Construction Time-Cost Optimization Modeling Using Ant Colony Optimization

Mohammed Nooruldeen Azeez (1), Angham Alsaffar (2)
mohammed.n.azeez@gmail.com, anghamalsaffar@yahoo.com
M.Sc. Candidate Baghdad University/ Civil Eng. Dep. (1), Prof. Ph.D. Baghdad University/ Civil Eng. Dep. (2)

ABSTRACT

In the field of construction project management, time and cost are the most important factors to be considered in planning every project, and their relationship is complex. The total cost for each project is the sum of the direct and indirect cost. Direct cost commonly represents labor, materials, equipment, etc.

Indirect cost generally represents overhead cost such as supervision, administration, consultants, and interests. Direct cost grows at an increasing rate as the project time is reduced from its original planned time. However, indirect cost continues for the life of the project and any reduction in project time means a reduction in indirect cost. Therefore, there is a trade-off between the time and cost for completing construction activities.

In this research, modeling of time-cost optimization, generating global optimum solution for time and cost problem, and lowering construction time and cost using ant colony optimization algorithm.

KEYWORDS: Time-Cost Optimization, Time-Cost Trade-off, Ant Colony Optimization.

موديل رياضي لأمثلية الوقت والكلفة الانشائية بأستخدام امثلية مستعمرة النمل

الخلاصة

ان الوقت والكلفة هي اهم العوامل المأخوذة في تخطيط اي مشروع، في اختصاص ادارة المشاريع الانشائية. وان العلاقة بين الوقت والكلفة معقدة. فأن الكلفة الكلية في اي مشروع تمثل مجموع الكلف المباشرة وغير المباشره، وتمثل الكلف المباشره كلف العمالة والمواد والمعدات ، الخ

بينما الكلف غير المباشره تمثل بصورة عامة مصاريف الاشراف و الاداريات والاستشارية اضافة الى الفوائد.

الكلف المباشرة تزداد بنسبة كلما طال عمر المشروع عن عمره المقرر ، بينما الكلف غير المباشره تستمر طيلة عمر المشروع وان اي تقليل في مدة المشروع عن المدة المحددة تعني تقليل الكلف غير المباشره ، لذا فان هنالك مقايضة بين الكلفة والوقت في اكمال الفعاليات الانشائية.

في هذا البحث ، سوف يتم انشاء موديل لحساب امثلية الوقت-الكلفة الانشائية. توليد حلول مثلى عالمية للوقت والكلفة، تقليل الوقت والكلفة بأستخدام طريقة مستعمرة النمل.

LITERATURE REVIEW

Time–cost trade-off analysis is as an important aspect of any construction project planning, and it is an interesting subject for both researchers and contractors, due to the academic / real field nature of the problem.

Construction time-cost problems were tackled repeatedly in the past decades using different methods and modeling techniques are classified into three types heuristic, mathematical and evolutionary based algorithms.

Ant Colony Optimization (ACO) is introduced as a new approach for deriving approximate solutions for computationally sophisticated problems, using the Traveler Salesman Problem TSP as an example application **(Dorigo et. al., 1991)**. Since that, ACO has been employed to solve various problems, such as no-wait flow shop scheduling routing problems, etc.

In addition, ACO Algorithms was and is still widely used to solve various discrete problems and still to date applied for solving complex optimization problems **(Blum, 2005)**, ACO is also widely used in Civil Engineering and for different applications

Even though ACO algorithms were developed to solve TSP problems, the extensive use of ACO in Civil Engineering and other sciences, and the outstanding performance of ACO algorithms provided the motive for applying ACO in TCO problems.

This research will use ACO searching behavior for developing a Time-Cost Optimization Model .

Heuristic Models:

(Prager, 1963) showed that Time-Cost algorithm can be given a structural interpretation, using concepts that are familiar to civil engineers.

(Siemens, 1971) introduced an algorithm (Siemens Approximate Algorithm) for efficiently shortening the duration of a project when the expected project duration exceeds a predetermined limit. The problem consists of determining which activities to expedite and by what amount. The objective is to minimize the cost of the project.

(Goyal, 1975) modified Siemens algorithm for shortening the duration of a project when the expected duration of the project exceeds a predetermined limit. Goyal redefined "effective cost slope", by selecting the activity with minimum effective cost slope, and simultaneously de-shortening appropriate activities on adequately shortened paths while shortening the selected activity,

(Al-Samaraai, 2005) used the typical cost slope approach that managers take in making time-cost trade-off and presented a crashing program.

Heuristic methods are widely used for their simplicity and general ability to produce good results, and they do not require a complicated calculations however they are problem dependent. Their results vary on different cases. In addition, despite the good solutions they provide, they do not guarantee optimality. Also most heuristic methods assume only linear time-cost relationships within activities. in addition, the solutions obtained by heuristic methods do not provide the range of possible solutions, making it difficult to experiment with different scenarios for what-if analyses **(Feng e. al., 1997)**, **(Feng e. al., 2000)**, and **(Hegazy, 2002).**

Mathematical Models:

(Kelly, 1961) established a mathematical basis For Crashing Cost in Critical-Path Scheduling Method. The essential ingredient of the technique is a mathematical model that incorporates sequence information, durations, and costs for each component of the project by using linear programming.

(Patterson et. al., 1974) studied minimum duration schedules for the resource constrained by using bounding techniques in conjunction with zero-one programming to solve project scheduling problems. The developed algorithms consist of examining the feasibility of a series of zero-one programming problems.

(Robinson, 1975) presented a model that involves a dynamic-programming approach to determine the allocation which minimizes the duration of the project (critical path). They presented a model able to determine the optimum allocation for networks of activities with computational shortcuts for functions with special properties used to increase the efficiency of their model.

(Hendrickson, et. al, 1989) used linear programming, and presented many solved examples for their method; however, the model was suitable only for problems with linear time-cost relationships.

(Liu et al, 1995) provided a hybrid method to solve Time-Cost trade-off problems using mathematical models. Their method takes advantage of linear programming for efficiency, and integer programming to find the exact solutions.

(Chassiakos et. al., 2005) incorporated important characteristics such as precedence relationships between activities, external time constraints, activity planning constraints, and bonuses/penalties for early/delayed project completion projects, which provide more realistic representation of actual construction in the analysis and two solution methods (exact and approximate) are developed, The exact method utilizes a linear/integer programming model to provide the optimal project time-cost curve and the minimum cost schedule considering all activity time-cost alternatives together.

The main criticisms to mathematical programming models is their complex formulations, computational-intensive nature, and applicability to small-size problems **(Feng et. al., 1997)**, and **(Feng et. al., 2000)**.

Although the heuristic methods and mathematical approaches have their specific strengths, their weaknesses are also obvious especially as both techniques may not always lead to optimal solutions. The future seems to favor EOAs **(Zheng et. al., 2005)**, and **(Ng. et. al, 2008).**

Another major deficiency of those methods is their inability to handle more than one objective. In addition, these methods are built upon the hill climbing algorithms, which has only one randomly generated solution exposed to some kind of variation to create a better solution. Therefore, it is questionable as to whether the solution is a "Global" optimal one **(Feng et. al., 1997)**, and **(Feng et. al., 2000)**.

Evolutionary-Based Optimization Algorithm (EOA) Models :

In an attempt to reduce processing time and improve the quality of solutions, particularly to avoid being trapped in local optima, EOAs have been introduced during the past 10 years **(Elbeltagi et. al, 2005)**.

EOAs are stochastic search methods that mimic the metaphor of natural biological evolution and/or the social behavior of species. The behavior of such species is guided by learning, adaptation, and evolution **(Hegazy, 2002)**, and **(Ng. et. al, 2008).**

(Feng et. al., 1997) presented an algorithm based on the principles of Genetic Algorithms (GAs) for construction time-cost trade-off, and a computer program that can execute the algorithm efficiently. The computer program used, TCGA, automates the execution of the new algorithm, and it provides a practical tool for practitioners to apply the algorithm in practice.

After that, **(Feng et. al., 2000)** developed their previous work by utilizing GAs with simulation techniques to imitate the probabilistic nature of project networks throughout the search of optimal solutions. The approach provides more realistic solutions for construction time-cost trade-off problem under uncertainty.

They also demonstrated that GAs can be integrated with simulation techniques to provide an efficient and practical means of assessing project time and cost risks.

(Li e. al., 1997) also introduced a genetic algorithm model to solve time-cost trade-off problems with less computation time of **(Feng et. al., 1997)**

(Hegazy, 2002) also developed GA model to solve time-cost trade-off problems and was able to minimize the number of calculation used to find the solution. Also he was able to present a computer program to solve TCT problems for both researchers and planners.

Despite its benefit, the time taken by a GA model to generate a near-optimum solution can be excessive. The main drawback of the GA-based applications is that they require large computational time for the search **(Feng et. al., 1997)**, and **(Ng. et. al, 2008).**

Another major drawback of GAs have to do with genetic drift which is typified by the existence of multiple peaks of equal height. When genetic drift occurs, it will converge to a single peak due to the stochastic errors during processing, and this is undesirable for any multi-objective TCO problems **(Feng e. al., 2000)**, and **(Zheng et. al., 2004)**.

GAs have been used extensively in the last decade to solve the TCT problem as mentioned above but Except for GA, other EOA techniques were inspired by different natural processes including the Ant Colony Optimization (ACO), Memetic Algorithms (MA), Particle Swarm Optimization (PSO), and Shuffled Frog Leaping Approach (SFL) etc. that were employed by **(Elbeltagi et. al, 2005)** for solving discrete time–cost trade-off problems.

(Elbeltagi et. al, 2005) developed five TCT models using all types of EOAs and provided better optimal solutions. They also conduct benchmark comparisons among the five algorithms for discrete time-cost trade-off problem to check the algorithm efficiency, in terms of processing time, convergence speed, and quality of the results. Based on this comparative analysis, some guidelines for determining the best operators for each algorithm were presented.

Although TCT problem has been extensively examined, all the researchers' only focused on minimizing the total cost for an early completion. This does not necessarily convey any reward to the contractor. However, clients and contractors are more concerned about the combined benefits and opportunities of early completion as well as cost savings. This has led to the development of the TCO concepts **(Zheng et. al., 2005)**, and **(Ng. et. al, 2008).**

TIME-COST OPTIMIZATION:

The time-cost optimization (TCO) problem is a multi-objective problem, which attempts to strike a balance between resource allocation costs and project schedule duration. TCO also is one of the greatest challenges in construction project planning, since the optimization of either time or cost would usually be at the expense of the other **(Afshar, et. al, 2009)**,and **(Kalhor et. al, 2011)**.

The goal of TCO is the same goal of any multi-objective optimization problems, which is to find the best compromise between multiple and conflicting objectives. In multi objective optimization, there is more than one solution which optimizes simultaneously all the objectives and there is no distinct superiority between these solutions. Therefore, we face a set of non-dominated solutions in these problems called Pareto optimal. Among the feasible solutions, a solution is identified as dominant if it is better than all other solutions in all of the considered objectives simultaneously. Among the feasible solutions, those belonging to Pareto front are known as non-dominated solutions, while the remainder solutions are known as dominated. Since none of the Pareto set solutions is absolutely better than the other non-dominated solutions, all of them are equally acceptable as regards the satisfaction of all the objectives **(Feng et. al., 2000)**, **(Zheng et. al., 2005)**, and **(Kasaeian, et. al, 2007).**

Time-Cost Optimization Models:

There are only few researches on TCO subject and they are summarized as follows:

(Zheng et. al., 2004) used GA and Pareto front approach; they developed a new algorithm for optimizing construction time-cost decisions. Their algorithm shows its efficiency by searching only a small fraction of the total search space. Its accuracy was verified by only small problems.

(Zheng et. al., 2004) compared their multi-objectives modified adaptive weight approach model with the previous single-objective models **(Hegazy, 2002)**; The test results of the deterministic scenario confirm that the new model can correctly locate the non-replaceable points on the segment of Pareto front within the limitation of time and cost.

They revealed that the model provides managers with greater flexibility to analyze their decisions in a more realistic manner.

Later on **(Zheng et. al., 2005)** applied a fuzzy sets theory to the original model to simulate uncertainty and produces better results especially as the risk increases, though it is not without weaknesses

Their model has significantly reduced the number of solutions generated for decision support, which is essential to multi-objective optimization, they proposed further refinements that are necessary to improve its efficiency when applied to large and complicated projects.

(Kasaeian, et. al, 2007) introduced a TCO model using Gas and a novel technique called Non-dominated Archiving to find the optimal solutions. Their model presented better solutions when compared to **(Zheng et. al., 2005)** with relatively higher computations.

(Xiong, et. al, 2008) presented a multi-objective TCO model using ACO as a searching tool, optimal solutions were generated, and outperformed (**Zheng et. al., 2004)** GA model results.

(Ng. et. al, 2008) also used ACO to find Time and cost optimality, the model was formulated and implemented on a commercial planning software. When performance compared with **(Zheng et. al., 2005)** model on large scale problems, results reveled better solutions for ACO model.

(Afshar, et. al, 2009) Adopted **(Kasaeian, et. al, 2007)** Non-Dominated Archiving technique and developed TCO model using multi colony ant algorithm. Results comparison with **(Ng. et. al, 2008)** model favored Non-dominant model, but no improvement in the solution from the original GA model, with the same calculation time.

Time-Cost Models can be summarized in **Table 1**

MODELING TCO

To solve the TCO problem, project network for TCO must be considered as a graphic network. Firstly the project is converted to an Activity-On-Arrow (AOA) network as shown in Fig. **(1).** The performance of ACO algorithms in TSP can be seen as a reference for ACO-based TCO model.

In this network, the events (1, 2, 3… etc.) could be regarded as nodes and the different options for each activity linking them would be the “distance” between these nodes.

Activity B in Fig. (1) could be taken as an example, there are three method options to complete this activity and they could be marked as B1, B2 and B3, and they are the different “distance” from event ①to event ④. Therefore, in ACO, the ants will travel from the first event ①to the event ⑦with proper options selected for each activity.

Like the shortest tour being set as the objective in TSP, the objectives for TCO would be the minimal time and lowest cost. On the other hand, there are many differences between TSP and TCO, for example, ants should not come back to the starting points in TCO which is otherwise necessary in TSP; despite the numerous differences, ACO could competently handle TCO problems.

MODEL DEVELOPMENT

To create an efficient optimization tool for Time and Cost and by considering the needs of the decision maker an evolutionary based model is created and developed, taking into account the strength and weaknesses of previous methods stated in the literature review for making this proposed model.

The developed model will transform time and cost from single option in an activity to optimum solutions that a construction project could be executed.

The model development process is divided into five important parts

- (i) Fitness function.
- (ii) Time and cost functions
- (iii) Ant Colony Solution Algorithm ACSA
- (iv) Pareto Front

• Fitness Function

Fitness Function (F.F.) represents planners and decision maker’s goal or what their needs are. To address a multi-objective optimization problem such as time, cost and to properly evaluate the solutions generated by the ACO, the fitness function consider both objectives and must be calculated during the ant colony solution algorithm part.

The FF. used is called Modified Adaptive Weight Approach (MAWA) which integrates both time and cost into one single function and prioritize the time and cost according to the solutions found by the ant colony solution algorithm part, accordingly to be used again in the algorithm for evaluating these solutions and finding the best ant in any iteration of the model calculation.

This approach is effective and able to optimize time and cost concurrently and generate optimal solution **(Zheng et. al., 2004),**and **(Kalhor et. al, 2011).**

The weights can be calculated using **(Zheng et. al., 2005)** *equations :*

- If $Z_t^{max} = Z_t^{min}$ and $Z_c^{max} = Z_c^{min}$:

Where:

Z_t^{max}, Z_c^{max} : maximal value for the objective of time and total cost in the current iteration.

Z_t^{min}, Z_c^{min} : minimal value for the objective of time and total cost in the current iteration.

$W_t = W_c = 0.5$ (eq.1)

Where:

W_t, W_c*:* the adaptive weight for the objective of time and total Cost.

- If $Z_t^{max} = Z_t^{min}$ and $Z_c^{max} \neq Z_c^{min}$:

$W_t = 0.1$, $W_c = 0.9$ (eq.2)

- If $Z_t^{max} \neq Z_t^{min}$ and $Z_c^{max} = Z_c^{min}$:

$W_t = 0.9$, $W_c = 0.1$ (eq.3)

- if $Z_t^{max} \neq Z_t^{min}$ and $Z_c^{max} \neq Z_c^{min}$:

$$V_t = \frac{Z_t^{min}}{Z_t^{max} - Z_t^{min}},\ V_c = \frac{Z_c^{min}}{Z_c^{max} - Z_c^{min}} \quad \text{(eq.4)}$$

$V = V_t + V_c$ (eq.5)

$$W_t = \frac{V_t}{V},\ W_c = \frac{V_c}{V} \quad \text{(eq.6)}$$

Where:

V_t, V_c, V represent Time, Cost, and project Value respectively.

This approach will generate optimal solution and will optimize both time and cost simultaneously.

The F.F. for any ant (k) in any given iteration will be:

$$f(k) = W_t \frac{Z_t^{max} - Z_t(k) + R}{Z_t^{max} - Z_t^{min} + R} + W_c \frac{Z_c^{max} - Z_c(k) + R}{Z_c^{max} - Z_c^{min} + R}$$

(eq.7) **(Zheng et. al., 2005)**

Where:

Z_t (k) and Z_c (k) is the time and cost function for ant (k) in the current iteration.

R is a positive random number between 0 and 1.

This approach imparts the ACO with greater freedom to search in the multi-objective space that overcomes the drawbacks of single objective and hill-climbing algorithms.

These weights will guide the algorithms to search through a wider range against the objectives that have a relatively small exploration space in previous generation.

- ## Time and cost Functions

To calculate Z_t (k) & Z_c (k) and prepare the necessary data to find the Non-dominant solution using Pareto front, two objectives were used for time and the total cost **designed for this purpose**

Time objective function

$$Z_t(k) = Max_{P_p \in P}\left[\sum_{i \in P_p} t_{ij}^{(k)} x_{ij}^{(k)}\right] \text{(eq. 8)}$$

(Researcher)

Where:

$t_{ij}^{(k)}$: Execution time of option j in the activity i selected by ant k.

$x_{ij}^{(k)}$: Index used to verify which option the ant selected to execute, if ant k selects option No. 4 then $x_{ij}^{(k)} = 1$, if not $x_{ij}^{(k)} = 0$.

P_p: Activity Sequence of certain path.

P: All the Paths in the network.

Cost Objective Function

$$Zc(k) = \left[\sum_{i \in L} C_{ij}^{(k)} x_{ij}^{(k)} + Z_t(k)(Ic)\right] \text{(eq. 9)}$$

(Researcher)

Where:

$c_{ij}^{(k)}$: Execution direct cost of option j in the activity i selected by ant k.

IC: project indirect cost rate.

L: number of project total activities.

- ## Ant Colony Solution Algorithm

This process is the key to find the optimal solutions and the beating heart (core) of the developed model that the decision makers need. It transforms the time and cost from being a single option in one activity to an optimal solution consisted of set of best options to execute the project with, throughout a sequence of intelligent and efficient steps to select the best option combinations and constantly update the resulted best solution found yet.

The closed cycle of searching, communicating, evaluating, and learning is the reason of solution continuous improvement. it is in a sensible balance between exploration of other solutions and exploitation of existing ones.

The Algorithm used in this model is ACS, due to its robustness in finding the shortest path and its low deviation from the optimal solution.

The ant colony solution algorithm is based upon **(Dorigo et. al. 2004)** ACS algorithm modified and developed to deal with construction projects networks ant to solve TCO

The ant colony solution algorithm contains four main steps :

1- Initiating ACS parameters
2- Creating the solutions
3- Path retracing and pheromone updating,

Figure (2) shows how and where the model development steps proceeds and interacts with each other

1- Initialization of ACSA parameters

Depending on **(Dorigo et. al., 2004)** and after conducting Parameter Sensitivity Analysis the necessary parameters are set to start ACSA:

m: is the number of ants in each iteration

t: no. of iterations in the model (optional)

L: no. of activities the user input

n: no. of options in each activity the user input

α: Coefficient represent the importance of the pheromone value (τ)

β: Coefficient represent the importance of the heuristic value (η)

τ0: initial pheromone value

ρ: global pheromone evaporation rate (0<ρ<1)

ε: local pheromone evaporation rate(0< ε <1)

q0: factor of the pseudorandom proportional action choice rule (0< q0<1)

Table (4.1) represents the parameters set by the researcher as the optimum parameter between exploration and exploitation; other parameters are variables and depend upon time-cost data that the user inputs (user oriented).

2- Creating the Solution

The strategy of the algorithm is to exploit information gathered from pervious iteration (pheromone trails) (τ_{ij}) and heuristic information (η_{ij}) calculated from input variables (option constructional properties) to construct candidate solutions and fold the information learned from constructing solutions again.

This step of the modeling starts by randomly generating solution using the first iteration (colony) to explore the environment and starts

learning what a good solution is. By evaluating the outcome of this iteration using MAWA to find the best solution, and then direct the next iteration (colony) towards the best solution found so far.
Fig. (2) shows that projects will have L activities where each activity has nL possibilities (construction options). To execute the selected activity, the model will have t number of iterations (colonies). In any colony there are m ants which will travel through the network to generate optimum solutions, This is the model complexity where Ant Colony Solution Algorithm can manage it efficiently yet it is easy for any user to use.
The bold line in fig. (3) represent the path taken by ant (K) selections and when the ant reaches the finishing point it will have a discrete time-cost solution. ACSA choices will be compared with other ants that crossed different paths to finish the project (each ant represents a candidate solution).
it can be seen that the ant does not travel from an activity to the next activity according to the network dependency no matter what they are (finish to start, start to start, start to finish, finish to finish). Ants travel in a numerical order, the model manage the ant solutions and transform the ant selected option to a candidate solution with respect to the activity relationships that the project is subjected to.
Ants select an option from n options by performing pseudorandom proportional rule , when the ant travel throughout the project network it will perform the same rule for every activity until all the activities have been passed.

Pseudorandom proportional rule:

$$Option_j = \begin{cases} Max.\left\{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}\right\} & if\ q \le q0; \\ J & otherwise, \end{cases}$$

(eq.10)

Where:

q : random variable between [0,1]

q0 : tunable parameter.

J: variable generated using random proportional rule (eq.12).

τ_{ij}: pheromone value of activity i option j .

η_{ij}: heuristic value of activity i option j and calculated according to this equation :

$$\eta_{ij} = \frac{1}{Cost_{ij} * time_{ij}} \text{ (eq.11)}$$

Fig. (4) depicts the variables in one single option (J) to execute an activity in a project, the figure also shows the variables that are subjected to change during model calculations from those which will not change. The construction variables (time and cost entered by user for this option) generates the ACSA (heuristic value and the pheromone value will be generated by the algorithm according to equations 11, 13,and15)respectively.

Random proportional rule:

$$P_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{j \in n_i^k}[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}} \text{ (eq. 12)}$$

Ant selects an option by generating a random No. (q) uniformly distributed between [0,1] where it is compared to q0 (0≤q0≤1).
If q≤q0, the ant selects the option with the higher value of both pheromone and heuristic value (using eq. 4.10); otherwise, the ant selects option from probability distribution created using (eq. 12)

3- Path Retracing and Pheromone Updating

This step contains two of the most important concepts in ACO (learning and communicating), which will contribute in improving the solution and exploring the solution environment, either by increasing the confidence in an option to become the most desirable choice or decrease it. This option will be the less desirable and almost neglected, without this step the solution will not change or improve.
The updating and evaporating occur only for the options selected by at least an ant, otherwise no changes will occur.
Pheromone Updating is divided in two types:

1- Local pheromone update**:** It retraces the ant k path and updates the pheromone value of the ant selection by applying equation (13). The importance of the local updating lies in using local evaporation rate to minimize the pheromone value of this option. To make this option less desirable for the ants in the same iteration, so they will not follow this ant directly to avoid premature convergence of the solution.
Therefore, achieving the maximum exploration possible in each iteration to and seek other options.

$$\tau_{ij}(t) = (1-\varepsilon)\tau_{ij} + \varepsilon.\tau 0 (eq.13)$$

Where:

ε: local evaporation rate (factor used to minimize the pheromone value of the selected option)

τ0: initial pheromone value of activity i and calculated by applying this equation:

$$\tau o_i = \frac{1}{L * \min(time_j * cost_j)} (eq.14)$$

The activity option which have min. (time * cost) will be selected to calculate the initial pheromone value because it has a rough estimation of the good solution.

τ_{ij} :pheromone trail of activity i option j in the previous iteration.

$\tau_{ij}(t)$: pheromone value of activity i option j in this iteration (t).

2- Global pheromone update: When all the ants of the colony (in single iteration) travel the network and generate discrete solutions, the algorithm points the ants of the next colony (iteration) to the path that achieved the best outcome so far by updating only the path of this ant, the so called iteration best solution or the iteration best ant which got the lowest value in the F.F.

The global pheromone updating is done according to the following equation:

$$\tau_{ij}(t) = (1-\rho)\tau_{ij} + \rho.\Delta\tau\ (eq.15)$$

Where:

ρ: global evaporation rate

Δτ: pheromone updating value for the option selected by the best ant in this activity and this iteration, calculated according to the following equation:

$$\Delta\tau = \frac{1}{F.F._{BS}\ value} (eq.16)$$

$F.F._{BS}$ value: result of the best ant Fitness function calculation.

This step is repeated for each activity until finishing the best ant traveling throughout its project path. When the next iteration starts the pheromone value in this option combination will be relatively higher and will lead to more ants selecting these options.

- **Optimal Non-Dominant Solution(s) Using Pareto Front**

In this step of the model development, the solution will be classified into optimum and not optimum to give the decision maker only the optimum solutions to be used. A solution pool created to contain the outcome of the Ant colony solution algorithm, as shown in Fig.(5).

In each iteration ten solutions will be generated (if the recommended parameter setting used) and these solutions will be added to the solution pool and then by applying Pareto front only the optimum solution will be selected.

For the next iteration there will be ten more options, If any new solution (x*) is better than that exists Pareto Front solution(x) with at least one objective, then the solution will be compared according to the performance of the other objective; otherwise, the new solution will be sent to the solution pool so as to reduce the unnecessary calculations.

But when a new solution is better than an existing PF selected solution for both objectives (time and cost) the inferior solution will be sent back to the solution pool. When the model stops, there will be only dominant optimal solutions left in the PF

Pareto Front constraints are:

If (x*) cost ≥ (x) cost and (x*) time ≤ (x) time ➔
add x* to Pareto front

If (x*) cost ≤ (x) cost and (x*) time ≥ (x) time➔
add x* to Pareto front

If (x*) cost ≥ (x) cost and (x*) time ≥ (x) time➔
add x* to solution pool

If (x*) cost ≤ (x) cost and (x*) time ≤ (x) time➔
add x* to Pareto front
add x to solution pool

MODEL EVALUATION

Model outcome tested and compared with credible and reliable references to confirm and validate the performance with other models.

This problem first will be applied to test the performance of the developed model, including seven activities problem presented by **(Feng, et. al., 1997)**, Table (3) represents the construction values of the problem, containing time and cost for each alternative option in every activity, and Fig. (6) represent the network of this problem, and the indirect cost was 1500$/day.

The activities contains 3 to 5 possible options (alternative), so the problem complexity will be [(3^5) * 4 * 5] = 4860 possible solutions. Although the project activities are only 7 but the problem complexity is considered as medium for civil engineering problems.

The project critical path calculated first to find the normal duration and for that the options selected were the normal time/normal cost, and the project time was 105 days, and 253700 $.

And after using TCO model the highest value of time was 68 days, and the cost was 220500$.

From the total cost 0.15%, and 54.4% of the project time were saved by using optimized values. This is achieved by using the saved the indirect cost to use more advanced equipment and increase the number of the crews or labors in the construction work, or a different construction method may be used.

As we can see from Table (4), the maximum possible total cost reached was 233500$ and it remains below both time and cost of the normal duration.

The comparison of the developed model solutions with two TCO models presented by **(Zheng, 2004)** using GA and **(Xiong, 2008)** using AS algorithm, is shown in Table (5).

By comparing the solutions of this problem with other reference solutions, the developed model shows the ability to generate global optimal solutions with an incredibly small time of 1 sec. using only 10 ants in 20 iterations while Xiong ACO model used 40 ants and 40 iterations to achieve the same solution, and Zheng used 5 as a population size in each of the 5 generations (iterations).

The developed model efficiency showed by searching only 4% of the possible solution [200/4860] and generated the global optimal solutions.

CONCLUTIONS

Ant Colony Optimization was able to generate optimal solutions in a fast and accurate way.

The developed model was able to generate global optimum solutions with less iterations and faster time compared to well-known time-cost optimization models.

Time and cost was optimized without dominating to only one function.

Time was saved by 54.4% while cost was 15% saved using the developed model.

Time-Cost Optimization Have A Great Effect On Lowering The Construction Time And Cost Of Construction Project In And Overcome The Delays And Cost Excess That Could Occurs During The Execution Of Any Construction Project

REFERENCES:

Afshar, A.; Ziaraty, A.; Keveh, A.; and Sharifi, F. , (2009) " Non-dominated Archiving Multi-colony Ant Algorithm in Time–Cost Trade-Off Optimization "Journal of Construction Engineering and Management, Vol. 135, No. 7, July 1, pages 668-674 ASCE,.

Al-Samaraai, H. Fulla (2005) "Developing Computer Program To Implement Time-Cost Relationship For Planning Construction Projects In Iraq" M.Sc. thesis submitted to the Civil Engineering Department, Al-Mustansria University.

Chassiakos A. P. and Sakellaropoulos S. P. (2005) "Time-Cost Optimization of Construction Projects with Generalized Activity Constraints" Journal of Construction Engineering and Management, Vol. 131, No. 10, October 1, 1115–1124 ASCE.

Dorigo M. and T. Stützle (2004). "Ant Colony Optimization". London, England, The MIT Press.

Dorigo M. and V. Maniezzo (1991). "Positive feedback as a search strategy". Dipartimento Di Elettronica - Politecnico Di Milano, pages 91-116.

Elbeltagi, E.; Hegazy, T. and Grierson, D. (2005), "Comparison among Five Evolutionary-Based Optimization Algorithms " Advanced Engineering Informatics, 19 , pages 43–53, Elsevier Ltd.

Feng C.-W. Liu L., and Burns S. A. (1997), "Using Genetic Algorithms To Solve Construction Time-Cost Trade-Off Problems", Journal of Computing in Civil Engineering, Vol. 11, No.3, July, Pages 184-189, ASCE.

Feng C.-W. Liu L., and Burns S. A. (2000) "Stochastic Construction Time-Cost Trade-Off Analysis" Journal Of Computing In Civil Engineering. - April 2000. - pp. 117-126 ASCE.

Goyal S. K. (1975), " A Note on A Simple CPM Time-Cost Tradeoff Algorithm" Management Science, Feb., Vol. 21, No. 6, Application Series jstor.

Hegazy, T. (2002). "Computer-based Construction Project Management" Upper Saddle River, NJ, Prentice Hall.

Hendrickson C. and Au T. (1989), "Project Management for Construction Fundamental Concepts for Owners, Engineers, Architects and Builders" first edition Prentice Hall.

Kalhor E., Khanzadi M., Eshtehardian E., Afshar A. (2011), "Stochastic time–cost optimization using non-dominated archiving ant colony approach", Automation in Construction. Elsevier.

Kasaeian, A.; Shoghli, O., and Afshar, A, (2007), "Non-dominated Archiving Genetic Algorithm for Multi-objective Optimization of Time-Cost Trade-off" Proceedings of the 8th WSEAS International Conference on Evolutionary Computing, Vancouver, British Columbia, Canada, June 19-21Pages 241-246.

Kelley J. E. (1961), "Critical-Path Planning and Scheduling: Mathematical Basis" Operations Research, May - Jun., Vol. 9, No.3 jstor

Li H., and Love P. (1997) "Using Improved Genetic Algorithms to Facilitate Time-Cost Optimization" Journal of Construction Engineering and Management, Vol. 123 September, No.3. , 0233-0237, ASCE.

Liu L, Burns S. A. and Feng C.-W. (1995) "Construction Time-Cost Trade-Off Analysis Using Lp/Ip Hybrid Method" Journal of Construction Engineering and Management, Vol. 121, No.4, 0446-0454 December,. ASCE,

Ng. Thomas S., and Zhang Y., (2008) "Optimizing Construction Time and Cost

Using Ant Optimization" Journal of Construction Engineering and Management, Vol. 134, No. 9, September 1, pages 721–728, ASCE.

Patterson J. H and Huber W. D. (1974) "A Horizon-Varying, Zero-One Approach to Project Scheduling". Management Science, Vol. 20, No. 6, Application Series, Feb., pp. 990-998 jstor.

Prager W. (1963) "A Structural Method of Computing Project Cost Polygons" Management Science, Vol. 9, No. 3 (Apr., 1963), pp. 394-404 jstor.

Robinson D. R. (1975), "A Dynamic Programming Solution to Cost-Time Tradeoff for CPM" Management Science, Oct., Vol. 22, No. 2 pp. 158-166Published

Siemens N. (1971) "A Simple CPM Time-Cost Tradeoff" Management Science, Vol. 17, No. 6, Application Series Feb., pp. B354-B363 jstor.

Xiong, Y.; and Kuang, Y. (2008), "Applying an Ant Colony Optimization Algorithm-Based Multi-objective Approach for Time–Cost Trade-Off", Journal of Construction Engineering and Management, Vol. 134, No. 2, February 1, Pages 153-156, ASCE.

Zheng, D.; Ng, T.; and Kumaraswamy. M. (2004)," Applying a Genetic Algorithm-Based Multi-objective Approach for Time-Cost Optimization" Journal of Construction Engineering and Management, Vol. 130, No. 2, April 1, pages 168-176, ASCE.

Zheng, D.; Ng, T.; and Kumaraswamy. M. (2005)," Applying Pareto Ranking and Niche Formation to Genetic Algorithm-Based Multi-objective Time–Cost Optimization" Journal of Construction Engineering and Management, Vol. 131, No. 1, January 1, pages 81–91, ASCE.

LIST OF ABBREVIATIONS

ACO:	Ant Colony Optimization
ACS:	Ant Colony System
ACSA:	Ant Colony Solution Algorithm
AOA:	Activity on Arrow
DC:	Direct Cost
FF:	Fitness Function
GA:	Genetic Algorithm
IC:	indirect cost rate.
MA:	Memetic algorithms
MAWA:	Modified Adaptive Weight Approach
PF:	Pareto Front
PSO:	Particle Swarm optimization
SFL:	Shuffled Frog Leaping
TCO:	Time-Cost Optimization
TCT:	Time-Cost Trade-off

$c_{ij}^{(k)}$:	Execution of direct cost of option j in activity i selected by ant (k)
J:	Variable generated using random proportional rule
K:	Random ant
L:	Total number of project activities
m:	The number of ants in each iteration
n:	No. of options in each activity the user input
P:	All the Paths in the network
P_p:	Activity Sequence of certain path
q:	Random variable between [0< q <1]
q0:	Factor of the pseudorandom proportional action choice rule [0<q0<1]
R:	Positive random number between [0< R <1]
S:	Total number iterations
T:	No. of iterations in the model (optional)
$t_{ij}^{(k)}$:	Execution time of option j in activity i selected by ant (k)
V_t, V_c, V:	Value of Time, Cost, and Project, respectively
W_t, W_c:	Adaptive weight for the objective of time and total cost
$x_{ij}^{(k)}$:	Index used to verify which option the ant selected to execute
Z_c (k):	Objective function of cost for ant (k) in the current iteration
Z_t (k):	Objective function of time for ant (k) in the current iteration
Z_t^{max}, Z_c^{max}:	Maximal value for objective of time and total cost in current iteration
Z_t^{min}, Z_c^{min}	Minimal value for objective of time and total cost in current iteration
α:	Coefficient represents the importance of the pheromone value (τ)
β:	Coefficient represents the importance of the heuristic value (η)
Δτ:	Best ant pheromone updating value
ε:	Local pheromone evaporation rate [0< ε <1]
η_{ij}:	Heuristic value of activity i option j
ρ:	Global pheromone evaporation rate [0<ρ<1]

τ0: Initial pheromone value

τ_{ij}: Pheromone value of activity i option j

TSP: Traveler salesman problem

FIGURES

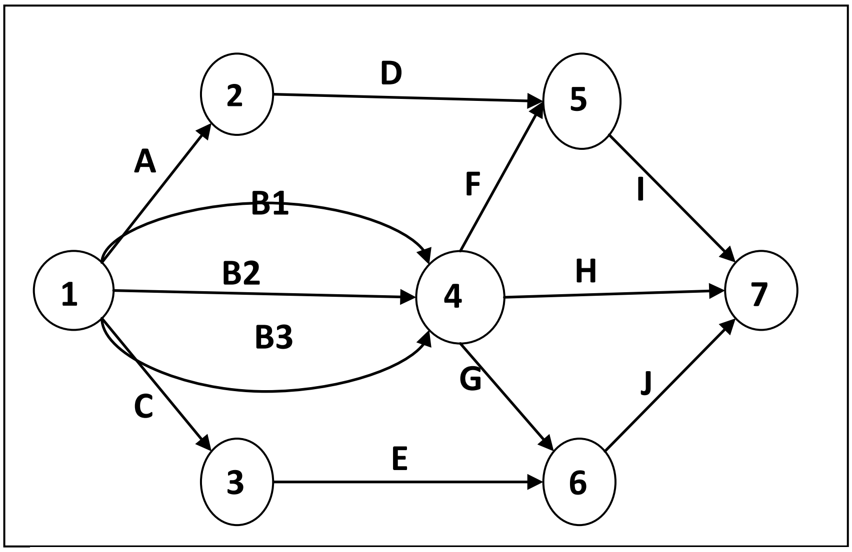

Fig. (1): TCO Project (AOA Network)**.**

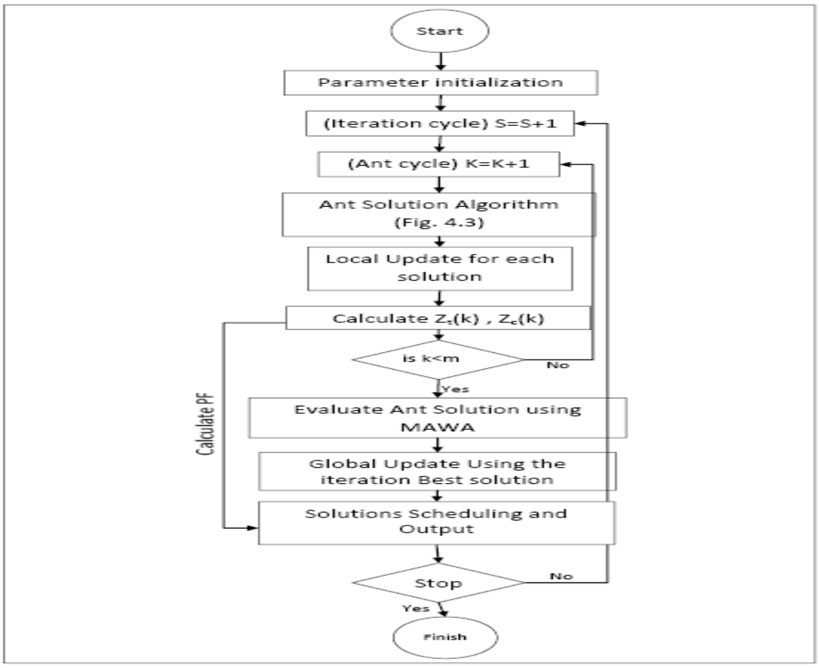

Fig (2): Developed Model Flowchart.

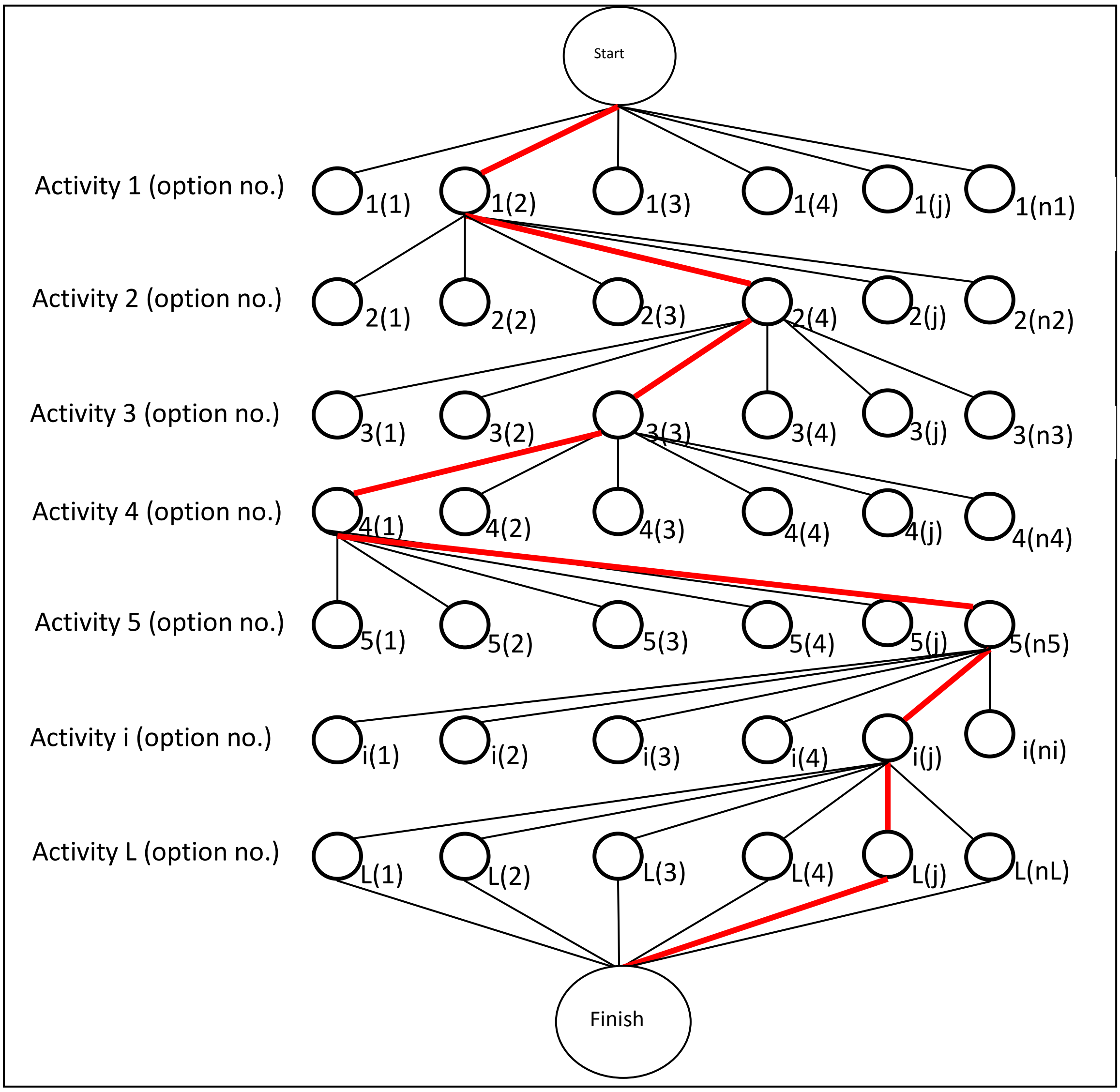

Fig. (3): Graphical Representation of Ants' traveling through construction projects Activities (Researcher).

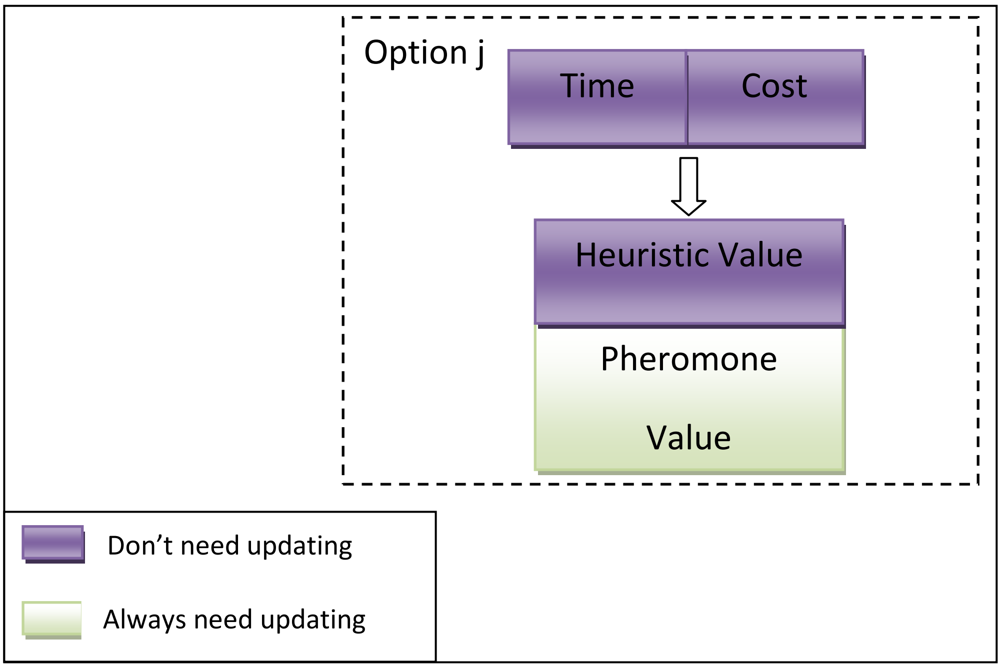

Fig. (4): Option $_j$ Variables (Researcher).

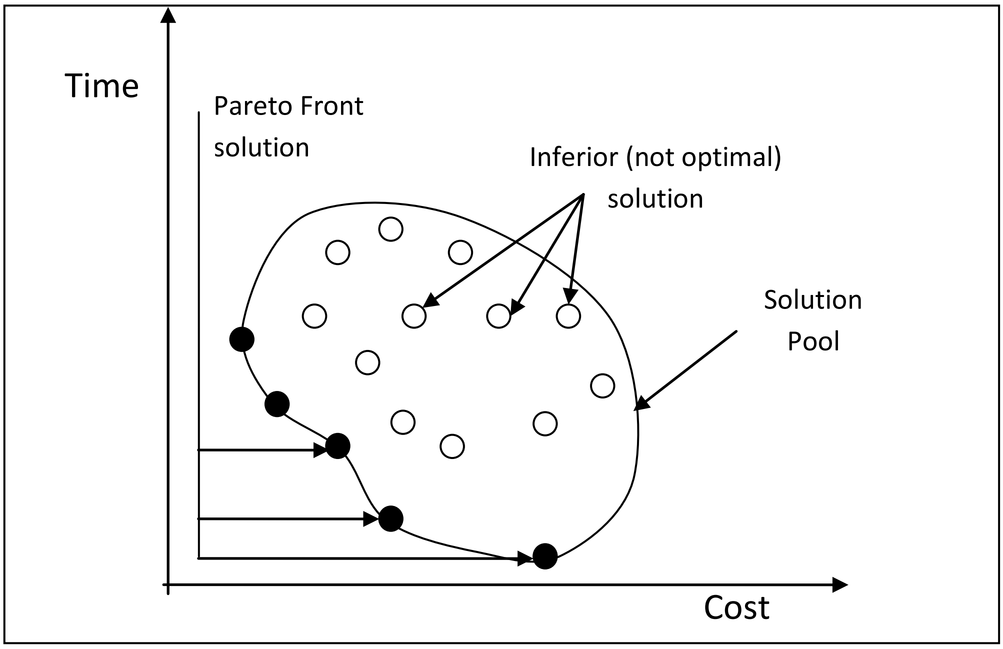

Fig.(5): Pareto front and solution pool.

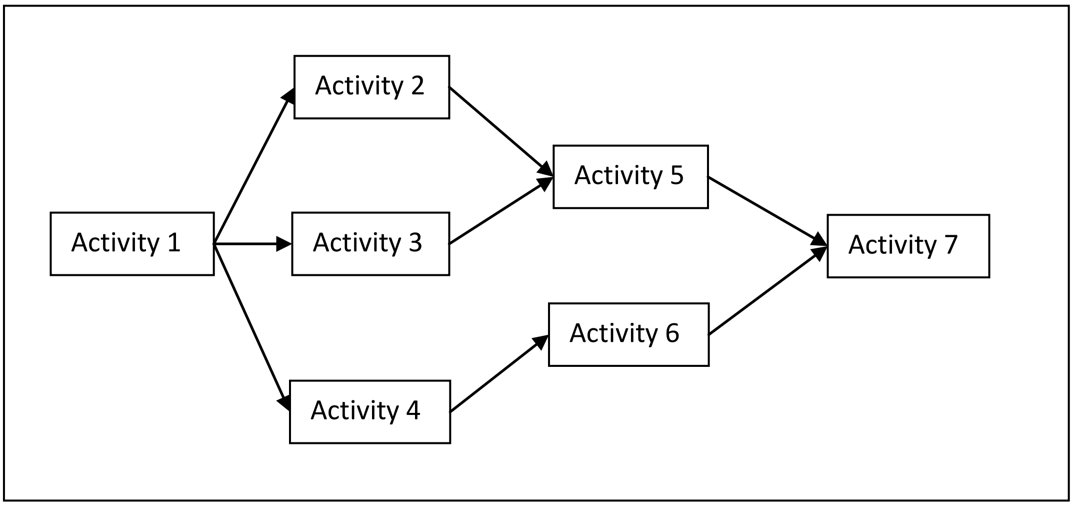

Fig. (6): Network representation of the 7 activity reference problem.

Table (1) Time-Cost Models (Researcher)

Time-Cost Trade-off			**Time-Cost Optimization**
Heuristic Models	**Mathematical Models**	**Evolutionary-Based Optimization Algorithm Models**	**Evolutionary-Based Optimization Algorithm Models**
Structural Interpretation **(Prager, 1963)**	Linear Programming **(Kelly, 1961)**	Genetic Algorithms **(Feng et. al. 1997)**	Genetic Algorithms **(Zheng et. al. 2004)**
	Zero-One Programming **(Patterson, 1974)**	Genetic Algorithms **(Feng et. al. 2000)**	Genetic Algorithms **(Zheng et. al. 2005)**
Cost Slope **(Siemens, 1971)**	Dynamic Programming **(Robinson, 1975)**	Genetic Algorithms **(Li et. al. 1997)**	Genetic Algorithms **(Kasaeian et. al. 2005)**
Effective Cost Slope **(Goyal, 1975)**	Linear Programming **(Handrickson, 1989)**	Genetic Algorithms, Particle Swarm, Ant Colony, Shuffled Frog Leaping and Mimetic Algorithms **(Elbaltagi, et. al 2005)**	Ant Colony Optimization **(Xiong, et. al. 2008)**
Cost Slope **(Al-Samaraai, 2005)**	Linear/Integer Programming **(Lui, et. al. 1995)**		Ant Colony Optimization **(Ng, et. al. 2008)**
	Linear Programming **(Chassiakos et. al. 2005)**		Ant Colony Optimization **(Afshar, et. al. 2008)**

Table (2): Model recommended parameters (Researcher)

Parameter	*Value*	*Parameter*	*Value*
m	*10*	*ρ*	*0.002*
α	*1*	*ε*	*0.1*
β	*2*	*q0*	*0.4*

Table (3): Reference Problem (7 Activity) ***(Feng, et. al., 1997)***

Activity	Precedence	Option No.1		Option No.2		Option No.3		Option No.4		Option No.5	
		Time (Days)	Cost ($)	Time (Days)	Cost ($)	Time (Days)	Cost ($)	Time (Days)	Cost ($)	Time (Days)	Cost ($)
1-Site Preparation	-	14	23,000	20	18,000	24	12,000	-	-	-	-
2-Forms and rebar	1	15	3,000	18	2,400	20	1,800	23	1,500	25	1,000
3-Excavation	1	15	4,500	22	4,000	33	3,200	-	-	-	-
4-Precast concrete girder	1	12	45,000	16	35,000	20	30,000	-	-	-	-
5-Pour foundation and piers	2.3	22	20,000	24	17,500	28	15,000	30	10,000	-	-
6-Deliver PC concrete	4	14	40,000	18	32,000	24	18,000	-	-	-	-
7-Erect girders	5.6	9	30,000	15	24,000	18	22,000	-	-	-	-

Table (4): Option selection and solution generated for the 7 activity reference problem (indirect cost is 1500$/day)

Solution	Project Time (Days)	Project Total Cost ($)	Options selected by the model to execute the corresponding activity						
			1	2	3	4	5	6	7
1	60	233500	1	1	1	1	1	3	1
2	62	233000	1	1	1	3	2	2	1
3	63	225500	1	1	1	2	2	3	1
4	67	224000	1	1	1	3	3	3	1
5	68	220500	1	1	1	3	4	3	1

Table (5): Solution comparison of (7activity) reference problem

Zheng Model 2004		Xiong Model		Developed Model	
Time (Days)	Total cost ($)	Time (Days)	Total cost ($)	Time (Days)	Total cost ($)
66	236500	60	233500	60	233500
73	251500	62	233000	62	233000
84	251000	63	225500	63	225500
-	-	67	224000	67	224000
-	-	68	220500	68	220500