Overall benefit-duration optimization and its genetic algorithm application in construction project scheduling

Pan, Heng

2008

Pan, H. (2008). Overall benefit-duration optimization and its genetic algorithm application in construction project scheduling. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/41774>

<https://doi.org/10.32657/10356/41774>

Overall Benefit-Duration Optimization and Its Genetic Algorithm Application in Construction Project Scheduling

Pan Heng

School of Civil & Environmental Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2008

Acknowledgements

I would like to express my sincere gratitude, first and foremost, to my supervisor, Associate Professor Ting Seng Kiong, for his patient inspirations and encouragements in the course of this study.

I would like to extend my gratitude to my senior, Dr Kong DeQuan for his valuable academic advices and continuous supports and encouragements.

I am fortunate to be a research student in School of Civil and Environmental Engineering (CEE), Nanyang Technological University (NTU). Thanks to its remarkable people and outstanding academic environment, my experience as a research student has been pleasurable and fruitful. It would not be possible to finish this research work without friendship and tremendous supports from all my friends in and outside of NTU.

Finally, I would like to thank my wife, Dr. Zeng Zhen and my family in China for their patience, understanding and supports in this chapter of my life.

Table of Contents

Acknowledgements..... I

Abstract.....X

Table of Contents II

List of Figures.....V

List of Tables.....VIII

Chapter 1 Introduction 1

 1.1 Background..... 1

 1.2 Research Objectives..... 3

 1.3 Research Methodology..... 4

 1.4 Research Scope 6

 1.5 Research Significances..... 8

 1.6 Organization of the Thesis..... 9

Chapter 2 Literature Review 11

 2.1 Duration-Cost Relationship of a Construction Project..... 11

 2.1.1 Introduction..... 11

 2.1.2 Duration-Cost Relationship of a Construction Project..... 12

 2.1.3 Time-Cost Relationship of a Task 13

 2.2 Time-Cost Trade-Off (TCT) Problems..... 15

 2.2.1 TCT and Existing Optimization Techniques..... 16

 2.2.2 Network Compression 18

 2.2.3 Literature Review of TCT Problems 18

 2.2.4 Development of Solutions to TCT Problems..... 21

 2.3 Overview of Genetic Algorithm (GA) 23

 2.3.1 General Introduction..... 23

 2.3.2 Operators and Process of Typical GA 24

 2.3.3 Mathematics of Typical GA 28

 2.3.4 GA for Optimization Problems in Construction Project Scheduling 29

 2.4 GA Applications in TCT Problems 30

 2.4.1 Introduction..... 31

 2.4.2 Background and Existing Research in GA 31

 2.5 GA Applications in Construction Project Scheduling..... 33

 2.5.1 GA Hybridization 33

 2.5.2 GA Applications in Commercial Software 35

Table of Contents

2.5.3 Computational Times in GA Applications.....	36
2.6 Summary	37
Chapter 3 Research Objectives Analysis.....	39
3.1 Characteristics of Construction Projects	39
3.2 Different Economic Perspectives of Project Owners and Contractors	40
3.3 Opportunity Income and Economic Incentive to Schedule Compression.....	41
3.4 Research Objectives.....	44
3.5 Necessity for a New Schedule Optimization Scheme.....	46
Chapter 4 Overall Benefit Duration Optimization.....	47
4.1 Definition and Assumptions of Overall Benefit-Duration Optimization	47
4.1.1 Definition of OBDO.....	47
4.1.2 Assumptions of OBDO.....	48
4.2 Objective Function of OBDO.....	49
4.3 Formulation of OBDO	52
4.4 An Illustrative Example.....	56
4.4.1 Time-Cost Trade-off (TCT) Solution	57
4.4.2 OBDO Solution.....	60
4.4.3 Conclusion	61
4.5 Significances of OBDO.....	61
Chapter 5 GAOBDO Model Formulation	65
5. 1. Introduction.....	65
5. 2. Genetic Algorithm Model for OBDO	66
5. 3. GAOBDO Model Formulation	67
5.3.1 Chromosome Structure.....	67
5.3.2 Two-point Crossover Operation.....	68
5.3.3 Mutation Operation	69
5. 4. Computer Implementation of GAOBDO	72
Chapter 6 Case Studies.....	77
6.1 Introduction	77
6.2 Sensitivity Analysis of Case Study 1	77
6.2.1. Sensitivity Analysis of Crossover Rate (Case Study 1).....	79
6.2.2. Sensitivity Analysis of Mutation Rate (Case Study 1).....	82
6.2.3. Sensitivity Analysis of Population Size (Case Study 1).....	84
6.2.4. Sensitivity Analysis of Generation Number (Case Study 1).....	86
6.2.5. Convergence Results of GA Parameters (Case Study 1).....	88
6. 3 Sensitivity Analysis of Case Study 2	91
6.3.1. Sensitivity Analysis of Crossover Rate (Case Study 2).....	92
6.3.2. Sensitivity Analysis of Mutation Rate (Case Study 2).....	95
6.3.3. Sensitivity Analysis of Population Size (Case Study 2).....	97
6.3.4. Sensitivity Analysis of Generation Number (Case Study 2).....	100

Table of Contents

6.3.5. Convergence Results of GA Parameters (Case Study 2)	102
6.4 Cross Sensitivity Analysis	104
6.4.1. Cross Sensitivity Analysis of Crossover Rate and Mutation Rate	105
6.4.2. Cross Sensitivity Analysis of Population Size and Generation Number	107
6.4.3. Summary of Cross Sensitivity Analysis	108
6.5 Case Study 3 – A Completed Commercial Construction Project.....	109
6.5.1. GAOBDO Result of Case Study 3	109
6.5.2. Convergence Results of GA Parameters (Case Study 3)	110
6.6. Summary of Case Studies.....	113
6.6.1 Comparison with Heuristic Methods.....	113
6.6.2 Comments on Crossover and Mutation Rates.....	114
6.6.3 Comments on Population Size and Generation Number.....	116
6.6.4 Comments on Computational Times and Sub-optimal Solutions	118
Chapter 7 Conclusions and Suggestions	120
7.1 Conclusions	121
7.2 Limitations of the Research.....	122
7.2.1 Research Scope	123
7.2.2 Complexity of Scheduling Optimization Problems	123
7.2.3 Limitations of OBDO Concept and GAOBDO Model	123
7.3 Future Directions of the Research.....	124
7.3.1 OBDO under Uncertainty	124
7.3.2 OBDO with Resource Allocation and Resource Leveling	125
7.3.3 Hybridization of GAOBDO Model.....	125
7.3.4 Other Influencing Economic Factors	126
7.3.5 Internet Accessibility of GAOBDO Model	127
References	128
Publications	136
Conference Papers:	136
Appendix	137
Appendix: Macro Program for Overall Benefit-duration Optimization Using Genetic Algorithm	137

List of Figures

Figure 1.1 Organization of the Thesis..... 9

Figure 2.1 Duration-Cost Relationship of a Construction Project..... 12

Figure 2.2 Time-Cost Relationships of a Task 14

Figure 2.3 Process of Typical Genetic Algorithm..... 27

Figure 3.1 Owner’s Opportunity Income 43

Figure 4.1 Overall Benefit-Duration Optimization..... 50

Figure 4.2 Maximization of Z through Network Compression 51

Figure 4.3 OBDO Process 55

Figure 4.4 An Illustrative Example: All-normal Solution..... 57

Figure 4.5 An Illustrative Example: First Compression 58

Figure 4.6 An Illustrative Example: Second Compression 59

Figure 4.7 An Illustrative Example: OBDO Solution..... 60

Figure 5.1 Chromosome Structure of GAOBDO 68

Figure 5.2 Two-point Crossover Operation of GAOBDO 69

Figure 5.3 Mutation Operation of GAOBDO..... 70

Figure 5.4 Flow Chart of GAOBDO..... 71

Figure 5.5 Data-entry for Options of Alternative Construction Methods 73

Figure 5.6 Data-entry for Daily Opportunity Incomes 73

List of Figures

Figure 5.7 Data-entry for Daily Interests Saving.....	73
Figure 5.8 Data-entry for Indirect Cost.....	74
Figure 5.9 Data-entry for Daily Incentive	74
Figure 5.10 Data-entry for Population Size.....	74
Figure 5.11 Data-entry for Crossover Rate	75
Figure 5.12 Data-entry for Mutation Rate.....	75
Figure 5.13 Data-entry for Generation Number	75
Figure 5.14 Customized Interface of GAOBDO Results.....	76
Figure 6.1 Project Network of Case Study 1	78
Figure 6.2 Custom Form of Case Study 1 in MS Project 2003	78
Figure 6.3 Sensitivity Analysis of Crossover Rate-1 (Case Study 1)	80
Figure 6.4 Sensitivity Analysis of Crossover Rate-2 (Case Study 1)	81
Figure 6.5 Sensitivity Analysis of Mutation Rate-1 (Case Study 1).....	83
Figure 6.6 Sensitivity Analysis of Mutation Rate-2 (Case Study 1).....	84
Figure 6.7 Sensitivity Analysis of Population Size-1 (Case Study 1).....	85
Figure 6.8 Sensitivity Analysis of Population Size-2 (Case Study 1).....	86
Figure 6.9 Sensitivity Analysis of Generation Number-1 (Case Study 1)	87
Figure 6.10 Sensitivity Analysis of Generation Number-2 (Case Study 1)	88
Figure 6.11 Convergence Result of Crossover Rate (Case Study 1)	89
Figure 6.12 Convergence Result of Mutation Rate (Case Study 1).....	90
Figure 6.13 Convergence Result of Population Size (Case Study 1)	90

List of Figures

Figure 6.14 Project Network of Case Study 2	91
Figure 6.15 Custom Form of Case Study 2 in MS Project 2003	92
Figure 6.16 Sensitivity Analysis of Crossover Rate-1 (Case Study 2)	93
Figure 6.17 Sensitivity Analysis of Crossover Rate-2 (Case Study 2)	94
Figure 6.18 Sensitivity Analysis of Mutation Rate-1 (Case Study 2).....	96
Figure 6.19 Sensitivity Analysis of Mutation Rate-2 (Case Study 2).....	97
Figure 6.20 Sensitivity Analysis of Population Size-1 (Case Study 2).....	98
Figure 6.21 Sensitivity Analysis of Population Size-2 (Case Study 2).....	99
Figure 6.22 Sensitivity Analysis of Generation Number-1 (Case Study 2)	100
Figure 6.23 Sensitivity Analysis of Generation Number-2 (Case Study 2)	101
Figure 6.24 Convergence Result of Crossover Rate (Case Study 2)	102
Figure 6.25 Convergence Result of Mutation Rate (Case Study 2).....	103
Figure 6.26 Convergence Result of Population Size (Case Study 2)	103
Figure 6.27 Cross Sensitivity of Crossover and Mutation Rates.....	107
Figure 6.28 Cross Sensitivity of Population Size and Generation Number	108
Figure 6.29 GAOBDO Results of Case Study 3.....	110
Figure 6.30 Convergence Result of Population Size (Case Study 3)	111
Figure 6.31 Convergence Result of Crossover Rate (Case Study 3)	111
Figure 6.32 Convergence Result of Mutation Rate (Case Study 3).....	112

List of Tables

Table 2.1 Existing Optimization Techniques for TCT Problems..... 16

Table 4.1 Variables/Parameters of OBDO Objective Function 49

Table 4.2 Task Crashing Data..... 57

Table 4.3 Comparisons between All-normal and TCT Solutions 59

Table 4.4 Comparisons among All-normal, TCT and OBDO Solutions..... 61

Table 6.1 Sensitivity Analysis of Crossover Rate-1 (Case Study 1)..... 80

Table 6.2 Sensitivity Analysis of Crossover Rate -2 (Case Study 1)..... 81

Table 6.3 Sensitivity Analysis of Mutation Rate-1 (Case Study 1) 82

Table 6.4 Sensitivity Analysis of Mutation Rate-2 (Case Study 1) 83

Table 6.5 Sensitivity Analysis of Population Size-1 (Case Study 1) 85

Table 6.6 Sensitivity Analysis of Population Size-2 (Case Study 1) 86

Table 6.7 Sensitivity Analysis Generation Number-1 (Case Study 1)..... 87

Table 6.8 Sensitivity Analysis of Generation Number-2 (Case Study 1)..... 88

Table 6.9 Sensitivity Analysis of Crossover Rate-1 (Case Study 2)..... 92

Table 6.10 Sensitivity Analysis of Crossover Rate-2 (Case Study 2)..... 94

Table 6.11 Sensitivity Analysis of Mutation Rate-1 (Case Study 2) 95

Table 6.12 Sensitivity Analysis of Mutation Rate-2 (Case Study 2) 96

Table 6.13 Sensitivity Analysis of Population Size-1 (Case Study 2) 98

List of Tables

Table 6.14 Sensitivity Analysis of Population Size-2 (Case Study 2)	99
Table 6.15 Sensitivity Analysis of Generation Number-1 (Case Study 2).....	100
Table 6.16 Sensitivity Analysis Generation Number-2 (Case Study 2)	101
Table 6.17 Cross Sensitivity of Crossover and Mutation Rates	106
Table 6.18 Cross Sensitivity of Population Size and Generation Number	108
Table 6.19 Cost and Schedule Input Parameters of Case Study 3	109

Summary

In this research, a new concept of Overall Benefit-Duration Optimization (OBDO) is proposed with objective function to maximize overall benefit of a construction project in scheduling process. An integrated solution model is developed by applying genetic algorithm and named as Genetic Algorithm Overall Benefit-Duration Optimization (GAOBDO). In order to facilitate practical usage, GAOBDO is coded in VBA (Visual Basic for Application) macro program in Microsoft Project 2003 as an application platform, which automates data input, model structure and outcome interfaces.

Given a normally-planned construction project with minimum cost, under a prerequisite that network compression-incurred opportunity income exceeds cost increment, it is lucrative to compress the project network to a desired duration where overall benefit, which refers to the difference between opportunity income and cost increment, is maximized. This problem is referred to as Overall Benefit-Duration Optimization (OBDO). Unlike time-cost trade-off problems that minimize cost, OBDO is a new scheduling optimization scheme for maximum overall benefit and prone to interests of a project owner. In addition, as a win-win solution, incentive fee is introduced in OBDO as an economic reward to contractor. As a combinatorial optimization problem, OBDO is complex and requires iterative selections of task crashing in network compression. Considering the number of possible iterations involved in this optimization process, optimal OBDO solution is difficult and time-consuming. Integrating extensive searching power of GA in combinatorial optimization problems, OBDO is modeled to maximize overall benefit and accounts for network compression-incurred influences to profitability of a construction project.

After experimenting on various cases drawn from journal papers and real practice, 3 typical case studies of various-sized project networks are applied to demonstrate feasibility of OBDO concept and practicability of GAOBDO model. Sensitivity analysis of four GA parameters, including crossover and mutation rates, population size and generation number, is conducted to provide guidelines for practical applications.

Moreover, cross sensitivity analysis of these four GA parameters is also conducted to explore their contributive effects to optimum OBDO solutions.

GAOBDO model provides a practical solution to maximum overall benefit and is proven applicable to various-sized problems. Research conclusions mainly include OBDO formulation with objective of maximum overall benefit. As a win-win solution, OBDO simultaneously considers profitability to project owner, as well as economic reward to contactor. With easy mechanism and accessibility in resolving combinatorial optimization problems, genetic algorithm is employed in GAOBDO solution model. Moreover, with its friendly interfaces and easy accessibility, Microsoft Project 2003 with embedded VBA programming function is chosen as an application platform to narrow the gap between theoretical research and practical usages. Future research developments include OBDO under uncertainties, OBDO with resource allocation and leveling, GAOBDO model hybridization, and introducing other influencing factors of OBDO in the process of project scheduling.

Chapter 1 Introduction

Overview

This chapter provides a brief introduction of research objective: overall benefit-duration optimization (OBDO) and its genetic algorithm (GA) application in construction project scheduling. It includes background, research objectives, methodology, research scope and thesis organization.

1.1 Background

In process of scheduling, profitability of a construction project is closely influenced by various factors including cost, duration and revenue and so on. Each of these factors is important and deserves simultaneous considerations. One traditional optimization method used in project scheduling is time-cost trade-off (TCT). This method has been applied to minimize cost or duration, considering influencing factors of cost and duration within construction phase. TCT is considered as one of the most important aspects of decision-making in project scheduling. However, it limits itself with objective of minimizing cost or duration. Despite its numerous research in literature and applications in practice, little research has been extended into domain of profitability maximization. In practice, objective of TCT analysis is to balance available resources and their associated cost, and then schedule a project network either within required duration at minimum cost or within budget at minimum duration. However, an owner considers not only total cost within construction phase, but also other related factors that influence profitability of a project after it is completed and commissioned. Considering related influencing factors, a new scheduling optimization scheme based on maximum profitability, which is proposed and named as overall benefit-duration optimization (OBDO), is proposed in this research.

For a construction project financed with bank loans, its profitability fluctuates with

various schedule alternatives. It is not difficult to understand that different schedules lead to different costs and investment allocations (Hastak and Baim 2001). Although schedule acceleration usually increases construction cost, earlier project completion introduces more opportunity income, which includes interest savings and revenue from earlier commission of the project. In such a circumstance, if opportunity income exceeds total cost increment, an owner prefers to compress a project network to maximize overall benefit. This creates an optimization problem named overall benefit-duration optimization (OBDO).

Construction project scheduling involves selecting fitting methods, work forces, equipments and technologies, to complete tasks on network. In general, there is a trade-off between time and cost to complete a task. For a task, on one hand, less expensive resource input takes longer time to finish. On the other hand, applying more resource saves time, but increases cost to finish it. Each option of resource input corresponds to its task time and cost. Each combination of time and cost represents an alternative construction method to finish a task. Each task has several alternative construction methods. Ultimately, selecting alternative construction method at task level decides duration and total cost of a construction project.

In the process of typical construction project scheduling, a project can be planned with normal duration at the least construction cost. Accordingly, time-cost trade-off (TCT) analysis with the objective of minimizing cost is applied. From economic perspective of project profitability, maximum benefit is of much more concern than minimum cost. In practice, to maximize overall benefit, besides duration and cost, more influencing factors need to be considered, which are beyond the reach of TCT analysis. As such, OBDO has different objective from TCT. In order to shorten duration and maximize overall benefit, if OBDO is applied, some tasks have to be crashed on its network.

Scheduling involves trade-offs between times and costs of dozens, or even hundreds of tasks on its network. Each task has several alternative construction methods to finish with associated time-cost combinations. Therefore, considering the number of combinations involved, finding optimum overall benefit is difficult and time-consuming. Moreover, evaluating each possible solution to OBDO requires recalculations of project duration and total cost. Such an exhaustive enumeration is not feasible for traditional methods. Therefore, in order to resolve OBDO problem and

secure optimal solutions to various sized project networks, a new model that integrates OBDO concept with appropriate optimization technique, is highly necessary. Moreover, this model application with clear interaction interfaces in commercial software that can be “computer language coded” and “run in fast computers”, is also highly desirable to end users in process of construction project scheduling.

As a powerful tool for search and optimization technique that has reached a mature stage, genetic algorithm (GA) has been found applicable in various optimization problems. GA works on Darwinian principle of natural selection and is noted as “survival of the fittest”. During the last five decades, GA-based models and their associated applications have been studied extensively to resolve optimization problems in construction project scheduling, such as time-cost trade-off (TCT).

In this research, considering network compression-incurred opportunity income, cost increment and duration, a new scheduling optimization scheme of OBDO is proposed with objective of maximum overall benefit. Furthermore, integrating GA and OBDO, an optimization solution model called GAOBDO is formulated for practical applications. Catering to easy usage of ender users, GAOBDO model is coded in macro programming language VBA (Visual Basic for Application) embedded in Microsoft Project 2003. Applied in this globally used project management software with its scheduling features, GAOBDO solution model can be applied for maximum overall benefit of a construction project in the process of scheduling.

1.2 Research Objectives

Considering influencing factors to project profitability, a new scheduling optimization scheme that maximizes overall benefit is proposed in this research. Integrating GA and OBDO concept, a solution model named Genetic Algorithm Overall Benefit-Duration Optimization (GAOBDO), is proposed to fulfill the following research objectives:

- to review existing techniques of time-cost trade-off (TCT) problems, then indicate their limited objectives;
- to identify characteristics of construction projects and compare different economic perspectives of project cost and overall benefit;

- to identify influencing factors to project profitability, and specify necessity of a new scheduling optimization scheme for maximized overall benefit;
- to propose a win-win solution to both owner and contractor by introducing network compression-incurred overall benefit and incentive fee;
- to formulate objective function of Overall Benefit-Duration Optimization (OBDO) problem, considering associated constraints;
- to apply genetic algorithms (GA) as an optimization technique, and establish a GA-based solution model to OBDO problem;
- to apply GA-based model to case studies, and prove its feasibility and practicability through sensitivity analysis and cross sensitivity analysis;
- to conclude this research and suggest a number of extensions in course of future research development.

1.3 Research Methodology

In order to fulfill the above-mentioned research objectives, various research approaches are applied in this study. Besides literature view and background study, research objectives analysis, concept definition and objective function formation are also applied. Other methods include model formulation, programming development and computer implementation, case studies and sensitivity analysis.

In order to develop an appropriate optimization model to fulfill the objective of maximizing overall benefit, relevant researches and associate optimization techniques are firstly reviewed in literature. For comparison purposes, TCT problems and their research approaches are reviewed with comments to their different objectives from those of this research. As a global searching and optimization technique, GA and its applications in TCT problems are reviewed. It is also suggested as an appropriate solution to a combinatorial optimization problem like OBDO.

With current literature review, research objectives analysis is carried out to address common characteristics of modern construction projects, as well as different economic

perspectives of minimum cost and maximum overall benefit, where main objective of this research are established. In addition, opportunity income is defined as summation of future revenue and interest savings by analyzing economic incentive of network compression for maximum overall benefit. Moreover, as a win-win approach, network compression-incurred incentive fee is introduced as economic reward to contractor. Research objective analysis ends in accentuating a necessity of formulating a new scheduling optimization scheme for maximum overall benefit.

With research objectives being established, a concept of Overall Benefit-Duration Optimization (OBDO) is defined with assumptions. This concept satisfies the need for a new scheduling optimization that maximizes overall profit. After concept definition, objective function of OBDO is mathematically formulated with associated constraints. It is used to integrate an appropriate optimization technique for modeling development.

Based on the formulated objective function, an integrated model called GA-based Overall Benefit-Duration Optimization (GAOBDO) is developed as an appropriate solution to OBDO problem. In this model, GA is employed to facilitate optimization efficiency that maximizes overall benefit. For conveniences, GAOBDO model is developed in macro programming language and implemented in Microsoft Project 2003. As final output, maximized overall benefit, corresponding cost and schedule are gathered and exported with friendly output interfaces for decision making of end users.

To verify its feasibility and practicability, GAOBDO model is applied to case studies. Sensitivity analysis of four GA parameters is conducted to discuss effectiveness of their performances in GAOBDO model. Moreover, cross sensitivity analysis is also conducted to explore contributive effects of these four GA parameters to global optimum solution. GAOBDO model performs effectively with varying combinations of these GA parameters when used on various-sized project networks. Hard guidelines are suggested in selecting the best combination of these GA parameters for various-sized project networks. Conclusions of this research and suggestions of future research extension are provided with guidelines for designing and implementing more effective GA applications to OBDO.

1.4 Research Scope

Research scope in this research is indicated as follows:

- Time-cost Trade-off (TCT) and Existing Optimization Techniques

Since 1960s, TCT has been studied extensively in literature of project scheduling. Existing optimization techniques to TCT, including heuristic, mathematical programming and artificial intelligence methods, are introduced extensively in literature. Heuristic and mathematical programming methods have been applied as two distinct categories of solution to TCT problems. Despite their easily accessible formulations and acceptable solutions for most project networks, heuristic methods can not guarantee optimum solutions. Mathematical programming methods, on the other hand, have been criticized for their cumbersome formulations and computational intensity. Moreover, these two categories of optimization techniques are only applicable to small-sized projects with local minimum solutions. With development in computer technologies, some artificial intelligent techniques, such as neural network, machine learning and genetic algorithm (GA), have emerged as more efficient solution techniques to TCT problems.

- Different Economic Perspectives of Project Cost and Profitability

In project scheduling, regarding project cost and profitability, different economic perspectives exist. Generally, through using more resources, cost tends to increase when a project completes in shorter duration. However, it may be beneficial to generate opportunity incomes as earlier project completion brings more incomes. As far as cost and profitability are concerned, their economic perspectives are inevitably different. In construction stage, a critical economic concern is cost minimization. However, after a project completes, in terms of profitability, main concern is overall benefit. For a project financed with bank loans, it faces a continuously declining cash flow that demands the earliest return on investment. Therefore, optimum solutions to scheduling would be different for minimum cost and maximum overall benefit. In other words, the most economical TCT solution of minimum cost may not be most beneficial to project profitability.

- A New Scheduling Optimization Scheme

In scheduling process, network compression inevitably increases cost and its influence to overall benefit is significant. An owner tends to plan a construction

project according to an accelerated schedule with maximized overall benefit. Reciprocally, accelerated schedule also brings incentive fee as an economic reward to contractor. Unlike conventional scheduling optimization methods that aim to select the most cost-effective way to compress a network within required duration at minimum cost, this research focuses on compressing schedule to a desirable limit where optimized overall benefit is reached. Therefore, this research extends findings of TCT analysis and network compression and develops a new scheduling optimization scheme, facilitating objective of maximizing overall benefit.

- Overall Benefit-Duration Optimization (OBDO)

Previous research on TCT is limited to construction phase, favoring economic perspective of minimum cost. This research extends TCT analysis beyond construction phase; not only considering influencing factors of cost and duration, but also overall benefit in whole lifecycle of project implementation. Given a normally-scheduled project network with a set of tasks to complete according to their precedence relationships, with a prerequisite that network compression-incurred opportunity income exceeds cost increment, the objective is to make a sequence of decisions to crash tasks and hence, compress the project network to a desired duration where overall benefit that equals to opportunity income minus cost increment, is maximized. This problem is referred to as overall benefit-duration optimization (OBDO). Objective function of OBDO, representing overall benefit, is maximized and formulated with constraints. An illustrative example is applied to validate feasibility of OBDO concept. Moreover, in order to resolve OBDO problem of a construction project with complicated network, it is necessary to formulate an effective model with appropriate optimization technique.

- Genetic Algorithm Overall Benefit-Duration Optimization (GAOBDO)

Integrating motivation of learning from efficiency and capability of solving large and complex combinatorial optimization problems, genetic algorithm (GA) is applied to solve OBDO problem. In order to facilitate easy usage of end users, GA-based model is coded in VBA (Visual Basic for Application) as a macro program in Microsoft Project 2003. After experimenting on various project networks, 3 case studies of various sized projects are used to test the applicability of GAOBDO. Sensitivity and cross sensitivity analysis of four GA parameters are

conducted to provide suggestions for implementation of proposed model. The GA-based model overcomes limitations of traditional methods, provides a synergy with more problem-solving power.

1.5 Research Significances

Significances of this research are summarized as follows:

- With characteristics of modern construction project, different economic perspectives of minimum cost and maximum overall benefit are elaborated. TCT problem and its objective of minimum cost are compared with economic perspective of profitability, indicating the necessity of a new optimization concept that aims to maximize overall benefit.
- Together with its objective function, a new concept of OBDO is proposed to fulfill an incentive of overall benefit maximization. An illustrative example is applied to prove the feasibility of OBDO. In order to solve complex OBDO problems, genetic algorithm (GA) is selected as an appropriate optimization technique to establish an optimization solution to OBDO.
- In order to narrow gap between theoretical research and practical usages, GAOBDO model is programmed in VBA and embedded in Microsoft Project 2003. Through customized dialog boxes and friendly interfaces, GAOBDO model is applied with basic understanding of GA mechanism.
- In order to facilitate end users with high efficiency, GAOBDO model is applied to case studies as activated projects in Microsoft Project 2003. Sensitivity and cross sensitivity analysis of four GA parameters are conducted to test their contributive effects to OBDO optimum accessibility. For various sized projects, guidelines of value selection of these four GA parameters are also suggested.
- In the course of future research, the author also suggests a number of possible research extensions of GAOBDO model development.

1.6 Organization of the Thesis

This thesis is presented in seven chapters, as illustrated in Figure 1.1.

Figure 1.1 Organization of the Thesis

Chapter 1 Introduction includes a brief introduction of background and objectives of this research. Moreover, research methodologies, research scope and organization of this thesis, are also outlined in this chapter.

Chapter 2 Literature Review provides an overview of TCT problems and their existing optimization techniques. Comments on traditional minimum cost objective of TCT with its pros and cons are also introduced. In addition, as an appropriate technique to solve combinatorial optimization problems, genetic algorithm (GA) and its applications in TCT problems are also reviewed with comments.

Chapter 3 Research Objectives Analysis addresses characteristics of modern construction project. Then, opportunity income and economic incentive to schedule

compression are defined after addressing different economic perspectives of contractor and owner. Finally, analysis ends in accentuating a necessity of formulating a new scheduling optimization concept for maximum overall benefit.

Chapter 4 Overall Benefit-Duration Optimization (OBDO) introduces a new scheduling optimization scheme as research objective. Assumptions and objective function of OBDO are formulated. Associated constraints and significance of OBDO are also elaborated in this chapter. An illustrative example is applied to prove feasibility of OBDO. Moreover, in order to resolve OBDO problems of various sized projects, a solution model with easy access and integrates appropriate optimization technique is highly necessary for applications of end users.

Chapter 5 GAOBDO Model Formulation integrates powerful searching capacities of genetic algorithm (GA) and OBDO concept, providing a VBA (Visual Basic for Applications) macro-programming model to resolve OBDO problem in Microsoft Project 2003. Computer implementation of the proposed GAOBDO model is also introduced to facilitate applications of end users.

Chapter 6 Case Studies verifies feasibility of OBDO concept and practicability of GAOBDO model with 3 various-sized project networks. Sensitivity analysis is performed to validate effectiveness of GAOBDO model with a self-developed VBA macro program in Microsoft Project 2003. Cross sensitivity and convergence analysis is conducted to explore contributive effects of four GA parameters to OBDO optimum solution. For practical applications to various-sized project networks, it also suggests suitable ranges of value selection for these four GA parameters.

Chapter 7 Conclusions and Suggestions summarizes research accomplishments and original contributions as conclusions. Limitations of this research scope are also presented. Suggestions are also provided as future research extensions for designing and implementing more powerful and practical GA-based applications to OBDO.

Chapter 2 Literature Review

Overview

In this chapter, duration-cost relationship of a construction project, as well as literature of researches on time-cost trade-off (TCT) problems, is elaborated. Overview of genetic algorithm, its applications in TCT and project scheduling are reviewed with summary.

2.1 Duration-Cost Relationship of a Construction Project

A construction project is composed of a network of tasks. Project duration is obtained by sequencing individual tasks on project network. Durations of these tasks are estimated from knowledge of costs allocated to each task and job conditions. Duration and cost of a construction project are inevitably related to each other.

2.1.1 Introduction

Total cost of a construction project consists of direct and indirect costs. Resources allocated to each task of a project determine direct cost. Indirect costs are overhead costs (Adeli and Karim 1997; Assaf *et al.* 2001). In general, total cost is summation of these two partitions and customarily increases, or decreases in certain circumstances, with fluctuations of project durations. Construction projects are large projects in terms of capital requirement; minimizing cost is therefore a primary goal in planning and scheduling of such projects (Adeli *et al.* 2001).

Project planners have to face decisions of selecting appropriate resources, including crew sizes, equipment, methods and technologies to perform tasks of a construction project (El-Rayes *et al.* 2001). Different project scheduling decisions ultimately cause different corresponding durations and costs of the project. For example, using more productive equipment or hiring more workers may shorten project schedule, but cost

could increase. Therefore, selecting the most cost-effective way to complete a project within time and resources limits is desirable for project planners (Alshawhi *et al.* 1999). Due to time-cost relationships of project tasks, it usually takes several iterations to select proper methods, equipment, and crew sizes to obtain acceptable overall project duration (Feng *et al.* 1997). Time-cost trade-off (TCT) analysis can be applied to lower cost without impacting project duration.

2.1.2 Duration-Cost Relationship of a Construction Project

A proper understanding of duration-cost relationship is highly critical to successful implementation of a construction project. Since duration and cost are inevitably related, it can be assumed that a close relationship exists between cost estimate and schedule (Lee and Kyoo 1999; Chang 2001). Fig. 2.1 shows relationship between duration and total cost of a construction project. As shown in Fig. 2.1, total cost is summation of direct cost and indirect cost. Direct cost, in general, has an inverse relationship with project duration. Indirect cost increases linearly with increment of project duration. Total cost is summation of these two partitions and can increase, or decrease along with fluctuations of project durations (Wang *et al.* 2000).

Figure 2.1 Duration-Cost Relationship of a Construction Project

Generally, a contractor prefers to schedule a project with duration at minimum cost, either extending or accelerating project schedule would cause high rises of total cost (Perry and Barens 2000). Economically, it is not wise to compress a schedule beyond duration of minimum total cost as it probably results in more investment of working capitals. It is noted that almost all researches about duration-cost relationship aim at

completing a project within required duration of minimum cost. Their research is limited within construction phase and focuses on minimum cost. As far as profitability is concerned, only considering minimum cost might not be the first priority.

2.1.3 Time-Cost Relationship of a Task

A project network is composed of a series of tasks according to their precedence relationships. Each task has its corresponding time and cost. Besides duration-cost relationship of a construction project, it is also indispensable to consider time-cost relationships of individual tasks on the network. Usually, task duration is decided in integral increments of time called scheduling time units such as days, weeks and months. Normal time of a task represents the time required to finish it under normal circumstances. Associated cost to the normal time is called normal cost, which is the lowest possible cost. It is possible to crash task time below its normal level by applying additional manpower or equipment and improved technologies. Correspondingly, crashing task time causes cost increment. In practice, crashing task is a trade-off approach between expedited task time and cost increment. It is often possible to obtain a relationship between task time and its associated cost. This relationship is determined by so called utility curve that governs time-cost relationship of a task. In process of task crashing, the more scheduling time units a task is crashed, the more cost it causes. In literature, a non-linear time-cost relationship of a convex nature is more appropriate to represent the real-life time-cost relationship of tasks (Nashwan 1998).

Alternative construction methods for each task determine its time-cost relationship. These alternative construction methods include normal, intermediate and crash times, as well as their associated normal, intermediate and crash costs. Normal task time refers to the longest time available to complete a task. In this study, crash time represents the shortest time possible to complete a task. Corresponding cost to this task crashing is called crash cost. It is the highest possible cost of the task. Intermediate times and costs are considered as other alternative construction methods, or other crashing possibilities of the task. As shown in Fig. 2.2, normal time and cost define “Normal Point” and crash time and cost define “Crash Point”. Together with “Intermediate Points” between normal and crash times and costs, they define utility curve of a task. Reda and Carr’s (1999) quadratic utility curve is known as an efficient model for representing actual nonlinear time-cost relations of a task (Li *et al.* 1999).

Figure 2.2 Time-Cost Relationships of a Task

In Reda and Carr’s model, normal time and cost for each task are specified as $T_n(i)$ and $C_n(i)$, respectively. As shown in Fig. 2.2, normal time $T_n(i)$ is the time required to complete task i with normally available resources without any extra resource input. Besides normal time, its associated normal cost is defined as $C_n(i)$. Crash time $T_c(i)$ is the time required to complete task i with extra resources. Crash cost $C_c(i)$ is the cost of completing task i by crashed time. Compressed duration is the time computed from compressed times of tasks. Crash cost for the project is total cost incurred in crashing these tasks. Besides these two times and their associated costs, it is useful to know intermediate time $T_i(i)$ and cost $C_i(i)$. Reda and Carr’s equation and its associated constraint are as follows:

$$C_t = \sum C_i = \sum (a_i T(i)^2 + b_i) \tag{2-1}$$

$$T_c(i) \leq T(i) \leq T_n(i) \tag{2-4}$$

Note: $C_n(i)$ = normal cost for task i $C_c(i)$ = crash cost for task i
 $T_n(i)$ = normal time for task i $T_c(i)$ = crash time for task i
 $T(i)$ = intermediate time for task i $C(i)$ = intermediate cost for task i

Parameters a_i and b_i are derived from values of normal and crash times and costs. A range of possible intermediate times and corresponding costs is based on its time-cost

relationship. Compared to normal time of a task, other available intermediate times and their corresponding costs can be considered as alternative construction methods through network compression. Accordingly, different combinations of alternative construction methods of tasks can be associated with their crashed times and costs of a construction project after network compression. Different combinations of alternative construction methods for tasks can be associated with a project cost and duration. Therefore, determining the best combination of times and costs of tasks with minimum project total cost is the goal of time-cost trade-off.

2.2 Time-Cost Trade-Off (TCT) Problems

Given a construction project network with a set of tasks to be completed according to their precedence relationships to one and another, the objective is to select appropriate resources, including crew sizes, equipment, methods and technologies, and to complete tasks of a project network within required times and minimum costs. This problem is referred to as time-cost trade-off (TCT) problem (Li *et al.* 1999), which is the primary approach to project scheduling and analysis of schedule changes. TCT analysis is a methodology that represents and optimizes construction cost over project duration. TCT assumes that costs are either direct cost of materials, labor, and equipments associated with network tasks or are indirect costs of site supervision, daily expenses, interest, and penalty payments. Direct costs are directly related to individual tasks. Indirect costs are related to project as an entity and are assumed to vary linearly over task duration. Optimization of construction cost and project duration is performed over a set of all tasks with their associated direct and indirect costs. These multi-task optimizations are called TCT problems (O'Brien and Fischer 2000). TCT problems in project scheduling are characterized by constraints on time and cost requirements. Typically, if a project is required to complete before normally planned schedule, TCT analysis can be conducted to crash some of the tasks on critical path of the project network to save time, whilst relaxing non-critical tasks to save duration. Results of this TCT analysis are based on relationship between duration and cost under different time-cost decisions, as well as selection of alternative construction methods that represents optimum compromise of project duration and cost. However, in practice, a real-life construction project consists of dozens, or even hundreds of tasks. Considering the number of possible permutations of alternative construction methods

for individual tasks on project network, obtaining optimum solution to TCT problems is difficult and time-consuming (Liu *et al.* 1995). Moreover, some TCT problems are difficult to solve because they do not have unique optimum solutions (Li 1996).

2.2.1 TCT and Existing Optimization Techniques

Traditionally, TCT analysis is concerned with minimizing project cost while maintaining desired project duration (Easa 1989). There are trade-offs between time and cost to complete each task of a construction project (Feng *et al.* 2000). Difficulties arise because, for hundreds of tasks of a construction project network, there are tremendous possible options of completing all tasks with various combinations of crew sizes, equipment or other resources. This creates a classic combinational optimization problem, which is difficult and confusing for a project planner to identify the best selections of resources that produce minimum cost possible to complete the project. Therefore, objective of TCT analysis is to find the minimum total cost for which the reduction of certain amount of time overrun can be achieved (Li and Love 1997). Existing optimization techniques of TCT are categorized into three areas: heuristic methods, mathematical programming methods, simulation and artificial intelligent methods. A brief review of these optimization techniques is summarized in Table 2.1.

Table 2.1 Existing Optimization Techniques for TCT Problems

	Heuristic Methods	Mathematical Programming Methods	Simulation and Artificial Intelligent Methods
Description	Simple heuristic rules of thumb	Linear, integer, dynamic programming and so on	Simulating genetic evolution and fittest survival mechanism
Principle	Try and repeat	Convert problems into mathematical models	Locate optimum using random yet directed searching operators
Applicability	Small-sized projects	Small-sized projects	Perform well in small/medium projects. Applicable in large projects with extended time
Objective	Single objective	Single objective	Single or multiple objectives
Pros	Simplicity and ease of application	Computational tractable for small-sized problem; Applicable in computer programs	Applicable to combinatorial optimization problems; Robust search algorithm that evolves improved solutions
Cons	Problem dependent; no optimality guaranteed	Complex formulation, computational-intensity, combinatorial explosion	Long processing time for large networks might obscure their robustness
Solution	Single or a few feasible solutions.	Single solution with local optimum	Single optimal solution or a few alternatively near-optimal solutions

Heuristic methods to TCT problems are based on rules of thumb that generally avoid rigorous mathematical efforts. These methods require project planners to visually identify several tasks on critical path and enumerate possible alternatives of allocating times to tasks. Each alternative represents a feasible solution in which tasks are compressed to meet the reduction of project duration. Costs associated with feasible alternatives are evaluated and the best one with minimum cost is selected as final solution to TCT problems. Heuristic methods have enjoyed the majority of research effort because they are applicable to find good solutions with far less computational effort than other methods (Feng *et al.* 2000). It is also noted that these solutions offer good solutions but do not guarantee optimum results.

Mathematical programming methods convert TCT to mathematical models and utilize linear programming, integer programming, or dynamic programming to solve them (Mattila *et al.* 1998). These methods establish project total cost as objective function with constraints, and then employ heuristic searching methods to identify local optimum solutions to TCT. Advantages of mathematical approaches include efficiency and accuracy. However, mathematical models require great computational effort and some approaches do not provide optimal solution either. Especially for large-scaled construction project networks, neither heuristic nor mathematical approaches are capable of obtaining optimal solutions to TCT problems efficiently (Feng *et al.* 2000).

With recent development in artificial intelligence and rapid advances in computer technology, simulation has been used to find optimal solutions to TCT problems in a stochastic network. Many studies focus on estimating project duration and expenditures. Some researches have been conducted to optimize a project in a stochastic network (Moselhi and El-Rayess 1993). However, simulation approaches do not attempt to optimize entire project TCT problem. Originally developed by Holland (1975) and based on the mechanics of natural selection and genetics to search through decision space for optimal solutions, following Darwin revolutionary logics, genetic algorithm (GA) has received increasing interests of researchers who strive to find solutions to TCT problems. During the last four decades, various optimization models utilizing GA have been developed as appropriate solutions to TCT problems.

2.2.2 Network Compression

Typically, when a construction project network is completed before normally planned duration, one method is to crash some tasks on critical path of the project network to save time, whilst relaxing non-critical tasks to save cost. This method refers to network compression, which is primarily concerned with trade-off between crashed task time and consequent increase in total cost. Network compression is originally developed along with critical path method (CPM) for scheduling and controlling construction projects (Abbasi *et al.* 2001). Traditionally, if task durations were crashed, costs of these tasks would increase due to more resources input. It is clear that some tasks can be crashed at a lower cost rate than others. When a choice of task crash is presented, less-cost tasks should be crashed rather than expensive ones. Therefore, objective of network compression is to crash selected tasks to balance direct and indirect costs, and thereby minimize increment of total construction costs until target duration is reached.

Network compression inevitably incurs additional expenditure since more resources are consumed for earlier completion of tasks. Final total cost is the summation of compression-incurred additional cost and total cost for normally planned duration. There are many compression alternatives since each individual task on project network can be possibly crashed. Cost of an alternative to do this is calculated from a time-cost relationship of each task. The least expensive alternative is the best solution (Randolph 2000). This procedure is repeated until the project network has been sufficiently shortened or the cost to shorten project network exceeds benefit derived. However, for owners of commercial construction projects, they prefer earliest return on investment. If network compression-incurred benefit exceeds cost increment, they would probably demand their projects being completed earlier than normally planned duration. Thus, a project planner can perform network compression to reduce project duration.

2.2.3 Literature Review of TCT Problems

TCT problems for construction project scheduling have been investigated since the 1960s. Some research accomplishments in literature are reviewed in this section.

Schedule optimization is one of the primary objectives in TCT problems. Moselhi and El-Rayes (1993) addressed that optimization criterion of existing dynamic programming formulations is limited to minimization of overall project duration. They

presented a model that incorporates cost as an important decision variable in optimization process with objective of overall minimum cost. Based on this objective, Liu *et al.* (1995) further developed a method by combining linear programming and integer programming to efficiently obtain optimal resource selection that optimize time and cost of a construction project. The authors indicated that obtaining an optimal solution with reasonable amount of computational effort is major motivating factor for the proposed method. Their hybrid method, along with spreadsheet tool, provides an efficient means of analyzing TCT problems. In addition, Li (1996) quantitatively analyzed factors that influence overall construction schedule and indicated that optimization must be closely combined with investment allocation, resource supply, and weather. A mathematical model and its associated computer program are established with objective of lowest construction cost. Adeli and Karim (1997) presented a mathematical formulation for highway construction scheduling. Repetitive and non-repetitive tasks, work continuity constraints, multiple-crew strategies, and effects of varying job conditions on crew performance, can be modeled. Nonlinear optimization is then solved by self-developed neural dynamics model that yields optimum schedule for minimum construction cost automatically. By varying construction duration, one can solve TCT problem and obtain global optimum schedule and corresponding minimum construction cost. Considering project scheduling characteristics of precedence relationships, multiple crew-strategies and TCT, Senouci and Adeli (2001) developed a mathematical model for resource scheduling. Different from previous resource scheduling formulations that focused on duration minimization, the proposed model considered cost minimization. Furthermore, unlike resource leveling and resource-constrained scheduling that have traditionally been solved independently, these two objectives are performed simultaneously with a case study under various resource availability profiles. Moreover, Hegazy and Ersahin (2001) indicated that overall schedule optimization, considering time, cost and resource constraints is a difficult task due to inherent complexity of projects. In their study, a spreadsheet-based model is presented for the optimization of overall construction schedule. It also considers difficulties associated with modeling all aspects combined, and inability of traditional optimization tools to solve large-scale problems. Nevertheless, Isidore and Back (2001) pointed out that major shortcoming of the widely used schedule optimization technique (least cost scheduling) lies in the fact that it depends on the deterministic relationship of duration and cost of project activities.

They presented a new procedure of optimal-cost scheduling, which simultaneously applies range estimating and probabilistic scheduling. With a view of systematization, Senouci and Adeli (2001) presented a mathematical model for resource scheduling, considering project scheduling characteristics including precedence relationships, multiple crew-strategies, and TCT problem. Their proposed model considers cost minimization, resource leveling and resource-constrained scheduling of non-repetitive tasks. In their research, nonlinear TCT problem is also resolved using patented neural dynamics model that yields optimum total cost.

Some influencing factors of TCT problems are considered in research, such as time value of money and construction quality. Sunde and Lichtenberg (1995) presented a heuristic TCT model that balances cost, time and resources with a net-present-value (NPV) representation. They indicated that NPV TCT considerably increases more net benefit than a traditional TCT method does. The authors advocate that the method is capable of handling relatively large task networks of projects. Similar to the above-mentioned research, Gardiner *et al.* (2000) indicated that many project planners recognize the need for project control as it applies to cost budgets and time schedule. However, it fails to integrate these two elements and is unable to answer the question of how project control affects project NPV as a result of time effect on cash. The authors proposed that NPV could be used as an ongoing monitor of project time-cost-quality trade-off problems. Similarly, Khang *et al.* (1999) also proposed to integrate quality into traditional TCT problems. Three models are designed in the paper to solve time, cost and quality trade-off problems. Results obtained from these three models, when presented using proper graphics, provide insightful information that helps project managers in making trade-off decisions.

Network compression is one of the major research branches for researchers who strive to solve TCT problems with different methods. Perera (1980) proposed a linear programming solution to network compression. The study aims at making use of durations of each task at minimum cost. Meanwhile, it fulfills CPM analysis objective of establishing a minimum time requirement. Vrat and Kriengkrairut (1986) presented a goal-programming model for network compression with piecewise linear TCT problem. The authors employed a convex type non-linear time-cost relationship approximated by piecewise linear function. Their proposed model considers four goals:

meeting a compressed schedule, ensuring no quality damage, budgeting and minimizing total cost. Furthermore, Reda and Carr (1989) described a pragmatic approach to TCT problems among relative tasks by defining time-cost curves for each task on a project network. They indicated that changes in one task are independent of changes in related tasks as they share methods and resources. Therefore, normal computerized TCT techniques are conceptually wrong and are not useful in practice.

Yau (1989) proposed a heuristic method for compressing a resource constrained project that preserves task schedule. Based on the fact that planned schedule compression can be considered as a reduction of normal experienced time or optimal time for projects. Noyce and Hanna (1997) presented a planned network compression concept file for electrical contractors. Each concept attempts to provide an executable objective for enhancing the construction process and minimizing the impacts of network compression. Laufer *et al.* (1998) reported on a study that examined how construction companies manage to cut housing construction time to half of the normal case. Their paper introduced an integrative approach to network compression and highlighted the role of environment. Chan and Kumaraswamy (2002) explored explanations for possible causative patterns and suggestions for strategies to network compression, on the basis of lessons learned from recent Hong Kong-based surveys and research findings. Specific technological and managerial strategies for reducing construction periods in particular buildings are formulated in order to improve construction time performance. In order to minimize project total cost, Moselhi (1993) presented a method for critical path scheduling that optimizes project duration. In addition, the proposed method could be employed to produce constrained schedules that accommodate contractual completion dates of projects and their milestones.

2.2.4 Development of Solutions to TCT Problems

Traditionally, various heuristic techniques and mathematical methods have been employed to find solutions to TCT problems. Out of the wide variety of mathematical programming models, linear programming, integer-linear programming and goal programming are popular tools used for TCT problems. Mathematical programming models are limited for various reasons. These methods require objective function with constraints being established, but this procedure is time-consuming and prone to errors. Moreover, mathematical programming knowledge is necessary to formulate these

mathematical methods correctly (Olusegun *et al.* 1999). In real practice, few project planners are trained to perform this type of formulation. In order to make them possible to be tackled, TCT problems solved by mathematical methods are over-simplified in most cases. Many researches indicate that their primary disadvantage is that they could not handle bigger and more complicated problems encountered in practice. In addition, solution times for these methods tend to grow exponentially as the number of decision variables increases because of common combinatorial nature of these problems. Even minor variations in small problems may lead to a significant increase in solution times, not to mention optimization problems in large-scale construction projects. Furthermore, chances of obtaining an optimal solution by mathematical method depend on proximity of initial solution to these problems, by which decision-makers are unable to provide.

Besides mathematical methods, due to their simple formats and application efficiencies, various heuristic methods have been used widely in practice. Heuristic methods have advantages of being simple to understand, easy to apply, and inexpensive to use in computer program. For TCT problems, heuristic methods are able to rationalize scheduling process and make it manageable for practical-sized projects. However, heuristic methods perform with varying effectiveness when used on different networks, and no hard guidelines can be helpful in selecting the best heuristic rule for a given network. Heuristic models are problem-dependent so that their rules of thumb can not be applied equally to all construction projects (Leu *et al.* 1999). Heuristic methods manage to obtain good or near optimal but do not guarantee optimal solutions. Moreover, heuristic solutions do not provide possible solution range. For large-scale networks with hundreds tasks and discrete time-cost relationships, which are common in practice, neither heuristic methods nor mathematical programming models can obtain optimal time-cost solutions efficiently (Feng *et al.* 1997).

With development of computer and intellectual technology (IT) in the last four decades, some advanced methods, such as simulation techniques, neural network and revolutionary techniques, had been applied to solve TCT problems with sound performances on large-scale and complicated construction projects. Adeli and Karim (1997) developed an optimization method for scheduling problem with objective of minimum direct construction cost. This branch of TCT problems is then resolved by patented neural dynamics model (Adeli and Park 1996, 1998; Park and Adeli 1997) that

automatically obtains optimal schedule with minimum construction cost. Based on all these optimization method formulations and patented model, particularly suitable for highway construction change order management, Karim and Adeli (1999 and 2001) developed an object-oriented (OO) information model for construction scheduling, cost optimization, and change order management (CONSCOM). As one of the most significant applications in the area of construction scheduling, applying Microsoft Foundation Class library in Visual C++, the OO model has been implemented in a prototype software system for management of construction projects. CONSCOM is used by the owner as an intelligent decision support system in schedule reviews, progress monitoring, and TCT analysis for change order approval. Unlike previous resource scheduling problem that focused on minimum duration, Senouci and Adeli (2001) developed a mathematical model with TCT objective of minimum total construction cost. Also applying patented neural dynamics model of Adeli and Park (1998), the proposed method considers TCT and was implemented in a FORTRAN computer program. It is able to perform resource scheduling efficiently regardless of construction project size.

Recently, as a global searching and optimization technique, mimicking Darwin's revolutionary genetic logics with programming-friendly characteristic, genetic algorithm (GA) has been developed as an appropriate solution to TCT problems.

2.3 Overview of Genetic Algorithm (GA)

Genetic algorithm (GA) is a search algorithm based on mechanics of natural selection and natural genetics. It combines survival of the fittest among string structures with a structured yet randomized information exchange to form search algorithms with some of the innovative flair of human search (Goldberg 1989).

2.3.1 General Introduction

GA is an example of a search procedure that uses random choice as a tool to guide a highly exploitative search through a coding of parameter space (Tam *et al.* 2001). In parameter space, a new set of artificial creatures is created using bits and pieces of the fittest of the old. An occasional new part is tried for good measure. While randomized, GA is no simple random walk. It efficiently exploits historical information to speculate

on new search points with expected improved performance. As the name suggests, these processes observed in natural evolution inspired GA. It mimics these processes and utilizes them for solving a wide range of optimization problems. GA performs directed random searches through a given set of alternatives with an aim of finding the best alternative with respect to given criteria of goodness.

As a powerful and broadly applicable stochastic search and optimization technique, GA is perhaps the most widely known type of evolutionary computation method applied in optimization problems today. In the past few decades, GA and its associated applications has turned much of its attention to optimization problems in industrial engineering, resulting in a fresh body of research and applications (Gen and Cheng 2000). It surpasses its more traditional cousins in quest for robustness and differs in some very fundamental ways. Basically, GA is different from more normal optimization and search procedures in four ways (Goldberg 1989):

1. GA works with a coding of parameter set, not parameters themselves;
2. GA searches from a population of points, not a single point;
3. GA uses payoff (objective function) information, not derivatives or other auxiliary knowledge;
4. GA uses probabilistic transition rules, not deterministic rules.

Unlike many other methods, GA uses probabilistic transition rules to guide its search. Random choice is used as a tool to guide a search toward regions of search space with likely improvements. These four differences, including direct use of a coding, search from a population, blindness to auxiliary information and randomized rules, contribute to robustness of GA and resulting advantages over other more commonly used techniques (Dandy and Engelhardt 2001).

2.3.2 Operators and Process of Typical GA

GA is a stochastic search technique derives from mechanism of Darwinian natural selection and natural revolution. In typical GA, possible solutions to an optimization problem are represented as a population of chromosomes. Each chromosome stands for a potential solution to the optimization problem. In terms of GA, number of chromosomes in a population is referred to as population size. These chromosomes reproduce through consecutive generations. GA operators applied to parent

chromosomes then generate offspring chromosomes. In each generation, chromosomes are evaluated on their performances with regard to their fitness functions (i.e., objective function of the optimization problem). Chromosomes with better fitness values have higher survival probabilities of being introduced into next generation. After a number of generations, chromosomes in new generation may be identical, or certain termination criterion is met. Final chromosome hopefully stands for optimal solution to optimization problem. Generally, typical GA that yields good results in optimization problems consists of the following three operators (Adeli and Hung 1995).

1. Reproduction

Reproduction is a process in which individual chromosomes are copied according to their fitness values. Fitness value can be considered as some measure of profit, utility, or goodness that needs to be optimized. Copying chromosomes according to their fitness values ensures that better chromosomes with better values have higher probabilities of contributing one or more offspring in next generation. This operator, of course, is an artificial version of natural selection. Reproduction may be implemented in algorithmic form in a number of ways. The easiest way is to create a biased roulette wheel where each current chromosome in the population has a roulette wheel slot sized in proportion to its fitness value.

2. Crossover

After reproduction, power of GA arises from crossover where a randomized exchange of genetic material on selected chromosomes is executed with a possibility that “good” solutions can generate “better” ones (Goldberg 1989). Take one-point crossover as an example; crossover may proceed in two steps after reproduction. First, members of newly reproduced chromosomes in mating pool are mated at random. Second, each pair of chromosomes undergoes crossover as follows: an integer position along the chromosome is selected uniformly at random between the first and the last chromosome position. Mechanics of reproduction and crossover are simple, involving random number generation, string copies, and some partial chromosome exchanges. Nonetheless, combined emphasis of reproduction and structured, though randomized information exchange of crossover give GA much of its power (Man *et al.* 2001).

3. Mutation

Mutation is applied to produce an offspring chromosome by introducing one or more random alterations in one chromosome. Mutation provides a means of introducing new genetic information into current population and plays a decidedly secondary role in GA process. However, mutation is necessary because, even though reproduction and crossover effectively search and recombine extant solutions, they may occasionally become overzealous and lose potentially useful genetic material. Mutation protects against such an irrevocable loss since it is occasional (with small probability) random alteration of value of a gene position on a chromosome. By itself, mutation is a random walk through gene space. When applied sparingly with reproduction and crossover, it is an insurance policy against premature loss of important solutions. Theoretically, crossover is employed in virtually all types of GA, but mutation is sometimes omitted. The role of mutation is to produce new chromosomes not on the basis of fitness function, but for the purpose of avoiding a local optimum. When mutation is employed, it is usually chosen with small probabilities. Mutation position, as well as chromosome mates in crossover, is selected randomly. When GA terminates, chromosome with highest fitness represents the optimal solution.

In terms of GA solutions to optimization problems in domain of project scheduling, these three operators have been proved to be both computationally simple and effective in resolving a number of optimization problems. GA searches the best alternative through chromosome evolutions. It is required that a set of alternatives to be searched through be finite. If a set of alternatives is applied to an optimization problem where this requirement is not satisfied, involved set has to be discretized and an appropriate finite subset needs to be selected. It is also required that the alternatives be coded in strings of some specific finite length, which consist of symbols from some finite alphabets. In terms of GA, these strings are called chromosomes; these symbols are called genes, and their set is called gene pool. Normally, GA formulation and solution processes adopted for optimization problems include following steps:

1. Initialize potential solution structure to optimization problem;
2. Code solutions in form of strings represented as chromosomes population;
3. Define GA parameters (crossover rate, mutation rate, generation number,

- population size and so on.) and set stopping criteria or termination conditions;
4. Evolve and create offspring chromosomes by merging two parent chromosomes by crossover and modifying chromosomes by mutation;
 5. Evaluate chromosomes of successive generations on their performance with regard to fitness function, until stopping criteria are met; and
 6. Decode evolved solution chromosomes into optimum solution.

Figure 2.3 Process of Typical Genetic Algorithm

Basic steps in process of typical GA are shown in Fig. 2.3. Firstly, an initial population of chromosomes is randomly selected. Next, each chromosome in population is evaluated in terms of its fitness value. Fitness value normally refers to as objective function of optimization problem. Then a new population of chromosomes is selected from given population by giving a greater change to select chromosomes with higher fitness values. Nevertheless, new population may contain duplicates. If given stopping criteria are not met, specific crossover and mutation operations are performed on chromosomes of new population. These operations produce new chromosomes called offspring. Same steps of this process, evaluation and natural selection are then applied to chromosomes of resulting population. This whole process is repeated in a number of

generations until given stopping criteria are met. Finally, the best chromosome in final population represents the optimal solution.

2.3.3 Mathematics of Typical GA

There are many mathematical variations of GA. To describe typical GA in detail, let G denote gene pool and n denote length of gene strings that form chromosomes. Population size of chromosomes is usually kept constant during GA execution. When new members are added to population, corresponding number of old members is excluded. Let m denote population size and f denote fitness function. Typical GA, which is iterative, consists of the following six steps:

1. Select an initial population $p^{(k)}$ of a given population size m , where k begins from $k = 1$. This selection is made randomly from gene pool G . Choice of value m is important. If it is too large, GA may not differ much from an exhaustive search. If it is too small, GA may not reach optimal solution;
2. In terms of fitness value, each chromosome in population $p^{(k)}$ is evaluated by determining value of fitness function $f(x)$ for each chromosome x in population;
3. Generate a new population $p_n^{(k)}$ from given population $p^{(k)}$ by reproduction. One procedure called roulette wheel selection is described here, which is referred to as deterministic sampling. Accordingly, calculate fitness value $f(x)$ for each chromosome x in population $p^{(k)}$, then select survival chromosomes with probability defined by $f(v) / \sum_{x=1}^{pop_size} f(x)$. Where $f(v)$ is fitness value of chromosome v . Obviously, chromosomes with higher fitness values will have higher probabilities of being selected into next generation. This step is to eliminate chromosomes with low fitness and duplicate those with high fitness;
4. If stopping criteria are not met, go to step 5; otherwise, stop;
5. Produce a population of new chromosomes $p_n^{(k+1)}$ by operating on chromosomes in population $p_n^{(k)}$. This step includes some or all of the following operations:

- Crossover: Given two chromosomes,

$$x = \langle x_1, x_2, \Lambda, x_n \rangle,$$

$$y = \langle y_1, y_2, \Lambda, y_n \rangle$$

Integer $i \in N_{n-1}$ is called crossover position; crossover operation is applied to x and y , then replaces these chromosomes with their offspring,

$$x' = \langle x_1, \Lambda, x_i, y_{i+1}, \Lambda, y_n \rangle,$$

$$y' = \langle y_1, \Lambda, y_i, x_{i+1}, \Lambda, x_n \rangle.$$

- Mutation: Given a chromosome $x = \langle x_1, x_2, \Lambda, x_n \rangle$ and an integer $i \in N_{n-1}$, which is called a mutation position, operation of mutation replaces x with

$$x' = \langle x_1, \Lambda, x_{i-1}, z, x_{i+1}, \Lambda, x_n \rangle,$$

Where z is a randomly chosen gene from gene pool G ;

6. Replace population $p_n^{(k)}$ with population $p_n^{(k+1)}$ produced in Step 5, increase k by one, and go to step 2.

GA converges quickly on optimal solutions after examining only a small fraction of search space. GA has been applied successfully to complex optimization problems in construction management and related areas. Recently, applications of GA have infiltrated the field of optimization problems in construction project scheduling.

2.3.4 GA for Optimization Problems in Construction Project Scheduling

Typical GA can possibly solve optimization problems that satisfy following criteria:

- Traditional heuristic and mathematical methods are inadequate or inefficient;
- Optimization problems are poorly understood, or very complicated as possible solution space is too large to analyze in a finite time;
- Additional information guiding the search is absent or not sufficient;
- Problem solutions can be coded in form of a string of characters; and
- Need for near-optimal solutions quickly to be used as starting points for other conventional optimization methods.

Based on natural selection and genetics in reproduction, GA is considered as a promising means for globally searching optimal solutions in a large problem domain. Some researches have been done in optimization problems of construction project scheduling using GA (Chan *et al.* 1996). For complicated optimization problems in construction project scheduling, such as TCT problems, where problem structure is not apparent or too complex, traditional heuristic and mathematical methods are far from ideal solutions. They are either problem-dependent with no guaranteed optimal solutions, or computational time consuming. In addition, for optimization problems in scheduling of a large-scale construction project, neither heuristic nor mathematical models are able to secure optimal solutions effectively. GA seems to be a better choice as it requires only objective function and solution boundary represented as a set of constraints. Moreover, GA tends to perform better for large-scale optimization problems and efficiently reaches to optimal solution. GA is robust, global and generally more straightforward in situations where there is little or no priori knowledge about the process to be controlled. This is crucial to many optimization problems in construction project scheduling where precise definition of the problem structure is too complex to illustrate. As one of the early GA applications to optimization problems in civil engineering, Adeli and Cheng (1993) applied GA to space structures optimization. Integrated GA with penalty-function method, GA was implemented in C language and applied to optimization of three space truss structures. Moreover, these two authors (1994) developed a hybrid GA for optimization of space structures using augmented Lagrangian method. It is noted that GA is not line search and avoids resolving optimization problem by computing derivatives of objective function and constraints. With their research, the authors advocated that GA is general and can be applied to a broad class of optimization problems.

2.4 GA Applications in TCT Problems

GA can be used to find sub-optimum, if not optimum solutions to a particular optimization problem (Al-Tabtabai and Alex 1999). Complicated problems in a specific optimization domain can be tackled effectively with a very modest knowledge of the theory behind GA (Dhanasekar 2000). In research field of TCT problems, GA applications have increased dramatically in the last few years.

2.4.1 Introduction

Optimization problems in construction project scheduling mainly involve a maximization or minimization problem subject to various influences and constraints that affect the decision (Al-Tabtabai *et al.* 1999). A project planner has to make decisions frequently in a construction project scheduling. However, defining and evaluating all feasible combinations of solutions based on constraints become cumbersome when the number of possible solutions increases. For TCT problems in scheduling of a real-life construction project, where the structure is either complicated or not apparent, GA seems to be a better choice as it requires only objective function and solution boundary represented as a set of constraints. Through literature review of existing research on TCT, GA is good at searching optimal solutions.

2.4.2 Background and Existing Research in GA

Recent researches of GA applications in TCT problems are reviewed in this section.

Feng *et al.* (1997) developed a new GA-based algorithm for TCT decisions. The algorithm demonstrates its effectiveness by searching only a small portion of total search space, providing an attractive alternative to TCT problems. To facilitate the GA-based applications in practice, a computer program called TCGA (Time-Cost GA) is developed as a practical tool for practitioners. However, Li and Love (1997) indicated that computational time required to complete medium to large-scale projects could be so long that GA advantages could be overridden. The authors adopted an improved GA to facilitate TCT problems with objectives of reducing computational time and increasing result reliability. The improved GA system is tested against conventional GA system to indicate its better performance in solving TCT problems. It also generates a whole class of alternative solutions close to the optimum.

To solve TCT problems and overcome its shortcoming of manually crafting linear time-cost curves of individual tasks, Li *et al.* (1999) presented a computer system called MLGAS (Machine Learning and GA-based System). In the system, a quadratic template is introduced to capture “non-linearity” time-cost relationships of tasks. Machine-learning method generates quadratic time-cost curves from historical data and also measures the credibility of each curve. The quadratic curves are then used to formulate objective function solved by the proposed GA model. Additionally, several

improvements are made to enhance GA capacity. A case study results from MLGAS is compared with those from an experienced project manager demonstrated that MLGAS generates better solutions to nonlinear TCT problems. In order to resolve real world problems in optimization process, Que (2002) presented an approach that makes GA-based TCT viable. In his research, practicability is incorporated through integration of a project management system to the proposed GA-based system. This approach ensures that all scheduling parameters are considered in determining project completion date, thus allowing comprehensive and realistic evaluations to be made.

Primary disadvantage of traditional analytical solutions to resource scheduling problems is that they could not solve bigger and more complex problems in large-scale projects effectively. GA has been employed to overcome this drawback. Thanks to GA strength of selection and recombination to learn large domain of specific project network, Chan *et al.* (1996) proposed a GA model that seems to be an appropriate alternative to resource scheduling problems. The proposed GA model is able to handle resource allocation and leveling, as well as “due-date” problems without extensive modifications. Unlike traditional algorithms that assume unlimited resources availability, Chua *et al.* (1997) presented a GA-based model to solve TCT problems with consideration of resource constraints. TCT problem is considered in terms of resource level to be deployed for each activity. GA searches both spaces of resource utilization and task starts to determine the optimal schedule that conforms to available resource profile. Moreover, Leu and Yang (1999) proposed a GA-based multi-criteria optimal model that integrates TCT, resources constrained allocation and unlimited resource leveling. Their GA-based scheduling model is capable of considering three categories: TCT, resource-constrained allocation, and resource leveling, respectively and simultaneously. Dandy and Engelhardt (2001) applied GA to find a near optimal schedule for water supply pipes replacement. Their goal is to minimize present value of capital repair and damage costs. The authors believe that the complexity of problem is increased to include the scheduling within budget constraints. However, GA is still able to identify optimal solution within limited project funds. With development of computer and programming technologies, Hegazy (1999) developed a procedure that searches for a near-optimal solution to resource allocation and leveling optimization simultaneously. In his study, a GA-based multi-objective optimization model is described and coded in a macro program. GA performance of the proposed procedure

is then evaluated on a case study with recommendations of further researches.

Considering uncertainties of task time and optimal allocation of limited resources with objective of minimum duration, some researches focus on hybrid solutions that integrate GA and uncertainty theories. Leu *et al.* (1999) adopted a new approach, integrating GA and fuzzy set theory to solve resource-constrained scheduling problem under uncertainties. In their research, fuzzy set theory is used to model uncertainties associated with time elements in a project networks. In order to achieve objective of minimum duration under uncertainties, GA is used to allocate multiple available resources to activities. Feng *et al.* (2000) presented a hybrid approach that combines simulation techniques and GA to solve TCT problems under uncertainty. They integrated GA with simulation techniques and provided an efficient and practical way of assessing project time and cost risks. Their approach provides project planners with a new way of analyzing stochastic construction project TCT decision in a more realistic manner. Leu *et al.* (2001) proposed a GA-based fuzzy optimal model for TCT problems, while considering effects of uncertain activity duration. In addition, fuzzy set theory is used to model uncertainties of task times. GA is adopted to search for optimal TCT profiles under different risk levels. The method provides an insight into optimal balance of time and cost under different risk levels defined by decision-makers.

2.5 GA Applications in Construction Project Scheduling

Certain issues of GA applications in construction project scheduling optimization are also reviewed in following sections.

2.5.1 GA Hybridization

From above-mentioned review, scheduling optimization models can be classified into two areas: (1) deterministic scheduling and (2) nondeterministic scheduling. Traditional heuristic and mathematical methods mostly focus on deterministic situations. However, none of these methods is able to solve non-deterministic problem where the structure is not apparent or uncertain. GA seems to be a better choice since it requires only objective function and initial solution pool to start with. However, most GA-based methods to scheduling optimization problems can not guarantee global

optimum solutions within efficient time frame. More recently, hybridizations of GA and other optimization techniques have been studied. In TCT problems, a hybrid approach of machine learning and GA is developed to deal with non-linear time-cost relationships of tasks (Li *et al.* 1999). As a subset of artificial intelligence research, machine learning is concerned with automatic generation and measurement of quadratic time-cost curves from historical data or solutions, which are then used to formulate objective function of subsequent GA procedure. Feng *et al.* (2000) hybridized simulation technique and GA to solve TCT problems under uncertainty. Simulation technique is used to provide efficient way of assessing time and cost risks. Their hybrid approach provides an innovative means of making stochastic TCT decisions in a more realistic manner. GA-based hybrid method with fuzzy set theory is also developed to consider both uncertain task time and TCT (Leu *et al.* 2001). In their model, GA is adopted to search for optimal TCT profiles under various risk levels, which are introduced by fuzzy set theory through modeling uncertainties of task times.

In category of resource-constrained construction scheduling, simulation and fuzzy set theory are also hybridized with GA to enhance their problem-solving capabilities. Leu *et al.* (1999) proposed a fuzzy optimal model for resource-constrained scheduling, in which fuzzy set theory is used to model uncertainties of activity durations. Then GA is adopted to search for fuzzy optimal project duration under resource constraints. Leu and Hung (2002) hybridized GA and Monte Carlo simulation and proposed an optimal scheduling model that considers both activity duration and resource constraints. In Monte Carlo simulation procedure, various probability distribution functions are used to generate optimal project duration profiles. GA is adopted to search for probabilistic optimal project duration under resource constraints. From above mentioned researches, it is noted that optimization problems encountered in construction project scheduling that satisfy the following criteria can be possibly solved with GA hybridization.

1. Sole GA-base techniques are inadequate;
2. Problem is very complicated or the possible solution time from GA is unreasonably consuming;
3. Additional information to guide GA search is unavailable or insufficient so that sole usage of GA method is not practical;

4. Possible for GA to generate coarse solutions quickly as triggering point for further refinement of hybridized optimization method;
5. Necessary to use a particular optimization model as a benchmark to evaluate the performance of GA-based searching for more problem-solving power.

GA can be readily hybridized with other optimization techniques to solve a complex optimization problem under particular circumstances. For large and complex combinatorial optimization problems, GA is not able or time-consuming to handle alone. But they are good at finding near-optimal solutions quickly. Rather than completely relying on GA, these near-optimal solutions from GA can be a starting engine for other optimization techniques to generate global optimal solutions. It is suggested that such hybridization could increase convergence possibility and efficiency of optimal solutions with more problem-solving power.

2.5.2 GA Applications in Commercial Software

In order to narrow gap between research and practical implementations, viable and solid GA-based optimization tools that facilitate convenient usage in practice have been developed. Based on the logic of “A good algorithm that is too difficult or too abstract to use is of little use to practitioners (Feng *et al.* 1997)”, many GA procedures were integrated in commercial software including Microsoft Excel, Microsoft Project, MATLAB and Primavera Project Planner (P3). Ting and Pan (2003) applied GA-based procedure of overall benefit-duration optimization in MATLAB. Interactive programming environment of the software allows GA procedure to manage its objective function, variables; import and export data; perform calculations; generate plots and figures; develop and manage files. GA solution from MATLAB overcomes limitations of traditional methods and presents a synergy with more problem-solving power. Feng *et al.* (1997) used Microsoft Excel as a front end of their proposed TCGA model for data entry, handling TCT problems. The macro GA program implanted in Microsoft Excel automates execution of TCGA, providing a practical tool for practitioners to apply optimization algorithm in practice. Also relied on powerful programmability and on-going control ability of Microsoft Excel, Hegazy and Petzold (2003) developed a genetic optimization for dynamic project control. Microsoft Excel extends searching capability of embedded GA procedure designed for project planning,

providing functions such as project tracking, progress reporting and corrective action optimization. Que (2002) incorporated practicability into GA-based time cost optimization. In his study, Primavera Project Planner (P3) is chosen for the project management system. With its powerful project scheduling functionality, P3 makes GA-based time-cost optimization viable for practical usage by facilitating comprehensive and realistic evaluations to be made in optimization process. In Hegazy's (1999a) GA-based optimization model of TCT, MS Project is selected for implementing the GA procedure due to its easy usage and programmability features. Using VBA (Visual Basic for Application) macro language of MS Project, GA optimization procedure is coded to facilitate construction planners with an automated tool that integrates the TCT model into their familiar commercial software. Moreover, Hegazy's (1999b) integrated a GA procedure with resource management features of MS Project for resource allocation and leveling. Coding GA procedure with macro programming language, it provides an automated evaluation of time and cost associated with each construction method of, in which resources are transferred to the scheduling software for resource allocation and leveling. From the above review, benefits of GA applications in commercial software can be concluded as follow:

1. It narrows gap between academic research and practical usage with integration of abstract GA methodology and widely used commercial software;
2. GA procedures are faster and easier to be programmed by users who are familiar with programming features of commercial software packages, which might be able to counteract drawback of computationally time-consuming feature of GA;
3. With friendly interfaces, integrated planning and control, and Internet capabilities in these software packages, GA applications ultimately result in actual efficiency improvement and time saving in optimization process.

2.5.3 Computational Times in GA Applications

Among optimization problems in construction project scheduling, various GA-based applications have been recognized as effective and applicable in small and medium-size projects. However, for large-scale projects, GA applications may consume much computational time that can obscure or even counteract the effectiveness of GA. Li and Love (1997) pointed out that major drawback of GA applications is that they require

much greater computational time than traditional methods. Computational times of GA methods depend on programming language used, computer platform, chromosomes and operators, population size, generation number and so on (Gad 1991). As GA search requires substantive analysis, computational time required in optimizing might be so high and overrides its advantages. It is suggested that following options can be adopted to improve time-efficiency of GA applications.

1. Code GA applications in faster programming language to improve time-efficiency of GA procedure;
2. Apply GA procedures in faster computers with parallel processing abilities. Faster computers perform large number of evolution and analysis in a short time, which is more beneficial to time-consuming GA applications;
3. Modify or introduce specific GA operators for particular optimization problems. Specially modified GA operators have been proved to be an option of reducing computational time and enhance result reliability.

2.6 Summary

Before embarking on a new research, main reasons of literature review include: 1) to provide background and justification for the research undertaken; 2) to review and comment on previous ground-breaking work in related research area; 3) to identify limitations of previous works and differentiate itself from them; 4) to build a theoretical foundation upon where this new research is based.

In this chapter, since the main research objective is overall benefit-duration optimization, duration-cost relationship is inevitably discussed in details. Together with existing problems of scheduling optimization and solution models, literature related to TCT is then reviewed. Research of three categorized optimization techniques for TCT problems, including heuristic, mathematical programming and artificial intelligent methods, is also introduced with comparisons of their pros and cons. As a promising optimization technique, GA is introduced in details including typical operators, process and mathematical foundations. Moreover, GA applications in TCT and other optimization problems in project scheduling are also reviewed in aspects of

hybridization, commercial software integration and computational time consumption.

TCT problems are concerned with minimizing construction cost while maintaining desired duration with available resources. Through literature review, heuristic, mathematical and GA-based methods to TCT have common objectives as follows:

1. TCT problems are characterized by constraints on time and cost with common objective of completing a project within minimum duration or minimum cost;
2. TCT analysis identifies the best selection of crew size and equipment that produces optimized duration without exceeding available resource limits. Existing solutions seem to provide very little support for objective of maximizing benefit;
3. Consider optimum solution within construction phase. Given a construction project, the objective is to complete the project within required duration in minimum cost. Profitability is beyond the concerns of the above-mentioned solutions.

In order to obtain maximum benefit, it needs a life-cycle schedule based on simultaneous considerations of influencing factors including revenue, duration, cost, which are more complicated than those considered in TCT problems. In this research, for the sake of project profitability, it aims to introduce more influencing factors that are considered beyond construction phase. Therefore, a new concept and its associated solution model that deal with optimization of project profitability, are highly desirable.

Chapter 3 Research Objectives Analysis

Overview

In this chapter, characteristics of construction projects are analyzed. Research objectives are formulated by analyzing different economic perspectives of cost and overall benefit. Incentive to schedule compression and necessity for a new scheduling optimization scheme are also elaborated.

Unlike TCT objective of scheduling at minimum cost, for a construction project with objective of maximum profitability, it concerns not only total cost and duration, but also other influencing factors including interest payment and future revenue.

3.1 Characteristics of Construction Projects

Construction projects are unique and vary in a number of characteristics including capital requirement, financing and construction method, which closely influence project cost, duration and profitability. Main characteristics of construction projects that are closely related to research objectives of this study are summarized as follows:

1. Capital Intensity

A construction project needs to be completed with substantive amount of resources including equipment, manpower and materials. Dozens or even hundreds of tasks on network have to be fulfilled synchronously. Each task has to be constructed with its corresponding working capital. Therefore, completing such a construction project is capital intensive.

2. Long-Term Implementation

Development of a construction project spreads over a long period of time. It involves completions of tasks on network with various construction methods. Due to technical reasons, an individual task on network can not be started after its

precedent task is completed. Therefore, completing all these tasks makes a modern construction project long-term implementary.

3. Project Network Complication

There are many different tasks involved in a schedule network of a construction project. Each task has several alternative construction methods to finish and has dependent or independent interrelationships with other tasks on network. Therefore, network of a construction project is complicated.

3.2 Different Economic Perspectives of Project Owners and Contractors

With inevitable increase in size and complexity of a modern construction project, in terms of project cost and benefit, comprehensive understanding of all associated economic aspects of the project is necessary. A project must be implemented in a high degree of cost and schedule control, not only in terms of schedule progress and cost, but also over the influences arising from external economic environment.

In the implementation of a modern construction project, a number of objectives exist regarding project financing courses and economic perspectives of practitioners (Harris and McCaffer 2001). For instance, similar to above-mentioned TCT problems, meeting a certain contracted date, controlling expenses to a fixed budget, minimizing total project cost and maximizing project profitability can all be objectives. However, in a real-life construction project, different economic perspectives of participants are incommensurate and often conflicting in terms of project cost, duration and profitability. Owners and contractors have different economic perspectives, which demand an appropriate balance between project costs and durations (Khosrowshahi 1997). A contractor has objective for his own cost control, which will differ from those of an owner's project management (Pilcher 1994; Chan 2001). As far as project cost and duration are concerned, economic priorities of contractors and owners inevitably differ. Given a construction project with a network of a set of tasks to be completed according to some precedence relationships, a contractor's objective is to complete tasks of a project within required duration at minimum cost (O'Brien and Fischer 2000). In such a circumstance, a construction solution with minimum total cost is the first priority to a contractor. TCT problems and their related methods are proposed to fulfill common

economic perspective of minimum cost within contracted duration.

On the other hand, completing a construction project with minimum cost may not be the most important economic perspective to a project owner. Except factors of construction safety and quality, in some circumstances, profitability catches more attention of owner. Although normal planned schedule with minimum cost is beneficial, it might not be the first economic concern for an owner who strives to obtain maximum project profitability. If a commercial construction project is financed with bank loans, comparing to TCT objective of minimum cost, owner faces a decision-making of whether accelerating project schedule would be more beneficial as it introduces earlier income flowing-in and more interest savings. In today's fast-paced construction industry environment, project owners have been increasingly placing greater demands on contractors to complete projects earlier (Rosenfeld *et al.* 1998). This is because time is critical for owners in obtaining 'first in the market' advantage over competitors. Owners will most likely suffer the loss of expected benefits as a direct result of delays in putting the facility or product into service (Kog *et al.* 1999; Abeyasinghe 2000). Therefore, a traditional project scheduling with minimum cost, such as TCT, might not cater to economic perspective of maximum project profitability.

In present inflationary economic environment, time is often the essence for the profitability of a construction project (Bakens 1997). Economic viability of a construction project is crucially dependent on shortening duration through network compression at expense of increasing cost by a minimized percentage. Therefore, if compression-incurred income exceeds increased cost, network compressing can be more lucrative for owners to generate more overall benefit. An owner may desire shorter project duration rather than demanding minimum construction cost, which might be different from TCT objective of minimum cost. Therefore, an owner's demand on maximum profitability from a project can not be realized by TCT solutions.

3.3 Opportunity Income and Economic Incentive to Schedule Compression

Modern construction projects with complicated networks are capital-intensive and long-term implementary. If a construction projects is financed with bank loans, early project completion means reimbursement of bank interest in advance and more revenue

Chapter 3 Research Objectives Analysis

flowing in. Summation of these two parts is defined as opportunity income. Therefore, capability of collecting opportunity income due to early project completion definitely influences profitability of a project.

In this research, opportunity income stands for summation of interest saving and future revenue from accelerated project completion. Interest saving is also considered as financing charges, which is the interest that must be paid every planning time unit on expenditure to date, until the construction project is producing an income or is ready for use (Ferry *et al.* 1999). Investment in a construction project, although it does not earn any profit before completion, it has potential of earning interest saving if the project is finished earlier than previously planned duration. In a large-scale construction project, interest saving from expedited schedule can reach to a significant amount. On a billion-dollar hydro power plant project with an estimated duration of 10 years, the funds invested during the first year at 7% interest will double by the end of the project (Ahujia 1976). A project has no return from investment until it is commissioned and produces income. Therefore, an owner wants the project finished in the shortest time that his financing ability will allow (Ahujia 1980). This involves decision-makings of project scheduling that directly influences project cost and profitability. Since capital has been spent on development, it is not available to invest elsewhere and is not accumulating interest. Interest saving and potential income from project expedition offers its owner an opportunity income. A construction can be shortened so the project can be commissioned earlier than normally scheduled and put facility into use at the earliest date. As strategic thinking in process of scheduling, an owner might focus on generating value added services for earlier commissioning of the project. Extra sources of opportunity income and project scheduling solutions that improve overall benefit will be sought instead of minimizing cost. Inevitably, with additional input of material, workforce and equipment, earlier completion of project also causes cost increment. In real practice, in order to encourage a contractor to complete project ahead of normally planned time, in some contractual clauses, an owner has option to pay contractor incentive fee for earlier finish. Contractual incentive fee enables the owner to incorporate the desire of project expedition into practical project scheduling, which is normally realized by network compression. After network compression, total project cost increment is composed of increased construction cost and incentive fee paid to contractor. In process of network compression, if the amount of opportunity income

exceeds total cost increment, each compression leads to its corresponding opportunity income and total cost increment. Thus, when compression-incurred opportunity income amounts to a considerable sum, it is beneficial to both owners and contractors to consider not only total cost increment, but also overall benefit for each alternative scheme of project scheduling. The problem is to select optimal trade-off between opportunity income and total cost increment. Although some owners are aware of the timing of profitability, they have not always been able to formalize methods of dealing with it. Moreover, their accounting systems do not recognize the significance of network compression-incurred overall benefit in practice. Project planners are not always able to demonstrate any advantage from optimizing overall benefit in relation to accelerated project schedule.

Figure 3.1 Owner's Opportunity Income

Fig. 3.1 is based on what has been the normal construction cost and normal duration. The above linear curve illustrates relationship between opportunity income and project duration. As duration is reduced, summation of interest saving and future revenue, which is assumed linearly related to project duration, is increased. Hence, the curve shows a negative slope as project duration extends. Total project cost is composed of construction cost and incentive fee. It generally increases with reduction on schedule. The lower and more sharply curved graph in Fig. 3.1 shows a relationship between project cost and duration. These two curves cross at a break-even point where the cost increment of schedule compression just balances opportunity income. This point

indicates that minimum duration can be traded-off for savings in interest on finance. More importantly, Fig. 3.1 indicates a different duration that is more profitable to the project than the duration shown at the break-even point. This is the duration point where maximum cut-off between opportunity income and total cost is realized.

Owners might have the greatest incentive to schedule compression. Since the sooner a project is completed and ready for commercial usage, the more it benefits. In construction of a hotel, a hospital, a manufacturing facility, a restaurant, etc., an owner can realize a definite cash payoff for every day gained in earlier completion of the project (O'Brien 1984). However, schedule compressing requires completing the work in less time or, in other words, performing more work in each planning time unit. This means that the cost per unit of time must increase (Degoff and Friedman 1985).

For a construction project financed with bank loans, in process of network compression, variables including interest saving, incentive fee, future revenue and cost increment are involved in overall benefit calculation. If compression-incurred opportunity income exceeds cost increment, project schedule can be accelerated to maximize the difference between opportunity income and cost increment, which is defined as overall benefit. Therefore, on one hand, reliable data of above mentioned variables for overall benefit calculation should be accessible to evaluate whether schedule compression is beneficial to an owner. On the other hand, contractual incentive fee can be considered as an economic reward to a contractor if the project is completed ahead of normally planned duration. Therefore, network compression scheme for maximum overall benefit is a win-win scheduling solution for both owners and contractors.

3.4 Research Objectives

In order to generate a feasible funding plan for a commercial construction project, several scheduling alternatives with different cost and duration contingencies are developed for decision-making (Mueller 1989). For a construction project financed with bank loans, an owner considers not only cost within construction phase, but also other economic factors such as interest saving and future revenue, which influence overall benefit of the project. Early project completion can be realized by network compression. It means crashing times of individual tasks on project network until

Chapter 3 Research Objectives Analysis

duration is accelerated to an optimum extent. However, it normally causes total cost increment including construction cost and incentive fee paid to contractor. In addition, it also brings more overall benefit due to earlier project completion. After compressing network to an optimal limit, summation of interest saving and future revenue, which is defined as opportunity income, amounts to a considerable sum. If compression-incurred opportunity income exceeds increased cost, difference of these two parts is defined as overall benefit. On one hand, it is advantageous to optimize project scheduling based on maximum overall benefit in stead of minimum-cost objective of TCT. On the other hand, as a win-win scheduling solution, it is also beneficial for contractor to gain incentive fee as a reward. It is therefore necessary to consider a new schedule optimization theme that favors overall benefit of a project and benefits both project owner and contractor.

Based on the above analysis, this research aims to optimize overall benefit by compressing project network to an optimal limit, balancing opportunity income and cost increment in the process of construction project scheduling. Therefore, research objectives are listed as follows:

- Review existing literature of TCT and its associated techniques, indicating its objective of minimum project cost with normally scheduled project duration;
- Indicate economic perspective of maximizing project profitability, clarifying difference between maximum overall benefit and minimum project cost;
- Elaborate owner's incentive to maximum overall benefit through network compression, whilst considering incentive fee as a reward to contractor and proposing a win-win project scheduling solution;
- Indicate necessity of establishing a new schedule optimization scheme and identify influencing variables to maximize overall benefit;
- Formulate objective function of overall benefit maximization, considering related variables, parameters and constraints; and
- Employ appropriate optimization technique to develop a solution model based on the proposed objective function and apply it to case studies, conduct sensitivity analysis and prove feasibility and practicability of the model.

3.5 Necessity for a New Schedule Optimization Scheme

Schedule optimization is an important task, but it is difficult to play its effective role in practice (Li 1996). Obviously, there is no one-optimization method that can “optimize” all schedules under all circumstances and is economically beneficial to all participants. Every optimization method is always under some certain assumptions and constraints. Although some optimization methods are used in scheduling such as TCT and time-constrained scheduling, they all have a common and single objective of minimizing cost. These methods provide little support for maximizing overall benefit regarding an owner’s concern of project profitability. Therefore, a new schedule optimization scheme is highly desirable when an owner, for instance, wishes to shorten the project duration so as to obtain more overall benefit with earlier revenue flowing in and more interest savings. For a commercial construction project financed with bank loans, one decisive objective is to maximize overall benefit through managing its relevant variables of ‘cost’ and ‘duration’ in parallel. As a win-win solution, for more incentive fee, a contractor is encouraged to construct project as early as possible whilst minimizing the increment of total project cost at the same time.

Based on above-mentioned research objectives, a new concept of schedule optimization scheme with maximum overall benefit and its associated solution model, which is different from TCT analysis, are highly necessary to resolve this particular scheduling optimization problem.

Chapter 4 Overall Benefit Duration Optimization

Overview

In this chapter, a new concept of overall benefit-duration optimization (OBDO) is proposed to optimize overall benefit in construction project scheduling. Assumptions, objective function and significances of OBDO are formulated and elaborated respectively. OBDO Concept is also applied in an illustrative example to validate its feasibility.

For a modern construction project, scheduling optimization with objective of maximum overall benefit is an important task. However, there are few effective optimization methods proposed both in literature and in practice. Although some scheduling optimization methods are used such as time-cost trade-off (TCT), resource-constrained scheduling, and time-constrained scheduling, they share same objective of minimizing cost or duration. In order to maximize overall benefit, more influencing factors need to be considered. In this research, a new optimization concept for maximum overall benefit and its associated model are proposed.

4.1 Definition and Assumptions of Overall Benefit-Duration Optimization

4.1.1 Definition of OBDO

For a construction project network with a set of tasks to be completed according to their precedence relationships, given a normally-planned schedule with minimum cost, under a prerequisite that network compression-incurred opportunity income exceeds total cost increment, a objective is to make decisions of task crashing and hence, compressing the network to a desired duration where overall benefit, which equals to

opportunity income minus cost increment, is maximized. This problem is referred to as Overall Benefit-Duration Optimization (OBDO).

When network compression is executed in OBDO process, the most economic way to compress a project network is adopted by inputting extra resources until overall benefit is maximized. An optimized allocation of investment in process of scheduling is considered to maximize overall benefit from network compression. Compared to a normally-planned schedule with minimum construction cost gained from TCT, a project owner is more pleased to accept an accelerated schedule since an earlier completion date often enables earlier facility operation and earlier revenues through recouping construction cost (Clough, *et al.* 2000). The owner receives more overall benefit from compressed duration and a shorter project-financing period.

4.1.2 Assumptions of OBDO

In general, solving OBDO problem is to maximize the difference between opportunity income and total cost increment through network compression. OBDO is valid only if the following assumptions are satisfied:

1. Opportunity income from network compression exceeds cost increment. Only with this indispensable assumption, it is beneficial to compress a project network with a minimum cost increment by buying time along the duration where maximum overall benefit is realized.
2. Adequate resource availability in process of network compression. OBDO assumes the amount of required resources is adequate while crashing tasks on network. This assumption enables crashing more parallel tasks independently. .
3. Interest rate, future revenue and incentive fee are accessible and predictable. Interest saving and future revenue compose opportunity income. Incentive fee and cost increment compose total cost. These three variables are important to formulate OBDO objective function of maximizing overall benefit. They are assumed predictable and accessible.
4. Time value effects of variables are not considered in OBDO objective function formulation. In real practice of cost estimation, data are collected and calculated

at time of scheduling where time value of direct and indirect cost is not necessarily considered. Therefore, in line with this practice, for integrity and simplicity purposes, NPV value and discount rate of the loan are not considered.

4.2 Objective Function of OBDO

Objective function of OBDO is based on microeconomic concept of marginally increasing or decreasing utility for varying levels of input. In OBDO, this marginal utility represents compressed time unit decided in time-cost relationships of individual tasks. In practice, since it is not practical to crash a task by 0.23 day, time-cost relationships of individual tasks are discrete. Therefore, this marginal compressed time unit should be a planning time unit, such as day, week or month.

In order to clearly formulate the objective function, variables and parameters of OBDO are defined and summarized in Table 4.1.

Table 4.1 Variables/Parameters of OBDO Objective Function

		Definition
Variables	$Z / \Delta Z =$	Overall benefit/Marginal overall benefit
	$I_c / \Delta I_c =$	Interest saving/Marginal interest saving
	$R_c / \Delta R_c =$	Future revenue/Marginal future revenue
	$C_t / \Delta C_t =$	Construction cost increment/Marginal construction cost increment
	$I_n / \Delta I_n =$	Incentive fee/Marginal incentive fee
Parameters	$C_o / \Delta C_o =$	Total project cost/Marginal total project cost
	$T_t / \Delta T_t =$	Compressed duration/Marginal compressed duration
	$i =$	Total number of tasks on project network
	$j \in i =$	Subset of crashed tasks on critical path of project network
	$T_n =$	Normally-planned duration with minimum cost
	$C_c / C_n =$	Compressed/Normally-planned construction cost
	$T_c / T_n =$	Compressed/Normally-planned project duration

Objective function of OBDO is demonstrated in Fig. 4.1. In the process of OBDO, by network compression, a project network can be completed before normally scheduled duration with minimum cost. Compressed project duration (T_t) is the summation of crashed time units of individual tasks on critical path. As shown in Fig. 4.1, interest saving (I_c) and future revenue (R_c) both increase with network compression. They

are assumed linearly related to compressed duration. Summation of these two values is defined as opportunity income. Marginal incentive fee (ΔI_n) is also assumed to linearly related to compressed duration. Marginal total project cost (ΔC_o) is the summation of marginal incentive fee (ΔI_n) and marginal construction cost increment (ΔC_t). Under a prerequisite that marginal opportunity income ($\Delta I_c + \Delta R_c$) exceeds marginal total project cost ($\Delta C_t + \Delta I_n$), difference between these two values is marginal overall benefit (ΔZ). As shown in Fig. 4.1, based on this network compression scheme, objective of OBDO is to maximize overall benefit (Z).

Figure 4.1 Overall Benefit-Duration Optimization

Based on the above analysis, objective function of OBDO is formulated as follows:

$$\text{Maximize } \Delta Z = (\Delta I_c + \Delta R_c) - (\Delta C_t + \Delta I_n) \tag{4-1}$$

Fig. 4.2 shows the objective of maximizing overall benefit (Z) through network compression. T_t denotes compressed project duration where maximized overall benefit is reached. As summarized in Table 4.1, four variables in OBDO objective function include marginal future revenue (ΔR_c), marginal interest saving (ΔI_c), marginal construction cost increment (ΔC_t) and marginal incentive fee (ΔI_n).

As for increased total project cost ($C_t + I_n$) shown in Fig. 4.2, it is composed of construction cost increment and incentive fee. Due to more investment of working capitals, relation between marginal increased construction cost and compressed time is

assumed to be non-linear since construction cost increment of succeeding compression on project network is normally more than that of preceding compression. In addition, relationship between marginal incentive fee and compressed time is assumed to be linear. Therefore, as shown in Fig. 4.2, as the summation of construction cost increment and incentive fee, total project cost increment is assumed to be a concave curve. To satisfy the prerequisite that opportunity income ($I_c + R_c$) obviously exceeds total project cost increment ($C_t + I_n$), as shown in Fig. 4.2, a curve of opportunity income is assumed to be obviously higher than the curve of total project cost increment. Since precise curve of opportunity income is difficult to obtain in either practical or theoretical analysis, the purpose of plotting Fig. 4.2 is to illustrate the objective of maximizing overall benefit (Z) through network compression.

Figure 4.2 Maximization of Z through Network Compression

Construction cost increment (C_t) is caused if a project completes ahead of normally planned duration, which is the difference between compressed construction cost and “normal-planned” construction cost. It can be expressed in Eq. (4-2) as follows:

$$C_t = (C_c - C_n) = \sum C_{ci} - \sum C_{ni} \tag{4-2}$$

$$T_t = \sum T(j) = T_n - T_c \tag{4-3}$$

$$I_n = T_t \times \Delta I_n = (T_n - T_c) \times \Delta I_n \tag{4-4}$$

$$C_o = C_c + I_n \quad (4-5)$$

Where i denotes total number of tasks on project network. Eq. (4-2) indicates that construction cost increment after network compression can be calculated by subtracting normally planned construction cost (C_n) from compressed construction cost (C_c). Eq. (4-3) indicates that compressed duration of project network (T_i) is a summation of crashed times of all individual tasks on its critical path. It is the difference between normally planned duration (T_n) and compressed duration (T_c). $j \in i$ denotes subset of crashed tasks on the critical path. Eq. (4-4) indicates linear relationship between marginal incentive fee (ΔI_n) and compressed duration (T_i) of the project. As part of total project cost after network compression, incentive fee is the multiplication of marginal incentive fee and compressed duration. Eq. (4-5) indicates that total project cost (C_o) after network compression is the summation of compressed construction cost (C_c) and incentive fee (I_n).

In addition, objective function of OBDO is subject to the following constraints:

$$\Delta R_c + \Delta I_c > \Delta C_t + \Delta I_n \quad (4-6)$$

$$\Delta Z, \Delta R_c, \Delta I_c, \Delta C_t, \Delta I_n \geq 0 \quad (4-7)$$

$$T(i) \geq 0 \quad (4-8)$$

$$T_c(i) \leq T(i) \leq T_n(i) \quad (4-9)$$

Eq. (4-6) indicates the first assumption of OBDO, which is only applicable under the prerequisite that opportunity income ($\Delta I_c + \Delta R_c$) exceeds cost increment ($C_t + I_n$). Eq. (4-6) and Eq. (4-7) indicate that values of all these five OBDO variables, as well as task duration on project network, are all assumed positive. Moreover, as a constraint of task crashing, Eq. (4-9) restricts all individual tasks being crashed within interval range between normal and crash times.

4.3 Formulation of OBDO

Three factors of OBDO formulation must be considered collectively including

Chapter 4 Overall Benefit-Duration Optimization

compressed duration, total cost and overall benefit. Compressed duration and total cost are determined by network length and summation of construction cost and compression-incurred incentive fee, respectively. Maximum overall benefit is introduced in OBDO as its objective in process of network compression.

Regarding the factor of duration, OBDO crashes times of individual tasks on network and hence, project duration, by increasing resource usage above its normal level. Tasks are linked according to their working sequences and formulate a project network. Critical Path Method (CPM) technique is used to analyze compressed network and identify critical path(s) and compressed duration. It is often the case that construction of some or all tasks can be crashed, or duration accelerated, by allocating more resources at expense of higher cost. In order to complete a task on network, a number of alternative construction methods are available to do so. Each method represents an alternative way to complete a task. A construction method is a combination of resources, such as cost, labor equipment, material and technology. According to time-cost relationship of individual task, construction method of a task with longest time costs the least. Other alternative construction methods with shorter times cost more. Comparatively, these can be considered as possible crashing possibilities of a task. In such a circumstance, when each task is completed by a construction method with longest duration, combination of all tasks formulates a network with so called “All-Normal” duration at the least cost. As a result, many different combinations of task crashing can yield particular compressed durations at different costs. Since various combinations of possible task durations can be associated in a project, the problem is to select the best combination. In the process of network compression, the objective of OBDO is to determine the best combination of construction methods for all tasks that yields maximum overall benefit.

Concerning the factor of total cost, it also depends on combinations of expenditures from task crashing on project network. Typically, construction cost consists of direct cost and indirect cost. Direct cost equals to cost summation of all individual tasks on project network. Cost of a task reflects its duration in accordance with their time-cost relationships. Indirect cost consists of overhead costs and is assumed linearly related to project duration. In some contractual agreements, if a project is completed ahead of normally planned schedule, incentive fee is paid to contractor as an economic reward.

Chapter 4 Overall Benefit-Duration Optimization

Total cost is composed of compressed construction cost and incentive fee. OBDO is applied to identify several tasks on critical path of a network. It enumerates possible alternatives of compressed time-units to these tasks. Each alternative represents a feasible solution in which tasks are crashed to meet reduction of duration. Among feasible alternatives to a particular compressed duration, their associated total costs are evaluated and the one with the least cost increment is more desirable. In OBDO, total project cost is the summation of additional expenditures from task crashing, construction cost at normally planned duration and compression-incurred incentive fee.

In OBDO, duration and cost are two inseparable factors in network compression. In real practice, there are trade-offs between time and cost to crash individual tasks. The more expensive resources are utilized in compression, the sooner it takes to complete a task. It is clear that some tasks can be crashed at lower cost increment than others. Hence, when a choice of tasks is considered for network compression, cheaper ones should be crashed instead of expensive ones. Difficulties arise because, for dozens, or even hundreds of tasks of a project network, there are considerable options of task crashing with various cost allocations.

For a construction project that requires substantive working capitals and takes long time to complete, in reference to the factor of overall benefit, it is normally financed with bank loans. In fact, in implementation of a construction project, its overall profit is closely influenced by various factors, such as duration, total cost, future revenue, which are of paramount importance and deserve simultaneous considerations. Since compression-incurred opportunity income exists, overall benefit changes with fluctuations of duration. An owner, for instance, may intend to shorten project duration so as to maximize overall benefit by collecting incomes after the project completes earlier than normally planned schedule.

Figure 4.3 OBDO Process

Process of OBDO is shown in Fig. 4.3. Given a project with tasks on its network, with critical path method (CPM) and time-cost trade-off (TCT) techniques, it is not difficult to initialize a project network with associated duration and total cost. Each task on a project network has several alternative construction methods with associated combinations of durations and costs. For an individual task, if an alternative construction method is set to its “all-normal duration”, minimum cost can be identified. When OBDO is applied, crashing individual tasks on network accelerates duration. Although it causes cost increment, it also brings opportunity income. OBDO is only applicable to the prerequisite that compression-incurred opportunity income exceeds cost increment. Thus, a saving is gained as marginal overall benefit. The network is compressed until any other task crashing results in an optimum accelerated schedule where increment of opportunity income can not be offset by cost increment. In each network compression, highest overall benefit is calculated and saved as

“sub-maximum”. Then, present value of sub-maximum overall benefit is compared with that of previous iteration of network compression. If present sub-maximum does not exceed the previous one, the network is further compressed. Sub-maximum of overall benefit is calculated in this iteration of further compression until overall benefit is finally maximized. In the end, maximum overall benefit, total cost and compressed schedule are gathered and exported as final OBDO output.

4.4 An Illustrative Example

Once OBDO objective function is formulated with its associated assumptions and constraints, in order to prove feasibility of OBDO, an illustrative example is applied to fulfill the following objectives:

1. To initialize “All normal” planned network of the example, applying TCT analysis to obtain minimum construction cost. Then apply OBDO to compress the network and obtain maximum overall benefit, comparing different scheduling results of TCT with OBDO solutions;
2. To compare different economic perspectives of TCT-oriented cost minimization with OBDO-oriented maximum overall benefit, accentuating economic incentive to network compression and objectives in this research;
3. To indicate the necessity of establishing an effective scheduling optimization model that is applicable to real-life projects. Integrating OBDO concept and appropriate optimization techniques, the proposed model should provide easy access to applications in real practice.

In this example, a project network comprising 11 tasks is represented in Fig. 4.4. Direct cost is \$200,200. Indirect cost is \$2,000/day (Shanmuganayagam, 2001). Relationship between cost and duration of individual tasks is linear. For each task, its crashing cost is assumed linear with its associated crashing time. Various task costs with crashing times define its cost slope. Task cost is calculated by multiplying task duration by its respective cost slope. Data regarding crashing range, which stands for maximum time unit each individual task can be possibly crashed, as well as its associated cost slope, are listed in Table 4.2.

Table 4.2 Task Crashing Data

Task	Crashing Range (Days)	Cost/Crashing Slope (\$/day)	Task	Crashing Range (Days)	Cost/Crashing Slope (\$/day)
1-2	0	1700	2-5	4	400
2-3	2	1200	5-9	5	1000
3-6	6	800	9-10	2	400
6-10	4	750			
2-4	4	1100			
4-7	2	950			
7-8	3	900			
8-10	2	500			

4.4.1 Time-Cost Trade-off (TCT) Solution

Given a construction project, several alternative construction methods are available to complete each task on its network. Various combinations of these construction methods for individual tasks lead to various construction durations. Normally, using more materials, workforce and equipments completes a construction earlier. Inevitably, cost also increases due to more workforce and capital input. In this example, regarding crashing ranges of each task, there are trade-offs between time and cost based on their linear relationships. Therefore, the objective of TCT is to find minimum construction cost for which the reduction of certain amount of time can be achieved.

4.4.1.1 All-normal Solution

Figure 4.4 An Illustrative Example: All-normal Solution

In this example, all-normal solution represents originally planned schedule that bears the longest duration. As shown in Fig. 4.4, all-normal solution to project schedule takes 90 days to complete. Critical path of the project network is 1-2-3-6-10. Non-critical paths are 1-2-4-7-8-10 and 1-2-5-9-10 with total time floats of 2 and 10 days,

respectively. Total construction cost, which is composed of direct and indirect cost, is \$380,200. After all-normal solution is initialized, within crashing range of each individual task shown in Table 4.2, two compressions of the network are conducted to reach TCT solution with its objective of minimum cost. Afterwards, further compression can be conducted to find OBDO solution of maximum overall benefit.

4.4.1.2 First Compression

Network compression means crashing critical task or tasks that bear the least cost increment. In this example, it means crashing task with the least cost slope. If n tasks on critical path were crashed by one day each, project duration would be n days shorter. To ensure no cost increment, summation of their cost slopes must be less than indirect cost (\$2,000/day $\times n$). In this compression, task 6-10 is selected and crashed by 2 days as it bares the least cost slope (\$750/days), which is less than indirect cost (\$2,000/day). As shown in Fig. 4.5, total construction cost after the first compression is \$377,700.

Figure 4.5 An Illustrative Example: First Compression

4.4.1.3 Second Compression

Two critical paths exist after the first compression. One is the original critical path of 1-2-3-6-10; the other is the path of 1-2-4-7-8-10. As shown in Fig. 4.6, to ensure project duration is accelerated with further compressions, these two critical paths must be crashed together.

Figure 4.6 An Illustrative Example: Second Compression

After the second compression, project network can be completed in 80 days, which is 10 days less than the original all-normal solution. Since all crashing ranges of selected tasks are crashed to zero, it seems that no further compressions can be possibly conducted. All these three task paths become critical after the second compression. If it is necessary to compress the network further, all these three critical paths must be compressed together. Tasks of 2-3, 2-4, and 2-5 respectively on these three critical paths are randomly selected to explore possibilities of further compressions. From data shown in Table 4.2, summation of cost slopes for these three tasks exceeds indirect cost slope ($\$1,200 + \$1100 + \$400 = \$2,700/\text{day} > \$2,000/\text{day}$). According to the minimum cost objective of TCT, further compression is uneconomical. After two compressions, comparison between all-normal solution and TCT solution is shown in Table 4.3.

Table 4.3 Comparisons between All-normal and TCT Solutions

---	All-normal Solution	TCT Solution
Duration	90 days	80 days
Indirect Cost	$90 \times \$2000 = \$180,000$	$80 \times \$2000 = \$160,000$
Direct Cost	\$200,200	Total Crashing Costs: Σ \$14,500 Task 6-10 by 4 days @ \$750/day = \$3,000, Task 3-6 by 6 days @ \$800/day = \$4,800, Task 8-10 by 2 days @ \$500/day = \$1,000, Task 7-8 by 3 days @ \$900/day = \$2,700, Task 4-7 by 2 days @ \$950/day = \$1,900, Task 2-4 by 1 day @ \$1100/day = \$1,100. Estimated Direct Cost + Crashing Cost = \$214,700
Total Cost	\$380,200	\$374,700

As shown in Table 4.3, total cost after two compressions is minimized to \$374,700, which is even less than that of all-normal solution. TCT solution after two compressions indicates that, crashing tasks on critical path within certain crashing range could decrease, or even minimize total construction cost. In terms of minimum

total cost, after two compressions, no further task crashing possibilities exist since there is no margin to decrease cost, but tend to increase.

4.4.2 OBDO Solution

Obviously, in this example, minimum total cost after two compressions is the best TCT solution since additional compressions cause cost increment. In practice, a network is scheduled based on TCT objective of minimum total cost. A project planner intends to schedule this network that can be completed in 80 days.

Figure 4.7 An Illustrative Example: OBDO Solution

However, based on OBDO, assume this example is a project financed with bank loans, the objective is to maximize overall benefit rather than minimize total cost. TCT solution seems not the best choice. Suppose the owner borrows bank loans that cover the total cost (\$374,700) with 10% interest rate per year. If time value of capital is ignored, interest reimbursement is approximately \$100/day. As for other variables in the objective function of OBDO, it is assumed that marginal future revenue after project completion is \$1,300/day. Marginal incentive fee is assumed to be \$200. As shown in Table 4.2, crashing ranges for these three tasks of 2-3, 2-5, and 2-4 are 2 days, 4 days, and 3 days, respectively. Task 2-4 has been crashed by 1 day in previous compressions. Therefore, only 3 days is available for further crashing. Maximum crashing range for further compression is only 2 days. As shown in Fig. 4.7, based on OBDO, after the third compression, each of these three tasks is crashed by 2 days. Project duration is

compressed to 78 days. Total cost is increased to \$376,300. Although it exceeds minimum construction cost (TCT solution) by \$1,600, OBDO solution introduces opportunity income of \$2,800. Based on the objective function of OBDO, although this compression costs more, it brings overall benefit of \$1,200.

4.4.3 Conclusion

As shown in Table 4.4, compared to “All-normal” and TCT solution, after OBDO application, opportunity income exceeds cost increment. Although first two compressions of TCT solution reach to minimum total cost, it still deserves further compressing for maximum overall benefit. This example illustrates different economic perspectives of cost and overall benefit. It also proves the feasibility of OBDO concept.

Table 4.4 Comparisons among All-normal, TCT and OBDO Solutions

---	All-normal Solution	TCT Solution	OBDO Solution
Project Duration	90 days	80 days	78 days ($\Delta = 2$)
Indirect Cost	\$180,000	\$160,000	\$156,000
Direct Cost	\$200,200	\$214,700	\$220,100
Total Construction/Project Cost	\$380,200	\$374,700	\$376,300
Revenue + Interest Savings ($\Delta R_c + \Delta I_c$)	0	0	\$2,800
Overall Benefit $\Delta Z = (\Delta R_c + \Delta I_c - \Delta C_t - \Delta I_n)$	---	---	\$1,200

It also needs to mention that individual tasks can not be crashed without limitations due to their physical properties. For example, it is not feasible to crash a task of pouring and cooling concrete by too many time units. Although extra resources and capital are available, cooling of poured concrete takes certain days and can not be further crashed. Moreover, for simplicity, time-cost relationships of each individual task on the network of this example are assumed linear. All results are manually calculated since the example network is quite simple. Nevertheless, it is acceptable if the objective of introducing this example is to illustrate different economic perspectives of cost and overall benefit, proving the feasibility of OBDO.

4.5 Significances of OBDO

After the objective function and assumptions of OBDO are formulated, an illustrative example is selected for experiment. Through network compression beyond TCT solution, OBDO is proved feasible with maximum overall benefit. Therefore,

significances of OBDO can be described as follows:

1. Unlike TCT objective of minimum cost within construction phase, OBDO aims to maximize overall benefit, considering influencing factors of duration, cost and opportunity income in entire project lifecycle.
2. Project owners might be the most beneficiary party in OBDO applications. Nevertheless, employing OBDO does not impair economic interest of contractors. To finish a project earlier than normally planned duration, incentive fee is introduced as an economic reward to a contractor. If a project is completed based on the schedule for maximum overall benefit, OBDO is not a concept that unilaterally benefits owner's interests and impairs contractor's. It is actually a win-win solution of project scheduling optimization to both parties.
3. OBDO circumvents some unpredictable risks through network compression. Generally, the longer a construction project lasts, the more risks project participants encounter. In present versatile economic and natural environment, development of a construction project spreads over a long period of time, during which risks related to weather, interest rate and other risks fluctuate or are difficult to predict. These risks directly influence cost, duration, and profitability of a project. Therefore, if a construction completes earlier than normally planned schedule with maximized overall benefit, all project participants are more confident and assured without worrying too much about negative influences of these risks to project profitability and duration.
4. Compressed schedule and maximum overall benefit obtained from OBDO can be also used as a criterion of contractor evaluation. Since TCT commonly assumes that a contractor crashes tasks with limited resources within target duration, traditional evaluation practice that considers selecting a contractor with minimum bidding price within budget and required duration. OBDO concept suggests a different tender evaluation criterion to contractor selection. An owner does not have to award a contractor with minimum bidding price and "all-normal" duration. The owner can select a contractor who can complete the project with project duration where overall benefit is maximized.

Chapter 4 Overall Benefit-Duration Optimization

In process of OBDO, to maximize overall benefit of a construction project with complicated network, project planner has to deal with crashing of dozens or even hundreds of tasks on project network. In order to compress network, a project planner has to enumerate possible alternatives of allocating crashing times to tasks. Each alternative represents a feasible solution in which tasks are crashed. Opportunity income associated with cost increment of each alternative is calculated and enumerated. The one with maximum overall benefit is the final solution. Since different enumerations of possible task crashing can be associated with network compression, in order to maximize overall benefit, the problem is to choose the best solution from these enumerations. Therefore, determining the best set of crashing enumeration is one of the goals of OBDO. Given a real-life construction project network with dozens, or even hundreds of tasks, if task A has a crashing possibilities with respective durations and costs, and task B has b crashing possibilities, and task C has c crashing possibilities, and so on and so forth, then total number of possible network compression enumerations is equal to $a \times b \times c \times K$. This is what is meant by a typical combinatorial optimization problem. Manually, it is cumbersome to enumerate all possible combinations of task crashing to identify an optimal network compression decision. It is also impractical to manually calculate cost increment and overall benefit.

Similar to TCT problems, three categories of existing techniques can be applied to solve OBDO problem. They are heuristic, mathematical programming, simulation and artificial intelligent methods. Heuristic methods are based on rules of thumb that do not need a great deal of mathematical effort. Given a normally-planned project network with minimum construction cost, if heuristic methods are applied to solve OBDO problem, project planner needs to visually identify several tasks on critical path and conduct network compression for maximum overall benefit. Due to simplicity and easy applicability, similar to their applications to TCT problems, heuristic methods perform well over a variety of optimization problems and are widely employed in practice. However, these methods are problem dependent with diversifying effectiveness. In addition, for a given real-life construction project network, there is no way of selecting the best heuristic rules to deal with. Although some heuristic methods provide good solutions, they can not guarantee global optimum solution.

Chapter 4 Overall Benefit-Duration Optimization

Another possible group of methods for OBDO problem are mathematical programming methods. These methods use mathematical programming techniques such as linear programming and integer programming. They require objective function and associated constraints and consume a great deal of computational effort. From literature review of their applications to TCT problems, they are not able to solve larger and more complicated problems in real practice. Although these methods are capable of locating local optimum solutions, they are only suitable for small-sized project networks. As for a typical combinatorial optimization problem like OBDO, mathematical methods are not able to deal with it.

Based on natural selection and mechanisms of population genetics originally developed by John Holland (1975), genetic algorithm (GA) emerged as an efficient method for searching global optimum in combinatorial optimization problems. After OBDO is formulated and proved feasible, in this research, a solution model incorporating GA is proposed as an appropriate approach to resolve OBDO problem.

Chapter 5 GAOBDO Model Formulation

Overview

In this chapter, a method called GA-based Overall Benefit-Duration Optimization (GAOBDO) is developed as an appropriate solution to OBDO problem. Computer implementation procedure of GAOBDO model is also introduced in details.

5. 1. Introduction

After the formulation of OBDO concept and its objective function, it is proved feasible in an illustrative example. In order to deal with real-life construction projects, it is necessary to establish an associated solution model. In the process of scheduling optimization of a project network with dozens, or even hundreds of tasks, neither traditional heuristic nor mathematical methods seems to be appropriate to resolve OBDO problem. Different from heuristic and mathematical methods that are only capable of dealing with simple networks without guaranteeing optimum solutions, genetic algorithm (GA) has been proved applicable to various combinatorial optimization problems in area of construction project scheduling.

Similar to Time-Cost Trade-off (TCT), considering the number of possible enumerations involved in task crashing, optimal decisions of OBDO is difficult to find and time-consuming in network compression. In the field of construction project scheduling, GA is capable of efficiently searching complex solution space in an intelligent manner to evolve optimal solutions to combinatorial optimization problems. Both in literature and practice, GA application tools are developed to resolve TCT problems and proved robust and straightforward. Unlike TCT objective of maximum

overall benefit, it is suggested that optimization problem like OBDO in scheduling can be tackled effectively with modest knowledge of the theory behind GA.

This research builds on the findings of previous GA applications to TCT problems and develops a practical solution model that circumvents some of their limitations. Development of the GA-based OBDO model is described in the following sections.

5. 2. Genetic Algorithm Model for OBDO

Genetic algorithm (GA) is a search procedure that combines an artificial survival-of-the-best strategy with genetic operators abstracted from nature. It aims to produce a mechanism that is capable of dealing with a variety of optimization problems (Goldberg, 1989). Basic theory behind GA is that a population of certain species adapts to live better in its environment after many generations of random evolutions. Applying GA to solve optimization problems is the same way. GA procedure starts by randomly generating an original population of possible solutions. They are then encoded in form of gene strings called chromosomes. Each chromosome in population represents one solution, which is better, or worse than the others. In one population, every chromosome has its fitness value to differentiate itself from the others. Fitness value of each chromosome is calculated by evaluating its performance with respect to its fitness function, which is often the objective function of the optimization problem. In this research, objective function of OBDO stands for maximum overall benefit.

To simulate natural survival-to-the-best process, “good” chromosomes of possible solutions exchange information to produce offspring chromosomes. These offspring chromosomes are evaluated and then better chromosomes replace less fit ones in population. This process proceeds for a number of generations in which population keeps evolving better solutions by replacing unfit solutions, until a stopping criterion is met where a local optimum solution or global optimum solution is achieved. In the end of GA process, chromosome with the best objective function value becomes optimum solution. Integrating random searching and coding-friendly characteristics of GA with objective function of OBDO, GA mechanism is applied to solve OBDO for better efficiency. In this research, this solution model is called GA-based Overall Benefit-Duration Optimization (GAOBDO).

5.3. GAOBDO Model Formulation

Applying GA to optimization problems normally includes the following basic steps:

- (1) Designing chromosome structure;
- (2) Deciding evaluation criteria (fitness function or objective function);
- (3) Generating an original population of chromosome (original solutions);
- (4) Selecting an offspring generation mechanism to produce new possible solutions;
- (5) Coding procedure in computer program.

As for OBDO, based on these five steps, chromosome structure is firstly designed as a string of gene contents. As shown in Fig. 5.1, each gene position stands for a task on project network. Content of a gene is an index to an alternative construction method. As such, a single chromosome represents an alternative construction method of the whole project with respective construction method of each task on the network.

5.3.1 Chromosome Structure

A construction project can be disassembled to dozens or hundreds of task on its network. Each task can be finished with various construction methods. Each construction method is a combination of task time and its associated cost. In this research, it is recognized that each task can be finished with five alternative construction methods. As for chromosome structure of GAOBDO model, numeric gene content in chromosome indicates alternative construction method of its associated task. Chromosome structure stands for task structure of its network. Each gene refers to an individual task on network. For each individual task, typical numeric gene index indicates its corresponding position on project network. Gene content refers to an available construction method for its corresponding task.

Chromosome structure of GAOBDO can be elaborated more specifically. As shown in Fig. 5.1, index “5” of the “fifth” task on project network represents “All-normal” construction method with the longest duration and the least cost. Consequently, “All-normal” planned schedule of the project can be represented when alternative construction method indices of all tasks are set to “5”. Accordingly, index “1” of a task

on project network represents “Crashed” construction method with the shortest duration and highest cost. Comparatively, each of the other index numbers stands for one of the alternative construction methods with associated intermediate times and costs. Since each task has 5 alternative construction methods, a chromosome consists of a group of index numbers ranging between “1” and “5”, lining up along with their respective tasks. To evaluate performance fitness of a chromosome, times and costs of tasks are first identified based on their indices on the chromosome. Critical path of project network is then determined and project duration is calculated, as well as total construction cost. Reflecting objective function of OBDO, fitness value of chromosome is then calculated as overall benefit.

Figure 5.1 Chromosome Structure of GAOBDO

Once chromosome structure and fitness function are formulated, evolutionary mechanism of GA takes place on a population of parent chromosomes. The simplest way is to randomly generate gene population. In GA applications, population size (number of chromosomes) is an important factor that influences solution and model processing time. Larger population size increases possibility of reaching optimum. However, it inevitably increases processing time. For GAOBDO, end users are given freedom to input number of population size by following suggestions of model developer or choosing themselves after familiarization. Once population is generated, fitness value (overall benefit of OBDO) of each chromosome in population is calculated. Correspondingly, its relative merit is calculated as fitness value of the chromosome divided by total fitness values of all chromosomes in population.

5.3.2 Two-point Crossover Operation

As discussed in Chapter 2, generation process among chromosomes in population is realized by reproduction, crossover and mutation. Simulating natural revolution, reproduction is a process in which individual chromosomes are duplicated according to

their fitness values. Copying chromosomes according to their fitness values results in that chromosome with higher value has higher probability of reproducing one or more offspring in next generation. Reproduction, of course, is an artificial version of natural selection. In GAOBDO, two-point crossover is applied by selecting two parent chromosomes, exchanging their gene contents and producing an offspring chromosome. Each of these two parent chromosomes is randomly selected in a manner such that its probability of being selected is proportional to its relative merit. This selection criterion ensures that better chromosomes have higher possibility of being selected, whilst maintaining diversity of random process.

Figure 5.2 Two-point Crossover Operation of GAOBDO

In GAOBDO, exchange of gene contents between two parent chromosomes is processed through two-point crossover operation. As shown in Fig. 5.2, two chromosomes (parent 1 and parent 2) are randomly selected and broken at two crossover points. One point is the position that connects gene 3 and 4. The other point is the position that connects gene 7 and gene 8. After genetic information exchange, two new offspring chromosomes (offspring 1 and 2) are generated.

5.3.3 Mutation Operation

Different from crossover that simulates natural reproduction, mutation is a rare GA operation that mimics process of accident generation of an odd offspring that happens to be a genius. The operation of mutation is to randomly select one chromosome from population and then arbitrarily changing one of its gene content. As shown in Fig. 5.3, alternative construction method of the sixth task on project network is arbitrarily changed from the “4th” construction method to the “1st” one. As shown in Fig. 5.3, after

mutation, offspring chromosome represents a project network with different construction method. Mutation can randomly break stagnation in evolutionary process, avoiding too-early convergence to unsatisfactory local optimum.

Once an offspring is reproduced by operations of reproduction, crossover and mutation, it is evaluated in turn and can be retained only if its fitness value is higher than others in population. This process normally proceeds for a large number of offspring generations until an optimum chromosome is reached. In this GAOBDO model, as a termination criterion for the process, end users are given freedom to input the number of population size by either following suggestions from model developer or by choosing themselves. Fig. 5.4 shows flow chart of proposed GAOBDO model.

Figure 5.3 Mutation Operation of GAOBDO

Any model without practical realization of convenient usage is less helpful to end users. Based on flow chart of GAOBDO model, as shown in Fig. 5.4, the GA-based procedure is coded in Visual Basic Application (VBA) language as a macro program. To maximize its benefits, catering to end users who know more of commercial project planning software than computer language, the author records the VBA-coded GAOBDO model into macro application environment of Microsoft Project 2003. Thanks to its user-friendly interfaces and customization function, end users without any VBA programming experiences are able to use recorded GAOBDO model to realize optimization with clear instructions and detailed outputs.

Figure 5.4 Flow Chart of GAOBDO

5. 4. Computer Implementation of GAOBDO

In Chapter 4, OBDO has been proved feasible with an illustrative example. After GAOBDO model is formulated, to maximize its benefit and prove its applicability in real-life construction project networks, GAOBDO is programmed in VBA (Visual Basic for Application). For further experiments and practical usage of end users, GAOBDO is integrated in macro programming environment of MS Project 2003 to facilitate computer implementation. Since CPM (Critical Path Method) engine and other computations like network compression and cost calculation are included as built-in functions in MS Project 2003, it is not necessary to program them additionally. GAOBDO model, as such, provides an automated tool that integrates OBDO concept into automated scheduling features of MS Project 2003, which is selected for its wide industrial usage and customized programmability functions. The VBA macro program follows detailed flow chart of GAOBDO model shown in Fig. 5.4.

As shown in Fig 5.5, GAOBDO model operates on an activated project in MS Project 2003. Data regarding project tasks with their precedence relationships are input using easy-to-use interfaces and default screens of the commercial software. However, MS Project 2003 does not facilitate end users specifying data regarding durations and costs of alternative construction methods for tasks. Therefore, special customization is required. The simplest option is to enter these data into some of blank database fields set aside by the software for user-specific applications. As shown in Fig. 5.5, GAOBDO model uses 10 task fields (Text 1 to Text 10) to store times and costs of up to five alternative construction methods for each task (odd-number fields for alternative durations and even-number fields for their corresponding costs). To enter data, as shown in Fig. 5.5, end user without any programming experiences can present the 10 fields on screen as a data-entry table designed as a special customization by the author. It requires more familiarity with customization options of the software and aims to develop a custom data-entry form that is linked to the same 10 fields.

Chapter 5 GAOBDO Model Formulation

Figure 5.5 Data-entry for Options of Alternative Construction Methods

Once data for alternative construction methods of each task are entered, they are saved into active project file and ready for GAOBDO application. End users are instructed by the macro program to input project information (opportunity income, interests saving, indirect cost and incentive fee), as shown in the following figures.

Figure 5.6 Data-entry for Daily Opportunity Incomes

Figure 5.7 Data-entry for Daily Interests Saving

Chapter 5 GAOBDO Model Formulation

Figure 5.8 Data-entry for Indirect Cost

Figure 5.9 Data-entry for Daily Incentive

As shown in the following figures, after inputting project information, end users also need to input four GA parameters (population size, generation number, and crossover and mutation rates) and then proceed with GAOBDO model, where macro program is run in MS Project 2003.

Figure 5.10 Data-entry for Population Size

Chapter 5 GAOBDO Model Formulation

Figure 5.11 Data-entry for Crossover Rate

Figure 5.12 Data-entry for Mutation Rate

Figure 5.13 Data-entry for Generation Number

At the end of GAOBDO, after GA searching is run through prescribed generations or reaches optimum solution, a customized interface pops up as output of GAOBDO. As the customized interface shown in Fig. 5.14, information of activated project network, including project duration, direct and total costs before and after GAOBDO, are shown for decision making of end users. Besides these data, parameters and information of GAOBDO procedure, including crossover and mutation rates, population size and

generation number, as well as processing time of the GA search, are also shown in Fig. 5.14.

Figure 5.14 Customized Interface of GAOBDO Results

Once end users confirm the output by clicking “OK” button on the customized interface, GAOBDO model stores crashed time of each task in column of “Duration”. End users can also check compressed CPM network in MS Project 2003 after GAOBDO application to the activated project. Different from original project network with “All-normal” duration at minimum cost, activated network after GAOBDO application in MS Project 2003 reflects the project schedule with maximum overall benefit.

Above-mentioned is the whole formulation process of GAOBDO model and its application procures. After that, in order to verify feasibility of OBDO concept and practicability of its associated GAOBDO model, it is suggested that case studies with sensitivity analysis is an appropriate way of validating effectiveness of GAOBDO. Besides model effectiveness validation, case studies and sensitivity analysis also provide end users with recommended values of GA parameter when GAOBDO is applied to various-sized projects.

Chapter 6 Case Studies

Overview

In this chapter, case studies and sensitivity analysis are performed to verify effectiveness of GAOBDO model as a macro program in Microsoft Project 2003. Suitable GA parameters values are suggested for practical applications of end users to various-sized construction projects.

6.1 Introduction

After OBDO concept its associated GAOBDO model are formulated, in order to assess performance of GAOBDO and its computer implementation in MS Project 2003, case studies and sensitivity analysis are conducted with convergence results. After applying GAOBDO to various sized projects, three cases with small, medium and large scale of network sizes are selected as typical construction projects encountered in the process of scheduling. The first case study is an 11-task project network. The second case study is an 18-task project network. The third case study is a large-scale commercial office building with a 78-task project completed in real practice. Sensitivity analysis and cross sensitivity analysis are conducted on these three case studies to achieve two objectives. One objective is to verify effectiveness of GAOBDO model in VBA programming environment of MS Project 2003. The other objective is to discuss suitable values and contributive efforts of GA parameters to optimum solutions. As discussed, these four GA parameters include crossover rate, mutation rate, population size and generation number. They play a decisive role in GA application to a combinatorial optimization problem like OBDO.

6.2 Sensitivity Analysis of Case Study 1

Project network of case study 1 is shown in Fig. 6.1. As an active project in MS Project 2003, time and cost data of alternative construction methods for individual tasks are

presented in Fig. 6.2. Before inputting GA parameters and running GAOBDO model in MS Project 2003, as shown in the “Duration” column in Fig. 6.2, each task has to be initially set to their least cost option with longest duration (All-normal duration). Consequently, the project has all-normal duration with minimum construction cost. As shown in Fig. 6.1, the critical path of this project network is task chain of 1-2-5-9.

Total direct cost of case study 1 is \$20,700 with project duration of 90 days. Daily indirect cost is \$100. Therefore, total project cost is calculated as \$29,700 by adding direct cost to indirect cost, which is the multiplication of daily indirect cost and project duration. The 90-day project duration with minimum construction cost is “All-normal” duration. Comparatively, this project may be completed either ahead of or drop behind the “All-normal” schedule. Therefore, in such a circumstance, penalty due to schedule delay is \$200. Incentive fee to network compression is \$700. Interest saving is not considered in such a small-scale project. Based on OBDO, due to network compression, with marginal opportunity income exceeding marginal total cost increment, it is lucrative to compress the 90-day network and search for maximum overall benefit.

Figure 6.1 Project Network of Case Study 1

Task	Duration	Preced.	Task1	Task2	Task3	Task4	Task5	Task6	Task7	Task8	Task9	Task10	Task11
1	10 d		10	2100	7	9500	8	1100	19	800	18	800	
2	16 d	1	10	3800	10	3200	12	1400	14	1400	15	1600	
3	26 d	1	10	4900	10	4200	20	4000	23	3000	26	3300	
4	20 d	1	10	4900	14	3900	16	3500	19	3200	20	3600	
5	38 d	2	20	13000	30	8600	33	7500	35	6200	38	5000	
6	22 d	3	12	3200	14	2700	16	2300	19	2000	22	1800	
7	24 d	4	10	3800	17	2700	19	2500	21	2400	24	2200	
8	18 d	5	10	2900	9	2200	10	2000	15	1900	18	1700	
9	26 d	5	10	2800	10	2700	20	1500	24	1200	26	1000	
10	12 d	8	10	1800	7	9500	8	1300	19	1000	12	800	
11	11 d	7	7	1800	8	9500	9	1200	19	1000	11	800	

Figure 6.2 Custom Form of Case Study 1 in MS Project 2003

In case study 1, once GAOBDO model is recorded as a macro program in MS Project 2003, it is used to conduct 8 experiments with data tables and output figures. 4 GA

parameters including crossover rate, mutation rate, population size and generation number are experimented respectively for sensitivity analysis. In each experiment, only one GA parameter is tested within a certain value range, while the other three GA parameters remain unchanged. It aims to find convergence result of the selected GA parameter to optimal solutions. For diversity and objectivity purposes, each GA parameter is tested twice to avoid disturbances. Outputs are evaluated to suggest reasonable value combinations of these 4 GA parameters, considering trade-offs between optimum accessibility and computational time. Compared to real-life projects, case study 1 is simple. It is not difficult to obtain global optimum of overall benefit with a heuristic method. Through comparing GA result with optimum solution obtained from heuristic method, sensitivity analysis is conducted to prove feasibility of OBDO concept and practicability of GAOBDO model in MS Project 2003.

6.2.1. Sensitivity Analysis of Crossover Rate (Case Study 1)

Theoretically, crossover operation plays a primary role in revolutionary process of typical GA. In case study 1, two experiments are conducted for sensitivity analysis of crossover rate. In the first experiment, 11 crossover rates in the range of $[0, 1]$ with value difference of 0.1 are tested on GAOBDO. In Hagazy's (1999) case study of an 18-task network running on a Pentium 233 MMX computer, population size and generation number were set as 100 and 1,000 respectively. For reference purpose, using a much faster computer of Pentium 1500MHz configuration, in this experiment, population size, generation number and mutation rate are set as 50, 100 and 0.02, respectively (Table 6.1). Compared to similar sized case studies found in journal paper, it is suggested that small population size and generation number are necessary since solution pool is small. High values of these two GA parameters cause too many duplicates in following generations and take unnecessarily long processing time. Value of mutation rate is also suggested to be within the range of $[0, 0.1]$. As shown in Fig. 6.3, value of overall benefit obtained from GAOBDO model reaches global optimum when crossover rate is set to 0.6.

Table 6.1 Sensitivity Analysis of Crossover Rate-1 (Case Study 1)

Population Size: 50			Mutation Rate: 0.02			
Generation Number: 100			All Normal Duration 90 Days			
Crossover Rate	Global Optimum (\$)	Maximum Overall Benefits (\$)	Compressed Total Cost (\$)	Compressed Duration (Days)	Time Consumed (Seconds)	Time to Optimum (Seconds)
0	9600	8500	41500	61	22	1
0.1	9600	8200	43200	59	24	1
0.2	9600	9000	40300	62	15	10
0.3	9600	8700	39200	64	15	8
0.4	9600	8600	39300	64	16	1
0.5	9600	9300	40700	61	17	2
0.6	9600	9600	41100	60	16	2
0.7	9600	9000	43100	58	17	4
0.8	9600	9300	41400	60	16	11
0.9	9600	9100	40900	61	17	4
1	9600	9300	40700	61	15	10

Figure 6.3 Sensitivity Analysis of Crossover Rate-1 (Case Study 1)

In the second experiment, population size is increased to 100, values of other 3 GA parameters are the same as those in the first experiment (Table 6.2). After population size increment, overall performance of the GAOBDO model is better than that in the first experiment. Comparatively, output is more stable with less variation gap between optimum values obtained from two consecutive crossover rates. Especially when crossover rate is set to 0.8, value of overall benefit reaches to global optimum.

Table 6.2 Sensitivity Analysis of Crossover Rate -2 (Case Study 1)

Population Size:				Mutation Rate:		
Generation Number:				All Normal Duration:		
100				90 Days		
Crossover Rate	Global Optimum (\$)	Maximum Overall Benefits (\$)	Compressed Total Cost (\$)	Compressed Duration (Days)	Time Consumed (Seconds)	Time to Optimum (Seconds)
0	9600	8600	42800	59	29	15
0.1	9600	8400	39500	64	25	25
0.2	9600	8600	41400	61	29	4
0.3	9600	8600	42100	60	30	28
0.4	9600	8700	41300	61	27	7
0.5	9600	8700	42000	60	26	3
0.6	9600	9100	39500	61	25	7
0.7	9600	9200	38000	63	28	1
0.8	9600	9600	41100	60	27	19
0.9	9600	9000	40300	65	25	12
1	9600	9300	40700	62	30	26

Figure 6.4 Sensitivity Analysis of Crossover Rate-2 (Case Study 1)

As shown in Fig. 6.3 and Fig. 6.4, among all trials in these two experiments, in average, optimum values of overall benefit in the range of [0.5, 1] are more satisfactory than those in the range of [0, 0.5]. Experiment results support finding of some GA theorists that crossover rate can be set within the range of [0.5, 1] (De Jong 1975). Theoretically, GA searching power arises mostly from crossover operation, where randomized exchange of genetic materials is performed to produce new chromosomes that represent new possible solutions. Crossover rate is a possibility that “good” chromosome solutions can produce “better” ones. In one population, within value range of [0, 1], higher crossover rate value allows more genetic materials being exchanged on chromosomes. Comparatively, within value range of [0.5, 1], it is more likely that higher values of crossover rate perform better in searching “better” solutions than lower

ones within the range of [0, 0.5]. It is noted that although crossover is primarily considered as a mechanism that improves performance of optimum solutions, it is also likely that higher crossover rates possibly break up a good chromosome structure that already exists in population. That is why value curve of maximum overall benefit do not maintain their increasing trend but oscillate within the range of [0.5, 1]. Therefore, more experiments on other case studies with larger project sizes are necessary to support the above results.

6.2.2. Sensitivity Analysis of Mutation Rate (Case Study 1)

Mutation involves arbitrarily changing values of selected genes on a chromosome. Main purpose of mutation operation is to enrich genetic diversity by introducing unexplored or retrieving lost genetic material to prevent GA search from getting stuck into a local optimum and prematurely converging to sub-optimal solutions.

15 mutation rate values ranging between [0, 0.3] are tested in two experiments. In the first experiment, population size, generation number and crossover rate are set as 50, 100 and 0.6, respectively (Table 6.3). The curve of optimum values fluctuates in the range of [0, 0.1]. In the range of [0.1, 0.3], the optimum value keeps decreasing and is no higher than that when mutation rate is set to 0 (Fig. 6.5).

Table 6.3 Sensitivity Analysis of Mutation Rate-1 (Case Study 1)

Population Size: 50				Crossover Rate: 0.6		
Generation Number: 100				All Normal Duration: 90 Days		
Mutation Rate	Global Optimum (\$)	Maximum Overall Benefits (\$)	Compressed Total Cost (\$)	Compressed Duration (Days)	Time Consumed (Seconds)	Time to Optimum (Seconds)
0	9600	8300	40300	65	9	7
0.01	9600	8600	37200	67	9	1
0.02	9600	9000	40300	62	11	3
0.03	9600	9500	41200	60	11	8
0.04	9600	8900	39000	64	10	6
0.05	9600	9000	41700	60	10	2
0.06	9600	8700	43400	58	11	6
0.07	9600	9200	40100	62	10	2
0.08	9600	9300	40700	61	11	8
0.09	9600	8500	40100	63	10	9
0.1	9600	8400	40200	63	11	1
0.15	9600	8300	41700	61	11	8
0.2	9600	8000	39200	65	10	6
0.25	9600	7900	37900	60	10	7
0.3	9600	7500	42500	61	10	4

Figure 6.5 Sensitivity Analysis of Mutation Rate-1 (Case Study 1)

In the second experiment, population size is increased to 100, while generation number and crossover rate remain the same (Table 6.4). On one hand, increasing population size may circumvent the risk of premature convergence of local optimum from small population size, whilst increasing generation number normally causes longer process time. On the other hand, crossover plays more decisive role than mutation in typical GA. Changing crossover rate values would confuse the objective of sensitivity study to mutation rate. With increment of population size, result of the second experiment is better than that in the first experiment.

Table 6.4 Sensitivity Analysis of Mutation Rate-2 (Case Study 1)

Population Size: 100				Crossover Rate: 0.6		
Generation Number: 100				All Normal Duration: 90 Days		
Mutation Rate	Global Optimum (\$)	Maximum Overall Benefits (\$)	Compressed Total Cost (\$)	Compressed Duration (Days)	Time Consumed (Seconds)	Time to Optimum (Seconds)
0	9600	8400	39500	59	20	15
0.01	9600	8700	39200	64	25	23
0.02	9600	8600	40000	63	22	12
0.03	9600	8900	43200	58	24	14
0.04	9600	8800	42600	59	22	15
0.05	9600	8700	39900	63	20	2
0.06	9600	9200	40100	62	23	8
0.07	9600	9500	41200	60	20	8
0.08	9600	9400	40600	61	22	8
0.09	9600	9400	40600	61	23	20
0.1	9600	8800	43300	58	21	3
0.15	9600	8600	40000	63	22	20
0.2	9600	8600	42100	60	21	14
0.25	9600	8500	41500	62	21	15
0.3	9600	8300	40300	63	23	17

Figure 6.6 Sensitivity Analysis of Mutation Rate-2 (Case Study 1)

Same as the first experiment, no obvious result shows that certain value of mutation rate affirmatively ensuring higher optimum values of overall benefit. Nevertheless, experiments show that global optimum values are all obtained in value range of $[0, 0.1]$. Moreover, optimum value obtained without mutation (mutation rate at 0) is lower than those with mutation being operated. Within the range of $[0.1, 0.3]$, optimum values obviously decrease and are even lower than that without mutation operation. Early convergence of premature optimum corresponds to usual practice and theoretical suggestion that mutation rate should not be “too high” (De Jong 1975). It corresponds to the result that optimum values are all obtained in the range of $[0, 0.1]$. This result indicates that high mutation rate makes a gene content of a single chromosome has high probability of being arbitrarily changed. It is likely that more “good” gene contents in chromosomes are mutated to “ordinary” ones.

6.2.3. Sensitivity Analysis of Population Size (Case Study 1)

Population size plays an important role in typical GA applications. Theoretically, increasing population size facilitates GA search obtaining optimum in bigger solution pool where more possible solutions are included. It is more likely to include optimum solution in possible solution pool with bigger population size. It is also more likely to produce optimum solution in following generations. In case study 1, 12 options of population sizes ranging between 20 and 1000 are experimented for sensitivity analysis. In the first experiment, generation number, crossover and mutation rates are set as 50,

0.6 and 0.08, respectively (Table 6.5). Result shows an increasing tendency of optimum values with increment of population size (Fig. 6.7). When population size is set to 800, optimum value of overall benefit converges at global optimum.

Table 6.5 Sensitivity Analysis of Population Size-1 (Case Study 1)

Generation Number: 50				Crossover Rate: 0.6		
Mutation Rate: 0.08				All Normal Duration: 90 Days		
Population Size	Global Optimum (\$)	Maximum Overall Benefits (\$)	Compressed Total Cost (\$)	Compressed Duration (Days)	Time Consumed (Seconds)	Time to Optimum (Seconds)
20	9600	8600	37200	67	5	3
50	9600	8400	38800	65	8	1
100	9600	8700	43400	58	14	4
150	9600	8800	37000	67	18	3
180	9600	8700	42700	59	24	2
200	9600	8900	40400	62	24	7
250	9600	9000	42400	59	35	22
300	9600	9000	41000	61	39	13
400	9600	9200	38000	65	64	19
500	9600	9400	41300	60	87	6
800	9600	9600	41100	60	155	54
1000	9600	9600	41100	60	213	38

Figure 6.7 Sensitivity Analysis of Population Size-1 (Case Study 1)

In the second experiment, generation number is set as 200, while crossover and mutation rates remain the same (Table 6.6). As shown in Fig. 6.8, there is an increasing tendency of optimum value with increase of generation number. Compared to the first experiment, optimum value of overall benefit converges earlier. When population size is set to 500, it converges at global optimum.

Comparatively, the second experiment took more computational time to reach to global

optimum due to generation number increment. Therefore, in order to apply GAOBDO model with high efficiency, simultaneously considering global optimum accessibility and tolerable processing time, selecting the best combination of population size and generation number is an issue that deserves further analysis.

Table 6.6 Sensitivity Analysis of Population Size-2 (Case Study 1)

Generation Number: 200				Crossover Rate: 0.6		
Mutation Rate: 0.08				All Normal Duration: 90 Days		
Population Size	Global Optimum (\$)	Maximum Overall Benefits (\$)	Compressed Total Cost (\$)	Compressed Duration (Days)	Time Consumed (Seconds)	Time to Optimum (Seconds)
20	9600	8300	40300	63	11	1
50	9600	8400	38800	65	18	2
100	9600	8900	40400	62	22	16
150	9600	8800	37000	67	36	14
180	9600	8900	40400	62	46	44
200	9600	9100	40900	61	54	27
250	9600	9100	40200	62	68	50
300	9600	9200	38000	65	111	16
400	9600	9200	40800	61	172	14
500	9600	9600	41100	60	264	32
800	9600	9600	41100	60	538	57
1000	9600	9600	41100	60	706	356

Figure 6.8 Sensitivity Analysis of Population Size-2 (Case Study 1)

6.2.4. Sensitivity Analysis of Generation Number (Case Study 1)

Generation number also plays a helpful but indeterminate role in typical GA applications. Increasing generation number facilitates GA searching for optimum solution since chromosomes in solution pool can be reproduced by more times. Large generation number leads to more recombination of genetic materials in possible

solutions. Therefore, despite its time consuming characteristic, it is more likely to produce optimum solution in experiments with larger generation numbers.

Table 6.7 Sensitivity Analysis Generation Number-1 (Case Study 1)

Population Size:		50		Crossover Rate:			0.8
Mutation Rate:		0.08		All Normal Duration:			90 Days
Generation Number	Global Optimum (\$)	Maximum Overall Benefits (\$)	Compressed Total Cost (\$)	Compressed Duration (Days)	Time Consumed (Seconds)	Time to Optimum (Seconds)	
100	9600	7900	37900	67	11	8	
150	9600	8800	43300	58	12	7	
200	9600	8300	36100	69	14	12	
250	9600	8600	40000	63	18	12	
300	9600	8600	42800	59	19	5	
400	9600	8600	37200	67	28	22	
500	9600	9400	41300	60	31	27	
700	9600	9300	40700	61	36	35	
900	9600	9300	40700	61	47	43	
1000	9600	9400	41300	60	51	20	
1200	9600	9600	41100	60	62	45	
1500	9600	9600	41100	60	81	23

Figure 6.9 Sensitivity Analysis of Generation Number-1 (Case Study 1)

In the first experiment, 12 generation numbers ranging between 100 and 1500 are tried while population size, crossover and mutation rates are set as 50, 0.8 and 0.08, respectively (Table 6.7). Result shows an increasing tendency of optimum values with generation number increment. Especially when generation numbers are set to 1200, optimum value converges at global optimum (Fig. 6.9).

Table 6.8 Sensitivity Analysis of Generation Number-2 (Case Study 1)

Population Size:			200	Crossover Rate:			0.8
Mutation Rate:			0.08	All Normal Duration:			90 Days
Generation Number	Global Optimum (\$)	Maximum Overall Benefits (\$)	Compressed Total Cost (\$)	Compressed Duration (Days)	Time Consumed (Seconds)	Time to Optimum (Seconds)	
100	9600	8400	39500	64	31	30	
150	9600	8900	41100	61	40	16	
200	9600	9000	43100	58	49	11	
250	9600	8700	39200	64	54	14	
300	9600	9000	40300	62	67	2	
400	9600	9200	40100	62	80	10	
500	9600	9200	38000	65	119	31	
700	9600	9100	43000	58	121	97	
900	9600	9400	41300	60	147	91	
1000	9600	9600	41100	60	163	139	
1200	9600	9600	41100	60	190	23	
1500	9600	9600	41100	60	242	85

Figure 6.10 Sensitivity Analysis of Generation Number-2 (Case Study 1)

In the second experiment, population size is increased to 200 with crossover and mutation rates remain the same (Table 6.8). Result shows that optimum value converges earlier. When generation number is set to 1000, optimum value of overall benefit converges at global optimum (Fig. 6.10).

6.2.5. Convergence Results of GA Parameters (Case Study 1)

Applying GAOBDO model to case study 1, sensitivity analysis of 4 GA parameters is conducted. Out of these 4 GA parameters, generation number only involves repetitions of reproduction and computational time that GA search consumes. The other 3 parameters, including crossover rate, mutation rate and population size, involve gene

content changes and variety of possible solutions, from where the GA power mainly rises. Therefore, generation number can be referred by the other 3 GA parameters with various values for convergence study. Additional experiments are carried out to obtain convergence results of these 3 GA parameters.

Figure 6.11 Convergence Result of Crossover Rate (Case Study 1)

In this convergence study, the parameter of crossover rate is decided by comparing 5 values with mutation rate being unchanged and generation number being set in the range of [100, 1000]. The convergence result of crossover rate is shown in Fig. 6.11. It is found that crossover rates of 0.5, 0.7 and 0.9 produce better convergence results that reach to global optimum value, whereas crossover rates of 0.1 and 0.3 produce much slower convergence. As a higher crossover rate creates a higher possibility of producing optimal results in a given number of generations, a crossover rate within the upper half range of [0.5, 1] is suggested in GAOBDO applications.

Similar to crossover rate, mutation rate is experimented with 6 different values in the range of [0.01, 0.30]. As shown in Fig. 6.12, mutation rates of 0.01, 0.03 and 0.07 produce better convergence results, whereas mutation rates of 0.10, 0.15 and 0.30 produce much slower and oscillating convergence results. Therefore, a mutation rate within the range of [0.01, 0.10] is suggested in GAOBDO application.

Figure 6.12 Convergence Result of Mutation Rate (Case Study 1)

Parameter of population size is experimented by 5 values in the range of [100, 800]. Convergence result is shown in Fig. 6.13. It is observed that slower convergence is obtained for population sizes of 300 and less. For population sizes of 500 and above, better performance of convergence is obtained.

Figure 6.13 Convergence Result of Population Size (Case Study 1)

Above sensitivity analysis results show that the concept of OBDO is feasible. Its associated GAOBDO model is proved capable of resolving OBDO problem of small-sized project. Nevertheless, GAOBDO model also needs to be verified and validated on larger project networks with more tasks on networks. More sensitivity analysis of GA parameters for various network sizes are necessary to suggest the best combinations of these GA parameters, which are helpful for practical usage of end users in GAOBDO application to practical-sized projects. Moreover, in order to apply GAOBDO model more effectively, it aims to provide end users with clear guidance on the best combination of GA parameters for global optimum solutions.

6.3 Sensitivity Analysis of Case Study 2

Project network of case study 2 is shown in Fig. 6.14. Duration and cost data associated with alternative construction methods of all tasks are also shown in Fig. 6.15. The 18-task project network is initially set to their longest duration option (All-normal duration), as shown in highlighted column of “Duration” in the active project.

Figure 6.14 Project Network of Case Study 2

In case study 2, direct cost is \$99,740 with duration of 169 days. Daily indirect cost is \$200. After simple calculation, total construction cost is \$133,540. Penalty due to schedule extension is \$20,000. Incentive fee due to network compression is \$1,000. Interest saving is not considered in case study 2 due to its simple project network. With marginal opportunity income exceeding marginal cost increment, it is feasible to apply GAOBDO and compress the 169-day network for maximum overall benefit.

Figure 6.15 Custom Form of Case Study 2 in MS Project 2003

Similar to case study 1, once GAOBDO model is recorded as a macro program in MS Project 2003, it is used to conduct 8 experiments. Testing on medium-sized project like case study 2 aims to support and validate tentative conclusions drawn from case study 1. Outputs with various values of 4 GA parameters are evaluated to determine a reasonable combination of optimum accessibility and computational time.

6.3.1. Sensitivity Analysis of Crossover Rate (Case Study 2)

In the first experiment, population size, generation number and mutation rate are set to 200, 500 and 0.08, respectively (Table 6.9). In case study 1, suggestion indicates that convergence result of crossover rate in the range of $[0.5, 1]$ are better than those in the range of $[0, 0.5]$. In order to approve its validity, for case study 2, 21 crossover rates in the range of $[0, 1]$ with value difference of 0.05 are tested for sensitivity analysis.

Table 6.9 Sensitivity Analysis of Crossover Rate-1 (Case Study 2)

Population Size: 200				Mutation Rate: 0.08		
Generation Number: 500				All Normal Duration 169 Days		
Crossover Rate	Global Optimum (\$)	Maximum Overall Benefits (\$)	Compressed Total Cost (\$)	Compressed Duration (Days)	Time Consumed (Seconds)	Time to Optimum (Seconds)
0	2069880	1723380	210160	109	130	8
0.05	2069880	1894930	218610	103	133	14
0.1	2069880	1896860	216680	103	135	7
0.15	2069880	1844550	208990	105	136	14
0.2	2069880	1895320	218220	103	134	7
0.25	2069880	1808940	214600	106	138	56
0.3	2069880	1779950	213590	107	136	25
0.35	2069880	1839450	214090	105	134	16
0.4	2069880	1814970	208570	106	138	38
0.45	2069880	1847750	205790	105	134	44
0.5	2069880	1953180	220360	101	135	66

0.55	2069880	1924900	218640	102	137	57
0.6	2069880	1924270	219270	102	133	55
0.65	2069880	1951600	221940	101	134	46
0.7	2069880	1924310	219230	102	136	58
0.75	2069880	1953880	219660	101	143	75
0.8	2069880	1926320	217220	102	136	52
0.85	2069880	1955750	217790	101	135	42
0.9	2069880	1982130	221410	100	137	98
0.95	2069880	1954070	219470	101	132	47
1	2069880	1958100	215440	101	135	84

Figure 6.16 Sensitivity Analysis of Crossover Rate-1 (Case Study 2)

Among all trials in the first experiment, when crossover rate is set to 0, optimum value is the lowest (Fig. 6.16). Then, in the range of [0, 0.5], optimum value increases with oscillations without any signs of stability and persistent augment. In the range of [0.5, 1], curve of optimum overall benefit picks up and stabilizes with slight variations.

In the first experiment, even with large generation number, overall performance of GAOBDO model is not so satisfactory since population size is relatively small. There are obvious gaps between local optimum values of overall benefit and global optimum (Fig. 6.16). Based on mechanism of typical GA, in one generation, small population size makes GA searching less likely to obtain optimum solution. Since possible solution pool is small, probability of reproducing and generating optimum solution from solution pool is also small. In order to improve overall performance of GAOBDO model without sacrificing too much computational time, in the second experiment, other GA parameter being the same, population size is increased to 300 (Table 6.10).

Table 6.10 Sensitivity Analysis of Crossover Rate-2 (Case Study 2)

Population Size: 300				Mutation Rate: 0.08		
Generation Number: 500				All Normal Duration 169 Days		
Crossover Rate	Global Optimum (\$)	Maximum Overall Benefits (\$)	Compressed Total Cost (\$)	Compressed Duration (Days)	Time Consumed (Seconds)	Time to Optimum (Seconds)
0	2069880	1844830	208710	105	195	7
0.05	2069880	1844570	208970	105	194	107
0.1	2069880	1868970	214570	104	210	58
0.15	2069880	1898730	214810	103	196	14
0.2	2069880	1840410	213130	105	213	114
0.25	2069880	1868520	215020	104	198	56
0.3	2069880	1925130	218410	102	204	44
0.35	2069880	1926450	217090	102	194	16
0.4	2069880	1896400	217140	103	196	25
0.45	2069880	1874470	209070	104	206	38
0.5	2069880	1929390	214150	102	196	152
0.55	2069880	1930700	212840	102	196	58
0.6	2069880	1954080	219460	101	197	98
0.65	2069880	1955110	218430	101	205	176
0.7	2069880	2009500	224040	99	203	75
0.75	2069880	2010200	223340	99	204	46
0.8	2069880	1951870	221670	101	213	175
0.85	2069880	2065900	227640	97	215	184
0.9	2069880	2041080	222460	98	205	152
0.95	2069880	1982130	221410	100	202	47
1	2069880	1955600	217940	101	197	57

Figure 6.17 Sensitivity Analysis of Crossover Rate-2 (Case Study 2)

In the second experiment, with increment of population size, convergence result is better than that of the first experiment. Especially for the particular trial when crossover rate is set as 0.85, obtained value of overall benefit is slightly lower than global optimum value (Fig. 6.17). From sensitivity analysis of crossover rate in these 2 case studies, it is noted that local optimum values of overall benefit in range of [0, 0.5] vary more significantly than those in the range of [0.5, 1]. Optimum values in the range

of [0, 0.5] is obviously worse than those in the range of [0.5, 1], which are closer to global optimum. In terms of crossover rate, sensitivity analysis results of case study 2 correspond to the tentative suggestion drawn from case study 1.

6.3.2. Sensitivity Analysis of Mutation Rate (Case Study 2)

15 values of mutation rate in the range of [0, 0.3] are experimented in case study 2. In the first experiment, population size, generation number and crossover rate are set as 100, 500 and 0.85, respectively (Table 6.11). Result shows no obvious increasing or decreasing trend of optimum values in the range of [0, 0.1] (Fig. 6.18). When mutation rate is set to 0, optimum value is lower than any other values. In this range, applying mutation obtains better values. However, in the range of [0.1, 0.3], obtained values decrease sharply. Same as case study 1, if mutation rate is set unreasonably high, it loses some “good” genetic materials in possible solution pool since “good” chromosomes are more likely to be mutated to “ordinary” ones.

Table 6.11 Sensitivity Analysis of Mutation Rate-1 (Case Study 2)

Population Size: 100				Crossover Rate: 0.85		
Generation Number: 500				All Normal Duration 169 Days		
Mutation Rate	Global Optimum (\$)	Maximum Overall Benefits (\$)	Compressed Total Cost (\$)	Compressed Duration (Days)	Time Consumed (Seconds)	Time to Optimum (Seconds)
0	2069880	1866820	216720	102	100	32
0.01	2069880	1868520	215020	101	111	24
0.02	2069880	1984220	219320	101	92	9
0.03	2069880	1951870	221670	100	105	19
0.04	2069880	1953570	219970	104	106	65
0.05	2069880	1953730	219810	104	100	80
0.06	2069880	1927090	216450	101	99	49
0.07	2069880	1896270	217270	103	103	66
0.08	2069880	1956850	216690	101	102	80
0.09	2069880	1954080	219460	101	103	61
0.1	2069880	1867470	216070	104	106	79
0.15	2069880	1844570	208970	105	100	64
0.2	2069880	1816050	207490	106	105	48
0.25	2069880	1787270	206270	107	104	22
0.3	2069880	1782580	210960	107	110	48

Figure 6.18 Sensitivity Analysis of Mutation Rate-1 (Case Study 2)

In the second experiment, population size, generation number remains the same as those in the first experiment, while crossover rate is changed to 0.6 (Table 6.12). Same as the first experiment, optimum value reaches the lowest when mutation rate is set to 0. Therefore, it shows that applying mutation helps obtain higher optimum values. However, if mutation rate is set too high, as shown in Fig 6.18 and Fig 6.19, it does not only guarantee higher optimum value of overall benefit, but also converge at optimum values that are even lower than that without applying mutation.

Table 6.12 Sensitivity Analysis of Mutation Rate-2 (Case Study 2)

Population Size:		100		Crossover Rate:		0.6
Generation Number:		500		All Normal Duration		169 Days
Mutation Rate	Global Optimum (\$)	Maximum Overall Benefits (\$)	Compressed Total Cost (\$)	Compressed Duration (Days)	Time Consumed (Seconds)	Time to Optimum (Seconds)
0	2069880	1897310	216230	103	104	39
0.01	2069880	1897310	216230	103	109	34
0.02	2069880	1895570	217970	103	109	45
0.03	2069880	1869250	214290	104	105	32
0.04	2069880	1897700	215840	103	101	36
0.05	2069880	2009720	223820	99	102	19
0.06	2069880	1924500	219040	102	105	41
0.07	2069880	1958320	215220	101	96	69
0.08	2069880	1984060	219480	100	104	42
0.09	2069880	1987020	216520	100	104	60
0.1	2069880	1838050	215490	105	109	18
0.15	2069880	1845350	208190	105	107	29
0.2	2069880	1838270	215270	105	105	52
0.25	2069880	1840410	213130	105	107	94
0.3	2069880	1838940	214600	105	103	18

Figure 6.19 Sensitivity Analysis of Mutation Rate-2 (Case Study 2)

6.3.3. Sensitivity Analysis of Population Size (Case Study 2)

In case study 2, 12 options of population size in the range of [20, 1200] are experimented twice with GAOBDO model. In the first experiment, generation number, crossover and mutation rates are set to 100, 0.85 and 0.08, respectively (Table 6.13). For values of population size, in the range of [20, 250], optimum value curve shows no obvious increasing trend. Due to small population size, number of possible solutions to the optimization problem is also limited. It is less likely for GAOBDO to search in limited solution pool and obtain promising increasing tendency of optimum value, not to mention global optimum value. Within population size in the range of [300, 1200], optimum value curve maintains an obvious increasing trend (Fig. 6.20). Nevertheless, there is still an obvious gap between the highest local optimum value and global optimum. Probably due to small generation number of 100, it prevents GAOBDO model from approaching global optimum. Therefore, in the second experiment, increasing generation number is suggested to help improving convergence result. Inevitably, computational time for GAOBDO application increases due to generation number increment. Therefore, how to balance population size and generation number and reach to global optimum solution, whilst considering factor of saving computational time, is one of the concerns that worth further investigating.

Table 6.13 Sensitivity Analysis of Population Size-1 (Case Study 2)

Generation Number:				100	Mutation Rate:		0.08
Crossover Rate:				0.85	All Normal Duration:		169 Days
Population Size	Global Optimum (\$)	Maximum Overall Benefits (\$)	Compressed Total Cost (\$)	Compressed Duration (Days)	Time Consumed (Seconds)	Time to Optimum (Seconds)	
20	2069880	1930700	212840	102	10	8	
50	2069880	1873080	210460	104	12	4	
100	2069880	1898350	215190	103	20	5	
150	2069880	1866410	217130	104	28	9	
200	2069880	1896400	217140	103	42	5	
250	2069880	1873080	210460	104	58	6	
300	2069880	1876700	206840	104	83	8	
400	2069880	1953180	220360	101	160	15	
500	2069880	1956600	216940	101	259	26	
700	2069880	1951820	221720	101	499	20	
1000	2069880	1982130	221410	100	987	38	
1200	2069880	2009720	223820	99	1387	168

Figure 6.20 Sensitivity Analysis of Population Size-1 (Case Study 2)

In the second experiment, crossover and mutation rates remain unchanged, generation number is increased to 600 (Table 6.14). Same as the first experiment, optimum value curve shows no obvious increasing trend within the values of population size in the range of [20, 250]. With increment of population size, optimum value maintains an increasing trend and converges at global optimum with population size of 1000.

Table 6.14 Sensitivity Analysis of Population Size-2 (Case Study 2)

Generation Number:				600	Mutation Rate:		0.08
Crossover Rate:				0.85	All Normal Duration:		169 Days
Population Size	Global Optimum (\$)	Maximum Overall Benefits (\$)	Compressed Total Cost (\$)	Compressed Duration (Days)	Time Consumed (Seconds)	Time to Optimum (Seconds)	
20	2069880	1757140	206400	108	35	23	
50	2069880	1844550	208990	105	43	42	
100	2069880	1816050	207490	106	80	25	
150	2069880	1867450	216090	104	115	19	
200	2069880	1868550	214990	104	172	33	
250	2069880	1844040	209500	105	260	9	
300	2069880	1929850	213690	102	401	78	
400	2069880	1958800	214740	101	765	34	
500	2069880	1926320	217220	102	1303	32	
700	2069880	1958800	214740	101	2150	47	
1000	2069880	2069880	223660	97	3845	190	
1200	2069880	2069880	223660	97	5023	96

Figure 6.21 Sensitivity Analysis of Population Size-2 (Case Study 2)

Therefore, it is noted that simply increasing population size does not help too much to obtain higher optimum values. Moreover, although overall performance after the increment of generation number is much better and reaches global optimum, it consumes much more computational time. Therefore, for effective applications of GAOBDO model in various sized projects, in order to obtain local optimum or global optimum values within tolerable computational time span, it is important to suggest end users with appropriate combination of population size and generation number.

6.3.4. Sensitivity Analysis of Generation Number (Case Study 2)

As a medium-sized project network, in case study 2, value range of generation number is selected broader than that in case study 1. 14 values of generation number in the range of [100, 10000] are experimented twice for sensitivity analysis of generation number. In the first experiment, population size, crossover and mutation rates are set to 100, 0.8 and 0.08, respectively (Table 6.15). Results show that optimum values oscillate within value of generation number in the range of [100, 1000]. Optimum value increases with increment of generation number. When generation number is set to 10,000, its optimum value almost reaches global optimum with slight difference.

Table 6.15 Sensitivity Analysis of Generation Number-1 (Case Study 2)

Population Size: 100				Crossover Rate: 0.8		
Mutation Rate: 0.08				All Normal Duration: 169 Days		
Generation Number	Global Optimum (\$)	Maximum Overall Benefits (\$)	Compressed Total Cost (\$)	Compressed Duration (Days)	Time Consumed (Seconds)	Time to Optimum (Seconds)
100	2069880	1819320	204220	106	22	8
150	2069880	1779950	213590	107	26	8
200	2069880	1778520	215020	107	30	6
300	2069880	1873530	210010	104	40	5
350	2069880	1844130	209410	105	42	10
500	2069880	1869720	213820	104	54	22
700	2069880	1847320	206220	105	69	59
1000	2069880	1897500	216040	103	98	70
1500	2069880	1895970	217570	103	138	92
2000	2069880	1903330	210210	103	182	151
3000	2069880	2011980	221560	99	329	27
5000	2069880	2039150	224390	98	449	389
8000	2069880	2041080	222460	98	685	685
10000	2069880	2067370	226170	97	859	757

Figure 6.22 Sensitivity Analysis of Generation Number-1 (Case Study 2)

In the second experiment, other parameters being the same, population size is increased to 200. Similar to the result from the first experiment, optimum value curve keeps its increasing trend after generation number is increased to 500. It finally reaches global optimum when generation number is set to 10,000.

Therefore, it is concluded that generation number should not be set “too small” or “too large”. On one hand, with small generation number, GAOBDO model has few chances to thoroughly reproduce more possible solutions and proceed to search for better solutions. On the other hand, with large generation number, global searching advantage of GA might be overturned by time-consuming inevitability.

Table 6.16 Sensitivity Analysis Generation Number-2 (Case Study 2)

Population Size:			200	Crossover Rate:			0.8
Mutation Rate:			0.08	All Normal Duration:			169 Days
Generation Number	Global Optimum (\$)	Maximum Overall Benefits (\$)	Compressed Total Cost (\$)	Compressed Duration (Days)	Time Consumed (Seconds)	Time to Optimum (Seconds)	
100	2069880	1815820	207720	106	44	9	
150	2069880	1837170	216370	105	58	19	
200	2069880	1866270	217270	104	69	58	
300	2069880	1922920	220620	102	88	65	
350	2069880	1927300	216240	102	99	79	
500	2069880	1896230	217310	103	135	68	
700	2069880	1901750	211790	103	179	58	
1000	2069880	1925780	217760	102	239	128	
1500	2069880	1932070	211470	101	361	44	
2000	2069880	1954080	219460	102	455	308	
3000	2069880	2010200	223340	99	679	672	
5000	2069880	2039150	224390	98	1135	769	
8000	2069880	2064900	228640	97	1789	1718	
10000	2069880	2069880	223660	97	2202	2082

Figure 6.23 Sensitivity Analysis of Generation Number-2 (Case Study 2)

6.3.5. Convergence Results of GA Parameters (Case Study 2)

Similar to case study 1, convergence results of GA parameters from case study 2 are introduced in this section.

Figure 6.24 Convergence Result of Crossover Rate (Case Study 2)

As shown in Fig. 6.24, with mutation rate being the same, applying values of generation number in the range of [100, 2000], parameter of crossover rate is tested by comparing 5 values in the range of [0.1, 0.9]. It is found that crossover rates at 0.5, 0.7 and 0.9 produce better convergence results than those at 0.1 and 0.3.

In the range of [0.01, 0.30], 6 values of mutation rates are experimented in case study 2. Convergence result is shown in Fig. 6.25. Similar to case study 1, mutation rates of 0.01, 0.03 and 0.07 converge at higher optimum values than those from the values of 0.15 and 0.30.

Figure 6.25 Convergence Result of Mutation Rate (Case Study 2)

Convergence result of population size for case study 2 is shown in Fig. 6.26. Parameter of population size is tested by comparing 5 values in the range of [100, 800]. It is observed that slower convergence is obtained for population sizes of 300 or less. For population sizes of 500 or above, better convergence result is obtained.

Figure 6.26 Convergence Result of Population Size (Case Study 2)

6.4 Cross Sensitivity Analysis

In sensitivity analysis, 4 GA parameters including crossover rate, mutation rate, population size and generation number are studied one at a time with separate experiments to validate feasibility of OBDO and applicability of GAOBDO. Moreover, as a systematic method, cross sensitivity analysis is applied to investigate contribution of each of these 4 parameters to GAOBDO performance.

Cross sensitivity analysis is a common and fundamental scientific method that measures success or failure of contributive effect of individual parameter to optimization solutions. One of the main objectives of cross sensitivity analysis is to study contributions among parameters. If the simple effects of various parameter A values are the same for those of all values of parameter B , these two factors are equally cross sensitive or contributive to optimum solutions. However, it may be the case that contributions of various parameters are different. One might expect greater contributions of parameter A that yield a higher potential than the applications of parameter B . In this section, various contribution effects of these 4 GA parameters employed in GAOBDO applications are studied in cross sensitivity analysis. In this research, objectives of cross sensitivity analysis are listed as follows:

- To investigate contribution effect of GA parameters to optimum solution. It provides an orderly and completed picture that shows whether these parameters are equally, or not equally contributive to overall optimum;
- To quickly determinate contributive effects of GA parameters. It is able to explore which GA parameter is more contributive and can therefore be more thoroughly analyzed in separate experiments;
- To lead to recommendations applicable to a wide range of conditions. By cross sensitivity analysis, “more-important” GA parameters can be applied to a broader range of circumstances.

In typical GA, in order to try more possible solutions to an optimization problem, crossover and mutation operations are involved in breaking chromosome structures and assembling them into new ones. These two parameters are related to realization of solution variety, which increases possibility of reaching global optimum solution to

optimization problem within the same chromosome population. One might expect different contributive effects of crossover and mutation rates to global optimum solution. Regarding population size and generation number, they are not involved in breaking and assembling gene structures, but enlarging possible solution pool and more repetition times of reproductions. They are closely related to efficient accessibility and processing time of global optimum solution. There might also be different contributive effects of these two GA parameters. Under such a circumstance, cross sensitivity analysis is conducted to investigate contributive effects of these two pairs of GA parameters respectively. Take crossover and mutation rates as an example, mechanism of cross sensitivity analysis of these 4 GA parameters can be described as follows:

Firstly, several pairs of experiments are conducted with GAOBDO model in these two case studies. Each pair of experiments with certain numbers of trials is conducted to explore contributive effects of crossover and mutation to global optimum solution. Each pair of experiments includes one experiment of cross sensitivity for crossover rate and the other one for mutation rate.

Secondly, in each pair of experiments, one experiment with certain number of trial runs is conducted to explore cross sensitivity of crossover rate with various values, while values of the other 3 GA parameters remain unchanged. Then, times of GAOBDO results that reach global optimum value are counted and recorded. The other experiment is conducted to explore cross sensitivity of mutation rate. Same as crossover, same number of trials are run with various mutation rates while values of the other 3 GA parameters remain unchanged. Then, times of GAOBDO result that reach global optimum value are also counted and recorded. In each of these two experiments with same number of trials, if times of reaching global optimum due to crossover are obviously higher than those due to mutation, it is noted that crossover is more cross sensitive and contributive to global optimum solution than mutation. If there is negligible difference between times of reaching global optimum for crossover and mutation, these two GA parameters are equally cross sensitive and equivalently contributive to global optimum solution.

6.4.1. Cross Sensitivity Analysis of Crossover Rate and Mutation Rate

Based on the above-mentioned mechanism design of cross sensitivity analysis,

contributive effects of crossover and mutation rates to global optimum solutions are analyzed in case study 2 with 10 experiments. In each of these 10 experiments, various numbers of trials ranging from 10 to 100 are run. Take cross sensitivity analysis of crossover rate as an example, 10 experiments are conducted in GAOBDO model with various values of crossover rates, whilst the other 3 GA parameters remain unchanged. In each of these 10 experiments, times of reaching global optimum out of its associated trial number are recorded in Table 6.17 and plotted in Fig. 6.27. Based on the same mechanism, in each of the other 10 experiments, times of reaching global optimum out of its associated trial number due to mutation rate variety are also recorded in table 6.17 and plotted in Fig. 6.27.

Table 6.17 Cross Sensitivity of Crossover and Mutation Rates

Order of Experiment	Number of Trials	Times of Reaching Optimum (Crossover Rate)	Times of Reaching Optimum (Mutation Rate)
1	10	9	2
2	20	19	3
3	30	26	5
4	40	32	7
5	50	42	8
6	60	49	10
7	70	60	12
8	80	75	15
9	90	81	18
10	100	92	20

In each experiment, various crossover rates in the value range of [0.5, 1] are applied in GAOBDO model, whilst the other 3 GA parameters remain unchanged. In addition, mutation rates in the value range of [0, 0.1] are also applied. It is noted that value ranges of crossover and mutation rates are all referred to conclusions of previous sensitivity analysis. As shown in Table 6.17, in the first experiment, 10 trail runs are applied with crossover and mutation rates respectively. In the first pair of experiments, times of reaching global optimum solution due to crossover and mutation rates out of 10 trials are 9 and 2 respectively. As shown in Fig. 6.27, in every pair of experiments, times of reaching optimum due to crossover rate variety are obviously higher than those due to mutation rate variety. It can be concluded that crossover rate is more cross sensitive and are more contributive than mutation rate to global optimum solution.

Figure 6.27 Cross Sensitivity of Crossover and Mutation Rates

6.4.2. Cross Sensitivity Analysis of Population Size and Generation Number

Similar to cross sensitivity analysis of crossover and mutation rates, 10 pairs of experiments are conducted with GAOBDO to analyze cross sensitivities of population size and generation number. Each pair of experiments includes certain number of trials ranging from 10 to 100. Each experiment is conducted 10 times in GAOBDO with various values of population size and generation number respectively, whilst the other 3 GA parameters remain unchanged. In each of these 10 experiments, times of reaching global optimum out of its associated trial number due to varieties of population size and generation number are recorded in Table 6.18 and plotted in Fig. 6.28.

As shown in Fig. 6.28, in every pair of experiments, times of reaching optimum due to varieties of population size and generation number are almost the same. In the second pair of experiments with 20 trial runs, times of reach global optimum due to varieties of population size and generation number are 14 and 13 respectively. Moreover, in the 6th pair of experiments with 60 trial runs, times of reach global optimum due to varieties of population size and generation number are 35 and 36 respectively. It can be concluded that population size and generation number are equally cross sensitive to global optimal solution in GAOBDO model.

Table 6.18 Cross Sensitivity of Population Size and Generation Number

Order of Experiment	Number of Trials	Times of Reaching Optimum (Population Size)	Times of Reaching Optimum (Generation Number)
1	10	6	5
2	20	13	14
3	30	17	15
4	40	25	28
5	50	32	30
6	60	35	36
7	70	40	42
8	80	50	48
9	90	56	58
10	100	65	64

Figure 6.28 Cross Sensitivity of Population Size and Generation Number

6.4.3. Summary of Cross Sensitivity Analysis

In this research, in order to further investigate the practicability and efficiency of proposed GAOBDO model, cross sensitivity analysis is conducted to explore contributive effects of crossover rate and mutation rate, as well as population size and generation number. In these experiments, from times of reaching global optimum solution out of its trial numbers, it is concluded that crossover rate is more cross sensitive and more contributive to global optimum solution. However, for cross sensitivity of population size and generation number, there is no obvious difference between times of reaching global optimum out of its trial numbers. It is concluded that population size and generation number are almost equally cross sensitive and are equivalently contributive to global optimum solution.

6.5 Case Study 3 – A Completed Commercial Construction Project

After case study 1 and case study 2, GAOBDO model has been proved feasible on small-sized and medium-sized project networks. In order to prove practicability and effectiveness of GAOBDO model on large-scale projects, a commercial construction project network completed in real practice is applied as case study 3. This project is a 4-storey commercial office building with a basement car park. Ground floor area of the construction project is 5,310 m². It is completed with duration of 377 days. Excluding land costs and other contingency fees, total construction cost is \$12,583,620 and indirect cost is \$620/day. Incentive fee for network compression and penalty for schedule extension are \$500 and \$5,000, respectively. Since it is a commercial construction project financed with bank loan at yearly interest rate of 6.5%, total interest is \$708,570. After divided by project duration of 377 days, interest saving is approximately \$1,880 per day. After the office building is completed, owner gained profit of \$6,183,390 by selling or leasing out office units and car park of the building. As input parameters of GAOBDO model, they are summarized in Table 6.19.

Table 6.19 Cost and Schedule Input Parameters of Case Study 3

Opportunity Incomes:	\$16,440/Day	Interests Saving/Day:	\$1,880/Day
Penalty:	\$5000/Day	All-normal Total Cost:	\$12,583,620
Indirect Cost:	\$620/Day	All-normal Duration:	377 Days
Incentive:	\$500/Day	Project Benefit:	\$6,183,390

In Microsoft Project 2003, the building construction is broken down to a project network of 78 tasks. Each task has five alternative construction methods. Each construction method has its corresponding cost and duration, as shown in Fig. 6.29 (column “Text 1” to column “Text 10”).

6.5.1. GAOBDO Result of Case Study 3

In this trial, following tentative conclusions drawn from sensitivity and cross sensitivity analysis of these 4 GA parameters in above-mentioned two cases, population size and generation number are set to 200 and 150; crossover and mutation rates are set as 0.8 and 0.08, respectively. GAOBDO model takes less than 17 minutes to reach global optimum solution. GAOBDO model result is shown as output interface in Fig. 6.29. After compressing the project network by 19 days (from 377 days to 358 days), maximum overall benefit reaches \$318,410 with a construction total cost increment of

\$28,910 (compressed project total cost minus all normal project total cost). After applying GAOBDO model to this completed project in Microsoft Project 2003, project owner would gain more benefit by \$318,410 (Maximum overall benefit from GAOBDO results), which is 5.1% increase of recorded profit (\$6,183,390) after the project is actually commissioned.

Figure 6.29 GAOBDO Results of Case Study 3

6.5.2. Convergence Results of GA Parameters (Case Study 3)

Based on the findings of GA parameters selection from previous sensitivity studies, convergence study is also conducted to case study 3. In previous two case studies, GA parameters of crossover rate and mutation rate are clearly suggested with appropriate values ranges. In addition, unlike GA parameters of population size and generation number, crossover and mutation rates do not decisively influence the efficiency and computational time of GAOBDO applications. In convergence study of case study 3, population size is firstly studied to determine the appropriate value for efficient applications in tolerable time span. Then convergence study of crossover and mutation rates are conducted with the suggested values of population size.

Figure 6.30 Convergence Result of Population Size (Case Study 3)

Five population sizes in the range of [100, 800] are studied with crossover mutation rates being set to 0.7 and 0.03 respectively. Due to the complexity of project network in this case, compared to the previous case studies, convergence speed is relatively slow even the selected values of population sizes are not too high. As shown in Fig. 6.30, faster convergence is obtained for population sizes of 500 or more. For population sizes of 300 or less, increase in convergence can still be obtained.

Figure 6.31 Convergence Result of Crossover Rate (Case Study 3)

After convergence study of population size is conducted, crossover rate is decided by comparing five values. Convergence result is shown in Fig. 6.31. Similar to the findings from case study 1 and 2, crossover rates of 0.5, 0.7 and 0.9 produce faster convergence results, while crossover rates of 0.1 and 0.3 yield slower convergence.

Figure 6.32 Convergence Result of Mutation Rate (Case Study 3)

Finally, convergence study of mutation rate is conducted and shown in Fig. 6.32. It shows that a lower mutation rate helps GAOBDO converge faster. However, it also shows that, for some runs, GA search terminates without achieving global optimum convergence if a mutation rate was set too high. This is because high mutation rates reduce the ability of GA to refine its search even though a wider range of solution is explored more quickly.

As a global searching and optimization tool, GA is applicable to solve combinatorial optimization problems in the construction project scheduling. Integrating GA and OBDO, GAOBDO model is developed as a macro program in VBA (Visual Basic for Application) and embedded in Microsoft Project 2003. Complicated project network of case study 3 is run in GAOBDO model and reaches global optimum results, which yields a better result than that in real practice, fulfilling objective of maximum overall benefit. It proves that GA is capable of solving OBDO problem in scheduling process of large-scale construction projects. Microsoft Project-embedded GAOBDO model is able to assist decision-making of compressing project network for maximum overall

benefit. Therefore, after applying GAOBDO model to the above 3 case studies, it is noted that the concept of OBDO is feasible and its associated GAOBDO model is applicable to various sized construction projects.

6.6. Summary of Case Studies

Sensitivity and cross sensitivity analysis of these 3 cases studies demonstrate that, with objective of maximum overall benefit, in the process of project scheduling, GAOBDO is able to generate optimized compressed networks in commercial project management software of Microsoft Project 2003.

6.6.1 Comparison with Heuristic Methods

Comparatively, if heuristic methods are applied to case study 1 and 2, it is not impossible to reach optimized solution through reiterations of manual network compressions and overall benefit calculations. Taking case study 1 as an example, from all-normal duration of 90 days, a well-known cost-slope heuristic solution can be derived manually in an iterative process. The process mainly includes “compress and calculate” individual tasks on critical path of project network. In each iterative process of “compression and calculation”, critical tasks are identified and the one with the least cost increment is selected and crashed to its shorter duration. The critical path and construction cost are recalculated, as well as overall benefit. Then next iteration is started. Traditional heuristic methods, however, are in most cases applicable to simple projects of small-sized networks since all compressions can be realized manually with inconvenient calculations of task costs and durations. Although heuristic methods are able to deal with small-sized projects, it probably takes too much time to finish these heuristic-based compressions and calculations, not to mention mistakes that might be made in these iterations. Besides, heuristic methods are almost impossible to handle practical-sized construction projects with dozens, or even hundreds of tasks on their networks. It is not wise to apply heuristic methods to solve such a complicated project network like case study 3. It definitely takes too much time and overturns advantages of heuristic methods. On the contrary, with GAOBDO model being applied to case study 1, it only takes at most several minutes to reach identical optimum solutions that might take half an hour for traditional heuristic approaches. For medium-sized or even

large-sized project networks, as can be seen from case study 2 and case study 3, GAOBDO model is able to search for a satisfactory local optimum solution, or even global optimum solution, in tolerable time span.

In De Jong's (1975) research on GA in function optimization, he studied a series of GA parameters and suggested that good GA performance requires the choice of "high" crossover rate, a "low" mutation rate, and a "moderate" population size. Following these suggestions, in GAOBDO applications to these 3 case studies, 4 GA parameters within various value ranges are adopted for sensitivity analysis. Comments on sensitivity and cross sensitivity analysis of these 4 GA parameters, as well as computational time and local optimum solutions, are introduced in following sections.

6.6.2 Comments on Crossover and Mutation Rates

As for sensitivity analysis of crossover rate, various values in the range of $[0, 1]$ are experimented in 3 case studies. Convergence results support De Jong's (1975) recommendation that crossover rate should be set "high". Among trials in 3 case studies, when crossover rate is set in the range of $[0.5, 1]$, overall performance of GAOBDO model is obviously better than that in the range of $[0, 0.5]$. In 3 case studies, global optimum values of overall benefit are all obtained from the same value range of $[0.5, 1]$. Through sensitivity analysis and convergence study of crossover rate, it is observed that proposed GAOBDO model performs consistently and achieves its objective in all these 3 cases. Genetically, applying GAOBDO model with higher crossover rate allows more chromosomes and gene materials being arbitrarily exchanged in a single population. Comparatively, with high crossover rate, it is easier and more accessible for GAOBDO model to search for more possible solutions in population. Consequently, with reproduction that individual chromosomes being copied according to its calculated overall benefit (objective function of OBDO), it is more likely for GAOBDO model to search for better optimal solutions. Therefore, it is concluded that, given potential benefits of GAOBDO model to various-sized project networks in above case studies, for further applications of end users in practice, a crossover rate within the range of $[0.5, 1]$ is recommended as a good option for OBDO.

As for mutation rate, various values within range of $[0, 0.3]$ are experimented for sensitivity analysis convergence study. Different from crossover rate, there is no

obvious result shows that certain range of mutation rate is responsible for obtaining higher optimum values. Nevertheless, in sensitivity analysis of case 1 and case 2, it is observed that, in the range of $[0, 0.1]$, optimum values without mutation operation (mutation rate equals to 0) are lower than those values obtained when mutation rates are applied. Moreover, mutation rate can not be set too “high” (De Jong 1975). It is also observed that optimum value obviously decreases in the range of $[0.1, 0.3]$, which is even lower than optimum value without applying mutation.

The convergence results from these 3 cases correspond to the suggestion that mutation rate should be “low”. Genetically, if mutation rate is set too high, it would make more chromosomes has higher probability of being arbitrarily changed. As a result, after a number of generations, it is more likely that some “good” gene contents of chromosomes are lost after random changes, which might cause some originally “good” chromosomes being accidentally downgraded to “ordinary” chromosomes. Sensitivity analysis of these 3 cases also proves a theoretical viewpoint that mutation only plays a decidedly secondary role in typical GA applications (De Jong’s 1975). Mutation is needed because, even though reproduction and crossover effectively search and recombine existing solutions in one generation, they may occasionally become overzealous and lose potentially “elite” genetic materials. Therefore, it is concluded that, for applications of end users in practice, a mutation rate in the range of $[0, 0.1]$ is necessary as a good option for GAOBDO model.

Cross sensitivity analysis is also conducted to explore contributive effects of crossover rate and mutation rate. It is concluded that, compared to mutation, crossover is more cross sensitive and contributive to global optimum solution. Results comply with many GA theorists’ (Gen and Cheng, 2000; Goldberg, 1989) common conclusion that, comparatively, crossover plays a decidedly primary role and mutation plays a secondary role in typical GA applications to combinatorial optimization problem like OBDO. It is also noted that the selection of crossover and mutation rates has no direct relationship with size and complexity of the project. For large-scale projects, larger generation number and population size can be chosen to find global optimum, but it inevitably exhibited more processing time.

6.6.3 Comments on Population Size and Generation Number

GAOBDO model is applied to the above 3 cases with various options of population size and generation number. Sensitivity analyses to these two GA parameters are also conducted. Results show that large population size facilitates higher optimum values in GAOBDO searching, so does large generation number. However, increasing population size and generation number also inevitably cause increment of computational time. From sensitivity analysis conducted on case 1 and case 2, it is concluded that increasing generation number are more accessible and more time-effective to search for optimum solutions than increasing population size. Comparatively, increasing population size causes too much computational time even it is also able to search for local optimum and global optimum solutions. But its time-consuming disadvantage might overturn practicability and effectiveness of GA search that makes GAOBDO model superior to traditional heuristic and mathematical methods. Take case study 2 as an example, regarding sensitivity analysis of population size, when generation number is set to 500 in the second experiment (Table 6.14), GAOBDO model took 4253 seconds (less than 71 minutes) to obtain global optimum solution with population size being set to 1200. Comparatively, regarding sensitivity analysis of generation number, when population size is set to 200 in the second experiment (Table 6.16), it takes 2202 seconds (less than 37 minutes) to obtain global optimum solution with generation number being set to 10,000.

Conclusion drawn from sensitivity analysis of population size and generation number corresponds to De Jong's (1975) suggestion that population size should be set "moderate". In the process of typical GA, in one generation, chromosomes of one population stand for possible solutions to optimization problem. No matter good or bad, these chromosomes are reproduced with crossover and mutation operations for the next generation. If generation number is not large enough, simply increasing population size only means that there are more possible solutions for GAOBDO model to search for better solutions. GA search is just random but no clear guidance in a big searching area, which does not make it more likely to guarantee global optimum solution even after reproduction, crossover and mutation. On the other hand, with appropriate population size and moderate generation number, chromosomes in a population would generate better solutions through generations. In each generation, through reproduction, crossover and mutation, better chromosomes with higher fitness values are kept and

worse chromosomes with lower fitness values are abandoned. Their empty positions in population pool are filled with new randomly generated chromosomes for the next generation. In the process of typical GA, overall performance of GA search in “offspring” generation is normally better than that of “parent” generation. Therefore, it keeps its trend and reaches global optimum solution based on a random but “better-offspring-survives” oriented searching mechanism.

From sensitivity analysis of population size and generation number drawn from 3 case studies, it is concluded that either increasing population size or generation number makes GAOBDO model more accessible to global optimum solutions. For the sake of computational time and effectiveness, increasing generation number with a moderate population size makes GA search more time-effective than enlarging population size with considerable generations. Trade-off relationship in computational time between generation number and population size is not quite clear based on case studies. However, it is concluded that increasing both of them moderately improve accuracy and gain better results than solely increasing one of them substantially. Moreover, for practical-sized projects with dozens or hundreds of tasks on its network, it exhibits increase in computational time, particularly with larger number of generation numbers. From these case studies, it is also observed that in large projects, not much improvement of finding global optimum is achieved by using large generation number. For OBDO problems, it is observed that large generation number consumes more computational time than large population size. If a construction project is small-scaled with simple task precedence relationships on the network, a population size in the range of [500, 800] and a generation number in the range of [100, 1200] are suggested as the best combination for GAOBDO application. As for a project with medium or large scaled project network, due to its task number and increased complexity, it is suggested that a population size within the range of [1000, 1200] and a generation number within the range of [100, 2000], can be considered as an appropriate compromise for practical applications.

However, it is necessary to indicate that value ranges of the best population size and generation number are suggested to provide referencing guidance to practical applications. The best compromise of these two GA parameters suggested in this research is problem-dependent due to complexity level of a project network and all

possible crashing contingences of individual tasks. Take case study 3 as an example, a very modest combination of population size and generation number (200, 150) is used to reach to global optimum solution within 17 minutes. Even a project network involves dozens of tasks; precedence relationships among tasks are comparatively simple. In addition, crashing possibilities of each task are quite limited since too much crashing makes cost increment exceeds opportunity income, which is against prerequisite condition of OBDO concept.

Several pairs of experiments are conducted with GAOBDO model respectively for cross sensitivity analysis of population size and generation number. In each pair of experiments, out of the same number of trial runs, times of reaching global optimum solution due to population size variety have negligible difference from those due to generation number variety. Therefore, it is concluded that population size and generation number are equally cross sensitive and have equivalent contributive effects to global optimum solution to OBDO problem.

In this study, it is observed that value selection of crossover and mutation rates does not rest with project complexity. Selection of value ranges for crossover and mutation rates are also highlighted in this study. As for values of population size and generation number, it is suggested that they can be set moderately high with increment of project complexity and project size. A complex and large-sized project has more tasks on its network, each task has several construction methods, and possible solutions of project scheduling are enormous. Large population number and generation number make it more possible for global optimum solution to be included in solution pool through generations. This facilitates GAOBDO model searching for more possible solutions and reaching to global optimum.

6.6.4 Comments on Computational Times and Sub-optimal Solutions

Feasibility of OBDO concept and applicability of GAOBDO model are proved through 3 case studies. Sensitivity and cross sensitivity analysis of 4 GA parameters demonstrate that GAOBDO works effectively to search for local optimum or even global optimum solutions. GAOBDO performs consistently and achieves its ultimate objective of maximizing overall benefit. Applications with large population size and generation number demonstrate increment in computational time, particularly in

large-sized project. Therefore, with a moderate generation number, population size can be reduced to save computational time. Based on De Jong's (1975) suggestion on selecting GA parameters and conclusions drawn from sensitivity analysis results, for a practical-sized project like case study 3, moderate population size and generation number can be used to find optimum solution within tolerable computational time span.

Regarding critical remarks on long computational time of typical GA applications, through various experiments of GAOBDO model on these case studies, it can be diluted by two options. One option is to increase CPU processing speed of computer where GAOBDO model is run. Higher CPU runs much faster with the same satisfactory results than lower ones. The other option is to optimize GA program coding and run more efficiently in the process of searching optimum values. It is suggested to code procedure of GAOBDO in a faster programming language than VBA language compatible with Microsoft Project 2003, such as C++. Accordingly, it can propose improvements to save computational time.

For OBDO problems of large-scale construction projects when global optimum solutions are not so compulsorily demanded, besides fast-running computers and highly efficient coding, trading-off between generation number and population size for sound local optimum solution within tolerable time span is also an available option. The proposed GA-based model can be used as a sub-optimal solution to OBDO problem. Then sub-optimal solutions obtained from GAOBDO model applications can be used as input data of other optimization techniques that is capable of reaching global optimum solution. Integration of GA-based model and other optimization techniques, which is conceived to ensure global optimum with tolerable time span, is suggested for further research development.

Chapter 7 Conclusions and Suggestions

Overview

This chapter concludes completed study and primary contribution of this research, as well as limitations of current research scope. Suggestions to future research extensions are also proposed

In the process of construction project scheduling, a project planner has to make optimization decisions frequently. To achieve ultimate objective of successfully implementing a construction project, optimization of various times, costs, resources and operations of tasks is required in scheduling process, which mainly involve a maximization or minimization problem subject to various influencing factors and constraints. Due to complexity and diversity of optimization problems in project scheduling, formulating and evaluating all possible combinations of solutions, based on constraints and interrelationships of various decision variables, has been increasingly time-consuming when number of feasible solutions increases.

Given a construction project network with tasks to complete according to their precedence relationships to one and another, in process of project scheduling, a common practice is to conduct the so-called time-cost trade-off (TCT) analysis, planning the project network with minimum construction cost. An all-normal project schedule with the least cost caters to economic perspective of minimum project cost. This research extends current research scope and proposes a new concept of overall benefit-duration optimization (OBDO) with its associated GA-based solution model for practical applications in Microsoft Project 2003. Based on this innovative scheme of

scheduling optimization, this research aims to maximum overall benefit. Integrating OBDO concept with GA in solving combinatorial optimization problems, a solution model called Genetic Algorithm Overall-Benefit Duration Optimization (GAOBDO) is formulated. In order to maximize the benefits of proposed model and facilitate applications of end users, it is coded in VBA (Visual Basic for Application) as a macro program in Microsoft Project 2003 for practical usage. Through case studies, sensitivity analysis and cross sensitivity analysis, proposed OBDO concept and its associate GAOBDO model are proved feasible and applicable to real-life construction project networks. This concluding chapter provides a brief summary and discusses some key features of this research. In addition, certain limitations of the research and suggestions to future research directions are also discussed.

7.1 Conclusions

OBDO concept and its associated GAOBDO model, together with their integration with Microsoft Project 2003, succeed in overcoming limitations of previous efforts. Conclusions of this research are as follow:

1. With objective of maximizing overall benefit, OBDO differentiates itself from TCT and caters to economic perspective of project profitability. Different from cost minimization, OBDO maximizes overall benefit by introducing future revenue, interest saving and incentive fee in its objective function;
2. For combinatorial optimization problems like TCT in construction project scheduling, GA-based applications have been employed successfully. Different from TCT objective, integrating OBDO concept with GA mechanism, GAOBDO is proposed as solution model that considers discrete time-cost relationships of tasks as alternative construction methods in the process of network compression;
3. As a critical part of modeling development, GAOBDO is coded in VBA (Visual Basic for Application) for its application in Microsoft Project 2003. Integrating GAOBDO with VBA, the solution model is converted into a macro program for

applications in the popular commercial software;

4. For a construction project network activated in Microsoft Project 2003, GAOBDO provides user-friendly interfaces for input of project information and values of GA parameters. With clear instructions, any end user can easily access and use GAOBDO as a recorded macro program to facilitate GA searching and provide clear interface for optimum solution output;
5. Any proposed model without easy accessibilities to its end users is of little help to its applications in real practice. Implementation of GAOBDO model in Microsoft Project 2003 brings direct conveniences to decision making in project scheduling. After inputting required data and running GAOBDO model in Microsoft Project 2003, it is able to search maximum overall benefit for final output. It is also able to execute project scheduling according to “Gantt Chart” output of the activated project file in Microsoft Project 2003;
6. In order to prove feasibility of OBDO concept and applicability of GAOBDO model in Microsoft Project 2003, 3 various-sized project networks are drawn from journal papers and real practice. Sensitivity and cross sensitivity analysis of four GA parameters are performed to explore their contributive effects to global optimum solution. Convergence results also suggest appropriate combinations of these four GA parameters in dealing with various-sized construction project networks. They can be considered as supportive references for practical applications of end users.

7.2 Limitations of the Research

This research basically fulfills the objectives of formulating a new scheduling optimization concept of OBDO and developing a GA-based solution model to maximize overall benefit. Nevertheless, there are still a number of issues on research limitations that should be addressed.

7.2.1 Research Scope

Main purpose of this research is to develop a new scheduling optimization concept with its associated application model. It is only a start of research on maximizing profitability of a construction project in the process of scheduling. On one hand, concept definition of OBDO prescribes the research scope of this study. On the other hand, assumptions of OBDO and constraints of its objective function also reflect some research areas that current study has not covered. As a prerequisite of OBDO concept, it assumes adequate resource availability in process of construction project scheduling. GAOBDO solution model in resource-constrained project scheduling is not included in current research scope. Therefore, broader research scope needs to be studied as one of directions of future research.

7.2.2 Complexity of Scheduling Optimization Problems

Modern construction projects have common features of capital intensity, long term implementation and complicated networks. These features increase complexity of optimization problems in process of project scheduling. Recently, in the area of scheduling optimization problems, GA-based models have been fast evolved and have good prospects for industrial acceptance. Due to complexity of scheduling optimization problems in modern construction project scheduling, major hindrances of utilizing GAOBDO model for optimum overall benefit of a construction project are immaturity of model and lack, or limited availability of optimization techniques that facilitate more effective applications.

7.2.3 Limitations of OBDO Concept and GAOBDO Model

Although OBDO concept and its associated GAOBDO model are proved feasible and applicable to various-sized project networks, better convergence results of GAOBDO remains to be a key challenge for wide adoption of GA applications in real practice. In conclusion, limitations of OBDO and GAOBDO model are as follows:

1. Concept Assumption

To simplify concept definition, OBDO and its associated GAOBDO model

assumes adequate resource availability in process of overall benefit optimization. This assumption enables completed network compression through task crashing without considering resource-constrained scenario. However, in project scheduling, a project planner often faces situation of maximizing overall benefit under circumstance of limited resource availability.

2. Objective Function Formulation

Influencing variables of project cost, duration and overall benefit are considered in the objective function of OBDO. Formulation of OBDO objective function simplifies the maximization of overall benefit. In this research, time values of future revenue, interest saving and discount rate of bank loan, are not considered.

3. Compatibility to Other Commercial Software

In this research, GAOBDO model is coded as a VBA macro program in Microsoft Project 2003. The solution model is only applicable when it is used together with the commercial project management software. For end users who use different commercial software as project scheduling tool, the GA-based solution model might not be compatible.

7.3 Future Directions of the Research

Further research directions are suggested to fine-tune OBDO concept and its associated GAOBDO model for more effective applications in practice. Emphases of further development in the course of future research are described in the following sections.

7.3.1 OBDO under Uncertainty

In the process of construction project scheduling, OBDO assumes that each task on project network has deterministic time-cost relationship. This assumption is realized by considering time-cost relationship of each task as alternative construction methods with their corresponding times and costs. Each method represents an alternative way to

complete a task. Scheduling optimization models can be classified into two areas: (1) deterministic scheduling; and (2) nondeterministic scheduling (Mulholland and Christian 1999). Traditional heuristic and mathematical methods mostly focus on deterministic situations. However, none of these methods is able to solve nondeterministic problems where time-cost relationships of individual tasks are not apparent or uncertain. Many uncertain variables such as weather, space congestion, and productivity level dynamically affect task durations, and corresponding costs could also change accordingly (Leu *et al.* 1999). Owing to variation of outside environment, task durations may not be so certain. In this case, time and cost of each option to complete tasks may be stochastically distributed (Isidore *et al.* 2001). Recently, researchers have paid more attention to non-deterministic optimization problems in project scheduling as many uncertain variables involved in real practice. Therefore, in future research, time-cost relationship of task under uncertain environment need to be further studied and incorporated in further development of GAOBDO model.

7.3.2 OBDO with Resource Allocation and Resource Levelling

One of the assumptions of OBDO is adequate resource availability in process of network compression. However, in real practice, resource can be limited in process of construction project scheduling. In resource-constrained scheduling problems, resource allocation and resource leveling are two different solutions regarding to their different assumptions. Resource allocation proceeds with premise of limited resource availability below maximum value, its associated solution aims to assign limited resources to project tasks and finish a construction project in shortest duration. On the other hand, objective of resource leveling is to reduce excessively high resource assignments and average period-to-period distribution within predefined project completion deadline. OBDO has to consider limited resource scenario in the course of future research work. Therefore, in future research work, it is suggested to enhance GAOBDO with multi-objectives, combining maximum overall benefit and smooth resource usage under the condition of constrained resource scheduling.

7.3.3 Hybridization of GAOBDO Model

In this research, GA is suggested as an appropriate optimization technique in solving

OBDO problem as it requires only objective function and initial solution pool to start with. In order to deal with random nature of GA applications that requires long computational time in large-scale problems, GA can be hybridized with other optimization approaches to provide a more efficient solution in future research.

More recently, hybridizations of GA and other optimization techniques have been studied in category of construction project scheduling problems (Adeli and Karim 1997; Adeli and Hung 1995; Goh 2000). It can be suggested that GAOBDO model can be also readily hybridized with other optimization techniques to solve a complex combinatorial optimization problem like OBDO. It is observed that GAOBDO is good at finding near-optimal solutions quickly. Rather than completely relying on GA, these near-optimal solutions from GAOBDO can be used as a starting engine for other optimization techniques to generate global optimal solutions more effectively in shorter time span. Therefore, in the course of future research, hybridization of GAOBDO model with more problem-solving power is suggested to increase convergence efficiency, especially in solving complex and large-scale OBDO problems.

7.3.4 Other Influencing Economic Factors

In order to perfect modeling development and make GAOBDO more realistic to real practice, besides interest rate and future revenue included in objective function of OBDO, other economic factors that influence overall benefit and take effects in practice also deserve consideration in the course of future research. For instance, Sunde and Lichtenberg (1995) presented a heuristic TCT model that balances cost, time and resources with a net-present-value (NPV) representation. Similar to their research, Gardiner et al. (2000) proposed that NPV could be used as an ongoing monitor of project time-cost-quality trade-off problems. Khang et al. (1999) also proposed to integrate construction quality into traditional TCT problems. Therefore, similar to TCT problems, some influencing factors related to OBDO problem, such as NPV, and construction quality are suggested to consider in the course of future research work. Besides, due to characteristic of long-term implementation in construction projects, time value of working capitals, inflation rate, discount rate of bank loan, and other economic factors that fluctuate with project duration should also be considered. These

factors can be identified as variables or parameters, and integrated into the objective function of OBDO.

7.3.5 Internet Accessibility of GAOBDO Model

Future study is recommended for extending GAOBDO application to support Internet accessibility. In this research, the proposed GAOBDO model has been integrated in Microsoft Project 2003 for practical usage. In order to narrow geographic distance between two end users who are involved in GAOBDO application but unable to meet each other, in the course of future research work, Internet accessibility of GAOBDO model should be considered. Since the GAOBDO model is coded as a macro program in Microsoft Project 2003, Internet accessibility of GAOBDO can be realized by functions of Internet Work Group in Microsoft Project 2003. After GAOBDO application to an active project in the software, project information with GAOBDO outputs can be exchanged and upgraded on Internet by end users from various geographic locations. No matter how far away these end users to one and another, they can easily keep contact and coordinate project scheduling optimization through Internet accessibility of GAOBDO model.

References

- Abbasi, G. Y., Adnan, M. M. (2001). "Crashing PERT networks using mathematical programming." *Intl. J. Proj. Mgmt.*, 19, 181-188.
- Abeyasinghe, M. C. L., Greenwood, D. J., and Johansen, D. E. (2000). "An efficient method for scheduling construction projects with resource constraints." *Intl. J. Proj. Mgmt.*, 19, 29-45.
- Adeli, H., and Cheng, N.-T. (1993) "Integrated Genetic Algorithm for Optimization of Space Structures" , *J. Aerospace Engrg.*, ASCE, 6(4), 315-328.
- Adeli, H. and Cheng, N.-T. (1994) "Augmented Lagrangian Genetic Algorithm for Structural Optimization" , *J. Aerospace Engrg.*, ASCE, 7(1), 104-118.
- Adeli, H., and Karim, A. (2001). *Construction Scheduling, Cost Optimization, and Management*, Spon Press, London.
- Adeli, H., and Karim, A. (1997). "Scheduling/Cost Optimization and Neural Dynamics Model for Construction." *J. Constr. Engrg. and Mgmt.*, ASCE, 123(4), 450-458.
- Adeli, H., and Hung, S. L. (1995). *Machine Learning Neural Network, Genetic Algorithms, and Fuzzy Systems*, John Wiley & Sons, Inc., Canada.
- Adeli, H., and Park, H. S. (1996). "Hybrid CPN-Neural Dynamics Model for Discrete Optimization of Steel Structures." *Microcomputers in Civ. Engrg.*, 11(5), 355-366.
- Adeli, H., and Park, H. S. (1998). *Neurocomputing for design automation*, CRC Press, Boca Raton, Fla.
- Ahuja, H. N. (1984). *Project Management: Techniques in Planning and Controlling Construction Projects*, Wiley-Interscience, John Wiley & Sons, Inc., Toronto.
- Ahuja, H. N. (1980). *Successful Construction Cost Control*, Wiley-Interscience

References

Publication, John Wiley & Sons, Inc., New York.

Ahuja, H. N. (1976). *Construction Performance Control by Network*, Wiley-Interscience, John Wiley & Sons, Inc., Toronto.

Alshaw, M., and Hassan, Z. (1999). "Integrated models for construction planning: object flow and relationship." *Engrg. Constr. and Archit. Mgmt.*, 6(2), 197-212.

Al-Tabtabai, H., and Alex, P. A. (1999). "Using genetic algorithms to solve optimization problems in construction." *Engrg. Constr. and Archit. Mgmt.*, 6(2), 121-132.

Al-Tabtabai, H., Kartam, N., Flood, I., and Alex, P. A. (1999). "Expert judgment in forecasting construction project completion." *Engr. Constr. and Archit. Mgmt.*, 4(4), 271-293.

Assaf, S. A., Bubshait, A. A., Atiyah, S., and Al-Shahri, M. (2001). "The management of construction company overhead costs." *Intl. J. Proj. Mgmt.*, 19, 295-303.

Bakens, W. (1997). "International trends in building and construction research." *J. Constr. Engrg. and Mgmt.*, ASCE, 123(2), 102-104.

Chan, A. P. C. (2001). "Time-cost relationship of public sector projects in Malaysia." *Intl. J. Proj. Mgmt.*, 19, 223-229.

Chan, D. W. M., and Kumaraswamy, M. M. (2002). "Compressing construction durations: lessons learned from Hong Kong building projects." *Intl. J. Proj. Mgmt.*, 20, 23-25.

Chan, W. T., Chua, D. K. H., and Govindan, K. (1996). "Construction resources scheduling with genetic algorithms." *J. Constr. Engrg. and Mgmt.*, ASCE, 122(2), 125-131.

Chang, S. T. A. (2001). "Defining cost/schedule performance indices and their ranges for design projects." *J. Mgmt. in Engrg.*, ASCE, 17(2), 122-130.

Chua, D. K. H., Chan, W. T., and Govindan, K. (1997). "A time-cost trade-off model with resource consideration using genetic algorithms." *J. Civ. Eng. Syst.*, 14, 291-311.

References

- Clough, R. H., Sears, G. A. and Sears, S. K. (2000) *Construction Project Management, Third Edition*, John Wiley & Sons, Inc., New York.
- Dandy, G. C., and Engelhardt, M. (2001). "Optimal scheduling of water pipe replacement using genetic algorithms." *J. Water. Res. Plan. and Mgmt.*, ASCE, 127(4), 214-223.
- Degoff, R. A., and Friedman, H. A. (1985). *Construction Management, Basic Principles for Architects Engineers and Owners*, John Wiley & Sons, New York.
- De Jong, K. A. (1975). *An analysis of the behavior of a class of genetic adaptive systems*. (Doctor dissertation, University of Michigan). Dissertation Abstracts International 36(10), 5140B. (University Microfilms No. 76-9381)
- Dhanasekar, M. (2000). "Identification of optimal size resources for a repetitive housing construction." *Engrg. Constr. and Archit. Mgmt.*, 7(4), 347-361.
- Easa, S. M. (1989). "Resource leveling in construction by optimization." *J. Constr. Engrg. and Mgmt.*, ASCE, 115(2), 302-316.
- El-Rayes, K., and Moselhi, O. (2001). "Optimizing resource utilization for repetitive construction projects." *J. Constr. Engrg. and Mgmt.*, ASCE, 127(1), 18-27.
- Feng, C. W., Liu, L., and Scott, A. B. (2000). "Stochastic construction time-cost trade-off analysis." *J. Comp. in Civ. Engrg.*, ASCE, 14(2), 117-126.
- Feng, C. W., Liu, L., and Scott A. B. (1997). "Using genetic algorithms to solve construction time-cost trade-off problems." *J. Comp. in Civ. Engrg.*, ASCE, 11(3), 184-189.
- Ferry, D. J., Brandon, P. S., and Ferry, J. D. (1999). *Cost Planning of Buildings Seventh Edition*, Blackwell Science, Ltd., U. K.
- Gad, E. E. (1991). *Neural network evolution using genetic algorithms: Pattern classification and logic gate recognition*, Master dissertation, State University of New York Institute of Technology at Utica/Rome, New York.
- Gardiner, P. D., and Stewart, K. (2000). "Revisiting the golden triangle of cost, time and

References

- quality: the role of NPV in project control, success and failure.” *Intl. J. Proj. Mgmt.*, 18, 251-256.
- Gen, M., and Cheng, R. (2000). *Genetic algorithms and engineering optimization*, John Wiley & Sons, Inc., Toronto.
- Goh, B. H. (2000). “Evaluating the performance of combining neural networks and genetic algorithms to forecast construction demand: the case of the Singapore residential sector.” *Constr. Mgmt. and Econ.*, 18, 209-217.
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley, Reading, Mass.
- Harris, F., and McCaffer, R. (2001). *Modern Construction Management, Fifth Edition*, Blackwell Science, MA.
- Hastak, M., and Baim, E. J. (2001). “Risk factors affecting management and maintenance cost of urban infrastructure.” *J. Infrs. Syst.*, ASCE, 7(2), 67-76.
- Hegazy, T. (1999). “Optimization of resource allocation and leveling using genetic algorithms.” *J. Constr. Engrg. and Mgmt.*, ASCE, 125(3), 167-175.
- Hegazy, T., and Ersahin, T. (2001). “Simplified spreadsheet solutions to overall schedule optimization.” *J. Constr. Engrg. and Mgmt.*, ASCE, 127(6), 469-475.
- Holland, J. H. (1975). *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, Mich.
- Hojjat Adeli., and Nai-Tsang Cheng. (1994). “Augmented Lagrangian Genetic Algorithm for Structural Optimization.” *J. Aerospace. Engrg.*, ASCE, 7(1), 104-118.
- Hojjat Adeli., and Nai-Tsang Cheng. (1992). “Integrated Genetic Algorithm for Optimization of Space Structures.” *J. Aerospace. Engrg.*, ASCE, 6(4), 315-329.
- Isidore, L. J., and Back, W. E. (2001). “Probabilistic optimal-cost scheduling.” *J. Constr. Engrg. And Mgmt.*, ASCE, 127(6), 431-437.
- Jaafari, A. (1996). “Time and priority allocation scheduling technique for projects.” *Intl. J. Proj. Mgmt.*, Vol.14 (5), 289-299, Elsevier Science Ltd and IPMA, U. K.

References

- Karim, A, and Adeli. H. (1999). "CONSCOM: An OO Construction Scheduling and Change Management System." *J. Constr. Engrg. and Mgmt.*, ASCE, 125(5), 368-376.
- Khang, D. B., and Myint, Y. M. (1999). "Time, cost and quality trade-off in project management; a case study." *Intl. J. Proj. Mgmt.*, 17(4), 249-256.
- Khosrowshahi, F. (1997). "The optimum project duration and cost curve for Hong Kong Public housing projects." *Engrg. Constr. and Archit. Mgmt.*, 4(4), 249-269.
- Kog, Y. C., Chua, D. K. H., Loh, P. K., and Jaselskis, E. J. (1999). "Key determinants for construction schedule performance." *Intl. J. Proj. Mgmt.*, 17(6), 351-359.
- Laufer, A., Shapira, A., and Goren, I. (1998). "Implementing an integrative approach to project schedule compression." *Engrg. Constr. and Archit. Mgmt.*, 5(1), 82-91.
- Lee, H. S., and Kyoo, J. Y. (1999). "Application of mathematical matrix to integrate project schedule and cost." *J. Constr. Engrg. and Mgmt.*, ASCE, 125(5), 339-346.
- Leen, S. K., and Paulson, B. C. (1998). "Information management to integrate cost and schedule for civil engineering projects." *J. Constr. Engrg. and Mgmt.*, ASCE, 124(5), 381-389.
- Leu, S. S., Chen, A. T., and Yang, C. H. (2001). "A GA-based fuzzy optimal model for construction time-cost trade-off." *Intl. J. Proj. Mgmt.*, 19, 47-58.
- Leu, S. S., Chen, A. T., and Yang, C. H. (1999). "Fuzzy optimal model for resource-constrained construction scheduling." *J. Comp. in Civ. Engrg.*, ASCE, 13(3), 207-216.
- Leu, S. S., and Hung, T. S. (2002). "A Genetic Algorithm-based Optimal Resource-constrained Scheduling Simulation Model." *Construction Management and Economics.*, 20, 131-141.
- Leu, S. S., and Yang, C-H. (1999). "GA-based multicriteria optimal model for construction scheduling." *J. Constr. Engrg. and Mgmt.*, ASCE, 125(6), 420-427.
- Li, H., Love, P. E. D., and Drew, D. S. (2000). "Effects of overtime work and additional resources on project cost and quality." *Engrg. Constr. and Archit. Mgmt.*, 7(3), 211-220.

References

- Li, H., Cao, J. N., and Love, P. E. D. (1999). "Using machine learning and GA to solve time-cost trade-off problems." *J. Constr. Engrg. and Mgmt.*, ASCE, 125(5), 347-353.
- Li, H., and Love, P. E. D. (1997). "Using improved genetic algorithms to facilitate time-cost optimization." *J. Constr. Engrg. and Mgmt.*, ASCE, 123(3), 233-236.
- Li, S. (1996). "New approach for optimization of overall construction schedule." *J. Constr. Engrg. and Mgmt.*, ASCE, 122(1), 7-14.
- Liu, L., Scott, A. B., and Feng, C. W. (1995). "Construction time-cost trade-off analysis using LP/IP hybrid method." *J. Constr. Engrg. and Mgmt.*, ASCE, 121(4), 446-454.
- Lu, M., and AbouRizk, S. M. (2000). "Simplified CPM/PERT simulation model." *J. Constr. Engrg. and Mgmt.*, ASCE, 126(3), 219-226.
- Man, K. F., Tang, K. S., and Kwong, S. (2001). *Genetic Algorithms: Concepts and Designs (Advanced Textbooks in Control and Signal Processing)*, First Edition, Springer, London.
- Mattila, K. G., and Abraham, D. M. (1998). "Linear scheduling: past research efforts and future directions." *Engrg. Constr. and Archit. Mgmt.*, 5(3), 294-303.
- Microsoft Project. (2002). *Reference manual, Version 2003 for Windows XP*, Microsoft Corporation, One Microsoft Way, Redmond, Ish.
- Moselhi, O. (1993). "Schedule compression using the direct stiffness method." *Can. J. Civ. Eng.*, 20, 65-72.
- Moselhi, O., and El-Rayes, K. (1993). "Scheduling of repetitive projects with cost optimization." *J. Constr. Engrg. and Mgmt.*, ASCE, 119(4), 681-696.
- Mueller, F. W. (1989). *Integrated Cost and Schedule Control for Construction Projects*, Van Nostrand Reinhold, New York.
- Mulholland, B., and Christian, J. (1999). "Risk Assessment in construction schedules." *J. Constr. Engrg. and Mgmt.*, ASCE, 125(1), 8-15.
- Nashwan, D. (1998). "Estimating project and activity duration: a risk management approach using network analysis." *Constr. Mgmt. and Econ.*, 16, 41-48.

References

- Noyce, D. A., and Hanna, A. S. (1997). "Planned schedule compression concept file for electrical contractors." *J. Constr. Engrg. and Mgmt.*, ASCE, 123(2), 189-197.
- O'Brien, J. J. (1984). *CPM in Construction Management, Third Edition*, McGraw-Hill, Inc., Toronto.
- O'Brien, W. J., and Fischer, M. A. (2000). "Importance of capacity constraints to construction cost and schedule." *J. Constr. Engrg. and Mgmt.*, ASCE, 126(5), 366-373.
- Olusegun, O. F., Love, P. E. D., and Li, H. (1999). "Optimal allocation of construction planning resources." *J. Constr. Engrg. and Mgmt.*, ASCE, 125(5), 311-319.
- Pan H. and Ting, S .K. (2002). "A genetic algorithms solution to overall benefit-duration trade-off for project owner." *Proceeding of 4th Asia-Pacific Simulated Evolution and Learning Conference, SEAL'02*, Singapore, 2002, 5pp.
- Park, H. S., and Adeli, H. (1997). "Distributed neural dynamics algorithms for optimization of large steel structures." *J. Struct. Engrg.*, ASCE, 123(7), 880-888.
- Perera, S. (1980). "Linear programming solution to network compression." *J. Constr. Div.*, 106(3), 315-326.
- Perry, J. G., and Barens, M. (2000). "Target cost contracts: an analysis of interplay between fee, target, share and price." *Engrg. Constr. and Archit. Mgmt.*, 7(2), 202-208.
- Pilcher, R. (1994). *Project Cost Control in Construction, Second Edition*, Blackwell Scientific Publications, U. K.
- Que, B. C. (2002). "Incorporating practicability into genetic algorithm-based time-cost optimization." *J. Constr. Engrg. and Mgmt.*, ASCE, 128(2), 139-143.
- Randolph, H. T. (2000). "Schedule acceleration, work flow, and labor production." *J. Constr. Engrg. and Mgmt.*, ASCE, 126(4), 261-267.
- Reda, R., and Carr, R. I. (1989). "Time-cost trade-off among related activities." *J. Constr. Engrg. and Mgmt.*, ASCE, 115(3), 475-486.
- Rosenfeld, Y., Navon, R., and Cohen, L. (1998). "Methodology for prioritizing

References

cost-effective construction technologies.” *J. Constr. Engrg. and Mgmt.*, ASCE, 124(3), 176-184.

Senouci, A. B., and Adeli, H. (2001). “Resource scheduling using neural dynamics model of Adeli and Park.” *J. Constr. Engrg. and Mgmt.*, ASCE, 127(1), 28-34.

Shanmuganayagam, V. (2001). Lecture notes for *Techniques of Project Planning & Control.*, School of Civil & Environmental Engineering, Nanyang Technological University, Singapore.

Son, J., and Skibniewski, M. J. (1999). “Multiheuristic approach for resource leveling problem in construction engineering: hybrid approach.” *J. Constr. Engrg. and Mgmt.*, ASCE, 125(1), 23-31.

Sunde, L., and Lichtenberg, S. (1995). “Net-present-value cost/time trade-off.” *Intl. J. Proj. Mgmt.*, 13(1), 45-49.

Tam, C. M., Tong, T. K. L., and Chan, W. K. W. (2001). “Genetic algorithm for optimizing supply locations around tower crane.” *J. Constr. Engrg. and Mgmt.*, ASCE, 127(4), 315-321.

Ting, S. K., Pan, H. (2003). “A MATLAB-based genetic algorithm solution to overall benefit-duration optimization (OBDO)”. *Proceeding of the 7th International Conference on the Application of Artificial Intelligence to Civil and Structure Engineering*, Egmond-ann-Zee, The Netherlands, 2-4th September.

Vrat, P., and Kriengkairut, C. (1986). “A goal programming mode for project crashing with piecewise linear time-cost trade-off.” *Constr. Mgmt. and Econ.*, 10, 161-172.

Wang, C. H., and Huang, Y. C. (2000). “A new approach to calculating project cost variance.” *Intl. J. Proj. Mgmt.*, 18, 131-138.

Yau, C. (1989). “Proejct compression: A method for speeding up resource constrained projects which preserve the activity schedule.” *Euro. J. Oper. Res.*, 49, 140-152.

Zouein, P. P., and Tommelein, I. D. (1999). “Dynamic layout planning using a hybrid incremental solution method.” *J. Constr. Engrg. and Mgmt.*, ASCE, 125(6), 400-408.

Publications

Conference Papers:

- Ting, S. K., Pan, H. (2003). "A MATLAB-based Genetic Algorithm Solution to Overall Benefit-Duration Optimization (OBDO)". *Proceeding of the 7th International Conference on the Application of Artificial Intelligence to Civil and Structure Engineering*, Egmond-ann-Zee, The Netherlands, 2-4th September.
- Pan H. and Ting, S .K. (2002). "A genetic algorithms solution to overall benefit-duration trade-off for project owner." *Proceeding of 4th Asia-Pacific Simulated Evolution and Learning Conference, SEAL '02*, Singapore, 2002, 5pp.

Appendix

Appendix: Macro Program for Overall Benefit-duration Optimization Using Genetic Algorithm

```
' This Macro facilitates Overall Benefits-Duration Optimization (OBDO)
' Macro Recorded Wed 8/25/04 by Pan Heng. p149870340@ntu.edu.sg
' Macro Program for Overall Benefit-Duration Optimization (OBDO) Using
Genetic Algorithms (GAs)
' *****
' Global Variables Definition
' *****

Public objTask As Object                                ' Global
Variables
Public vntMo, intA, intB As Variant
Public vntStart As Variant, vntLast As Variant
Public sngIndi_Cost As Single, sngPenalty As Single, sngOppo_Income As
Single
Public sngIncentive As Single, sngIntere_Saving As Single
Public sngInitial_Dir_Cost As Single, sngDir_CostSec As Single
Public sngNormalDu As Single, sngSecondCrashDu As Single, sngCrashDu As
Single, sngFirstCrashDu As Single
Public sngInitial_Tot_Cost As Single, sngTot_CostSec As Single,
sngTot_CostSec1 As Single
Public sngOverall_Benefits As Single, sngBest_Option As Single
Public sngCrossover_Rate As Single, sngMutation_Rate As Single
Public intGene() As Integer
Public field_no, nof As Integer
Public intRelFitness() As Integer
Public sngFit_Value() As Single
Public vntTotFitness As Variant
Public intDur_Divisor As Integer
Public intDefDuration_Units As Integer

Sub GAOBDO_Optimize()

    Randomize                                            'Start
Macro by Initializing Variables
    Dim intDur_Divisor As Integer
    Dim i As Integer, j As Integer, m As Integer, k As Integer
    Dim intGene1 As Integer, intGene2 As Integer
    Dim intBest As Integer, intWorst As Integer
    Dim intG As Integer, intGeneration As Integer, intPopulation As
Integer
    Dim blnIdentical As Boolean
    Dim vntSumFitness As Variant
    Dim strMsg As String, strTitle_Bar As String

    strMsg = "Macro of GAOBDO developed by research student Pan Heng, email:
p149870340@ntu.edu.sg"
```

Appendix

```

strMsg = strMsg + vbCr + vbCr
strMsg = strMsg + "This macro facilitates Overall Benefit-Duration
Optimization (OBDO) with genetic algorithms (GAs)"
strMsg = strMsg + vbCr + vbCr
strMsg = strMsg + "End users need to input up to 5 alternative
construction methods for each task with its respective duration and cost"
strMsg = strMsg + vbCr + vbCr
strMsg = strMsg + "Then let GAOBDO help us with the optimized
solution.....shall we proceed?"
vntMB = MsgBox(strMsg, vbInformation + vbYesNo, "GA for Overall
Benefit-Duration Optimization ")
If vntMB = vbNo Then End

intDefDuration_Units = ActiveProject.DefaultDurationUnits
intDur_Divisor = GetDur_Divisor()

OptionsEdit DayLabelDisplay:=0
OptionsSchedule TaskType:=0, DurationUnits:=7
ViewApply Name:="&Gantt Chart"

SelectBeginning ' Selects
the First Cell in the Active Table
For Each objTask In ActiveProject.Tasks
    Do
        k = Form("Alternative Construction Methods")
        If k = 0 Then End
        Loop While Val(objTask.Text1) = 0
        SelectCellDown 1
    Next objTask

' *****
'   User Inputs
' *****

SelectBeginning

strTitle_Bar = "GAOBDO Model"
strMsg = ""

strMsg = "Please Input Daily Opportunity Incomes for Owner due to
Schedule Compression ($)"
sngOppo_Income = Int(Val(InputBox(strMsg, strTitle_Bar, "16400")))

strMsg = "Please Input Daily Interests Saving for Owner due to Schedule
Compression ($)"
sngIntere_Saving = Int(Val(InputBox(strMsg, strTitle_Bar, "1880")))

strMsg = "Please Input Daily Incentive for Contractor due to Schedule
Compression ($)"
sngIncentive = Int(Val(InputBox(strMsg, strTitle_Bar, "500")))

strMsg = "Please Input Daily Penalty for Contractor due to Schedule
Compression ($)"
sngPenalty = Int(Val(InputBox(strMsg, strTitle_Bar, "5000")))

strMsg = "Please Input Total Indirect Cost per Day ($)"
sngIndi_Cost = Int(Val(InputBox(strMsg, strTitle_Bar, "620")))

strMsg = "How Many Genes Would You Like Each Population to be
(Population Number)?"
intPopulation = Int(Val(InputBox(strMsg, strTitle_Bar, "200"))): If

```

Appendix

```

intPopulation = 0 Then End

    strMsg = "Please Input Crossover Rate (Recommended between 0.5-0.9)"
    sngCrossover_Rate = Val(InputBox(strMsg, strTitle_Bar, "0.8"))

    strMsg = "Please Input Mutation Rate (Recommended between 0.01-0.09)"
    sngMutation_Rate = Val(InputBox(strMsg, strTitle_Bar, "0.08"))

    strMsg = "How Many Generations per Cycle Would You Prefer (Generation
Number)?"
    IntGeneration = Val(InputBox(strMsg, strTitle_Bar, "200"))

    LevelingOptions Automatic:=True, Order:=2

    sngInitial_Dir_Cost = 0
    sngNormalDu = ActiveProject.Duration \ GetDur_Divisor

    For Each objTask In ActiveProject.Tasks
        'If objTask.Number1 = 0 Then objTask.Number1 = 1
        Call Get_field_id(objTask.Number1 * 2 - 1)
        objTask.Duration = Val(objTask.GetField(FieldID:=field_no)) *
GetDur_Divisor
        Call Get_field_id(objTask.Number1 * 2)
        sngInitial_Dir_Cost = sngInitial_Dir_Cost +
Val(objTask.GetField(FieldID:=field_no))
    Next objTask

    sngInitial_Tot_Cost = (sngInitial_Dir_Cost + sngIndi_Cost *
sngNormalDu)

    strMsg = "Project All-Normal Direct Cost = " &
Format(sngInitial_Dir_Cost, "$#####0.0") & vbCr + vbCr
    strMsg = strMsg + "Project All-Normal Total Cost = " &
Format(sngInitial_Tot_Cost, "$#####0.0") & vbCr + vbCr
    strMsg = strMsg + "Project All-Normal Duration = " &
Format(sngNormalDu, "###0.0") + " Days."
    vntMB = MsgBox(strMsg, , "GAOBDO Model")

    ReDim intGene(intPopulation + 1, ActiveProject.Tasks.Count) As
Integer
    ReDim intRelFitness(intPopulation) 'As Integer
    ReDim sngFit_Value(intPopulation + 1) 'As Single

    strMsg = "This GAOBDO processing may take a few minutes. Please be
patient and wait !"
    vntMB = MsgBox(strMsg, vbInformation + vbYesNo, "GA for Overall
Benefit-Duration Optimization")
    If vntMB = vbNo Then End

' *****
' Initialize Crashed Duration and Cost
' *****

    vntStart = Time ' Used to Calculate Computational
Time
    LevelingOptions Automatic:=True, Order:=2
    sngDir_CostSec = 0

    For Each objTask In ActiveProject.Tasks
        objTask.Number1 = Int(Rnd * (objTask.Number1) + 1)
        Call Get_field_id(objTask.Number1 * 2 - 1)

```

Appendix

```

    objTask.Duration = Val(objTask.GetField(FieldID:=field_no)) *
GetDur_Divisor
Next objTask

Dim sngFirstCrashDu As Single
Dim sngCostCom As Single
sngFirstCrashDu = ActiveProject.Duration \ GetDur_Divisor
strMsg = strMsg + vbCr + "First Compressed Project Duration : " &
Format(sngFirstCrashDu, "###0.0") & " Days."
vntMB = MsgBox(strMsg, , "GAOBDO")

For Each objTask In ActiveProject.Tasks
    For k = 1 To 5
        Call Get_field_id(k * 2 - 1)
        objTask.Duration = Val(objTask.GetField(FieldID:=field_no)) *
GetDur_Divisor
        Call Get_field_id(k * 2)
        sngCostCom = Val(objTask.GetField(FieldID:=field_no))
        Call Get_field_id(objTask.Number1 * 2)
        If ActiveProject.Duration \ GetDur_Divisor <= sngFirstCrashDu
And sngCostCom < Val(objTask.GetField(FieldID:=field_no)) Then
            objTask.Number1 = k
        End If
    Next k
    Call Get_field_id(objTask.Number1 * 2 - 1)
    objTask.Duration = Val(objTask.GetField(FieldID:=field_no)) *
GetDur_Divisor
    Call Get_field_id(objTask.Number1 * 2)
    sngDir_CostSec = sngDir_CostSec +
Val(objTask.GetField(FieldID:=field_no))
Next objTask

sngSecondCrashDu = ActiveProject.Duration \ GetDur_Divisor
strMsg = strMsg + vbCr + "Second Compressed Project Duration : " &
Format(sngSecondCrashDu, "###0.0") & " Days."
vntMB = MsgBox(strMsg, , "GAOBDO")

LevelingOptions Automatic:=True, Order:=2

If ActiveProject.Duration \ GetDur_Divisor <= sngNormalDu Then
    sngDir_CostSec = sngDir_CostSec + (sngNormalDu -
ActiveProject.Duration \ GetDur_Divisor) * sngIncentive
    'If sngDir_CostSec >= sngInitial_Dir_Cost And
    If (sngNormalDu - ActiveProject.Duration \ GetDur_Divisor) *
(sngOppo_Income + sngIntere_Saving + sngIndi_Cost) >= _
(sngDir_CostSec - sngInitial_Dir_Cost) Then
        sngOverall_Benefits = (sngNormalDu - ActiveProject.Duration \
GetDur_Divisor) * (sngOppo_Income + sngIntere_Saving + sngIndi_Cost) _
- (sngDir_CostSec -
sngInitial_Dir_Cost)
        sngBest_Option = sngOverall_Benefits
    Else
        vntMB = MsgBox("The overall benefits increment is less than cost
increment, which does not satisfy the " _
& "assumptions of OBDO concept. You may choose to quit
the Macro program or run it again with different alternative " _
& "construction methods", vbInformation + 1,
strTitle_Bar)
    Exit Sub
End If

```


Appendix

```

    ElseIf ActiveProject.Duration \ GetDur_Divisor >= sngNormalDu Then
        sngDir_CostSec = sngDir_CostSec - (ActiveProject.Duration \
GetDur_Divisor - sngNormalDu) * sngPenalty
    End If
    sngTot_CostSec = (sngDir_CostSec + sngIndi_Cost *
ActiveProject.Duration \ GetDur_Divisor)

    vntTotFitness = 0 ' Generate Population
of Random Genes
    For k = 1 To intPopulation ' Population Number
is Inputed by User
        For i = 1 To ActiveProject.Tasks.Count
            vntMo = Rnd: intGene(k, i) = Int(Rnd * 5 + 1) ' Genes Contents
Lie in the Range of [1, 5]
        Next i ' Next Task in the
Activated Project
        Call OverallBenefits_Calculate(k) ' Check Every IntGene
by Calculating its Overall Benefits (Fitness Value)
        sngFit_Value(k) = sngOverall_Benefits ' Check Every Gene
    Next k ' Next IntGene in
the Same Population

' *****
' Evaluate Initial Population
' *****

    For intG = 1 To IntGeneration 'Start Genetic
Evolution by Evaluating Current Population
        intG = intG + 1
        vntTotFitness = 0
        For i = 1 To intPopulation: vntTotFitness = vntTotFitness +
sngFit_Value(i): Next i
        For i = 1 To intPopulation: intRelFitness(i) = CInt(100 *
sngFit_Value(i) / vntTotFitness): Next i
        'For i = 1 To intPopulation: intRelFitness(i) = CInt(sngFit_Value(i)
/ vntTotFitness): Next i

' *****
' Two-Point Crossover Operation
' *****

        vntMo = Rnd: intA = Int(Rnd * 100 + 0.5)
        If intA <= Int(sngCrossover_Rate * 100) Then
            'Do
                vntSumFitness = 0: intGene1 = 1: intGene2 = 1 'Select Two Genes
to Marry
                vntMo = Rnd: intA = Int(Rnd * 100 + 0.5)
                vntMo = Rnd: intB = Int(Rnd * 100 + 0.5)
                For i = 1 To intPopulation
                    If intA >= vntSumFitness And intA < vntSumFitness +
intRelFitness(i) Then intGene1 = i
                    If intB >= vntSumFitness And intB < vntSumFitness +
intRelFitness(i) Then intGene2 = i
                    vntSumFitness = vntSumFitness + intRelFitness(i)
                Next i
                blnIdentical = True 'Check if Gene1
and Gene2 are Identical
                For i = 1 To ActiveProject.Tasks.Count
                    If intGene(intGene1, i) <> intGene(intGene2, i) Then
                        blnIdentical = False: Exit For

```

Appendix

```

        Next i
        If blnIdentical = True Then intGene1 = intGene2
    'Loop While intGene1 = intGene2

    Do
        'Two-Point Cross-over
        Operation to Generate an Offspring
        vntMo = Rnd: intA = Int(Rnd * ActiveProject.Tasks.Count) +
1
        vntMo = Rnd: intB = Int(Rnd * ActiveProject.Tasks.Count) +
1
        If intB < intA Then j = intA: intA = intB: intB = j
        Loop While intA = intB
        For i = 1 To intA: intGene(intPopulation + 1, i) =
intGene(intGene1, i): Next i
        For i = intA + 1 To intB: intGene(intPopulation + 1, i) =
intGene(intGene2, i): Next i
        For i = intB + 1 To ActiveProject.Tasks.Count:
intGene(intPopulation + 1, i) = intGene(intGene1, i): Next i
    Else

' *****
'           Mutation Operation
' *****

        vntMo = Rnd: intA = Int(Rnd * 100 + 0.5)
        If intA <= Int(sngMutation_Rate * 100) Then
            vntMo = Rnd: intA = Int(Rnd * intPopulation) + 1
            For i = 1 To ActiveProject.Tasks.Count:
intGene(intPopulation + 1, i) = intGene(intA, i): Next i
            vntMo = Rnd: intA = Int(Rnd * ActiveProject.Tasks.Count) +
1
            intGene(intPopulation + 1, intA) = Int(Rnd * 5 + 0.5)
        End If
    End If

'
*****
*****
'   Check if Offspring is Better than the Worst Population Member
'
*****
*****

        Call OverallBenefits_Calculate(intPopulation + 1)
        sngFit_Value(intPopulation + 1) = sngOverall_Benefits
    '
    Check Offspring
        intWorst = 1: intBest = 1

        For i = 1 To intPopulation
            '
            Find the Worst Network and the Best Network
            If sngFit_Value(i) < sngFit_Value(intWorst) Then intWorst = i
            If sngFit_Value(i) >= sngFit_Value(intBest) Then intBest = i
        Next i

        strMsg = ""
        'Check
        if Offspring is Better than the Worst
        For i = 1 To intPopulation
            strMsg = strMsg + vbCr + "Gene" & i & ": "
            For j = 1 To ActiveProject.Tasks.Count: strMsg = strMsg &
intGene(i, j): Next j
            strMsg = strMsg + vbCr + "GAOBDO Performance = " +

```

Appendix

```

Format(sngFit_Value(i), "#####0.0") + ";      Relative Fitness= " &
intRelFitness(i)
Next i

    strMsg = strMsg + vbCr + vbCr
    strMsg = strMsg + "Offspring No. " & intG
    strMsg = strMsg + vbCr + "                genes " & intGene1 & "
and " & intGene2 & " are selected"
    strMsg = strMsg + vbCr + "                and perform two-point cross-over
from position " & intA & " to " & intB
    strMsg = strMsg + vbCr + vbCr + "Offspring: "

    For j = 1 To ActiveProject.Tasks.Count: strMsg = strMsg &
intGene(intPopulation + 1, j): Next j
    strMsg = strMsg & "      GAOBDO Performance      =" +
Format(sngFit_Value(intPopulation + 1), "#####0.0")

    If sngFit_Value(intPopulation + 1) >= sngFit_Value(intBest) Then
        strMsg = strMsg + vbCr + "                and will replace Gene"
& intBest
        sngFit_Value(intBest) = sngFit_Value(intPopulation + 1)
        For i = 1 To ActiveProject.Tasks.Count: intGene(intBest, i) =
intGene(intPopulation + 1, i): Next i
    Else
        strMsg = strMsg + vbCr + "                and will be discarded"
    End If
    'k = MsgBox(strMsg, , "GAOBDO Results")
Next intG

,
*****
*****
'      Post-processing to Relax Non-Critical Tasks of the Solution
,
*****
*****

'For Each objTask In ActiveProject.Tasks
'Call Get_field_id(objTask.Number1 * 2 - 1)
'objTask.Duration = Val(objTask.GetField(FieldID:=field_no)) *
GetDur_Divisor
'Next objTask                                ' Get Project
Duration of the Optimum Solution
'sngCrashDu = ActiveProject.Duration \ GetDur_Divisor

'Dim Best_Du As Single
'Dim sngTask_Cost As Single

'For Each objTask In ActiveProject.Tasks                                ' Get Task
Durations and Costs of Alternative Construction Methods
'For k = 1 To 5
'Call Get_field_id(k * 2 - 1)                                ' Field
1, 3, 5, 7, 9 Denote Task Durations
'sngTaskDuraion = Val(objTask.GetField(FieldID:=field_no)) *
GetDur_Divisor
'If sngTaskDuraion = 0 Then Exit For
'objTask.Duration = sngTaskDuraion
'Call Get_field_id(k * 2)                                ' Field
2, 4, 6, 8, 10 Denote Task Costs
'sngTask_Cost = Val(objTask.GetField(FieldID:=field_no))
'Call Get_field_id(objTask.Number1 * 2)

```

Appendix

```

        'If ActiveProject.Duration \ GetDur_Divisor <= sngCrashDu And
Val(objTask.GetField(FieldID:=field_no)) > sngTask_Cost Then
        'objTask.Number1 = k
        'sngTot_CostSec1 = sngTot_CostSec1 -
        (Val(objTask.GetField(FieldID:=field_no)) - sngTask_Cost)
        'sngBest_Option = sngBest_Option +
        (Val(objTask.GetField(FieldID:=field_no)) - sngTask_Cost)
        'vntLast = Time
        'sngCrashDu = ActiveProject.Duration \ GetDur_Divisor
    'Else
        'Call Get_field_id(objTask.Number1 * 2 - 1)
        'objTask.Duration =
Val(objTask.GetField(FieldID:=field_no)) * GetDur_Divisor
    'End If
    'Next k
'Next objTask

' *****
'      Report of Final Results
' *****

    strMsg = ""
    strMsg = strMsg + "GA Population Number: " + Format(intPopulation,
"###0") & " Genes      Generation Number: " & Format(IntGeneration,
"###0") + vbCr
    strMsg = strMsg + "Crossover Rate: " + Format(sngCrossover_Rate) &
"      Mutation Rate: " & Format(sngMutation_Rate) +
vbCr
    strMsg = strMsg + vbCr + "All Normal Project Duration: " &
Format(sngNormalDu, "###0.0") & " Days. All Normal Project Total Cost:
" & Format(sngInitial_Tot_Cost, "$#####0.0") + vbCr
    strMsg = strMsg + "All Normal Project Direct Cost: " &
Format(sngInitial_Dir_Cost, "$#####0.0") & "      Project's Indirect
Cost/day: " & Format(sngIndi_Cost, "$#####0.0") + vbCr
    'strMsg = strMsg + vbCr + "All Normal Project Total Cost: " &
Format(sngInitial_Tot_Cost, "$#####0.0")
    strMsg = strMsg + vbCr + "Penalty/day due to Delayed Completion: " &
"$" & sngPenalty
    'strMsg = strMsg + vbCr + "Project's Indirect Cost/day: " & "$" &
sngIndi_Cost
    strMsg = strMsg + vbCr + "Contractor's Incentive/day due to Schedule
Compression: " & "$" & sngIncentive
    strMsg = strMsg + vbCr + "Owner's Opportunity Incomes/day due to
Schedule Compression: " & "$" & sngOppo_Income
    strMsg = strMsg + vbCr + "Owner's Interests Saving/day due to Schedule
Compression: " & "$" & sngIntere_Saving
    strMsg = strMsg + vbCr + vbCr + "GAOBDO Results:"
    strMsg = strMsg + vbCr + "Compressed Project Duration: " &
Format(sngCrashDu, "###0.0") & " Days."
    strMsg = strMsg + vbCr + "Compressed Poject Total Cost: " &
Format(sngTot_CostSec, "$#####0.0")
    strMsg = strMsg + vbCr + "Maximum Overall Benefits: " &
Format(sngBest_Option, "$#####0.0")
    strMsg = strMsg + vbCr + vbCr + "If this solution is not satisfactory
yet, please re-run the macro with different GAS parameters."
    strMsg = strMsg + vbCr + vbCr + "Start of GAOBDO Procedure: " & vntStart
    strMsg = strMsg + vbCr + "The Best GAOBDO Solution Reached on: " &
vntLast
    strMsg = strMsg + vbCr + "End of GAOBDO Procedure on: " & Time
    vntMB = MsgBox(strMsg, , "GAOBDO Results")

```

Appendix

```

End Sub

' *****
' Subroutine For Overall Benefits Calculations
' *****

Sub OverallBenefits_Calculate(geno)
    Dim m As Integer, oi As Integer, k As Integer
    Dim sngTaskDuration 'As Single
    Dim sngTaskCrashDu 'As Single
    Dim sngCrashedDu 'As Single
    Dim sngCrashCost

    sngDir_CostSec = 0
    LevelingOptions Automatic:=False, Order:=2

    For oi = 1 To ActiveProject.Tasks.Count
        'Do
            'm = 0:
            Call Get_field_id(intGene(geno, oi) * 2 - 1)
            'ActiveProject.Tasks(oi).Duration =
            Val(ActiveProject.Tasks(oi).GetField(FieldID:=field_no)) *
            GetDur_Divisor
            sngTaskDuration =
            Val(ActiveProject.Tasks(oi).GetField(FieldID:=field_no)) *
            GetDur_Divisor
            'If sngTaskDuration = 0 Then m = 1: intGene(geno, oi) =
            intGene(geno, oi) - 1
            'Loop While m = 1
            ActiveProject.Tasks(oi).Duration = sngTaskDuration \
            GetDur_Divisor
            'Call Get_field_id(intGene(geno, oi) * 2)
            'sngDir_CostSec = sngDir_CostSec +
            Val(ActiveProject.Tasks(oi).GetField(FieldID:=field_no))
            'Next oi
            sngCrashedDu = ActiveProject.Duration \ GetDur_Divisor

            'LevelingOptions Automatic:=True, Order:=2
            'sngDir_CostSec = 0
            'For oi = 1 To ActiveProject.Tasks.Count
            For k = 1 To 5
                Call Get_field_id(k * 2 - 1)
                ActiveProject.Tasks(oi).Duration =
                Val(ActiveProject.Tasks(oi).GetField(FieldID:=field_no)) *
                GetDur_Divisor
                'ActiveProject.Tasks(oi).Duration = sngTaskCrashDu
                Call Get_field_id(k * 2)
                sngCrashCost =
                Val(ActiveProject.Tasks(oi).GetField(FieldID:=field_no))
                Call Get_field_id(intGene(geno, oi) * 2)
                If ActiveProject.Duration \ GetDur_Divisor <= sngCrashedDu And
                sngCrashCost < Val(ActiveProject.Tasks(oi).GetField(FieldID:=field_no))
                Then
                    intGene(geno, oi) = k
                End If
            Next k
            Call Get_field_id(intGene(geno, oi) * 2 - 1)
            sngTaskCrashDu =
            Val(ActiveProject.Tasks(oi).GetField(FieldID:=field_no)) *
            GetDur_Divisor
            ActiveProject.Tasks(oi).Duration = sngTaskCrashDu

```

Appendix

```

        Call Get_field_id(intGene(geno, oi) * 2)
        sngDir_CostSec = sngDir_CostSec +
Val(ActiveProject.Tasks(oi).GetField(FieldID:=field_no))
    Next oi

'sngDir_CostSec = sngDir_CostSec + (sngNormalDu -
ActiveProject.Duration \ GetDur_Divisor) * sngIncentive

LevelingOptions Automatic:=True, Order:=2

If ActiveProject.Duration \ GetDur_Divisor < sngNormalDu Then
    'If sngDir_CostSec >= sngInitial_Dir_Cost
        sngDir_CostSec = sngDir_CostSec + (sngNormalDu -
ActiveProject.Duration \ GetDur_Divisor) * sngIncentive
        If (sngNormalDu - ActiveProject.Duration \ GetDur_Divisor) _
            * (sngOppo_Income + sngIntere_Saving + sngIndi_Cost) >=
(sngDir_CostSec - sngInitial_Dir_Cost) Then

            sngOverall_Benefits = (sngNormalDu - ActiveProject.Duration \
GetDur_Divisor) * (sngOppo_Income + sngIntere_Saving + sngIndi_Cost) _
                - (sngDir_CostSec -
sngInitial_Dir_Cost)
        Else
            sngOverall_Benefits = 0
        End If
    ElseIf ActiveProject.Duration \ GetDur_Divisor >= sngNormalDu Then
        sngDir_CostSec = sngDir_CostSec - (ActiveProject.Duration \
GetDur_Divisor - sngNormalDu) * sngPenalty
        sngTot_CostSec = sngDir_CostSec + (sngIndi_Cost * sngCrashedDu)
        sngOverall_Benefits = 0
    End If

    If sngOverall_Benefits > sngBest_Option Then
        sngBest_Option = sngOverall_Benefits
        'sngCrashDu = ActiveProject.Duration \ GetDur_Divisor
        sngTot_CostSec = sngDir_CostSec + (sngIndi_Cost *
ActiveProject.Duration \ GetDur_Divisor)
        'strMsg = strMsg + "Compressed Project Duration : " &
Format(sngCrashedDu, "###0.0") & " Days."
        'strMsg = strMsg + vbCr + "Overall Cost : " &
Format(sngTot_CostSec, "$#####0.0")
        'strMsg = strMsg + vbCr + "Overall Benefits : " &
Format(sngBest_Option, "$#####0.0")
        'vntMB = MsgBox(strMsg, , "GAOBDO")
        For oi = 1 To ActiveProject.Tasks.Count
            ActiveProject.Tasks(oi).Number1 = intGene(geno, oi)
        Next oi
        For Each objTask In ActiveProject.Tasks
            Call Get_field_id(objTask.Number1 * 2 - 1)
            objTask.Duration = Val(objTask.GetField(FieldID:=field_no)) *
GetDur_Divisor
        Next objTask
        ' Get Project
Duration of the Optimum Solution
        sngCrashDu = ActiveProject.Duration \ GetDur_Divisor

        vntLast = Time
    End If
End Sub

```

```

' *****

```

Appendix

```

'      Subroutine for identifying data fields
'      *****
Sub Get_field_id(nof)
  Select Case nof
    Case 1
      field_no = pjTaskText1
    Case 2
      field_no = pjTaskText2
    Case 3
      field_no = pjTaskText3
    Case 4
      field_no = pjTaskText4
    Case 5
      field_no = pjTaskText5
    Case 6
      field_no = pjTaskText6
    Case 7
      field_no = pjTaskText7
    Case 8
      field_no = pjTaskText8
    Case 9
      field_no = pjTaskText9
    Case 10
      field_no = pjTaskText10
  End Select
End Sub

Private Function GetDur_Divisor() As Integer
  Select Case intDefDuration_Units
    Case pjMinute '0 denotes minutes
      GetDur_Divisor = 1
    Case pjHour '1 denotes hours
      GetDur_Divisor = 60
    Case pjDay '2 denotes days
      GetDur_Divisor = ActiveProject.HoursPerDay * 60
    Case pjWeek '3 denotes weeks
      GetDur_Divisor = ActiveProject.HoursPerWeek * 60
  End Select
End Function

```