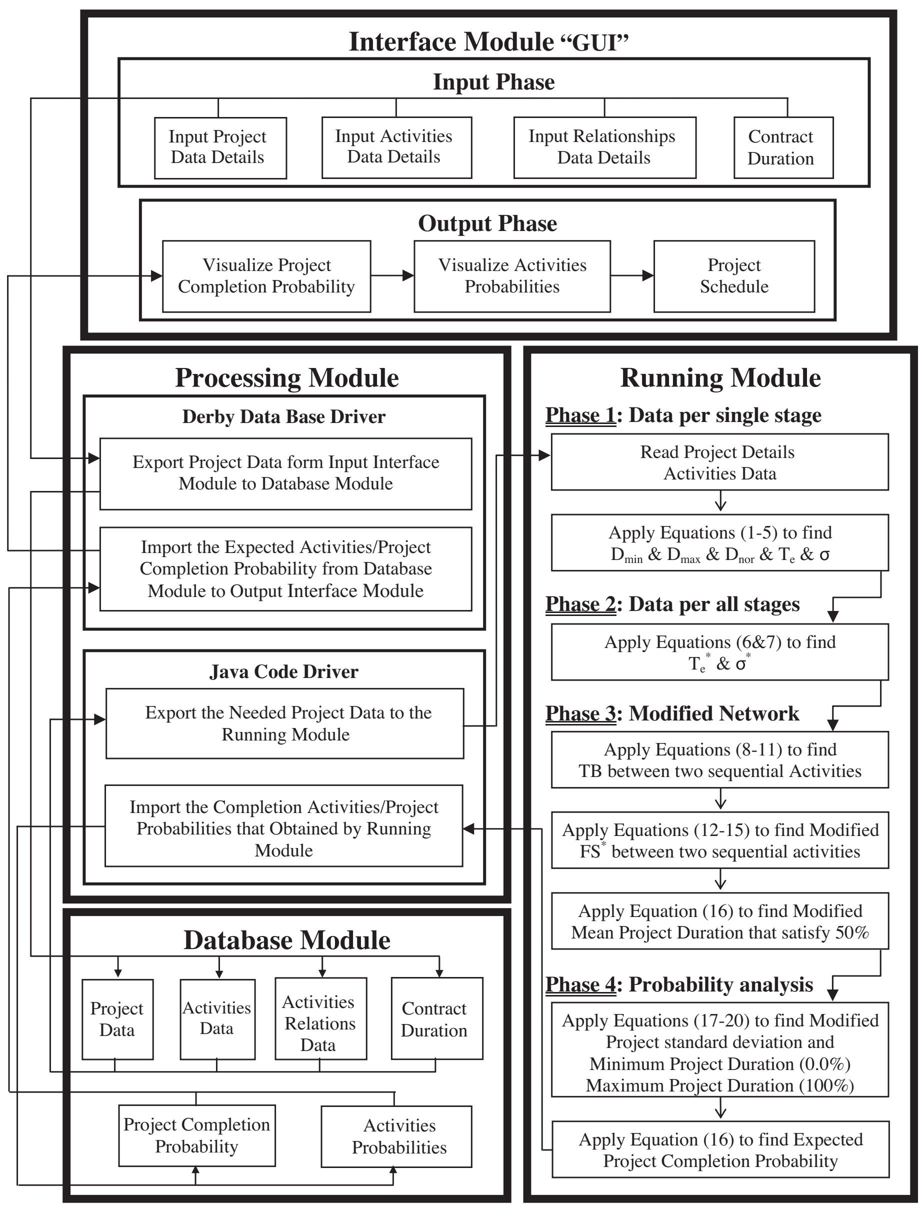

Figure 4 **RPERT** main modules.

contract duration) and storing produced output data (e.g., expected project completion probability within a specified/certain duration). This module is composed of main groups that are designed to store the following construction planning details: (a) Project data; (b) Holidays data; (c) Exceptions data; (d) Activities data; (e) Relationships among activities data; and (f) Contract data specially contract duration. The present database module is developed using Derby database management system by Java programming code to facilitate its integration with the remaining modules of RPERT using processing module; (3) A running module can be defined as a class of programming code system especially java coding and its applications that is designed to allow different calculation runs such as: (a) Activity optimistic, pessimistic, and most likely durations per stage for each activity into studied construction project; (b) Activity equivalent duration per stage for each activity into studied construction project; (c) Activity standard deviation per stage for each activity into studied construction project; (d) Activity equivalent duration per all stages as a single unit for each activity into studied construction project; (e) Activity standard deviation per all stages as a single unit for each activity into studied construction project; (f) Modified finish to start relationship between two sequential activities per all stages as a single unit for each relationship into studied construction project; (g) Mean project duration; (h) Minimum project duration; (i) Maximum project duration; and (j) Expected completion probability of the studied project within a specified/certain duration (contract duration); and (4) A processing module can be defined as a class of programming code system especially java coding and its applications that is designed to communicate and exchange data from available modules with a seamless integration. The present processing module is developed in RPERT to enable: (a) The Java programming code driver; and (b) The Derby data base driver. First, the java programming code driver is utilized to perform two main functions: (i) Export data from database module to running module; and (ii) Import the generated result from running module to database module. Second, the derby driver is used in RPERT to perform two main functions: (i) Export the existing project data from input interface module to database module; and (ii) Import the solution data from the database module to output interface module. The main data transferred using the two drivers in the present processing module are the inserted data in RPERT using a newly developed user interface module.

8. Verification of software

This section presents the results of using an illustrative example, which is solved manually and analyzed by RPET for making a comparison between manual result and RPERT result to test the software, to illustrate the use of present software, and to demonstrate its capabilities. The main objective of these results, related to the present system, is to provide expected completion probabilities for any typical-repetitive construction projects within a specified/certain duration as stakeholders need. Input data for single stage of illustrative example are shown in Fig. 5 and Table 1; the project example consists of four activities with different relationships among them. These activities have three mode options, each mode has its own production rate (units/day) using a single number of crew, while the activities data for each mode option take into consideration single stage from studied example as shown in Table 1. Number of repetitive stages of analyzed example is equal to five typical stages; Project start date is Tuesday 1st October 2013; weekends are Fridays and Sundays; holidays are Wednesday 9th October 2013 and Tuesday 12th November 2013; and exceptions are Friday 11th October 2013 and Sunday 10th November 2013. It is required to find the (1) Expected completion probability for studied project at Tuesday 31th December 2013; (2) Expected completion probability for studied project at 60 working days; and (3) Expected completion date that satisfies probability is equal to 90%.

Tables 2,3 and Fig. 6 show the application of Eqs. (1)–(16); then the mean project duration is equal to 62.4 working days. By applying Eqs. (17)–(20), the project standard deviation is equal to 10.5 days; the range between

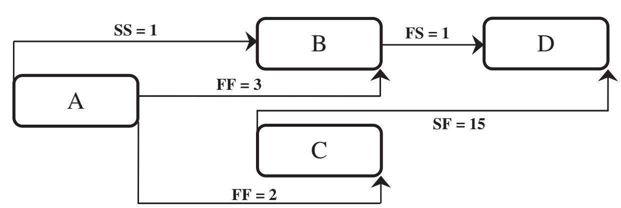

Figure 5 Network of illustrative example.

Table 1 Available project activities data and mode options per stage.

Activity ID	Depends on	Relation type	Lag value	Stage buffer	Mode options	Quantity (units)	Production rate (units/day)
A	–	–	–	–	1	1000	500
					2		400
					3		200
B	A	SS	1	2	1	2000	600
		FF	3		2		400
					3		300
C	A	FF	2	1	1	500	100
					2		50
					3		30
D	B	FS	1	0	1	200	80
	C	SF	15	3	2		40
					3		40

Table 2 Analyzed project activities data.

Activity ID	D_{min} (days)	D_{nor} (days)	D_{max} (days)	T_e (days)	T_e^* (days)	σ	σ^*
A	2.0	2.5	5.0	2.8	14.0	0.5	2.5
B	2.0	5.0	6.7	4.8	24.0	0.8	3.9
C	5.0	10.0	16.7	10.3	51.5	2.0	10.0
D	2.5	5.0	5.0	4.6	23.0	0.4	2.0

Table 3 Modified project activities relations.

Activity ID	A	B	C	D
A		TB = −1.80 FS* = −8.40	TB = −8.30 FS* = −11.2	– –
B	TB = −1.80 FS* = −8.40		– –	TB = +1.00 FS* = −17.4
C	TB = −8.30 FS* = −11.2	– –		TB = +0.10 FS* = −14.6
D	– –	TB = +1.00 FS* = −17.4	TB = +0.10 FS* = −14.9	

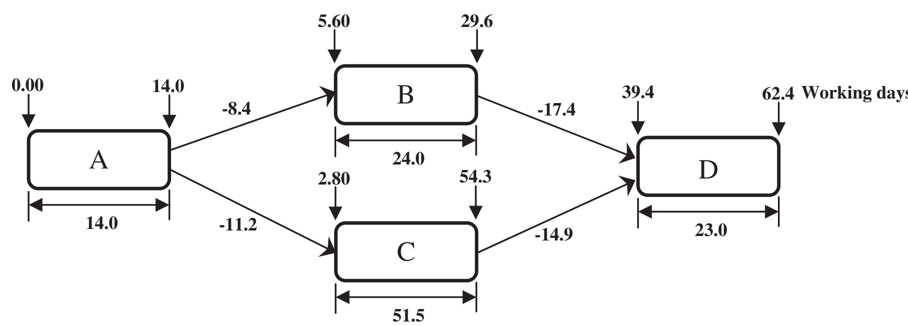

Figure 6 Solved network of illustrative example.

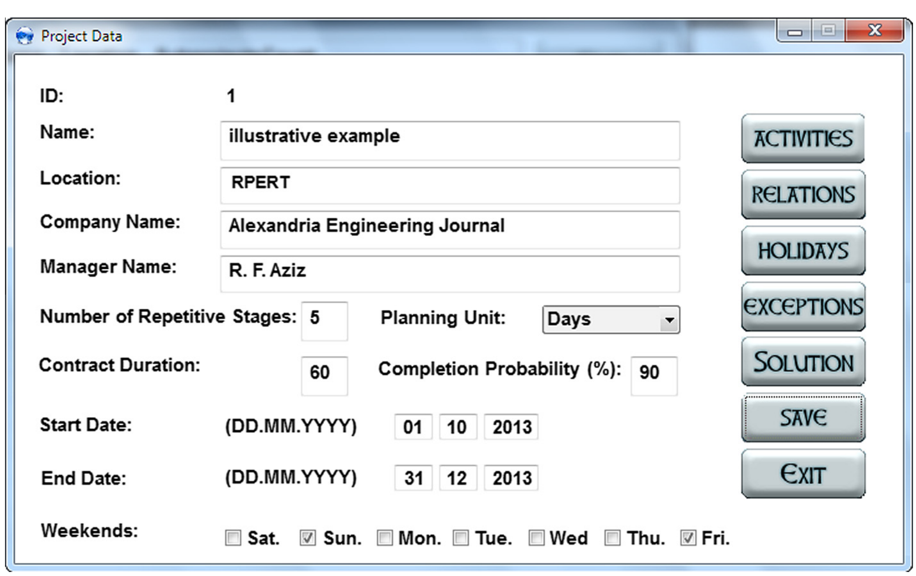

Figure 7 Project details form.

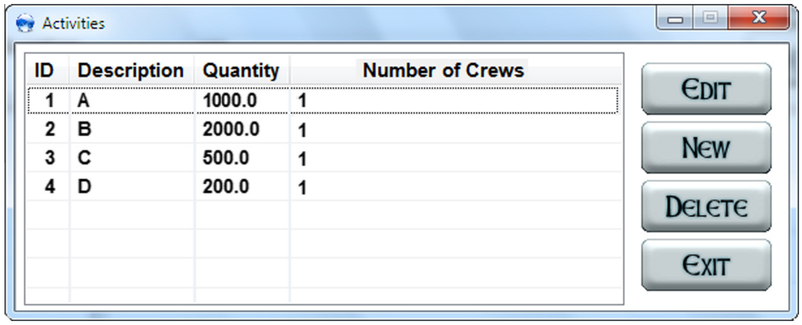

Figure 8 Project activities form.

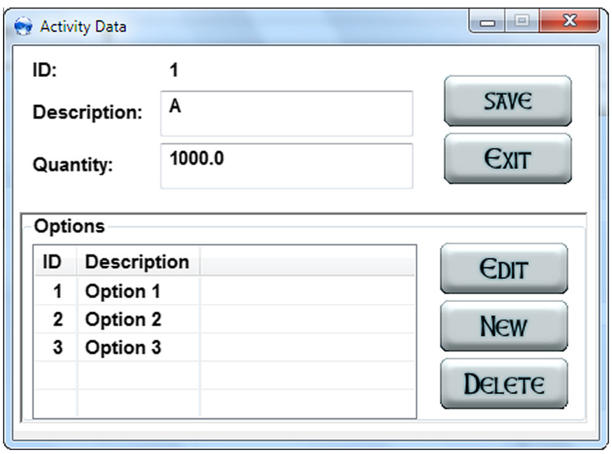

Figure 9 Project activities details form.

minimum and maximum project duration are equal to 30.9 and 93.9 working days, respectively. From Eq. (21), the planner can get the (1) Z that is equal to 0.34 with corresponding expected completion probability equals to 63.3%; (2) Z that is equal to -0.23 with corresponding expected

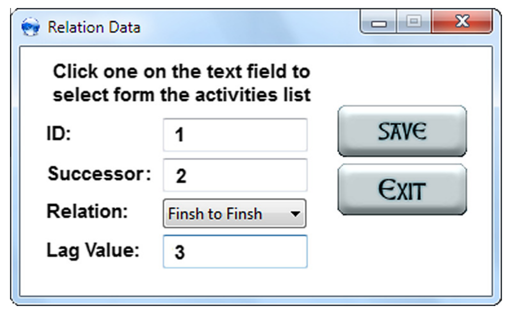

Figure 11 Project activities relationships details form.

completion probability equals to 40.9%; and (3) Expected completion probability that is equal to 90% with corresponding Z equals to 1.28 and required duration (contract duration) is equal to 75.8 working days or completion date will be at Tuesday 14th January 2014 as are required respectively.

Inserting the input data by using RPERT input wizards as shown in Figs. 7–14, while Fig. 15 shows the output wizard that finalizes the number of expected probabilities solutions for completion of repetitive construction project

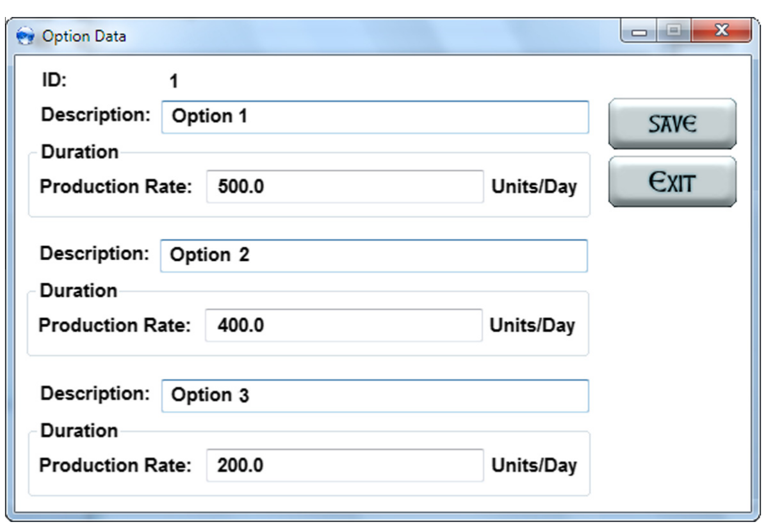

Figure 10 Project activities options form.

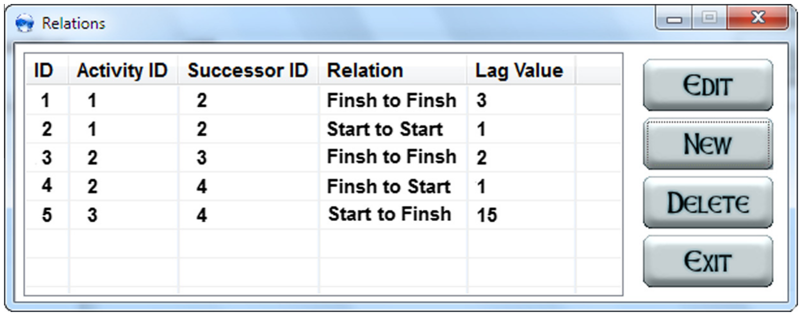

Figure 12 Project activities relationships form.

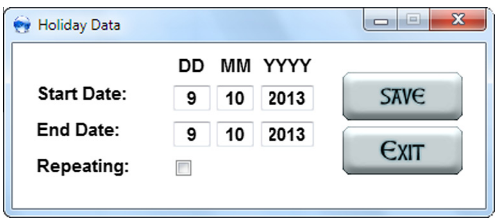

Figure 13 Project holidays details form.

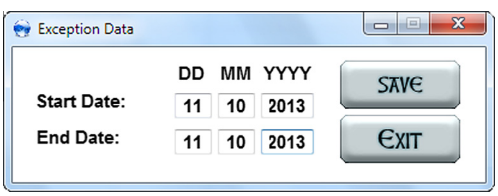

Figure 14 Project exceptions details form.

according to completion sets of specified durations in the analyzed project.

9. Conclusion

Author proposed the development of simplified software called Repetitive-Projects Evaluation and Review Technique (RPERT), which is processed by Program Evaluation and Review Technique (PERT) integrated with Line Of Balance technique (LOB) for repetitive construction projects with identical activities in order to find the expected completion probability within a specified/certain duration (contract duration). It was developed in four main modules: (1) A user interface module to facilitate inserting the input of project data and visualizing the output expected project probabilities solutions; (2) A database module to facilitate data storage and retrieval of data; (3) A running module can be defined as a class of programming code system and is designed to allow different calculation runs; and (4) A processing module can be defined as a class of programming code system, especially java coding and its applications that is designed to communicate and exchange data from available modules with a seamless integration. RPERT was designed by java programming code system to provide a number of new and unique capabilities, including: (1) Viewing the expected project completion probability according to a set of specified durations per each identical activity (optimistic time, most likely time, and pessimistic time) in the analyzed project; and (2) Providing seamless integration with available project

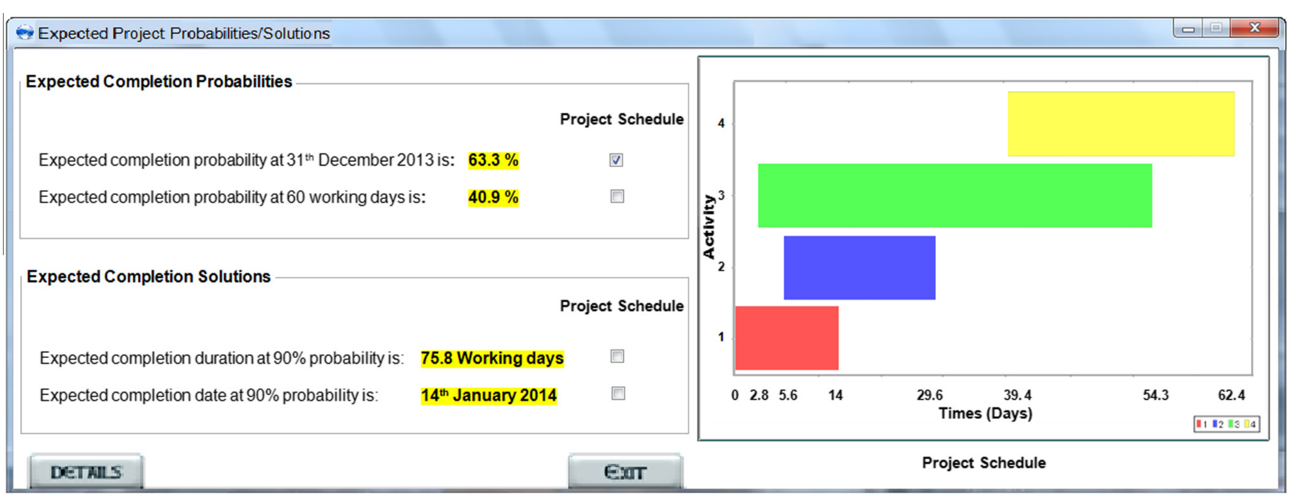

Figure 15 Expected project probabilities solutions form.

time calculations. An illustrative example was presented to demonstrate and verify the applications of proposed software (RPERT), by using probabilistic calculations for repetitive construction projects.

References

[1] F. Davidson, The new project management: tools for an age of rapid change, complexity, and other business realities, San Francisco, John Wiley & Sons, Inc, 2002

[2] C. Ronald, Blueprint for project recovery: a project management guide: the complete process for getting derailed projects back on the track, New York, AMACOM, 2002.

[3] K. Mehtotra, J. Chai, S. Pillutla, A study of approximating the moments of job completion time in PERT networks, Journal of Operations Management 14 (1996) 277–289.

[4] D. Trietsch, B. Kenneth, PERT 21: fitting PERT/CPM for use in the 21st century, International Journal of Project Management 30 (2012) 490–502.

[5] N. Mouhoub, A. Benhocine, H. Belouadah, A new method for constructing a minimal PERT network, Applied Mathematical Modeling 35 (2011) 4575–4588.

[6] W. Haga, K. Marold, A simulation approach to the PERT/CPM time-cost trade-off problem, Project Management Journal 35 (1) (2004) 31–37.

[7] P. Pontrandolfo, Project duration in stochastic networks by the PERT-path technique, International Journal of Project Management 18 (2000) 215–222.

[8] A. Pillai, A. Tiwari, Enhanced PERT for Programme Analysis, Control and Evaluation: PACE, International Journal of Project Management 13 (1) (1995) 39–43.

[9] F. Williams, PERT completion times revisited, INFORMS Transactions on Education 6 (1) (2005) 21–34.

[10] S. Baradaran, S. Ghomi, M. Mobini, S. Hashemin, A hybrid scatter search approach for resource-constrained project scheduling problem in PERT-type networks, Advances in Engineering Software 41 (2010) 966–975.

[11] P. Chretienne, F. Sourd, PERT scheduling with convex cost functions, Theoretical Computer Science 292 (2003) 145–164.

[12] J. Velasco, R. Pleguezuelo, J. Dorp, Revisiting the PERT mean and variance, European Journal of Operational Research 210 (2011) 448–451.

[13] E. Hahn, Mixture densities for project management activity times: a robust approach to PERT, European Journal of Operational Research 188 (2008) 450–459.

[14] A. Kumar, P. Reddy, R. Tagore, Permutation and combinations approach to program evaluation and review technique, ARPN Journal of Engineering and Applied Sciences 2 (6) (2007) 20–26.

[15] A. Azaron, H. Katagiri, M. Sakawa, K. Kato, A. Memariani, A multi-objective resource allocation problem in PERT networks, European Journal of Operational Research 172 (2006) 838–854.

[16] S. Yaghoubi, S. Noori, A. Azaron, R. Moghaddam, Resource allocation in dynamic PERT networks with finite capacity, European Journal of Operational Research 215 (2011) 670–678.

[17] J. Castro, D. Gomez, J. Tejada, A rule for slack allocation proportional to the duration in a PERT network, European Journal of Operational Research 187 (2008) 556–570.

[18] A. Azaron, R. Moghaddam, Multi-objective time-cost trade-off in dynamic PERT networks using an interactive approach, European Journal of Operational Research 180 (2007) 1186–1200.

[19] Y. Ghaleb, A. Abbasi, M. Adnan, Crashing PERT networks using mathematical programming, International Journal of Project Management 19 (2001) 181–188.

[20] J. Castro, D. Gomez, J. Tejada, A polynomial rule for the problem of sharing delay costs in PERT networks, Computers & Operations Research 35 (2008) 2376–2387.

[21] L. Wenying, L. Xiaojunb, Progress Risk Assessment for Spliced Network of Engineering Project Based on Improved PERT, 2011 International Conference on Risk and Engineering Management (REM), Systems Engineering Procedia, vol. 1, 2011, pp. 271–278.

[22] K. Xiangxing, Z. Xuan, H. Zhenting, MARKOV skeleton process in PERT networks, Acta Mathematica Scientia 30B (5) (2010) 1440–1448.

[23] A. Banerjee, A. Paul, On path correlation and PERT bias, European Journal of Operational Research 189 (2008) 1208–1216.

[24] K. Seal, A generalized PERT/CPM implementation in a spreadsheet, INFORMS, Transcations on Education 2 (1) (2003) 16–26.

[25] A. Azaron, S. Ghomi, Lower bound for the mean project completion time in dynamic PERT networks, European Journal of Operational Research 186 (2008) 120–127.

[26] M. Lu, S. AbouRizk, Simplified CPM/PERT simulation model, Journal of Construction Engineering and Management 126 (3) (2000) 219–226.

[27] W. Cottrell, Simplified Program Evaluation AND Review Technique (PERT), Journal of Construction Engineering and Management 125 (1) (1999) 16–22.

[28] W. Haga, O. Tim, Crashing PERT networks: a simulation approach. in: 4th International conference of the Academy of Business and Administrative Sciences Conference Quebec City, Canada, 2001.

[29] W. Haga, Crashing PERT networks, Unpublished Doctoral dissertation, Applied Statistics Department, University of Northern Colorado, Greeley, Colorado, 1998.

[30] I. Premachandra, An approximation of the activity duration distribution in PERT, Computers & Operations Research 28 (2001) 443–452.

[31] S. Ghomi, E. Teimouri, Path critical index and activity critical index in PERT networks, European Journal of Operational Research 141 (2002) 147–152.

[32] H. Khan, Exemplary Program Evaluation and Review Technique (EXPERT) Using Projects, in: 32nd ASEE/IEEE Frontiers in Education Conference, Session T3B, 6–13, 2002

[33] D. Zhong, J. Zhang, New method for calculating path float in Program Evaluation and Review Technique (PERT), Journal of Construction Engineering and Management 129 (5) (2003) 501–506.

[34] N. Shankar, V. Sireesha, An approximation for the activity duration distribution, supporting original PERT, Applied Mathematical Sciences 3 (57) (2009) 2823–2834.

[35] D. Golenko, Ginzburg, On the distribution of activity time in PERT, Journal of the Operations Research Society 39 (1988) 767–771.