SIMULATION AND OPTIMIZATION MODEL FOR ELECTRICAL SUBSTATION CONSTRUCTION

SUBMITTED: February 2016
REVISED: January 2017
PUBLISHED: September 2018 at https://www.itcon.org/2018/11
EDITOR: Ruikar K.

Rania A. Fayed, Ph.D. Candidate,
Department of Construction Engineering, School of Science and Engineering, American University in Cairo;
rfayed@aucegypt.edu

A. Samer Ezeldin, Professor and Chairman,
Department of Construction Engineering, School of Science and Engineering, American University in Cairo;
aezeldin@aucegypt.edu

SUMMARY: *One of the most complex construction projects is electrical substations. Construction of electrical substation includes civil works and electromechanical works. The scope of civil works includes construction of several buildings divided into parallel and overlapped working phases that require variety of resources and are generally quite costly and consume a considerable amount of time. The construction industry is becoming increasingly competitive over the years, whereby the need to continuously find ways to improve construction performance. In order to address the aforementioned adversities, this paper takes the initial steps and presents simulation and optimization model for the execution processes of civil works for an electrical substation using a data base excel file for input data entry. The input data include bill of quantities, resources breakdown, maximum available resources, production rates, unit cost of resources and indirect cost. The model is built on Anylogic software using discrete event simulation method. The model is divided into three zones working in parallel to each other. Each zone includes a group of buildings related to the same construction area. Each zone- model describes the execution process of each building in the zone, the time consumed, percentage of utilization of equipment and manpower crews, amount of materials consumed, total direct and indirect cost. The model is then optimized to mainly minimize the project duration using a parameter variation experiment and a genetic code implemented using Anylogic platform. The models were then tested on a real case study in Egypt. The models prove efficiency by attaining a reduction in model time units between simulation and optimization experiments of 10.25% and reduction in total cost of 4.7%. An analysis to determine the effect of each resource on reduction in cost was presented. However, the model is limited to the scope of civil works, without considering the electromechanical works and the effect of risk factors on time and cost.*

KEYWORDS: *electrical substation, construction, simulation, optimization, genetic algorithm, time, cost.*

REFERENCE: *Rania A. Fayed & A. Samer Ezeldin(2018). Simulation and optimization model for electrical substation construction. Journal of Information Technology in Construction (ITcon), Vol. 23, pg. 215-242, http://www.itcon.org/2018/11*

1. INTRODUCTION

Nowadays the electrical power demand is increasing very rapidly. For fulfilling these huge power demands the modern time requires creation of bigger power generating stations. These power generating stations may be hydro – electric, thermal or nuclear. Depending upon the availability of resources these stations are constructed in different places. These places may not be nearer to load centres where the actual consumption of power takes place. So it is necessary to transmit these huge power blocks from generating station to their load centres. Long and high voltage transmission networks are needed for this purpose. Power is generated comparatively in low voltage level. It is economical to transmit power at high voltage level. Distribution of electrical power is done at lower voltage levels as specified by consumers. For maintaining these voltage levels and for providing greater stability a number of transformation and switching stations have to be created in between generating station and consumer ends. These transformation and switching stations are generally known as electrical substations.

A substation is a high-voltage electric system facility. It is used to switch generators, equipment, and circuits or lines in and out of a system. It also is used to change alternating current (AC) voltages from one level to another, and/or change alternating current to direct current or direct current to alternating current. Some substations are small with little more than a transformer and associated switches. Others are very large with several transformers and dozens of switches and other equipment.

Construction of electrical substation projects includes civil works and electromechanical works. Scheduling and optimization the execution processes of civil works are complex and challenging tasks (König and Beißert, 2009). Construction of electrical substation consists of phases working in parallel to each other and each phase includes a group of buildings. All these buildings share resources at the same time (eg, equipment, labours and raw materials). The selection of resources is the most important part of scheduling and should be considered incongruence with site restrictions and the work to be undertaken (Jaskowski and Sobotka, 2006). As electrical substations projects are unique in nature, the creation of a schedule for construction tasks, should consider an array of conditions such as technological and organizational methods and constraints, as well as the availability or limitation of resource to ensure that a client's needs and requirements in terms of time, cost and quality are met (Zhou el al, 2013).

Substation construction scheduling is concerned with optimally sequencing activities over time and allocating resources accordingly. Construction of substations faces complicated time-cost-resource optimization problems. These multivariate problems require simultaneous minimization of total project duration and total project cost while considering issues related to optimal resource allocation and resource levelling. In other words, substation construction project scheduling should be performed under resource constraints with considering flexibility for time and cost savings through proper resource allocation. However, conventional project scheduling methods and most notably, the critical path method (CPM) have limitations. The limitations are that they do not consider assigning resource availability constraints to project activities or the possibility of time and cost savings through changing project schedules and resource adjustments (Kim and Ellis 2010). The major problem is that the focus of these conventional methods is on local optimality at activity level and not on global optimality at project level; i.e., these methods are not capable of simultaneous time-cost-resource optimization (Ashuri and Tavakolan, 2015).

To overcome this limitation, several simulation and modelling techniques have been developed to solve computer scheduling and optimization of construction projects. Also, the application of evolutionary algorithms like genetic algorithm (GA) and Simulated Annealing (SA) Algorithm as multi-objective optimization methods to solve construction project planning problems (Zheng, et al 2005). The advantage of these new techniques is that they consider the possibility of interrupting project activities and moving shared resources across project activities to simultaneously optimize total project duration, total project cost, and total variation of resource allocation (Ashuri and Tavakolan, 2015).

This paper presents simulation and optimization models for the civil works of an electrical substation using Anylogic software and a data base excel file for input data entry to estimate the time and cost of the substation using each model and determine the differences. The simulation model shows the real time sequence of events that the project would run through to finish all activities of substation using shared resources, depending on their availability in resource pools as constraint, and maintaining different relationships between the activities. The optimization model is created on the same model using parameter variation experiment and genetic algorithm (GA) java code implemented using Anylogic platform. The objective function of optimization model is to minimize project time using resources allocation parameters as decision variables and maximum number of resources available as constraints. The model estimates the cost of the substation after each experiment (simulation and optimization) including direct cost and indirect cost, percentage of utilization of each resource (equipment and labours) and consumed amount of each material. The differences in time, cost and resource allocation between both models are discussed. Also, an analysis was created to present the variation in effect of parameters on project time and cost.

The proposed model has the following distinctive characteristics when compared to previous studies in the literature as follows:

- Event-trigger procedure is used as a way to schedule some action in the model such as condition trigger, time out trigger or rate trigger event. This help in modeling delays and timeouts which will subsequently help in presenting the real time sequence of the process and activities, instead of depending only on the normal activities relationships (predecessors and successors).
- The resources are seized and released by activities through a resource pool procedure, in which the same resources are shared between activities and move inside the project as entities. This procedure provides better resources utilization and prevents resource fluctuations through the project.
- The model accommodates materials supplying schedule, storage procedure and equipment working plan which reflect the real operation process.
- The model effectively uses JAVA parameters types.
- The proposed model utilizes functions with double roles to perform actions or to obtain the result of function calculations.

2. PREVIOUS WORK

A plethora of methods have been propagated to address the construction schedule and cost optimization problem, which can be categorized into four areas: heuristics, mathematical programming, simulation, and genetic algorithms (Bayraktar, 2006).

2.1 Heuristic methods

Heuristic methods are based on the past experience for problem solving. Heuristic methods utilize rules of thumb that generally do not have a mathematical background. Such methods do not guarantee optimal solutions (Feng et al, 2000). Prevalent heuristic methods include Fondahl's method (1961), Structural model (Prager, 1963), Siemens approximation (Siemens, 1971) and structural stiffness (Moselhi, 1993).

Zhang et al (2006) developed a heuristic method for scheduling the multiple-mode repetitive construction project subject to resource constraints. The heuristic algorithm ranks all alternative combinations of activities and selects the one leading to a minimal increase in project duration. Elazouni (2009) proposed a heuristic method for scheduling multiple projects subject to cash constraints. The method determines cash availability during a given period and identifies the schedules for all possible activities as well as the cash requirements for each schedule.

2.2 Mathematical methods

Mathematical programming methods use critical path method, linear programming, integer programming, or dynamic programming to find optimal solutions (Feng et al, 2000). Liu and Li (2003) developed a method to accommodate resource constraints and repetitive scheduling known as the Resource-Activity Critical-Path Method. However, minimization of the overall project cost was not considered in this approach. Kallantzis and Lambropoulos (2004) present a scheduling method for determining the critical path in linear projects, which takes in consideration maximum time and distance constraints in addition to the commonly used minimum time and distance constraints. Gomar et al (2002) illustrated a method that includes optimization of the allocation of multi-skilled work crews by the use of integer linear programming. Moussourakis and Haksever (2004) developed an integer programming tool for use in optimization of linear or discrete cost functions. Elazouni and Gab-Allah (2004) proposed an IP finance-based scheduling method to produce feasible schedules that balance the financing requirements of activities at any period with the cash available during that same period. Robinson (1975) presented a dynamic programming approach to solve time–cost trade-off problems. Moselhi and El-Rayes (1993) proposed a dynamic programming model by introducing a cost variable into the optimization process.

2.3 Simulation methods

Simulation modeling and visualization are increasingly used tools in the field of construction. Simulation modeling consists of creating a model that is the "imitation of the operation of a real-world process or system over time" (Banks et al, 2001). It takes data from the model creator, as input, based on historical data and assumptions. The model itself consists of logical relationships and processes defined by the modeler. The output consists of results based on the interaction of the input data with the designed model. There are three main types of simulation which are discrete-event, agent-based and system dynamics (Zankoul et al, 2015).

A discrete-event simulation (DES) models the operation of a system as a discrete sequence of events in time. Each event happens at a specific moment in time and denotes a change of state in the framework (Robinson, 2004). An agent-based model (ABM) is one of a class of computational models for simulating the activities and cooperation of independent agents, both individual and collective entities, for example, organizations or group, with a perspective to surveying their consequences for the system as a whole (Grimm and Railsback, 2005). System dynamics (SD) is one formal computational dynamic modelling method where the system is represented as stocks, flows and endogenous feedback loops to account for how systems change over time, including accumulations, delays and non-linear interactions (Flynn et al, 2014).

The discrete-event simulation (DES) technique is the most suitable and effective approach for analysing construction operations. The concepts in which the discrete event simulation operation is based on can effectively describe the construction process, which are; entity, event, resource, control element and operation. The entity assigns a unit of movement (an exchange), entities prompt and react to events. An event is an occurrence that progressions the condition of the model. The term resource designates a system element that provides a service (e.g. earth excavation). Usually an entity uses resources. The term control element designates a construct (e.g. logic or constraints in the schedule) based on a system's state. An operation is a step carried out by an entity while it moves through a system (Hossain, 2010).

Ashley (1980) proposed a simulation model using GPSS simulation language for scheduling of construction projects that adopts a queuing model to resolve the crew availability problem. Kavanagh (1985) presented SIREN (Simulation of Repetitive Networks), a repetitive construction model coded in the GPSS language including deterministic analysis and employs Monte-Carlo simulation to account for the probability distributions for values of activity durations and weather conditions. Isidore and Back (2002) combined the techniques of range estimating and probabilistic scheduling using multiple simulation analysis to help mitigate the risks in construction projects. Peña-Mora et al (2008) developed a Discrete Event and System Dynamics hybrid simulation model to simulate the combined effect of operational and strategic management decisions on infrastructure projects performance. Labban el al (2013) developed a special purpose discrete event simulation model to estimate, plan and manage asphalt paving operations, and describe a proposed alternative method currently being research to rapidly build construction simulation models. Zankoul and Khoury (2014) created a generic and dynamic Discrete Event simulation model that can be adopted by contractors in order to plan and optimize the construction process of on-shore wind farms in Lebanon and elsewhere. Zankoul et al (2015) developed an Agent based and Discrete Event hybrid simulation model to capture and visualize the interaction among the different resources or entities in an earthmoving operation and define the weak links in order to improve the efficiency of such activities onsite.

Zaheer (2000) and Hossain (2010) illustrate the advantages and disadvantages of simulation methods in solving optimization problems as follows:

Advantages of simulation methods:

- The model configuration empowers or compels the investigator to look at all the components of the framework in subtle element.
- This examination might prompt a superior comprehension and changes of the framework.
- "What-if' inquiries can be replied which is imperative in outlining of new frameworks.
- Bottleneck analysis can be performed demonstrating where work-in-procedure, data, and materials are being unnecessarily deferred.

Disadvantages of simulation methods:

- Simulation is utilized as a part of a few situations when an analytical solution is conceivable or even best.
- Rigorous validation is required.
- Simulation is a device that just translates information and can't decide.
- Model parameters might be hard to gather or decide.
- Simulation is not an exact science.

2.4 Genetic algorithm (GA) methods

GA belongs to the larger class of evolutionary algorithms (EA) that solves optimization problems using techniques based on natural evolution. There are other members of EA such as genetic programming, evolutionary programming and evolution strategy. A GA can be defined as random searching algorithm which depends on the mechanism of natural selection and survival of the fittest solution. The three most important operators of GA are selection, crossover and mutation (Zhou el al, 2013).

Genetic algorithms have the following advantages over other optimization methods (Hossain, 2010):

- GAs operate on a coding set of variables and not with the variables themselves.
- GA is a global optimization method and suitable for large-scale problems that have numerous nearby optima. It can discover optimal solutions inside of an extensive search space without requiring much user supplied data.
- GAs search for a population of solutions, instead of enhancing a single solution. This implies that GAs are different directional and all around pursuit through the upkeep of a population of potential solutions from generation to generation.
- A Pareto GA has the interesting capacity to look for a set of solutions by means of rank, rather than by the function values of a point.
- GAs don't require any helper data aside from the objective function values.
- GAs don't have numerous numerical necessities and can deal with a (linear or non-linear, convex or non-convex, etc.) functions and constraints.
- GAs' transitions schemes are probabilistic; while, conventional techniques use gradient information.
- GAs' are intrinsically parallel, while most other algorithms are serial and can only explore the solution space to a problem in one direction at a time.
- GAs' have the ability to manipulate many parameters simultaneously, which is ineffective by other algorithms.

Hegazy and Kassab (2003) used genetic algorithm procedure together with flowchart based simulation tool in order to solve resource optimization problems. Hegazy et al (2004) proposed a distributed scheduling model using genetic algorithms to select the optimal construction methods, work order, resource planning and project costs for large infrastructure networks projects. . Zheng et al (2004) used genetic algorithms technique illustrated in a simple manual simulation model that aims to optimize total time and total cost simultaneously in construction projects. El-Rayes and Kandil (2005) proposed a GA-based approach to solve a highway construction scheduling problem. The objective of the optimization problem is to minimize construction time and cost, while maximize its quality. Dawood and Sriprasert (2006) used GA technique to solve multi-constraint scheduling optimization problem. Given multiple constraints, the GA alters tasks' priorities and construction methods so as to arrive at an optimum solution. Kim and Ellis (2008) used a permutation-based elitist GA to solve large-scale resource constrained construction scheduling problem. The algorithm employs the elitist strategy to preserve the best individual solution for the next generation so an improved solution can be obtained. Chen and Weng (2009) proposed a two-phase GA module for resource constrained project scheduling problem.

3. DESCRIPTION OF ELECTRICAL SUBSTATION

Electrical substations are unique projects, consists mainly of the same buildings and components with the same design. Sometimes the areas of the buildings and number of components differ from substation to another according to the total area of the substation and the electrical power the substation is required to deliver. The scope of civil works includes all construction, finishing and sanitary works. The main components of an electrical substation are:

- Gas insulated switchgear (GIS) building, this building is designed to house the gas insulated switchgear and auxiliary equipment, distribution panels, cables and raceways, and the portion superstructure is serviced by a crane. The GIS portions are designed to accommodate the bus/breaker arrangement and the space requirements for maintenance area. The GIS building is a single story reinforced concrete building with large height ranges from 11.0 to 14.0 m, width ranges from 10.0 to 12.0 m and length varies according to number of cells and switchgear that will be erected;
- Control building, it is also a single story reinforced concrete building of height less than GIS building. This building is designed to house medium voltage switchgear, auxiliary transformers, batteries, power supply system, telecommunication system, and control and protection system;
- Switchyard portals foundations, these are concrete footings designed to carry out the steel structure of various apparatus for all switchyard equipment;
- The foundations of the outdoor main transformers, they are designed to carry the loading from the oil filled transformers with any additional loads due to transformers operation, seismic and other natural phenomes. There are two types of transformer’s foundations; foundations for outdoor switchyard power system and foundations for switchgear power system;
- Fire walls, these walls are protection reinforced concrete walls constructed between every two transformers foundations;
- Underground reinforced concrete fire footings, they are constructed for outside galvanized steel firefighting water tank;
- The galvanized steel fence surrounds the outdoor switchyard and the outdoor transformers area and rested on concrete footings;
- Cable trenches, reinforced concrete trenches are required for installation of buried cables, conduits, piping, etc.
- Roads are required for operation and maintenance of the substation and to get easy access to any part inside the substation. Roads are divided into main roads and secondary roads;
- Firefighting buildings, they are reinforced concrete skeleton buildings with suitable area constructed to maintain firefighting system for transformers provided with all necessary equipment and accessories;
- Workshop building, it is a reinforced concrete building constructed for storage, maintenance and workshop works;
- Ground and elevated tank, water is stored in the ground tank and water supply is provided with water pressure maintained by an elevated tank. The volume of the water in the elevated tank is dimensioned for both potable and water for firefighting and with necessary steps taken to secure the water supply for firefighting system;
- Drainage tanks, those are underground reinforced concrete tanks (septic tank, filter tank and collecting tank) connected with manholes constructed for the drainage of sewage water;
- Shed roof for vehicles, sun shed parking for vehicles is arranged, so that, one shed for buses and trucks and the second shed for cars. The shed skeleton (columns and roof) is constructed either of reinforced concrete or steel structure. Both types of skeleton lay on concrete foundations.

4. METHODOLOGY

The simulation model of electrical substation is developed using AnyLogic software connected with data base excel file for input data entry. AnyLogic has been generally utilized in simulation for simplicity of model building, a visual model improvement environment, adaptability, and animation. The interesting adaptability of the modeling language empowers the user to capture the multifaceted nature and heterogeneity of business, financial and social systems to any craved level of detail. The graphical interface, apparatuses, and library objects permit user to rapidly show differing ranges, for example, assembling and logistics, business processes, human resources, purchaser and patient conduct. The object-oriented model outline worldview accommodates particular, hierarchical, and incremental development of expansive models.

AnyLogic bolsters the consistent combination of discrete and continuous simulations. The native Java environment bolsters boundless extensibility including custom Java code, outside libraries, and outer information sources. A broad statistical distribution function set gives an incredible stage to mimicking the instability natural in all systems. Furthermore, the powerful experimental framework, bolster a wide assortment of simulation approaches.

AnyLogic includes a graphical modeling language i.e. the model is built in a graphical editor that allows the user to edit the diagram of the model graphically. The graphical model elements are called “Active objects”, they are the main building elements of AnyLogic models. Active objects can be used to model very diverse objects of the real world. These elements are shown in a palette view grouped by categories in a number of stencils (palettes) and libraries such as Agent, Presentation, State chart, Connectivity, Process Modelling Library and Pedestrian Library.

The substation model is built by discrete event simulation method. Discrete-Event simulation modeling also, referred to as “Process-Centric” in AnyLogic, consists of dividing a continuous process into discrete parts to simplify analysis. The process in question is modeled as a sequence of discrete events or activities. Entities arrive into the model at a certain arrival rate and go through its activities one at a time sequentially. The AnyLogic Process Modelling Library elements are used to build this model as it supports discrete-event, or process-centric modeling paradigm.

The execution process of civil works of the substation depends on dividing the total construction area into three zones (phases) working in parallel to each other as they are not related to or dependent on each other. Therefore, the model is divided into three zones; each zone includes a group of buildings which are related to the same construction area as following:

Zone 1: This zone is related to gas insulated switchgear area which includes GIS building and control building.

Zone 2: This zone is related to switchyard area which includes portals foundation, transformers foundations, fire footings, fire walls, chain link fences, cable trenches, main roads and secondary roads.

Zone 3: This zone is related to service buildings area which includes firefighting buildings, workshop, elevated tank, drainage tank and vehicles shed.

Building the model goes through ten main stages as follows:

- Creating a data base excel file;
- Creating a new model called “OzModel”;
- Creating a new java class “Util”;
- Creating the flow charts diagrams of Zone 1 model- agent type;
- Creating the flow charts diagrams of Zone 2 model- agent type;
- Creating the flow charts diagrams of Zone 3 model- agent type
- Cost estimation of the substation (direct and indirect cost);
- Creating the “Simulation” experiment;
- Creating the “Optimization” experiment;
- Case study and results.

5. MODEL DEVELOPMENT

5.1 Creating a data base excel file

To facilitate the data entry to the Model, a Microsoft Excel file is used, so that the user can easily enter or change all input data needed to operate the model. The data base file consists of two sheets. Sheet1 includes the bill of quantities of all clauses of the substation, categorized by type of building, a resource breakdown for each clause; manpower, equipment and materials, unit cost for each raw material and indirect cost. The resource breakdown for each activity shows the quantity of each raw material needed for this activity, and the quantity needed to be done by each equipment and manpower crew. The cell connecting each row and columns refers to the quantity needed from the resource mentioned in this column to the activity mentioned in this row. For example the value in cell (1, 110, 8) = 5427; the value 1 corresponds to the spread sheet number 1, the value 110 corresponds to the row number 110 and value 8 corresponds to the column number 8. This means that in spread sheet (1), the quantity needed from the resource type mentioned in column (8) to the activity described in row (110) is 5427 (*see Appendix A*).

Sheet 2 in data base excel file includes the production rates of resources (equipment and manpower crews), the maximum number available from each type of resource for each zone in the model, the number required from each type of resource for each building and the unit cost of each resource per day (*see Appendix B*).

5.2 Creating a New Model "OzModel"

A new model is created by clicking on file button in the Anylogic window and choose New > Model. When the new model wizard is displayed, the name of the model "OzModel" is written in the "Model name" field and Model time unit is chosen from the "Model time units" drop box to be days.

5.3 Creating a Java Class "Util"

A java class called "Util" is created in order to write all java codes used to set the type of resource pools and number of resource units needed from each resource pool to be seized by activates in order to be executed, each code will be called by the active object (activity) which utilizes this type of resource. Java class is created by right click on the model name in the projects view, then choose New > Java Class. In the Java Class wizard, the name "Util" was added in the "Name" field. This name will be used to identify and access this object from code. Also, java class fields are added, the name, type, access and initial value for each class field are entered.

5.4 Creating "Zone 1" model- agent type

As mentioned previously, Zone 1 is the model representing gas insulated switchgear area in the substation which includes GIS building and control building. In Zone 1 agent type, the flow chart for each of these buildings is represented, resource pools used by active objects, parameters indicating the number of resource units seized by each active object from each resource pool, all types of raw material utilized by active objects, parameters indicating unit cost and total cost of each resource pool, parameters indicating the quantity utilized, unit cost and total cost of each raw material , parameters indicating indirect cost and all java codes need to calculate all these values .

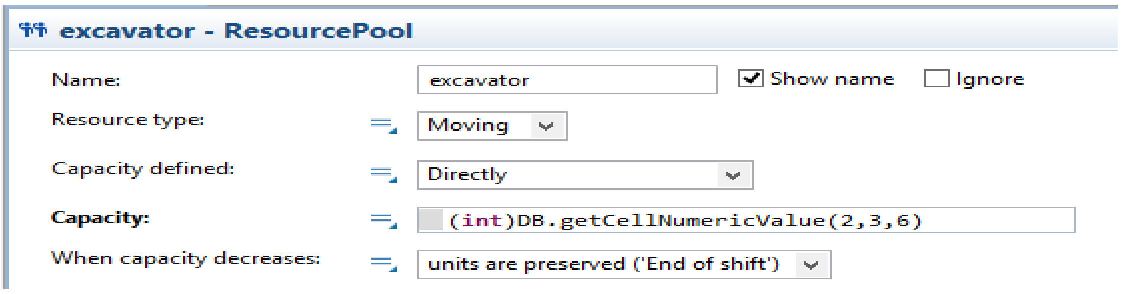
Creating new Zone-Agent type by right click on the model name in the projects view, then choose New > Agent Type. In the new agent type wizard, the name of the agent "Zone1" is written in the "The name of new type" field and "Do not use template" check box is used. After completing the process, the graphical editor will be opened to build the model as following:

5.4.1 Data base

The first step is to make the model read input data from the data base excel file and this by done by dragging the "Excel File" element from the Connectivity palette of the Palette view onto the graphical diagram. In the Properties view, the name of the element is modified to "DB" in the Name field. This name will be used to identify and access this object from code. Excel file in which this object will work with is specified in "File" field. In order to read any cell from “DB” excel file, the following code is written in the place where this cell is called: “*DB.getCellNumericValue(sheet number, row number, column number)*”. For example “DB.getCellNumericValue(2,3,6)”; this code reads the number written in the cell located in “DB” excel file, in sheet number 2, row number 3 and column number 6.

5.4.2 Resource pools

The equipment and manpower crews are defined in the model as resource pools. A resource pool defines a set of resource units that can be seized and released by entities using flowchart objects, in which the same resources are shared between activities and move inside the project as entities. Creating a resource pool object is done by dragging the “Resource Pool” element from the Process Modeling Library onto the graphical diagram. In the Properties view, the name of the element is modified to the name of resource in the “Name” field. The type of resource is chosen from Resource type drop box, either Moving, Static or Portable. The capacity of each resource means the maximum number of units available from this resource. The capacity is defined using “Capacity defined” drop box methods The resource pool capacity is read from the data base file, sheet number 2, the column referring to the maximum number of resources available for “Zone 1”using java code (*see Fig. 1*).



excavator - ResourcePool

Name: excavator ☑ Show name □ Ignore

Resource type: Moving

Capacity defined: Directly

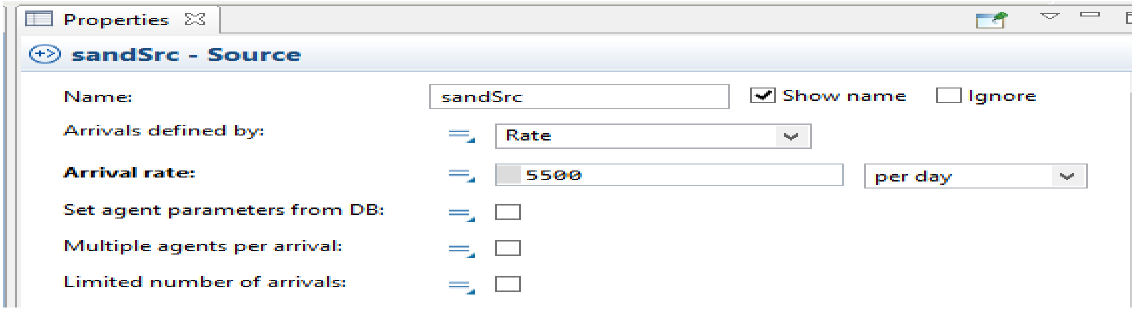
Capacity: (int)DB.getCellNumericValue(2,3,6)

When capacity decreases: units are preserved ('End of shift')

FIG. 1: Excavator resource pool object properties window

5.4.3 Raw materials

Each type of raw materials is represented in the model by two objects: a “Source” object and “Queue” object. Source object is used to start generating of raw material, while Queue object is used to store the raw material in the specified order until it is utilized by active objects. Creating the raw material objects is done by dragging the Source and Queue elements from the Process Modeling Library of the Palette onto the graphical diagram. The elements are connected together with a connector. In the Properties view of Source object, the name of the element is modified to the name of raw material in the “Name” field. Arrival method of raw material is chosen from “Arrivals defined by” check box to be Rate and the rate is defined in “Arrival rate” field by units per day (*see Fig. 2*).



Properties

sandSrc - Source

Name: sandSrc ☑ Show name □ Ignore

Arrivals defined by: Rate

Arrival rate: 5500 per day

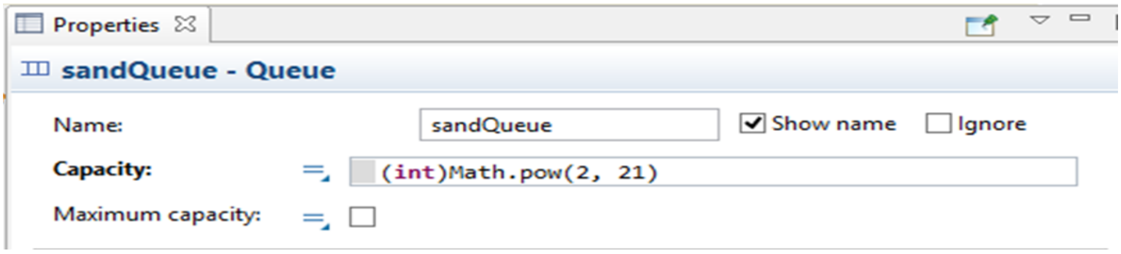
Set agent parameters from DB: □

Multiple agents per arrival: □

Limited number of arrivals: □

FIG. 2: Sand material source object properties window

In the Properties view of Queue object, the name of the element is modified in the” Name” field. Capacity of Queue is the maximum amount of raw material in which the Queue can store. The capacity is defined in “capacity” field by java code *(see Fig. 3).*



Properties

sandQueue - Queue

Name: sandQueue ☑ Show name □ Ignore

Capacity: (int)Math.pow(2, 21)

Maximum capacity: □

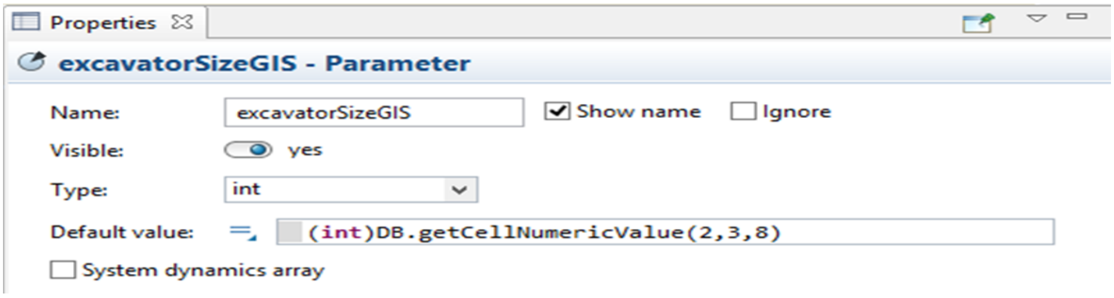
FIG. 3: *Sand material queue object properties window*

5.4.4 Resource parameters

The number of resource units needed by each active object (activity) from each resource pool in order to be executed is defined as a parameter. In simulation experiment, the values of these parameters are read from the data base excel file, while in optimization experiment, the values of these parameters are considered decision variables varies within a specified range to obtain the minimum optimum time. Resource parameters are created for each building in the zone. Creating resource parameter is done by dragging the Parameter element from the Agent palette onto the graphical diagram. In the Properties view, the name of the element is modified in the “Name” field. The type of resource unit is chosen from the “Type” field drop box to be integer number (int). The number of resource units needed is defined in the “Default value” field. This number is read from the data base excel file using java code *(see Fig. 4).*

5.4.5 Flowcharts

Flowchart is the sequence of work for each building represented by active objects connected together by connectors indicating the relationship between objects. Zone1 consists of two flowcharts, GIS building flow chart and control building flowchart. The active objects used to build these flowcharts from process modeling library are source, split, service, assembler, combine, select output, batch and sink.



Properties

excavatorSizeGIS - Parameter

Name: excavatorSizeGIS ☑ Show name □ Ignore

Visible: yes

Type: int

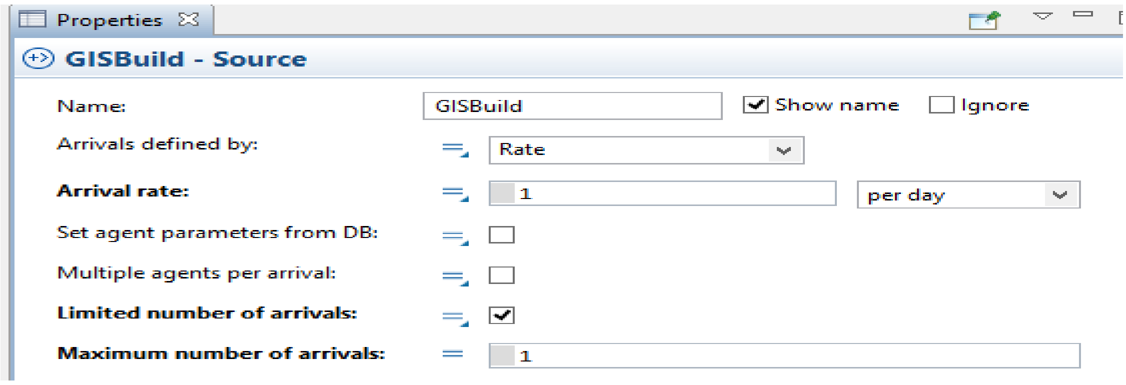
Default value: (int)DB.getCellNumericValue(2,3,8)

□ System dynamics array

FIG. 4: *Excavator of GIS building parameter properties window*

5.4.5.1 Source object

This object is used to generate entities; it is usually a starting point of a process model. In the Properties view of Source object, the name of the element is modified to the name of building in the “Name” field to be name of the building. Method of arrival of entities is chosen from “Arrivals defined by” checkbox to be “Rate”. Arrival rate is defined in “Arrival rate” field as number of entities arrived per model unit time which is days. If there are multiple entities arrived in the same arrival, “Multiple entities per arrival” box is checked. If a specific number of entities will arrive, the “Limited number of arrivals” box is checked, and the number of entities is defined in “Maximum number of arrivals” field *(see Fig. 5).*



Properties

GISBuild - Source

Name: GISBuild ☑ Show name □ Ignore

Arrivals defined by: Rate

Arrival rate: 1 per day

Set agent parameters from DB: □

Multiple agents per arrival: □

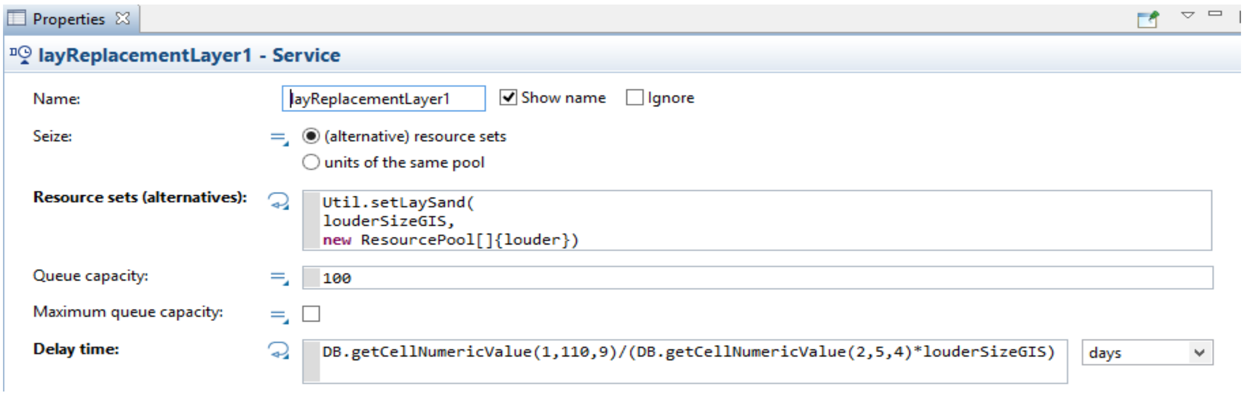
Limited number of arrivals: ☑

Maximum number of arrivals: 1

FIG. 5: GIS building source object properties window

5.4.5.2 Service object

This object is used to seize a given number of resource units, delays the entity, and releases the seized units. It refers to execution process of an activity. In the Properties view of Service object, the name of the element is modified to the name of activity to be done in the “Name” field. In the seize field, two options are introduced to seize resources, either resources just of one single type, so “units of the same pool” option is chosen, or resources of different types, so “(alternative) resource sets” option is chosen. The type of resources and the number of resource units (resource parameters) to be seized for this service are set in “Resource sets” field by calling the activity resource set code from “Util” java class and specifying the resource parameters and types of resource pools. The time this service will take to be executed is defined in “Delay time” field by using this equation (Quantity / (resource production rate x Number of resources parameter)). All fields of the equation are read from data base excel file by using java code (*see Fig. 6*). The quantity of raw materials required for this service is called from each raw material queue by using a java code written in “On enter” field. This code reads the required quantity of each raw material from data base excel file and calls it from the raw material queue store (*see Fig. 7*).



Properties

layReplacementLayer1 - Service

Name: layReplacementLayer1 ☑ Show name □ Ignore

Seize: ● (alternative) resource sets ○ units of the same pool

Resource sets (alternatives):

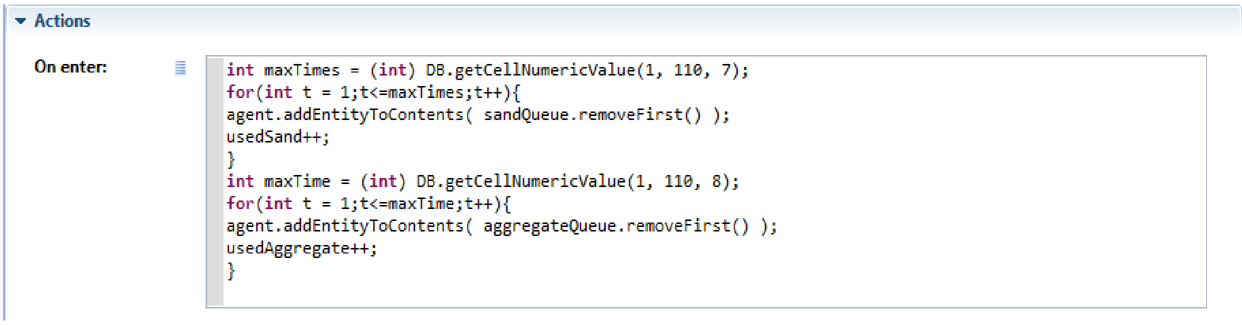
```
Util.setLaySand(
louderSizeGIS,
new ResourcePool[]{louder})
```

Queue capacity: 100

Maximum queue capacity: □

Delay time: `DB.getCellNumericValue(1,110,9)/(DB.getCellNumericValue(2,5,4)*louderSizeGIS)` days

FIG. 6: Part 1 of lay replacement layer service object properties window



Actions

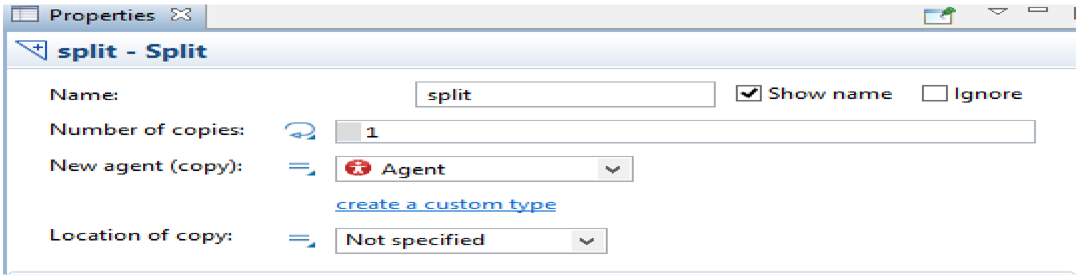
On enter:

```
int maxTimes = (int) DB.getCellNumericValue(1, 110, 7);
for(int t = 1;t<=maxTimes;t++){
agent.addEntityToContents( sandQueue.removeFirst() );
usedSand++;
}
int maxTime = (int) DB.getCellNumericValue(1, 110, 8);
for(int t = 1;t<=maxTime;t++){
agent.addEntityToContents( aggregateQueue.removeFirst() );
usedAggregate++;
}
```

FIG. 7: Part 2 of lay replacement layer service object properties window

5.4.5.3 Split object

For each incoming entity ("original"), Split object creates one or several other entities and outputs them via outCopy port. A new agent created may have a meaning of a copy, a sibling, etc. and can be of arbitrary type. The number of new entities is changed dynamically. This is used to make the entity pass in several parallel paths at the same time. In the Properties view, the number of paths is defined in “Number of copies” field (*see Fig. 8*).



split - Split

Name: split ☑ Show name □ Ignore

Number of copies: 1

New agent (copy): Agent

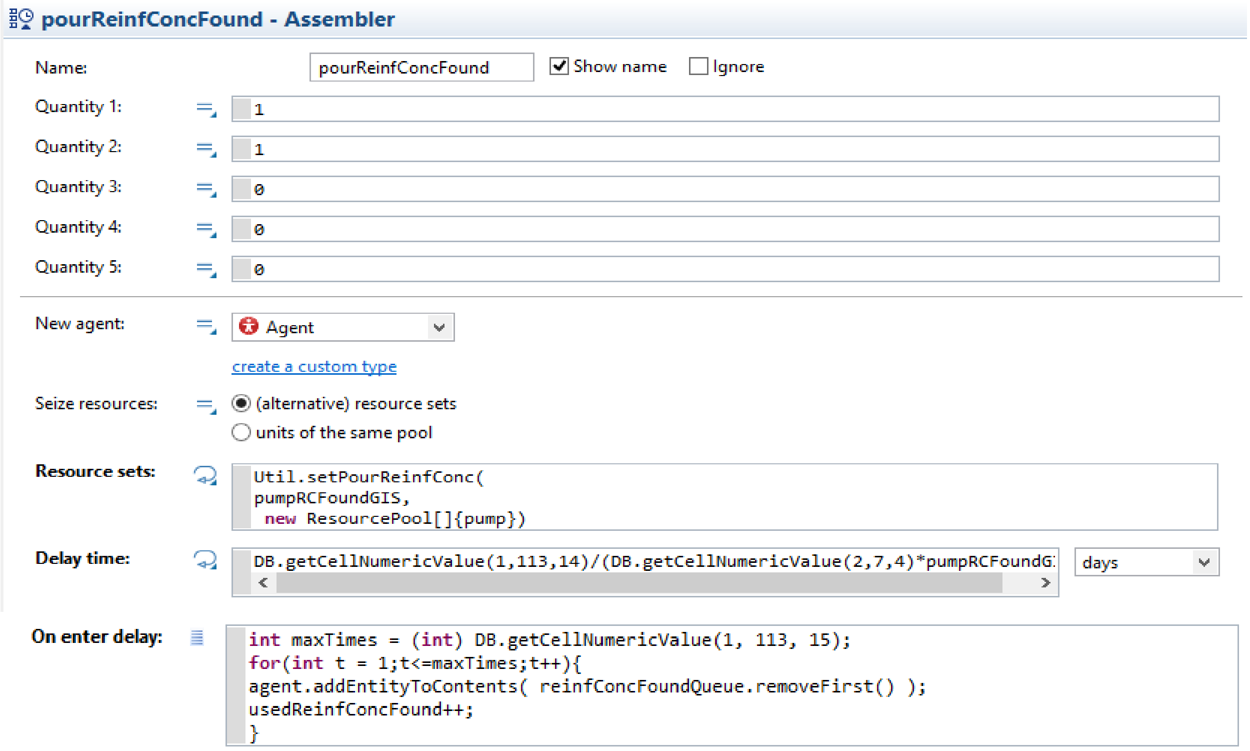
create a custom type

Location of copy: Not specified

FIG. 8: Split object properties window

5.4.5.4 Assembler object

The object allows certain number of entities from several sources (5 or less) to be joined into a single entity. The type of the new entity, as well as its initialization, is specified in the new object. The number of entities for each input required to produce one new entity is also specified using the object parameters (Quantity 1, Quantity 2, etc.). All arrived entities wait inside the object until all required entities arrive. Once the new entity can be build, the assembly operation starts. This operation takes the time specified in the Delay time field. In the Properties view of Assembler object, the name of the element is modified to be the name of activity in the “Name” field. The quantity of entities required to arrive from each input to create the new entity is specified in “Quantity 1, Quantity 2, Quantity 3, Quantity 4 and Quantity 5” fields. Seize resources, resource sets, delay time and quantities of raw materials are set following the same steps as mentioned previously in section 5.4.5.2 (*see Fig. 9*).



pourReinfConcFound - Assembler

Name: pourReinfConcFound ☑ Show name □ Ignore

Quantity 1: 1

Quantity 2: 1

Quantity 3: 0

Quantity 4: 0

Quantity 5: 0

New agent: Agent

create a custom type

Seize resources: ● (alternative) resource sets ○ units of the same pool

Resource sets:

```
Util.setPourReinfConc(
pumpRCFoundGIS,
 new ResourcePool[]{pump})
```

Delay time:

```
DB.getCellNumericValue(1,113,14)/(DB.getCellNumericValue(2,7,4)*pumpRCFoundG:
```

days

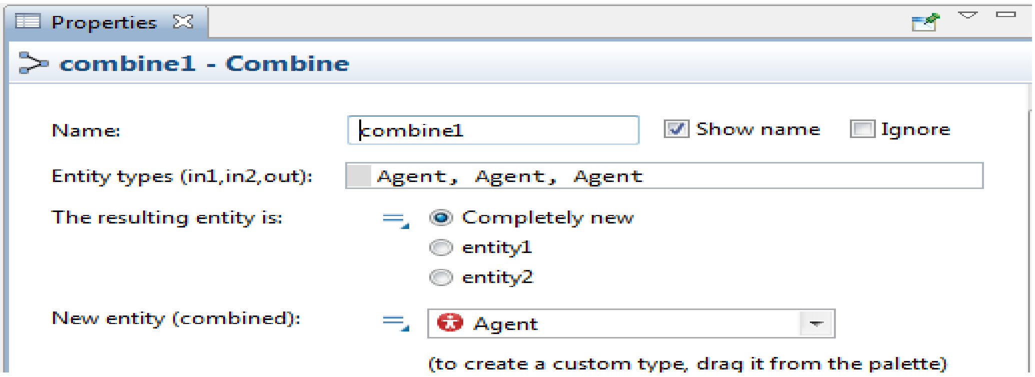
On enter delay:

```
int maxTimes = (int) DB.getCellNumericValue(1, 113, 15);
for(int t = 1;t<=maxTimes;t++){
agent.addEntityToContents( reinfConcFoundQueue.removeFirst() );
usedReinfConcFound++;
}
```

FIG. 9: Pour concrete assembler object properties window

5.4.5.5 Combine object

This object waits for the two entities to arrive (in arbitrary order) at ports in1 and in 2, produces a new entity and outputs it. The new entity may be a "completely new", i.e. a newly constructed object whose properties possibly depend on the original entities, or it may be one of the original entities. Once the two entities are ready, the combine operation takes zero time. In the Properties view of Combine object, the type of the new entity is specified in “The resulting entity is” field either Completely new, entity 1 or entity 2 (*see Fig. 10).*



combine1 - Combine

Name: combine1 ☑ Show name □ Ignore

Entity types (in1,in2,out): Agent, Agent, Agent

The resulting entity is:
● Completely new
○ entity1
○ entity2

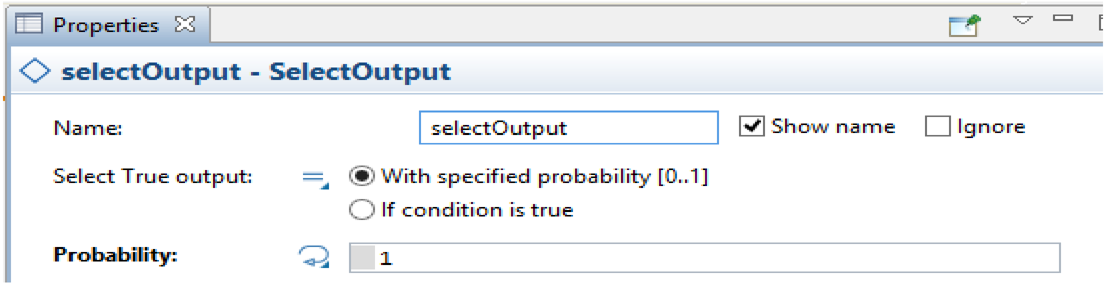
New entity (combined): Agent

(to create a custom type, drag it from the palette)

FIG. 10: Combine object properties window

5.4.5.6 Select output object

This object routes the incoming entities to one of the two output ports depending on (probabilistic or deterministic) condition. The condition may depend on the entities as well as on any external factors. The entities spend zero time in SelectOutput. In the Properties view of SelectOutput object, the name of the element is modified in the “Name” field. In “Select True output” field, there are two options introduced, either the entities are routed randomly with specified probability, in this case “With specified probability [0..1]” is chosen, and the probability of true route is defined in “Probability” field. The other option routes the entities depending on specified boolean condition, in this case “If condition is true” field is chosen, and the condition is defined in Condition field using java code (*see Fig. 11).*



selectOutput - SelectOutput

Name: selectOutput ☑ Show name □ Ignore

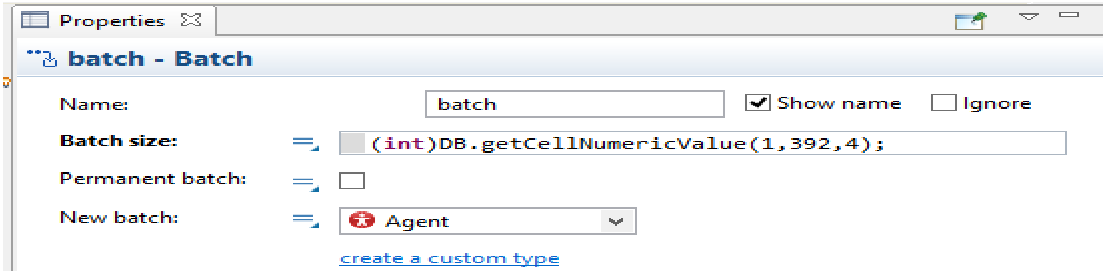
Select True output:
● With specified probability [0..1]
○ If condition is true

Probability: 1

FIG. 11: Select output object properties window

5.4.5.7 Batch object

This object converts a number of entities into one entity by either discarding the original entities and creating a new one - permanent batch (whose properties possibly depend on the original ones), or by adding the original entities to the contents of the new entity - temporary batch. This object embeds a Queue object where it stores the incoming entities. Once the number of the stored entities reaches the batch size, the batch is created and leaves the object immediately. In the Properties view of Batch object, the number of entities that will be batched into one entity is defined in “Batch size” field (*see Fig. 12).*



Name: batch ☑ Show name □ Ignore

Batch size: (int)DB.getCellNumericValue(1,392,4);

Permanent batch: □

New batch: Agent

FIG. 12: Batch object properties window

5.4.5.8 Sink object

This object disposes entities. It is usually an end point in a process model. Creating the Sink object is done by dragging the Sink element from the Process Modeling Library of the Palette onto the graphical diagram.

Fig. 13 shows a flowchart template for control building. All buildings flowcharts are built following the same template according to the activities of each building.

Creating Zone 2 and Zone 3 model-agent types are done by following the same steps of creating Zone 1 as illustrated in section 5.4. Zone 2 includes eleven flowcharts, each flowchart represent a building /component in the switchyard area. These components are; portals foundations model F1, portals foundations model F2, transformers foundations 220kv, transformers foundations 66kv, fire footings, fire walls, chain link fences, cable trenches inside switchyard, cable trenches outside switchyard, main roads and secondary roads. While Zone 3 includes five flowcharts represent five buildings/ components in the service buildings area. These components are firefighting buildings, workshop, elevated tank, drainage tank and vehicles shed.

5.5 Cost estimation

Cost of the model consists of direct cost and indirect cost and each cost consists of fixed cost and variable cost. Direct cost is the cost of materials, equipment and manpower crews, while indirect cost is the cost of salaries, overhead and other site expenses. Fixed cost is a constant value which does not change by model time variation. While variable cost is a function of time, it varies according to the total model time units. Direct fixed cost is the cost of raw materials and direct variable cost is the cost of equipment and manpower crews. Indirect fixed cost includes site offices, mobilization, taxes and deductions of Employer. Indirect variable cost includes staff wages, head office overhead and site facilities (flats rent, cars, water, electricity, phones, site expenses and etc).

5.5.1 Estimation of direct variable cost (resource pools)

Equipment and manpower crews are defined in the model as resource pools. Determining the cost of a resource unit in a resource pool is done by creating a custom resource type to obtain the utilization of a unit as a percent of time the unit was busy during simulation. Knowing it, the total number of days each unit was in work can be calculated as follows: time()*myResources.get(i).getUtilization()/day() - where "myResources" is a population of resource units. After multiplying it by the cost of one unit per day, total cost will be got. These steps are implemented as following:

5.5.1.1 Creating custom resource type

Custom resource type is created by dragging the Agent element from the General Library palette into the graphical editor. In the new agent wizard, the type of agents to be created is specified as population of agents and “I want to create a new agent type” box is checked. The name of the resource type is defined in “The name of the new type” field. “Create initially empty population, I will add agents at the model runtime” box is checked. In the properties view of the new resource type, Resource unit is chosen from “Use in flowchart as” drop box.

5.5.1.2 Adjusting resource pool properties

In the properties view of resource pool, the new resource type is chosen from “New resource unit” drop box. In “Add units to” field, custom population box is checked, and the new resource type population is chosen from “Population” drop box.

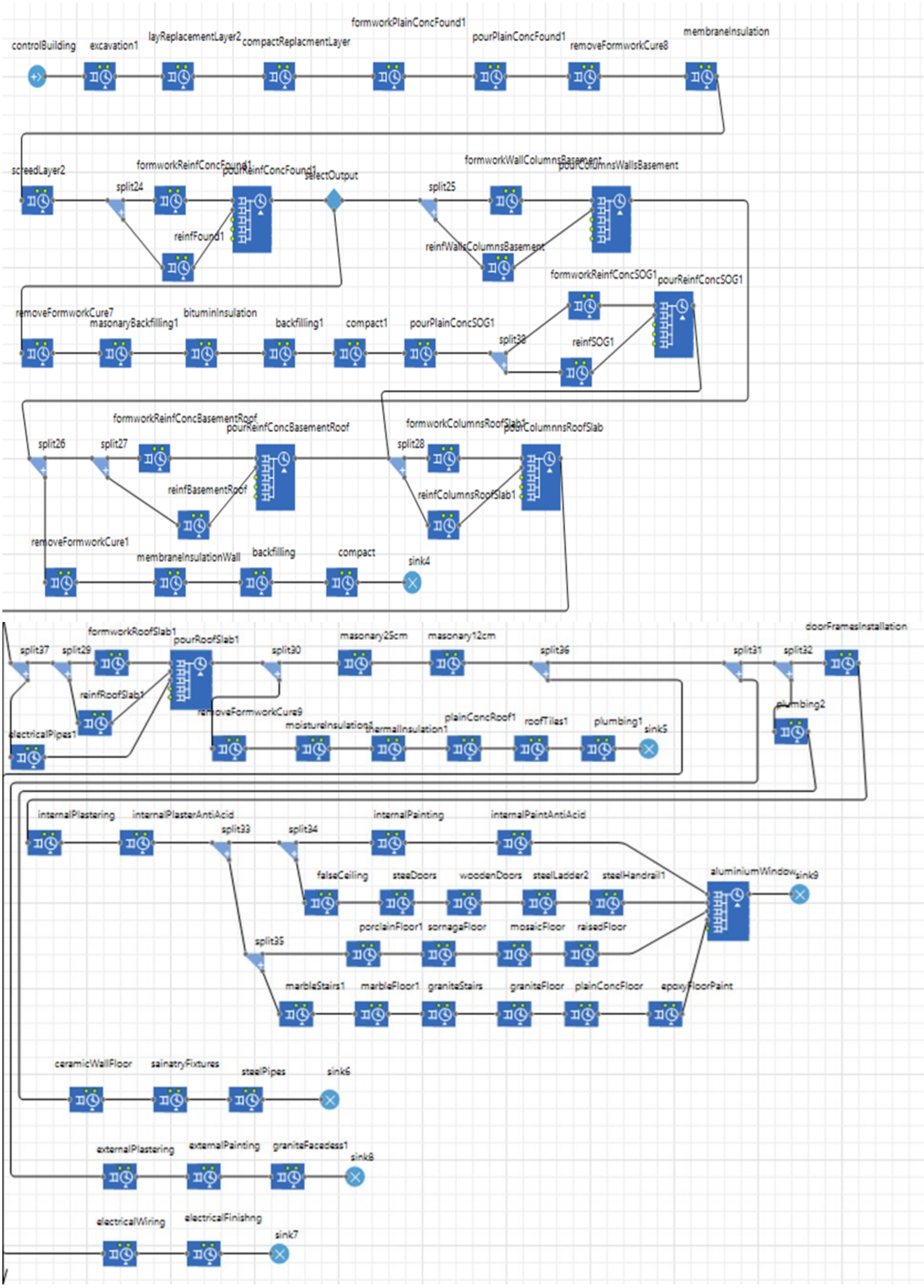


FIG. 13: Control building flow chart

5.5.1.3 Creating Cost Variables

Two variables are created for each resource pool; the first variable is created to specify the unit cost of this resource, the cost is read from data base excel file using java code. The second variable is created to determine the total cost of this resource pool at the end of model simulation or at any time the simulation is terminated. Creating a variable is done by dragging the Variable object from the General Library palette into the graphical editor. In the properties view, the name of the variable is specified in the "Name" field. The value of the variable is specified in "Initial value" field. For the variables specifying the unit cost of resource, the unit cost is read from data base excel file using java code. While for variables indicating the total cost, the initial value is set to be zero.

5.5.1.4 Estimating Resource Pool Cost

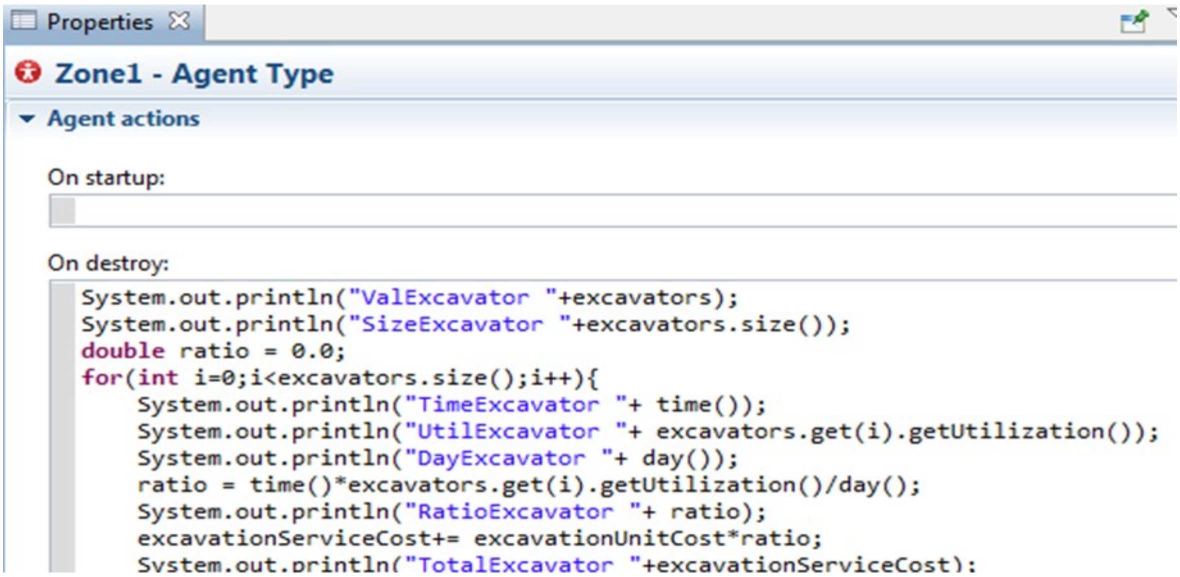
After creating the custom resource type for each resource pool, adjusting resource pool properties and creating cost variables, each resource pool cost is calculated using a java code written in on destroy field, in the properties view of each Zone-Agent type. The aim of the code is to determine the total time units of simulation model, the percentage of utilization of each resource in the resource pool, the total time units this resource was busy during the model by multiplying the percentage of utilization of the resource by the total time units of the model, the cost of the resource by multiplying the unit cost of this resource (unit cost variable) by the total time units this resource was busy and finally the total cost of all units in the resource pool (*see Fig. 14)*.

5.5.2 Estimation of direct fixed cost (raw materials)

The cost of each raw material is determined by first determining the amount of material has been withdrawn from the storage (queue of this material) and then multiplying this amount by the unit cost of this material which is read from the data base excel file to obtain the total cost of each material. These steps are implemented as following:

5.5.2.1 Estimation of material quantity

To determine the utilized quantity of each material, a variable is created for each material and then increased by increment (1) each time an active object withdraw this material from the queue to add this amount to the variable. This is done by first dragging the Variable object from the General Library palette into the graphical editor. In the properties view, the name of the variable is specified in the "Name" field. The type of the variable is specified in the "Type" field to be integer (int). The initial value of the variable is specified in "Initial value" field to be zero. In "On enter" field of the activity object properties, where the quantity of each material is called to be utilized, the variable name is called using java code in order to add this utilized quantity in the material variable (for example, in each time, calling sand material quantity, the following code is written "usedSand++").



```
System.out.println("ValExcavator "+excavators);
System.out.println("SizeExcavator "+excavators.size());
double ratio = 0.0;
for(int i=0;i<excavators.size();i++){
    System.out.println("TimeExcavator "+ time());
    System.out.println("UtilExcavator "+ excavators.get(i).getUtilization());
    System.out.println("DayExcavator "+ day());
    ratio = time()*excavators.get(i).getUtilization()/day();
    System.out.println("RatioExcavator "+ ratio);
    excavationServiceCost+= excavationUnitCost*ratio;
    System.out.println("TotalExcavator "+excavationServiceCost);
```

FIG. 14: Part of calculating resource pools cost java code in zone-agent type properties

5.5.2.2 Create material unit Cost variable

To read the unit cost of material from the data base excel file, a variable is created for each material. In the properties view, the type of the variable is specified to be double and the initial value of the variable is specified using java code to read the unit cost of this material from the excel file.

5.5.2.3 Estimation of total material cost

Determining the total material cost in each zone, by multiplying the unit cost variable of each material by the total quantity variable of this material. This equation is written in on destroy field, in the properties view of each Zone-Agent type.

5.5.3 Estimation of indirect fixed cost

To set the indirect fixed cost in the model, a variable is created with type double to read this value from the data base excel file using java code.

5.5.4 Estimation of indirect variable cost

The indirect variable cost is defined in the model as a variable; cost per unit time, and the total indirect cost is determined at the end of the model run by multiplying the indirect variable cost per day by the total model time days. A variable is created with type double to read this value from the data base excel file using java code. The total indirect variable cost is determined by writing the following equation in on destroy field, in the properties view of each Zone-Agent type; totalIndirectVariableCost = indirectVariableCost*time();.

5.5.5 Estimation of total cost

Total cost of each zone is determined by adding the total direct fixed cost (materials), the total direct variable cost (resource pools), the total indirect fixed cost and the total indirect variable cost. This equation is written in on destroy field, in the properties view of each Zone-Agent type.

5.6 Create optimization experiment

The aim of creating an optimization experiment for the substation model is to find the minimum possible time for the model by getting the optimal combination of resource parameters that affect the result under resource constraints. So, the objective function of this model is to minimize time, the decision variables are the resource parameters defined in section 5.4.4 and the constraints are the capacities of resource pools (maximum available number of resources) defined in section 5.4.2. The optimization model is constructed on the same simulation model using parameter variation experiment introduced by Anylogic software and genetic algorithm (GA) java code written on a java class and implemented using Anylogic platform.

The concept of GA starts from a population of randomly generated individuals, in an iterative process, with the population in each iteration called a generation. In each generation, the fitness of every individual in the population is evaluated; the fitness is usually the value of the objective function. The more fit individuals are stochastically selected from the current population, and each individual's genome is modified (recombined and possibly randomly mutated) to form a new generation. The new generation of candidate solutions is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population.

GA can be considered as four major steps: chromosome representation style, design and define fitness function, design and develop genetic operator and control genetic operator and determine termination condition (Yu and Buyya, 2008).

5.6.1 Chromosome and population representation

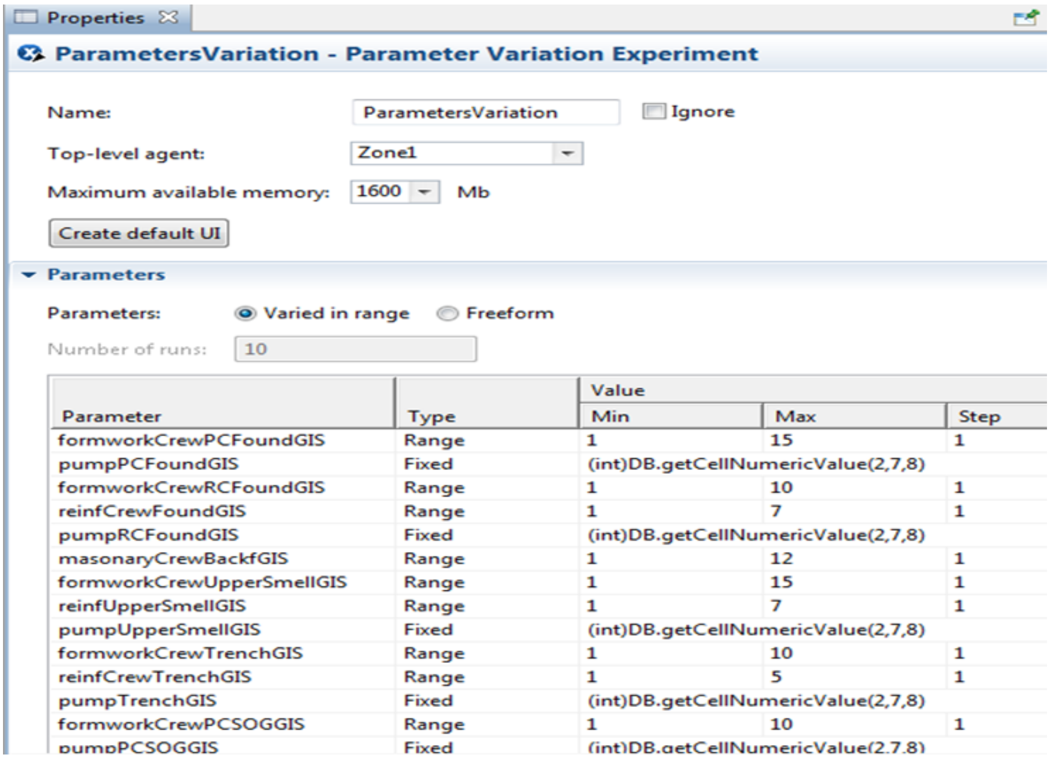
Because, chromosome represents a problem solution and solution of the substation optimization problem is array of values for the different parameters affecting the total time, chromosome has a length equals to total number of parameters in the zone model. Each parameter is represented by a genome. Genome has a value between minimum and maximum specified as possible range of values. These values are elected randomly at runtime over iterations.

A population is a group of chromosomes; each chromosome has its own fitness that is calculated using a fitness function defined in section 5.6.2.

5.6.2 Fitness function definition

Fitness function represents objective function of model optimization problem. The fitness function is based on minimization of total consumed time for substation construction. Model time unit is defined in day as mentioned in section 5.2. Different affecting parameters are merged into a single chromosome. Each chromosome is evaluated to get total time in days. Evaluation of chromosome is performed using Anylogic feature called Parameter Variation Experiment. In parameter variation experiments, the proposed model is executed implicitly using the parameter combination provided by the chromosome. The parameters are passed in order to match variables of the parameter variation experiment. Anylogic platform is able to detect the parameters of the proposed model automatically and allow the developer to customize setting of parameter variation experiment.

Creating parameter variation experiment is done by right-click the model item and choose New > Experiment from the popup menu. Parameter variation option is chosen from the Experiment Type list. The type of model-zone of the experiment is chosen and changed from the Top-level agent drop-down list. In the properties view window of the experiment, all resource parameters of the chosen model-zone (decision variables) are displayed and there are two options to deal with variable parameters, either to be varied in range or freedom. Varied in range option, means each parameter is to be varied through range of values which should be defined. Namely, minimum and maximum values for a parameter and also the increment step this parameter will increase its value to reach the maximum are defined *(see Fig. 15)*. The table lists all the parameters of the top level agent. The Type field drop box has two options for each parameter; range or fixed. For varied parameters, range is specified for the parameter and the parameter's lower bound in the "Min" field, the parameter's upper bound in the "Max" field are entered, and the increment value in the "Step" field is specified. For unvaried parameters, fixed option is chosen and fixed parameter value is specified.



Name: ParametersVariation □ Ignore

Top-level agent: Zone1

Maximum available memory: 1600 Mb

Parameters: ● Varied in range ○ Freeform

Number of runs: 10

Parameter	Type	Value		
		Min	Max	Step
formworkCrewPCFoundGIS	Range	1	15	1
pumpPCFoundGIS	Fixed	(int)DB.getCellNumericValue(2,7,8)		
formworkCrewRCFoundGIS	Range	1	10	1
reinfCrewFoundGIS	Range	1	7	1
pumpRCFoundGIS	Fixed	(int)DB.getCellNumericValue(2,7,8)		
masonaryCrewBackfGIS	Range	1	12	1
formworkCrewUpperSmellGIS	Range	1	15	1
reinfUpperSmellGIS	Range	1	7	1
pumpUpperSmellGIS	Fixed	(int)DB.getCellNumericValue(2,7,8)		
formworkCrewTrenchGIS	Range	1	10	1
reinfCrewTrenchGIS	Range	1	5	1
pumpTrenchGIS	Fixed	(int)DB.getCellNumericValue(2,7,8)		
formworkCrewPCSOGGIS	Range	1	10	1
pumpPCSOGGIS	Fixed	(int)DB.getCellNumericValue(2,7,8)		

FIG. 15: Parameter variation experiment properties view

5.6.3 Genetic operators

Genetic operators are operations that motivate and make difference between generations. Generations means populations which are filled from evolving cycle to another by performing three distinct operators; reselection, cross-over mutate and single genome mutation. Each operator is responsible for filling a ratio of total number required for producing a new population. In each population, three operators cooperate together to produce population chromosomes with predefined share ratio. In reselection operator, the best fit chromosomes are copied into the new population. Best fit chromosomes are determined by total minimum number of required days. These chromosomes are healthy parents that must be kept safe until more fit solution is achieved during evolving operation. In Single genome mutation operation, chromosome is selected for being mutated. Then, a random genome is selected to be mutated. Genome is mutated by generating a random value within possible range. Because of genome represents a variable of optimization problem, the genome has valid range identical for parameter range. In cross-over mutate, two chromosomes are selected for being mutated. Then, a random genomes are selected to be mutated cross over with other genomes from second chromosome. Genomes are mutated by changing their parts by each other.

5.6.4 Termination of evolving iterations

In general, genetic algorithm has stop condition for evolving iterations, either reaching predefined max number of iterations or obtaining optimal fitness value. For substation construction optimization problem with time objective function, optimal is to perform the required efforts within 1 day which is impossible. In turn, best termination condition is seeking max number of iterations. The substation model is designed to have population containing 80262500 chromosomes. Each chromosome consists of fixed number of parameters (genomes) which are; 117 parameter for Zone 1, 132 parameter for Zone 2 and 140 parameter for Zone 3. For each Zone, genetic algorithm runs at least 2000 iteration evolving chromosomes and generates new populations between evolving cycles till get best fit chromosome. Generated population is derived from applying three operators; reproduction, cross-over mutation and single genome mutate. The substation model is zone based structure, which is applied to each zone separately changing length of chromosome, population size and range of each parameter per zone.

5.6.5 Integrating of Genetic Algorithm & Anylogic: Implementation View

Genetic algorithm is implemented using exported parameter variation model per zone from Anylogic platform into stand-alone java application. In parameter variation experiment, the simulation model is executed implicitly using the parameters combination provided by the chromosome. The GA is implemented inside NetBeans software by creating a new project for each zone separately. The project is configured by scientific library for genetic implementation known as Java Genetic Algorithm Programming (JGAP) and its dependencies. Beside, JGAP, the project is configured using the exported Anylogic parameter variation to fit as the fitness function. Inside the created project, a new java class is created called GAFeatureEvaluator to write the genetic algorithm java code.

5.7 Model results validation

The proposed model was validated using a real case study.

5.7.1 Case study description

The project selected for the case study is Elnabq substation which is located in Sharm El-Sheikh city in Egypt. This substation transformers power of 220/66 kv. The land area of this substation is 35,960 m^2, the switchyard area (220 kv) is 24158 m^2, the gas insulated switchgear area (66 kv) is 7,710 m^2 and service buildings area is 4,092 m^2. The substation consists of (1) GIS building, (1) control building, (211) portals Foundations model F1, (25) portals Foundations model F2, (2) transformers Foundations 220 kv, (4) transformers Foundations 66 kv, (2) fire walls, (2) fire footings, (2) chain link fences, one for switchyard and the other for transformers, cable trenches inside switchyard with total lengths 626.2 m, cable trenches outside switchyard with total lengths 155.8 m, main roads with total lengths 552.15 m and secondary roads with total lengths 522 m.

5.7.2 Results and evaluation

The first step to run the model is to enter the input data of Elnabq substation in the data base Excel file. These data are the actual data of Elnabq substation. As mentioned previously in section 5.1, the bill of quantities, resources breakdown, unit cost of materials and indirect cost data are entered in sheet 1 of the excel file. The production rates of equipment and manpower crews, the maximum available number of resources for each zone, the number of resources needed by each building and unit cost per day for each resource are entered in sheet 2 of the excel file.

After running the simulation and optimization models for each zone and obtaining time and cost results, it can be observed that the time of Zone 1 model decreased from 390 days after simulation experiment to 350 days after optimization experiment, the difference in model time between both experiments is 40 days which is nearly 10.25% of the total time of Zone 1. For Zone 2 model, the time decreased from 238 days after simulation experiment to 215 days after optimization experiment, the difference in model time units between both experiments is 23 days which is nearly 9.7% of the total time of Zone 2. While the time in Zone 3 model decreased from 152 days after simulation experiment to 135 days after optimization experiment, the difference in model time between both experiments is 17 days, which is nearly 11.2% of the total time of Zone 3. As all zones work parallel to each other, this means that Zone 1 is the critical path and the total substation model time units decreased by 10.25% which is the value of Zone 1*(see Table. 1).*

TABLE. 1: Comparison between model time results after simulation and optimization experiments

Description	Time after simulation (days)	Time after optimization (days)	Reduction in time (days)	Percentage of reduction in time	CPU time (seconds)
Zone 1	390	350	40	10.25%	2602.86
Zone 2	238	215	23	9.70%	2936.56
Zone 3	152	135	17	11.20%	3114.53
Total	390	350	40	10.25%	8653.95

It is also observed that, due to reduction in model time units after optimization, the model cost also was decreased. So, for Zone 1, the direct variable cost decreased from 1,017,905 L.E. after simulation experiment to 855,990 L.E. after optimization experiment, the cost of materials is fixed in both experiments which is 3,475,183.5 L.E., so the difference in total direct cost of Zone 1 between both experiments is 161,915 L.E. which is nearly 3.6% of the total direct cost of Zone 1 and 15.9% of direct variable cost of Zone 1. Also for Zone 2, the direct variable cost decreased from 940,825 L.E. after simulation to 789,975 L.E. after optimization; the cost of materials is fixed in both experiments which is 2,906,513.75 L.E. So the difference in total direct cost of Zone 2 between both experiments is 150,850 L.E. which is nearly 3.9% of the total direct cost of Zone 2 and 16.03% of direct variable cost of Zone 2. For Zone 3, the direct variable cost decreased from 164,810 L.E. after simulation to 153,645 L.E. after optimization, the cost of materials is fixed in both experiments which is 677,884.4 L.E., so the difference in total direct cost of Zone 3 between both experiments is 11,165 L.E. which is nearly 1.32% of the total direct cost of Zone 3 and 6.8% of direct variable cost of Zone 3. So, generally the total direct cost of the substation model decreased by 323,930 L.E., which represents 3.53% of the total direct cost *(see Table. 2)*. As the indirect variable cost is defined in the model as a cost per unit time, so this cost was decreased as a result of reduction in model time units. The indirect variable cost was decreased from 2,2880,01.3 L.E. after simulation experiment to 2,053,334.5 L.E. after optimization experiment. The cost of indirect fixed cost is the same in both experiments which is 430,000 L.E. So, the total indirect cost of the model was decreased by 234,666.8 L.E. which is 8.63% of the total indirect cost and 10.26% of the indirect variable cost. Finally, the total cost of the substation model decreased from 11,901,122.95 L.E. after simulation experiment to 11,342,526.15 L.E. after optimization experiment, this means that the total cost decreased by 558,596.8 L.E. which represent 4.7% of the total cost of the substation model. The total reduction in direct cost represents 2.72% of the total cost and the total reduction in indirect cost represents 1.98% of the total cost of the model.

TABLE. 2: Comparison between model cost results after simulation and optimization experiments

Description	Cost after Simulation (L.E.)	Cost after Optimization (L.E.)	Reduction in Cost (L.E.)	Percentage of Reduction in Cost
Zone 1 (Direct Cost)	4,493,088.50	4,331,173.50	161,915.00	3.60%
Zone 2 (Direct Cost)	3,847,338.75	3,696,488.75	150,850.00	3.90%
Zone 3 (Direct Cost)	842,694.40	831,529.40	11,165.00	1.32%
Total (Direct Cost)	9,183,121.65	8,859,191.65	323,930.00	3.53%
Indirect Cost	2,718,001.30	2,483,334.50	234,666.80	8.63%
Total	11,901,122.95	11,342,526.15	558,596.80	4.70%

In the simulation model, the values of resource parameters *(see section 5.4.4)* were read from the data base excel file as input data. Each resource parameter represents the number of resources assigned to each activity from each type of resource. These values are the actual data of Elnabq substation. So, during simulation experiment, the model was searching for the best work sequence that achieves this resources allocation under resources constraints. While, in optimization experiment, the values of resource parameters are considered variables that range between minimum and maximum values *(see section 5.6.2).* The minimum value for any parameter was always 1, while the maximum value for each parameter was the capacity of resource pool *(see section 5.4.2)* in which this resource parameter withdraws from it (i.e. the maximum available number of resources in this resource pool in which this resource parameter withdraws its value from it) and the increment step was 1 *(see Fig. 15).* Due to the implementation of genetic algorithm and parameter variation experiment as discussed in section 5.6, the process was repeated using different combinations of resource parameters and genetic operators to enhance the results under resources constraints until the optimum resource parameters that attain the minimum model time were obtained. This optimum resource allocation affected the sequence of work, so some buildings started before other buildings or vice versa, also some activities were postponed and others started earlier to finally achieve the optimum work sequence, while still satisfying project logic constraints and activities logical relationships . This variation in time, in return affected the cost. The cost of equipment and manpower crews was defined in the model as unit cost per day. As the number of working days for equipment or manpower crews decreased, the cost also was decreased. Also, the indirect variable cost was defined in the model as unit cost per day, so this cost was decreased as a result of reduction in total model time units.

The combined simulation/ optimization process to find the optimum solution took approximately; 2602.86 seconds, 2936.56 seconds and 3114.53 seconds for zone 1, zone 2 and zone 3 respectively *(see Table. 1).* The total time consumed was 8653.95 seconds (2.4 h) on a Pentium 900 MHz machine, which is reasonable considering the potential time and cost savings.

It is important to analyse the effect of parameters on reduction of total cost of the model to determine the parameters that have great effect and those that have moderate, low or no effect. Fig. 16 shows the percentage of reduction of each resource pool cost on total direct cost of the model after optimization, sorted in a descending order. Generally, it can be said that the resource pools that are related to concrete and skeleton activities like Formwork Crews, Steel Reinforcement Crews, Trucks and Pump are more effective on reduction in cost than those pools related to finishing activities like Masonry Crews, Tiles Crews and Aluminium Crews. While Plaster and Paint crews pools have moderate effect. And this returns to three reasons. First, the concrete and skeleton clauses in electrical power substations projects represent higher percentage of direct cost than finishing clauses. Second, the difficult nature of substations buildings like GIS building, Control Building or Elevated Tank requires high skilled structure work labours which need more cost and time, because they are one story buildings with large heights and areas, connected with large section beams located on different levels along the height of the building. Third, the construction of substation projects is horizontal construction work, not vertical work as construction of conventional buildings, hospitals or factories, which in turn, make it difficult to make such great overlap between structure and finishing works.

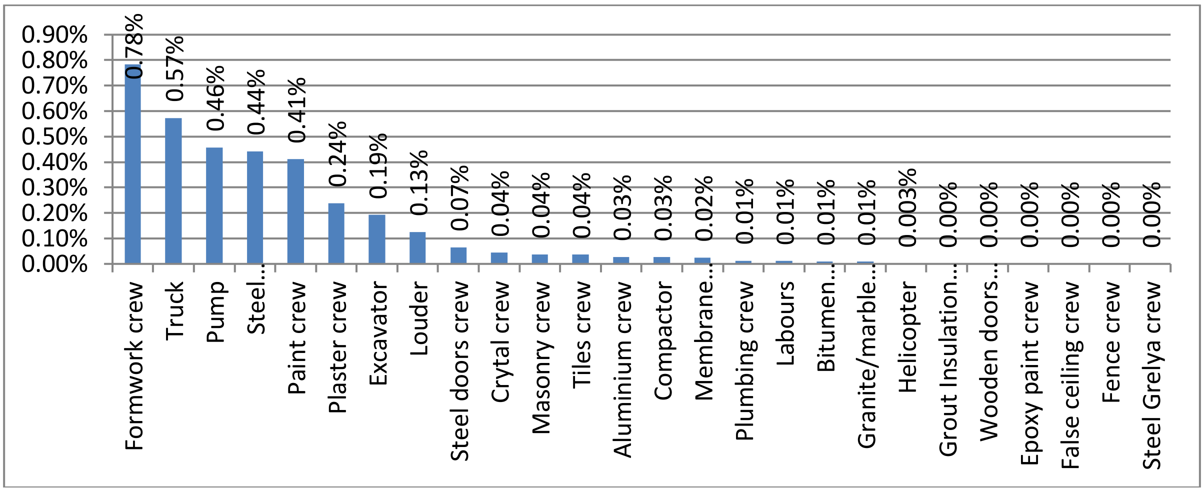


FIG. 16: Distribution of percentage of reduction in direct cost of model on resource pools

5.7.3 Comparison between proposed model and previous research

Table. 3 shows a comparison between the developed model and previous similar research. All the previous studies used for the comparison were based on genetic algorithm approach for solving the optimization problem in order to make the comparison more accurate. All these studies were trying to solve time-cost-resource optimization problem except for the third study (Haque and Hasin, 2012), this approach was solving time-cost optimization problem without considering resource factor. It can be observed from Table. 3 that the proposed model attained saving values in time and cost more than the previous approaches. As observed, when applying Dawood and Sriprasert (2006) approach on a case study including 9 activities and 3 types of resources, it attained 6.6% saving in time from the original project time, but without any saving or increase in cost. Chen and Weng (2009) approach was applied on a case study including 37 activities and 1 type of resource, the approach achieved 2.08% saving in time and 0.23% saving in cost. By applying Haque and Hasin (2012) approach on a 7-activities case study, it attained 6.8% saving in time and 5.45% saving in cost, but the approach didn’t consider resource limited availability or resource levelling as an effective factors. Also, Kim et al (2016) approach that was applied on a case study of 9 activities and 3 types of resources, obtained 8.9% saving in time, while 26.3% increase in cost. But, this approach depends on unlimited resources availability. The proposed model when applied on Elnabq substation that includes 571 activities and 28 types of resources, it obtained saving in time and cost by 10.25% and 4.7% respectively. It is noticed from this discussion that the developed model achieved better results on solving the time-cost-resource optimization problem than the previous approaches; however, the size of the case study used in this research is extremely larger than those used in the other studies.

The added value of the proposed approach compared to the previous studies returns to the following:

- All the previous mentioned studies depend on the traditional methods in project scheduling, like; critical path method (CPM), which is ineffective in actualizing multiple resources sharing procedure as it deals predominantly with activities’ durations not with resources which might bring about large resource fluctuations. On the contrary, the proposed model is based on the discrete event simulation method in scheduling, which is able to imitate the real construction operation over time and deal with the concept of multiple resource allocation and resource sharing according to their availability as well as time. This is done through the concept of resource pools which actualize the real process of resource sharing between activities, and obtains more accurate results.
- The previous approaches focus in dealing with changing project schedules and resource modification on nearby optimality at activity level, while the proposed model is dealing with resource allocation problem on global optimality at project level.

TABLE. 3: Comparison of results between the proposed model and previous research

No.	Source	Problem	Objective function	Constraints	No. of activities	No. of resource types	Percentage of saving/ increase in time	Percentage of saving/ increase in cost
1	Dawood and Sriprasert (2006)	Multi- constraint optimization	Minimize time, cost, resource fluctuation	Activity dependency, limited working area, resource and information readiness	9	3	6.6% saving	0%
2	Chen and Weng (2009)	Time–cost-resource optimization	Minimize time, cost	Resource availability constraints	37	1	2.08% saving	0.23% saving
3	Haque and Hasin (2012)	Time-cost optimization with unlimited resources availability	Minimize cost	Min. and max. activities durations and costs and targeted project duration	7	NA	6.8% saving	5.45% saving
4	Kim et al (2016)	Time–cost-resource levelling optimization with unlimited resources availability	Minimize time, cost, resource fluctuation	Min. and max. activities durations and costs	9	3	8.9%* saving	26.3%* increase
5	Proposed Model	Time–cost-resource optimization	Minimize time	Resource availability constraints	571	28	10.25% saving	4.7% saving

* Most achieved results.

- In the previous approaches, the relationship between time, cost and resources relied on the idea of normal cost, crashed cost, normal duration and crashed duration for each activity. So, during optimization process, when the activity duration decreased, the cost increased due to assigning more resources, increasing shifts or inadequate resource levelling and vice versa. It can be observed from Table.3 that the approaches that achieved saving in project time with adequate resource leveling, resulted in increasing in project cost if the approach depends on unlimited resources availability or nearly no saving in cost if it is resource constrained. The proposed model tried to solve the optimization problem from more efficient view, the model focuses on searching for the optimum resource allocation through changing the work sequence while keeping the logical relationships. This optimum work sequence that allocates the resources by the way that minimizes the number of working days for resources. When the project is completed with less number of working days, the total project duration will be decreased. And, as the cost of each resource is defined in the model as unit per day, so decreasing the number of working days, in return, decreases cost. Also, the indirect variable cost is defined as unit cost per day, so it is decreased by decreasing the total project duration.
- Parameter variation experiment allows a wide range of combinations between parameters values to find the optimum combination. While in the previous research, the total number of possible combinations is restricted by the entered possible options for duration of each activity and the entered possible options for the combinations between the assigned numbers from each type of resource for each activity. All these data are input data should be entered by the user, which restrict the wide possibility of solutions.
- In the previous research, the model is restricted by the specific alternative modes of execution or work sequence entered as inputs by the user, while in the proposed model all possible work sequences are available, so the model can choose the optimum work sequence that attains the best resource allocation and minimum project time.

6. CONCLUSION

Electrical substation construction projects are considered very important projects. An electrical substation is a subsidiary station of an electricity generation, transmission and distribution system where voltage is transformed from high to low or the reverse using transformers. Electric power may flow through several substations between generating plant and consumer, and may be changed in voltage in several steps. Scheduling and optimization the execution process of civil works of electrical substation are considered complex and challenging tasks. The electrical substation consists of several phases working in parallel to each other; each phase includes a lot of components and requires different resources which are shared by different buildings at the same time.

In this paper, two models are presented for electrical substation construction projects; a simulation model and an optimization model. The models are constructed by discrete event simulation method using Anylogic software. A data base excel file is constructed for input data entry. So, the model can be considered as a template for execution process of civil works of any substation project. The aim of the simulation model is to obtain the total time units and cost of the substation according to the maximum available resources specified by user and according to input data. The optimization model is built also on the same model by using parameter variation experiment and implementing genetic algorithm (GA) java code on a java class using Anylogic platform to obtain the minimum model time units, the optimum sequence of work corresponding to this time, the optimum resource allocation and the corresponding cost. The available resources and the resource parameters are considered the model constraints and decision variables respectively.

The models are tested on a real case study which is Elnabq substation, located in Sharm Elsheikh city, Egypt. The models demonstrate efficiency by attaining a reduction in model time units between simulation and optimization experiments of 10.25% and reduction in total cost of 4.7%. An analysis to determine the effect of each resource on reduction in cost is included. The analysis indicates that the resources that are related to concrete and skeleton activities are more effective on reduction in cost than finishing activities. The model is limited to the scope of civil works without considering the electromechanical works or the effect of risk factors on time and cost.

7. REFERENCES

Ashley D.B. (1980). Simulation of repetitive unit construction, *Journal of the construction division*, Vol. 106, No. C02, 185-194.

Ashuri B. and Tavakolan M. (2015). Shuffled frog-leaping model for solving time-cost-resource optimization problems in construction project planning, *ASCE Journal of computing in civil engineering*, Vol. 29, No. 1.

Banks J., Carson J., Nelson B. and Nicol D. (2001). Discrete-Event System Simulation, *Prentice Hall.*

Bayraktar M.E. (2006). A decision model to determine the optimal time-cost trade off in highway work zone projects, *Partial fulfilment of the requirements for the degree of doctor of philosophy, Purdue University,*

Chen P.H. and Weng H. (2009). A two-phase GA model for resource constrained project scheduling, *Automation in Construction,* Vol. 18, No. 4, 485–498.

Dawood N. and Sriprasert E. (2006). Construction scheduling using multi-constraint and genetic algorithms approach, *Construction management and economics*, Vol. 24, No. 1, 19-30.

Elazouni A. (2009). Heuristic method for multi-project finance based scheduling, *Construction management and economics,* Vol. 27, No. 2, 199–211.

Elazouni A.M. and Gab-Allah A.A. (2004). Finance-based scheduling of construction projects using integer programming, *ASCE Journal of construction engineering and management,* Vol. 130, No. 1, 15–24.

El-Rayes K. and Kandil A. (2005). Time-cost-quality trade-off analysis for highway construction, *ASCE Journal of construction engineering and management*, Vol. 131, No. 4, 477–486.

Feng C.W., Liu L. and Burns S.A. (2000). Stochastic construction time-cost trade-off analysis, *ASCE Journal of computing in civil engineering,* Vol. 14, No. 2, 117-126.

Flynn T., Tian Y., Masnick K., Huynh E., Mair A., McDonnell G. and Osgood N. (2014). Discrete choice, agent based and system dynamics simulation of health profession career paths, *Proceedings of the 2014 winter simulation conference* (Tolk A., Diallo S. Y., Ryzhov I. O., Yilmaz L., Buckley S. and Miller J.A., editors), Institute of Electrical and Electronics Engineers, 1700-1711.

Fondahl J. W. (1961). A non-computer approach to the critical path method for the construction industry, *Technical report No. 9, the construction institute, Department of civil engineering, Stanford University, Stanford, CA.*

Gomar J.E., Haas C.T. and Morton D.P. (2002). Assignment and allocation optimization of partially multi skilled workforce, *ASCE Journal of construction engineering and management,* Vol. 128, No. 2, 103–109.

Grimm V. and Railsback S.F. (2005). Princeton series in theoretical and computational biology: Individual-based modeling and ecology, *Princeton University Press,* Princeton, NJ, USA.

Haque K.M.A. and Hasin M.A.A. (2012). Genetic algorithm for project time-cost optimization in fuzzy environment*, Journal of industrial engineering and management,* Vol. 5, No. 2, 364–381.

Hegazy T. and Kassab M. (2003). Resource optimization using combined simulation and genetic algorithms, *ASCE Journal of construction engineering and management*, Vol. 129, No. 6, 698-705

Hegazy T., Elhakeem A. and Elbeltagi E. (2004). Distributed scheduling model for infrastructure networks, *ASCE Journal of construction engineering and management*, Vol. 130, No. 2, 160- 167.

Hossain L. (2010). An optimization model for the dynamic multi-project environment in construction, *Ph.D. dissertation*, University of Galgary, Alberta.

Isidore L.J. and Back W.E. (2002). Multiple simulation analysis for probabilistic cost and schedule integration, *ASCE Journal of construction engineering and management*, Vol. 128, No. 3, 211-219.

Jaskowski P. and Sobotka A. (2006). Scheduling construction projects using evolutionary algorithms, *ASCE Journal of construction engineering and management*, Vol. 132, No. 8, 861–870.

Kallantzis A. and Lambropoulos S. (2004). Critical path determination by incorporating minimum and maximum time and distance constraints into linear scheduling, *Engineering, construction and architectural management*, Vol. 11, No. 3, 211–222.

Kavanagh D.P. (1985). SIREN: A Repetitive construction simulation model*, ASCE Journal of construction engineering and management*, Vol. 111, No. 3, 308-323.

Kim J.L. and Ellis Jr. R.D. (2008). Permutation-based elitist genetic algorithm for optimization of large-sized resource-constrained project scheduling, *ASCE Journal of construction engineering and management*, Vol. 134, No. 11, 904–913.

Kim J.L. and Ellis R. (2010). Comparing schedule generation schemes in resource-constrained project scheduling using elitist genetic algorithm, *ASCE Journal of construction engineering and management*, Vol. 136, No. 2, 160–169.

Kim K., Walewski J. and Cho Y.K. (2016). Multiobjective construction schedule optimization using modified niched pareto genetic algorithm, *ASCE Journal of management in engineering*, Vol. 28, No. 3, 265–272.

König M. and Beißert U. (2009). Construction scheduling optimization using simulated annealing, *Proceedings of the 26th international symposium on automation and robotics in construction (ISARC).* 24–27 June, Austin, TX.

Labban R., AbouRizk S., Haddad Z. and Elsersy A. (2013). A discrete event simulation model of asphalt paving operations, *Proceedings of the 2013 winter simulation conference* (Pasupathy R., Kim S.-H., Tolk A., Hill R., and Kuhl M. E., editors).

Liu M. and Li H. (2003). Resource-activity critical-path method for construction planning. *ASCE Journal of construction engineering and management,* Vol. 129, No. 4, 412–420.

Moselhi O. (1993). Schedule compression using the direct stiffness method, *Canadian journal of civil engineering,* Vol. 20, No. 1, 65–72.

Moselhi O. and El-Rayes K. (1993). Scheduling of repetitive projects with cost optimization, *ASCE Journal of construction engineering and management,* Vol. 119, No. 4, 681–697.

Moussourakis J., and Haksever C. (2004). Flexible model for time/cost tradeoff problem, *ASCE Journal of construction engineering and management,* Vol. 130, No. 3, 307–314.

Peña-Mora F., Han S., Lee S., and Park M. (2008). Strategic-operational construction management: hybrid system dynamics and discrete event approach, *ASCE Journal of construction engineering and management*, Vol. 134, No. 9, 701–710.

Prager W. (1963). A structured method of computing project cost polygons, *Management science,* Vol. 9, No. 3, 394–404.

Robinson D. R. (1975). A dynamic programming solution to cost time trade-off for CPM, *Management science* Vol. 22, No. 2, 158–166.

Robinson S. (2004). Simulation: the practice of model development and use, *Wiley*.

Siemens N. (1971). A simple CPM time-cost trade off algorithm, *Management science,* Vol. 17, No. 6, 354–363.

Yu J. and Buyya R. (2008). Workflow scheduling algorithms for grid computing, *Grid computing and distributed systems laboratory at melbourne university, Australia [Online]*, URL:http://www.buyya.com/papers/MHS-Springer-Jia2008.pdf.

Zaheer N. (2000). Project/Process simulation modeling using discrete event construction simulation language stroboscope, *Master thesis*, King Fahd University of Petroleum and Minerals.

Zankoul E. and Khoury H. (2014). Simulation of on-shore wind farm construction process in lebanon, *The 31st international symposium on automation and robotics in construction and mining (ISARC).*

Zankoul E., Khoury H. and Awwad R. (2015). Evaluation of agent-based and discrete-event simulation for modeling construction earthmoving operations. *Conference paper*, http://www.anylogic.com/upload/iblock/2f7/2f7dd9078751699626333a930ab9c87d.pdf.

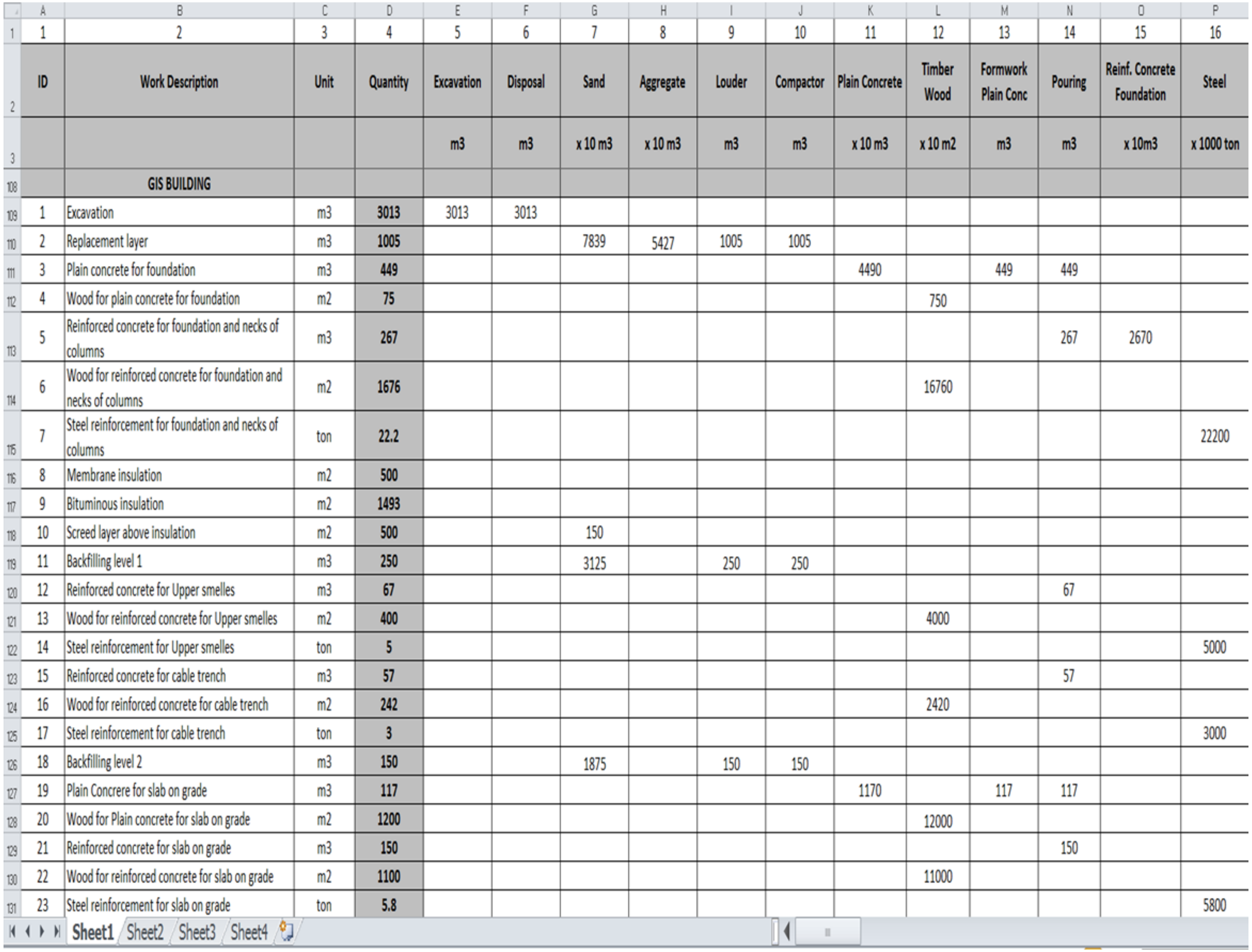
Zhang H., Li H. and Tam C. M. (2006). Heuristic scheduling of resource-constrained, multiple-mode and repetitive projects, *Construction management and economics*, Vol. 24, No. 2, 159–169.

Zheng D.X.M., Ng S.T. and Kumaraswamy M. M. (2005). Applying pareto ranking and niche formation to genetic algorithm-based multiobjective time–cost optimization, *ASCE Journal of construction engineering and management*, Vol. 131, No. 1, 81–91.

Zheng D.X.M., Ng S.T. and Kumaraswamy M.M. (2004). Applying a genetic algorithm-based multiobjective approach for time-cost optimization, *ASCE Journal of construction engineering and management*, Vol. 130, No. 2, 168–176.

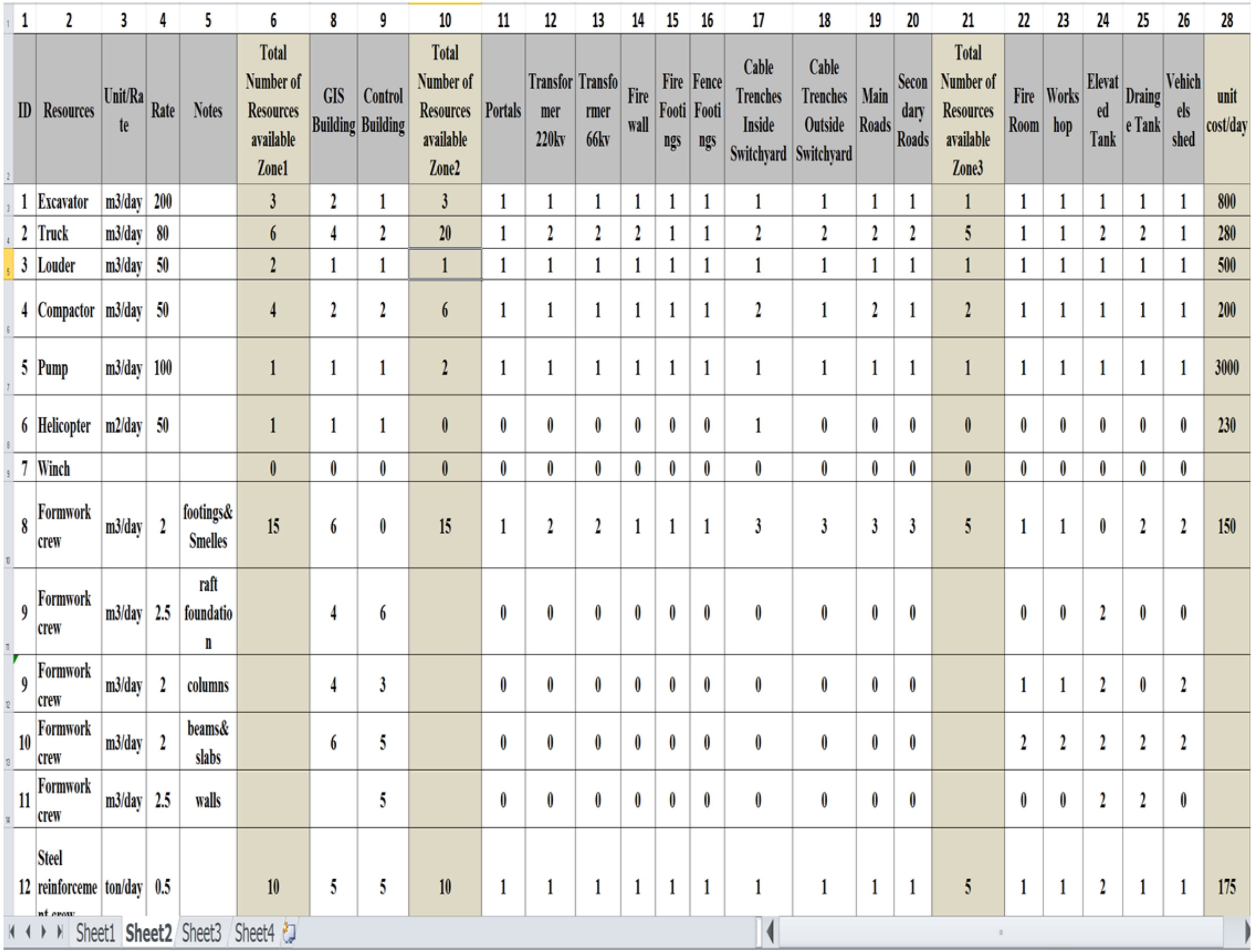
Zhou J., Love P., Wang X., Teo K. and Irani Z. (2013). A review of methods and algorithms for optimizing construction scheduling, *Journal of the operational research society,* Vol. 64, 1091–1105.

APPENDIX A: PART OF DATA BASE SHEET 1 EXCEL FILE



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	ID	Work Description	Unit	Quantity	Excavation	Disposal	Sand	Aggregate	Louder	Compactor	Plain Concrete	Timber Wood	Formwork Plain Conc	Pouring	Reinf. Concrete Foundation	Steel
3					m3	m3	x 10 m3	x 10 m3	m3	m3	x 10 m3	x 10 m2	m3	m3	x 10m3	x 1000 ton
108		GIS BUILDING														
109	1	Excavation	m3	3013	3013	3013										
110	2	Replacement layer	m3	1005			7839	5427	1005	1005						
111	3	Plain concrete for foundation	m3	449							4490		449	449		
112	4	Wood for plain concrete for foundation	m2	75								750				
113	5	Reinforced concrete for foundation and necks of columns	m3	267										267	2670	
114	6	Wood for reinforced concrete for foundation and necks of columns	m2	1676								16760				
115	7	Steel reinforcement for foundation and necks of columns	ton	22.2												22200
116	8	Membrane insulation	m2	500												
117	9	Bituminous insulation	m2	1493												
118	10	Screed layer above insulation	m2	500			150									
119	11	Backfilling level 1	m3	250			3125		250	250						
120	12	Reinforced concrete for Upper smelles	m3	67										67		
121	13	Wood for reinforced concrete for Upper smelles	m2	400								4000				
122	14	Steel reinforcement for Upper smelles	ton	5												5000
123	15	Reinforced concrete for cable trench	m3	57										57		
124	16	Wood for reinforced concrete for cable trench	m2	242								2420				
125	17	Steel reinforcement for cable trench	ton	3												3000
126	18	Backfilling level 2	m3	150			1875		150	150						
127	19	Plain Concrere for slab on grade	m3	117							1170		117	117		
128	20	Wood for Plain concrete for slab on grade	m2	1200								12000				
129	21	Reinforced concrete for slab on grade	m3	150										150		
130	22	Wood for reinforced concrete for slab on grade	m2	1100								11000				
131	23	Steel reinforcement for slab on grade	ton	5.8												5800

APPENDIX B: PART OF DATA BASE SHEET 2 EXCEL FILE



1	2	3	4	5	6	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	28
ID	Resources	Unit/Rate	Rate	Notes	Total Number of Resources available Zone1	GIS Building	Control Building	Total Number of Resources available Zone2	Portals	Transformer 220kv	Transformer 66kv	Fire wall	Fire Footings	Fence Footings	Cable Trenches Inside Switchyard	Cable Trenches Outside Switchyard	Main Roads	Secondary Roads	Total Number of Resources available Zone3	Fire Room	Workshop	Elevated Tank	Draing e Tank	Vehichels shed	unit cost/day
1	Excavator	m3/day	200		3	2	1	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	800
2	Truck	m3/day	80		6	4	2	20	1	2	2	2	1	1	2	2	2	2	5	1	1	2	2	1	280
3	Louder	m3/day	50		2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	500
4	Compactor	m3/day	50		4	2	2	6	1	1	1	1	1	1	2	1	2	1	2	1	1	1	1	1	200
5	Pump	m3/day	100		1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3000
6	Helicopter	m2/day	50		1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	230
7	Winch				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8	Formwork crew	m3/day	2	footings& Smelles	15	6	0	15	1	2	2	1	1	1	3	3	3	3	5	1	1	0	2	2	150
9	Formwork crew	m3/day	2.5	raft foundation		4	6		0	0	0	0	0	0	0	0	0	0		0	0	2	0	0	
9	Formwork crew	m3/day	2	columns		4	3		0	0	0	0	0	0	0	0	0	0		1	1	2	0	2	
10	Formwork crew	m3/day	2	beams& slabs		6	5		0	0	0	0	0	0	0	0	0	0		2	2	2	2	2	
11	Formwork crew	m3/day	2.5	walls			5		0	0	0	0	0	0	0	0	0	0		0	0	2	2	0	
12	Steel reinforcement crew	ton/day	0.5		10	5	5	10	1	1	1	1	1	1	1	1	1	1	5	1	1	2	1	1	175