Automation in Construction 20 (2011) 491–505

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Review

Metaheuristics for project and construction management – A state-of-the-art review

T. Warren Liao [a,*], P.J. Egbelu [a], B.R. Sarker [a], S.S. Leu [b]

[a] *Department of Construction Management and Industrial Engineering, Louisiana State University, Baton Rouge, LA 70803, United States*
[b] *Department of Construction Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan*

ARTICLE INFO

Article history:
Accepted 11 December 2010
Available online 14 January 2011

Keywords:
Project management
Construction management
Optimization
Metaheuristic
Ant colony optimization
Evolutionary algorithm
Genetic algorithm
Particle swarm optimization

ABSTRACT

This paper carries out a detailed review of previous studies that employed metaheuristics to address problems/issues encountered in the life time of a construction or engineering project. The review is organized around critical problems/issues including engineering, cost estimation, planning, scheduling, and monitoring and control of project operations, with the objectives to optimize cost and time through the efficient uses of constrained or unconstrained resources. Metaheuristics are emphasized because they are especially suitable when approximate solutions are good enough in the case that the global optimum is either unknown or computationally too expensive to obtain it; both are characteristics of a typical large scale project. Based on the review, we offer our view of this research area and their potential for practical use. An attempt is also made to identify project management related topics where further advancement needs to be made.

Contents

* Corresponding author. Tel.: +1 225 578 5365; fax: +1 225 579 5109.
E-mail address: ieliao@lsu.edu (T.W. Liao).

0926-5805/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.autcon.2010.12.006

1. Introduction

Regardless of its type (linear or nonlinear), scale (small or large), and duration (short or long), each project generally has three objectives to meet, i.e., performance (with respect to the specification), time, and cost. Since these three objectives are conflicting in nature, the primary task of a project manager is to manage the tradeoffs between them. Each project has its life cycle just like any organic entity or manufactured product. Typically, a project life cycle starts with (1) the initialization of a need or an idea, then followed by (2) the engineering phase that can be further divided into preliminary design and detailed design, (3) the planning phase that includes elements such as selecting construction methods, subcontracting, resource requirement planning, determining project evaluation methods, and performing risk analysis, (4) the implementation phase that involves budgeting, scheduling, and resource allocation, (5) the monitoring and control phase that include information gathering, tracking, and taking corrective actions, and finally (6) the evaluation, auditing and termination phase. Project management (PM) shall begin with the initial concept for the project and continue on throughout the project life cycle to ensure that the project objectives are met in an optimal manner.

Understandably there are complicated engineering and management problems/issues related to a project life cycle. This is especially true for a large scale project that takes a long time to complete. Construction and engineering researchers have been at the forefront in developing various techniques and tools to support construction and engineering decision making. Among all topics mentioned above, project scheduling has been most studied. Techniques been developed to schedule *linear projects* include the line of balance technique, linear scheduling method, vertical production method, and so on. *Non-linear projects* are usually scheduled based on networking techniques. [76] presented and discussed the major contributions of operational research (OR) to project management (PM). His review emphasizes on the use of traditional OR techniques to project scheduling and project monitoring. In a separate review, [77] was particularly concerned with the contribution that OR has made to real, practical problems by building mathematical models and the results of their analysis to inform decision making. He pointed out the divergence in the OR and PM camps in 80s and 90s and urged OR researchers on in developing new theories and models that are more useful and better explain the behavior of modern, complex, and dynamic real world projects.

Heuristic rules and exact solution methods dominate earlier operational research to support construction and engineering decision making. However they are deficient to deal with large scale problems. Various metaheuristic algorithms grounded on biological and animal behavior have become popular lately. Metaheuristics are general purpose high level search frameworks that can be applied to any optimization problem with the use of appropriate local problem-dependent solution procedures. Examples of metaheuristics include simulated annealing (SA), ant colony optimization (ACO), evolutionary algorithm (EA), genetic algorithm (GA), particle swarm optimization (PSO), and shuffled frog-leaping (SFL). Theoretically, each metaheuristic can be applied to optimize each project life related topic, though the effectiveness might vary. The reason for focusing on metaheuristics is their superiority in handling highly nonlinear, multi-modal, constrained, discontinuous, and non-differentiable optimization models often encountered in project management.

This review paper focuses on the use of various metaheuristics to help determine the optimal operation mode in carrying out every phase of a project life cycle, as depicted in Fig. 1. Many relevant and practical project life cycle problems/issues will be reviewed in this paper, except the category of resource constrained project scheduling that is intentionally left out to be covered in another review due to the high numbers of papers reporting on the topic. Due to space limitation, for each paper reviewed the emphasis will only be on the problem being solved and how metaheuristic is used. Interested readers should refer to the original paper for more details. Differing from previous reviews, our contributions mainly lie in two areas: (1) the coverage of wider topics with emphasis on metaheuristic algorithms - new breeds of OR techniques with high potential, and (2) the offering of our personal view on past, current, and future research on this subject.

The remaining of the paper is organized as follows. Section 2 provides an overview of uses of metaheuristics in project and construction management. Our review is organized around subjects encountered in a typical project cycle starting from engineering design in Section 3, cost estimation in Section 4, planning in Section 5, scheduling in Section 6, and monitoring and control in Section 7. A Discussion Section follows with topics for future research highlighted. Finally, the paper is concluded.

2. Overview

Before presenting the details, an overview is given first. Table 1 summarizes various metaheuristics that have been applied to study various project life cycle topics, specifically to determine the optimal solution(s) for the problem of concern to meet some specific objective. The numbers in the table indicate the numbers of papers reviewed. Table 1 clearly indicates that evolutionary algorithms/genetic algorithms (EAs/GAs) is the most popular metaheuristic among all. In our opinion, their popularity can be attributed to, at least, the following four reasons: (1) EAs/GAs are the earliest population-based metaheuristics developed; (2) EAs/GAs are capable of dealing with both continuous function and combinatorial optimization problems; (3) EAs/GAs generally find good solutions; and (4) there are several non-commercial EA/GA tools available for free download. Note that

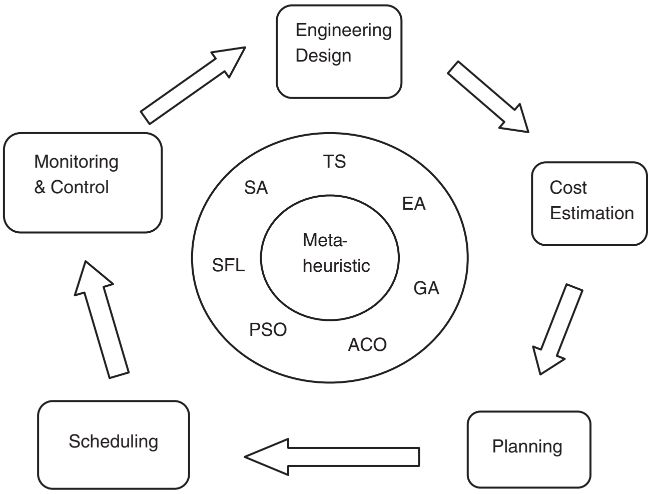

Fig. 1. Optimization of project life cycle related problems with metaheuristics.

Table 1
Summary of usages of metaheuristics in construction management.

Topic	SA	TS	ACO	EA/GA	PSO	HS	SFL	Hybrid	More than 1
3. Engineering Design				5					
4. Cost Estimation				3					
5. Planning									
5.1. Site preparation	1								
5.2. Site/floor layout		1	1	9	1			1[3]	
5.3. Site routing				1					
5.4. 1D stocking cutting				2					
5.5. Supply chain/logistics			1	1					
5.6. Equipment selection				3(1)					
6. Scheduling									
6.1. Resource unconstrained	2		3	2					1[2]
6.2. Resource constrained									
6.2.1. Time-cost tradeoff			3(3)	12(8)		1(1)	1		
6.2.2. Resource allocation			2(1)	8(3)	2(1)				
6.2.3. Resource leveling	1			3					
6.2.4. Integrated models				5(2)					
7. Monitoring and Control				1					

Note: (1) Numbers in parenthesis indicate number of papers that consider multiple objectives.
(2) This study employs two metaheuristics, i.e., SA and TS.
(3) This is a ant-GA hybrid.

each metaheuristic is continuously improved upon since it was originally developed. Therefore, there are numerous versions of the same metaheuristic that follow very much the same principle but differ slightly from each other in some details. New metaheuristics are also being developed. But the day that a new metaheuristic would overtake the popularity of EAs/GAs in project and construction management is nowhere in sight.

Also clear in Table 1 is that among all the project life cycle related problems that employed metaheuristic, site/floor layout, time-cost tradeoff analysis, and resource allocation are most frequently studied. Nevertheless, the numbers of studies on each of these three topics are still far short of those studies on resource constrained project scheduling, which is intentionally left out to be covered in another review due to the high numbers of papers reporting on the topic.

For some problems, it might be more appropriate to consider more than one objective. Most likely those objectives are conflicting in nature. Time-cost tradeoff is a case in point. [41] presented a review of current research efforts on the field of construction multi-objective optimization. One simple approach to deal with multi-objective problem is to convert multiple objectives into one single objective by weighting them. However, correctly assigning the weights is often a challenge. The alternative approach that is gaining popularity is to generate the entire set of non-dominated solutions. The numbers in parentheses in Table 1 indicate the number of papers that consider multiple objectives attempting to find the set of non-dominated solutions.

Depending upon the topic, the decision variables might be continuous, discrete, or mixture of both. This often determines whether the optimization problem is continuous or combinatorial. The site layout problem to assign departments/facility/offices to predetermined locations is a typical combinatorial optimization problem. However, if the locations are not predetermined, then the site layout problem is not combinatorial because it needs to determine the coordinates of facility in real values instead. The project scheduling problem involved in determining activity starting times is another typical continuous optimization problem. The stock cutting problem is one example involving both continuous variables (different cut lengths) and discrete variables (numbers of different cut lengths).

The majority of optimization problems reviewed in this paper is constrained. Solutions that violate constraints are infeasible and different ways to handle infeasible solutions have been used. Rejecting and randomly regenerating a replacement is one simple strategy. However, this approach might not work on a highly constrained problem in which the numbers of feasible solutions are relatively few. For an optimum located close to infeasible solutions, it might be easier to find it by modifying an infeasible one than a feasible one using some repair strategy. In such cases, the rejection strategy is also at a disadvantage. The penalty method is yet another approach, which retains infeasible solutions after penalizing them. Determining the appropriate penalty factors, however, is non-trivial and problem dependent. The parameterless penalty method was developed by [19] to avoid this very need to specifying penalty factors. Surprisingly, this excellent strategy has not been employed in any work reviewed in this paper. It appears that the research community on the subjects reviewed in this paper is lagging behind in this regard.

Regardless of the metaheuristic chosen for a study, there are common issues that need to be addressed as briefly described below.

(1) Solution representation. In deciding how to represent a solution, the number and types of decision variables, and the resolution of solution values must all be considered.
(2) Generation of initial solution(s). Depending upon the metaheuristic, one or more solutions must be initialized to start the search process. Random generation is commonly used. However, a specially designed procedure might be more appropriate to start with a solution close to the optimum or if randomly generated solution might be infeasible.
(3) Evaluation function. Each solution must be evaluated for its goodness. The evaluation function is problem-dependent and must be defined in consideration of the objective(s) to be met, and in some cases constraint violations, too.
(4) Generation of neighborhood solutions. Neighborhood solutions are often obtained by perturbing (selected) current solution(s) by some generation methods. Some neighborhood solution generation methods are generic while others are unique. For example, pairwise exchange and insertion operators are generic and used often in permutation-based solution representation. On the contrary, the way that ants construct solutions by communicating through pheromone is unique only to the ant colony optimization algorithm.
(5) Handling constraint violations. Violations of equality or inequality constraints and violations of domain values are often distinguished and handled differently. Some commonly used strategies have been mentioned earlier.
(6) Stopping criteria. One or more criteria must be specified to stop the algorithm. In comparing different metaheuristics, consistent and computing platform independent stopping criteria such as maximal numbers of evaluations should be applied to

each metaheuristic for fair comparison. However, picking the appropriate value is problem dependent and requires some trial and error so that the algorithm does not stop either too early or too late.

3. Engineering design

Engineering design could be a standalone project or part of a larger project. Numerous engineering design problems of widely different natures exist and only a selected few have been reported in the open literature, as presented below.

[61] developed a GA-based methodology for selecting/assembling the optimal robotic system to strip paint off bridges of complex and varied geometries. The criteria used to determine the optimal parameters of a robot are: (i) collision avoidance, (ii) singularity avoidance, (iii) percentage of coverage, (iv) dexterity, and (v) productivity. Due to computing limitations, optimizing a robot's kinematic parameters is decomposed into a two-stage problem. The first stage optimizes the robot's major configuration based on a fitness function, which comprises of the criteria of collision and arm singularity avoidance, and percentage of workspace coverage. The second stage optimizes the wrist type, the movement sectors of all six joints and their optimal acceleration and velocity parameters based on another fitness function that considers arm and wrist singularity avoidance, dexterity and productivity. Subsequently, [59] developed a GA-based tool, which can be used for rapid and easy testing of different geometries and different robot's configurations. They tested the tool's effectiveness on a selected geometry and for a choice of two most probable robot's configurations: RRR (all three joints are of the revolute type) and RRP (the third joint is of the prismatic type). The scale of the computations greatly restricted the efficiency in obtaining results.

To eliminate the need to specify a priori the penalty parameter in GA search for pipe network design, [3] proposed a method that uses the ratio of the best feasible and infeasible designs at each generation to adapt the value of the penalty parameter. Their method is based on the assumption that the optimal solution of a pipe network design problem lies somewhere on, or near, the boundary of the feasible region. The GA employed is binary coded and based on tournament selection, one-point crossover, and bit mutation operations. The fitness function is the inverse of total cost. The rehabilitation of the New York City water supply network with 21 pipes, 20 demand nodes, and one reservoir was tested.

Foundations are crucial to the safety and serviceability of supported structures such as bridges and buildings. [8] developed a hybrid GA that incorporates a local search operator into a binary-coded GA for optimizing the size and topology of pile foundations, subject to various design constraints. The objective is to minimize the material volume taking the configuration, number, and cross-sectional dimensions of the piles as well as the thickness of the pile cap as design variables. Infeasible solutions are penalized by using a parameterized penalty function. Determining penalty factors truly reflecting the relative importance of design constraints is nontrivial. In addition, binary coding would limit the size of problem that could be handled.

In a new product design and development project, it is critical to perform project activities in an appropriate sequence. Otherwise, essential information will not be available when it is needed, and activities that depend on it will have to proceed with assumptions instead. This often precipitates a cascade of rework once the real information is finally available. To address this issue, [60] applied GAs to a digraph-based information flow model represented as a design structure matrix (DSM) to find an optimal sequence of engineering design activities. The cells on the diagonal of a DSM represent activities to be performed and off-diagonal cells represent information flows or dependencies among the activities. Cells below the diagonal represent forward dependencies and those above backward dependencies. They formulated design process sequencing or DSM sequencing as a quadratic assignment problem. Four GA algorithms including a simple GA (SGA), a hybrid SGA (SGA + 2-opt local search), a competent GA called ordered messy GA (OmeGA), and a hybrid competent GA, were tested. They demonstrated that the difficulty of sequencing DSMs depends upon DSM characteristics such as size, density, and objective function.

4. Cost estimation

The accuracy of cost estimation is critical to the success of any project. The current practice generally relies on experts' experience. Computer assistance is quite feasible with the use of a wide range of statistical or machine learning methods such as case based reasoning if historical data is available. This section reviews only those that employ metaheuristic.

Using the cost information of 498 projects of residential buildings built by general contractors between 1997 and 2000, [43] selected eight input variables and applied three hybrid models of neural networks (NNs) and genetic algorithms to make preliminary cost estimates of residential buildings. The three models applied differ in determining the NN parameters by trial-and-error in Model I, determining the NN parameters by GA in Model II; and training the weights of NNs using genetic algorithms in Model III. The data was divided into training data (80%), cross-validation data (10%), and test data (10%). The training process is stopped when the mean square error in cross-validation does not improve for 100 iterations. Model II was found most effective among all.

To improve the conceptual cost estimation accuracy, [13] proposed the Evolutionary Fuzzy Neural Inference Model (EFNIM) with the combined use of genetic algorithms, fuzzy logic (FL), and neural networks. Within EFNIM GAs are used to optimize NN topology/weightings and to define FL membership functions. Two estimators were developed to produce overall conceptual cost and categorical cost. The inputs to the overall estimator are six quantitative factors and four qualitative ones. The critical factors impacting each category as identified serve as inputs to each categorical estimator. 23 training and 5 testing cases were used, which seem to be few relative to the model size.

To estimate cost of a software project, [51] proposed the use of simple genetic algorithm for project selection based on analogy based estimation (PSABE) and for the simultaneous optimization of feature weights and project selection based on analogy based estimation (FWPSABE). Each chromosome consists of two parts. The first part codes feature weights with the length of $14 \times n$, where n is the number of features. The second part of the codes is for project selection: "1" for selected and "0" for not selected. The accuracies of cost estimating methods are measured by mean magnitude of relative error (MMRE), median magnitude of relative error (MdMRE), and percentage of predictions that fall within 25% of the actual cost (PRED(0.25)).

5. Planning

The importance of preproject planning and its potential impact on project success has been recognized by industrial practitioners. Based on their analysis of more than 200 projects, [31] reported positive relationship between thorough preproject planning and enhanced project performance. After analyzing data collected from 202 project managers, [90] concluded that the level of risks at the beginning of projects has no effect on their final success and improving the project plan is a more effective managerial tool in dealing with high risk projects. This section reviews studies on planning activities such as site preparation, site/floor layout, routings of on-site vehicles and people, material preparation, and supply chain and logistics.

5.1. Site preparation

A large engineering and construction project often requires a modification of the terrain where the construction site is to be located. Terrain modifications involve moving large amounts of earth, either from one area to another area of a construction site, or in some cases from external locations to a construction site or from a construction site to other external locations. Excavating earth from cut locations and hauling it for deposit into fill locations requires earthmoving vehicles that are usually expensive to operate and maintain. To reduce the cost of such operations, planners must develop a strategy that minimizes the total distance traveled by earthmoving vehicles between cut and fill locations. This problem is known as the shortest route cut and fill problem (SRCFP) in the research literature. [37] presented results obtained by a local search algorithm, i.e., simulated annealing, applied to the SRCFP. Assuming the number of unit cuts and the number of unit fits are equal ($=m$), the size of the solution space is $m!(m-1)!/2$. Given a set of cut locations, c_i, and a set of fill locations, f_i, $i=1, \ldots, m$, the problem is to find a route $\mathrm{R}=(\mathrm{c}_1,\mathrm{f}_1,\ldots,\mathrm{c}_\mathrm{m}, \mathrm{f}_\mathrm{m})$ so that objective function *g(R)* is minimized with the assumption that the vehicle must be returned to the first cut position.

$$g(R) = \sum_{i=1}^{m-1}\left[d(c_i,f_i) + d\left(f_i,c_{i+1}\right)\right] + d(c_m,f_m) + d(f_m,c_1) \qquad (1)$$

where $d(c_i, f_i)$ denotes the Euclidean distance between cut location c_i and fill location f_i. To determine if the solution found by the simulated annealing algorithm was indeed optimal, CPLEX 6.6 was used to solve the integer programming model formulated.

5.2. Site layout and floor layout

Each construction project needs adequate amount of spaces for temporary facilities such as warehouses, job offices, various workshops, lay down area for materials, and batch plants in order to execute the construction activities in a safe and efficient manner. Construction site-level facilities layout is thus an important activity in site planning. Planning construction site spaces to allow for safe and efficient working conditions is a complex and multi-disciplinary task as it involves accounting for a wide range of scenarios. Site layout planning can be distinguished as a static problem or a dynamic problem, depending upon whether non-changing or changing site facilities and site space are considered in different phases of the project. Some studies assume that locations are predetermined while others do not.

5.2.1. Site layout with predetermined locations

[49] described a construction site-level facility layout problem as allocating a set of predetermined facilities into a set of predetermined locations, while satisfying layout constraints and requirements. They employed a genetic algorithm to solve the problem by assuming that the predetermined locations are in rectangular shape and are large enough to accommodating the largest facility. If there are more locations than facilities, dummy facilities are added to make both numbers equal. To ensure that the final layout will not be affected by dummy facilities, distance and frequency associated with a dummy facility are set to zero. Each chromosome represents a string of locations where the facilities are placed. Chromosomes are selected with probabilities proportional to their fitness values according to the roulette wheel principle. To generate new offspring, the modified edge recombination operator and the symmetric genes exchange operator are used. The objective is to minimize the total traveling distance of site personnel between facilities. Subsequently, [50] applied a genetic algorithm similar to that used in [49] to solve site-level unequal-area layout problem for the case that predetermined locations have two different sizes of areas and some of the predetermined locations are only able to accommodate selected facilities. Both studies mentioned above did not address the effects of GA parameters.

The production of standardized modular precast units within a precast yard involves repeated movements of resources between the essential facilities necessary for production. Following the same assumptions made in [14], [49] described the use of a GA package, the Evolver, to solve a pre-cast yard layout problem. The order chromosome is used to encode facilities assigned to locations. Evolver employs a steady-state approach, meaning that only one chromosome is replaced at a time, rather than an entire generation being replaced as normally done. The genetic operators used are the order crossover operator and the swap mutation operator. Parents are chosen with a rank-based mechanism. The fitness of a chromosome is assessed by the total cost per day for transporting all resources necessary to achieve the anticipated output.

[75] developed a site layout genetic algorithm to determine the optimal tower crane and supply locations in the structural concrete-frame construction stage of public housing projects. Each location is selected from a corresponding predetermined candidate set of 3D coordinates. Each possible solution is thus encoded with a number of chromosomes equivalent to the number of locations to be determined. The demand points are assumed fixed. The crane travel time is computed according to the model developed by [86]. The objective is to minimize the total cost.

[85] presented a particle swarm optimization (PSO)-based methodology to solve the construction site unequal-area layout problem, formulated as a quadratic assignment problem (QAP). The objective is to minimize the total costs including the communication cost and the setup cost. Each particle encodes the priority-based representation of a layout solution, which is then transformed to a feasible layout plan in consideration of geometric constraints between facilities and locations. In addition, a modified solution space boundary handling approach is used to ensure that the updated particle position is inside the solution space. The velocity of particles is limited to the predetermined boundary values to prevent explosion. An 11 facilities 11 locations problem was used for testing and the results showed that the proposed PSO outperformed the GA algorithm of [50].

[52] proposed a multi-searching tabu search procedure based on efficient diversification and intensification strategies to effectively improve the various arrangements in the facility layout problem. The diversification strategy relies on three moves: neighboring move, backward move, and left-right move. The intensification strategy uses random exchange to search local area. It was shown that better results were obtained by the proposed procedure compared with the GA of [50] and the GA of [14].

Using a hypothetical construction site involving the assignment of 6 facilities to 24 equal sized locations, [44] showed that better solution could be obtained if the initial population of solutions for a genetic algorithm was generated by a max-min ant system, instead of randomly generated. To arrange the precast facilities in the construction site, [78] developed a genetic algorithm and a mixed integer programming (MIP) to generate optimal layout solutions. An example of 11 facilities and 11 locations was used to illustrate that MIP produced a better solution than GA. To solve a dynamic site layout planning problem, [64] proposed a new method that uses the continuous dynamic search scheme to guide the max-min ant system to find the optimal layout. The objective is to minimize the weighted sum of safety concern and construction cost.

5.2.2. Site layout without predetermined locations

[35] presented a GA for site layout planning, which makes use of a grid-based representation of site and facility. Using the macro language of Microsoft Excel, the procedure was coded as part of a complete site layout planning system called EvoSITE. The area of a grid unit is computed as the greatest common divisor of all facility areas. The location reference of each grid cell is calculated as (row position − 1) × total columns +

column position. Three methods for placing a facility are considered: horizontal, vertical, and rectangular. The closeness relationships between facilities are qualitatively determined by giving subjective proximity weight, in exponential scale. The GA procedure is used to search for a layout that minimizes the total travel distance. Each chromosome encodes the location references of facilities.

[89] investigated the applicability of a genetic approach for solving the construction site layout problem with locations not predetermined. The layout problem considered is characterized by affinity weights used to model transportation costs between facilities and by two-dimensional geometric constraints that limit their relative positions on site. The shape and size of facilities are considered fixed. Three types of constraints considered are distance, orientation, and non-overlap. Each chromosome is coded as a vector with length equal to the number of facilities that exist on site. The objective is to find a feasible arrangement for all layout objects within the site space that minimizes the sum of the weighted distances separating the layout objects. Their approach uses nine genetic operators to vary the positions of objects around the site. The genetic operators used call for the Find-a-Set-of-Possible-Positions function that generates a set of feasible positions for an object in the neighborhood of a randomly selected location on site. In most cases where the total objects-to-site-area ratio did not exceed 60%, the algorithm returned close to optimal solutions in reasonable time (less than 2 minutes) after 250 generations. In problems with higher total-objects-to-site-area ratio the algorithm failed to find “good” and in some cases even feasible solutions.

[58] described the general site layout problem with locations not predetermined from both theoretical and practical points of view. Assuming rectangular site area and facility shapes, and facility sides paralleling with the coordinate axes, they proposed a genetic algorithm to determine the positions of facilities on site to minimize a fitness function, which is consisted of four cost components: material transportation cost, setup cost, removal cost, and personnel movement cost. Each chromosome encodes an array of real coordinate values of facilities. Specific genetic operators developed for the study include arithmetic combination crossover operator, coordinate swap crossover operator, random offset mutation operator, and swap mutation operator. The proposed algorithm was found to be sensitive to the relative costs assigned to setup and transport.

5.2.3. Floor layout

For multiple-floor building where multiple construction-works occur simultaneously in close proximity, especially in a congested and densely populated city where spaces for constructions are limited, floor-level construction material layout is needed to provide project managers with a valuable technique for developing efficient sequences of works that optimally defines how to efficiently position the construction materials and thus minimizes the travel distance between work spots and special construction materials in a multiple-floor space.

[39] applied a genetic algorithm to the floor-level construction material layout in multiple-floor buildings. While sizes and shapes of construction material may vary widely in reality, it is assumed that these materials are in square or rectangular form. Each floor is divided into grids with size the same as the smallest construction material. The objective is to minimize the total number of construction material handling movements, in which the priority weight of a construction material relative to a construction work must be determined. Specially designed crossover and mutation operators are used to evolve solutions. They compared the total distances of construction material handling between an actual case and the output of genetic algorithm and reported that inefficiencies in positioning construction materials at the floor-level could led to an increase in the construction material handling distance by 14%. However, determining priority weights is tedious and needs to be simplified to improve their method.

5.3. Routing of on-site vehicles and people

The issue of planning vehicles and people's movement within a construction site has increasingly become a significant area of interest to site planners. Currently, site planners rely mainly on their experience and common sense when considering the allocations of paths for logistics, vehicles, and site operatives. Advances in information technology could be used to help site planners in their decision-making process by allowing them to express various scenarios that take into consideration several interrelated factors for path finding.

[73] described a framework for supporting path planning analysis in construction sites based on multi-criteria evaluation of transportation, safety, and visibility measures. Their modeling strategy has the following four steps: (i) site layout representation; (ii) path evaluation criteria; (iii) multi-criteria path cost evaluation; and (iv) path search optimization. The site layout is divided into cells in a two-dimensional grid, which is then used to develop a directed graph whose vertices and edges are the location of the discrete nodes and the associated numerical node values. A composite objective value of a grid node on the site layout is obtained by summing the normalized values of distance, hazard, and visibility by weighted sum. The distance between two nodes is measured by Euclidean. On site objects are hazard and different hazard levels around an object are distinguished using three different hazard distributions including constant hazard, linearly decreasing hazard, and nonlinearly decreasing hazard. The average visibility on a grid is computed as the ratio between the number of isovist lines and the number of rays. Three path search optimization techniques used are Dijkstra, A*, and genetic search. In the case of Dijkstra and A*, the path cost evaluation function is the summation of the composite objective values of the grid nodes on the path. In the case of GA, the fitness function consists of two parts, the objective of minimizing the overall path cost and the penalty for grid cells that lie on on-site objects and coincide with the path. The GA's performance limitations lie mainly in obtaining less accurate solutions and the time consuming fine-tuning process to guide the search.

5.4. One dimensional stock cutting

Construction waste accounts for a significant portion of the municipal waste stream of the United States. One-dimensional stocks are one of the major contributors to construction waste. Although part of such waste such as steel waste is recyclable, reduction in waste can maximize the stock material usage and thereby increase the profit potential. Major causes of steel waste generation are mainly due to production errors, calculation errors, on-site waste, and cutting losses. Cutting losses occur during cutting the standard steel (rebars and sections) lengths to meet the specified lengths. Because of the large number of possible cutting patterns, cutting stock problem is a combinatorial problem.

[72] presented a genetic algorithm model for solving the one-dimensional cutting stock problem. The chromosomes are integer coded in a series of pairs with length equaling twice the number of different lengths demanded. The first number in each pair denotes the pattern number to be used, and the second number gives the number of times this pattern would be used in the solution. For a pattern to be feasible, the total length of the pattern (utilized length) must be less than or equal to the standard length of the stock. To reduce the search space, a pattern must also be efficient in the sense that the associated unused (wasted) length must be less than the smallest demanded unit length. To further decrease the search space, the maximum allowable repetition of each pattern is limited to a certain level. The population size is set 2.5 times the chromosome length. The population is initialized randomly based on a special scheme. The fitness of each chromosome is evaluated with a function, which is designed to minimize waste and to penalize unmet demand. To generate new offspring, two-point crossover, a mutation operator, and the roulette-

wheel selection integrated with elitist strategy and linear normalization of fitness values are used. The evolution continues until the maximum number of generations is reached. The results of three case studies showed that the amount of waste experienced in the actual cutting of the steel sections from the workshops was reduced, on average, by 28.92% using the proposed method. Later, [70] compared GA with linear programming and integer programming approaches for essentially the same problem, though different cases were solved.

5.5. Supply chain and logistics

One key operation within the supply chain for the construction industry is the Ready Mixed Concrete (RMC) delivery process. Unlike materials such as pipes that can be manufactured in advance and stored in the construction sites, RMC can only be produced at the time that the job site manager calls for delivery because of the quick solidification of the cement. In other words, the delivery process must be coordinated with the working processes at construction sites. Two major types of costs associated with RMC trucks are cost of ownership and maintenance and opportunity cost due to improper dispatching schedule. An efficient and balanced dispatching schedule of RMC trucks not only improves the productivity of the RMC delivering process but also minimizes the cost of owning and maintaining RMC trucks. The current practice relies mainly on the dispatcher to determine the schedule based on his/her experience, which might not be optimal and presents the potential loss of profits.

[27] developed an optimization procedure that integrates the fast messy genetic algorithms (fmGA) with the CYCLONE simulation technique to determine the optimal dispatching schedule which minimizes the total waiting time at construction sites without interrupting the operations of casting concrete. The number of deliveries needed at a construction site is first determined in consideration of the amounts requested, the loading capacity of the truck, and the road bearing limit allowed by the regulation. Once the number of RMC deliveries required by different construction sites is identified, the dispatching schedule can be determined by deciding when each RMC truck is dispatched to which construction site. In their GA procedure, only the dispatching sequence of RMC trucks is considered as the decision variable. The departing time of each truck is decided by simulating the delivery process according to the dispatching sequence generated. The batch plant is assumed to keep on dispatching; as a result, RMC trucks tend to wait at construction sites. The fitness function has two major components: total truck wait time and penalty of interrupting casting concrete.

To solve the logistics problem arising in disaster relief activities, [80] proposed a two-phase procedure. The first phase employs ant colony optimization to build stochastic vehicle paths under the guidance of pheromone trails. A network flow based solver is used in the second phase for the assignment between different types of vehicle flows and commodities. The two phases are sequentially solved in an iterative manner until convergence. The objective is to minimize service delay in terms of the weighted sum of unsatisfied demand over all commodities and wounded people waiting at demand nodes and emergency units, subject to nine sets of constraints. The ACO was tested on 28 randomly generated problems and was found promising to solve large scale instances.

5.6. Equipment selection

Equipment is often needed to carry out a construction project and there are always a variety of options differing in technical capabilities and costs to choose from. There is thus an equipment selection issue. [32] developed a decision support system XpertRule for selecting opencast mining equipment (XSOME). XpertRule is a hybrid of knowledge base system and genetic algorithms. The knowledge base system within XSOME selects equipment in broad categories utilizing a decision tree that represents several nested production rules. XSOME also applies genetic algorithms to find the set of equipment that can achieve the minimum cost. The search for better solutions is controlled through three operators: one-point crossover, random mutation, and adaptation that is mutation-like but retains only improved values.

Earthmoving operations are commonly encountered in heavy civil engineering projects such as dams and highway. [56] presented a genetic algorithm designed to minimize the total cost of earthmoving operations, accounting for efficient use of the selected equipment fleet (numbers of different types of equipment) with a constraint on minimal equipment utilization. Total cost, duration, and utilization of the equipment fleets involved are estimated via computer simulation and passed to the developed algorithm to select the optimal fleet. The algorithm provides many options in performing operations such as fitness normalization, chromosome selections, crossover, elitism, and storing chromosome information to prevent performing the same simulation previously performed. In a separate study, [57] presented a framework for optimizing earthmoving operations utilizing computer simulation and genetic algorithms. The proposed framework was applied to the selection of earthmoving equipment fleets, taking into account their availability to contractors, with the objective to minimize time and cost of earthmoving operations. The simulation program was designed by utilizing discrete event simulation and object-oriented modeling coded in MS Visual C++. The optimization module was developed using genetic algorithms and Pareto optimality.

6. Project scheduling

Basic Program Evaluation and Review Technique (PERT) and Critical Path Method (CPM) are helpful scheduling techniques only when the project deadline is not fixed and the resources are not constrained by either availability or time. Studies to solve more realistic project scheduling problems can be classified into the following categories: (1) time-cost tradeoff; (2) resource allocation; (3) resource-constrained project scheduling; (4) resource leveling; and (5) integrated models that consider more than one category above. The objective of time-cost tradeoff analysis is to find a time-cost tradeoff curve showing the relationship between activity duration and cost. The objective of resource allocation is to allocate limited resources to activities so as to optimize a certain goal such as cost minimization. The objective of resource-constrained project scheduling is to determine the sequence of project activities and to allocate available resources to project activities in an attempt to optimize a given objective function such as minimizing project makespan. The objective of resource leveling is to smooth period-to-period resource demand. The common assumption in project scheduling studies is that a precise cost and duration for each activity can be defined beforehand by managers and planners. However, as construction activities are seldom entirely identical and it would be unrealistic to determine the exact time and cost values until the activity in question is complete. So far very few metaheuristic-based studies exist that have considered stochastic cost and/or duration.

Other than resource-constrained project scheduling (intentionally left out due to high numbers of studies on the topic), previous metaheuristic-based optimization research that focuses on each category mentioned above is reviewed in the subsections below.

6.1. Resource unconstrained scheduling

This section reviews eight metaheuristic-based studies of project scheduling without considering resource constraints. The first three consider cash flows with the objective to maximize net present value (NPV). The next three try to find the critical path like CPM/PERT. The last two studies focus on *linear projects*.

[24] developed a simulated annealing based solution procedure for scheduling the starting time of a project to maximize its net present

value (NPV) for a given discount rate, subject to precedence relations among the activities and the constraints on the event occurrence times. Each net cash flow magnitude depends on the event's occurrence time and is assumed known a priori. Three different strategies for generating neighborhood solutions were compared and the best one was identified.

To address the problem of simultaneously determining the amount, timing and location of progress payments in resource unconstrained projects in order to maximize net present value, [18] proposed a two-stage heuristic where simulated annealing is used in the first stage to determine a set of payments, and activities are rescheduled to improve project net present value in the second stage. They adopted the activity-on-arc (AoA) representation and assumed that expenses were incurred at the beginning of each activity. The initial solution of the simulated annealing algorithm is based on an early time schedule. Neighborhood solutions are generated by randomly selecting one time period with a payment from the current solution for removal and by randomly selecting one time period without a payment from the current solution for inclusion.

[33] developed two metaheuristics, SA and TS, to solve the multi-mode project payment scheduling problem (MPPSP) with the objective to maximize the net present value of the contractor under the constraint of project deadline but no constraint on resource. The project is represented as an AoA network and cash flows in the project are all attached to events. A feasible solution of MPPSP is represented by three vectors: a vector denoting to which events the payments are attached, a vector of modes of activities, and a vector of occurrence times of events. Using instances generated by ProGen, the test results showed that SA outperformed the other three approaches, i.e., TS, multi-start iterative improvement, and random sampling.

[15] presented an ant colony optimization based methodology as a possible improvement to traditional CPM techniques for determining the critical (longest) path in construction networks. Two advantages of ACO over the traditional method were noted: allowing for the simultaneous calculation of both the longest and the shortest path during the same path-traversal procedure, thus minimizing the computational load, and the shortest and longest path can be found for any chosen start node and end node. In addition, it was shown that ACO could find the critical path in agreement with the traditional CPM-based calculations. However, their ACO is not able to provide total float value for each activity. A similar study was carried out by [1], in which they used ant colony optimization to solve and calculate both deterministic and probabilistic CPM/PERT networks. It was shown that ACO could produce optimal and good suboptimal solutions compared to conventional methods. Both studies adopted the AoA representation scheme. [21] proposed AoN-based ACO algorithms and showed that they are better than AoA-based ACO algorithms.

[55] developed a genetic algorithm for scheduling repetitive construction projects in consideration of precedence and constraints of work continuity. The objective is weighted sum of project duration and cost. The proposed method can be applied to both discrete and continuous relationships between durations and direct costs of activities. Two types of activities are distinguished: type α activities desired to be continuously performed to maintain the work continuity of crews (resources) and type β activities with violations of the crew work continuity constraints allowed. Each chromosome encodes durations per unit quantity of activity work in binary. For a given set of durations, activities are scheduled in two stages: stage 1 for type α and stage 2 for type β including the backward step and the forward step. The genetic operators are one-point crossover and uniform mutation. Two notable assumptions of their method are the same production rate for each activity throughout all units and only one crew working at a time.

[38] developed an optimization model for workgroup based repetitive scheduling and solved it using a genetic algorithm with the objective to maximize total project net present value while conforming to the proper work sequence between workgroups and maintaining the work continuity of resources. The assumption is that an arbitrary work order can be assigned to each workgroup. Each chromosome consists of the same number of segments as the number of workgroups identified in the repetitive project. In each segment, a pair of genes represents the assigned resource number and work sequence number of an activity. The genetic operators employed include external crossover, internal crossover, external mutation, internal mutation, and roulette wheel selection.

6.2. Realistic project scheduling

6.2.1. Time-cost tradeoff analysis

There are tradeoffs between time and cost to complete project activities. Time-cost tradeoff analysis is one of the most important aspects of project management. In general, the less expensive the resources used, the longer it takes to complete an activity. The "normal time" for completing activity i, $T_n(i)$, is determined by calculating the minimum cost ("normal cost") for the activity, $C_n(i)$. The minimum duration for an activity is known as the "crash time", $T_c(i)$, and the cost associated with the crash time is called "crash cost," $C_c(i)$. Any intermediate point in between the above-mentioned two points can be computed in accordance with the activity's utility curve. Using the critical path method (CPM), the overall project cost can be reduced by using less expensive resources for noncritical activities without impacting the project duration.

Time-cost optimization problems in construction projects are characterized by the constraints on the time and cost requirements. Typically, if a project is running behind the scheduled plan, one option is to compress some compressible activities on the critical path so that the target completion time can be met. It is unavoidable that the cost of these compressed activities will be increased as a result. It should be noted that time reductions on the critical path may affect other paths throughout the network. In compressing activities, resource-constrained problems may also arise when the amount of required resources at any instant exceeds the available resources. In this case, integrated models of time-cost tradeoff analysis and resource constrained project scheduling are necessary.

Activity time and cost are commonly considered exact but many situations in the real world could affect them. Therefore, activity time and cost are realistically uncertain. Uncertain activity time and cost can be treated statistically if there is enough data to justify its use; else, fuzzy set and possibility theory is more appropriate. Previous studies on time-cost tradeoff analysis are reviewed below in two groups depending upon whether single objective or multiple objectives are considered.

6.2.2. Single objective time-cost tradeoff

Assuming a linear relationship between time and cost, [48] presented several improvements to the basic GA, particularly improved crossover and mutation, to significantly increase the efficiency in search for optimal solutions. In this case, the objective is to minimize the total cost, C_t, incurred by compressing some activities to shorten the total duration to a targeted value, T_t, as formulated below.

$$Min\ C_t = \sum_{\forall i} C_i = \sum_{\forall i} [a_i T(i) + b_i] \tag{2}$$

Subject to

$$\sum_{\forall i} T(i) = T_t$$
$$a_i = [C_n(i) - C_c(i)] / [T_n(i) - T_c(i)]$$
$$b_i = [C_c(i) T_n(i) - C_n(i) T_c(i)] / [T_n(i) - T_c(i)]$$
$$T_c(i) \le T(i) \le T_n(i).$$

Their study treats crash times as continuous, which may be impractical, as in the construction industry the minimum time fraction is normally half a day.

Also assuming a linear relationship between time and cost, [68] developed a GA-based time-cost optimization program, which integrates a project management system, i.e. Primavera, with a GA system called SUGAL. The project completion date is used as the first criterion to determine whether a solution is valid or not. The project cost is used as a measure of the solution fitness. Since only crashable activities are encoded, the total computed only includes the cost of activities that have not been started and is equivalent to the remaining cost to complete the project. An invalid solution is given a fitness value higher than that of the all-crashed solution and is allowed to participate in subsequent optimization process.

[42] studied a project scheduling problem with the following assumptions: (1) each activity is non-preemptive; (2) each activity can only be processed if the loan needed is allocated and all the precedent activities are finished; (3) all of the costs needed are obtained via loans with some given interest rate; and (4) all activity durations are stochastic. The project is represented as an AOA directed network. The schedule is described as the allocating time of all the loans needed for activities. Their method integrates stochastic simulation with genetic algorithm to solve three types of stochastic models: the expected cost model, the α-cost model, and the probability maximization model.

For time-cost tradeoff analysis, [22] introduced a modified shuffled frog-leaping algorithm (MSFL), which involves adding a new search-acceleration parameter to the original shuffled frog-leaping algorithm. The search acceleration parameter, which can be a positive constant value, linear, or nonlinear function of time, provides the means to balance between global search and local search. The MSFL was applied to find the optimal set of construction options that meet the deadline for minimum total cost for an 18-activity construction project. Each activity has five optional methods of construction varying from cheap-and-slow one to fast-and-expensive one. The effect of the search-acceleration parameter was investigated and a value between 1.3 and 2.1 was found to have the best chance of finding the global optimum with the least number of iterations.

[69] performed an analysis of time-cost relationship using time coupling method *TCM III*, which was designed to ensure minimum project lead time. They employed a hybrid evolutionary algorithm to find a schedule such that the maximum completion time is minimal. The algorithm employs three procedures: a local search procedure to determine a set of local minima, a procedure to determine a set of fixed elements and positions in next iteration, and a procedure to generate a new population in next iteration.

6.2.3. Multi-objective time-cost tradeoff

Alternatively, a time-cost tradeoff problem can be treated as a multi-objective optimization process to minimize both project duration and cost. Following this line, [25] developed a computer program called TCGA, which uses Microsoft Excel as a front end for data entry and a macro to execute a multi-objective GA implementing the Pareto front approach. Their model is based on the convex hull concept developed from linear programming; and hence the time-cost tradeoff curve may not truly reflect the real optimal solution set. Moreover, only direct costs are considered in their tradeoff curve; thus the optimal solution found might not be optimal in terms of the total cost. Subsequently, they presented a hybrid approach that combines simulation with genetic algorithm to solve the stochastic time-cost tradeoff problem [26]. In this stochastic case, the convex hull is defined by the means values of the project durations and costs of the solutions in the population. The fitness function is defined as the solution's average minimal distance to the convex hull.

[87] proposed an improved GA-based multi-objective model, which implements a modified adaptive weight approach (MAWA) that derives adaptive weights used in fitness evaluation based on four different combinations of extreme points on the time-cost tradeoff curve. Each chromosome encodes the option for each activity with length equal to the total number of activities. Finite numbers of options are considered with each varying in activity duration and cost. It was shown that the modified adaptive weight approach could help guide GA to obtain a better Pareto front than the original adaptive weight approach [29]. To address the phenomenon of “genetic drift” typified by the existence of multiple peaks of equal height, [88] introduced Pareto ranking, niche formation, and adaptive mutation rate into the improved GA-based multi-objective model. They implemented the proposed model using the macro language of Microsoft Project 2000 for Windows.

[5] formulated the time-cost tradeoff problem in PERT network as a multi-objective problem that involves four conflicting objective functions: minimizing project direct cost, minimizing the mean and variance of the project completion time, and maximizing the probability that the project completion time does not exceed a given deadline. The activity durations are assumed independent with generalized Erlang distributions. An analytical method is used to compute the distribution function of the project completion time in PERT networks. The decision variables are the allocated resource quantities. They applied a “genetic Algorithm for Numerical Optimizations of Constrained Problems (GENOCOP)” to solve it using a goal attainment technique. Factorial experiments were performed to identify appropriate GA parameters. The limitation of their model is that the state space can grow exponentially with the size of the project network.

[40] developed a parallel multi-objective GA framework to optimize construction planning in a large scale construction project. The framework incorporates four modules: (1) a multi-objective optimization module that is capable of simultaneously minimizing both the project duration and cost through the optimal selection of construction methods, crew formations, and overtime policies, (2) a global parallel GA module in which one process is designated as the manager processer to perform the generation evolution task and the others as worker processers to evaluate the fitness of solutions, (3) a coarse-grained parallel GA module in which each processor evolves a subpopulation independently but the best found solutions are allowed to share among processers through the migration process, and (4) a performance evaluation module. Test results were obtained using 183 experiments of various combinations of 3 construction projects of different sizes (with 180, 360, and 720 activities, respectively), 11 different clusters of parallel processors, 2 migration rates, and 3 migration intervals.

[66] presented a neural network embedded multi-objective GA to solve non-linear time-cost tradeoff problem of real world engineering projects. It employs a multi-objective GA to search for the set of non-dominated solutions and a neural network model for rapid estimation of time-cost relationship by invoking its function approximation capability. The proposed algorithm was tested with an 18-activity project; 7 non-dominated solutions were found and used to construct a 1-3-1 neural network model of TCT profile for each activity. The numbers of training data points is 5, which seems low compared with the model size.

[79] employed ant colony optimization (ACO) to deal with the time-cost tradeoff problem. Their ACO incorporates the modified adaptive weight approach and fitness function proposed by [87]. Based on the test results of a 7-activity project, it was shown that all Pareto optimal solutions derived from the ACO algorithm were better than those obtained by the GA of [87]. [62] applied an ant colony optimization algorithm to solve time-cost optimization problems, called ACO-TCO. The modified adaptive weight approach was used to convert bi-objective of time and cost into one single objective. Test results showed that ACO-TCO could find better solutions in shorter time than [88].

[2] developed a multicolony ant algorithm to solve the time-cost multiobjective optimization problem. Their algorithm employs one colony of ants for each objective and both colonies iterates alternatively. All solutions produced in each colony for any cycle iteration are evaluated based on both objectives and the nondominated ones are transferred to an offline archive. The solutions in the archive at the end of the process contain the final set of nondominated solutions. An 18-activity project with varying indirect unit cost was tested to show that the proposed algorithm outperformed that of [88].

Considering activity time and cost as fuzzy numbers, [23] presented a multi-objective genetic algorithm specially tailored to solve the discontinuous and multi-objective fuzzy time-cost model with relatively large search space. Depending upon the risk level and manager's degree of optimism, the set of nondominated solutions can be found.

[28] applied a harmony search algorithm to solve time-cost tradeoff problems. The objective is to minimize total cost, which consists of direct cost and time related indirect cost. A 7-activity network and an 18-activity network were used to show that the proposed method was competitive in comparison with [79,87].

6.2.4. Resource allocation

6.2.4.1. Single objective/weighted multi-objective resource allocation. Focusing on software development projects, [10] utilized GALib, available at http://www.lancet.mit.edu/ga/, to automatically determine a near-optimal allocation of resources and schedule that satisfies a given task structure and resource pool. It is assumed that the estimated effort for each task is known *a priori* and can be obtained from any known estimation method such as COCOMO [6]. Each solution is represented as a two-dimensional array with employee enumerated along the rows and the tasks along the columns. Each employee's effort that could be assigned to any single task is constrained to the set {0%, 25%, 50%, 75%, 100%}. The fitness function comprises of four components: penalty of infeasible solution, minimum level of overload, minimum cost, and minimum time span. The default genetic operators provided by GALib are used, which include UniformInitializer, ElementComparator, FlipMutator, and OnePointCrossover.

[16] presented a genetic algorithm for solving-resource allocation problems with the objectives to maximize system reliability and minimize testing cost, subject to resource constraint. Each chromosome encodes testing time for each individual module in testing a modular software system. The modular structure is assumed to be parallel and serial. The failures among different modules are assumed independent. A special procedure was developed for generating the initial population. The fitness function is defined as weighted objectives. The genetic operators include roulette wheel selection, one-point crossover, and alternating mutation. Two examples were tested to illustrate the method.

Construction projects require resources to carry out the necessary operations. To simplify the problem, the demands are often assumed static, i.e., not changing with time. In reality, the demands on these construction facilities may change over the project duration, especially for larger projects involving interdisciplinary fields. Hence, it is desirable to be able to formulate an optimal strategy for allocating site-level facilities at different times of the project so as to minimize the total cost. [11] developed a two-stage dynamic model to assist construction planners in formulating the optimal strategy to establish potential intermediate temporary site-level facilities that support construction such as batch plants, lay-down yards, receiving warehouses, and various workshops. The problem-solving process is split into two stages, namely, a lower-level stage solved by a standard linear programming method and an upper-level stage solved by a genetic algorithm in consideration of the time-dependent dynamic allocation. The principal objective is to identify which combination of a pre-enumerated set of construction site-level transfer facilities are worth establishing for any time interval during the course of a project to minimize the total system cost. In addition to the transportation and variable operating costs acquired from the lower-level step, the total cost also comprises costs of establishment, closedown, and fixed operation costs for different combinations of transfer centers in different periods. The lower-level step focuses on the instantaneous transportation problem to minimize the total transportation cost plus variable operation cost for each period and facility type. It was shown that the two-stage algorithm found the same solution as that by the mixed integer programming and required much less amount of computational effort.

[81] presented an ant colony optimization algorithm for solving the nonlinear resource allocation problem with the objective to minimize the total cost, subject to the constraint that the quantity of resource allocated to activity i is in a specified range and the last activity consumes all the remaining resource. To ensure that the constraint is satisfied, an adaptive resource bound updating rule was developed. The effects of evaporation rate and number of ants were studied. Experimental results showed that the proposed algorithm outperformed GA and naïve exhaustive search. The convergence of the ACO-based algorithm was analyzed using node transition entropy to validate that the obtained quality solution was due to the consensus knowledge possessed by the whole ant colony instead of by the wandering of a lucky ant. In [82], the same study was repeated using particle swarm optimization in place of ant colony optimization. The convergence behavior was analyzed by observing the variations of particle entropy.

[4] used a GA to tackle the problem of software project scheduling in deciding who does what during the software project lifetime. Assuming no overtime, each entry in the two-dimensional employee-task matrix takes one of eight equally distributed values between 0 and 1. Each chromosome thus encodes in binary values of length equal to 3 times the product of total number of employees and total number of tasks. The fitness function is defined as the inverse of weighted project cost and project duration for a feasible solution or the inverse of weighted project cost and project duration plus weighted sum of penalties for an infeasible solution. They developed an instance generator (http://www.tracer.lcc.uma.es/problems/psp) to create fictitious software projects for testing.

[63] studied the problem of optimal assignment of resources to realize the features of a sequence of software releases called Resource Allocation for Software Release Planning (RASORP). The assumption is that only one developer can work on one task at any given time. For its solution, they developed a two-phase approach called Focused Search (FS), which combines the strength of special structure integer linear programming with the power of GA. The GA in phase 2 can be applied alone, called Unfocused Search (UFS) in this case, but the search space would be substantially larger. Both FS and UFS were tested with 600 randomly generated cases and the results were compared with greedy search.

6.2.4.2. Multi-objective resource allocation. [65] investigated the possibility of using genetic algorithms to solve a multi-objective resource allocation problem, which is formulated as a network model, where limited supply represented by stages and the resource available gives possible state in each stage. A set of Pareto-optimal solutions are found to avoid the need to specify the weight for each objective. The initial population is randomly generated. Infeasible solutions are rejected and replaced with randomly generated feasible ones. A procedure is used to find the non-dominated set of solutions. For evolving next generation, the elitist and roulette wheel selection, one-point crossover, and alternating mutation are used. The random weight approach is used to aggregate multiple objectives into one and the selection probability is computed based on the aggregated objective value. An example from the literature was tested.

Formulating multi-criteria human resource allocation problem as a network model, [54] developed a hybrid GA for its solution. Two objectives are to minimize the total cost and to maximize the total efficiencies of the human resource decisions subject to resource constraints. Each chromosome encodes a vector of allocation path. Infeasible solutions are rejected. The GA operators include roulette wheel selection, uniform crossover, random replacement mutation, and an iterative hill climbing local search method. A numerical example was used to illustrate the method.

[83] proposed a PSO and a hybrid variant which embeds a hill-climbing heuristic into the PSO for searching Pareto-optimal solutions of multi-objective resource allocation problem. To guarantee that all resource constraints are satisfied, they devised an adaptive resource bounding technique to construct feasible solutions. A simulated data set, with two objectives: one maximizing benefit and the other minimizing cost, was generated for testing. The results showed that the proposed algorithm outperformed the original NSGA II [20] and another variant that incorporates the proposed adaptive resource bounding technique.

[7] proposed a modified ACO to obtain a set of Pareto solutions efficiently for the multi-objective resource allocation problem. The modifications include a special heuristic information, pheromone updating rule, and selection probability equation. The proposed algorithm was tested with the same example used in [54] and found to perform better than the hybrid GA developed in [54].

[17] developed a multi-objective evolutionary algorithm, called NSGA-II-RAP, for solving two resource allocation problems: university class timetabling and land-use management. NSGA-II-RAP follows the basic procedure of NSGA-II, but with differences in the chromosome representation, EA operators and run-time guidance. Each chromosome is a two-dimensional matrix and the EA operators are tailored to it. A problem based run time guidance is used to handle infeasible solutions generated at the time of initialization, and/or after crossover operation.

6.2.5. Resource leveling

[74] developed a multiheuristic-based local optimizer and a hybrid model combining the local optimizer with simulated annealing for the solution of resource leveling problems. Let y_t be the sum of resource requirements of the activities performed at time unit t, and $\bar{y}_t$ be a uniform resource level over the project duration. The objective of resource leveling is to minimize total deviations of resource requirements $Z = \sum_{t=1}^{T} \left(y_t - \bar{y}_t\right)^2$ or simply $Z = \sum_{i=1}^{T} y_t^2$. The local optimizer has four independent searches and the best schedule out of the four independent searches is recorded as the local optimum. Note that more than one schedule might have the same objective value. To find a more suitable schedule, the above objective can be modified to include daily deviation. In this study, activities are assumed non-preemptive and require only one resource type.

[47] proposed a GA-based resource leveling scheduling (GARLS) system. The GARLS assumes that (1) activities cannot be split; (2) activity durations and the network logic are fixed; and (3) demands on resources are kept constant throughout the duration of an activity. The objective of the GARLS is to minimize $RLI = \sum_{k=1}^{m} w_k Z_k$ subject to $t_i - ES_i \leq TF_i,\ t_i \geq 0,\ i = 1,\ldots,n$,where w_k is the weight assigned to resource type k, Z_k is the total deviations of resource type k, t_i is the starting date of activity i , ES_i is the early starting date of activity i, and TF_i is the total floating time of activity *i*. Each chromosome encodes the starting dates of activities. The selection strategy is roulette wheel combined with the elitist. The genetic operators include one-point crossover and uniform mutation.

[45] proposed a resource leveling model, in which the combinative effect of both uncertain activity duration and resource leveling are taken into consideration. In that model, Monte Carlo simulation is used to model the uncertainties of activity duration (beta distribution assumed) and GA is adopted to search for the best schedule that minimizes the averaged resource leveling index, $RLI_{avg} = \sum_{p=1}^{SN} RLI_p \Big/ SN$,where SN is the total simulation cycles. The GA used in this study is identical to that used in [47].

[30] presented a genetic algorithm-based system for performing resource leveling in *linear projects* under the linear scheduling method scheme. The objective is to minimize the absolute daily (or other appropriate time unit) fluctuations and the absolute deviation from the average resource usage throughout the project, i.e. to minimize

$$Z = \sum_{i=1}^{T} \left(y_i - \bar{y}_i\right)^2 + \beta \sum_{i=1}^{T-1} \left(y_{i+1} - y_i\right). \tag{3}$$

Each chromosome encodes an integer-valued string to represent the resource usage throughout the project. The fitness function is defined as the inverse of the objective function. The GA evolutions utilize roulette wheel selection, one-point crossover, and a specifically-devised mutation operator. It is assumed that activities are non-preemptive and only one type of resource is required.

6.2.6. Integrated models

This section reviews seven studies that consider more than one aspect of realistic project scheduling. The first four studies try to optimize single objective function while the last three optimize multiple objectives.

[9] presented a resource scheduling approach using Genesis, which is a commercial GA package. A general optimization model was formulated to encompass both resource allocation and resource leveling problems. Given the resource availability profile, the model tries to minimize the deviations of required resources from the available resource profiles, which include resource underutilization and overutilization and penalty on project extension, subject to the precedence relationships among project activities. For chromosome representation, the random key concept is adopted. Accordingly, the gene values are real numbers which code for two decisions: the scheduling priority of an activity, and the percentage of current float to be utilized before the start of the activity. A schedule builder was developed as a supplement to the GA routines available in the package. The schedule builder contains the logic to construct a valid schedule from the decoded chromosome in consideration of the constraints of the application, including hard constraints such as precedence relationships between project activities, and soft constraints such as project durations and resource availability. In applying GA to scheduling problems, illegal schedules introduced by the GA operations must be dealt with.

[34] improved resource allocation and leveling heuristics by using a GA technique to search near-optimum solution in consideration of both aspects simultaneously. The objective is to minimize the total project duration under resource constraints while also minimizing the appropriate moments of selected resources important to resource leveling. The moment around the horizontal axis should be chosen if the focus is on reducing daily resource fluctuations. One of two moments around the vertical axis could be chosen to reduce either the resource utilization period or the release of the resource at an early date. Each chromosome encodes a string of elements, with each element denoting the assigned activity priority level. The fitness of a schedule generated based on the set of priority levels is determined by the relative improvements on weighted duration and moments over the initial schedule. Using the macro language of Microsoft Project, the GA procedure was coded and then used to search for an optimum schedule for the case study at hand. This implementation takes

advantage of the software's feature of allowing user-specified activity priorities.

[71] presented an augmented Lagrangian genetic algorithm model for resource scheduling in consideration of resource leveling and resource-constrained scheduling simultaneously. The objective is to minimize the total cost subject to precedence relationship constraints, resource allocation and leveling constraints, project duration constraint, and non-negativity constraint. Each activity is allowed finish-start, start-start, finish-finish, and start-finish precedence relationships with its preceding/succeeding activities. The quadratic penalty function is used to convert a constrained scheduling problem into an unconstrained one.

[67] developed a genetic algorithm to solve multi-mode resource-constrained discrete time/cost tradeoff model in consideration of (i) constrained renewable resources, (ii) direct and indirect costs computed according to the usages, and (iii) time-cost tradeoff in executing an activity in a selected mode. The assumptions include activities may not be preempted and mode cannot be changed once selected. The objective is either to minimize total cost subject to constraints including precedence relationships, one mode for each activity, resource availability, and project deadline, or to minimize project duration subject to budget constraint. Each chromosome is represented by three vectors for activity priority, mode, and time-cost tradeoff option. The serial schedule scheme is used to generate project plan from the chromosome. Infeasible solutions are penalized. Special selection and genetic operators are used.

[46] proposed a multi-criteria scheduling model, which integrates the time-cost tradeoff, resource constrained allocation, and unlimited resource leveling into a unified framework. In this integrated architecture, a resource constrained time-cost tradeoff model is first analyzed. The GA techniques are used to search optimal or near-optimal solutions. The non-dominated solutions with the objectives of project durations and costs are found using Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), which is a multi-criteria decision making technique. Next, a GA-based resource leveling system receives information about the optimal schedule and related resource requirements from the resource-constrained time-cost tradeoff model, and then adjusts activity starting dates to obtain a more even resource profile. The objective is to minimize the deviation between actual resource usage and the uniform resource usage plus a penalty term, used to prevent the violation of precedence relationships and resource overutilization. The GA chromosomes encode a string of activity durations in time-cost tradeoff analysis, a string of activity starting dates in unconstrained resource leveling, and a sequence of activities in resource-constrained allocation. One-point crossover and uniform mutation apply to time-cost tradeoff and unconstrained resource leveling. For resource-constrained allocation, a new crossover operator called UX3 and a new mutation operator called UM3 were developed. The chromosomes are selected based on the roulette wheel principle.

[12] proposed a two-phase GA model in consideration of both time-cost tradeoff and resource scheduling. A GA-based multi-objective optimization technique is adopted for the time-cost tradeoff analysis. Each chromosome in this GA encodes the mode for each activity. For each such chromosome, the 2nd phase GA is called to determine the best schedule that minimizes the project makespan subject to precedence and resource constraints. Each chromosome in the 2nd GA encodes an activity list. A schedule is constructed following the order of the activities in the topological sort subject to resource availability. Finally, the shortest project duration and associated cost are returned as the fitness values for the chromosome of 1st phase GA.

Considering fuzzy activity time and cost, [84] proposed a time-cost-resource utilization optimization (TCRO) model. Based on a specified risk level, fuzzy activity and cost are converted into a time interval and a cost interval based on α-cut. A discrete numbers of time-cost options are then selected. They used NSGA-II [20] to find the set of nondominated solutions that simultaneously minimize total cost, time, and required resource moments. Each chromosome is encoded by two vectors: vector of options and vector of activity start time. Each new generation of solutions is produced by standard roulette wheel selection, one-point crossover, and altering mutation operators. Adding the resource leveling function to time-cost optimization was shown to produce more practical solutions in terms of resource allocation and utilization.

7. Project monitoring and control

Project monitoring and control are necessary to measure the deviation of actual from planned progress so that corrective actions can be taken to meet project budget and deadlines. [36] presented a practical project control model in the form of a spreadsheet integrated with commercial project management software, and a genetic optimization procedure. Once the actual progress is entered, their model can be used to determine optimal corrective actions for the remaining portion of the project activities during the execution of construction projects, considering time, cost, and resource constraints, concurrently. Their model incorporates some aspects of critical chain project management concept without much change to the traditional earned value (EV) analysis.

8. Discussion

The above review of literature clearly indicates that the use of metaheuristics in project management is a relatively young (not yet two decades old) and very promising. Literally, any problem/issue that cannot be easily solved by conventional optimization techniques and specifically designed heuristics is a candidate problem for metaheuristics. Most existing commercial project management software packages only provide limited project scheduling, project tracking and reporting aids and fall short in their computational capabilities. It is our belief that metaheuristics offer a rich set of computational techniques that can greatly enhance current project management tools. However, before this can happen the practitioners must be convinced and are willing to pay for such products. Undoubtedly most test cases shown in the literature are small compared to real world projects. Able applied researchers would love to have the opportunity to work with real world companies on real world projects. The question is whether there is any company interested in such a large scale testing and willing to pay for it. In our opinion, it is in their interest for big software companies in this market to step up to the plate to make this happen.

Each metaheuristic has its own set of tunable parameters. An important issue in using a metaheuristic is selecting a good set of values. On each topic reviewed above, whether the metaheuristic chosen for a particular study is absolutely the best is unknown because most studies did not carry out a comprehensive testing of a wide range of possible parameter settings. It is time consuming to determine the best set of parameter settings for any metaheuristic. At the minimum, a recommend range of values shall be provided for the users. This practice will be similar to the recommended range of tool speed and cutting speed for any combination of cutting tool and material in the machining world. Furthermore, very few studies employed more than one metaheuristic and evolutionary algorithm seems to be the dominant metaheuristic in this area. Hence there are opportunities for obtaining better results by using other metaheuristics that have never been applied before such as differential evolution or devising a new and better metaheuristic tailored to it. For the latter, hybrid algorithms are the trend; but it might mean more parameters to select [53].

To bridge the gap between a real world problem and a formulated model, the model must be formulated based on reasonable assumptions.

Often time assumptions were made to simplify the problems for easier solutions. This is particularly true for traditional mathematical approach. In the cases the assumptions made were too remote from reality the solutions found would become useless. To make research results more useful in practice, efforts should be made to develop solution methodologies that can be used to solve real world problems without imposing unnecessary and unrealistic restrictive assumptions. On this issue, metaheuristics are not constrained as much as traditional mathematical approaches and are more suitable to solve more realistic models and problems.

More specific comments and some ideas to further advance each area reviewed in this paper are laid out in the following paragraphs.

A possible extension to the robot selection problem is to use micro-GA, which has the potential to reduce the population size and the computing requirements. For the pile group foundation design problem, binary coding must be changed in order to design pile groups consisting of hundreds of piles and the parameterless penalty function proposed by [19] should be considered to avoid the difficulty in determining the values of penalty parameters. For design process sequencing, a desirable study is on the classification of DSM difficulty prior to performing optimization.

For cost estimation, the first two papers reviewed applied GA to tune model parameters with neural network and fuzzy neural networks used as the modeling methods, respectively. The same problem can definitely be modeled by using any statistical or machine learning method with parameters tuned by other metaheuristics other than GA.

Possible extensions to site preparation include considering unequal cut and fit locations, incorporating haul roads and terrain features into the distance calculation, and adding penalties to routes that force the equipment to negotiate undesirable terrain. Moreover, multiple types of equipment could be considered and additional constraints such as time of repair and availability of equipment could also be imposed under special situations.

Possible extensions to site-level layout problems include considering the most general case that each location and facility has a different shape and size, and considering site layout as a dynamic problem with the project progresses in phases. Ideally, in a dynamic layout problem the determination of layout and the scheduling procedure need to be carried out concurrently to demonstrate optimality.

For the on-site routing and path planning problem, possible extensions include investigating the applicability of fuzzy based multi-criteria evaluation as well as hybrid optimization search algorithms. The one-dimensional cutting stock problem solved by GA can be extended to two-dimensional cutting stock problems such as cutting a set of rectangular items from steel plates with arbitrary sizes for constructing steel bridges.

Possible extensions to the dispatching of RMC trucks are (1) to consider the uncertainty involved with input parameters such as traveling duration and casting duration using either the probability theory or the fuzzy set theory, (2) to provide real time optimization of the dispatching process by incorporating a global positioning system, and (3) to expand the single batch plant model into multiple batch plants and construction sites. The routing problem in disaster relief logistics can be extended to consider location problem as well.

Most project scheduling studies reviewed in this paper considered that activity's duration and cost are deterministic, which are quite uncertain in reality. The exceptions are those of [5,26,42,45], in which stochastic activity durations are considered. Two studies that consider fuzzy time and cost are [23,84]. Possible extensions to construction scheduling in general include (1) considering the uncertainty of activity's duration and cost, (2) providing guideline for selecting algorithmic parameters, and (3) conducting sensitivity analysis of changing constraints.

Possible extensions to resource unconstrained project scheduling include (1) solving the problem for the case where uncertain cash flows are dependent on the time of their occurrence in accordance with various payment methods, (2) conducting additional testing and analysis of different neighborhood approaches that take advantage of the network characteristics as well as the cash flow characteristics for the payment scheduling problem, (3) finding a way to provide total float value for each activity when using ACO to find the critical path; (4) allowing multiple crews working simultaneously in different units at the same activity in scheduling repetitive construction projects, and (5) considering additional constraint in the work order in each workgroup in workgroup-based repetitive scheduling.

For time-cost tradeoff analysis possible extensions include (1) considering the stochastic nature of the construction project, (2) adding the resource constraints that might arise as a result of compressing the activities, (3) ensuring a steady performance when applying to large-scale projects, (4) examining whether time-cost savings can be realized by adopting new construction methods/techniques or re-sequencing certain construction activities, (5) developing a decision support system to assist project managers in choosing the best solution from a long list of non-dominated solutions according to his/her peculiar condition, and (6) optimization of not only time-cost but also other project goals such as quality and safety.

Possible topics for future studies on resource allocation problems include (1) considering dependent failures of modular software in testing-resource allocation, (2) developing a decision support system to assist project managers in choosing the best solution from a long list of non-dominated solutions according to his/her peculiar condition as in time-cost tradeoff analysis, and (3) developing benchmark test instances for evaluating the performance of a particular algorithm and for comparing the performances of different algorithms.

Possible extensions to resource leveling studies include (1) allowing crashing of a schedule using time/cost curves, (2) allowing splitting of activities, (3) leveling of multiple resources, (4) allowing the extension of project deadline with penalty, and (4) integrating resource leveling with resource allocation and time-cost tradeoff analysis.

Among the seven studies that make use of integrated models, two integrate resource leveling and resource allocation and both combine multiple objectives into one; two integrates time-cost tradeoff analysis with resource constrained project scheduling; one considers resource leveling and resource constrained scheduling; one considers time-cost tradeoff and resource leveling; and one considers time-cost tradeoff, resource allocation, and resource leveling. All of them employ genetic algorithm as the optimization technique. As more aspects are considered, the effort required for their solution increases. It is thus desirable to derive guideline as to when it is beneficial to consider what.

9. Conclusions

This paper has presented an overview and detailed reviews of previous studies that employed one or more metaheuristic(s) as the optimization technique(s) to determine the near optimal solutions for project life cycle related problems. All the publications included in this review appeared in archival journals. Despite of our effort to include all journal publications in the targeted areas, unintentional misses are expected and apologized. Due to the already high number of papers, articles published in conference proceedings are intentionally left out.

The previous studies are organized around topics and each individual paper is analyzed. The results clearly indicate that the use of metaheuristics in project management is very promising. Literally, anything that cannot be easily solved by conventional exact optimization techniques and specifically designed heuristics is a candidate problem for metaheuristics. Among all metaheuristics, EAs / GAs are most popular and four reasons were offered to explain the reason why. Among all problems studied by metaheuristics with resource-constrained project scheduling problems excluded, top the list are three problems: site/floor layout, time-cost tradeoff analysis, and resource allocation.

General deficiencies of previous studies are highlighted. First, most test examples are small compared to real world problems. Secondly, assumptions made in some models are far from reality, rendering the results of little value. Thirdly, experiments are rarely carried out to produce useful guide for selecting algorithmic parameter values. Despite of these deficiencies, it is our belief that the potential of metaheuristics is high. However, efforts need to be made to correct the above deficiencies, to show that metaheuristics can be useful to solve complicated real world problems, and to develop new metaheuristics with fewer parameters. The power of metaheuristics has not been utilized in commercial project management software packages today, but the picture could be different in the future. It is in the best interest of major software developers or project management firms to work with researchers to further validate the potential benefits that metaheuristics can bring to real world project management.

Lastly, some topics required for further studies are identified with the hope that researchers, both new comers and experienced veterans, would pick up some of these ideas and work on them to further advance this area of research.

Acknowledgement

The authors acknowledge the constructive comments made by anonymous reviewers that have help improve the quality of the paper.

References

[1] H. Abdallah, H.M. Emara, H.T. Dorrach, A. Bahgat, Using ant colony optimization algorithm for solving project management problems, Expert Systems with Applications 36 (2009) 10004–10015.

[2] A. Afshar, A.K. Ziaraty, A. Kaveh, F. Sharifi, Nondominated archiving multicolony ant algorithm in time-cost tradeoff optimization, Journal of Construction Engineering and Management 135 (7) (2009) 668–674.

[3] M.H. Afshar, M.A. Mariño, A parameter-free self adapting boundary genetic search for pipe network optimization, Computer Optimization and Applications 37 (2007) 83–102.

[4] E. Alba, J.F. Chicano, Software project management with GAs, Information Sciences 177 (2007) 2380–2401.

[5] A. Azaron, C. Perkgoz, M. Sakawa, A genetic algorithm approach for the time-cost tradeoff in PERT networks, Applied Mathematics and Computation 168 (2005) 1317–1339.

[6] B. Boehm, Software Engineering Economics, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[7] S.K. Chaharsooghi, A.H.M. Kermani, An effective ant colony optimization algorithm for multi-objective resource allocation problem, Applied Mathematics and Computation 200 (2008) 167–177.

[8] C.M. Chan, L.M. Zhang, J.T.M. Ng, Optimization of pile groups using hybrid genetic algorithms, Journal of Geotechnical and Geoenvironmental Engineering 135 (4) (2009) 497–505.

[9] W.-T. Chan, D.K.H. Chua, G. Kannan, Construction resource scheduling with genetic algorithms, Journal of Construction Engineering and Management 122 (2) (1996) 125–132.

[10] C.K. Chang, M.J. Christensen, T. Zhang, Genetic algorithms for project management, Annals of Software Engineering 11 (2001) 107–139.

[11] K.W. Chau, A two-stage dynamic model on allocation of construction facilities with genetic algorithm, Automation in Construction 13 (2004) 481–490.

[12] P.-H. Chen, H. Weng, A two-phase GA model for resource-constrained project scheduling, Automation in Construction 18 (2009) 485–498.

[13] M.-Y. Cheng, H.-C. Tsai, W.-S. Hsieh, Web-based conceptual cost estimates for construction projects using evolutionary fuzzy neural inference model, Automation in Construction 18 (2009) 164–172.

[14] S.-O. Cheung, T.K.-L. Tong, C.-M. Tam, Site precast yard layout arrangement through genetic algorithm, Automation in Construction 11 (2002) 35–46.

[15] S. Christodoulou, Construction imitating ants: Resource-unconstrained scheduling with artificial ants, Automation in Construction 18 (2009) 285–293.

[16] Y.S. Dai, M. Xie, K.L. Poh, B. Yang, Optimal testing-resource allocation with genetic algorithm for modular software systems, The Journal of Systems and Software 66 (2003) 47–55.

[17] D. Datta, C.M. Fonseca, K. Deb, A multi-objective evolutionary algorithm to exploit the similarities of resource allocation problems, Journal of Scheduling 11 (2008) 405–419.

[18] N. Dayanand, R. Padman, A two stage search heuristic for scheduling payments in projects, Annals of Operations Research 102 (2001) 197–220.

[19] K. Deb, An efficient constraint handling method for genetic algorithms, Computer Methods in Applied Mechanics and Engineering 186 (2000) 311–338.

[20] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2002) 182–197.

[21] Q. Duan, T.W. Liao, Improved ant colony optimization algorithms for determining project critical paths, Automation in Construction 19 (2010) 676–693.

[22] E. Elbeltagi, T. Hegazy, D. Grierson, A modified shuffled frog-leaping optimization algorithm: applications to project management, Structure and Infrastructure Engineering 3 (1) (2007) 53–60.

[23] E. Eshtehardian, A. Afshar, R. Abbasnia, Fuzzy-based MOGA approach to stochastic time-cost tradeoff problem, Automation in Construction 18 (2009) 692–701.

[24] R. Etgar, A. Shtub, L.J. LeBlanc, Scheduling projects to maximize net present value – the case of time-dependent, contingent cash flows, European Journal of Operational Research 96 (1996) 90–96.

[25] C.-W. Feng, L. Liu, S.A. Burns, Using genetic algorithms to solve construction time-cost tradeoff problems, Journal of Computing in Civil Engineering 11 (3) (1997) 184–189.

[26] C.-W. Feng, L. Liu, S.A. Burns, Stochastic construction time-cost tradeoff analysis, Journal of Computing in Civil Engineering 14 (2) (2000) 117–126.

[27] C.-W. Feng, H.-T. Wu, Integrating fmGA and CYCLONE to optimize the schedule of dispatching RMC trucks, Automation in Construction 15 (2006) 186–199.

[28] Z.W. Geem, Multi-objective optimization of time-cost tradeoff using harmony search, Journal of Construction Engineering and Management 136 (6) (2010) 711–716.

[29] M. Gen, R. Cheng, Genetic Algorithms & Engineering Design, John Wiley & Sons, New York, 2007.

[30] M.E. Georgy, Evolutionary resource scheduler for linear projects, Automation in Construction 17 (2008) 573–583.

[31] G.E. Gibson Jr., Y.-R. Wang, C.-S. Cho, M.P. Pappas, What is preproject planning, anyway? Journal of Management in Engineering 22 (1) (2006) 35–42.

[32] A. Haidar, S. Naoum, R. Howes, J. Tah, Genetic algorithms application and testing for equipment selection, Journal of Construction Engineering and Management 125 (1) (1999) 32–38.

[33] Z. He, N. Wang, T. Jia, Y. Xu, Simulated annealing and tabu search for multi-mode project payment scheduling, European Journal of Operational Research 198 (2009) 688–696.

[34] T. Hegazy, Optimization of resource allocation and leveling using genetic algorithms, Journal of Construction Engineering and Management 125 (3) (1999) 167–175.

[35] T. Hegazy, E. Elbeltagi, EVOSITE: Evolution-based model for site layout planning, Journal of Computing in Civil Engineering 13 (3) (1999) 198–206.

[36] T. Hegazy, K. Petzold, Genetic optimization for dynamic project control, Journal of Construction Engineering and Management 129 (4) (2003) 396–404.

[37] D. Henderson, D.E. Vaughan, S.H. Jacobson, R.R. Wakefield, E.C. Sewell, Solving the shortest route cut and fill problem using simulated annealing, European Journal of Operational Research 145 (2003) 72–84.

[38] R.-Y. Huang, K.-S. Sun, A GA optimization model for workgroup-based repetitive scheduling (WoRSM), Advances in Engineering Software 40 (2009) 212–228.

[39] H. Jang, S. Lee, S. Choi, Optimization of floor-level construction material layout using genetic algorithms, Automation in Construction 16 (2007) 531–545.

[40] A. Kandil, K. El-Rayes, Parallel genetic algorithms for optimizing resource utilization in large-scale construction projects, Journal of Construction Engineering and Management 132 (5) (2006) 491–498.

[41] A. Kandil, K. El-Rayes, O. El-Anwar, Optimization research: enhancing the robustness of large scale multi-objective optimization in construction, Journal of Construction Engineering and Management 136 (1) (2010) 17–25.

[42] H. Ke, B. Liu, Project scheduling problem with stochastic activity duration times, Applied Mathematics and Computation 168 (2005) 342–353.

[43] G.H. Kim, D.S. Seo, K.I. Kang, Hybrid models of neural networks and genetic algorithms for predicting preliminary cost estimates, Journal of Computing in Civil Engineering 19 (2) (2005) 208–211.

[44] K.-C. Lam, X. Ning, M.C.-K. Lam, Conjoining MMAS to GA to solve construction site layout planning problem, Journal of Construction Engineering and Management 135 (10) (2009) 1049–1057.

[45] S.-S. Leu, T.-H. Hung, An optimal construction resource leveling scheduling simulation model, Canadian Journal of Civil Engineering 29 (2002) 267–275.

[46] S.-S. Leu, C.-H. Yang, GA-based multicriteria optimal model for construction scheduling, Journal of Construction Engineering and Management 125 (6) (1999) 420–427.

[47] S.-S. Leu, C.-H. Yang, J.-C. Huang, Resource leveling in construction by genetic algorithm-based optimization and its decision support system application, Automation in Construction 10 (2000) 27–41.

[48] H. Li, P. Love, using improved genetic algorithms to facilitate time-cost optimization, Journal of Construction Engineering and Management 123 (3) (1997) 233–237.

[49] H. Li, P.E.D. Love, Site-level facilities layout using genetic algorithms, Journal of Computing in Civil Engineering 12 (4) (1998) 227–231.

[50] H. Li, P.E.D. Love, Genetic search for solving construction site-level unequal-area facility layout problems, Automation in Construction 9 (2000) 217–226.

[51] Y.F. Li, M. Xie, T.N. Goh, A study of project selection and feature weighting for analogy based software cost estimation, The Journal of Systems and Software 82 (2009) 241–252.

[52] L.Y. Liang, W.C. Chao, The strategies of tabu search technique for facility layout optimization, Automation in Construction 17 (2008) 657–669.

[53] T.W. Liao, Two hybrid differential evolution algorithms for engineering design optimization, Applied Soft Computing 10 (2010) 1188–1199.

[54] C.-M. Lin, M. Gen, Multi-criteria human resource allocation for solving multistage combinatorial optimization problems using multiobjective hybrid genetic algorithm, Expert Systems with Applications 34 (2008) 2480–2490.

[55] L.D. Long, A. Ohsato, A genetic algorithm-based method for scheduling repetitive construction projects, Automation in Construction 18 (2009) 499–511.

[56] M. Marzouk, O. Moselhi, Constraint-based genetic algorithm for earthmoving fleet selection, Canadian Journal of Civil Engineering 30 (2003) 673–683.

[57] M. Marzouk, O. Moselhi, Multiobjective optimization of earthmoving operations, Journal of Construction Engineering and Management 130 (1) (2004) 105–113.

[58] M.J. Mawdesley, S.H. Al-jibouri, H. Yang, Genetic algorithms for construction site layout in project planning, Journal of Construction Engineering and Management 128 (5) (2002) 418–426.

[59] A. McCrea, R. Navon, Application of GA in optimal robot selection for bridge restoration, Automation in Construction 13 (2004) 803–819.

[60] C. Meier, A.A. Yassine, T.R. Browning, "Design process sequencing with competent genetic algorithms", Journal of Mechanical Design, Transactions of the ASME 129 (2007) 566–585.

[61] R. Navon, A. McCrea, Selection of an optimal construction robot using genetic algorithm, Journal of Computing in Civil Engineering 11 (3) (1997) 175–183.

[62] S.T. Ng, Y. Zhang, Optimizing construction time and cost using ant colony optimization approach, Journal of Construction Engineering and Management 134 (9) (2008) 721–728.

[63] A. Nge-The, G. Ruhe, Optimized resource allocation for software release planning, IEEE Transactions on Software Engineering 35 (1) (2009) 109–123.

[64] X. Ning, K.-C. Lam, M.C.-K. Lam, Dynamic construction site layout planning using max-min ant system, Automation in Construction 19 (2010) 55–65.

[65] M.S. Osman, M.A. Abo-Sinna, A.A. Mousa, An effective genetic algorithm approach to multiobjective resource allocation problems (MORAPs), Applied Mathematics and Computation 163 (2005) 755–768.

[66] B.K. Pathak, S. Srivastava, K. Srivastava, Neural network embedded multiobjective genetic algorithm to solve non-linear time-cost tradeoff problems of project scheduling, Journal of Scientific & Industrial Research 67 (2008) 124–131.

[67] W. Peng, C. Wang, A multi-mode resource constrained discrete time-cost tradeoff problem and its genetic algorithm based solution, International Journal of Project Management 27 (2009) 600–609.

[68] B.C. Que, Incorporating practicability into genetic algorithm-based time-cost optimization, Journal of Construction Engineering and Management 128 (2) (2002) 139–143.

[69] M. Rogalska, W. Bożejko, Z. Hejducki, Time/cost optimization using hybrid evolutionary algorithm in construction project scheduling, Automation in Construction 18 (2008) 24–31.

[70] O. Salem, A. Shahin, Y. Khalifa, Minimizing cutting wastes of reinforcement steel bars using genetic algorithms and integer programming models, Journal of Construction Engineering and Management 133 (12) (2007) 982–992.

[71] A.B. Senouci, N.N. Eldin, Use of genetic algorithms in resource scheduling of construction projects, Journal of Construction Engineering and Management 130 (6) (2004) 869–877.

[72] A.A. Shahin, O.M. Salem, Using genetic algorithms in solving the one-dimensional cutting stock problem in the construction industry, Canadian Journal of Civil Engineering 31 (2004) 321–332.

[73] A.R. Soltani, H. Tawifik, J.Y. Goulermas, T. Fernando, Path planning in construction sites: performance evaluation of the Dijkstra, A*, and GA search algorithms, Advanced Engineering Informatics 16 (2002) 291–303.

[74] J. Son, M.J. Skibniewski, Multiheuristic approach for resource leveling problem in construction engineering: Hybrid approach, Journal of Construction Engineering and Management 125 (1) (1999) 23–31.

[75] C.M. Tam, T.K.L. Tong, W.K.W. Chan, Genetic algorithm for optimizing supply locations around tower crane, Journal of Construction Engineering and Management 127 (4) (2001) 315–321.

[76] L.V. Tavares, A review of the contribution of operational research to project management, European Journal of Operational Research 136 (2002) 1–18.

[77] T.M. Williams, The contribution of mathematical modeling to the practice of project management, IMA Journal of Management Mathematics 14 (1) (2003) 3–30.

[78] C.K. Wong, I.W.H. Fung, C.M. Tam, Comparison of using mixed-integer programming and genetic algorithms for construction site facility layout planning, Journal of Construction Engineering and Management 136 (10) (2010) 1116–1128.

[79] Y. Xiong, Y. Kuang, Applying an ant colony optimization algorithm-based multiobjective approach for time-cost tradeoff, Journal of Construction Engineering and Management 134 (2) (2008) 153–156.

[80] W. Yi, A. Kumar, Ant colony optimization for disaster relief operations, Transportation Research Part E 43 (2007) 660–672.

[81] P.-Y. Yin, J.-Y. Wang, An colony optimization for the nonlinear resource allocation problem, Applied Mathematics and Computation 174 (2006) 1438–1453.

[82] P.-Y. Yin, J.-Y. Wang, A particle swarm optimization approach to the nonlinear resource allocation problem, Applied Mathematics and Computation 183 (2006) 232–242.

[83] P.-Y. Yin, J.-Y. Wang, Optimal multi-objective resource allocation using hybrid particle swarm optimization and adaptive resource bounds technique, Journal of Computational and Applied Mathematics 216 (2008) 73–86.

[84] B. Zahraie, M. Tavakolan, Stochastic time-cost-resource utilization optimization using nondominated sorting genetic algorithm and discrete fuzzy sets, Journal of Construction Engineering and Management 135 (11) (2009) 1162–1171.

[85] H. Zhang, J.Y. Wang, Particle swarm optimization for construction site unequal-area layout, Journal of Construction Engineering and Management 134 (9) (2008) 739–748.

[86] P. Zhang, F.C. Harris, P.O. Olomolaiye, G.D. Holt, Location optimization for a group of tower cranes, Journal of Construction Engineering and Management 125 (2) (1999) 115–122.

[87] D.X.M. Zheng, S.T. Ng, M.M. Kumaraswamy, Applying a genetic algorithm-based multiobjective approach for time-cost optimization, Journal of Construction Engineering and Management 130 (2) (2004) 168–176.

[88] D.X.M. Zheng, S.T. Ng, M.M. Kumaraswamy, Applying Pareto ranking and niche formation to genetic algorithm-based multiobjective time-cost optimization, Journal of Construction Engineering and Management 131 (1) (2005) 81–91.

[89] P.P. Zouein, H. Harmanani, A. Hajar, Genetic algorithm for solving site layout problem with unequal-size and constrained facilities, Journal of Computing in Civil Engineering 16 (2) (2002) 143–151.

[90] O. Zwikael, A. Sadeh, Planning effort as an effective risk management tool, Journal of Operations Management 25 (2007) 755–767.