INTEGRATED RESOURCE LEVELING AND CASH FLOW OPTIMIZATION FOR CONSTRUCTION PROJECTS USING SYMBIOTIC ORGANISMS SEARCH ALGORITHM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

BAŞAK SEYİSOĞLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CIVIL ENGINEERING

JANUARY 2020

Approval of the thesis:

INTEGRATED RESOURCE LEVELING AND CASH FLOW OPTIMIZATION FOR CONSTRUCTION PROJECTS USING SYMBIOTIC ORGANISMS SEARCH ALGORITHM

submitted by **BAŞAK SEYİSOĞLU** in partial fulfillment of the requirements for the degree of **Master of Science in Civil Engineering, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Ahmet Türer
Head of the Department, **Civil Engineering** ________________

Prof. Dr. Rıfat Sönmez
Supervisor, **Civil Engineering, METU** ________________

Assist. Prof. Dr. Saman Aminbakhsh
Co-Supervisor, **Civil Engineering, Atılım University** ________________

Examining Committee Members:

Prof. Dr. İrem Dikmen Toker
Civil Engineering, METU ________________

Prof. Dr. Rıfat Sönmez
Civil Engineering, METU ________________

Assist. Prof. Dr. Aslı Akçamete Güngör
Civil Engineering, METU ________________

Assist. Prof. Dr. Güzide Atasoy Özcan
Civil Engineering, METU ________________

Assist. Prof. Dr. Saman Aminbakhsh
Civil Engineering, Atılım University ________________

Date: 27.01.2020

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Başak Seyisoğlu

Signature :

ABSTRACT

INTEGRATED RESOURCE LEVELING AND CASH FLOW OPTIMIZATION FOR CONSTRUCTION PROJECTS USING SYMBIOTIC ORGANISMS SEARCH ALGORITHM

Seyisoğlu, Başak
Master of Science, Civil Engineering
Supervisor : Prof. Dr. Rıfat Sönmez
Co-Supervisor: Assist. Prof. Dr. Saman Aminbakhsh

January 2020, 101 pages

Construction projects are well-known for their inherent complexity as they involve a considerable amount of challenging tasks to be performed in conformance with the contractual documents. Unrealistic schedules that disregard the prominent components such as resources are one of the major reasons for failure of the construction projects. The most widely used scheduling technique is the Critical Path Method (CPM) which usually provides a schedule with unfavorable resource fluctuations. To reduce the negative effects of variations in resource demand, resource leveling methods are usually applied. Moreover, the primary cause of contractors' failure is identified as poor financial management. Cash flow related problems can be overcome by virtue of proper cash flow analysis which secures the contractor's cash flow. To deal with issues appertaining to each constituent part of a construction project, a thorough project schedule, including resources and cash flow analysis, should be readily available prior to project execution. Despite a multitude of endeavors in the literature, preparation of a comprehensive project schedule under consideration of different goals is hard to achieve due to computational expensiveness. In this study, it is suggested that a practical approach is needed for industry practitioners who suffer from lack of an optimization model that integrates resource leveling and cash flow optimization in project scheduling. For this purpose, this thesis introduces a new integrated optimization method, named “Combined Resource and Cash flow (CRC)” that considers resource leveling and cash flow optimization simultaneously. The application of proposed integrated method is demonstrated using an example project from the literature. The solutions are obtained with the implementation of Symbiotic Organisms Search (SOS) algorithm. The solutions revealed that the use of CRC method produces promising results compared to the existing methods.

Keywords: Construction Project Management, Resource Leveling, Cash Flow Optimization, Symbiotic Organisms Search Algorithm

ÖZ

SİMBİYOTİK ORGANİZMA ARAMA ALGORİTMASI KULLANILARAK İNŞAAT PROJELERİ İÇİN ENTEGRE KAYNAK DENGELEMESİ VE NAKİT AKIŞI OPTİMİZASYONU YÖNTEMİ GELİŞTİRİLMESİ

Seyisoğlu, Başak
Yüksek Lisans, İnşaat Mühendisliği
Tez Yöneticisi: Prof. Dr. Rıfat Sönmez
Ortak Tez Yöneticisi: Dr. Öğretim Üyesi Saman Aminbakhsh

Ocak 2020, 101 sayfa

İnşaat projeleri, sözleşmelerle belirlenen gerekliliklere uygun olarak yapılması gereken çok miktarda zorlu işler içerdiğinden, doğasında birçok karmaşıklık barındırmaktadır. Kaynaklar gibi önemli bir bileşeni göz ardı eden gerçekçi olmayan iş planları, inşaat projelerinin başarısız olmasının başlıca nedenlerinden biridir. Kritik Yol Yöntemi (CPM) inşaat projelerinin programlamasında en yaygın kullanılan teknik olmakla birlikte, bu yöntem genellikle istenmeyen kaynak dalgalanmalarını içeren bir iş planı vermektedir. Bu dalgalanmaların olumsuz etkilerini azaltmak için genellikle kaynak dengelemesi yöntemi uygulanır. Öte yandan, zayıf finansal yönetim yüklenicilerin başarısızlığının önemli nedenlerinden biridir. Para akışıyla ilgili sorunların bir kısmı, yüklenicinin nakit akışını sağlayan dengeli nakit akışı analizi sayesinde aşılabilir. Bir inşaat projesinin doğru bir şekilde planlanması için, kaynaklar ve nakit akışı analizinin dahil edildiği kapsamlı bir iş programı, projenin yürütülmesinden önce hazır bulundurulmalıdır. Literatürdeki çabalara rağmen, farklı hedefler göz önünde bulundurularak kapsamlı bir proje programının hazırlanması yoğun hesaplamalar gerektirmesinden dolayı fazla çalışılmamıştır. Bu tezde, kaynak planlaması ve nakit akışı optimizasyonunu proje planına entegre eden mevcut yöntemlerin eksikliği vurgulanmış ve bu amaçla, kaynak dengeleme ve nakit akışı optimizasyonunu aynı anda dikkate alan “birleşik kaynak ve nakit akışı (CRC)” adlı yeni bir entegre optimizasyon yöntemi sunulmuştur. Önerilen entegre yöntemin uygulaması literatürde bulunan örnek bir proje kullanılarak gösterilmiştir. Simbiyotik Organizma Arama (SOS) algoritması kullanılarak elde edilen çözümler, CRC yönteminin kullanımının mevcut yöntemlere kıyasla tatmin edici sonuçlar verdiğini ortaya koymuştur.

Anahtar Kelimeler: İnşaat Proje Yönetimi, Kaynak Dengeleme, Nakit Akışı Optimizasyonu, Simbiyotik Organizma Arama Algoritması

Dedicated to my beloved family…

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor Prof. Dr. Rıfat Sönmez and my co-supervisor Assist. Prof. Dr. Saman Aminbakhsh for their contributions, guidance, encouragement, and invaluable support throughout my research.

I would like to thank to the members of my thesis examining committee; Prof. Dr. İrem Dikmen Toker, Assist. Prof. Dr. Aslı Akçamete Güngör, Assist. Prof. Dr. Güzide Atasoy Özcan for their constructive comments, suggestions, and feedbacks. I also want to thank to the research assistants in the Construction Engineering and Management Divison; Murat Altun and Görkem Eken, for sharing their knowledge and giving support.

I am sincerely thankful to my parents Elmas Seyisoğlu and Arman Seyisoğlu for their endless love and patience throughout my life and my research.

I would like to give special thanks to my dearest friends Övsev Dörtlemez, Cansu Coşkun, Saniye Öktem, and Pınar Bilgin for their lovely friendship and unlimited support. They always make me smile and feel strong. I also want to thank to my friends; Batu Türksönmez, Berk Karakuş, Beste Özyurt, Gizem Bilgin, Gökben Demir, Güncel Vara, Hüseyin Erol, İrem Aktaş, Mert Pehlivan, Murat Vargün, Mücahit Gürbüz, Müge Özgenoğlu, and Volkan Kınay for providing me cheerful motivation.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	
CPM	Critical Path Method
CRC	Combined Resource and Cash Flow
MRD	Maximum Resource Demand
RID	Resource Idle Day
RLP	Resource Leveling Problem
RRH	Release-and-rehire
SOS	Symbiotic Organisms Search

CHAPTER 1

INTRODUCTION

Construction projects are deemed to be complex in terms of a myriad of criteria such as high degree of interdependency among project activities, uncertain project nature, involving participants from different professions, and so forth. This complexity necessitates, even mandates, the use of advanced scheduling techniques in lieu of traditional methods to achieve project success. Apart from having an inherently complex nature, construction projects entail different goals to be achieved in a specific period of time defined in contractual documents. These goals often encompass more than one objective to be optimized simultaneously. All these objectives define the project success criteria, albeit conflicting in general. Efficient project management with the integration of leading-edge technologies would enable to improve the project accomplishment process.

The majority of construction projects are scheduled based on the Critical Path Method (CPM), which results in preparing an early start schedule for project activities. However, this traditional scheduling method lacks the incorporation of resources, which are excessively consumed during execution of a construction activity. Even assigning associated resource usage to project activities is not sufficient enough to obtain a reasonable schedule since the resource demand varies along project duration, which might cause both a decrease in productivity and an increase in cost. Thus, after assigning resources to activities, resource leveling techniques should be performed as a further step to minimize resource fluctuations observed in the CPM schedule.

Besides inadequate project schedules, contractors generally suffer from cash deficit throughout the project life-cycle since the execution of almost every construction activity requires a significant amount of cash. This paucity of cash arises mainly from poor cash flow management. In some construction projects, negative values of net cash flow balance are inevitable due to intrinsic nature of work, such as excessive amount of early payments, lag period between disbursements and receipt money. Being aware of these unfavorable negative cash flow values, a contractor should be prepared against cash deficit by evaluating different alternatives for financing the project. In that case, the contractor has to pay for the interest charge, which, in the end, causes making less profit than anticipated. Hence, efficient cash flow management can be regarded as another indispensable ingredient of project success in addition to preparation of a project schedule with its resouces assigned and leveled.

Although there are numerous researches on resource leveling and cash flow optimization, only one study has attempted to investigate the performance of a project schedule while optimizing both resource fluctuations and cash flow profile (Elazouni & Abido, 2014), which can further be improved. Moreover, this comprehensive optimization problem requires the use of efficient optimization algorithms since different objectives are aimed to be considered simultaneously. Although there exists a growing body of literature that recognizes the superior performance of the SOS algorithm, a dearth of research focuses on the application of the SOS algorithm in scheduling problems within the scope of construction project management domain. In addition, despite the advent of powerful optimization methods, the most widely used project scheduling software packages, still, are not capable of generating optimum schedules considering both resource leveling and cash flow optimization. To this end, the main objective of this research is to provide a practical approach that gives satisfactory results for construction practitioners who seek integrating resource leveling and cash flow optimization into project scheduling techniques.

In this study, the research question of how to develop a practical method that integrates resource leveling and cash flow optimization for construction projects is intended to be answered.

A new optimization method, named as "*combined resource and cash flow (CRC)*" that accounts for resource leveling and cash flow optimization simultaneously, is introduced herein. The use of this method simplifies the existing solution approach designed for multiple objectives in terms of computational effort, significantly.

The comprehensive optimization approach includes the integration of scheduling, resource leveling and cash flow models. To demonstrate the application of the proposed method, an example benchmark project from the literature is used. The optimization algorithm is chosen as a recently introduced meta-heuristic algorithm, called SOS, which gives promising results for many optimization problems studied within the scope of different fields. The proposed integrated method is coded using C# programming language and compiled within Microsoft Visual Studio 2017. As a final step, the results are evaluated for each objective and compared with the solutions of the existing method.

The rest of this thesis is organized as follows. Chapter 2 presents the existing literature including the most relevant studies on resource leveling problem, cash flow optimization, and applications of the SOS algorithm in construction project management field. In Chapter 3, a concise description of the resource leveling problem is given together with the mathematical formulation. In Chapter 4, the fundamentals of cash flow optimization with calculations of necessary parameters to perform cash flow analysis is provided. Chapter 5 presents the SOS algorithm. Chapter 6 introduces the proposed integrated resource leveling and cash flow optimization method. Chapter 7 includes the results associated with solutions of the optimization problem and comparison of these results with the existing solution method. Finally, Chapter 8 concludes this thesis by summarizing the major parts of the study, and by indicating both the limitations of the present work and the suggestions for possible future research.

CHAPTER 2

LITERATURE REVIEW

This chapter presents previous research on the resource leveling problem, cash flow optimization, and applications of a recently introduced meta-heuristic method, Symbiotic Organisms Search (SOS) algorithm, in the construction project management field.

2.1 Resource Leveling Problem

In construction projects, activities consume a significant amount of resources for execution. Therefore, the efficient utilization of resources is essential to achieve success throughout the entire life-cycle of a project. As an effective optimization technique, resource leveling has been widely used to obtain a reasonable resource usage profile; hence, a substantial amount of research on the resource leveling problem (RLP) has been conducted. The aim of the classical RLP is to establish a project schedule in which the fluctuations in resource usage are minimized by means of changing start times of project activities within an allowable range while satisfying precedence relationships among activities and project deadline constraints (Neumann, Schwindt, & Zimmermann, 2003).

The classical RLP and its variants/extensions are categorized into three main groups in terms of solution methods: (1) exact, (2) heuristic, and (3) meta-heuristic methods. It was well-proven by Neumann and Zimmermann (1999) that the general RLP with precedence constraints is nondeterministic polynomial-time hard (NP-hard) combinatorial optimization problem, which triggers researchers to benefit from heuristic and meta-heuristic methods to solve NP-hard problems rather than exact methods. Nevertheless, some researchers used exact methods with the motivation of obtaining optimal solutions.

As a pioneering work, Easa (1989) presented an integer programming formulation to solve the RLP for construction projects. In this study, minimization of absolute value of deviation in resource usage from average resource demand and absolute deviation of resources in consecutive time periods were used as objectives. Bandelloni, Tucci, and Rinaldi (1994) used non-serial dynamic programming and enumeration-based approaches presented by Younis and Saad (1996) to reach an optimal solution of a given RLP. Mattila and Abraham (1998) and Elwany, Korish, Barakat, and Hafez (1998) studied the classical RLP for repetitive projects. Son and Mattila (2004) provided a binary resource leveling model in which allowance for activity splitting is present. Focusing on costs incurred due to resource fluctuation and activity splitting, Hariga and El-Sayegh (2011) proposed a mixed integer programming formulation where the goal is to minimize costs of shutdown and restart in case an activity is split, and costs caused by fluctuations in resource demand. Rieck, Zimmermann, and Gather (2012) applied mixed integer linear programming method and domain-reducing preprocessing techniques to solve the RLP. Since it requires a considerable amount of time to reach an optimum solution, preprocessing technique and linearization of nonlinear objective function were applied to reduce and simplify the problem. In another study, branch-and-bound algorithm was adapted to reach optimum solution of the RLP with minimal lags (Ponz-Tienda, Salcedo-Bernal, & Pellicer, 2017). This study incorporates parallel computing which makes use of cloud computing and multicore network computing to analyze simultaneously a number of sub-problems compromising the entire optimization problem. More recent studies involve soft precedence relations between activities by formulating a mathematical model (Jaskowski & Biruk, 2018), and float loss impact caused by shifting non-critical activities by using non-linear integer programming (El-Sayegh, 2018).

Although exact methods offer an optimum solution for a given problem, they need extensive computational effort, especially for large-scale and complex projects;

hence, are not practical. Therefore, some researchers focused on heuristic methods to solve the RLP, one of which was introduced by Harris (1990). In this study, the minimum moment of the resource usage histogram was used to quantify the resource fluctuations. The start times of activities are determined based on the priority order such that the resource usage histogram tends to have a rectangular shape, i.e., a uniform resource profile, leading to a minimum moment value. Later, the minimum moment approach was modified such that the activities are shifted based on free float and resource rate (Hiyassat, 2000) and using entropy maximization (Christodoulou, Ellinas, & Michaelidou-Kamenou, 2010). Kim, Kim, Jee, and Yoon (2005) proposed a model to enhance the minimum moment approach by extending some characteristics of earlier studies such as a range of resource availability rather than a single fixed resource supply. Also, the Analytical Hierarchy Process (AHP) was used for evaluating the relative importance of distinct resource leveling metrics. Neumann and Zimmermann (1999) presented a priority-rule based heuristic method for the solution of RLP with resource constraints and time lags between project activities. He and Zhang (2013) put forward a priority rule-based forward-backward heuristic so as to overcome the limitations of early studies which are based on static priority rules. In a more recent work, Abdel-Basset, Ali, and Atef (2019) developed a Neutrosophic-Burgess heuristic method, in which the objective function seeks to minimize cost of daily resource fluctuations in lieu of deviations in resource usage, considering stochastic nature of activity durations.

In addition to exact and heuristic methods, many researchers have developed efficient methods such as meta-heuristic algorithms, which possess capability of generating satisfactory results in solving complex optimization problems. Genetic Algorithms (e.g., Chan, Chua, & Kannan, 1996; Hegazy, 1999; Leu, Chen, & Yang, 1999; Leu & Yang, 1999; Leu, Yang, & Huang, 2000; Hegazy & Ersahin, 2001; Leu & Hung, 2002; Senouci & Eldin, 2004; Zhao, Liu, Zhao, & Zhou, 2006; Razavi & Mozayani, 2007; Georgy, 2008; El-Rayes & Jun, 2009; Jun & El-Rayes, 2011; Iranagh & Sonmez, 2012; Ponz-Tienda, Yepes, Pellicer, & Moreno-Flores, 2013; Kaiafa & Chassiakos, 2015; Kim et al., 2016; Li & Demeulemeester, 2016; Li, Xiong, Liu, & Li, 2018), Particle Swarm Optimization (Nikoofal Sahl Abadi, Bagheri, & Assadi, 2018), Simulated Annealing (e.g., Anagnostopoulos & Koulinas, 2010; Piryonesi, Nasseri, & Ramezani, 2019), Evolutionary Algorithms (e.g., Cheng, Tran, & Hoang, 2017; Li, Wang, & Dong, 2019) Shuffled Frog-leaping (Ashuri & Tavakolan, 2015), Ant Colony Optimization (Geng, Weng, & Liu, 2011), Harmony Search (e.g., Ponz-Tienda, Salcedo-Bernal, Pellicer, & Benlloch-Marco, 2017), Symbiotic Organisms Search (Cheng, Prayogo, & Tran, 2016; Prayogo, Cheng, Wong, Tjandra, & Tran, 2018), Bat Algorithm (Li et al., 2019) are among the meta-heuristics implemented for the RLP. Some researchers developed hybrid algorithms by combining existing meta-heuristics to blend the strengths of the individual algorithms while compensating for their weaknesses (e.g., Son & Skibniewski, 1999; Doulabi, Seifi, & Shariat, 2011; Ashuri & Tavakolan, 2012; Kyriklidis, Vassiliadis, Kirytopoulos, & Dounias, 2014).

In order to reflect real-project nature which engages conflicting objectives to be optimized, some researchers considered the single-objective RLP with the inclusion of other objectives. Jun and El-Rayes (2011) established a bi-objective optimization model that minimizes both resource total duration and fluctuations in resource profile under resource availability constraints. The multi-objective GA was implemented to generate Pareto optimal solutions, and the proposed model was integrated in MS Project 2007, which is a frequently used software for scheduling purposes. Similarly, Menesi and Hegazy (2015) considered minimization of project duration as a second objective with the demonstration on practical-size projects. The results were obtained by using Constraint Programming (CP), which incorporates operations research and logic programming. To compare the performances of commonly used multiple objective optimization algorithms, namely Strength Pareto Evolutionary Algorithm-II (SPEA-II), Non-dominated Sorting Genetic Algorithm-II (NSGA-II), and Multi-objective Particle Swarm Optimization (MOPSO), Nikoofal Sahl Abadi et al. (2018) presented a bi-objective optimization model considering minimization of discounted cost of resource fluctuations and minimization of project duration. Instead of project duration, some researchers focused on other objectives, such as minimization of resource availability cost (Koulinas & Anagnostopoulos, 2013) and minimization of changes in schedule (Tang, Liu, & Sun, 2014), with the implementation of a Tabu Search-based yperheuristic algorithm and Constraint Programming method, respectively.

In some studies, researchers further extended the bi-objective optimization problems such that they include more than two objectives. Lue and Yang (1999) proposed an integrated optimization model consisting of time-cost trade-off model and resource-related models. The model includes two phases, namely integration of time-cost trade-off and resource allocation, and resource leveling. In the first phase, duration and cost values are evaluated by Technique for Order Preference by Similariy to Ideal Solution (TOPSIS), which is a multi-attribute decision-making approach to order preferences based on similarity with the ideal solution. In the second phase, the schedule and resource requirement output of the first phase is used to level resources. Then, the best schedule is obtained based on cost output of the first phase and schedule output after the second phase. Using NSGA-II, Zahraie and Tavakolan (2009) studied the time-cost-resource utilization optimization (TCRO) model which aims to minimize total duration, total cost and resource fluctuations. Besides, they applied fuzzy set theory where duration and cost variables are fuzzy to account for uncertainties resulted from unexpected events. Ashuri and Tavakolan (2012) conducted a similar study in terms of problem description and fuzziness of duration and cost variables. However, instead of NSGA-II, they used a hybrid of GA and PSO algorithms, which were proven to be powerful in solving advanced optimization problems. In another study, NSGA-II was employed to solve the multi-mode version of the TCRO problem with resource availability constraints (Ghoddousi, Eshtehardian, Jooybanpour, & Javanmardi, 2013). Considering the possibility of interruption during execution of an activity, Ashuri and Tavakolan (2015) applied SFL algorithm that utilizes strengths of PSO and shuffling complex evolution algorithm, to solve the TCRO problem with activity splitting allowed. The study uses Monte Carlo Simulation to characterize uncertainty in duration of activities, thus, enables capturing several possible combinations to execute an activity. Kim et al.

(2016) drew attention to the limitation caused by fragmented optimization process where resource leveling is performed after finding time-cost trade-offs, which may result in generating sub-optimal solutions. To achieve simultaneous schedule optimization, they proposed a heuristic approach that eliminates this drawback. The presented modified Niched Pareto Genetic Algorithm (NPGA) in this study enhances the quality of solutions compared with traditional methods. Kaiafa and Chassiakos (2015) treated the multiple objective resource-driven optimization problem as a single objective cost minimization problem. To simplify this complex optimization problem, costs associated with resource overallocation, project deadline exceedance and daily resource fluctuations are expressed as a single cost function. To solve this cost optimization problem, they applied the basic GA, which is the most prevalent algorithm among meta-heuristic methods.

2.2 Cash Flow Optimization

Although there exists a plethora of reasons that causes bankruptcy of construction companies, surveys made among industry practitioners revealed financial and budgetary factors as the leading causes of business failures (Arditi, Koksal, & Kale, 2000). These causes are primarily due to inefficient financial and cash flow management (Zayed & Liu, 2014). Therefore, controlling cash transactions and rectifying issues raised due to poor cash flow management are crucial for the success of projects in the construction industry. Accordingly, several studies have linked scheduling techniques with cash flow management to maintain a balance between workflow and available cash.

The finance-based scheduling method was first introduced by Elazouni and Gab-Allah (2004) to prepare schedules of construction projects exposed to cash-related constraints. Based on the Critical Path Method (CPM), the problem was formulated as it gives the optimum solution with the maximum project duration by using integer programming method. The model starts with a basic CPM schedule, then devises a schedule extension scheme to add an extension time increment to the project duration obtained by CPM. This extension scheme facilitates scheduling the activities such that negative cash flow is less than the credit limit while minimizing the extension increment. After this phase, the integer programming model was employed to minimize the shift from last activity under precedence relations, activity shifting and credit limit constraints. Later, Elazouni (2009) presented a heuristic for finance-based scheduling of multiple construction projects. The proposed heuristic identifies available cash for a defined time period, discovers candidate activity schedules, ascertains corresponding required amount of cash, sorts schedules according to the increase in project duration, and schedules every activity according to the chosen schedule alternative. Another study on finance-based scheduling in a multi-project environment also includes a heuristic approach in which activities' start times are determined by polynomial shifting algorithm (Gajpal & Elazouni, 2015). Al-Shihabi and AlDurgam (2017) developed three Max-Min Ant System (MMAS) algorithms in which different heuristic information are used to produce solutions. Again, the objective function is constructed as minimization of project duration constrained by the credit limit. Discussing the effect of credit limit on financing costs and indirect costs, and finance-based scheduling on profit, Elazouni and Metwally (2007) used GA to obtain a financially feasible schedule while maximizing profit. Liu and Wang (2008) pointed out the effect of resource utilization on project cash flow, and proposed a Constraint Programming (CP) model that involves resource-constrained project scheduling and cash flow management to maximize profit from contractor's perspective. Some other examples of finance-based scheduling with profit maximization include studies on repetitive projects with the implementation of commonly used meta-heuristics such as GA (Ali & Elazouni, 2009) and SA (Lucko, 2011). Recently, Alavipour and Arditi (2019a) developed an integrated profit maximization model that evaluates a number of financing alternatives such as short-term loans, long-term loans and lines of credit. In addition, they showed the effect of allowance for extension in project duration on profit considering contractor's point of view. By using Linear Programming (LP), Alavipour and Arditi (2018) presented an explicit formulation of finance-based scheduling problem with the objective of minimizing financing cost while satisfying credit limit and cumulative net balance constraints.

Admitting the fact that developing overall optimized project schedules, where major scheduling and financial components are considered, represent a commonly encountered challenge, increasingly more attention is being paid to the finance-based scheduling problems that incorporate more than one objective. As an example of a bi-objective optimization in cash flow-related studies, Alavipour and Arditi (2019b) analyzed time-cost tradeoff problems with different financing alternatives for contractors. Within the context of this study, GA and Linear Programming (LP) technique were combined to obtain a hybrid algorithm. Profit maximization and financing cost minimization are formulated as objective functions, and constraints are related to cumulative net balance and maximum amount of money defined by lenders of financing alternatives which are short-term loan, long-term loan and line of credit. Tavakolan and Nikoukar (2019) seek to find Pareto optimal solutions that can be used for assisting decision-makers in construction projects. For this purpose, they considered two objectives that need to be optimized simultaneously to achieve project success. The proposed bi-objective model includes minimization of project duration and financing cost as objectives of the problem. Moreover, a sensitivity analysis is performed to measure how changes in financial input parameters affect the cost of financing.

The complexity of construction projects gives rise to the need for analyzing scheduling problems with more than two objectives. Afshar and Fathi (2009) investigated the finance-based scheduling problem considering project duration, required credit, and financing cost as three objectives to be minimized by NSGA-II. In order to reflect uncertainty in project input parameters, they applied fuzzy set theory when modeling direct costs incurred during execution of an activity. In addition, they integrated the risk level accepted by the decision-maker into the model formulation using the α-cut method. In a similar study, with the use of same objectives and solution method, alternatives of line of credit were evaluated for cash procurement (Fathi & Afshar, 2010). The study presented by Abido and Elazouni

(2011) also includes minimization of the same three objectives and the use of SPEA, yet, it differs in terms of the number of projects, i.e., more than one projects involved rather than a single project. In another study carried out in a multi-project environment, El-Abbasy, Zayed, and Elazouni (2012) built an optimization model in which the focus is on project duration, financing cost, and maximum negative cumulative balance minimization by using NSGA-II method.

The importance of resource utilization attracts some researcher's interest, resulting in an increase in trend towards combining resource allocation techniques with cash flow optimization problems. Elazouni and Abido (2014) combined clustering technique with SPEA to solve the multiple objective scheduling problem that aims to minimize accumulated finance cost and resource fluctuations while maximizing profit. To achieve the best compromise solution, they applied fuzzy approach that enables evaluation of relative importance of desired objectives based on membership functions. Considering duration, cost, resource fluctuation and cash flow simultaneously, Elbeltagi, Ammar, Sanad, and Kassab (2016) studied the multi-objective scheduling problem in a broader sense. They formulated four objective functions which are minimization of duration, resource fluctuation, total project cost, and sum of financial charge and fluctuation between overdrafts. PSO algorithm was used to obtain both the Pareto-compromise and best alternative solutions. In a more comprehensive study, El-Abbasy, Elazouni, and Zayed (2016) developed an automated system, called MOSCOPEA, that optimizes multiple objectives related to financial and resource management for multiple construction projects with the use of NSGA-II algorithm. The proposed automated system aims to aid industry practitioners in evaluating trade-offs between different objectives which are associated with portfolio duration, fluctuations in resource usage, total cost, cost of financing, maximum amount of required credit, and profit. Besides, the developed model provides flexibility in choosing a set of objectives to be optimized together and observing the effects of desired objectives on schedules. In a similar study, a generic optimization model was presented to solve the scheduling problems of prioritizing the projects with shared resources, allocating a limited amount of resources efficiently among projects that are simultaneously executed, and satisfying multiple objectives of multiple projects under constraints associated with resources and cash (El-Abbasy, Elazouni, & Zayed, 2017).

As a means of handling the inherent complexity of multi-objective optimization problems, some researchers suggested some practical approaches as well. For instance, Hegazy and Ersahin (2001) proposed a simplified spreadsheet-based model that integrates time-cost trade-off analysis, resource allocation, resource leveling, and cash flow management. To this end, they formulated the objective function as the minimization of total project cost, and utilized GA to find solutions that give minimum total cost under duration, cash, and resource constraints. Instead of total cost, Elazouni and Metwally (2007) constructed a mathematical model with the objective of profit maximization to output solutions of overall-optimized project schedules by using GA. The objective function comprises adjustments to the profit calculation such that both completion penalty/incentive and resource unleveling penalty are included.

2.3 Applications of Symbiotic Organisms Search (SOS) Algorithm in Construction Project Management Problems

The intricate nature of construction projects demands the use of efficient algorithms to solve the optimization problems that are within the scope of construction project management. Observing the potential in many research areas (Ezugwu & Prayogo, 2019), the SOS algorithm has gained interest throughout the past few years among researchers who are studying optimization problems pertinent to construction projects. The preeminent advantage of the SOS algorithm among other meta-heuristic algorithms is being parameter-free, i.e., it does not require any parameter tuning except for maximum number of evaluations and population size, which needs to be set in any other meta-heuristic as well (Cheng & Prayogo, 2014). Motivated by the promising results obtained by the SOS algorithm, which was introduced with an application on a benchmark civil engineering design problem, some researchers have diversified the application areas by opening the gate of construction project scheduling problems.

Cheng et al. (2016) applied the SOS algorithm to resource leveling problem when multiple projects with multiple resources are involved. The discrete version of classical SOS algorithm, in which continuous solutions are transformed into discrete solutions, was first formulated in this study. The discrete version includes developing a function that converts the real-value variables into integer values. To account for differences between levels of resource demand among multiple projects, first, the absolute demand was transformed into relative demand in order to compare resources in terms of quantity. Then, the Analytical Hierarchy Process (AHP) was applied to measure the degree of importance for each resource which provides a pairwise comparison. A two-project case study with three resource types (manpower, fund and equipment) was used to investigate the effectiveness of the algorithm. They compared the performance of the discrete SOS algorithm with Differential Evolution (DE), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO), and the results indicated that the discrete SOS algorithm performed better than other methods in terms of performance measures; which are accuracy, solution stability, and satisfaction.

In a recent study presented by Prayogo et al. (2018), an alternative version of the standard SOS algorithm, named modified SOS, was developed to provide an optimization model to resource leveling problem. The problem was formulated such that minimum value of daily resource fluctuations in resource usage profile is searched under resource availability constraints without changing total project duration. They integrated a neighborhood search mechanism and crowding-based selection operator to one of the phases of the standard version of the SOS algorithm to improve the quality of solutions. A case study of a construction project with 44 activities adapted from Sears, Sears, and Clough (2008) was used for performance evaluation. The solutions obtained by modified SOS algorithms were compared to four optimization algorithms, which are Dynamic Stochastic Selection Multi-member Differential Evolution (DSS-MDE), Self-adaptive Differential Evolution

(SaDE), Resource Leveling based on Differential Evolution (RLDE) and SOS. Value of overall fitness function, moment of resource histogram, peak resource usage, cumulative and maximum variation of resource demand between two consecutive time intervals were used as performance measurement metrics. The results show that the modified SOS converges faster than other algorithms and gives the best fitness value.

Prayogo and Kusuma (2019) investigated the performance of the SOS algorithm on nine different resource leveling objectives, each of them seeks to minimize resource fluctuations by using nine different metrics which were obtained from the previous studies related to resource leveling problems. These metrics are mainly related to quantification of deviations in resource usage, either exist throughout project duration or between two consecutive periods, and peak resource demand. The performance of the SOS algorithm was compared with the PSO method which is a widely used and powerful optimization algorithm. Both algorithms were simulated 30 times with 100 iterations on an example construction project adapted from Sears et al. (2008). Each objective function generates different resource demand histograms as each of them uses different metrics to level resource demand. The best, average and worst solutions were presented together with standard deviations for each fitness value. In eight out of nine objective functions, the average fitness and the standard deviation of the SOS algorithm are significantly lower than that of the PSO algorithm, whereas both algorithms give same results for one objective function, which is minimization of maximum absolute deviation. In addition to objective function evaluations, the convergence rate was recorded and presented for further comparison. The convergence curves show that the SOS algorithm is capable of producing optimal solutions faster than PSO in four out of nine objective functions.

Tran, Chou, and Luong (2019) implemented a bi-objective SOS algorithm to solve the time-cost trade-off problem in repetitive construction projects in which total duration and cost are optimized simultaneously. To describe relationships between activity, time, and production, they used singularity functions which help to define piece-wise linear and nonlinear relations. Additionally, they modified benefit factors, which represent the level of improvement obtained between any two solutions, in the classical version of the SOS algorithm with the intention of achieving a balance between exploration and exploitation. The solutions are selected by the use of non-dominated sorting and crowding entropy approaches, yielding a Pareto front which includes a non-dominated set of solutions. Two case studies of repetitive projects were adopted to validate the proposed method and to examine its competence against other algorithms. For performance measurement purposes, solutions of the developed method were compared with four widely used algorithms; which are Multi-objective Particle Swarm Optimization (MOPSO), the Multi-objective Artificial Bee Colony (MOABC) algorithm, Multi-objective Differential Evolution (MODE), and Non-dominated Sorting Genetic Algorithm-II (NSGA-II). The performance evaluation revealed that the proposed algorithm produces the best quality of solutions along the Pareto front, which proves the superiority of the proposed method against the basic SOS algorithm and other well-performed algorithms.

Tran, Cheng, and Prayogo (2016) addressed the importance of including labor utilization as an additional objective in time-cost trade-off problems since multiple work shift policies are frequently applied in construction projects to fulfill predefined success criteria. To this end, they introduced a multi-objective SOS algorithm that enables trade-off optimization between duration, total cost and utilization of multiple work shifts while preserving network logic and satisfying the constraint of available resources. Minimization of total project duration, total project cost, labor utilization during evening and night shifts were defined as three conflicting objectives. The solutions were expressed as a set of three decision variables which are shift option, priority of each activity, and labor constraint for shift systems. The shift option refers to the feasible shift alternative of an activity, the priority reflects the preference for an activity among other activities, and labor constraint serves for defining total available labors per time period for the selected shift. A set of non-dominated solutions, which helps project managers to select a favorable plan among all alternatives, were obtained by employing non-dominated sorting and crowding entropy approaches. A case study of a construction project adapted from Jun and El-Rayes (2010) was used for testing purposes, and the performance of the algorithm was investigated by comparing the results with widely used state-of-the-art algorithms which are Multi-objective Artificial Bee Colony (MOABC), Multi-objective Differential Evolution (MODE), Multi-objective Particle Swarm Optimization (MOPSO), and Non-dominated Sorting Genetic Algorithm-II (NSGA-II). The results together with statistical analyses indicated that the multi-objective SOS algorithm outperformed the solutions obtained by other methods significantly.

Repetitive construction projects necessitate continuity of tasks performed in different units. Therefore, it is essential to maintain work continuity by means of eliminating waste caused by waiting for common resources consumed by preceding activities to complete their task. Pointing out the significance of increasing the learning curve effect and reducing the idle amount of resources, Tran, Loung-Duc, Doung, Le, and Pham (2018) proposed a multi-objective SOS to schedule repetitive projects considering workflow. This method uses opposition-based learning technique introduced by Tizhoosh (2005) for population initialization and generation jumping. With the target of achieving a better approximation to a solution, this technique involves the use of both current and opposite estimates. The scheduling module captures the objectives of minimization of project duration, cost, and total interruption, and maximization of total project quality. Besides, non-dominated sorting and crowding entropy approaches were applied for population selection. In order to evaluate the capability of the algorithm in generating solutions to scheduling problems of repetitive construction projects, the results obtained from two case studies were compared with Multi-objective Differential Evolution (MODE), Non-dominated Sorting Genetic Algorithm-II (NSGA-II), Multi-objective Particle Swarm Optimization (MOPSO), and Multi-objective Artificial Bee Colony (MOABC). The results of the statistical comparison indicated that the approach is superior to the compared algorithms based on statistical evaluation metrics.

2.4 Discussion on Literature Review

It is inferred from the existing literature presented in the previous sections that resource leveling problem and cash flow optimization have been investigated as two independent research topics in the existing studies, except that of Elazouni and Abido (2014). This study incorporates profit, finance, and resource leveling with the use of a fuzzy approach which yields best compromise solution for a decision-maker. This approach is introduced to handle imprecise nature of decision-makers' judgment, and necessitates assigning subjective weights to each objective function (Dhillon, Parti, & Kothari, 1993); hence, is not capable of giving a rational solution when different optimization problems are combined.

In addition, the previous literature pays meager attention to the construction project management applications of the SOS algorithm. There is a need for further investigation of application areas in the construction management field to come up with promising results for challenging optimization problems. In recognition of these gaps, this study contributes to the body of knowledge in the construction project management domain by proposing a meta-heuristic method based on a recently developed SOS algorithm for solution of the integrated resource leveling and cash flow optimization problem.

CHAPTER 3

RESOURCE LEVELING PROBLEM

Construction project planners often use the Critical Path Method (CPM) for scheduling purposes. Despite being simple and practical for practitioners, CPM disregards resource optimization while analyzing a project network. However, it is incontrovertible that resources play a vital role in project execution as almost every construction activity consumes a significant amount of resources. Therefore, resource management techniques and scheduling methods need to be considered together to achieve project success without causing delays and/or cost overruns.

When resources are assigned to a CPM schedule, undesired resource fluctuations are inevitable primarily due to different amount of resources consumed by activities. Furthermore, overlapping activities lead to resource demand deviations between time periods throughout the project duration. A contractor has to either keep excess resource idle on site or release-and-rehire temporary excess resources. Therefore, these undesirable fluctuations are generally costly for the contractor. To prevent waste of resources on a construction site, resources should be managed efficiently starting from the planning phase and extending over the entire life-cycle of a project.

As a resource management technique, resource leveling has emerged as an effective solution method that enables creating a balanced resource profile. Researchers have developed powerful solution methods to obtain satisfactory results for the Resource Leveling Problem (RLP). The mathematical formulation of the classical RLP, which is adapted from Rieck et al. (2012), is given in the remaining parts of this chapter.

Let project network N be represented by an activity-on-node (AoN) network with V = $\{0, 1, \ldots, n, n+1\}$ consisting of n activities that have to be executed without interruption. 0 and $n+1$ represent two dummy activities for the beginning and the end of the project, respectively.

The start time of activity $i \in V$ is denoted by S_i. If it is assumed that a project starts at time zero, then $S_0 = 0$, and S_{n+1} is equal to the predefined project completion time denoted by T_{max}. The finish time of $i \in V$ is denoted by F_i which is equal to $S_i + d_i$, where d_i is the duration of activity $i \in V$. Based on the network logic, activity $i \in V$ can be started only if its predecessors are completely finished. This constraint is defined by precedence relationships between project activities. Hence, the precedence constraint is expressed by $S_i \geq S_j + d_j$, where $j \in B_i$ and B_i is a set of all predecessors of activity $i \in V$.

A sequence of start times $S = (S_0, S_1, \ldots, S_{n+1})$, where $S_i \geq 0$, $i \in V$, and $S_0 = 0$, is called "schedule". A schedule is called feasible if it satisfies all precedence constraints.

The mathematical formulation of finding an optimal schedule for an objective function f(S) to be minimized is expressed as follows:

$$\text{minimize}\, f(S) \tag{3.1}$$

subject to

$$S_i \geq S_j + d_j \tag{3.2}$$

$$S_0 = 0 \tag{3.3}$$

$$S_{n+1} \leq T_{max} \tag{3.4}$$

$$S_i \geq 0 \tag{3.5}$$

The set of feasible start times of activity $i \in V$ lie within the interval $[ES_i, LS_i]$, where ES_i is the earliest and LS_i the latest start times of activity i under given precedence constraints, respectively. In addition, $ES_0 = LS_0 = 0$ by definition.

The total float, defined by $TF_i = LS_i - ES_i$, $i \in V$, gives the maximum time interval within which activity i can be started without causing a delay in predefined project deadline, T_{max}. An activity i is named "critical" if a delay in its early start time leads to a certain amount of delay in project duration. Thus, the total float equals to zero for critical activities, whereas it is greater than zero for non-critical activities.

The objective functions frequently used for solution of the classical resource leveling problem were revealed by Damci and Polat (2014) which were found as a result of a comprehensive review of the existing literature. The nine different commonly practiced objective functions found by these authors are provided in Table 3.1.With the use of these functions, one can accomplish solving the classical RLP formulated above that gives a more balanced resource profile than that obtained from CPM.

In the formulations presented in Table 3.1, k denotes time period within predefined project duration T_{max}; R_k and R_{k+1} are resource demand in time period k and $k+1$, respectively; $R_{inc,k}$ is defined as the increase in resource demand between time periods k and $k+1$; and Avg represents the average of resource demand during project duration.

It is noteworthy to mention that using a single objective among aforementioned resource leveling objectives could be misleading due to uniqueness of construction projects (Damci & Polat, 2014). In other words, one objective might give satisfactory results for a given construction project, whereas using the same objective to level resources of another project might lead to obtaining poor results. Therefore, it would be better to evaluate the results of different objectives according to the quality of solutions for a specific construction project.

Table 3.1. Most Common Resource Leveling Objectives

Objective	**Formula**
Sum of absolute deviations in resource usage	$f(S) = \sum_{k=1}^{T_{max}} \lvert R_{k+1} - R_k \rvert$
Sum of increases in resource usage	$f(S) = \sum_{k=1}^{T_{max}} R_{inc,k}$
Sum of absolute deviations between resource usage in a time period and average resource usage	$f(S) = \sum_{k=1}^{T_{max}} \lvert R_k - Avg \rvert$
Maximum resource usage	$f(S) = \min[\max(R_k)]$
Maximum deviation in resource usage	$f(S) = \max(\lvert R_{k+1} - R_k \rvert)$
Maximum absolute deviation between resource usage in a time period and average resource usage	$f(S) = \max(\lvert R_k - Avg \rvert)$
Sum of square of resource usage	$f(S) = \sum_{k=1}^{T_{max}} {R_k}^2$
Sum of square of deviation in resource usage	$f(S) = \sum_{k=1}^{T_{max}} (R_{k+1} - R_k)^2$
Sum of square of deviations between resource usage in a time period and average resource usage	$f(S) = \sum_{k=1}^{T_{max}} (R_k - Avg)^2$

In summary, the main idea behind resource leveling can be defined as the effort to minimize undesired resource demand fluctuations by shifting start time of non-critical activities based on their available float while satisfying both precedence and project deadline constraints. It should be noted that some researchers have extended the classical RLP in such a way that different considerations are involved, such as activity splitting, float loss cost, soft precedence relationships, and uncertainty in activities' duration.

CHAPTER 4

CASH FLOW OPTIMIZATION

In construction industry, projects necessitate effective cash flow management since the execution of project activities highly depends on available cash amount. Proper cash flow management ensures that the project can be completed within time and budget. On the other hand, poor financial management might lead to an unfavorable situation where the contractor is unable to finance the project due to cash deficit; hence, has to take out excessive loans to finance the project which causes a decrease in the anticipated profit, or even to go bankrupt due to indebtedness. In some situations, the contractor has to terminate the project because of intensive financial problems. Therefore, financing becomes one of the main concerns in almost every construction project.

In order to carry out cash flow analysis, first, the cash flow components should be identified. The project expenses mainly include direct and indirect costs, and the income of a contractor consists of periodic receipts from the owner depending on the work progress. Figure 4.1 represents a schematic cash flow profile for a typical construction project from the contractor's perspective (Au & Hendrickson, 1986).

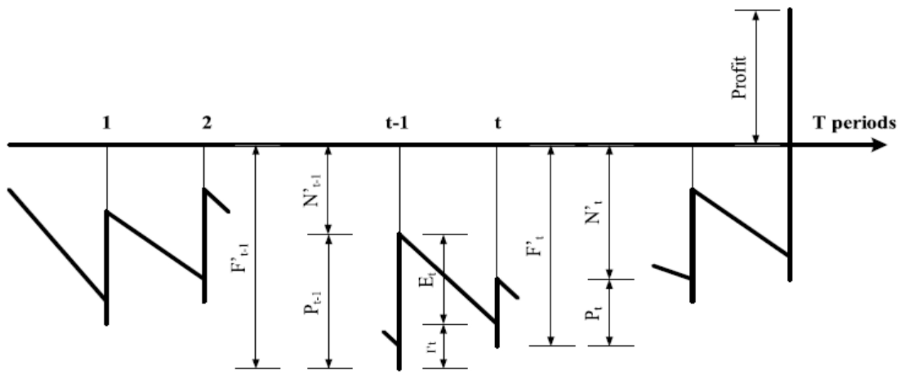

Figure 4.1. Typical Cash Flow Profile for a Construction Project (adapted from Au and Hendrickson (1986))

As can be observed from the cash flow profile, the contractor may suffer from negative cash flow balance during early periods of project life-cycle if the advance payment amount is not sufficient to cover the initial expenses. This is mainly due to initial activities such as mobilization, site preparation and excavation. Furthermore, construction projects are well-known for intensive use of heavy machinery on sites. In addition, there might be cases where installation of heavy equipment, often expensive, is necessary within the scope of a construction project. In these cases, procurement of such equipment generally requires early payment of a significant amount before receiving the corresponding amount from the owner.

The aim of the cash flow optimization is to improve the cash flow profile of the contractor by reducing the negative values, leading to a more financially-feasible project schedule. As mentioned earlier, the first step to perform a reasonable cash flow analysis is to calculate cash flow components accurately, which can only be achieved through the use of a well-defined cash flow model.

The rest of this chapter describes the model formulation associated with cash flow transactions during execution of a construction project. Each term in Figure 4.1 is described in the following parts in detail. The parameters and their corresponding calculations are in conformance with the financial terminology which is introduced by Au and Hendrickson (1986) and elaborated by Elazouni and Metwally (2005).

In a typical construction project, the primary expenses are directly related to the execution of an activity. These expenses are referred to as direct costs, which mainly include labor, material, equipment, and subcontractor expenses.

If there exist n_i overlapping activities on day i, the total direct cost of these activities executed on day i would be calculated as:

$$y_i = \sum_{p=1}^{n_i} y_{pi}, \qquad i = 1,2, \dots, T \tag{4.1}$$

where; y_{pi} represents the direct cost of activity p in day i; and T denotes the total project duration. Assuming the direct disbursement as uniformly incurred throughout the duration of an activity, the daily direct cost can be calculated by dividing the total direct cost by the duration of that activity.

Other than direct costs, the contractor pays the overheads, taxes, bonds etc. which comprise a part of total disbursements. These expenses are called indirect costs, and as the name implies, they are not directly related to the project activities. Instead, they mainly incur during the entire life-cycle of a project. Hence, the total project cost, which represents the cash outflow, is calculated as the summation of total direct and indirect costs:

$$E_t = \sum_{i=1}^{m} y_i + O_t \tag{4.2}$$

where; m is the number of days comprising period t; y_i is the direct cost incurred within that period t; and O_t accounts for the indirect expenses allocated in period t.

The contractor submits payment requests regularly in accordance with the work progress. Once the owner approves progress payments, the contractor receives the monetary value of accomplished work which represents the cash inflow for that period. The following formula gives the value of this receipt amount represented by P_t:

$$P_t = KE_t \tag{4.3}$$

where; K is a multiplier to calculate the amount of payment for a given amount of disbursement E_t based on the contract prices ($K > 1$).

It is a common practice that contractors deposit the payments into a credit-line account so that the outstanding debit, i.e., cumulative negative balance, is reduced. The cumulative balance at the end of period t (for $t > 1$), F_t, is calculated by adding the net cash flow at the end of the previous period, N_{t-1}, and the cash outflow of period t:

$$F_t = N_{t-1} + E_t \tag{4.4}$$

It should be noted that the net cash flow at the end of period t after receiving payment P_t is denoted by N_t, which is also referred to as the net cumulative balance at the end of period t. At the end of period $(t-1)$, F_{t-1} is the cumulative balance, P_{t-1} is the amount of receipt, N_{t-1} becomes the net cumulative balance, i.e., the net cash flow. The following equation gives the relationship between cumulative balance, payment received and net cumulative balance, which can also be observed from Figure 4.1.

$$N_{t-1} = F_{t-1} + P_{t-1} \tag{4.5}$$

Most often, cash procurement through the use of credit line alternatives offered by banks incurs financing costs in accordance with the agreed terms. The financing cost charged by the bank at the end of period t, represented by I_t, can be calculated by the following relations:

$$I_t = \begin{cases} rN_{t-1} + r\dfrac{E_t}{2} & if\ N_{t-1} \leq 0 \\ r\left(\dfrac{E_t - N_{t-1}}{2}\right) & if\ N_{t-1} > 0\ and\ N_{t-1} - E_t < 0 \\ 0 & if\ N_{t-1} - E_t \geq 0 \end{cases} \tag{4.6}$$

where; r is the interest rate per period t.

A positive net cumulative balance of the previous period, N_{t-1}, means that the contractor has no debit. Therefore, the contractor can use the available surplus cash to finance project activities during period t. If the surplus cash can totally cover the amount of total expense at that period, E_t, there is no need to borrow cash; otherwise, the contractor has to pay cost of financing for the borrowed amount exceeding the surplus cash. In situations where N_{t-1} is negative, the contractor has to pay cost of financing which consists of the financing costs charged both on the cumulative net balance N_{t-1} and the cash outflow E_t during period t. The second term represents an approximation of the financing cost using E_t.

In this cash flow model, it is assumed that the contractor pays all the finance costs at the end of the project. The following equation can be used to calculate the compounded periodical financing costs.

$$I'_t = \sum_{l=1}^{t} I_l (1+r)^{t-l} \tag{4.7}$$

where; r represents the interest rate per period t.

The cumulative balance at the end of period t including accumulated financing costs (I'_t), represented by F'_t, is calculated by the equation below:

$$F'_t = F_t + I'_t \tag{4.8}$$

For a given period, the amount of negative cumulative balance F'_t shows the indebtedness of a contractor to the bank, i.e., periodical required finance. The minimum of *F'* values is termed as the "*finance*", which is one of the optimization parameters used in the integrated method proposed in this study.

Finally, the calculation of net cumulative balance at the end of period t including accumulated financing cost, N'_t, is given below:

$$N'_t = F'_t + P_t \tag{4.9}$$

As displayed in Figure 4.1, the positive value of N'_t at the end of project completion is referred to as "*anticipated profit"* of the contractor, which is another objective parameter taken into consideration within the context of this study.

CHAPTER 5

SYMBIOTIC ORGANISMS SEARCH (SOS) ALGORITHM

This chapter is devoted to the description of Symbiotic Organisms Search (SOS) algorithm that is implemented to solve the optimization problem studied in this study. Starting with the basic concept of symbiosis, which forms the phenomenal foundation of SOS, algorithm steps along with the formulations are presented.

In order to eliminate the limitations of the conventional optimization algorithms, researchers among different fields have developed nature-inspired meta-heuristics, which are fast, robust and can produce satisfactory solutions to complex optimization problems. Introduced by Cheng and Prayogo (2014), the SOS algorithm has attracted researchers from various domains, mainly due to its implementation simplicity that results from its parameter-free nature (Ezugwu & Prayogo, 2019).

Cheng and Prayogo (2014) developed the SOS algorithm by inspiring from the interactive relations observed among certain organisms. It is a biological fact that organisms often live together in nature due to reliance on other organisms for sustenance, and even survival. This interdependent relationship is defined as symbiosis, which is derived from the Greek word meaning “living together” (Cheng & Prayogo, 2014). Symbiotic relationships are divided into two classes: (1) obligate, and (2) facultative. Obligate relationship describes a mandatory absolute dependency of an organism to another for survival, whilst facultative relationships involve two distinct organisms cohabitating in a mutually beneficial relationship, but not necessarily.

Like most of the notable metaheuristic algorithms (e.g., Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), etc.) which imitate natural phenomena, the SOS mimics the symbiotic behavior between two different organisms that are used to search for the best solution to a given optimization problem. As a result, this algorithm is constructed based on the following three most typical symbiotic relationships that organisms undergo in nature:

- Mutualism,
- Commensalism,
- Parasitism.

Mutualism indicates a symbiotic relationship between two different organisms in which both get benefit from this interaction. The most well-known mutualism relationship is observed between bees and flowers. Bees benefit from flowers by gathering nectar from them to produce honey. In return, flowers also benefit from this process because bees spread pollens while gathering nectar from many flowers, which helps pollination.

Commensalism represents a symbiotic relationship between two different organisms in which one side receives benefit and other side is not affected either positively or negatively. Generally, the smaller organism is the beneficiary of this type of relationship in certain ways such as providing nutrients or shelter from the larger organism. One common example of commensalism is the relationship between remora fish and sharks. During the interaction, the remora fish adheres itself to the shark and eats food leftovers of its preys; hence, receiving a benefit that is necessary for its existence. The shark is unaffected by this relationship even it provides food for the remora fish.

If one organism benefits and the other gets damage as a result of a symbiosis relationship, this interaction is called parasitism. The organism that harms the other one is called a parasite. A prevalent occurrence of this kind is the relation between Plasmodium parasite and a human. This parasite invades human body through the Anopheles mosquitoes, and reproduces itself inside human host very quickly. Consequently, the human host suffers from a serious disease called malaria, and may even die.

As a result of inspiration from symbiosis relationships, Cheng and Prayogo (2014) defined phases of the SOS, which constitute the main structure of the algorithm, such that they reflect the real-world biological interaction among different species. The feature of behaviors in aforementioned symbiosis relationships is an apt description of the primary principle used in each phase. In the mutualism phase, two solutions are intended to be enhanced based on mutualistic behaviors. In the commensalism phase, only one of the solutions has a chance to improve after the interaction of two solutions. The parasitism phase involves creation of a parasite organism from a solution, and this parasite competes with another organism in the population. Each organism in the ecosystem sequentially interacts with a different randomly selected organism in each phase until the predetermined termination criterion is satisfied.

Apart from being a nature-inspired algorithm, the SOS is also a population-based search algorithm that facilitates iterative use of a population of candidate solutions in the search space during the course of finding the global optimum solution. Similar to other population-based algorithms, the SOS starts with creation of an initial population called the "ecosystem". The initial ecosystem contains randomly generated organisms, each of which represents a candidate solution to a predefined problem. This initial ecosystem can be expressed as the following vector of individuals:

$$X = \{X_1, X_2, \dots, X_{ecosize}\} \quad (5.1)$$

where; *ecosize* denotes the number of organisms in the population.

Every organism in the ecosystem has a fitness value which indicates degree of adaptation, i.e., how "good" the solution is, to the desired objective. The selection of organisms for the next generation depends on the fitness value that is associated with evaluated organisms.

The flowchart of the SOS algorithm is depicted in Figure 5.1. In summary, the process starts with the ecosystem initialization; then, best organism is identified prior to organism interactions defined by mutualism, commensalism, and parasitism phases until stopping condition is reached. Detailed explanations of the algorithm phases along with mathematical formulations are given in the following sections.

5.1 Mutualism Phase

In the first phase of the algorithm, two new organisms are produced in accordance with the characteristics of mutualistic relationships with the aim of increasing their survival advantage within the ecosystem.

An organism X_j is selected randomly from the ecosystem to interact with the current organism X_i. During this phase, both organisms undergo mutualistic changes, resulting in new candidate solutions, X_{inew} and X_{jnew} which are calculated by using Equations 5.2 and 5.3.

$$X_{inew} = X_i + rand(0,1) \times (X_{best} - X_{mutual} \times BF_1) \qquad (5.2)$$

$$X_{jnew} = X_j + rand(0,1) \times (X_{best} - X_{mutual} \times BF_2) \qquad (5.3)$$

where; X_i and X_j represent the i^{th} organism and the randomly selected j^{th} organism in the ecosystem, respectively; organism with the best fitness value is referred to as X_{best}; and *rand(0,1)* denotes uniformly distributed random numbers within the range of [0, 1]. The mutualistic relationship is expressed with a mutual vector X_{mutual} and a benefit factor. The mutual vector is defined as the average of organisms X_i and X_j as shown in Equation 5.4.

$$X_{mutual} = \frac{X_i + X_j}{2} \qquad (5.4)$$

Benefit factors, BF_1 and BF_2, reflect the variation in the level of benefit achieved from one organism. In nature, a mutualistic relationship might give a greater beneficial advantage to one organism than another, or equal beneficial advantage might be received from both organisms. For example, organism X_i might receive a significant benefit during interaction with X_j, whereas organism X_j might only get a slight benefit, or vice versa. Therefore, benefit factors are determined randomly as either 1 or 2 which refers to receiving partial and full benefits, respectively.

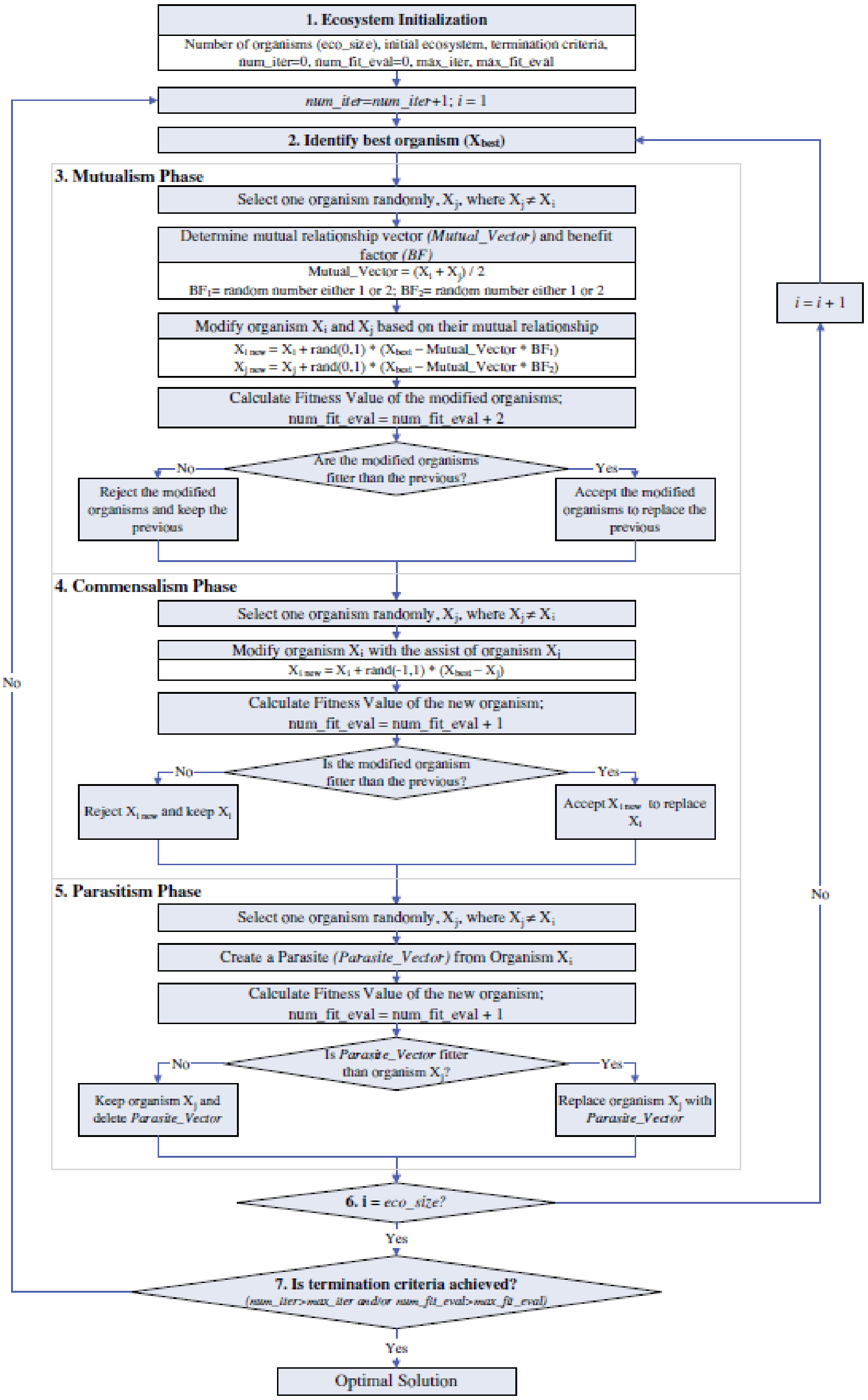

Figure 5.1. Flowchart of the SOS Algorithm (adapted from Cheng and Prayogo (2014))

In Equations 5.2 and 5.3, (X_{best} − X_{mutual} × BF) represents the intrinsic behavior of organisms to increase their survival advantage through mutualistic relationships, which helps them to increase their degree of adaptation to the ecosystem. Hence, X_{best} is used in Equations 5.2 and 5.3 to lead organisms X_i and X_j to a point with a higher degree of adaptation to the surroundings in nature.

At the end of this phase, organisms X_i and X_j are replaced by new organisms only if their post-interaction fitness value is better than their pre-interaction fitness value.

5.2 Commensalism Phase

The mutualism phase is followed by the commensalism phase in which a new organism, X_j, is selected randomly from the ecosystem to interact with X_i as in the mutualism phase. In this phase, organism X_i tries to take advantage from the interaction with X_j. Meanwhile, organism X_j neither benefits nor get damaged from this commensal relationship.

The new candidate solution, X_{inew}, is calculated based on the commensal interaction between organisms X_i and X_j, which is formulated as Equation 5.5.

$$X_{inew} = X_i + rand(-1,1) \times (X_{best} - X_j) \tag{5.5}$$

where; X_i and X_j are the i^{th} organism and randomly selected j^{th} organism in the ecosystem, respectively; X_{best} denotes the organism possessing the best fitness value; and *rand(-1,1)* represents uniformly distributed random numbers within the range of [−1, 1]. The expression (X_{best} – X_j) reflects the beneficial advantage provided by organism X_j in order to aid organism X_i in enhancing its survival advantage in the ecosystem to the highest degree of adaptation, represented by X_{best}.

After commensal interaction, the new organism X_{inew} replaces the organism X_i only if its fitness value is better than that of X_i.

5.3 Parasitism Phase

Once the commensalism phase is finished, organism X_i enters the parasitism phase. In this phase, organism X_i plays a role akin to that of the Anopheles mosquito through creation of an artificial parasite vector, $X_{parasite}$. $X_{parasite}$ is generated by first duplicating organism X_i, then modifying randomly selected dimensions of organism X_i. Similar to previous phases, an organism X_j is selected randomly from the ecosystem, and it serves as a host to the parasite like human body in the case of plasmodium parasite interaction mentioned above.

Without generating a new candidate solution with interaction equations, $X_{parasite}$ simply attempts to replace the position of X_j in the ecosystem. If $X_{parasite}$ has a better fitness value, it will paralyze organism X_j and owns its position in the ecosystem. If the fitness value of X_j is better than $X_{parasite}$'s, X_j gains immunity from the parasite and $X_{parasite}$ will no longer exist in that ecosystem.

CHAPTER 6

PROPOSED METHOD FOR INTEGRATED RESOURCE LEVELING AND CASH FLOW OPTIMIZATION

This chapter includes a description of the method that is proposed in this thesis. Firstly, the scheduling model, resource leveling model, and cash flow model, which lay the foundation of the optimization problem within the context of this study, are provided. Afterwards, the proposed method that incorporates resource leveling and cash flow optimization is presented prior to implementation of the Symbiotic Organisms Search (SOS) algorithm.

The research methodology presented in Figure 6.1 is followed. The proposed method integrates three models: (1) scheduling model, (2) resource leveling model, and (3) cash flow model. The details of model features are given in the next three sections. An example project from the literature is used both for demonstration and comparison purposes. Being a newly introduced powerful meta-heuristic algorithm, the SOS algorithm is employed to obtain solutions. Finally, the results are discussed and compared with the existing method.

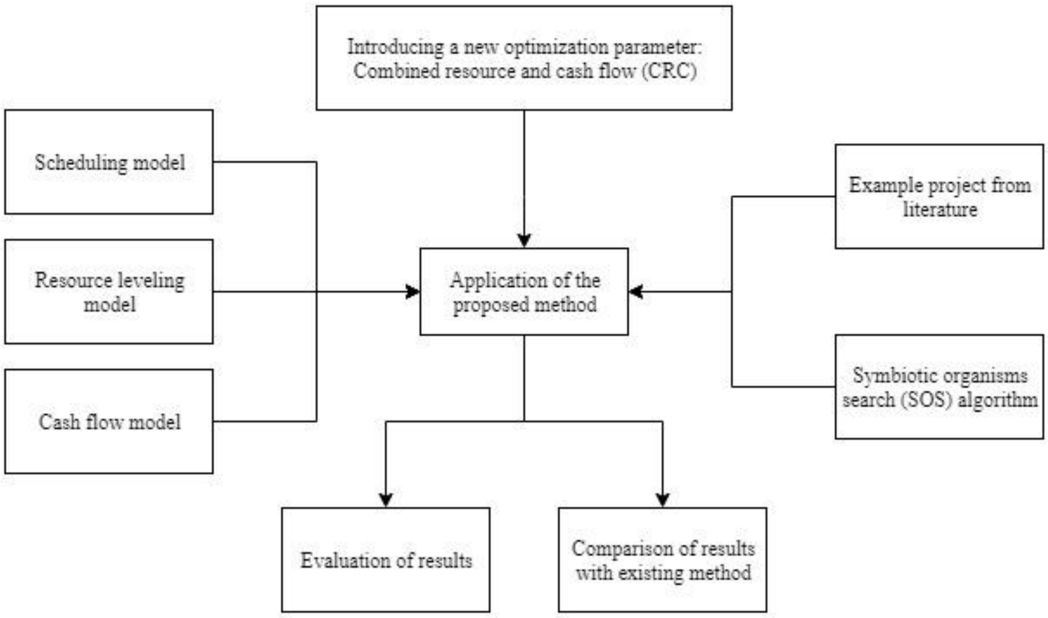

Figure 6.1. Research Methodology

6.1 Scheduling Model

The scheduling model involves the use of the Critical Path Method (CPM) which is the most widespread scheduling network analysis technique. Since its development, project planners have applied this method in order to create a thorough schedule.

The CPM mainly consists of two sequential phases; forward pass and backward pass. By assuming that each activity starts as soon as possible, the former gives the early start time (EST) and early finish time (EFT), whereas the latter outputs the late start time (LST) and late finish time (LFT) of an activity.

To carry out CPM calculations, the following activity-related information should be known:

- ID number,
- Duration,
- Predecessor(s) or successor(s).

With the above information in hand, first, forward pass is performed to determine the EST of each activity considering precedence relationships. Once the EST of each activity *i* is identified, the EFT values are computed by simply adding the duration (D) of that activity to the EST as given below.

$$EFT_i = EST_i + D_i \tag{6.1}$$

After calculating the early start and finish times of project activities, backward pass is employed to determine late start and finish times. The backward pass computations are initiated by equating the EFT of the last activity, i.e., the activity with no successor(s), to the LFT of that activity, then, the LFT of remaining activities can be found by following precedence relationships. The computations then proceed by subtracting duration of activity *i* from its LFT to obtain LST as given in the following equation.

$$LST_i = LFT_i - D_i \tag{6.2}$$

Apart from identifying start and finish times, the CPM enables measuring flexibilities of certain activities according to floats. The total float (TF) of activity *i*, which can be calculated by Equation 6.3, defines the range an activity can be started within, without causing any delay in total project duration.

$$TF_i = LST_i - EST_i \tag{6.3}$$

The activities with zero total float are named as critical activities, which form the critical path for a given project network. The remaining activities are defined as non-critical activities, which have the flexibility to be shifted within their EST and LST without affecting completion time of the project. The method proposed in this study can be applied to a project network with a number of activities having multiple execution modes, each of which has different duration, resource demand, and direct cost values. Thus, the duration, resource usage, and direct cost of an activity depend on the selected execution mode of that activity. Once the execution modes are selected for each activity, EST, LST, EFT, and LFT values can be found by CPM calculations. Then, the total float of each activity can be computed using Equation 6.3.

In this thesis, it is aimed to benefit from changing the start times of non-critical activities so that an optimized schedule is obtained with the integration of resource leveling and cash flow optimization. Therefore, the decision variables of the optimization problem represent both the selected execution modes and selected start times of project activities. This is mainly achieved by benefitting from total float values of non-critical activities. The execution modes and start times are selected based on the interval rules defined by Equations 6.5 and 6.7.

If m_i is the number of available execution modes of activity i, then mode selection interval, I_m, is defined as follows.

$$I_i^m = {}^{1}/_{m_i} \tag{6.4}$$

The execution mode of an activity i is chosen by using the following equation:

$$SM_i = \begin{cases} 1 \quad if \quad 0 < rnd_i^m \leq I_i^m \\ 2 \quad if \quad I_i^m < rnd_i^m \leq 2 \times I_i^m \\ \cdot \qquad\qquad\qquad \cdot \\ \vdots \qquad\qquad\qquad \vdots \\ m_i \quad if \quad (m_i - 1) \times I_i^m < rnd_i^m \leq m_i \times I_i^m \end{cases} \tag{6.5}$$

where; rnd_i^m represents random numbers between 0 and 1; and SM_i is the selected execution mode of activity i.

If tf_i is the total float of activity i obtained from CPM calculations, then selection interval for start time, I_s, is defined as follows:

$$I_i^s = {}^{1}/_{(tf_i + 1)} \tag{6.6}$$

The start time of an activity i is selected according to Equation 6.7.

$$SST_i = \begin{cases} 1 & if \quad 0 < rnd_i^s \leq I_i^s \\ 2 & if \quad I_i^s < rnd_i^s \leq 2 \times I_i^s \\ \cdot & \qquad\qquad \cdot \\ \vdots & \qquad\qquad \vdots \\ tf_i + 1 & if \quad tf_i \times I_i^s < rnd_i^s \leq (tf_i + 1) \times I_i^s \end{cases} \tag{6.7}$$

where; rnd_i^s represents random numbers between 0 and 1.

In summary, the steps followed by scheduling model can be outlined as follows:

i. Generate random numbers between 0 and 1 for each activity to be used in execution mode selection.
ii. Select execution mode of each activity based on interval rule defined in Equation 6.5.
iii. Carry out CPM calculations to determine EST, LST, EFT, LFT, and TF of each activity.
iv. Generate random numbers between 0 and 1 for each activity to be used in start time selection.
v. Select start time of each activity based on interval rule defined in Equation 6.7.
vi. Schedule activities between start and end times.

The output information of scheduling model is used as input data for resource leveling and cash flow calculations, which are explained in the next two consecutive sections of this chapter.

6.2 Resource Leveling Model

There are various resource leveling metrics used as optimization objectives to minimize fluctuations in resource profile. El-Rayes and Jun (2009) proposed two alternative metrics for resource leveling which are specific to construction projects. Therefore, this study adopts these resource leveling metrics in order to improve the resource profile in terms of fluctuations and idle times. In this section, the use and calculation of these metrics are explained in detail.

A construction activity often consumes resources during execution. Once an activity is scheduled, the resources related to that activity are assigned to the corresponding time intervals. In other words, a project activity is accompanied by resources allocated to that activity. The resource usage profile, typically of a histogram chart format, can be obtained from a project schedule with assigned resources. When activities are scheduled with the use of CPM technique, one can observe inevitable fluctuations between certain time periods, mainly due to different amounts of resource demands and overlapping activities. These fluctuations are undesirable because a contractor has to either release the excess resources during low-demand periods and rehire them during high-demand periods, or keep the idle resources on site until they are needed (El-Rayes & Jun, 2009). Using resource leveling techniques, a contractor aims to minimize these undesirable fluctuations since they cause not only a decrease in productivity but also an increase in project cost. To this end, El-Rayes and Jun (2009) developed two metrics; namely release and re-hire (RRH) and resource idle day (RID), which enable contractors to measure resource fluctuations and take actions accordingly.

6.2.1 Release and Re-Hire (RRH)

RRH gives the total amount of resources that need to be temporarily released during low-demand periods and rehired during high-demand periods (El-Rayes & Jun, 2009). The value of RRH depends on total increases in daily resource demand (*H*) and maximum resource demand (*MRD*).

The maximum resource demand (*MRD*), as expressed in the following equation, represents the maximum amount of resources throughout the entire project duration, T.

$$MRD = Max(r_1, r_2, \dots, r_T) \tag{6.8}$$

The total increases in daily resource demand (*H*) is half of the total daily resource fluctuations, represented by *HR*. The formulation of *HR* is given in Equation 6.9.

$$HR = \left[r_1 + \sum_{t=1}^{T-1} | r_t - r_{t+1} | + r_T \right] \tag{6.9}$$

where; r_t and r_{t+1} are the amount of resources needed on day t and $(t+1)$, respectively.

After finding the values of *MRD* and *HR*, the amount of released and rehired resources can be determined by subtracting *MRD* from *H* as shown in the equation below:

$$RRH = H - MRD = \frac{1}{2} \times HR - MRD \tag{6.10}$$

El-Rayes and Jun (2009) pointed out that the usage of RRH metric is reasonable for projects where release and rehire of resources are allowed.

6.2.2 Resource Idle Days (RID)

In projects where releasing and rehiring of resources are restricted, contractors generally have to keep the excess resources idle on site during low-demand periods. Therefore, RID metric is developed to quantify the total number of days in which resources remain idle and nonproductive due to undesirable resource fluctuations (El-Rayes & Jun, 2009).

Idle resources appear on day *t* when the resource demand on that day lowers to a level which is less than the peak demand levels occurred before and after day *t*. Thus, the number of idle resources on day *t* is calculated by subtracting its resource demand level from the minimum of the peak demands that exist before and after that day.

The formulation of total RID calculation is given in Equation 6.11.

$$RID = \sum_{t=1}^{T} \{ Min[Max(r_1, r_2, \ldots, r_t), Max(r_t, r_{t+1}, \ldots, r_T)] - r_t \} \tag{6.11}$$

where; r_t represents the resource demand on day t; and T denotes the total project duration.

The resource leveling model uses the following output information of scheduling model as input parameters:

- Start and finish time of each activity,
- Resource demand for the selected mode of each activity,
- Project duration.

The step-wise procedure given below is followed to reach RRH and RID values as output:

i. Assign resources between start and end times of each activity.
ii. Obtain resource demand on each day by adding amount of resources planned to be consumed by each activity on that day. This gives the resource usage profile of a given project.
iii. Calculate release-and-rehire (*RRH*) using Equations 6.8, 6.9, and 6.10.
iv. Calculate resource idle day (*RID*) using Equation 6.11.

In this study, "*RID*" is chosen to be used as one of the optimization parameters of the proposed integrated method as it is independent of releasing and rehiring policy of the companies.

6.3 Cash Flow Model

The cash flow model receives execution mode, start and end time of project activities, and project duration as input parameters from the scheduling model to calculate the parameters associated with project expenses. In addition to these activity-related information, cash flow analysis requires project-related information which encompasses financial data and contract provisions to develop a thorough cash flow profile for a given project, including:

- Interest rate per payment period,
- Overhead costs percentage,
- Mobilization costs percentage,

- Tax percentage,
- Markup percentage,
- Bond premium percentage,
- Advance payment percentage,
- Retention percentage,
- Lag to pay retained money after last payment,
- Submission period of pay requests,
- Lag to pay payment requests.

The preliminary step for cash flow analysis is the calculation of direct and indirect costs. As the name implies, the former refers to the expenses that can directly be assigned to project activities such as labor, material, equipment/machinery and subcontractor costs, if there exists any subcontracted work. On the contrary, indirect costs are usually a function of project duration, and account for expenses incurred during the life-cycle of a project, e.g., expenses pertaining to overhead, administrative and management, project manager's salary, etc. Mostly, the total project cost refers to the sum of total direct and indirect expenses as formulated in Equation 6.12.

$$Total\ cost = Direct\ cost + Overhead\ expense \tag{6.12}$$

where;

$$Overhead\ expense = Overhead\ percentage \times Direct\ cost \tag{6.13}$$

When the cash flow model introduced by Au and Hendrickson (1986) is referred (Chapter 4), it can be noticed that the total expenditures consist of mobilization and tax expenses in addition to direct and indirect costs. Mobilization cost accounts for the amount of money spent to prepare the site to commence construction such as project equipment, machinery, facilities, and personnel needed to start and continue with the early phases of the project. The mobilization cost (C_M) can be calculated as a percentage (MP) of total project cost (TC):

$$C_M = MP \times TC \tag{6.14}$$

Furthermore, the contractor pays for taxes (C_T) which is computed by the equation below:

$$C_T = TR \times (TC + C_M) \tag{6.15}$$

where; TR represents the tax rate (%).

The sum of direct cost, overhead expenses and tax represents the periodic expenditures (E_t) defined in Chapter 4.

After including tax, markup (C_{MU}) is added as a percentage of project cost including mobilization and tax (Equation 6.16).

$$C_{MU} = MU \times (TC + C_M + C_T) \tag{6.16}$$

where; MU is the markup percentage (%).

The contractor also pays the bond premium expressed as a percentage for the bond. The cost for bond (C_B) is calculated by the use of Equation 6.17.

$$C_B = BP \times (TC + C_M + C_T + C_{MU}) \tag{6.17}$$

where; BP is the bond premium percentage (%).

The total cost, mobilization cost, tax expense, markup and bond premium constitute the contract amount (Equation 6.18), which determines the income of contractor.

$$Contract\ amount = TC + C_M + C_T + C_{MU} + C_B \tag{6.18}$$

The contractor receives money from the owner in accordance with the work progress reported in progress payment documents. The procedure is as follows: (1) contractor should submit progress payment requests regularly at due times stated in the contract, (2) owner approves the requested payment based on the word accomplished in compliance with the specifications, (3) contractor receives the approved amount of money with a certain deduction, which accounts for advance payment and retainage, one payment period after submission of payment request, in general.

Since construction projects involve a large number of works, it is often encountered that some details of the project remain not completed towards the end of the project. Therefore, it is a common practice in construction projects that a predefined percentage of money is retained by the owner to compel the contractor to complete unfinished works. This retainage, sometimes referred to as retention, is calculated as a percentage of contract amount as shown in Equation 6.19.

$$Retention = RP \times Contract\ amount \tag{6.19}$$

where; RP is the retention percentage (%) defined in the contract.

In a typical construction project, contractors engage in a significant amount of payments during early stages. Therefore, another common practice is advance payment which represents the amount of money paid to the contractor by the owner at the very beginning of the project to compensate for the early payments. This advance payment (Equation 6.20), which is a percentage of the contract amount, is paid back to the owner as a deduction from each progress payment amount until the end of the project duration.

$$Advance\ payment = AP \times Contract\ amount \tag{6.20}$$

With the definitions of above parameters in hand, the cash flow model presented by Au and Hendrickson (1986) can be implemented to establish the cash flow profile from the contractor's perspective.

The following steps summarize the process of obtaining cash flow components:

i. Compute total direct cost which is the sum of direct cost values of each activity.
ii. Calculate overhead expenses by using Equation 6.13.
iii. Calculate mobilization expense by using Equation 6.14.
iv. Obtain contract amount with the use of Equation 6.18.
v. Calculate retention by using Equation 6.19.
vi. Calculate advance payment by using Equation 6.20.
vii. Assign direct cost values between the start and end times of each activity

viii. Calculate cash flow parameters as defined in Chapter 4.

The last value of cumulative net balance represents the profit, and the minimum of cumulative balance including accumulated financing costs is defined as the finance expense within the context of this study. The proposed integrated model seeks to optimize these two dependent variables together with resource leveling metric defined in Section 6.2.

6.4 Integrated Resource Leveling and Cash Flow Optimization

As discussed earlier, resource leveling and cash flow optimization are of great importance for project success as both resources and cash flow form the skeleton of a construction project. While the resource leveling techniques attempt to shift non-critical activities within their available floats, the cash flow optimization aims to bring forward costly activities in a way that a contractor would be able to receive the corresponding amount of money earlier. In addition, it enables improving the cash flow by reducing the negative values so that the contractor does not suffer from significant financial damages. To strike a balance between these conflicting goals, this study proposes an integrated resource leveling and cash flow optimization method.

6.4.1 Objective Function

The proposed integrated method aims to optimize the following objectives simultaneously:

- Maximization of profit (obtained from the cash flow model),
- Minimization of finance expense (obtained from the cash flow model),
- Minimization of resource idle day (obtained from the resource leveling model).

Recognizing the fact that cash flow-related parameters are monetary values, whereas resource idle day is sole a numeric value without unit, a cost parameter can be attributed to resource idle days. This conversion allows expressing the resource idle days as a monetary value. Accordingly, the combined objective function, named as "*combined resource and cash flow (CRC)*" function is formulated below:

$$\text{maximize}\ CRC = Profit - Finance\ Expense - Resource\ Idle\ Day\ Cost \tag{6.21}$$

This combined objective representation eliminates the need for subjectively assigning weights to dissimilar objectives. Furthermore, this representation significantly reduces the computational effort devoted to solving the multi-objective optimization problem.

The resource idle day cost in Equation 6.22 is calculated by multiplying the resource idle day (RID) by unit RID cost (Equation 6.25).

$$Resource\ Idle\ Day\ Cost = RID \times Unit\ RID\ cost \tag{6.22}$$

The unit RID cost is computed based on the average unit cost of project activities, denoted by *i*. The calculations are shown in Equations 6.23, 6.24, and 6.25.

$$Unit\ cost_i = \frac{Direct\ cost_i}{Duration_i \times Resource\ amount_i} \tag{6.23}$$

$$Average\ unit\ cost = \frac{\sum_{i=1}^{N} Unit\ cost_i}{N} \tag{6.24}$$

$$Unit\ RID\ cost = \frac{Average\ unit\ cost}{Reduction\ coefficient} \tag{6.25}$$

where; *N* is the number of activities in a given project.

6.4.2 Decision Variables

A candidate solution to the optimization problem can be expressed as a vector with the decision variables representing (1) selected execution mode; and (2) selected start time of a set of project activities, as follows.

$$\vec{X} = [X_{i,1}, X_{i,2}, \dots, X_{i,N},\ \ X_{i,N+1}, X_{i,N+2}, \dots, X_{i,2N}] \tag{6.26}$$

where; *i* denotes the i^{th} individual in the population; and *N* is the number of activities in the project. The first and second sets of *N* decision variables represent selected execution modes and selected start times, respectively.

6.4.3 Constraints

The precedence relationships between activities should be satisfied to maintain network logic.

The mathematical formulation of the optimization problem is given below:

$$\text{maximize } CRC(\vec{X}) \tag{6.27}$$

subject to

$$s_j \geq s_k + d_k, \quad \forall k \in P_j \quad and \quad \forall j \in \{1,2,\dots,N\} \tag{6.28}$$

$$s_0 = 0 \tag{6.29}$$

where; $CRC(\vec{X})$ = combined resource and cash flow as a function of a vector of candidate solution; s_j = start time of activity *j*; s_k = start time of activity *k*; d_k = duration of activity *k*; P_j = set of immediate predecessors of activity *j*.

The summary of proposed method that incorporates the scheduling, resource leveling and cash flow models is shown in Figure 6.2.

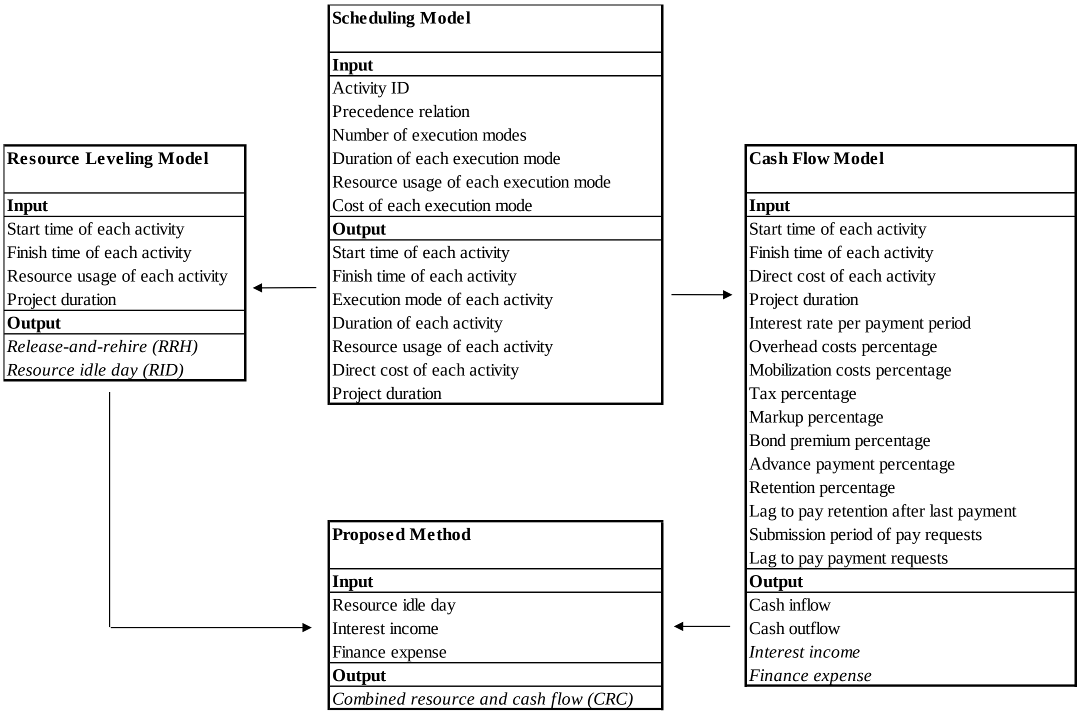

Figure 6.2. Summary of the Proposed Method

6.5 Implementation of SOS Algorithm

To solve the integrated resource leveling and cash flow optimization problem, the SOS algorithm is employed. The details of SOS algorithm together with descriptions of the main phases are given in Chapter 5. In this section, a concise summary of algorithm steps is provided to give an understanding of the application of the algorithm on the optimization problem.

Step 0: Definition of algorithm parameters

The SOS algorithm necessitates setting values for termination criterion, either maximum number of iterations (G_{max}) or maximum number of fitness evaluations (F_E).

Step 1: Ecosystem initialization

Firstly, the number of organisms in the ecosystem (*ecosize*) is set prior to population initialization. Then, a group of organisms is generated randomly in the search space, each of which is a member of the initial ecosystem.

Step 2: Identification of the best organism

The best organism, X_{best}, is determined as having the minimum fitness value for minimization problems or the maximum fitness value for maximization problems.

After this step, each and every organism *i* undergoes the following consecutive phases of the algorithm: (1) mutualism phase, (2) commensalism phase, and (3) parasitism phase. In other words, the algorithm phases start with the first organism in the ecosystem (*i* = 0), and proceeds until *i* reaches *ecosize*.

Step 3: Mutualism phase

An organism X_j ($X_j \neq X_i$) is selected from the ecosystem, randomly, to interact with the current organism X_i. In this phase, both organisms attempt to increase their survival advantage in the ecosystem which is defined by Equations 5.2 and 5.3.

In this step, both organisms have a chance to benefit from its partner, either partially or fully. The BF_1 and BF_2 in Equations 5.2 and 5.3 represent the level of benefit gained from mutualistic interaction. The benefit factors are determined randomly from {1, 2}; the value of 1 indicates partial benefit, whereas the value of 2 means the mutualistic interaction involves full benefit at least for one of the organisms.

Since the decision variables of the optimization problem lie within a range of [0, 1], the minimum and maximum values of interaction equations can be calculated. It can be recognized that the output of interaction equations might fall out of the selection range [0, 1] for execution mode and start time selection. Moreover, depending on the benefit factors, the minimum and maximum values may vary. This relationship is shown in the following expressions.

$$X_{inew} \in [-1, 2], if\ BF_1 = 1 \quad \& \quad X_{jnew} \in [-1, 2], if\ BF_2 = 1 \tag{6.30}$$

$$X_{inew} \in [-2, 2], if\ BF_1 = 2 \quad \& \quad X_{jnew} \in [-2, 2], if\ BF_2 = 2 \tag{6.31}$$

The output values of mutualistic relation functions are normalized with the use of Equation 6.32. The minimum value of interaction function is taken as –1 if benefit factor is 1, and –2 if benefit factor is 2.

$$X_{normalized} = \frac{X - \min(X)}{\max(X) - \min(X)} \tag{6.32}$$

where; X represents the decision variable of the optimization problem.

The normalized values are used for execution mode and start time selection.

The updated organisms (X_{inew} and X_{jnew}) replace the position of original organisms (X_i and X_j) if their fitness values are better than pre-mutualism values. The fitness value refers to the result of objective function obtained by following the procedures explained in the preceding sections.

Step 4: Commensalism phase

Similar to the mutualism phase, an organism X_j ($X_j \neq X_i$) is selected from the ecosystem randomly in order to devise a commensal relationship. In this phase, the current organism has a chance to improve its fitness value, i.e., adaptation to the ecosystem, whereas organism X_j remains unaffected. This commensal relationship is expressed mathematically in Equation 5.5.

The output of interaction function (Equation 5.5) ranges between –1 and 2, similar to that of mutualistic relationship with partial benefit. Therefore, the same approach is utilized for bounding the values within the range [0, 1] with normalization function given in Equation 6.32.

At the end of this phase, only fitness value of one organism (X_i) can be enhanced due to the nature of commensal relationship described above.

Step 5: Parasitism phase

This phase involves creation of a parasite vector, $X_{parasite}$, by duplicating organism X_i and modifying randomly selected dimensions. This parasite organism contends for the position of a randomly selected organism X_j.

The procedure for creation of the parasite organism is as follows. Initially the organism X_i is replicated. Thus, the parasite vector can initially be expressed as a set of decision variables:

$$Initial\ parasite\ vector = [X_{p,1}, X_{p,2}, \dots, X_{p,N},\ X_{p,N+1}, X_{p,N+2}, \dots, X_{p,2N}] \qquad (6.33)$$

where; N is the number of activities of a given project.

After replication, randomly selected dimensions of $X_{parasite}$ are modified according to the rule presented in Equation 6.34. The following rule requires generation of a random number (rnd_n) between 0 and 1 for each activity $n = \{1,2,...,N\}$ prior to application.

$$X_{p,k} = \begin{cases} X_{p,n}\ and\ X_{p,N+n} & if\ rnd_n \leq 0.5 \\ X_{p,k} & otherwise \end{cases} \qquad (6.34)$$

where; $k = \{1,2,...,2N\}$ denotes the k^{th} decision variable of parasite organism; $X_{p,n}$ and $X_{p,N+n}$ represent the newly generated random numbers within the range of [0, 1]; and $X_{p,k}$ is the decision variable of initial parasite vector located in k^{th} dimension. In short, the modification rule implies that if the random number generated for an activity n (rnd_n) is less than or equal to 0.5, a new random number replaces position k in parasite vector; otherwise, the k^{th} decision variable remains the same as the one in vector X_i. The final parasite organism has the following vector form.

$$X_{parasite} = [X_{p,1}, X_{p,2}, \dots, X_{p,N},\ X_{p,N+1}, X_{p,N+2}, \dots, X_{p,2N}] \qquad (6.35)$$

The solutions of X_j and $X_{parasite}$ are evaluated based on their fitness values. If $X_{parasite}$ has better fitness value, it replaces the position of X_j in the ecosystem. Otherwise, X_j preserves its position and eliminates $X_{parasite}$.

The algorithm starts its phases with the first organism in the ecosystem ($X_{i=0}$). After completing the parasitism phase for $X_{i=0}$, it proceeds with Step 2 to identify the best organism in the updated ecosystem, and with Steps 3 to 5 for each organism consecutively until all organisms in the ecosystem go through each phase.

The end of above procedure designates completion of one iteration. The algorithm continues to search for the optimum solution until a predefined termination criterion is met. In other words, the optimization process terminates when the stopping condition is reached. In this study, the maximum number of iterations, G_{max}, is used as the termination parameter.

The end output of the algorithm is the solution that gives best fitness value, i.e., objective function value, corresponding to the best organism (X_{best}) of the final ecosystem.

CHAPTER 7

RESULTS AND PERFORMANCE COMPARISON

This chapter consists of a brief description of the existing solution method, the example problem used to demonstrate the application of the proposed approach, the solutions obtained by implementing the proposed method, and finally, comparison of results with the existing method.

7.1 Existing Method

The optimization problem studied within the scope of this study involves minimization of resource fluctuations, minimization of financing cost, and maximization of cumulative net balance at the end of the project. The existing solution method developed by Elazouni and Abido (2014) mainly includes the implementation of Genetic Algorithm (GA), Strength Pareto Evolutionary Algorithm (SPEA), and fuzzy approach to generate solutions for this optimization problem.

The problem was formulated as a minimization problem under precedence constraints for project activities. The optimization parameters are resource idle days (RID) which is introduced by El-Rayes and Jun (2009), financing cost and anticipated profit which are defined by Au and Hendrickson (1986). The decision variables are expressed as a vector of execution modes and start times of each activity.

Since the problem consists of optimization of more than one objective, Elazouni and Abido (2014) suggested obtaining non-dominated solutions employing the concept of dominance (Zitzler & Thiele, 1998) and SPEA which gives a Pareto front at the end of optimization process. These solutions represent the indifference of a decision-maker among different objectives, i.e., no solution is better than the other with respect to any other objective from the point of view of the decision-maker. Elazouni and Abido (2014) used fuzzy approach introduced by Dhillon et al. (1993), which reflects the imprecise nature of the decision maker's judgment, with the aim of assisting decision-makers in selecting the best compromise solution among non-dominated solutions. This approach necessitates defining membership functions (μ_i) for each objective G_i as follows:

$$\mu_i = \begin{cases} 1 & if\ G_i \le G_i^{min} \\ \dfrac{G_i^{max} - G_i}{G_i^{max} - G_i^{min}} & if\ G_i^{min} < G_i < G_i^{max} \\ 0 & if\ G_i \ge G_i^{max} \end{cases} \tag{7.1}$$

where; G_i^{min} and G_i^{max} represent the minimum and maximum value of i^{th} objective among non-dominated solutions, respectively.

The value of membership function shows how much a non-dominated solution *k* satisfies objective G_i. The sum of the membership function values for all objectives gives the "accomplishment" of each solution in satisfying the objectives. The accomplishment of a non-dominated solution can be evaluated relative to all non-dominated solutions by normalizing its value over the sum of the accomplishments of all non-dominated solutions, which is given by Equation 7.2.

$$\mu^k = \frac{\sum_{i=1}^{N_{obj}} \mu_i^k}{\sum_{k=1}^{M} \sum_{i=1}^{N_{obj}} \mu_i^k} \tag{7.2}$$

where; *M* is the number of non-dominated solutions; N_{obj} is the number of objective functions.

The solution with the maximum value of μ^k is the best compromise solution.

In summary, the existing method combines the use of GA, SPEA, and fuzzy approach to produce solutions that give minimum RID, minimum financing cost, maximum profit including accumulated financing cost, and best compromise solution. Readers are referred to the study conducted by Elazouni and Abido (2014) for the details on the solution approach and the results.

7.2 Example Problem

Elazouni and Abido (2014) used a case study of a 9-activity project to demonstrate the application of their solution method. Although the original version introduced by Leu and Yang (1999) includes three resource types for each activity, Elazouni and Abido (2014) consider only the first resource type for application purposes. The example project includes 9 activities, some of which have different execution modes accompanied by different time, cost, and resource utilization alternatives. The activity information consisting of duration, direct cost and resource demand of each execution mode along with predecessor(s) are given in Table 7.1.

Table 7.1. Activity Information of the Example Project (adapted from Leu and Yang (1999))

Activity	**Predecessor(s)**	**Mode**	**Duration (days)**	**Direct cost ($)**	**Resource (crew/day)**
A	-	1	5	480	5
		2	6	300	3
B	A	1	9	450	4
C	B, D	1	12	850	4
		2	13	600	3
D	A	1	15	420	5
E	D, F	1	12	1,860	1
		2	13	1,450	1
		3	14	1,050	1
F	A	1	16	3,860	6
		2	17	3,220	5
		3	18	2,600	4
		4	19	2,000	3
G	F	1	13	1,900	3
		2	14	1,200	3
H	C, E	1	7	950	6
		2	8	640	6
I	E, G, H	1	9	560	5

In addition to activity-related information given above, project-related information including financial data and contract provisions to perform cash flow analysis are provided in Table 7.2.

Table 7.2. Financial Data and Contract Provisions of the Example Project (adapted from Elazouni and Abido (2014))

Category	**Item**	
Interest rate	Interest rate percentage per week	0.8
Financial data	Overheads percentage	15
	Mobilization costs percentage	10
	Tax percentage	2
	Markup percentage	10
	Bond premium percentage	1
	Advance payment percentage of total contract amount	5
	Weeks to retrieve advance payment	[a]
Contract provisions	Retained percentage of pay requests	5
	Lag to pay retained money after last payment (weeks)	0
	Weeks to submit payment requests regularly	1
	Lag to pay payment requests (weeks)	1
	Late completion penalty per day	150
	Lag to pay payment requests (weeks)	1

[a]Number of weeks encompassing the total project duration.

It is worthwhile mentioning that the problem description contains the following underlying assumptions:

- The network logic is assumed to remain unchanged.
- Activities are assumed to be time continuous. Once an activity is started, it will continue without interruption.
- The precedence relationships among activities are deterministic. Each activity cannot start until its predecessor activities have finished.

- The duration of each activity is known and fixed. Once the duration of an activity is established, no reduction or extension in activity duration is permitted.
- The resource of each activity is known and fixed. Resource usage of each activity is also assumed to remain constant throughout the duration of the activity, i.e., each activity will have a constant rate of utilization of the resource.
- The cost of an activity is known, fixed, and uniformly distributed along its duration.
- Each activity must be performed using only one of the available execution modes, and mode switching is not allowed once an activity is started.
- Payments and receipts are assumed to be delivered on time in accordance with contract provisions.

7.3 Results of the Proposed Method

The proposed optimization method was coded using C# programming language and compiled within Microsoft Visual Studio 2017 on a 64-bit platform. The analyses were performed on a desktop computer with a P9X79 Chipset motherboard, 16 GB 667 MHz DDR3 random-access memory (RAM), central processing unit (CPU) of Intel Core i7-3.40 GHz, and 64-bit Windows 10 operating system.

The results presented in this part are obtained using the information given for the example project in Section 7.2.

The optimization problem defined in this study was solved in such a way that the project schedules that yield:

(1) minimum RID,
(2) maximum profit,
(3) minimum finance expense,
(4) best compromise solution,

(5) maximum CRC,

are captured separately.

Using the solution of the optimization problem enables creation of the following output:

- Start and end times of project activities,
- Resource usage profile throughout the project duration,
- Parameters used in the cash flow analysis and cash flow profiles.

In the remaining parts of this section, the results are provided for each of the foregoing outcomes.

7.3.1 Schedule Output

The solution of optimization problem gives the selected execution modes and start times of project activities. Each execution mode is accompanied by different duration, resource and direct cost combination for a project activity. Thus, knowing the execution mode implies knowing abovementioned activity-related information, leading to a schedule output for the whole project.

Figure 7.1 shows the project schedules where start and finish times of each activity can be identified for each objective.

Once the start and finish time of an activity is known, resource usage amount associated with selected execution mode of that activity can be assigned along its duration. Completing the resource assignment of each project activity leads to establishing a complete resource usage profile for the project, as depicted in Figure 7.2.

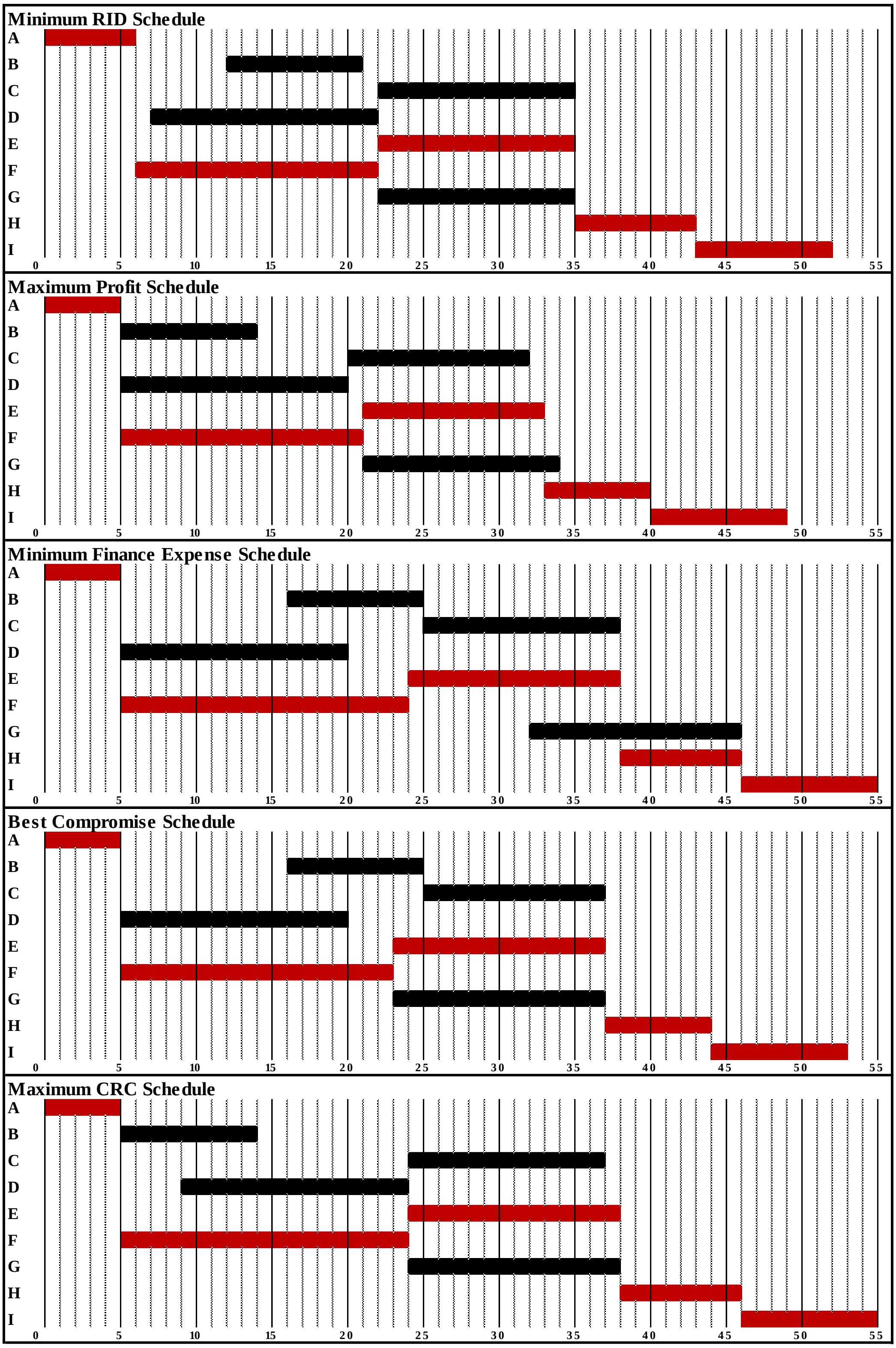

Figure 7.1. Project Schedules of Each Solution

7.3.2 Resource Usage Profile

Once the start and finish time of an activity is known, resource usage amount associated with selected execution mode of that activity can be assigned along its duration. Completing the resource assignment of each project activity leads to establishing a complete resource usage profile for the project, as depicted in Figure 7.2.

Using the daily resource usage values, the calculations of resource idle day (RID) and release-and-rehire (RRH) can be carried out in accordance with equations provided in Section 6.2.

It can be observed from Figure 7.2 that the solutions of minimum RID and best compromise objectives give the minimum value of RID, which is zero. The minimum finance expense solution has the maximum value of RID, which is equal to 59. Therefore, it can be deduced that if a contractor seeks to minimize project's finance expense, he/she will end up with having a considerable amount of resource fluctuation in resource usage histogram. Moreover, a slight difference in maximum resource demand (MRD) values among all solutions can be noticed.

The solution with maximum CRC outputs the RID value as 2, which is very close to the minimum value of RID. Thus, it can be concluded that the use of CRC as an optimization parameter could be preferred against best compromise solution as it requires less computational effort to reach the solution.

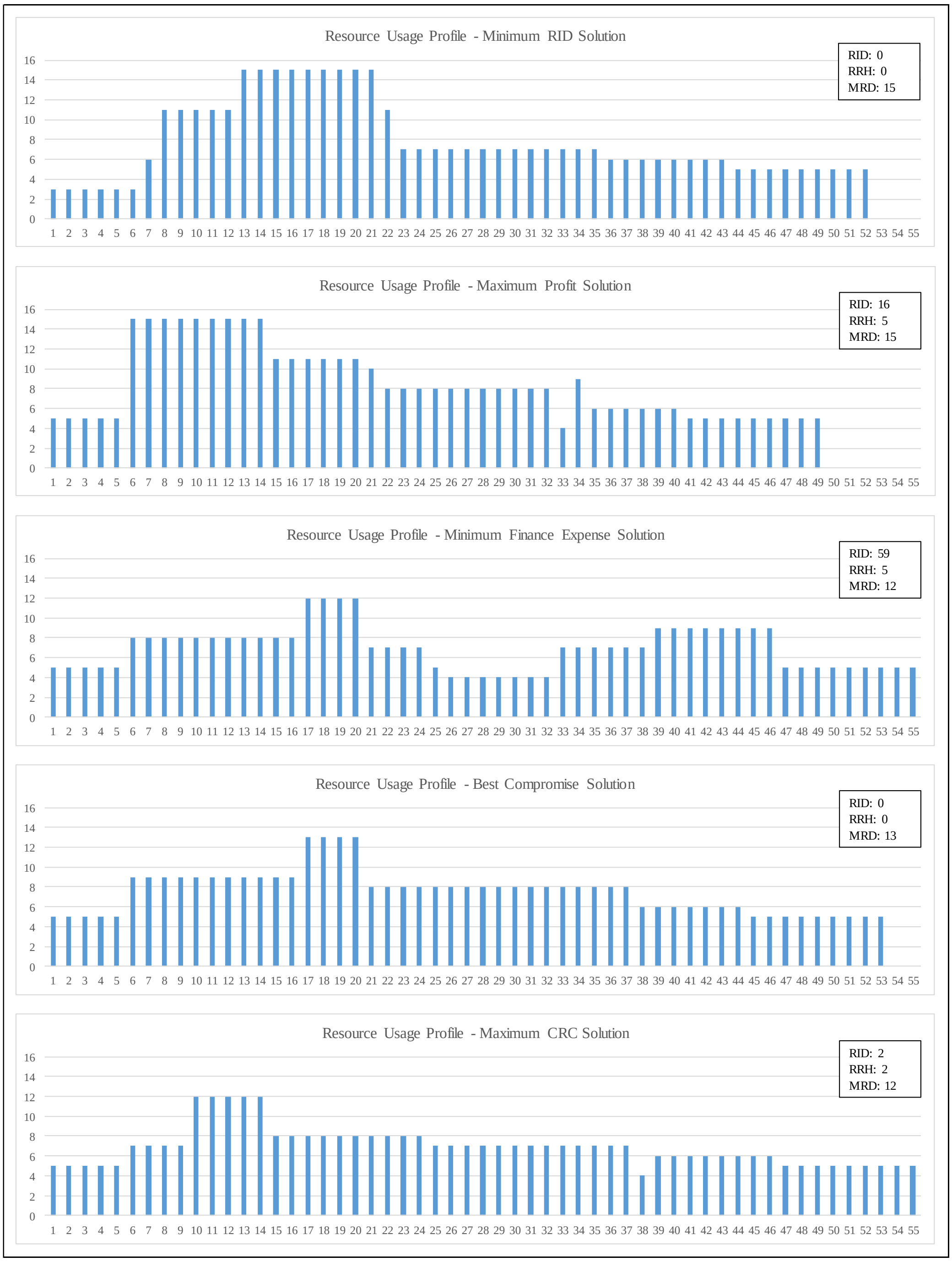

Figure 7.2. Resource Usage Profiles of Each Solution

7.3.3 Cash Flow Analysis

The first step to carry out cash flow analysis is to determine direct expenses incurred during execution of project activities since the whole analysis depends on the total direct cost of the project together with financial data and contract provisions. Similar to activity durations and resource usages, direct costs are known once the execution modes are decided.

After computing total direct cost, the contract amount can be found by applying the procedure described in Section 6.3. Table 7.3 shows the components of contract amount calculated for each solution.

Table 7.3. Bid Analysis

	Minimum RID	**Maximum profit**	**Minimum finance**	**Best compromise**	**Maximum CRC**
Total direct cost ($)	10,180.0	11,330.0	7,400.0	8,560.0	7,400.0
Overhead expense ($)	1,527.0	1,699.5	1,110.0	1,284.0	1,110.0
Subtotal *(direct cost + overhead)*	11,707.0	13,029.5	8,510.0	9,844.0	8,510.0
Mobilization expense ($)	1,170.7	1,303.0	851.0	984.4	851.0
Subtotal *(direct cost + overhead + mobilization)*	12,877.7	14,332.5	9,361.0	10,828.4	9,361.0
Tax ($)	257.6	286.6	187.2	216.6	187.2
Subtotal *(direct cost + overhead + mobilization + tax)*	13,135.3	14,619.1	9,548.2	11,045.0	9,548.2
Markup ($)	1,313.5	1,461.9	954.8	1,104.5	954.8
Subtotal *(direct cost + overhead + mobilization + tax + markup)*	14,448.8	16,081.0	10,503.0	12,149.5	10,503.0
Bond premium ($)	144.5	160.8	105.0	121.5	105.0
Contract amount ($)	14,593.3	16,241.8	10,608.1	12,271.0	10,608.1
Bid factor	1.434	1.434	1.434	1.434	1.434

The contractor receives the monetary equivalence of the accomplished work based on the contract amount. Thus, the value of cash inflow for a given period is dependent on the earned value of the completed percentage of work appurtenant to the previous period. At the very beginning of the project, the sole income of the contractor is the advance payment paid by the owner as per contract agreement. In the remaining periods until project completion, the owner deducts a certain amount of money from earned value to account for retention and reimbursement of advance payment. Therefore, the receipt amount equals the multiplication of direct expenses incurred within previous period by the bid factor (Equation 7.3), i.e., earned value (Equation 7.4), with a deduction (Equation 7.5) due to retained money and advance payment, except the last period. This received money refers to the net cash inflow (Equation 7.6) of the contractor. Upon approval of the project completion by the owner, the contractor is able to receive the retained money at the end of the project. Accordingly, the cash inflow corresponding to the last period is equal to the sum of total retained amount and earned value with deductions.

$$Bid\ factor = \frac{Contract\ amount}{Total\ direct\ cost} \tag{7.3}$$

$$Earned\ value = Bid\ factor\ \times\ Direct\ expense \tag{7.4}$$

$$Deductions = Earned\ value\ \times Retention\ \% + \frac{Advance\ payment}{Number\ of\ weeks} \tag{7.5}$$

$$Net\ cash\ inflow = Earned\ value - Deductions \tag{7.6}$$

The cash outflow refers to the expenses incurred during execution of a project. In general, the direct cost of project activities is the main contributor to the cash outflow. In addition to direct expenses, disbursements due to overhead and taxes constitute cash outflow as well. Consequently, the value of total cash outflow (Equation 7.7) for a given period is equal to the sum of direct cost, overhead expense, and tax, except commencement of the project. The initial cost consists of mobilization and bond expenses, which comprise the first cash outflow of the contractor.

$$Total\ cash\ outflow = Direct\ expense +\ Overhead\ expense + Tax \tag{7.7}$$

The weekly cash outflow and inflow values are provided in Tables 7.4, 7.6, 7.7, 7.8, and 7.9 for the solutions of minimum RID, maximum profit, minimum finance expense, best compromise, and maximum CRC, respectively.

The total cash outflow and net cash inflow values described above refer to the "*expenditures*" and "*income*" defined in Chapter 4, respectively. In the remaining parts of this section, the cash flow parameters are named as per the cash flow model introduced by Au and Hendrickson (1986), which is presented in Chapter 4. These parameters are listed as follows.

- Expenditures, *E* (Equation 4.2),
- Income, *P* (Equation 4.3),
- Cumulative balance, *F* (Equation 4.4),
- Net cumulative balance, *N* (Equation 4.5),
- Financing costs, *I* (Equation 4.6),
- Accumulated financing costs, *I'* (Equation 4.7),
- Cumulative balance with financing costs, *F'* (Equation 4.8),
- Net cumulative balance with financing costs, *N'* (Equation 4.9).

The results of cash flow calculations for obtaining the aforementioned parameters are tabulated in Tables 7.5, 7.7, 7.9, 7.11, and 7.13 for the solutions of minimum RID, maximum profit, minimum finance expense, best compromise, and maximum CRC, respectively.

The minimum value of cumulative balance including accumulated financing costs (*F'*) and the net balance including accumulated financing cost (*N'*) at the end of the project represent optimization parameters pertinent to cash flow analysis, named as "*finance expense*" and interest "*income*", respectively.

Lastly, with the use of related cash flow parameters, the cash flow profile can be constructed as shown in Figures 7.3, 7.4, 7.5, 7.6, and 7.7 for the solutions of minimum RID, maximum profit, minimum finance expense, best compromise, and maximum CRC, respectively.

Table 7.4. Weekly Cash Outflow and Inflow (Minimum RID Solution)

Week	Outflow	Amount ($)	Inflow	Amount ($)
0	Mob. + Bond	1,315.2	Adv. Pay	729.7
1	Direct	250.0	Earned value	-
	Overhead + Tax	171.6	Deductions	-
	Total	421.6	Net	-
2	Direct	1,099.0	Earned value	358.4
	Overhead + Tax	171.6	Deductions	84.3
	Total	1,270.6	Net	274.1
3	Direct	1,496.3	Earned value	1,575.4
	Overhead + Tax	171.6	Deductions	145.1
	Total	1,667.8	Net	1,430.3
4	Direct	1,596.3	Earned value	2,144.9
	Overhead + Tax	171.6	Deductions	173.6
	Total	1,767.8	Net	1,971.3
5	Direct	1,500.0	Earned value	2,288.3
	Overhead + Tax	171.6	Deductions	180.7
	Total	1,671.6	Net	2,107.5
6	Direct	1,519.2	Earned value	2,150.3
	Overhead + Tax	171.6	Deductions	173.9
	Total	1,690.8	Net	1,976.5
7	Direct	1,519.2	Earned value	2,177.9
	Overhead + Tax	171.6	Deductions	175.2
	Total	1,690.8	Net	2,002.6
8	Direct	400.0	Earned value	2,177.9
	Overhead + Tax	171.6	Deductions	175.2
	Total	571.6	Net	2,002.6
9	Direct	364.4	Earned value	573.4
	Overhead + Tax	171.6	Deductions	95.0
	Total	536.0	Net	478.4
10	Direct	311.1	Earned value	522.4
	Overhead + Tax	171.6	Deductions	92.5
	Total	482.7	Net	430.0
11	Direct	124.4	Earned value	446.0
	Overhead + Tax	68.6	Deductions	88.6
	Total	193.1	Net	357.4
12	Direct	-	Earned value	178.4
	Overhead + Tax	-	Deductions	75.3
	Total	-	Additions	729.7
		-	Net	832.8

Table 7.5. Cash Flow Parameters (Minimum RID Solution)

Week	E ($)	P ($)	F ($)	N ($)	I ($)	I' ($)	F' ($)	N' ($)
0	-1,315.2	729.7	-1,315.2	-585.5	0.0	0.0	-1,315.2	-585.5
1	-421.6	0.0	-1,007.1	-1,007.1	-6.4	-6.4	-1,013.5	-1,013.5
2	-1,270.6	274.1	-2,277.7	-2,003.6	-13.1	-19.6	-2,297.3	-2,023.1
3	-1,667.8	1,430.3	-3,671.4	-2,241.1	-22.7	-42.4	-3,713.8	-2,283.5
4	-1,767.8	1,971.3	-4,008.9	-2,037.6	-25.0	-67.8	**-4,076.7***	-2,105.4
5	-1,671.6	2,107.5	-3,709.2	-1,601.7	-23.0	-91.3	-3,800.5	-1,693.0
6	-1,690.8	1,976.5	-3,292.5	-1,316.0	-19.6	-111.6	-3,404.1	-1,427.6
7	-1,690.8	2,002.6	-3,006.9	-1,004.2	-17.3	-129.8	-3,136.6	-1,134.0
8	-571.6	2,002.6	-1,575.8	426.8	-10.3	-141.1	-1,717.0	285.7
9	-536.0	478.4	-109.2	369.2	0.0	-142.3	-251.5	226.9
10	-482.7	430.0	-113.5	316.4	0.0	-143.4	-256.9	173.0
11	-193.1	357.4	123.4	480.7	0.0	-144.6	-21.2	336.2
12	0.0	832.8	480.7	1,313.5	0.0	-145.7	335.0	**1,167.8****

**The value of finance expense.*

***The value of net cumulative balance including accumulated financing cost.*

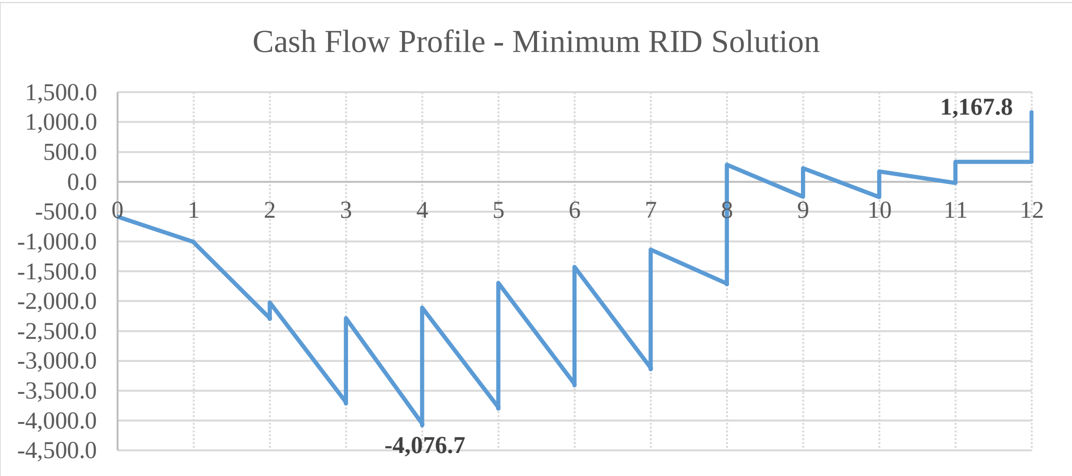

Figure 7.3. Cash Flow Profile (Minimum RID Solution)

Table 7.6. Weekly Cash Outflow and Inflow (Maximum Profit Solution)

Week	Outflow	Amount ($)	Inflow	Amount ($)
0	Mob. + Bond	1,463.8	Adv. Pay	812.1
1	Direct	480.0	Earned value	-
	Overhead + Tax	202.7	Deductions	-
	Total	682.7	Net	-
2	Direct	1,596.3	Earned value	688.1
	Overhead + Tax	202.7	Deductions	115.6
	Total	1,798.9	Net	572.5
3	Direct	1,546.3	Earned value	2,288.3
	Overhead + Tax	202.7	Deductions	195.6
	Total	1,748.9	Net	2,092.6
4	Direct	1,346.3	Earned value	2,216.6
	Overhead + Tax	202.7	Deductions	192.0
	Total	1,548.9	Net	2,024.5
5	Direct	1,800.0	Earned value	1,929.9
	Overhead + Tax	202.7	Deductions	177.7
	Total	2,002.7	Net	1,752.2
6	Direct	1,859.9	Earned value	2,580.4
	Overhead + Tax	202.7	Deductions	210.2
	Total	2,062.6	Net	2,370.2
7	Direct	1,462.7	Earned value	2,666.3
	Overhead + Tax	202.7	Deductions	214.5
	Total	1,665.4	Net	2,451.7
8	Direct	678.6	Earned value	2,096.8
	Overhead + Tax	202.7	Deductions	186.1
	Total	881.2	Net	1,910.8
9	Direct	311.1	Earned value	972.7
	Overhead + Tax	202.7	Deductions	129.8
	Total	513.8	Net	842.9
10	Direct	248.9	Earned value	446.0
	Overhead + Tax	162.1	Deductions	103.5
	Total	411.0	Net	342.5
11	Direct	-	Earned value	356.8
	Overhead + Tax	-	Deductions	99.0
	Total	-	Additions	812.1
		-	Net	1,069.8

Table 7.7. Cash Flow Parameters (Maximum Profit Solution)

Week	E ($)	P ($)	F ($)	N ($)	I ($)	I' ($)	F' ($)	N' ($)
0	-1,463.8	812.1	-1,463.8	-651.7	0.0	0.0	-1,463.8	-651.7
1	-682.7	0.0	-1,334.3	-1,334.3	-7.9	-7.9	-1,342.3	-1,342.3
2	-1,798.9	572.5	-3,133.3	-2,560.8	-17.9	-25.9	-3,159.1	-2,586.7
3	-1,748.9	2,092.6	-4,309.7	-2,217.1	-27.5	-53.6	**-4,363.3***	-2,270.6
4	-1,548.9	2,024.5	-3,766.0	-1,741.4	-23.9	-77.9	-3,843.9	-1,819.4
5	-2,002.7	1,752.2	-3,744.1	-1,992.0	-21.9	-100.5	-3,844.6	-2,092.4
6	-2,062.6	2,370.2	-4,054.6	-1,684.4	-24.2	-125.5	-4,180.0	-1,809.9
7	-1,665.4	2,451.7	-3,349.8	-898.0	-20.1	-146.6	-3,496.4	-1,044.7
8	-881.2	1,910.8	-1,779.3	131.5	-10.7	-158.5	-1,937.8	-27.0
9	-513.8	842.9	-382.3	460.6	0.0	-159.8	-542.0	300.9
10	-411.0	342.5	49.6	392.1	0.0	-161.1	-111.4	231.0
11	0.0	1,069.8	392.1	1,461.9	0.0	-162.3	229.7	**1,299.6****

**The value of finance expense.*

***The value of net cumulative balance including accumulated financing cost.*

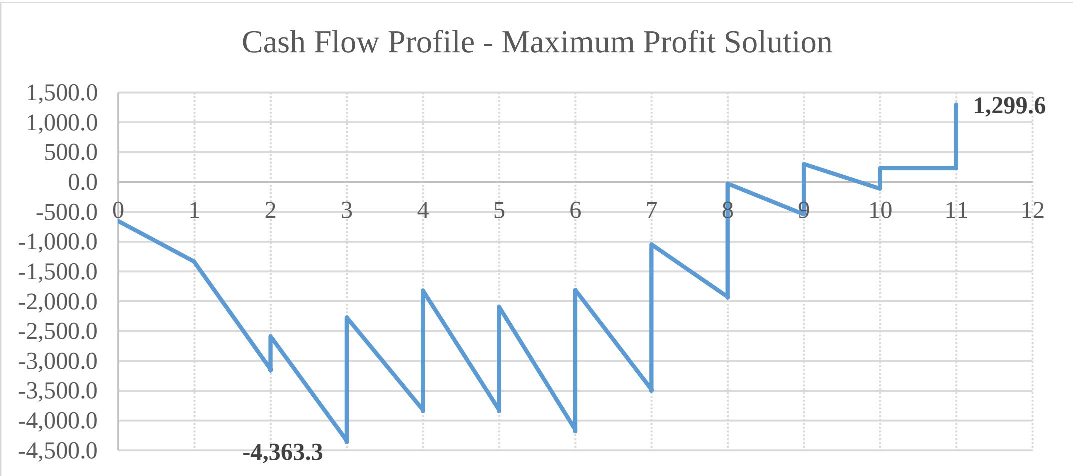

Figure 7.4. Cash Flow Profile (Maximum Profit Solution)

Table 7.8. Weekly Cash Outflow and Inflow (Minimum Finance Expense Solution)

Week	Outflow	Amount ($)	Inflow	Amount ($)
0	Mob. + Bond	956.0	Adv. Pay	530.4
1	Direct	480.0	Earned value	-
	Overhead + Tax	117.9	Deductions	-
	Total	597.9	Net	-
2	Direct	666.3	Earned value	688.1
	Overhead + Tax	117.9	Deductions	82.6
	Total	784.2	Net	605.5
3	Direct	666.3	Earned value	955.2
	Overhead + Tax	117.9	Deductions	96.0
	Total	784.2	Net	859.2
4	Direct	866.3	Earned value	955.2
	Overhead + Tax	117.9	Deductions	96.0
	Total	984.2	Net	859.2
5	Direct	746.1	Earned value	1,241.9
	Overhead + Tax	117.9	Deductions	110.3
	Total	864.0	Net	1,131.6
6	Direct	605.8	Earned value	1,069.5
	Overhead + Tax	117.9	Deductions	101.7
	Total	723.7	Net	967.8
7	Direct	862.9	Earned value	868.4
	Overhead + Tax	117.9	Deductions	91.6
	Total	980.8	Net	776.7
8	Direct	952.0	Earned value	1,237.0
	Overhead + Tax	117.9	Deductions	110.1
	Total	1,070.0	Net	1,126.9
9	Direct	828.6	Earned value	1,364.8
	Overhead + Tax	117.9	Deductions	116.5
	Total	946.5	Net	1,248.3
10	Direct	414.6	Earned value	1,187.8
	Overhead + Tax	117.9	Deductions	107.6
	Total	532.5	Net	1,080.2
11	Direct	311.1	Earned value	594.3
	Overhead + Tax	117.9	Deductions	77.9
	Total	429.0	Net	516.4
12	Direct	-	Earned value	446.0
	Overhead + Tax	-	Deductions	70.5
	Total	-	Additions	530.4
		-	Net	905.9

Table 7.9. Cash Flow Parameters (Minimum Finance Expense Solution)

Week	E ($)	P ($)	F ($)	N ($)	I ($)	I' ($)	F' ($)	N' ($)
0	-956.0	530.4	-956.0	-425.6	0.0	0.0	-956.0	-425.6
1	-597.9	0.0	-1,023.6	-1,023.6	-5.8	-5.8	-1,029.4	-1,029.4
2	-784.2	605.5	-1,807.8	-1,202.3	-11.3	-17.2	-1,825.0	-1,219.5
3	-784.2	859.2	-1,986.6	-1,127.4	-12.8	-30.1	-2,016.6	-1,157.4
4	-984.2	859.2	-2,111.6	-1,252.4	-13.0	-43.3	-2,154.9	-1,295.7
5	-864.0	1,131.6	-2,116.4	-984.8	-13.5	-57.1	**-2,173.5***	-1,041.9
6	-723.7	967.8	-1,708.5	-740.7	-10.8	-68.3	-1,776.8	-809.0
7	-980.8	776.7	-1,721.6	-944.8	-9.8	-78.7	-1,800.3	-1,023.5
8	-1,070.0	1,126.9	-2,014.8	-887.9	-11.8	-91.2	-2,106.0	-979.0
9	-946.5	1,248.3	-1,834.4	-586.1	-10.9	-102.8	-1,937.1	-688.8
10	-532.5	1,080.2	-1,118.6	-38.4	-6.8	-110.4	-1,229.0	-148.8
11	-429.0	516.4	-467.5	49.0	-2.0	-113.3	-580.8	-64.4
12	0.0	905.9	49.0	954.8	0.0	-114.2	-65.3	**840.6****

**The value of finance expense.*

***The value of net cumulative balance including accumulated financing cost.*

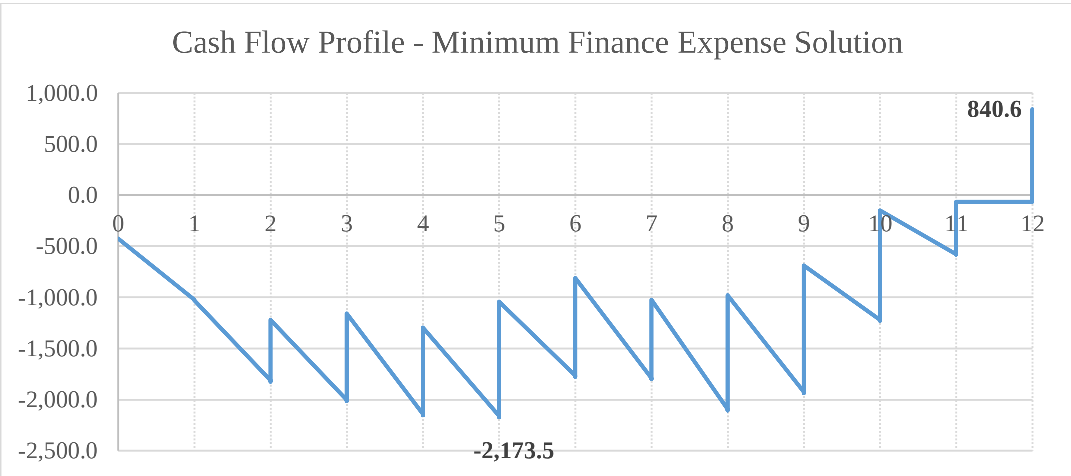

Figure 7.5. Cash Flow Profile (Minimum Finance Expense Solution)

Table 7.10. Weekly Cash Outflow and Inflow (Best Compromise Solution)

Week	Outflow	Amount ($)	Inflow	Amount ($)
0	Mob. + Bond	1,105.9	Adv. Pay	613.5
1	Direct	480.0	Earned value	-
	Overhead + Tax	141.6	Deductions	-
	Total	621.6	Net	-
2	Direct	862.2	Earned value	688.1
	Overhead + Tax	141.6	Deductions	90.2
	Total	1,003.8	Net	597.9
3	Direct	862.2	Earned value	1,236.0
	Overhead + Tax	141.6	Deductions	117.6
	Total	1,003.8	Net	1,118.4
4	Direct	1,062.2	Earned value	1,236.0
	Overhead + Tax	141.6	Deductions	117.6
	Total	1,203.8	Net	1,118.4
5	Direct	1,004.8	Earned value	1,522.7
	Overhead + Tax	141.6	Deductions	131.9
	Total	1,146.3	Net	1,390.8
6	Direct	1,157.7	Earned value	1,440.3
	Overhead + Tax	141.6	Deductions	127.8
	Total	1,299.3	Net	1,312.6
7	Direct	1,157.7	Earned value	1,659.6
	Overhead + Tax	141.6	Deductions	138.8
	Total	1,299.3	Net	1,520.9
8	Direct	870.2	Earned value	1,659.6
	Overhead + Tax	141.6	Deductions	138.8
	Total	1,011.8	Net	1,520.9
9	Direct	605.1	Earned value	1,247.5
	Overhead + Tax	141.6	Deductions	118.2
	Total	746.6	Net	1,129.4
10	Direct	311.1	Earned value	867.4
	Overhead + Tax	141.6	Deductions	99.1
	Total	452.7	Net	768.2
11	Direct	186.7	Earned value	446.0
	Overhead + Tax	84.9	Deductions	78.1
	Total	271.6	Net	367.9
12	Direct	-	Earned value	267.6
	Overhead + Tax	-	Deductions	69.2
	Total	-	Additions	613.5
		-	Net	812.0

Table 7.11. Cash Flow Parameters (Best Compromise Solution)

Week	E ($)	P ($)	F ($)	N ($)	I ($)	I' ($)	F' ($)	N' ($)
0	-1,105.9	613.5	-1,105.9	-492.3	0.0	0.0	-1,105.9	-492.3
1	-621.6	0.0	-1,113.9	-1,113.9	-6.4	-6.4	-1,120.3	-1,120.3
2	-1,003.8	597.9	-2,117.7	-1,519.8	-12.9	-19.4	-2,137.1	-1,539.2
3	-1,003.8	1,118.4	-2,523.6	-1,405.1	-16.2	-35.7	-2,559.3	-1,440.9
4	-1,203.8	1,118.4	-2,608.9	-1,490.5	-16.1	-52.1	-2,661.0	-1,542.6
5	-1,146.3	1,390.8	-2,636.8	-1,246.0	-16.5	-69.0	**-2,705.8***	-1,315.0
6	-1,299.3	1,312.6	-2,545.3	-1,232.7	-15.2	-84.7	-2,630.0	-1,317.5
7	-1,299.3	1,520.9	-2,532.0	-1,011.2	-15.1	-100.5	-2,632.5	-1,111.6
8	-1,011.8	1,520.9	-2,023.0	-502.1	-12.1	-113.4	-2,136.4	-615.5
9	-746.6	1,129.4	-1,248.7	-119.4	-7.0	-121.3	-1,370.0	-240.7
10	-452.7	768.2	-572.0	196.2	-2.8	-125.0	-697.1	71.2
11	-271.6	367.9	-75.4	292.5	0.0	-126.0	-201.4	166.5
12	0.0	812.0	292.5	1,104.5	0.0	-127.0	165.5	**977.4****

**The value of finance expense.*

***The value of net cumulative balance including accumulated financing cost.*

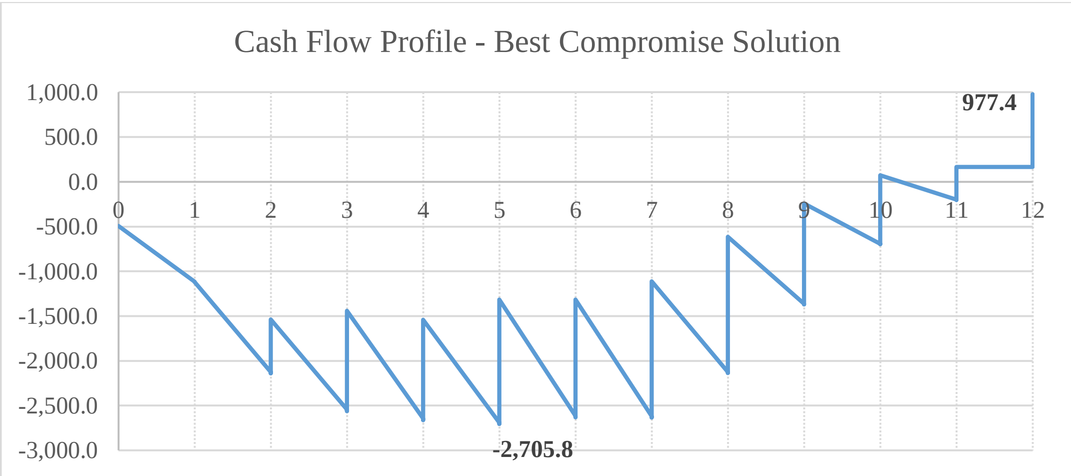

Figure 7.6. Cash Flow Profile (Best Compromise Solution)

Table 7.12. Weekly Cash Outflow and Inflow (Maximum CRC Solution)

Week	**Outflow**	**Amount ($)**	**Inflow**	**Amount ($)**
0	Mob. + Bond	956.0	Adv. Pay	530.4
1	Direct	480.0	Earned value	-
	Overhead + Tax	117.9	Deductions	-
	Total	597.9	Net	-
2	Direct	804.3	Earned value	688.1
	Overhead + Tax	117.9	Deductions	82.6
	Total	922.2	Net	605.5
3	Direct	866.3	Earned value	1,153.0
	Overhead + Tax	117.9	Deductions	105.9
	Total	984.2	Net	1,047.1
4	Direct	666.3	Earned value	1,241.9
	Overhead + Tax	117.9	Deductions	110.3
	Total	784.2	Net	1,131.6
5	Direct	739.9	Earned value	955.2
	Overhead + Tax	117.9	Deductions	96.0
	Total	857.8	Net	859.2
6	Direct	1,034.3	Earned value	1,060.7
	Overhead + Tax	117.9	Deductions	101.3
	Total	1,152.3	Net	959.4
7	Direct	1,034.3	Earned value	1,482.8
	Overhead + Tax	117.9	Deductions	122.4
	Total	1,152.3	Net	1,360.4
8	Direct	734.5	Earned value	1,482.8
	Overhead + Tax	117.9	Deductions	122.4
	Total	852.4	Net	1,360.4
9	Direct	400.0	Earned value	1,052.9
	Overhead + Tax	117.9	Deductions	100.9
	Total	517.9	Net	952.0
10	Direct	328.9	Earned value	573.4
	Overhead + Tax	117.9	Deductions	76.9
	Total	446.8	Net	496.5
11	Direct	311.1	Earned value	471.5
	Overhead + Tax	117.9	Deductions	71.8
	Total	429.0	Net	399.7
12	Direct	-	Earned value	446.0
	Overhead + Tax	-	Deductions	70.5
	Total	-	Additions	530.4
		-	Net	905.9

Table 7.13. Cash Flow Parameters (Maximum CRC Solution)

Week	**E ($)**	**P ($)**	**F ($)**	**N ($)**	**I ($)**	**I' ($)**	**F' ($)**	**N' ($)**
0	-956.0	530.4	-956.0	-425.6	0.0	0.0	-956.0	-425.6
1	-597.9	0.0	-1,023.6	-1,023.6	-5.8	-5.8	-1,029.4	-1,029.4
2	-922.2	605.5	-1,945.8	-1,340.3	-11.9	-17.7	-1,963.5	-1,358.1
3	-984.2	1,047.1	-2,324.6	-1,277.4	-14.7	-32.5	**-2,357.1***	-1,310.0
4	-784.2	1,131.6	-2,061.7	-930.1	-13.4	-46.1	-2,107.8	-976.3
5	-857.8	859.2	-1,788.0	-928.8	-10.9	-57.4	-1,845.3	-986.1
6	-1,152.3	959.4	-2,081.0	-1,121.6	-12.0	-69.9	-2,150.9	-1,191.5
7	-1,152.3	1,360.4	-2,273.9	-913.5	-13.6	-84.0	-2,357.9	-997.5
8	-852.4	1,360.4	-1,765.8	-405.5	-10.7	-95.4	-1,861.3	-500.9
9	-517.9	952.0	-923.4	28.6	-5.3	-101.5	-1,024.9	-72.9
10	-446.8	496.5	-418.2	78.3	0.0	-102.3	-520.5	-24.0
11	-429.0	399.7	-350.7	49.0	0.0	-103.1	-453.8	-54.2
12	0.0	905.9	49.0	954.8	0.0	-103.9	-55.0	**850.9****

**The value of finance expense.*

***The value of net cumulative balance including accumulated financing cost.*

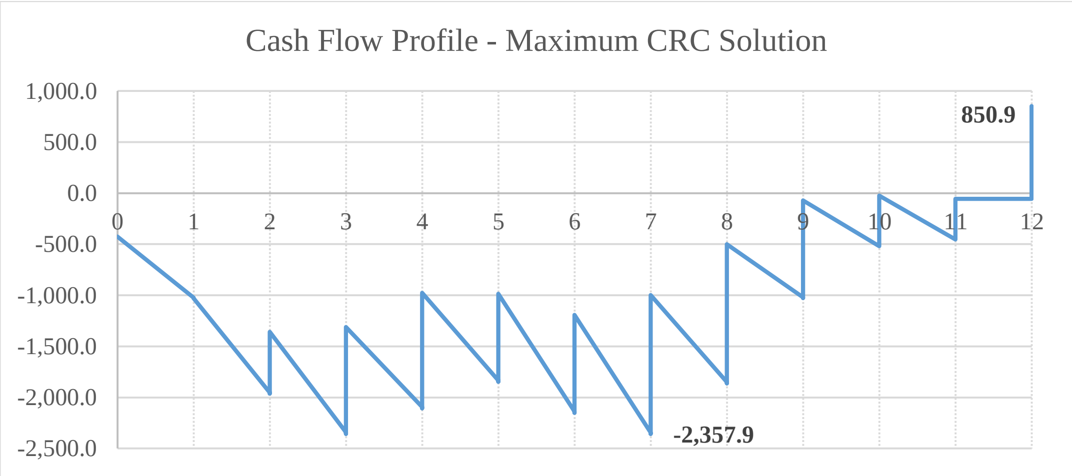

Figure 7.7. Cash Flow Profile (Maximum CRC Solution)

As can be observed from tables presenting cash flow parameters, the net balance without interest charges (*N*) at the end of the project has a lower value than the net balance when accumulated finance costs are included (*N'*). Furthermore, the results reveal that "finance expense" is inversely proportional to "profit". In other words, the solution with the minimum "finance expense" leads to maximum "profit", and vice versa.

The solution with maximum CRC gives considerably less "finance expense" than that of best compromise solution, which could promote the use of this parameter as an objective instead of best compromise solution. However, the "profit" value of maximum CRC solution is less than the value obtained from the best compromise solution.

7.4 Comparison of Results with Existing Method

The existing solution method for the optimization problem studied in this thesis study involves the use of Genetic Algorithm (GA) and Strength Pareto Evolutionary Algorithm (SPEA), whereas this thesis study implements Symbiotic Organisms Search (SOS) algorithm to obtain optimized schedules for the same problem. The algorithm parameters are given in Table 7.14. It should be noted that the maximum number of iterations (G_{max}) is taken as 30 for the best compromise solution in order to reduce the computational burden.

Table 7.14. Algorithm Parameters

Strength Pareto Evolutionary Algorithm (SPEA)		**Symbiotic Organisms Search (SOS) Algorithm**	
Parameter	**Value**	**Parameter**	**Value**
Population size	400	Number of organisms (*ecosize*)	400
Number of generations	500	Maximum number of iterations (G_{max})*	500
Crossover probability	90%		
Mutation probability	20%		

**G_{max} is taken as 30 for the best compromise solution.*

Out of 50 non-dominated solutions, Elazouni and Abido (2014) presented four remarkable solutions that give minimum "*finance*", maximum "*profit*", minimum "*RID*", and "*best compromise*" solution. Comparison of results for each objective is given in Table 7.15. The solutions of minimum "*finance*", maximum "*profit*", minimum "*RID*" and *best compromise* solution are provided as presented by Elazouni and Abido (2014) for the sake of consistency. In addition, Table 7.15 includes the solution which corresponds to maximum "*CRC*", a parameter that takes into account objectives related to resource leveling and cash flow optimization, simultaneously. It should be noted that the reduction coefficient used in the calculation of resource idle day cost (Equation 6.25) is taken as 2, i.e., the resource idle cost is considered as 50% of the average unit resource cost.

Table 7.15. Comparison of Results

		RID	RID Cost ($)	Profit ($)	Finance ($)	Total ($)
Minimum RID	*Existing*	0	-	-	-	-
	Proposed	0	-	-	-	-
Maximum profit	*Existing*	-	-	1,294.9	-	-
	Proposed	-	-	1,299.6	-	-
Minimum finance	*Existing*	-	-	-	2,302.3	-
	Proposed	-	-	-	2,173.5	-
Best compromise	*Existing*	1	-	976.8	2,743.8	-
	Proposed	0	-	977.4	2,705.8	-
Maximum CRC	*Existing*	1	12.8	976.8	2,743.8	-1,779.8
	Proposed	2	24.1	850.9	2,357.9	-1,555.2

For the single objective solutions, the SOS algorithm gives less "finance expense" and higher "profit" value than SPEA, which are favorable from the point of view of the contractor. Moreover, both algorithms are able to reach the global minimum value of RID, which is 0. For the best compromise solution, the SOS algorithm, again, outperforms the results obtained by SPEA. Besides reaching the minimum value of RID, the best compromise solution outputs a lower value for finance expense and a slightly higher value for profit as in comparison with the existing method. All in all, when the results are evaluated for all of the comparable optimization objectives, it can be seen that the SOS algorithm yields solutions with better quality than SPEA.

In the maximum CRC solution, the CRC value is calculated as -$-1,779.8 by using solutions of the existing method, whereas, the value of CRC is obtained as -$1,555.2 with the proposed method, indicating a significantly better solution for overall optimized schedule, which shows the superiority of the proposed method over the existing method. The comparison results reveal the potential for preferring a practical optimization method that gives satisfactory results for industry practitioners.

Lastly, the processing time of both methods are given in Table 7.16 for further comparison. The processing time is 1,104.3 seconds in the existing method, using a desktop computer having a 3.2 GHz processor and 24 GB RAM, whereas it lasts 768 seconds, in total, to obtain all solutions by the use of the proposed method, using a desktop computer having a 3.4 GHz processor and 16 GB RAM.

Table 7.16. Comparison of Processing Time

	Minimum finance	**Maximum profit**	**Minimum RID**	**Best compromise**	**Maximum CRC**
Existing	1,104.3 s				-
Proposed	97.0 s	94.0 s	95.0 s	385.0 s	97.0 s

CHAPTER 8

CONCLUSIONS

In construction industry, it is becoming more essential to benefit from advanced scheduling techniques to achieve competitiveness in the industry. Consequently, this creates a growing interest in developing comprehensive optimization models that are capable of considering different goals related to project success criteria. However, while both resource management and cash flow analysis compromise two major components of a construction project to attain a satisfactory outcome, many researchers have investigated the problems related to resources and cash flow separately. Therefore, to address the need for an integrated optimization model, this study focuses on combining resource leveling and cash flow optimization for construction projects.

In this thesis study, an integrated method that incorporates scheduling, resource leveling, and cash flow models is presented. The scheduling model gets activity information and financial data as inputs, and carries out Critical Path Method (CPM) calculations to output parameters of early start schedule. In the resource leveling model, two recently introduced metrics are used to measure and quantify the resource fluctuations observed along project duration. The cash flow model is designed to calculate the parameters required for construction of a cash flow profile. The following parameters are used as optimization objectives within the context of this study: (1) resource idle day, (2) minimum of finance expense, (3) net cumulative balance including accumulated financing cost at the end of the project. The first parameter is retrieved from the resource leveling model, and the remaining parameters are obtained from the cash flow model.

A vast majority of existing research offers solutions as a Pareto front, which consists of non-dominated solutions, when multiple objectives are considered. However, the

Pareto front leaves the decision of choosing the end solution to the decision-maker; hence, it might not be practical and may not yield the overall optimal solution. The main contribution of this study is deemed as introduction of a novel optimization method that includes a parameter named *"combined resource and cash flow (CRC)"*, which enables providing a practical approach for construction practitioners when both resource leveling and cash flow optimization are aimed to be considered simultaneously. The use of this parameter leads to a single solution, which promotes its practicality, and indeed, substantially reduces the computational time required to obtain the optimized schedules.

The solution for the optimization problem defined in this study demands an efficient algorithm to generate optimum schedules in a short time with satisfactory results. For this purpose, a recently developed meta-heuristic algorithm, named Symbiotic Organisms Search (SOS) algorithm is utilized in this study. The primary merit of the SOS algorithm is having no parameters to be fine-tuned. Besides, the outstanding performance was well-proven against other commonly used meta-heuristic algorithms. Though, similar to any other meta-heuristic, optimality of solutions cannot be guaranteed. Moreover, very few researchers implement this algorithm for problems associated with construction project management. In that sense, another contribution of this study can be regarded as extending the application areas of this recently introduced algorithm to construction project management, towards which, a case construction example is practiced.

The results are obtained for minimum resource idle day, maximum net cumulative balance including accumulated financing cost at the end of the project, minimum finance expense, best compromise solution, and maximum combined resource and cash flow, separately. The solution with maximum CRC exhibits considerably less finance expense value than that of best compromise solution, which could promote the use of this parameter instead of best compromise solution while considering different objectives simultaneously. On the other hand, maximum CRC solution gives less net cumulative balance value compared to best compromise solution. The solutions obtained by the SOS algorithm are compared to the solutions of the existing method in which the Strength Pareto Evolutionary Algorithm (SPEA) is implemented. The results of the comparison reveal that the SOS algorithm is capable of finding better solutions for cash flow related objectives, and reaches the global minimum value of resource idle day.

Although this study involves many realistic aspects that are consistent with practical applications, there is, in essence, a significant potential for improvement. Firstly, for the problem under consideration, unlimited cash is assumed to be available in case of negative cash flow. In a future study, constraints related to available cash can be included in order to reflect the realistic cases. Uncertain nature of activity information can also be embedded into the problem. In another future research, more complex project networks with large-scale projects can be studied since construction projects have many activities, in general. Moreover, this problem can be extended such that multiple projects with shared resources of different kinds are considered. Last but not least, the proposed method can be integrated into commonly used project scheduling software by means of an add-in to provide construction practitioners an optimization module that combines resource leveling and cash flow optimization.

REFERENCES

Abdel-Basset, M., Ali, M., & Atef, A. (2019). Resource levelling problem in construction projects under neutrosophic environment. *Journal of Supercomputing*.

Abido, M., & Elazouni, A. (2011). Multiobjective evolutionary finance-based scheduling: Entire projects' portfolio. *Journal of Computing in Civil Engineering*, *25(1)*, 85-97.

Afshar, A., & Fathi, H. (2009). Fuzzy multi-objective optimization of finance-based scheduling for construction projects with uncertainties in cost. *Engineering Optimization*, *41(11)*, 1063–1080.

Al-Shihabi, S.T., & AlDurgam, M.M. (2017). A max-min ant system for the finance-based scheduling problem. *Computers and Industrial Engineering*, *110*, 264–276.

Alavipour, S.M.R., & Arditi, D. (2018). Optimizing financing cost in construction projects with fixed project duration. *Journal of Construction Engineering and Management*, *144(4)*.

Alavipour, S.M.R., & Arditi, D. (2019a). Maximizing expected contractor profit using an integrated model. *Engineering, Construction and Architectural Management*, *26(1)*, 118–138.

Alavipour, S.M.R., & Arditi, D. (2019b). Time-cost tradeoff analysis with minimized project financing cost. *Automation in Construction*, *98*, 110–121.

Ali, M.M., & Elazouni, A. (2009). Finance-based CPM/LOB scheduling of projects with repetitive non-serial activities. *Construction Management and Economics*, *27(9)*, 839–856.

Anagnostopoulos, K.P., & Koulinas, G.K. (2010). A simulated annealing hyperheuristic for construction resource levelling. *Construction Management and Economics, 28(2)*, 163–175.

Arditi, D., Koksal, A., & Kale, S. (2000). Business failures in the construction industry. *Engineering, Construction and Architectural Management, 7(2)*, 120–132.

Ashuri, B., & Tavakolan, M. (2012). Fuzzy enabled hybrid genetic algorithm-particle swarm optimization approach to solve TCRO problems in construction project planning. *Journal of Construction Engineering and Management, 138(9)*, 1065–1074.

Ashuri, B., & Tavakolan, M. (2015). Shuffled frog-leaping model for solving time-cost-resource optimization problems in construction project planning. *Journal of Computing in Civil Engineering, 29(1)*, 1–11.

Au, T., & Hendrickson, C. (1986). Profit measures for construction projects. *Journal of Construction Engineering and Management, 112(2)*, 273–286.

Bandelloni, M., Tucci, M., & Rinaldi, R. (1994). Optimal resource leveling using non-serial dynamic programming. *European Journal of Operational Research, 78(2)*, 162–177.

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

Cheng, M.-Y., & Prayogo, D. (2014). Symbiotic organisms search: A new metaheuristic optimization algorithm. *Computers and Structures, 139*, 98–112.

Cheng, M.-Y., Prayogo, D., & Tran, D.H. (2016). Optimizing multiple-resources leveling in multiple projects using discrete symbiotic organisms search. *Journal of Computing in Civil Engineering, 30(3)*.

Cheng, M.-Y., Tran, D.H., & Hoang, N.D. (2017). Fuzzy clustering chaotic-based differential evolution for resource leveling in construction projects. *Journal of Civil Engineering and Management, 23(1),* 113–124.

Christodoulou, S.E., Ellinas, G., & Michaelidou-Kamenou, A. (2010). Minimum moment method for resource leveling using entropy maximization. *Journal of Construction Engineering and Management, 136(5),* 518–527.

Damci, A., & Polat, G. (2014). Impacts of different objective functions on resource leveling in construction projects: A case study. *Journal of Civil Engineering and Management, 20(4),* 537–547.

Dhillon, J.S., Parti, S.C., & Kothari, D.P. (1993). Stochastic economic emission load dispatch. *Electric Power Systems Research, 26(3),* 179–186.

Doulabi, S.H.H., Seifi, A., & Shariat, S.Y. (2011). Efficient hybrid genetic algorithm for resource leveling via activity splitting. *Journal of Construction Engineering and Management, 137(2),* 137–146.

Easa, S.M. (1989). Resource leveling in construction by optimization. *Journal of Construction Engineering and Management, 115(2),* 302–316.

El-Abbasy, M.S., Elazouni, A., & Zayed, T. (2016). MOSCOPEA: Multi-objective construction scheduling optimization using elitist non-dominated sorting genetic algorithm. *Automation in Construction, 71(Part 2),* 153–170.

El-Abbasy, M.S., Elazouni, A., & Zayed, T. (2017). Generic scheduling optimization model for multiple construction projects. *Journal of Computing in Civil Engineering, 31(4),* 1–17.

El-Abbasy, M.S., Zayed, T., & Elazouni, A. (2012). Finance-based scheduling for multiple projects with multimode activities. *Construction Research Congress 2012: Construction Challenges in a Flat World, Proceedings of the 2012 Construction Research Congress*, 386–396.

El-Rayes, K., & Jun, D.H. (2009). Optimizing resource leveling in construction projects. *Journal of Construction Engineering and Management*, 135(11), 1172–1180.

El-Sayegh, S. (2009). Resource levelling optimization model considering float loss impact. *Engineering, Construction and Architectural Management, 25(5)*, 639–653.

Elazouni, A. (2009). Heuristic method for multi-project finance-based scheduling. *Construction Management and Economics, 27(2)*, 199–211.

Elazouni, A., & Abido, M. (2014). Enhanced trade-off of construction projects: Finance-resource-profit. *Journal of Construction Engineering and Management, 140(9)*, 1–10.

Elazouni, A., & Gab-Allah, A.. (2004). Finance-based scheduling of construction projects using integer programming. *Journal of Construction Engineering and Management, 130(1)*, 15–24.

Elazouni, A., & Metwally, F.G. (2007). Expanding finance-based scheduling to devise overall-optimized project schedules. *Journal of Construction Engineering and Management, 133(1)*, 86–90.

Elbeltagi, E., Ammar, M., Sanad, H., & Kassab, M. (2016). Overall multiobjective optimization of construction projects scheduling using particle swarm. *Engineering, Construction and Architectural Management, 23(3)*, 265–282.

Elwany, M.H., Korish, I.E., Barakat, M.A. & Hafez, S.M. (1998). Resource smoothening in repetetive projects. *Computers and Industrial Engineering, 35(3-4)*, 415–418.

Ezugwu, A.E., & Prayogo, D. (2019). Symbiotic organisms search algorithm: Theory, recent advances and applications. *Expert Systems with Applications, 119*, 184–209.

Fathi, H., & Afshar, A. (2010). GA-based multi-objective optimization of finance-based construction project scheduling. *KSCE Journal of Civil Engineering, 14(5)*, 627–638.

Gajpal, Y., & Elazouni, A. (2015). Enhanced heuristic for finance-based scheduling of construction projects. *Construction Management and Economics, 33(7),* 531–553.

Geng, J.-Q., Weng, L.-P., and Liu, S.-H (2011). An improved ant colony optimization algorithm for nonlinear resource-leveling problems. *Computers and Mathematics with Applications, 61(8),* 2300–2305.

Georgy, M.E. (2008). Evolutionary resource scheduler for linear projects. *Automation in Construction, 17(5),* 573–583.

Ghoddousi, P., Eshtehardian, E., Jooybanpour, S., & Javanmardi, A. (2013). Multi-mode resource-constrained discrete time-cost-resource optimization in project scheduling using non-dominated sorting genetic algorithm. *Automation in Construction, 30*, 216–227.

Hariga, M., & El-Sayegh, S.M. (2011). Cost optimization model for the multiresource leveling problem with allowed activity splitting. *Journal of Construction Engineering and Management, 137(1),* 56–64.

Harris, M. (1990). Packing method for resource leveling (PACK). *Journal of Construction Engineering and Management, 116(2),* 331–350.

He, L., & Zhang, L. (2013). Dynamic priority rule-based forward-backward heuristic algorithm for resource levelling problem in construction project. *Journal of the Operational Research Society, 64(8),* 1106–1117.

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

Hegazi, T., & Ersahin, T. (2001). Simplified spreadsheet solutions. II: Overall schedule optimization. *Journal of Construction Engineering and Management, 127(6)*, 469–475.

Hegazi, T., & Ersahin, T. (2001). Simplified spreadsheet solutions. II: Overall schedule optimization. *Journal of Construction Engineering and Management, 127(6)*, 469–475.

Hiyassat, M.A.S. (2000). Modification of minimum moment approach in resource leveling. *Journal of Construction Engineering and Management, 126(4)*, 278–284.

Iranagh, M.A., & Sonmez, R. (2012). A genetic algorithm for resource leveling of construction projects. *Association of Researchers in Construction Management, ARCOM 2012 – Proceedings of the 28th Annual Conference 2*, 1047–1054.

Jaskowski, P., & Biruk, S. (2018). Reducing renewable resource demand fluctuation using soft precedence relations in project scheduling. *Journal of Civil Engineering and Management, 24(4)*, 355–363.

Jun, D.H., & El-Rayes, K. (2010). Optimizing the utilization of multiple labor shifts in construction projects. *Automation in Construction, 19(2)*, 109–119.

Jun, D.H., & El-Rayes, K. (2011). Multiobjective optimization of resource leveling and allocation during construction scheduling. *Journal of Construction Engineering and Management, 137(12)*, 1080–1088.

Kaiafa, S., & Chassiakos, A.P. (2015). A genetic algorithm for optimal resource-driven project scheduling. *Procedia Engineering, 123*, 260–267.

Kim, K., Walewski, J., & Cho, Y.K. (2016). Multiobjective construction schedule optimization using modified niched Pareto genetic algorithm. *Journal of Management in Engineering, 32(2)*, 1–12.

Kim, J., Kim, K., Jee, N., & Yoon, Y. (2005). Enhanced resource leveling technique for project scheduling. *Journal of Asian Architecture and Building Engineering, 4(2)*, 461–466.

Koulinas, G.K., & Anagnostopoulos, K.P. (2013). A new tabu search-based hyper-heuristic algorithm for solving construction leveling problems with limited resource availabilities. *Automation in Construction, 31*, 169–175.

Kyriklidis, C., Vassiliadis, V., Kirytopoulos, K., & Dounias, G. (2014). Hybrid nature-inspired intelligence for the resource leveling problem. *Operational Research, 14(3)*, 387–407.

Leu, S.-S., Chen, A.-T., & Yang, C.-H. (1999). A fuzzy optimal model for construction resource leveling scheduling. *Canadian Journal of Civil Engineering, 26(6)*, 673–684.

Leu, S.-S., & Hung, T.-H. (2002). An optimal construction resource leveling scheduling simulation model. *Canadian Journal of Civil Engineering, 29(2)*, 267–275.

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

Leu, S.-S., Yang, C.-H., & Huang, J.-C. (2000). Resource leveling in construction by genetic algorithm-based optimization and its decision support system application. *Automation in Construction, 10(1)*, 27–41.

Li, H., & Demeulemeester, E. (2016). A genetic algorithm for the robust resource leveling problem. *Journal of Scheduling, 19(1)*, 43–60.

Li, H., Wang, M., & Dong, X. (2019). Resource leveling in projects with stochastic minimum time lags. *Journal of Construction Engineering and Management, 145(4)*, 1–13.

Li, H., Xiong, L., Liu, Y., & Li, H. (2018). An effective genetic algorithm for the resource levelling problem with generalised precedence relations. *International Journal of Production Research, 56(5)*, 2054–2075.

Liu, S.S., & Wang, C.J. (2008). Resource-constrained construction project scheduling model for profit maximization considering cash flow. *Automation in Construction, 17(8)*, 966–974.

Lucko, G. (2011). Optimizing cash flows for linear schedules modeled with singularity functions by simulated annealing. *Journal of Construction Engineering and Management, 137(7)*, 523–535.

Mattila, K.G., & Abraham, D.M. (1998). Resource leveling of linear schedules using integer linear programming. *Journal of Construction Engineering and Management*, *124(3)*, 232–243.

Menesi, W., & Hegazy, T. (2015). Multimode resource-constrained scheduling and leveling for practical-size projects. *Journal of Management in Engineering, 31(6)*, 1–7.

Neumann, K., Schwindt, C., & Zimmermann, J. (Ed.) (2003). *Project scheduling with time windows and scarce resources: Temporal and resource-constrained projectsScheduling with regular and nonregular objective functions* (2nd ed.). Germany: Springer.

Neumann, K., & Zimmermann, J. (1999). Resource levelling for projects with schedule-dependent time windows. *European Journal of Operational Research*, *117*, 591–605.

Nikoofal Sahl Abadi, N., Bagheri, M., & Assadi, M. (2018). Multiobjective model for solving resource-leveling problem with discounted cash flows. *International Transactions in Operational Research, 25(6)*, 2009–2030.

Piryonesi, S.M., Nasseri, M., & Ramezani, A. (2019). Resource leveling in construction projects with activity splitting and resource constraints: A simulated annealing optimization. *Canadian Journal of Civil Engineering, 46(2)*, 81–86.

Ponz-Tienda, J.L., Salcedo-Bernal, A., & Pellicer, E. (2017). A parallel branch and bound algorithm for the resource leveling problem with minimal lags. *Computer-Aided Civil and Infrastructure Engineering, 32(6),* 474–498.

Ponz-Tienda, J.L., Salcedo-Bernal, A., Pellicer, E., & Benlloch-Marco, J. (2017). Improved adaptive harmony search algorithm for the resource leveling problem with minimal lags. *Automation in Construction, 77,* 82–92.

Ponz-Tienda, J.L., Yepes, V., Pellicer, E., & Moreno-Flores, J. (2013). The resource leveling problem with multiple resources using an adaptive genetic algorithm. *Automation in Construction, 29*, 161–172.

Prayogo, D., Cheng, M.-Y., Wong, F.T., Tjandra, D., & Tran, D.H. (2018). Optimization model for construction project resource leveling using a novel modified symbiotic organisms search. *Asian Journal of Civil Engineering, 19(5)*, 625–638.

Prayogo, D., & Kusuma, C.T. (2019). Optimization of resource leveling problem under multiple objective criteria using a symbiotic organisms search. *Civil Engineering Dimension, 21(1*), 43–49.

Razavi, N., & Mozayani, N. (2007). A resource leveling model based on genetic algorithms: Activity splitting allowed. *Lecture Notes in Engineering and Computer Science*, 109–114.

Rieck, J., Zimmermann, J., & Gather, T. (2012). Mixed-integer linear programming for resource leveling problems. *European Journal of Operational Research, 221(1),* 27–37.

Sears, S.K., Sears, G.A., & Clough, R.H. (Ed.) (2008). *Construction project management: A practical guide to field construction management* (5th ed.). New Jersey: Wiley.

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

Son, J., & Mattila, K.G. (2004). Binary resource leveling model: activity splitting allowed. *Journal of Construction Engineering and Management, 130(6),* 887–894.

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

Tang, Y., Liu, R., & Sun, Q. (2014). Schedule control model for linear projects based on linear scheduling method and constraint programming. *Automation in Construction, 37,* 22–37.

Tavakolan, M., & Nikoukar, S. (2019). Developing an optimization financing cost-scheduling trade-off model in construction project. *International Journal of Construction Management, 1–16.*

Tizhoosh, H. R. (2005). Opposition-based learning: A new scheme for machine intelligence. *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*. IEEE Computer Society, USA, 695–701.

Tran, D.-H., Cheng, M.-Y., & Prayogo, D. (2016). A novel multiple objective symbiotic organisms search (MOSOS) for time-cost-labor utilization tradeoff problem. *Knowledge-Based Systems, 94,* 132–145.

Tran, D.-H., Chou, J.-S., & Luong, D.-L. (2019). Multi-objective symbiotic organisms optimization for making time-cost tradeoffs in repetitive project scheduling problem. *Journal of Civil Engineering and Management*, *25(4)*, 322–339.

Tran, D.-H., Luong-Duc, L., Duong, M.-T., Le, T.-N., & Pham, A.-D. (2018). Opposition multiple objective symbiotic organisms search (OMOSOS) for time, cost, quality and work continuity tradeoff in repetitive projects. *Journal of Computational Design and Engineering*, *5*, 160–172.

Younis, M.A., & Saad, B. (1996). Optimal resource leveling of multi-resource projects. *Computers and Industrial Engineering*, *31(1-2)*, 1–4.

Zahraie, B., & Tavakolan, M. (2009). Cash flow modeling for construction projects. *Journal of Construction Engineering and Management*, *135(11)*, 1162–1171.

Zayed, T., & Liu, Y. (2014). Cash flow modeling for construction projects. *Engineering, Construction and Architectural Management*, *21(2)*, 170–189.

Zhao, S.-L, Liu, Y., Zhao, H.-M., & Zhou, R.-L. (2006). GA-based resource leveling optimization for construction project. *Proceedings of the 2006 International Conference on Machine Learning and Cybernetics*, Dalian, China, 2363–2367.

Zitzler, E., & Thiele, L. (1998). An evolutionary algorithm for multiobjective optimization: The strength pareto approach. *TIK-Report, No. 43*. Swiss Federal Institute of Technology, Zurih, Switzerland.