Construction Scheduling using Non–Traditional Optimization.

A thesis submitted
in partial fullfillment of the requirements
for the Degree of
BACHELOR OF TECHNOLOGY
in
CIVIL ENGINEERING

By
Samdani Saurabh Arun
(02010422)

Department of Civil Engineering
Indian Institute of Technology Guwahati
Guwahati, India. 781039
April - 2005

Abstract

The prime objective in construction industry is to complete the planned activities of project in time. Arranging these activities in proper time sequence is a formidable task. Various methods like network based and non network based have been applied for scheduling in Construction Industry. Activities can be scheduled to comply with constrained resource usage, after the critical path has been identified. This involves determining the start time of non critical activities within the total float. The other aim is to make the resource usage as smooth as possible. Some times different time estimates are available for activities. Although lower project duration leads to higher direct costs, indirect costs are also decreased. In such a case it is important to study the trade off between completion time and cost of the project. These problems, when formulated mathematically as optimization problems, are $\mathcal{NP} - complete$. Since no solution techniques which can guarantee the optimal solution in polynomial time are available, non traditional approximation algorithms like Genetic Algorithms, Ant Colony Optimization (ACO) have to be used.

The objective of this work is to develop methods for scheduling problems in construction utilizing the power of non traditional optimization methods. As part of the project, a software to find the critical path of a network based construction schedule has been developed in C on Linux operating system. This program can takes the input as activity precedence relationships and duration of the activities; and gives the project duration and critical path as output. A software in C for optimization of a general mixed integer non linear programming problem using Ant Colony Optimization has also been developed. The constrained resource allocation problem has been solved using the ACO technique, where the objective is to minimize the total cost of the project with resource availability and precedence relationships as constraints. Results on a test problem indicate that the present model could find better solutions than reported in literature. A formulation for the multiobjective time-cost trade-off problem has also been presented. Here two objectives of project duration and direct cost are considered and the aim is to find a trade-off curve for these two objectives by choosing a combination of available resource utilization options for activities of the network. To solve this multiobjective time-cost trade-off problem, a genetic algorithm called NSGA II was used and case study on a popular test problem indicates that the method could successfully obtain a diverse set of solutions on the time-cost trade-off curve.

CERTIFICATE

It is certified that the work contained in the thesis entitled **"Construction Scheduling Using Non Traditional Optimization"** by **Samdani Saurabh Arun** , has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(Arbind Kumar Singh)
Assistant Professor
Department of Civil Engineering
Indian Institute of Technology, Guwahati

April, 2006
IIT Guwahati.

ACKNOWLEDGMENTS

I would like to express my deep gratitude and hearty thanks to my guide, **Dr. Arbind Kumar Singh**, for his guidance, encouragement and gracious support throughout the course of my under graduate study, for his expertise and insight in the field of construction scheduling and genetic algorithms, which motivated me to work in this area, and for his faith in me during most difficult stages of the research. I have benefited immensely from the long discussions I had with him on many topics in scheduling, optimization, programming and IT in construction.

I also take this opportunity to thank **Dr. Sujit Kumar Dash**, who constantly motivated me and allowed me to work from his room and use his computer. I am also thankful to other faculty members for helpful suggestions during my entire course and the Department of Civil Engineering at Indian institute of Technology Guwahati for providing all the facilities needed for this project.

I specifically thank Suresh Kataria for his help during this project. I heartily thank all my other friends for their kind cooperation during my short tenure at IIT Guwahati. Without charming company of many of them, my small endeavor could not have been so pleasurable.

Last but not the least, I give my dedicated thanks to my parents and my sister for their love and never ending support.

Contents

List of Figures

Nomenclature

N	–	set of events
A	–	set of activities
s	–	start event
t	–	terminal event
i	–	optional event
(i, j)	–	activity from event i to j
i	–	optional event
D_{ij}	–	duration of activity from event i to event j
$E(i)$	–	early event time of the i^{th} event
$L(i)$	–	late event time of the i^{th} event
$\pi(i)$	–	arbitrary event time of the i^{th} event between $E(i)$ and $L(j)$
$ES(i, j)$	–	earliest start time of activity (i, j)
$EF(i, j)$	–	earliest finish time of activity (i, j)
$LS(i, j)$	–	latest start time of activity (i, j)
$LF(i, j)$	–	latest finish time of activity (i, j)
p	–	project duration

Abbreviations

ACO	Ant Colony Optimization
GA	Genetic Algorithm
CPM	Critical path method
PERT	Program evaluation and review technique
ES	Early start time
EF	Early Finish time
LS	Late start time
LF	Late finish time
TS	Total slack
FS	Free slack
FF	Free Float
TF	Total Float
IF	Independent Float
IF	Independent Float

Chapter 1

Introduction

1.1 Introduction

Construction planning is a fundamental and challenging activity in the management and execution of construction projects. It involves the choice of technology, the definition of work tasks, the estimation of the required resources and durations for individual tasks, and the identification of any interactions among the different work tasks [26]. A good construction plan is the basis for developing the budget and the schedule for work. Developing the construction plan is a critical task in the management of construction, even if the plan is not written or otherwise formally recorded. In addition to these technical aspects of construction planning, it may also be necessary to make organizational decisions about the relationships between project participants and even which organizations to include in a project. For example, the extent to which sub-contractors will be used on a project is often determined during construction planning.

Forming a construction plan is a highly challenging task. As Sherlock Holmes noted:

> Most people, if you describe a train of events to them, will tell you what the result would be. They can put those events together in their minds, and argue from them that something will come to pass. There are few people, however, who, if you told them a result, would be able to evolve from their own inner consciousness what the steps were which led up to that result. This power is what I mean when I talk of reasoning backward [12].

Like a detective, a planner begins with a result (i.e. a facility design) and must synthesize the steps required to yield this result. Essential aspects of construction planning include the generation of required activities, analysis of the implications of these activities, and choice among the various alternative means of performing activities. In contrast to a detective discovering a single train of events, however, construction planners also face the normative problem of choosing the best among numerous alternative plans. Moreover, a detective is faced with an observable result, whereas a planner must imagine the final facility as described in the plans and specifications.

Project management is the principle of planning different projects and keeping them on track within time, cost and resource constraints. Making a schedule for construction means: making a plan with the sequence of operations and the list of resources, i.e., work force materials, amchines as they corresopnd to the project [21]. Scheduling of a construction project is challenging because:

- A construction project involves hundreds sometimes thousands of activities and the proper sequencing of so many activities demands a lot of knowledge and years of experience.
- Hundres of independent resources can be usedin a construction project and these resources are often limited both in time and size.
- Many constraints have to be satisfied e.g. deadline of major phases, the completion date of construction, budget constraints etc. TO stay within the deadline and budget is usually the most serious problem.
- Finally, the reason why it is so difficult to make a schedule is the complexity and the computational work involved!

Modern techniques of project management are made use of by managers who deal with planning scheduling and control of the projects. A well organized construction project is finished quicker and cheaper than a badly organized.

1.2 Scheduling Techniques

Scheduling techniques can be classified in many ways, traditional or non traditional and network based or non network based.

1.2.1 Traditional Scheduling Techniques

Some of these techniques are

1. Bar chart method
2. Mile stone charts

3. Linear Scheduling method.

All these techniques are non network based.

1.2.2 Non Traditional Scheduling Techniques

Following Non Traditional Scheduling Techniques are currently in use

1. Critical path method(CPM)
2. Programme evaluation and review technique(PERT)
3. Precedence Diagramming Method(PDM)

All these techniques are network based, so one can say that modern techniques are network based while traditional are non network based scheduling techniques.

1.3 Network Based Scheduling Techniques

Network Based Scheduling Techniques have evoled due to overcome the limitations traditional scheduling techniques. Two most important problems to be addressed were

1. Emphasis on logical relationships.
2. Division of planning and scheduling into two separate phases

The heart of a network based technique is always a graph, a set of nodes and connected arrows. Observing Fig. 1.1, it can be seen how clearly the graph presents the network logic. For example in Fig. 1.1 activity A has to be completed before activity B can start. Generally we say that all activities preceding a given activity must be completed before the activity in question can be started.

1.3.1 Basic Steps in Network Based Scheduling Techniques

The network based project management methodology embodies the following steps in order of appearance:

Planning phase	Step	1	Defining Activities
	Step	2	Defining Activitiy Interdepencies
	Step	3	Drawing the network
	Step	4	Time and Resource Estimation
Scheduling Phase	Step	5	Basic Calculations
	Step	6	Advanced Calculations
Control Phase	Step	7	Project Control
	Step	8	Project Review

Planning and scheduling is a dynamic process and iterative process. If results obtained are not satisfactory, one may have to go back to previous steps.

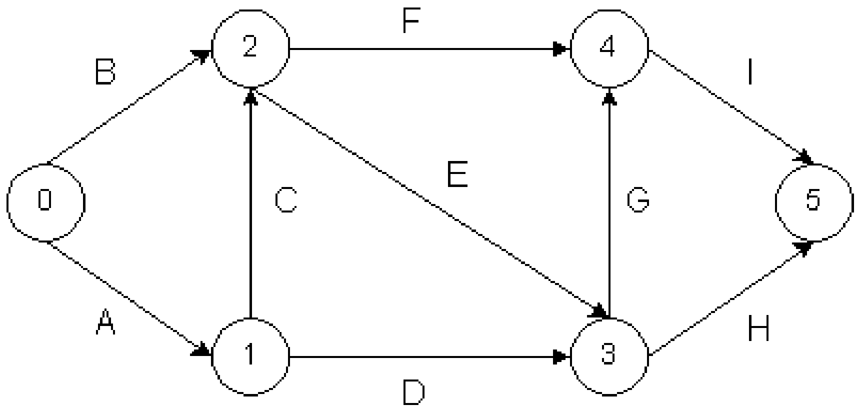

Figure 1.1: Activity-on-Branch Representation of a Nine Activity Project

1.4 Resource Constrained Scheduling

In a real project construction resources like crew sizes, equipment and materials are limited. The unlimited resources assumption in the traditional CPM technique is not feasible [31]. In this section, we shall review some general approaches to integrating both concerns in scheduling.

Two problems arise in developing a resource constrained project schedule. First, it is not necessarily the case that a critical path schedule is feasible. Because one or more resources might be needed by numerous activities, it can easily be the case that the shortest project duration identified by the critical path scheduling calculation is impossible. The difficulty arises because critical path scheduling assumes that no resource availability problems or bottlenecks will arise. Finding a feasible or possible schedule is the first problem in resource constrained scheduling. Of course, there may be a numerous possible schedules which conform with time and resource constraints. As a second problem, it is also desirable to determine schedules which have low costs or, ideally, the lowest cost [26].

Numerous heuristic methods have been suggested for resource constrained scheduling. Many begin from critical path schedules which are modified in light of the resource constraints. Others begin in the opposite fashion by introducing resource constraints and then imposing precedence constraints on the activities. Still others begin with a ranking or classification of activities into priority groups for special attention in scheduling. One type of heuristic may be better than another for different types of problems. Certainly, projects in which only an occasional resource constraint exists might be best scheduled starting from a critical path schedule. At the other extreme, projects with numerous important resource constraints might be best scheduled by considering critical resources first. A mixed approach would be to proceed simultaneously considering precedence and resource constraints.

1.4.1 Resource Levelling

The resource leveling problem arises when there are sufficient resources available and it is necessary to reduce the fluctuations in the pattern of resource usage [13]. These fluctuations are very undesirable because they often present labor, utilization, and financial difficulties for the contractor. The scheduling objective is to make the resource requirements as uniform as possible. In resource leveling there are no resource limits and the process is accomplished by shifting only the onocritical activities within their available float, the project duration of the original critical path remains fixed.

1.4.2 Resource allocation

Resource allocation is also called constrained resource scheduling. Resource allocation attempts to reschedule the project tasks so that a limited number of resources can be efficiently utilized while keeping the unavoidable extension of the project to a minimum. The variables considered in this problem are the priorities for resource allocation of the activities. These priorities in turn determine the start time of the non critical activities. Hence one can also take the activity start times of the non critical activities as the variables in such a problem. Constraint is imposed on the available resource of each type. The objective is to find the minimum project duration.

1.4.3 Current approaches to resource scheduling

An approach to resource allocation with integer linear programming was presented in [28]. It was assumed that some resources are limited and when the requirement is exceeded then those resources have to be borrowed / leased with associated leasing cost. A predetermined pattern of resource usage for each activity was used to find the resource deficit. The objective function used is minimization of total leasing cost over the project duration subject to the network logic. However resource leveling was not attempted. The model has the capability of mapping continuous, intermittent, uniform and non uniform activities. The variables considered were start and stop times of each activity. The model can be used to help plan the preliminary resource requirements and make lease versus buy decisions. All integer linear programming problems are $\mathcal{NP}$ – Complete [2, 6], so there is no guarantee that this model will guarantee optimal solution. The number of variables are very large due to the binary mapping of each day during the total float, hence the model may not be very useful for large projects.

A integer linear programming model for resource leveling was proposed in [13] for small to medium sized projects. Easa (1989) has assumed that resource availability is unlimited, and the total amount of resource consumption due to an activity is known. The variables considered are the starting time of non critical activities subject to the limits of earliest start time and latest start time. Once the start times of all the activities are known the resource usage pattern can be determined. Required (or uniform) resource usage rate of each resource is calculated as total resource consumption divided by the project duration. The objective function is to minimize the cumulative difference resource usage from the uniform resource usage for the project duration. The model was implemented for two sample problems with good results, however the earlier qualifications for integer linear programming still apply to this model.

An integrated approach to resource leveling and constrained resource scheduling was presented in [5]. Chan et al. (1996) has considered the ordering of scheduling activities as variables. Based upon the ordering of activities, start times are obtained and used to find the resource usage pattern. The objective is to minimize the weighted sum of the deviation of resource usage from availability for each resource from start to end of the project. A penalty is imposed upon the objective function if the usage of any resource crosses a predetermined resource availability. A schedule builder was designed which can take care of network logic while "constructing" the solution. The solution technique used is Genetic Algorithms [20], a biologically inspired optimization technique. Encouraging performance was observed on small as well as large projects. However, Genetic algorithms *may* not converge to the optimal solution quickly and one does not know whether the "best" found solution is an optimal solution. But GAs have been found always to give *good* solutions.

Hegazy (1999) [22] presented a different approach to simultaneous resource leveling and constrained resource scheduling using Genetic Algorithms. The methodology was implemented in a commercial software, Microsoft Project [39], thereby making it easier for practitioners to adopt the technique. This technique makes use of the "Activity Priority" feature of MS Project. After a set of priorities are assigned for all the activities, the software calculates the resource usage pattern. Two objective functions were suggested:

1. Moment of the resource histogram about X (time) axis, M_x.
2. Moment of the resource histogram about Y (Resource) axis, M_y.

Objective 1 can be used when the focus is on reducing project resource fluctuations, while objective 2 can be used when the focus is on reducing the resource utilization period. The sum of both the objectives can be used when the focus is on both aspects. Each gene in the Genetic Algorithm chromosome represents the activity priority (from Very High to Lowest). A macro program in MS Project was coded. Although "Multiobjective search using Genetic Algorithms" is discussed, in reality only one objective was used in implementation . It is not clear how the resource constraints are incorporated, since it is mentioned only during introduction but not in the formulation. Hegazy et al.(2000) [25] presented an algorithm for scheduling using multi skilled constrained resources, where resource constraints for multiple and multi skilled resources are taken into account. Hegazy and Ersahin(2001) [23] presented the spread sheet model for integrating critical-path network scheduling with time-cost trade-off analysis, resource allocation, resource levelling and cash flow management. The model uses the total project cost as the objective function to be minimized. To facilitate this large-size optimization, a nontraditional optimization technique genetic algorithms is used to locate the globally optimal solution.
Hegazy and Kassab (2003) [24] presented a new approach for resource optimization by combining a flow-chart based simulation tool with a powerful genetic optimization procedure. The proposed approach determines the least costly, and most productive, amount of resources that achieve the highest benefit/cost ratio in individual construction operations. To further incorporate resource optimization into construction planning, various genetic algorithms GA-optimized simulation models are integrated with commonly used project management software. Accordingly, these models are activated from within the scheduling software to optimize the plan. The result is a hierarchical work-breakdown-structure tied to GA-optimized simulation models. It was proposed that computer simulation and genetic algorithms can be an effective combination with great potential for improving productivity and saving construction time and cost.

Mattila and Abraham (1998) [36] presented a method to level the resources of a typical highway construction project that was scheduled using the linear scheduling method. The integer linear programming resource leveling formulation presented uses the concepts of rate float and activity float. The objective was to find a set of start times of activity to minimize the deviations of resource usage above and below the desired usage rate. There is no limit on resource usage. In an extension to this model, Mattila and Park (2003) [37] proposed linear scheduling model and the repetitive scheduling method. This paper discusses basic linear scheduling techniques and then calculates critical activities of basic linear scheduling elements using two methods. The results of two techniques are then compared.

El-Rayes and Moselhi (2001) [15] has developed an automated model for optimizing resource utilization for repetitive construction projects. The model is based on a dynamic programming formulation, designed to identify an optimum crew formation and interruption option for each activity in the project that leads to minimum project duration. The model incorporates a scheduling algorithm and an interruption algorithm. The scheduling algorithm complies with job logic, crew availability, and crew work continuity constraints. The interruption algorithm generates a set of feasible interruption vectors for each crew formation to be used as a second-state variable in the dynamic programming formulation. A project example from the literature was analyzed in order to demonstrate the use of the model and illustrate its capabilities. Extending this methodology, Moselhi and Hassanein (2003). [38] presented a model, designed to optimize scheduling of linear projects. The model employs a two-state-variable, N-stage,dynamic programming formulation, coupled with a set of heuristic rules. The model is resource-driven, and incorporates both repetitive and non repetitive activities in the optimization process to generate practical and near-optimal schedules. The model is implemented in a prototype software that operates in Windows environment. It is developed utilizing object-oriented programming and provides for automated data entry. Several graphical and tabular output reports are be generated.

Elazouni and Gab-Allah(2004) [16] introduced an integer-programming finance based scheduling method to produce financially feasible schedules that balance the financing requirements of activities at any period with the cash available during that same period. The proposed method offers two fold benefits of minimizing total project duration and fulfilling finance availability constraints. This model was extended for use with Genetic Algorithms in [17] with considerable improvements.

1.5 Time cost trade off

The previous sections discussed the duration of activities as either fixed or random numbers with known characteristics. However, activity durations can often vary depending upon the type and amount of resources that are applied [26]. Assigning more workers to a particular activity will normally result in a shorter duration. Greater speed may result in higher costs and lower quality, however. In this section, we shall consider the impacts of time, cost and quality tradeoffs in activity durations. In this process, we shall discuss the procedure of project crashing as described below.

A simple representation of the possible relationship between the duration of an activity and its direct costs appears in Fig. 1.2. Considering only this activity in isolation and without reference to the project completion deadline, a manager would undoubtedly choose a duration which implies minimum direct cost, represented by D_{ij} and C_{ij} in the figure. Unfortunately, if each activity was scheduled for the duration that resulted in the minimum direct cost in this way, the time to complete the entire project might be too long and substantial penalties associated with the late project start-up might be incurred. This is a small example of sub-optimization, in which a small component of a project is optimized or improved to the detriment of the entire project performance. Avoiding this problem of sub-optimization is a fundamental concern of project managers.

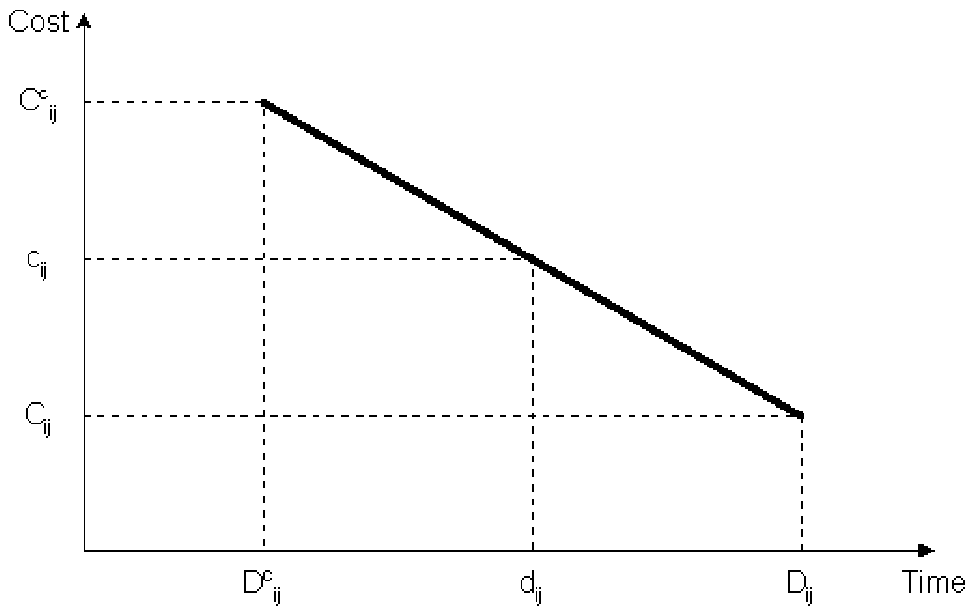

Figure 1.2: Illustration of a Linear Time/Cost Tradeoff for an Activity (from [26])

At the other extreme, a manager might choose to complete the activity in the minimum possible time, D^c_{ij}, but at a higher cost C^c_{ij}. This minimum completion time is commonly called the activity crash time. The linear relationship shown in the figure between these two points implies that any intermediate duration could also be chosen. It is possible that some intermediate point may represent the ideal or optimal trade-off between time and cost for this activity.

What is the reason for an increase in direct cost as the activity duration is reduced? A simple case arises in the use of overtime work. By scheduling weekend or evening work, the completion time for an activity as measured in calendar days will be reduced. However, premium wages must be paid for such overtime work, so the cost will increase. Also, overtime work is more prone to accidents and quality problems that must be corrected, so indirect costs may also increase. More generally, we might not expect a linear relationship between duration and direct cost, but some convex function such as the nonlinear curve or the step function shown in Fig. 1.3. A linear function may be a good approximation to the actual curve, however, and results in considerable analytical simplicity

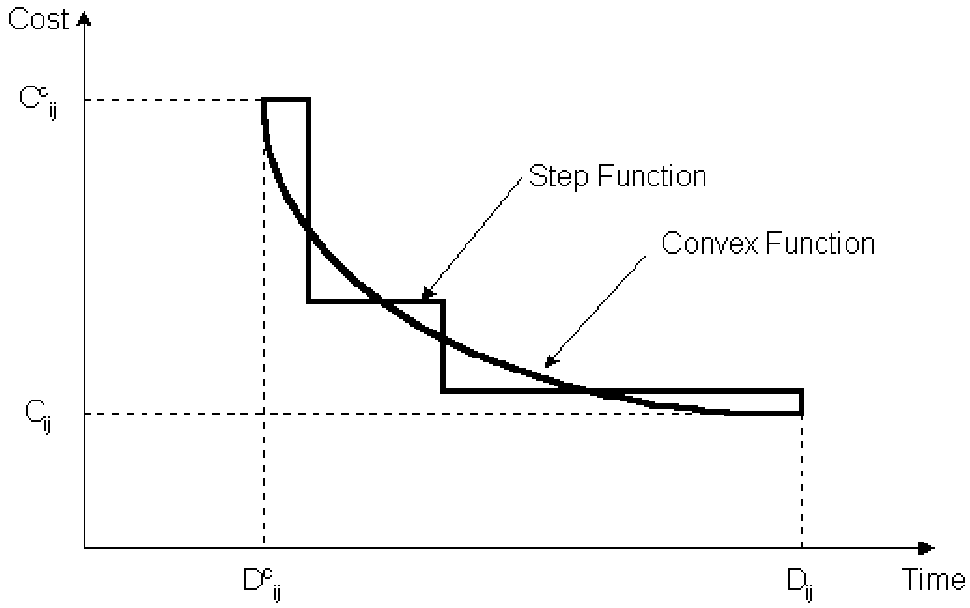

Figure 1.3: Illustration of Non-linear Time/Cost Tradeoffs for an Activity (from [26])

A time cost optimization model using integer programming / linear programming (LP/IP) hybrid was developed and presented in [35]. It was assumed that an activity can be completed in various durations with associated cost. Duration of various activities are assumed to be discrete for a project . Hence the time cost relationship for an activity is discrete. In the linear programming formulation, the objective function is the total cost which is the sum of direct cost and indirect cost. The variables in LP formulation are continuous. Indirect cost is taken function of the duration. The constraints imposed are the network logic and constraints from the convex hull of the time cost relationship of activity. The linear programming formulation gives lowerbound to the time cost trade off, while the exact solution is found out using integer programming which can handle the discrete time cost relationships. The problem was solved using solver in Microsoft Excel. The previous LP/IP model was extended in [18] The objective is to know what choice of time-cost alternative for each activity would produce the solution with minimum total cost. In the first stage, time cost trade off surface is generated using genetic algorithms.

The direct cost is considered as one objective while the project duration as other. The fitness of each individual is calculated by finding the distance of the solution from the convex hull. The closer a solution to the convex hull, higher the fitness. In the next stage, minimum total cost solution is found out by adding the direct cost and indirect cost (which is proportional to the duration). A 18 activity network was used to demonstrate the software developed in Microsoft Excel.

Li and Love (1997) [33] formulated a different type of time cost trade off problem which occurs when the project is running behind schedule. To meet the deadline, the project duration should be reduced to a certain limit. It was assumed that all the activities are not crashable. The objective is to find which activities should be crashed and by what duration to shorten the total duration to the targeted limit with minimum compressing cost. Each gene represents the crash duration of an activity in the chromosome coding. The proposed method was explained for a network with 10 crashable activities. Li et al. [32] also presented a machine learning and genetic algorith to solve this problem.

A Genetic Algorithms based multicriteria scheduling model which integrates the time/cost trade-off model, resource leveling and resource allocation model was proposed in [31]. The model is implemented in two phases. The GA crashing engine subsystem at Phase 1 manipulates project time/cost trade-off variables. In the time/cost trade-off model, project costs are a function of project durations, whose lengths are dependent on resource availabilities. Correspondingly, the resource-constrained model needs activity durations from the time/cost trade-off model as basic input data for the computation of minimum project duration under resource constraints. Therefore, these two models interact with each other at Phase 1. In the final step, both results (i.e., project costs and durations) are evaluated by technique for order preference by similarity to ideal solution. The survivors are fed back to the GA-based time/cost trade-off algorithm and the GA based resource-constrained algorithm for the next generation. Phase 2 focuses on resource leveling. In this paper, resource leveling is a subsidiary to Phase 1 of the integrated model. The GA-based resource leveling system receives information about the optimal schedule and related resource requirements from Phase 1, and then adjusts activity starting dates to obtain a more even resource profile. The model is very versatile and can successfully integrate resource scheduling and time cost trade off. The final output is the duration and starting date of each activity.

Que (2002) [40] presented an approach that makes GA-based time-cost optimization viable for practical application by integrating a project management system to the GA system and taking full advantage of its powerful scheduling functionality. The approach ensures that all scheduling parameters, including activity relationships, lags, calendars, constraints, resources, and progress, are considered in determining the project completion date corresponding to a solution, allowing more comprehensive and realistic evaluations during optimization. Since the benefits of the approach are qualitative rather than quantitative, experimental results were not included.

Leu et al. (1999) [30] presented a fuzzy optimal model for resource-constrained construction scheduling. The model took into consideration both uncertain activity duration and resource constraints. A genetic algorithm-based technique was used to search for optimal fuzzy profiles of project duration and resource amounts under the constraint of limited resources. However, the method did not attempt to seek project total cost minimization. Senouci and Adeli (2001)[42] presented a mathematical model for resource scheduling, which handled resource-constrained scheduling, resource leveling, and project total cost minimization simultaneously. The patented neural dynamics model of Adeli and Park (1998) [1] was used to solve the optimization model. However, the model dealt with continuous variables only and did not consider the case of discrete variables. A model which considers all precedence relationships, multiple crew strategies, total project cost minimization, and time-cost trade-off was proposed in [43]. In the new formulation, resource leveling and resource-constrained scheduling are performed simultaneously. The model presented uses the quadratic penalty function to transform the resource-scheduling problem to an unconstrained one. The objective function used is the total cost minimization subject to network precedence constraints and resource availability limits. All type of precedence relationships like finish-start, start-finish, start-start, finish-finish can be used. The model uses Genetic Algorithms as the optimization technique to drive the search towards good solutions. The effectiveness was demonstrated on a 12 activity network.

Zheng et al. (2004) [44] presented a novel multiobjective approach that aims to optimize total time and total cost simultaneously. GAs concepts and tools. The model introduces a MAWA (modified adaptive weight approach) to replace traditional fixed or random weights, and integrates time and total cost into a single objective for simulation. This approach imparts the GAs with greater freedom to search in the multiobjective space that overcomes the drawbacks of single objective TCO, i.e., a local optimum in HCA (hill-climbing algorithms), and the previously proposed multiobjective approach developed by Gen and Cheng (2000)[19]. The main focus of this paper was on improving the solution technique used by earlier researchers. The model was similar to the one adopted in [35]. A similar attempt was made by Zheng et al. (2005) [45] to improve the Genetic Algorithm solution procedure using pareto ranking and niche formation.

A multiobjective optimization model was developed to transform the traditional two-dimensional time-cost trade-off analysis to an advanced three-dimensional time-cost-quality trade-off analysis in [27, 14]. The model was designed to search for optimal resource utilization plans that minimize construction time and cost while maximizing its quality. The optimization model is developed in three main stages: 1. Model formulation stage that incorporates all major decision variables and optimization objectives; 2. quality quantification stage that formulates new functions to enable the consideration of construction quality in this optimization problem; and 3. model implementation stage that implements a multiobjective GA for highway construction to enable the simultaneous optimization of construction time, cost, and quality. An application example is analyzed to illustrate the use of the model and demonstrate its capabilities in considering quality in the optimization process and in developing optimal trade-offs among construction time, cost, and quality. These new capabilities should prove useful to decision makers in highway construction and rehabilitation projects, especially those who are involved in new types of contracts that demand high-quality performance.

References on Time Cost trade-off and Construction Resource Scheduling

Time-cost Optimization	[34], [26], [35], [18], [33], [32], [31], [40], [30], [42], [1], [43], [44], [19], [35], [45], [27, 14],
Resource Scheduling	[31], [26], [13], [28], [2, 6], [13], [5], [20], [22], [39], [25], [23], [24], [36], [37], [15], [38], [16], [17],

1.6 Summary and Layout of the Report

In this introductory chapter, time cost trade off and resource scheduling problems in construction scheduling were introduced. These problems when formulated mathematically as optimization problems are $\mathcal{NP} - complete$ since they are all mixed integer non linear programming problems [6]. As no polynomial time algorithms to solve such problems are available [6, 2], approximation algorithms like meta heuristics have to be used. In the next chapter a discussion on one such method called Ant Colony Optimization is taken up. Many other non traditional methods are available like genetic algoritihms [20], simulated annealing [7], particle swarm optimization [29], evolutionary strategies [41]. A discussion on these techniques can be found in popular textbooks [20, 7, 41, 29]. A formulation of multiobjective time-cost trade-off problem is present in Chapter 4 and solution to the problem is attempted using non-dominated sorting genetic algorithm. Chapter 3 presents the formulation for constrained resouce allocation and leveling problem, which is solved with Ant colony optimization Finally a summary of the report and future plans are discussed in Chapter 5.

Chapter 2

Ant Colony Optimization

2.1 Introduction

In the early 90's, ant colony optimization (ACO) [10, 9] emerged as a novel nature-inspired metaheuristic method for the solution of combinatorial optimization problems. The inspiring source of ACO is the foraging behavior of real ants. When searching for food, ants initially explore the area surrounding their nest in a random manner. As soon as an ant finds a food source, it evaluates quantity and quality of the food and carries some of the food found to the nest. During the return trip, the ant deposits a chemical pheromone trail on the ground. The quantity of pheromone deposited, which may depend on the quantity and quality of the food, will guide other ants to the food source. The indirect communication between the ants via the pheromone trails allows them to find shortest paths between their nest and food sources. This behaviour of real ant colonies is exploited in artificial ant colonies in order to solve discrete optimization problems.

2.2 Real Ants Behaviour

Ants are social insects, that is, insects that live in colonies and whose behavior is directed more to the survival of the colony as a whole than to that of a single individual component of the colony. Social insects have captured the attention of many scientists because of the high structuration level their colonies can achieve, especially when compared to the relative simplicity of the colony's individuals. An important and interesting behavior of ant colonies is their foraging behavior, and, in particular, how ants can find the shortest paths between food sources and their nest.

Although individual ants move in a quasi-random fashion, performing relatively simple tasks, the entire colony of ants can collectively accomplish sophisticated movement patterns and can find the shortest route between their nest and a food source . Ants accomplish this by depositing a substance called *pheromone* as they move. This chemical trail can be detected by other ants, which are probabilistically more likely to follow a path rich in pheromone. This chemical trail can be detected by other ants, which are probabilistically more likely to follow a path rich in pheromone.To show how trail information can be utilized to adapt to sudden unexpected changes in the terrain, a brief example is given below (see Fig.2.1). In Fig.(2.1a) a colony of ants is traveling in both directions between point A and point B. Each ant knows which direction to take because of the pheromone trail that is present from point A to point E Fig. (2.1 a). Fig. (2.1 b)show what happens when an object is placed in the middle of the path. Since the object is not placed symmetrically on the trail, the path B-C-D is shorter than the path B-F-G-H-D. The ants moving from point B to point D and vice versa ,will have to make a decision whether to turn left or right. Since there is no pheromone in either direction, the ant has an equal probability of turning right or left. Initially the first ants turn left or right equally, which means that the equal number of ants are taking path B-C-D and path B-F-G-H-D . The ant traveling along any path will leave pheromone along it. The ants traveling the B-C-D path will arrive at point D earlier than the ones traveling along path B-F-G-H-D . An ant traveling in the opposite direction and is at point D will detect more pheromone along the D-C path, since not only are ants going from D to C, but ants also have started to arrive from C to D. On the other hand the path B-F-G-H-D being longer , the ants have not yet arriving on path D-H. The exact same thing is occurring at point B. Therefore due to more pheromone deposition , probabilistically more ants will begin taking path D-C-B . Eventually the pheromone level on the D-C-B path will become so dominant that all of the ants will choose this path as in Fig. (2.1 c). This will also hold true for the ants traveling from point B to D. Hence the ants, using their highly effective pheromone based communication method, are able to find the shortest path between point B and D.

This particular behaviour of ant colonies has inspired the Ant Colony Optimization meta-heuristic algorithm, in which a set of artificial ants co-operate to find solutions to a given optimization problem by depositing pheromone trails throughout the search space.

2.3 Ant colony optimization

ACO algorithms are metaheuristic methods for tackling combinatorial optimization problems [11]. The central component of an ACO algorithm is the pheromone model, which is used to probabilistically sample the search space. As outlined in [3], the pheromone model can be derived from a model of the CO problem under consideration. A model of a CO problem can be stated as follows.

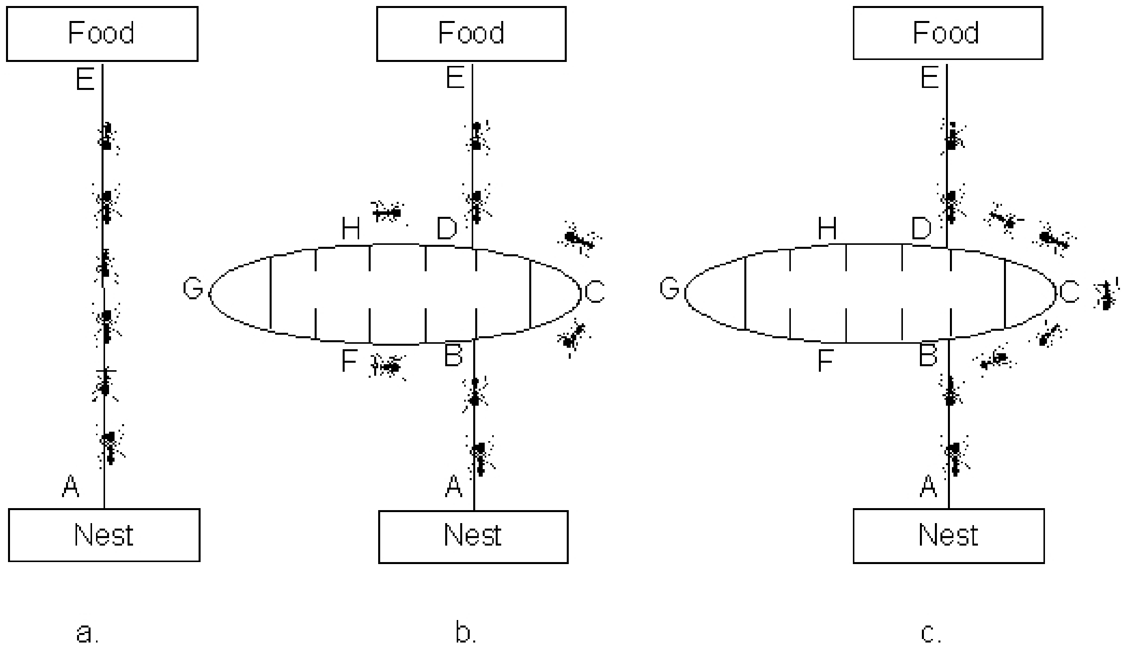

Figure 2.1: Biological ants find shortest path by using pheromone based communication.

Definition 1 *A model $P = (S, \Omega, f)$ of a CO problem consists of:*

- *a* ***search (or solution) space*** *$\mathcal{S}$ defined over a finite set of discrete decision variables and a set Ω* ***constraints*** *among the variables;*
- *an* ***objective function*** *$f : \mathcal{S} \rightarrow \mathcal{R}^+$ to be minimized.*

The search space $\mathcal{S}$ is defined as follows: Given is a set of n ***discrete variables*** *$\mathcal{X}_i$ with domains $D_i = v_i^1, \cdots, v_i^{|Di|}, i = 1, \cdots, n$. A variable instantiation, that is, the assignment of a value v_i^j to a variable $\mathcal{X}_i$, is denoted by $\mathcal{X}_i = v_i^j$. A feasible solution $s \in \mathcal{S}$ is a complete assignment (i.e., an assignment in which each decision variable has a domain value assigned) that satisfies the constraints. If the set of constraints Ω is empty, then each decision variable can take any value from its domain independently of the values of the other decision variables. In this case we call $\mathcal{P}$ an* ***unconstrained*** *problem model, otherwise a* ***constrained*** *problem model. A feasible solution $s^* \in \mathcal{S}$ is called a* ***globally optimal solution****, if $f(s^*) \leq f(s) \;\; \forall s \in \mathcal{S}$ The set of globally optimal solutions is denoted by $\mathcal{S}^* \subseteq \mathcal{S}$ To solve a CO problem one has to find a solution $s \in \mathcal{S}^*$*

A model of the CO problem under consideration implies the finite set of solution components and the pheromone model as follows [3]. First, we call the combination of a decision variable $\mathcal{X}_i$ and one of its domain values v_i^j a *solution component* denoted by $\mathfrak{c}_i^j$. Then, the pheromone model consists of a *pheromone* trail parameter $\mathcal{T}_i^j$ for every solution component $\mathfrak{c}_i^j$. The value of a *pheromone* trail parameter $\mathcal{T}_i^j$ called pheromone value – is denoted by τ_i^j. The set of all pheromone trail parameters is denoted by $\mathcal{T}$. As a CO problem can be modelled in different ways, different models of the CO problem can be used to define different pheromone models. Alg. 1 captures the framework of a basic ACO algorithm, as outlined in [3]. It works as follows. At each iteration, n_a ants probabilistically construct solutions to the combinatorial optimization problem under consideration, exploiting a given pheromone model. Then, optionally, a local search procedure is applied to the constructed solutions. Finally, before the next iteration starts, some of the solutions are used for performing a pheromone update. The details of this framework [3] are explained in more detail in the following.

InitializePheromoneValues($\mathcal{T}$). At the start of the algorithm the pheromone values are all initialized to a constant value $c > 0$.

```
Algorithm:ACO procedure
Input: An instance of the problem P of a CO problem model P = (S, f, Ω)
InitializePheromoneValues(T) ;
s_bs ← NULL ;
while Termination criteria not true do
    S_iter ← φ ;
    for each ant k do
        s ← ConstructSolution(T) ;
        s ← LocalSearch(s) ;
        if (f(s) < f(s_bs)) or (s_bs == NULL ) then
            s_bs = s_bs
        end
        S_iter ← S_iter ⋃{s};
    end
    ApplyPheromoneUpdate(T, S_iter, s_bs) ;
end
Output: The best– so– far solution s_bs
```

Algorithm 1: Framework of a basic Ant Colony Optimization Algorithm

ConstructSolution($\mathcal{T}$). The basic ingredient of any ACO algorithm is a constructive heuristic for probabilistically constructing solutions. A constructive heuristic assembles solutions as sequences of elements from the finite set of solution components $\mathfrak{C}$. A solution construction starts with an empty partial solution $\mathfrak{s}_p = \langle\rangle$. Then, at each construction step the current partial solution $\mathfrak{s}_p$. is extended by adding a feasible solution component from the set $\mathfrak{N}(s_p) \subseteq \mathfrak{C} \setminus \mathfrak{s}_p$, which is defined by the solution construction mechanism. The process of constructing solutions can be regarded as a walk (or a path) on the so-called construction graph $\mathcal{G}_C = (\mathfrak{C}, \mathfrak{L})$, which is a fully connected graph whose vertices are the solution components $\mathfrak{C}$ and the set $\mathfrak{L}$ are the connections. The allowed walks on $\mathcal{G}_C$ are implicitly defined by the solution construction mechanism that defines the set $\mathfrak{N}(s_p)$ with respect to a partial solution $\mathfrak{s}_p$. We denote the set of all solution that may be constructed in this way by $\mathfrak{S}$. The choice of a solution component from $\mathfrak{N}(s_p)$ is, at each construction step, done probabilistically. In most ACO algorithms the probabilities for choosing the next solution component – also called the *transition probabilities* – are defined as follows [3]:

$$p(\mathfrak{c}_i^j|\mathfrak{s}_p) = \frac{\tau_i^{j^\alpha}\eta(\mathfrak{c}_i^j)^\beta}{\sum_{\mathfrak{c}_k^l \in \mathfrak{N}(s_p)} \tau_i^{l^\alpha}\eta(\mathfrak{c}_i^l)^\beta} \quad \forall \mathfrak{c}_k^l \in \mathfrak{N}(s_p) \tag{2.3.1}$$

where η weighting function that assigns, at each construction step, a heuristic value $\eta(\mathfrak{c}_i^j)$ to each feasible solution component $(\mathfrak{c}_i^j)$. The values that are given by the weighting function are commonly called the *heuristic information*. α and β are positive parameters whose values determine the relative importance of pheromone and heuristic information.

ApplyPheromoneUpdate($\mathcal{T}, \mathfrak{S}_{iter}, \mathfrak{s}_{bs}$) Most ACO algorithms use the following pheromone value update rule:

$$\tau_i^j \leftarrow (1-\rho)\tau_i^j + \rho \sum_{\{\mathfrak{s}\in\mathfrak{S}_{upd}|\mathfrak{c}_i^j\in\mathfrak{s}\}} F(\mathfrak{s}), \tag{2.3.2}$$

for $i = 1, \cdots, n$, and $j \in \{1, \cdots, |\mathcal{D}_i|\}$. $\rho \in (0, 1]$ is a parameter called evaporation rate. $F : \mathfrak{S} \mapsto R^+$ is a function such that $f(\mathfrak{s}) < f(\mathfrak{s}\prime) \Rightarrow F(\mathfrak{s}) \geq F(\mathfrak{s}\prime), \quad \forall \mathfrak{s} \neq \mathfrak{s}\prime \in \mathfrak{S}$. $F(\mathfrak{s})$ is commonly called the quality function. Instantiations of this update rule are obtained by different specifications of $\mathfrak{S}_{upd}$, which is a subset of $\mathfrak{S}_{iter} \bigcup \{\mathfrak{s}_{bs}\}$, where $\mathfrak{S}_{iter}$ is the set of solutions that were constructed in the current iteration, and $\{\mathfrak{s}_{bs}\}$ is the best-so-far solution. A well-known example of update rule Eq.2.3.2 is the AS-update rule [9] (i.e., the update rule of Ant System (AS)) which is obtained from Eq.2.3.2 by setting

$\mathfrak{S}_{upd} \leftarrow \mathfrak{S}_{iter}$ The goal of the pheromone value update rule is to increase the pheromone values on solution components that have been found in high quality solutions. In AS the quality function $F(\mathfrak{s})$ is defined as follows,

$$F(\mathfrak{s}) = \frac{Q}{f(\mathfrak{s})}, \quad \forall \mathfrak{s} \quad \in \quad \mathfrak{S}_{iter} \tag{2.3.3}$$

where Q is a constant. In [9] elitist strategy for trail update was also suggested as per Eq.2.3.4.

$$F(\mathfrak{s}_{bs}) = n_e.\frac{Q}{f(\mathfrak{s}_{bs})} \tag{2.3.4}$$

where $\mathfrak{s}_{bs}$ is the the best–so–far solution and n_e is the number of elitist ants, a parameter of the algorithm.

Another method for trail update was introduced in [4] and called as Rank-based Ant system (AS_{rank}) In the AS_{rank}, the solutions created by the ants are ranked according to how well they solve the problem. Let μ_i denote the rank of the solution of ant i. Then $\mathfrak{s}_{iter}$ (the best solution in the current iteration) will have rank 1. $\mathfrak{S}_{upd}$ is defined as [4],

$$\mathfrak{S}_{upd} = \mathfrak{S}_{rank} \bigcup \{\mathfrak{s}_{bs}\}, \tag{2.3.5}$$

where $\mathfrak{S}_{rank}$ is defined by Eq.2.3.6 as

$$\mathfrak{S}_{rank} = \ \{ \ \mathfrak{s}_i \mid \mu_i < \lambda\} \tag{2.3.6}$$

where λ denotes the number of top ranked ants used for trail update.

In AS_{rank} , the quality function $F(\mathfrak{s})$ is defined as follows,

$$F(\mathfrak{s}_i) = Q.\frac{\lambda - \mu_i}{f(\mathfrak{s}_i)}, \quad \forall \mathfrak{S}_i \in \mathfrak{S}_{rank} \tag{2.3.7}$$

where $\mathfrak{S}_i$ has rank λ_i.

2.4 Conclusions

In this chapter, underlying biological foundations of ACO metaphor were explained and framework for solution of a problem was discussed. A software in the C programming language for solution of such a generic problem has been prepared and tested. The ACO metaphor would now be used for solution of Constrained Resource Scheduling in next chapter.

Chapter 3

Constrained Resource Scheduling using ACO

The resource constrained scheduling problem was introduced in Sec. 1.4. In this chapter a mathematical formulation of the problem is presented with variables as the duration and the starting time of each activity [43]. Cost and resource distributions as functions of activity duration are assumed to be known for each activity. The objective is then to determine the starting time and duration of each activity to optimize the total project cost subject to limited resource constraint.

3.1 Problem formulation

Consider a network with n_a activities. Any precedence relationship between the activities is permissible (finish-finish, start-start, finish-start, start-finish). We will use the following notation

- $\mathfrak{d}_i$: duration of activity i; $\mathfrak{d}_i^{min} \leq \mathfrak{d}_i \leq \mathfrak{d}_i^{min}$, $\forall i = 1, 2, \ldots, n_a$.
- $\mathfrak{s}_i$: start time of activity i;
- $\mathfrak{c}_i$: direct cost of of activity i for duration $\mathfrak{d}_i$; $i = 1, 2, \ldots, n_a$.
- $\mathfrak{l}_{ij}$: lag/ lead time between activities i and j.
- $\mathcal{S}_i$: set of activities succeding activity i
- $\mathcal{C}_d$: direct project cost; $\mathcal{C}_d = \sum_i^{n_a} \mathfrak{c}_i(\mathfrak{d}_i)$
- $\mathcal{C}_i$: indirect project cost
- $\mathcal{C}_t$: total project cost; $\mathcal{C}_t = \mathcal{C}_d + \mathcal{C}_i$

- $\mathcal{S}_t$: set of activities in progress at time t
- r_{ki}: daily requirement of kth resource for activity i.
- R_{kt}: maximum availability of kth resource at time t.

For each activity, a cost distribution curve could be constructed from the allocated resources. Depending on the type of activity, such curves could be either continuous or discrete. They could also be linear or nonlinear as shown in Figs.3.1 (a and b), respectively. Similarly, curves describing resource profiles (resource-duration) can be constructed as shown in Figs. 3.2 (a and b). In these two figures, the Y-axis denotes a type k resource as required by activity i for duration of $\mathfrak{d}_i$. In this study, continuous linear and nonlinear cost - duration curves and resource-duration curves are considered.

Since indirect costs are time dependent, the relationship between $\mathcal{C}_i$ and the project duration $\mathcal{D}$ can be expressed as

$$\mathcal{C}_i = \mathcal{C}_0 + b\mathcal{D} \tag{3.1.1}$$

where $\mathcal{C}_0$ are intial costs for the project and b is the slope of the cost distribution function (i.e. daily expenditures) Consider $X = \{\mathfrak{s}_1, \mathfrak{s}_2, \ldots, \mathfrak{s}_{n_a}, \mathfrak{d}_1, \mathfrak{d}_2, \ldots, \mathfrak{d}_{n_a}, \}$ as the vector of decision variables. Now the resource constrained scheduling problem can be written as

$$\text{minimize} \qquad \mathcal{C}_t(X) \tag{3.1.2}$$

subject to:
Precedence constraints

- Finish to start (FS)

$$\mathfrak{s}_i + \mathfrak{d}_i + \mathfrak{l}_{ij} \leq \mathfrak{s}_j \qquad \forall j \in \mathcal{S}_i \tag{3.1.3}$$

- Start to start (SS)

$$\mathfrak{s}_i + \mathfrak{l}_{ij} \leq \mathfrak{s}_j \qquad \forall j \in \mathcal{S}_i \tag{3.1.4}$$

- Start to Finish (SF)

$$\mathfrak{s}_i + \mathfrak{l}_{ij} \leq \mathfrak{s}_j + \mathfrak{d}_j \qquad \forall j \in \mathcal{S}_i \tag{3.1.5}$$

- Finish to Finish (FF)

$$\mathfrak{s}_i + \mathfrak{d}_i + \mathfrak{l}_{ij} \leq \mathfrak{s}_j + \mathfrak{d}_j \qquad \forall j \in \mathcal{S}_i \tag{3.1.6}$$

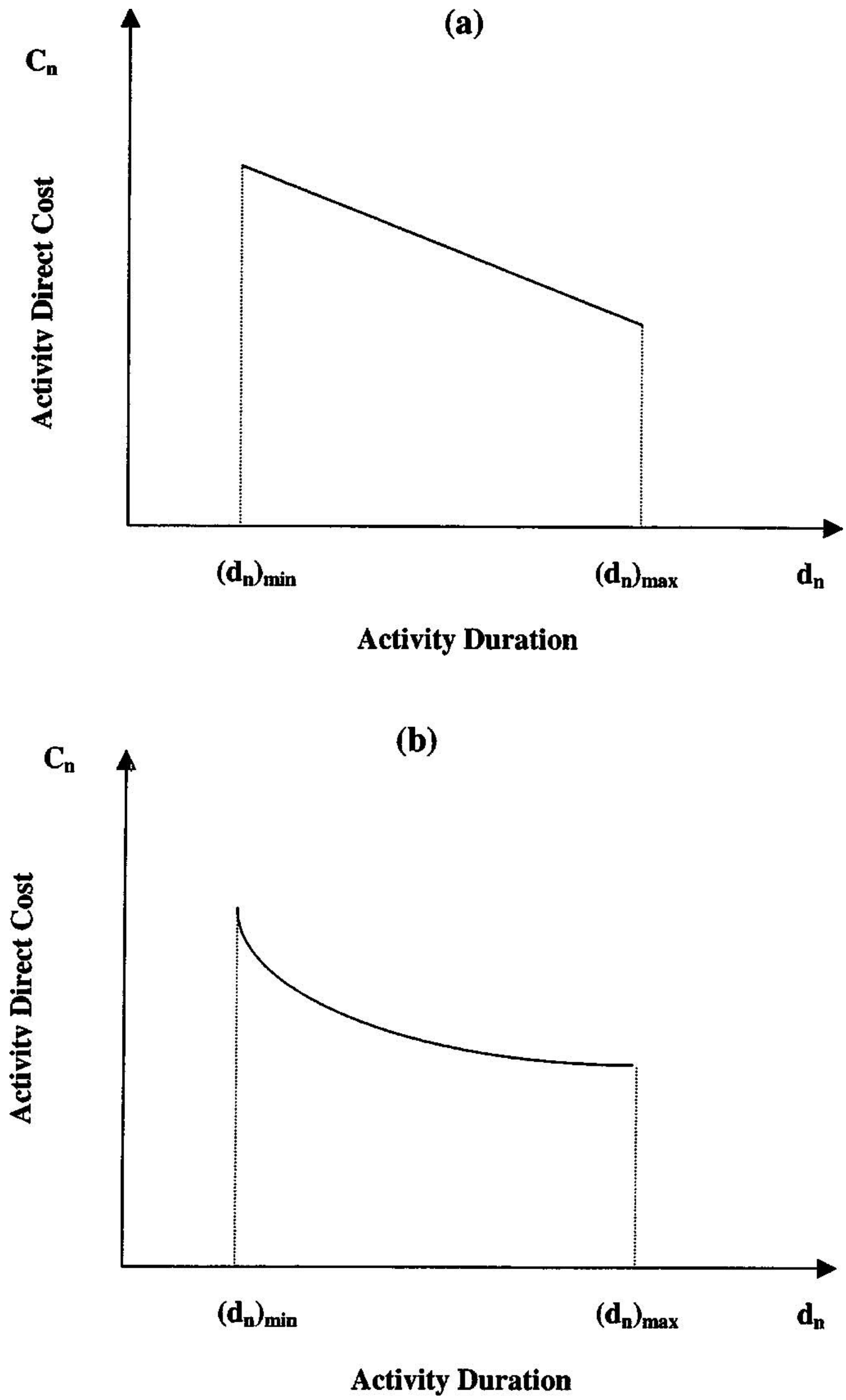

Figure 3.1: Activity cost distribution curves: (a) linear and (b) discrete (from [43])

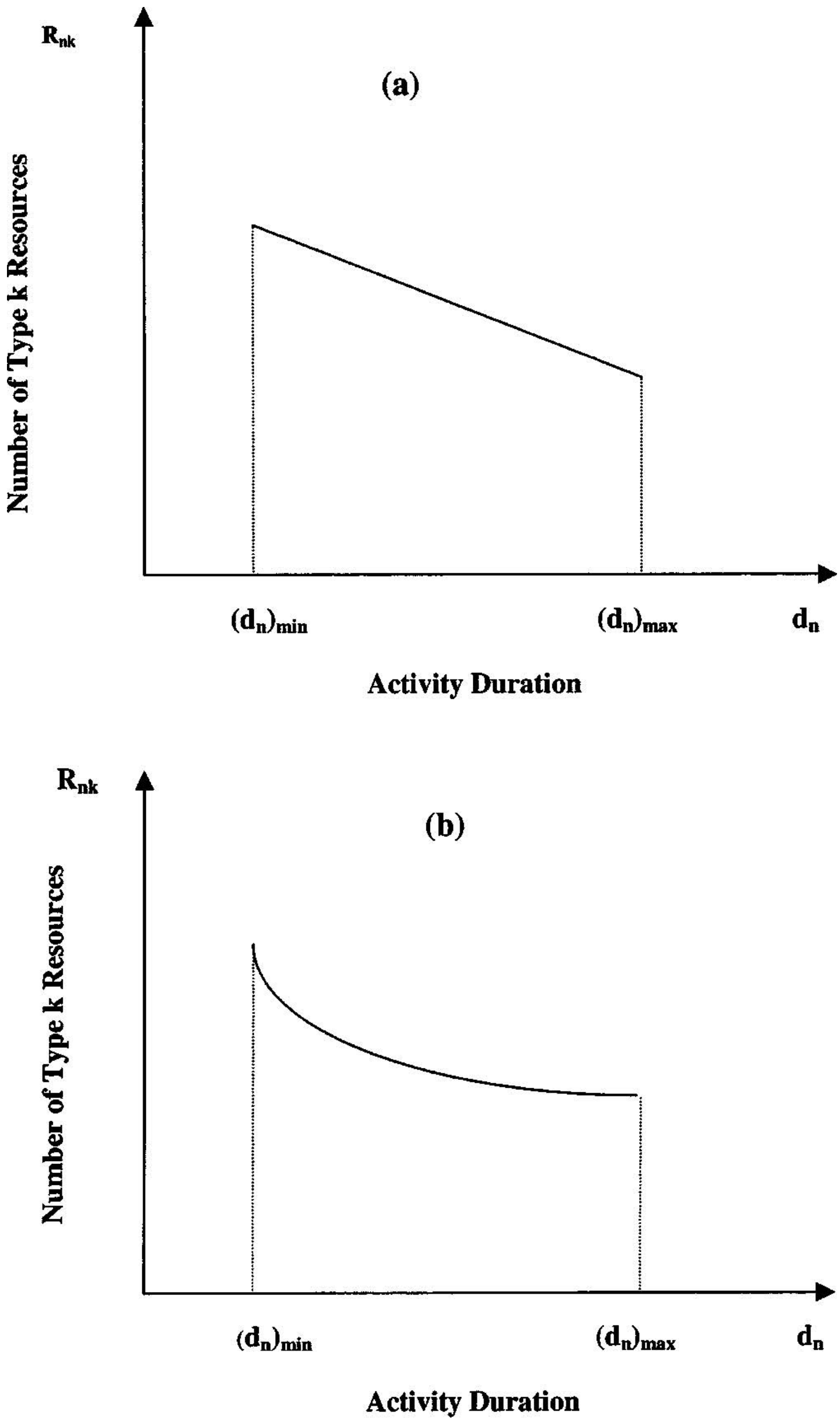

Figure 3.2: Activity resource usage distribution curves: (a) linear and (b) discrete(from [43]

Maximum Resource Constraint

$$\sum_{i \in \mathcal{S}_t} r_{ki} \leq R_{kt} \tag{3.1.7}$$

Peak resource usage deviation constraint

$$\left| \sum_{i \in \mathcal{S}_t} r_{ki} - \sum_{i \in \mathcal{S}_{t+1}} \right| \leq RL \tag{3.1.8}$$

where RL is the desired resource leveling limit.

Variable bounds

$$\mathfrak{d}_i^{min} \leq \mathfrak{d}_i :\leq \mathfrak{d}_i^{min} \tag{3.1.9}$$

3.2 Computer implementation

A numerical example is presented in order to illustrate the steps and capabilities of the model presented. The critical path method (CPM) network shown in Fig. 3.3 describes a project that consists of 12 activities (labeled A-L). It is assumed that there are no limitations on precedence relationships between succeeding activities. For the example project, these relationships are shown in Table 3.2. Cost-time tradeoff data are provided in Table 3.1. This includes the relationships between resources and durations and relationships between activities' direct costs and durations. An initial cost of \$6,000 and a daily cost of \$2,500 are used in this example. It is also assumed that each activity requires a particular type of resource, and different activities require different resources.

A computer program was developed in C on Linux operating system to implement the ACO algorithm explained in Alg. 1 of Chapter 2. The program starts with reading input parameters for the ACO algorithm like number of ants, pheromone evaporation constant, maximum number of ants etc. Each duration and start time for an activity and the corresponding resource utilization for executing an activity can be regarded as a solution component. The trail on each solution component is initially initialized to a small value. In the solution construction routine, ants decide upon a solution component (i.e. start time and duration for an activity) using the available pheromone information. After all ants construct their solutions, the objective function values are evaluated in the apply pheromone update routine of Alg. 1. The objective function evaluation procedure starts with reading of the activity precedence relationships and assignment of the time duration and start time for each activity. Using this information, the project duration is calculated and critical path is identified. Calculations for activity costs and resource usage are

Table 3.1: Relationship between activity duration and cost, resouce usage

Activity	Minimum duration	Maximum duration	Number of resources	Direct cost ($)
A	1	2	3-d	3000- 100d- 50d^2
B	4	5	9-d	7000- 300d- 75d^2
C	1	3	8-d	6000- 500d- 25d^2
D	1	2	3-d	8000- 600d- 50d^2
E	2	4	5-d	11000- 400d- 20d^2
F	2	3	5-d	11000- 400d- 75d^2
G	1	2	6-d	7000- 500d- 10d^2
H	1	2	4 -d	3500- 300d- 75d^2
I	2	4	9-d	3500- 300d- 50d^2
J	7	8	9-d	2500- 100d- 15d^2
K	4	6	7-d	5000- 200d- 25d^2
L	2	3	4-d	2000- 200d- 30d^2

done next. Then, the direct project cost is calculated as sum of costs of all the activities. After these calculations, the start time and duration of each activity are checked for precedence relationship constraints and a weighted normalized penalty is applied for each violation [8]. Now, the calculations for resource usage for each day till the project completion can be performed and the maximum resource usage constraint can be checked. The solution quality (objective function value) is penalized for each constraint violation using the penalty function method. Finally the complete project cost with the penalty for constraint violation is returned to the apply pheromone update routine.

3.3 Case study

In this study a maximum resouce usage limit of 7 was used and a solution satisfying this constraint was searched using the implementation ACO algorithm. Table 3.4 shows the starting time and activity durations obtained with this constraint. The project duration is 16 days. The total direct cost for the solution is $ 49775, while the indirect cost is $ 46000. Total cost of the project is $ 95775. Fig. 3.4 shows the daily resource usage profile for this schedule. In Fig. 3.4, the X-axis shows the number of days completed and the Y-axis shows the total resources used for that particular day. It can be observed that the total resource usage never exceeds the availability of 7 per day and this proves the capability of the ACO algorithm to find a feasible solution. The convergence of the ACO algorithm along with iteration is show in Fig. 3.5 There is a very quick improvement in the objective function value

Table 3.2: Activity precedence relationships.

Activity	Succeeding activity	relationship type	lag time (days)
A	B	SS	2
A	D	SS	2
B	C	FF	3
C	G	FS	0
D	E	SF	2
D	F	FF	4
E	H	SS	1
F	K	FS	0
G	I	FF	4
G	J	FF	2
H	K	FS	2
I	L	FS	1
J	L	FS	0
K	L	FS	0

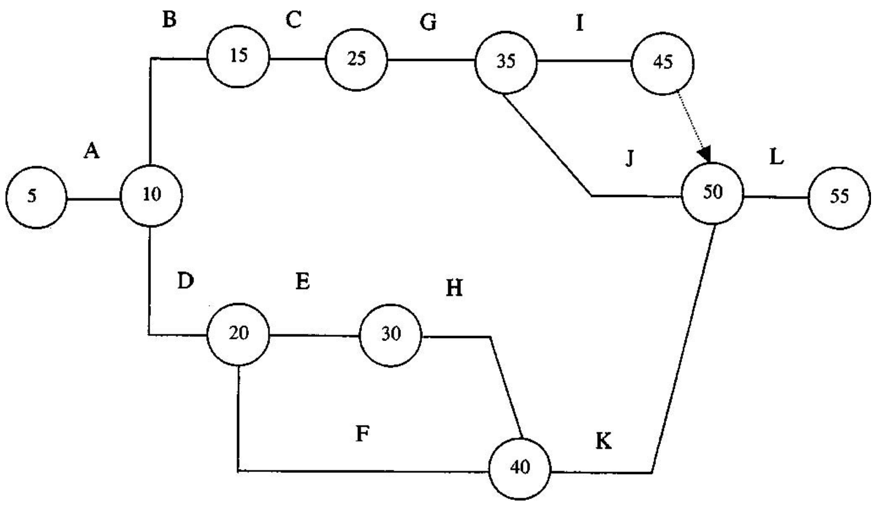

Figure 3.3: Network for test problem (from [43])

Table 3.3: Algorithm details for test problema

Algorithm used:	Rank Based Ant System
Ranks Used	5
Elitist Ants	1
No of variables	23,
No of ants	50
Max no of cycles	50
Max no of runs	3
Evaporation (rho)	0.200000
Initial trail	1.000000
Local update	used
α	1.000000,
Local update evap(γ)	0.200000
Local search used with	Global Best Ant
In local search	
Max no of Cycles	15
No of Ants	25
Evaporation ($ls\rho$)	0.500000

Table 3.4: Schedule with maximum resource usage 7.

Activity	$\mathfrak{d}_i$	# resources	$\mathfrak{s}_i$	$\mathfrak{c}_i$
A	2	1	0	2600
B	5	4	2	3625
C	3	5	7	4275
D	2	1	3	6600
E	4	1	0	9080
F	3	2	4	9125
G	2	4	10	5600
H	2	2	1	2600
I	4	5	12	1500
J	8	1	8	740
K	6	1	6	2900
L	3	1	12	1130

in the initial iterations and the best solution is found at the end of 5th iteration. The construction graph used by the ants is show in Fig. 3.6. Fig. 3.6 shows the available solution components for each activity lined up vertically for each variable (X-Axis). Ants start from their home (point [0,0]) and then take a path to a solution component of each variable as shown. For example, the ants chose the last solution component for first two variables (duration of activity A and B) and the first solution component for last variable (start time of L) before reaching the destination food source.

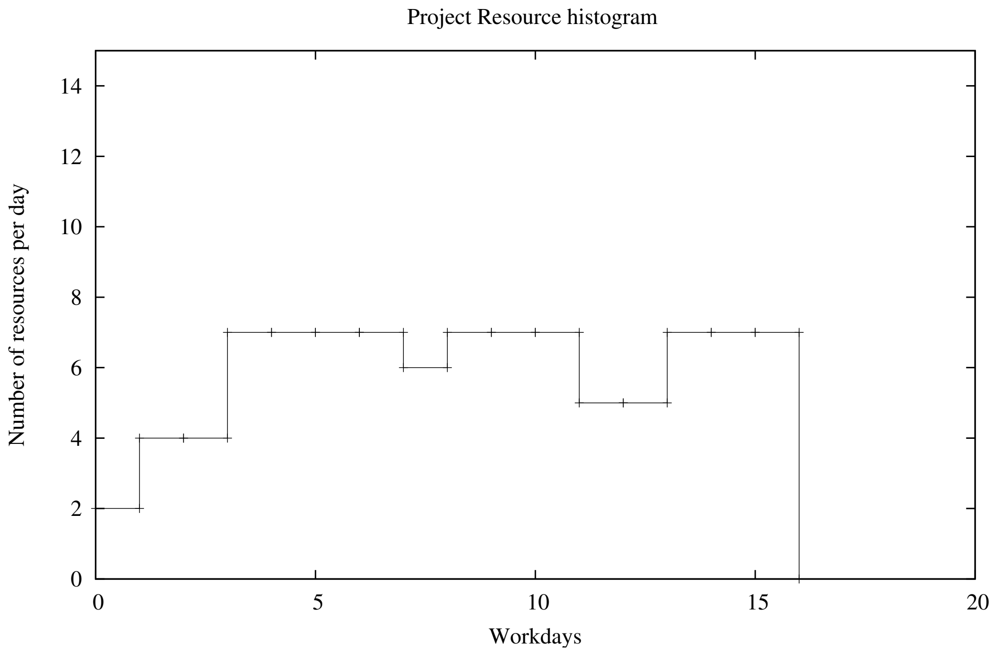

Figure 3.4: Project resource histogram for case study

3.4 Conclusions

This chapter presents the formulation and solution of constrained resource allocation and leveling problem in construction scheduling. A solution is attempted using ACO. The variables considered are the start time and duration of each activity. Resource usage is constrained by a predefined upper limit. A unique feature of this formulation is that all types of relationship constraints in activities can be accomodated. A case study is presented to demonstrate the successful use of ACO for this problem.

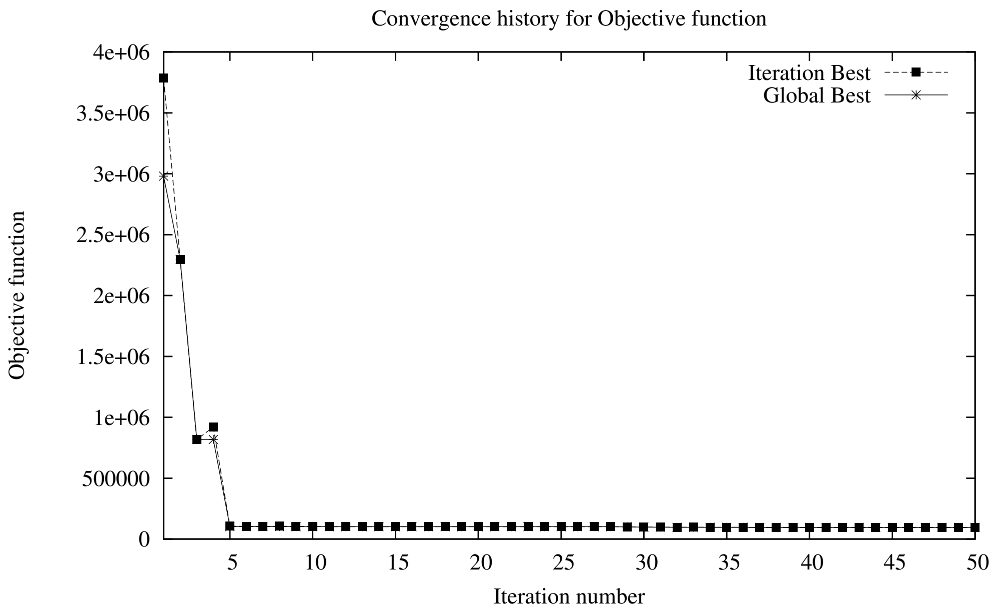

Figure 3.5: Convergence of algorithm for case study

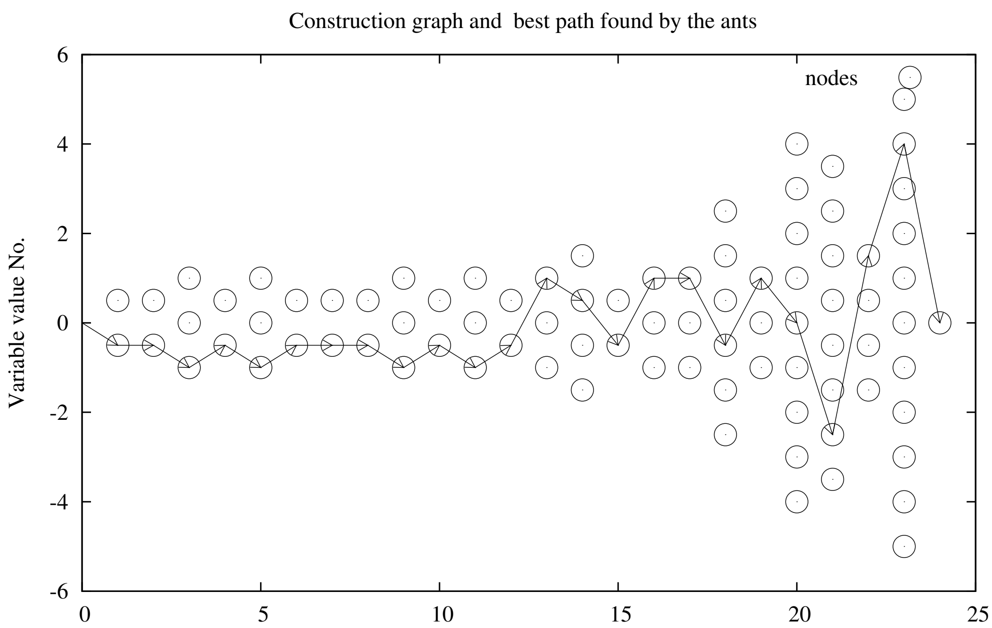

Figure 3.6: Construction Graph for case

Chapter 4

Time-Cost Trade-Off using Genetic Algorithms

The time cost trade-off (TCTO) problem was introduced in 1.5 The objective of this chapter is to present the formulation and results of a multi-objective optimization model that supports minimizing construction time and cost. Here it is assumed that there is a set of options for carrying out each activity and the engineer is required to choose an option for each activity so as to simulataneously minimize the cost of carrying out activities (direct cost) as well as the project duration. A higher project duration contributes to a higher indirect cost and the optimum solution balances direct and indirect cost to obtain a minimum project cost as shown in Fig. 4.1. However to calculate the totalcost, we need to know the exact mathematical relationship between the project duration and indirect cost. Such information is not always known in advance, so the aim in solving this problem would be to find the time-cost trade-off curve and leave the task of deciding a solution belonging to the optimal time-cost trade-off curve to the decision maker/engineer. The TCTO curve is obtained using a multiobjective genetic algorithm called NSGA-II [8].

4.1 Problem Formulation

Solving construction TCTO problem involves making a sequence of decisions to choose the proper methods, resources, and equipment to perform each activity of a project, which optimizes the overall performance of the project in terms of time and cost. Let us consider a network with n_a activities. Each activity i can be performed with θ_i combinations of methods, resources and equipment with a corresponding cost $\mathfrak{c}_i$ and time duration $\mathfrak{t}_i$. x_i is the option chosen for activity i For the optimization problem, the vector of decision variables $X = \{x_1, x_2, \ldots, x_{n_a}\}$. Let EST_i be the earliest start time of the i th activity calculated according to the precedence

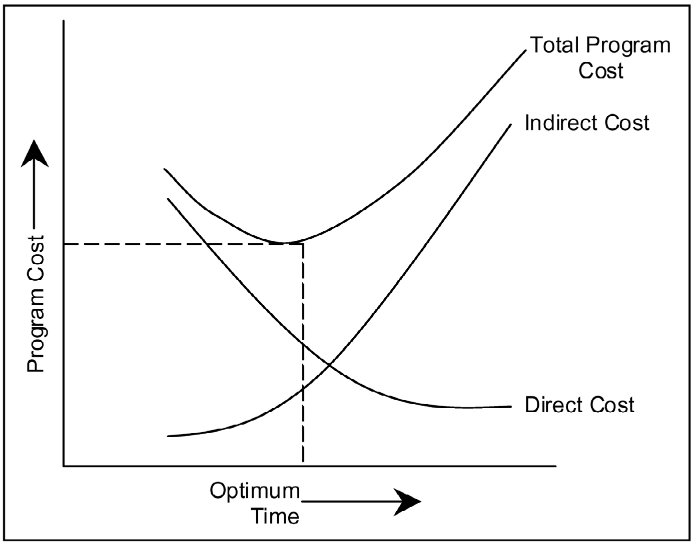

Figure 4.1: Time cost trade off

relationships for the network.

The multi-objective TCTO problem is:

$$\text{minimize} \qquad \mathfrak{C}(X) = \sum_{i=1}^{n_a} \mathfrak{c}_i(x_i) \tag{4.1.1}$$

$$\text{minimize} \qquad \mathcal{T} = max\{EST_i + \mathfrak{t}_i(x_i) \quad | \quad i = 1, 2, \ldots, n_a\} \tag{4.1.2}$$

subject to

$$1 \leq x_i \leq \theta_i \tag{4.1.3}$$

The project duration $\mathcal{T}$ is calculated using the critical path method.

4.2 Non dominated sorting Genetic Algorithm - II (NSGA II)

The multi-objective optimization problem of Eq. 4.1.1 - 4.1.3 is solved using the Non dominated sorting Genetic Algorithm - II (NSGA II) [8]. As such, the present model is implemented in three major phases:

1. Initialization phase that generates an initial set of S possible solutions for the problem;
2. fitness evaluation phase that calculates the cost, and time of each generated solution;
3. population generation phase that seeks to improve the fitness of solutions over successive generations.

4.3 Computer implementation and case study

The above formulation for finding out the project duration and cost was implemented in C on Linux operating system. Source code for NSGA II was obtained from KanGAL, IIT Kanpur (`www.iitk.ac.in/kangal/soft.htm`). The NSGA II software from KanGAL requires the user to change only the objective function for a specific application. The project duration is calculated using the CPM routine, which takes actvitiy precedence and duration as input. The direct cost of the project can be calculated by simply summing up the costs of the individual activities for a particular vector of options (X). As an example, the 18-activity network of [18] was solved. Precedence relationships between the different activities is shown in Fig. 4.2. Various resource utilization options for all the activities along with their associated cost and duration are shown in Table 4.1 Binary solution encoding is used for every activity option variable depending upon the number of options for that activity. The parameters used for a sample run were

- Population size = 500
- Number of generations = 150
- Number of objective functions = 2
- Number of constraints = 0
- Number of real variables = 0
- Number of binary variables = 18
- Probability of crossover of binary variable = 0.8
- Probability of mutation of binary variable = 0.02
- Seed for random number generator = 3.782000e-01
- Number of crossover of binary variable = 536517
- Number of mutation of binary variable = 59641

Table 4.1: Resource utilization options for test problem

i	$\mathfrak{t}_i$	$\mathfrak{c}_i$	i	$\mathfrak{t}_i$	$\mathfrak{c}_i$
1	14	2400	9	18	240
1	15	2150	9	20	180
1	16	1900	9	23	150
1	21	1500	9	25	100
1	24	1200	10	15	450
2	15	3000	10	22	400
2	18	2400	10	33	320
2	20	1800	11	12	450
2	23	1500	11	16	350
2	25	1000	11	20	300
3	15	4500	12	22	2000
3	22	4000	12	24	1750
3	33	3200	12	28	1500
4	12	45000	12	30	1000
4	16	35000	13	14	4000
4	20	20000	13	18	3200
5	22	20000	13	24	1800
5	24	17500	14	9	3000
5	28	15000	14	15	2400
5	30	10000	14	18	2200
6	14	40000	15	16	3500
6	18	32000	16	20	3000
6	24	18000	16	22	2000
7	9	30000	16	24	1750
7	15	24000	16	28	1500
7	18	22000	16	30	1000
8	14	220	17	14	4000
8	15	215	17	18	3200
8	16	200	17	24	1800
8	21	208	18	9	3000
8	24	120	18	15	2400
9	15	300	18	18	2200

Fig. 4.3 shows the objective function values of the individuals of the first generation. It can be observed that there are many dominated individuals in the first generation. The minimum project duration is around 124 days and the minimum project cost is around $100,000. Fig. 4.4 shows the plot of project duration and direct cost for the last generation. In this generation the algorithm has searched large number of solutions along the non-dominated front. Best non-dominated solutions from all the generations are shown in Fig. 4.5. It can be observed that the algorithm has successfully found a diverse set of solutions on the trade-off curve.

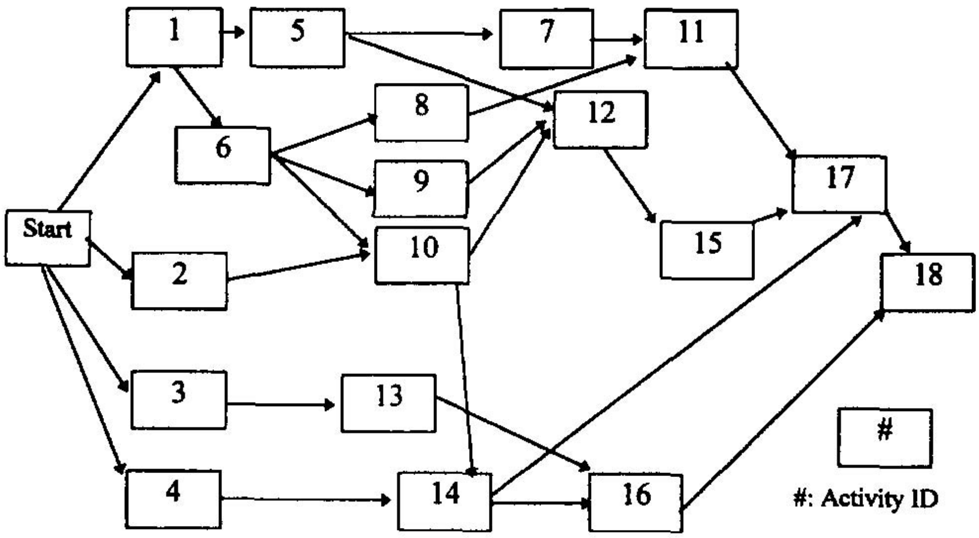

Figure 4.2: Network for test problem (from [18])

4.3.1 Validation of the model

In order to validate the results provided by the present model, they are compared to those reported in the literature for the same application example [18]. The comparison confirms that the present model is capable of generating the same set of optimal solutions as those reported in [18] for the time-cost trade-off analysis. In, Fig. 4.5 we can see that there are disjoint regions along the non-dominated curve. Similar sets were obtained by the implementation in [18].

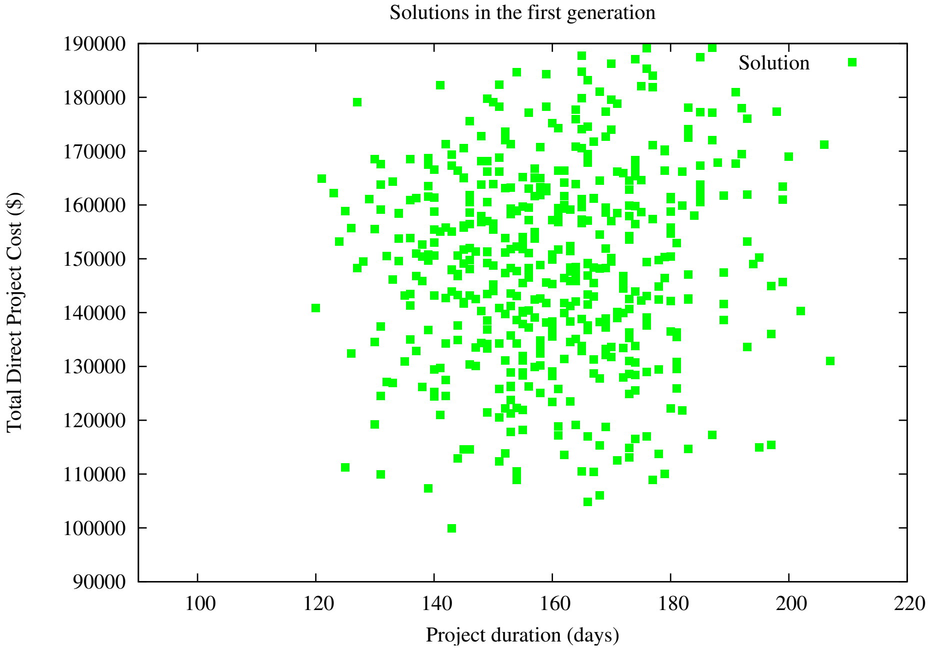

Figure 4.3: The initial generation

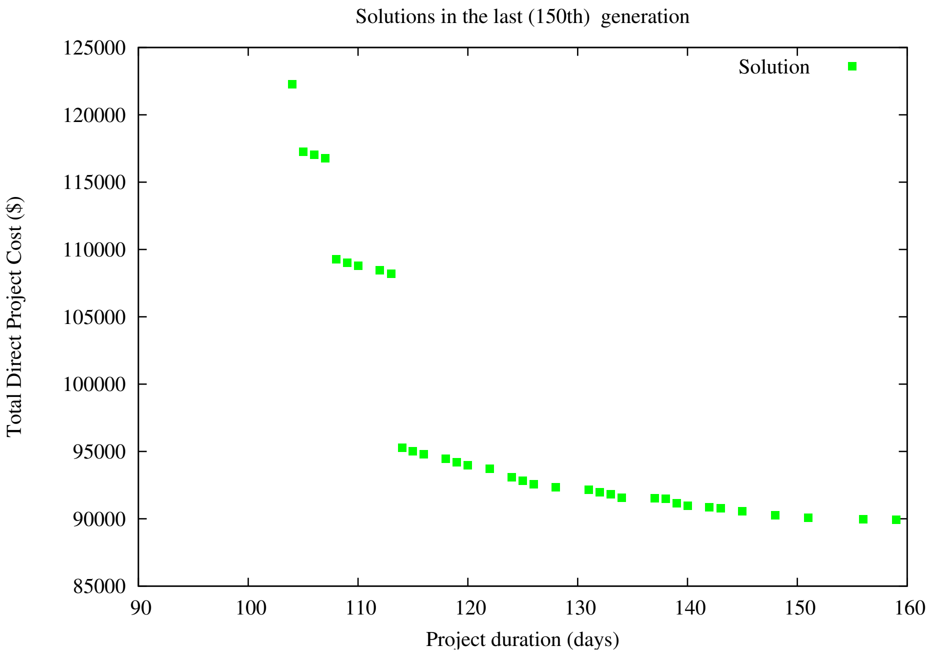

Figure 4.4: The last generation

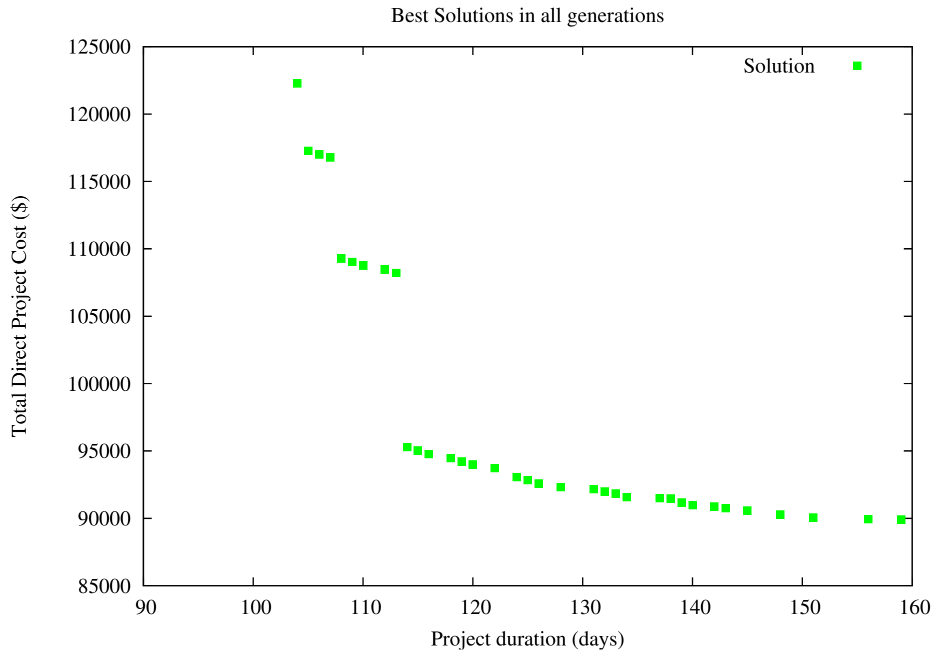

Figure 4.5: Best solutions

4.4 Conclusions

This chapter presents a multi-objective time cost trade-off problem in construction scheduling and a solution is attempted using NSGA II. First objective is project duration and the second objective is project direct cost. It is assumed that different resource utilization options for each activity leading to different activity costs and duration are known in advance. The aim is to find the pareto optimal solutions by choosing a combination of different resource utilization options. The efficacy of the implementation was demonstrated on a 18 activity network. The computer implementation could successfully obtain many solutions on the non-dominated front. In this formulation resource constraints are not considered. It would be interesting to formulate an integrated model which can handle objectives time, cost and resource leveling along with resource usage constraints.

Chapter 5

Summary

From the discussion in previous chapters it can be observed that although time cost trade off and resource scheduling are problems of equal interest to project managers, they have been treated separately. One of the reasons for the separate treatment is that efficient techniques were not available for solution of individual problems, so increasing the complexity of the problem would have been only of theoretical interest. But now approximate solution techniques which can deal with the complexity are available. An integrated framework for simulataneous solution of time cost trade off and resouce scheduling problems is required. Non traditional optimization techniques like Genetic Algorithms and Ant Colony Optimization can easily handle discrete and continuous variables, non linear objectives and constraints and are ideally suited multiobjective optimization problems [8]. Simulataneous solution of the time-cost trade-off problem and constrained resource leveling problem is difficult because in time-cost trade-off problem, aim is to find the duration of each activity. Duration of each activity decides the critical path and the activity floats. Since activity floats are not known in advance, it is not possible to put tight bounds on starting time of each activity. As deciding starting of each activity is aim of resource allocation, these two problems have to be solved sequentially. The aims of the present study are

- Formulation with multiple objectives of time cost trade off and resouce scheduling problems in construction.

- Solving the above formulation with multiobjective non traditional optimization techniques.

- Implementation of the above framework and demonstration on case studies.

5.1 Main Contributions

As a part of the project following softwares were developed in the C programming language on Linux operating system

- Critical path method software for activity-on-node and activity-on-edge networks
- ACO software for optimization of a general mixed integer non linear programming problem using Ant Colony Optimization

The above two softwares were used to solve the multiobjective time-cost trade-off problem and the constrained resource scheduling problem.

5.2 Future work

Many interesting possibilities arise out of the present work. It would be interesting to study the effect of parameter settings of the ACO and NSGA II algorithm on the performance on case studies. Recently quality aspect of construction has been incorporated into the time-cost trade-off problem [27, 14]. A problem formulation which takes into account the project time, cost and quality along with the resource constraints would be a challenging problem to work on. Further, one can explore the effect of converting the resource leveling constraint (as in Eq. 3.1.8) into an objective and studying this effect on the quality of solutions obtained.

Bibliography

[1] Hojjat Adeli and H.S. Park. *Neurocomputing for design automation*. CRC Press, 1998. 1.5

[2] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley Series in Computer Science and Information Processing. Addison-Wesley, 1974. 1.4.3, 1.5, 1.6

[3] Christian Blum and Marco Dorigo. The hyper-cube framework for ant colony optimization. *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, 34(2):1161–1172, 2004. 2.3, 2.3, 2.3

[4] B. Bullnheimer, R. Hartl, and C. Strauss. A new rank based version of the ant system — a computational study, 1997. 2.3

[5] Weng-Tat Chan, David K. H. Chua, and Govindan Kannan. Construction resource scheduling with genetic algorithms. *Journal of Construction Engineering and Management*, 122(2):125–132, 1996. 1.4.3, 1.5

[6] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. MIT Press, 1990. 1.4.3, 1.5, 1.6

[7] Kalyanmoy Deb. *Optimization for Engineering Design*. Prentice Hall, New Delhi, India, 1995. 1.6

[8] Kalyanmoy Deb. *Multiobjective Optimization using Evolutionary Algorithms*. Wiley–Interscience, New York, 2001. 3.2, 4, 4.2, 5

[9] M. Dorigo, V. Maniezzo, and A. Colorni. The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics*, 26:29–41, 1996. 2.1, 2.3, 2.3

[10] Marco Dorigo. Optimization, learning and natural algorithms. Technical Report MI-92-03, PhD thesis, Dip. Elettronica e Informazione, Politecnico di Milano, Italy, 1992. 2.1

[11] Marco Dorigo and Thomas Stützle. *Ant Colony Optimization*. MIT Press, Cambridge , MA, 2004. 2.3

[12] Arthur C. Doyle. *Complete works of Sherlock Holmes*. Rupa Paperbacks, 1990. 1.1

[13] Said M. Easa. Resource leveling in construction by optimization. *Journal of Construction Engineering and Management*, 115(2):302–316, 1989. 1.4.1, 1.4.3, 1.5

[14] Khaled El-Rayes and Amr Kandil. Time-cost-quality trade-off analysis for highway construction. *Journal of Construction Engineering and Management*, 131(4):477–486, 2005. 1.5, 5.2

[15] Khaled El-Rayes and Osama Moselhi. Optimizing resource utilization for repetitive construction projects. *Journal of Construction Engineering and Management*, 127(1):18–27, 2001. 1.4.3, 1.5

[16] Ashraf M. Elazouni and Ahmed A. Gab-Allah. Finance-based scheduling of construction projects using integer programming. *Journal of Construction Engineering and Management*, 130(1):15–24, 2004. 1.4.3, 1.5

[17] Ashraf M. Elazouni and Fikry Gomaa Metwally. Finance-based scheduling: Tool to maximize project profit using improved genetic algorithms. *Journal of Construction Engineering and Management*, 131(4):400–412, 2005. 1.4.3, 1.5

[18] Chung-Wei Feng, Liang Liu, and Scott A. Burns. Using genetic algorithms to solve construction time-cost trade-off problems. *Journal of Computing in Civil Engineering*, 11(3):184–189, 1997. (document), 1.5, 4.3, 4.2, 4.3.1

[19] M. Gen and R. Cheng. *Genetic algorithms and engineering optimization*. Wiley–Interscience, New York, 2000. 1.5

[20] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley Publishing Company, 1989. 1.4.3, 1.5, 1.6

[21] Miklos Hajdu. *Network Scheduling Techniques for Construction Project Management*. Kluwer Academic, 1997. 1.1

[22] Tarek Hegazy. Optimization of resource allocation and leveling using genetic algorithms. *Journal of Construction Engineering and Management*, 125(3):167–175, 1999. 1.4.3, 1.5

[23] Tarek Hegazy and Tolga Ersahin. Simplified spreadsheet solutions. ii: Overall schedule optimization. *Journal of Construction Engineering and Management*, 127(6):469–475, 2001. 1.4.3, 1.5

[24] Tarek Hegazy and Moustafa Kassab. Resource optimization using combined simulation and genetic algorithms. *Journal of Construction Engineering and Management*, 129(6):698–705, 2003. 1.4.3, 1.5

[25] Tarek Hegazy, Abdul Karim Shabeeb, Emad Elbeltagi, and Tariq Cheema. Algorithm for scheduling with multiskilled constrained resources. *Journal of Construction Engineering and Management*, 126(6):414–421, 2000. 1.4.3, 1.5

[26] Chris Hendrickson and Tung Au. *Project Management for Construction*. Prentice Hall, 1998. (document), 1.1, 1.4, 1.5, 1.2, 1.3, 1.5

[27] Amr Kandil. *Multi-Objective Optimization for Large-Scale Highway Construction Projects*. PhD thesis, University of Illinois at Urbana-Champaign, Urbana Champaign, IL, USA, 2005. 1.5, 5.2

[28] Fadi Karaa and Anas Nasr. Resource management in construction. *Journal of Construction Engineering and Management*, 112(3):346–357, 1986. 1.4.3, 1.5

[29] James Kennedy and Russell C. Eberhart. *Swarm intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001. 1.6

[30] Sou-Sen Leu, An-Ting Chen, and Chung-Huei Yang. Fuzzy optimal model for resource-constrained construction scheduling. *Journal of Computing in Civil Engineering*, 13(3):207–216, 1999. 1.5

[31] Sou-Sen Leu and Chung-Huei Yang. GA-based multicriteria optimal model for construction scheduling. *Journal of Construction Engineering and Management*, 125(6):420–427, 1999. 1.4, 1.5

[32] Heng Li, J.-N. Cao, and P. E. D. Love. Using machine learning and ga to solve time-cost trade-off problems. *Journal of Construction Engineering and Management*, 125(5):347–353, 1999. 1.5

[33] Heng Li and Peter Love. Using improved genetic algorithms to facilitate time-cost optimization. *Journal of Construction Engineering and Management*, 123(3):233–237, 1997. 1.5

[34] Jinming Liu and Fred Rahbar. Project time-cost trade-off optimization by maximal flow theory. *Journal of Construction Engineering and Management*, 130(4):607–609, 2004. 1.5

[35] Liang Liu, Scott A. Burns, and Chung-Wei Feng. Construction time-cost trade-off analysis using lp/ip hybrid method. *Journal of Construction Engineering and Management*, 121(4):446–454, 1995. 1.5

[36] Kris G. Mattila and Dulcy M. Abraham. Resource leveling of linear schedules using integer linear programming. *Journal of Construction Engineering and Management*, 124(3):232–244, 1998. 1.4.3, 1.5

[37] Kris G. Mattila and Amy Park. Comparison of linear scheduling model and repetitive scheduling method. *Journal of Construction Engineering and Management*, 129(1):56–64, 2003. 1.4.3, 1.5

[38] O. Moselhi and A. Hassanein. Optimized scheduling of linear projects. *Journal of Construction Engineering and Management*, 129(6):664–673, 2003. 1.4.3, 1.5

[39] Microsoft Project. *Reference Manual*. Microsoft Corp. Redmond WA, 1995. 1.4.3, 1.5

[40] Bryan Christopher Que. Incorporating practicability into genetic algorithm-based time-cost optimization. *Journal of Construction Engineering and Management*, 128(2):139–143, 2002. 1.5

[41] Hans-Paul Paul Schwefel. *Evolution and Optimum Seeking: The Sixth Generation*. John Wiley & Sons, Inc., New York, NY, USA, 1993. 1.6

[42] Ahmed B. Senouci and Hojjat Adeli. Resource scheduling using neural dynamics model of adeli and park. *Journal of Construction Engineering and Management*, 127(1):28–34, 2001. 1.5

[43] Ahmed B. Senouci and Neil N. Eldin. Use of genetic algorithms in resource scheduling of construction projects. *Journal of Construction Engineering and Management*, 130(6):869–877, 2004. (document), 1.5, 3, 3.1, 3.2, 3.3

[44] Daisy X. M. Zheng, S. Thomas Ng, and Mohan M. Kumaraswamy. Applying a genetic algorithm-based multiobjective approach for time-cost optimization. *Journal of Construction Engineering and Management*, 130(2):168–176, 2004. 1.5

[45] Daisy X. M. Zheng, S. Thomas Ng, and Mohan M. Kumaraswamy. Applying pareto ranking and niche formation to genetic algorithm-based multiobjective time–cost optimization. *Journal of Construction Engineering and Management*, 131(1):81–91, 2005. 1.5