DEVELOPMENT OF HIGH PERFORMANCE HEURISTIC AND META-HEURISTIC METHODS FOR RESOURCE OPTIMIZATION OF LARGE SCALE CONSTRUCTION PROJECTS

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

BY
MAHDI ABBASI IRANAGH

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
CIVIL ENGINEERING

JULY 2015

Approval of the thesis:

DEVELOPMENT OF HIGH PERFORMANCE HEURISTIC AND META-HEURISTIC METHODS FOR RESOURCE OPTIMIZATION OF LARGE SCALE CONSTRUCTION PROJECTS

submitted by **MAHDI ABBASI IRANAGH** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Civil Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Ahmet Cevdet Yalçıner
Head of Department, **Civil Engineering**

Assoc. Prof. Dr. Rıfat Sönmez
Supervisor, **Civil Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. M. Talat Birgönül
Civil Engineering Dept., METU

Assoc. Prof. Dr. Rıfat Sönmez
Civil Engineering Dept., METU

Assoc. Prof. Dr. Ali Murat Tanyer
Architectural Dept., METU

Assoc. Prof. Dr. Mehmet Kabak
Industrial Engineering Dept., Gazi University

Assist. Prof. Dr. Tankut Atan
Industrial Engineering Dept., Işık University

Date: 10.07.2015

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 : Mahdi Abbasi Iranagh

Signature :

ABSTRACT

DEVELOPMENT OF HIGH PERFORMANCE HEURISTIC AND META-HEURISTIC METHODS FOR RESOURCE OPTIMIZATION OF LARGE SCALE CONSTRUCTION PROJECTS

Abbasi Iranagh, Mahdi
PhD., Department of Civil Engineering
Supervisor: Assoc. Prof. Dr. Rıfat Sönmez

July 2015, 154 Pages

Despite the importance of resource optimization in construction scheduling, very little success has been achieved in solving the resource leveling problem (RLP) and resource constrained discrete time-cost trade-off problem (RCDTCTP), especially for large-scale projects. The major objective of this thesis is to design and develop new heuristic and meta-heuristic methods to achieve fast and high quality solutions for the large-scale RLP and RCDTCTP.

Two different methods are presented in this thesis for the RLP, including a memetic algorithm with simulated annealing (MASA) that is adequately generic for unraveling RLPs incorporating any type of known objective functions, and a hybrid genetic algorithm which limits the searching space to only quasistable schedules (QHGA). QHGA is capable of minimizing the sum of squares of daily resource usage or total overloaded amount from a desired level of resource consumptions, for large-scale projects in a very short computation time. The computational experiments reveal that both MASA and QHGA outperform the state-of-art methods for the RLP. QHGA is also integrated to Microsoft Project to enhance the use of the proposed leveling method in practice

The final proposed algorithm within the thesis is a heuristic method which is designed and developed to achieve fast and high quality solutions for the large-scale RCDTCTP. The proposed heuristic consists of two parts including the scheduling and the crashing parts. The scheduling part adopts backward-forward scheduling technique for the resource constrained project scheduling problem. In the second part, the critical sequence including the activities that determine the project duration for a resource constrained schedule are crashed. The computational experiment results reveal that the new critical sequence crashing heuristic outperforms the other state-of-art methods, both in terms of the solution quality and computational time. The main contribution of the thesis is that it provides fast and effective methods for optimal scheduling and resource allocation of real-life-size construction projects.

Keywords: Resource Optimization; Resource Leveling; Project Scheduling; Genetic Algorithm; Simulated Annealing; Memetic Algorithm; Heuristics.

ÖZ

BÜYÜK ÖLÇEKLİ İNŞAAT PROJELERİNDE KAYNAK OPTİMİZASYONU İÇİN YÜKSEK PERFORMANSLI SEZGİSEL VE ÜST-SEZGİSEL ALGORİTMALAR GELİŞTİRİLMESİ

Abbasi Iranagh, Mahdi
Doktora, İnşaat Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Rıfat Sönmez

Temmuz 2015, 154 Sayfa

Kaynak optimizasyonun inşaat projelerinin planlaması ve programlaması aşamalarında kritik önem teşkil etmesine rağmen, özellikle büyük ölçekli inşaat projeleri için kaynak dengelemesi problemi (KDP) ve kaynak kısıtlı zaman-maliyet ödünleşim probleminin (KKZMÖP) çözümünde çok sınırlı başarı elde edilebilmiştir. Bu tezin temel amacı büyük ölçekli projelerde KDP ve KKZMÖP için kısa sürede kaliteli çözümler elde edebilen sezgisel ve üst-sezgisel yöntemler tasarlanması ve geliştirilmesidir.

Bu tezde kaynak dengeleme problemi için iki farklı yöntem geliştirilmiştir. Bunlardan ilki, farklı amaç fonksiyonları için çözüm üretebilen bir tavlama benzetimli memetik algoritmadır (MASA). Diğer yöntemse, literatürde quasistable terimi ile tanımlanan iş programlarını tarayan ve böylece çözüm kümesini küçülterek kısa sürede kaliteli çözümler elde etmeyi hedefleyen bir melez gen algoritmasıdır (QHGA). QHGA büyük ölçekli projeler için günlük kaynak kullanım karelerinin toplamının veya hedeflenen günlük kaynak miktarı üzerindeki toplam kaynak kullanım miktarının çok kısa sürede minimize edilmesi amacıyla geliştirilmişir. Geliştirilen bu iki yöntem literatürde yer alan problemlerle test edilmiştir. Bu testler sonucunda önerilen kaynak dengeleme yöntemleri, literatürdeki mevcut yöntemlerden daha iyi sonuçlar elde etmişir. QHGA’nın sektörde kullanımı artırmak amacıyla, bu algoritma Microsoft Project programına entegre edilmiştir.

Tez kapsamında geliştirilen üçüncü bir yöntem ise, büyük ölçekli KKZMÖP için kısa sürede kaliteli sonuçlar elde edilebilmesini hedeflemektedir. Bu kapsamda önerilen sezgisel yöntem iki kısımdan oluşmaktadır. İş programlaması kısmında, geri-ileri iş programlaması yöntemi kaynak kısıtlı iş programlaması problemi için kullanılmıştır. İkinci kısım ise, kaynak kısıtlı iş programı için proje süresini belirleyen kritik iş sırasının kırılmasından oluşmaktadır. Yapılan testler önerilen sezgisel yöntemin özellikle büyük ölçekli projelerde, literatürdeki mevcut yöntemlere göre KKZMÖP çözümü için hem daha az bir işlem süresi gerektirdiğini hem de daha kaliteli çözümler elde ettiğini göstermektedir. Tez kampsamında geliştirilen yöntemler özellikle gerçek inşaat projelerinin ölçeği mertebesindeki büyük ölçekli problemlerde iş programı ve kaynak optimizasyonu için hızlı ve etkili metotlar geliştirilmesi doğrultusunda önemli katkılar sağlamaktadır.

Anahtar Kelimeler: Kaynak Optimizasyonu; Kaynak Dengeleme; İş Programlaması; Gen Algoritması; Tavlama Benzetimi; Memetik Algoritma, Sezgiseller.

To my beloved family

ACKNOWLEDGEMENTS

I would like to express my special appreciation and thanks to my advisor Assoc. Prof. Dr. Rıfat Sönmez. You have been a tremendous mentor for me. I would like to thank you for encouraging my research and for allowing me to grow as a researcher. Your advice on both research as well as on my career have been priceless. I would also like to thank Assist. Prof. Dr. Tankut Atan for his valuable and endless supports throughout my thesis study. I would like to thank to Prof. Dr. Talat Birgönül for his continuous supports from the beginning of my PhD. It has been a great privilege for me to meet with him. I would also like to thank my committee members, Assoc. Prof. Dr. Ali Murat Tanyer and Assoc. Prof. Dr, Mehmet Kabak for serving as my committee members even at hardship.

The financial supports of TÜBİTAK to the research projects of 111M140 and 213M253 are gratefully acknowledged. Without the financial support it would be difficult for me to afford living miles away from my homeland.

A special thanks to my dear friend co-research assistant, PhD candidate Saman Aminbakhsh for his comments and share of his experience. He always responded without hesitation when I needed help. I would also like to thank my dear friend, PhD candidate Maryam Daneshvar for her valuable comments on writing my thesis and for the time she spent for me.

Words cannot express how grateful I am to my mother for all of the sacrifices that she has made on my behalf. Your pray for me was what sustained me thus far. I would also like to express my great thanks to my beloved father that even if he is not physically with me, his spiritual support always encourages me.

I would also like to thank all of my friends who supported me in writing, and incented me to strive towards my goal. At the end I would like express appreciation to my beloved wife Elham Gholamian who spent sleepless nights with and was always my support in the moments when there was no one to answer my queries.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ACO	Ant Colony Optimization
ACA	Ant Colony Algorithm
ADIF	Absolute Deviation Metric
AGA	Adaptive Genetic Algorithm
AHP	Analytical Hierarchy Process
ANN	Artificial Neural Network
APD	Average Percent Deviation
CP	Constraint Programing
CPM	Critical Path Method
CPU	Central Processing Unit
CSCH	Critical Sequence Crashing Heuristic
D	Duration
DSS	Decision Support System
DTCTP	Discrete Time-Cost Trade-Off Problem
DUR	Duration
EF	Early Finish Time
ES	Early Start Time
ESA	Early Start Ascending Priority
ESD	Early Start Descending Priority
FS	Finish to Start
GA	Genetic Algorithm
GB	Gigabyte
GHz	Gigahertz
ID	Identity
IDA	Identity Ascending Priority

ADD	Identity Descending Priority
LF	Late Finish Time
LFA	Late Finish Ascending Priority
LFD	Late Finish Descending Priority
LS	Late Start Time
MA	Memetic Algorithm
MASA	Memetic Algorithm with Simulated Annealing
MRD	Maximum Daily Resource Demand
MSP	Microsoft Project
NN	Neural Network
NP-HARD	Non-deterministic Polynomial Time Hard
OVLD	Overload Resource Metric
PACK	Packing Method
PD	Percent Deviation
PSO	Particle Swarm Optimization
PSPLIB	Project Scheduling Problem Library
QHGA	Quasistable Hybrid Genetic Algorithm
RAM	Random Access Memory
RCPSP	Resource Constrained Project Scheduling Problem
RCDTCTP	Resource Constrained Discrete Time-Cost Trade-Off Prooblem
RES	Resource
RID	Resource Idle Days Metric
RLP	Resource Leveling Problem
RR	Resource Requirement
RRH	Release and Rehire Metric
S	Second(s)
SA	Simulated Annealing
SS	Scheduled Start Time
SF	Scheduled Finish Time

SSRR	Sum of Squares Metric
STD	Standard
TCT	Time-Cost Trade-Off
TCTP	Time-Cost Trade-off Problem
TF	Total Float
TFA	Total Float Ascending Priority
TFD	Total Float Descending Priority
USD	United States Dollar

CHAPTER 1

INTRODUCTION

Construction business has become seriously competitive through the recent years. As a result, effective management has turned out to be a major prerequisite for survival of the companies. In addition, construction management constantly deals with challenge of making a project successful within the triple constraints of quality (scope), cost (resources) and schedule (time). Considering the trade-offs between the afore-mentioned three constraints, while ensuring the acceptable quality level, the role of planning and scheduling within the construction project management area of knowledge cannot be overlooked. In project planning and scheduling, the work tasks are defined and a sequential program for consumption of the available resources is developed, while the successful completion of the project within the least possible time is aimed. In other words, planning and scheduling help companies to complete the project on time and within the budget with respect to the predetermined level of quality. Undoubtedly, without adequate planning and scheduling beforehand, the main goals of project management cannot be achieved. Therefore, many construction researchers have focused on optimized scheduling, an area where the construction optimization problems arise.

Critical path method (CPM) is one of the most extensively applied techniques for scheduling of construction projects. CPM performs scheduling by only considering the precedence relationships and does not theoretically consider resource allocation. As a result, two types of resource scheduling problem occur as the resource leveling problem (RLP) and resource constrained project scheduling problem (RCPSP).

Regarding the RLP, it is assumed that there are unlimited available resources within the project. The schedules prepared by CPM unavoidably encompass undesired fluctuations in the resource usage profile, since all the activities are scheduled to be started on their earliest times. These fluctuations in the manpower and machinery diagrams often cause extra labor or financial expenditure (Ballestín, Schwindt, & Zimmermann, 2007; Easa, 1989). Hence, resource leveling which aims to minimize the aforementioned fluctuations is one of the essential aspects of construction scheduling to have an efficient resource allocation and reduced project cost.

RCPSP on the other hand, occurs when there are limited resources available in the project and it is aimed to complete the project within the shortest possible time with respect to the available amounts of resources. The general RCPSP aims to achieve the minimum project duration that satisfies both the precedence and resource constraints. More specifically, RCPSP solution ensures efficient allocation of the available resources in such a way that the project is completed in the shortest possible time period without exceeding the resource limitations.

The other type of project scheduling optimization problem is time cost trade off problem (TCTP). Time and cost are both aimed to be minimized, however, due to the inherent trade-off relationship between time and cost, the impact of both shall be taken into account simultaneously. As a result, in this case, the original single objective time or cost optimization problem is shifted to the multi objective TCT optimization problem. Since many resource types such as manpower and equipment exhibit discrete nature, numerous researches have focused on the discrete version of the TCTP, called as discrete time cost trade off problem (DTCTP). Within the relevant literature, Discrete TCTP has also been studied under three categories of deadline, budget, and time-cost curve problems. In the deadline problem, the total project cost is minimized while an upper bound completion time is considered as the project deadline. Whereas, the budget type of DTCTP aims to minimize the project duration without exceeding a budget amount as the upper bound. Finally, in the time-cost curve problem, a set of solutions are mapped, which represent optimal total costs related to any feasible completion time called as non-dominated solutions.

If resource constrained problem is taken into account in discrete TCT problem, the new multi objective problem of resource constrained discrete time cost trade-off problem (RCDTCTP) is formed, which is also covered within the scope of this thesis. The objective of RCDTCTP is to settle a time/cost/resource option with a start date for each activity in such a way that, the precedence and resource constraints are satisfied, and the total project cost is minimized.

1.1. Scope of the Thesis

Within the scope of this study, two types of problems related to projects with completion deadline including RLP and RCDTCTP have been focused. All the relationships between the activities have been assumed to be finish-to-start (FS) with zero lag time. The whole parameters have been supposed to be deterministic with static structure.

1.1.1. Resource Leveling Study

Resource leveling is crucial for optimal planning of construction resources, particularly manpower and machinery types of resources, to minimize project overall costs. Despite the importance of resource leveling in practice, commercial project management software use simple priority based heuristics, and have very limited capabilities for solving the resource leveling problem (Iranagh & Sonmez, 2012; Son & Mattila, 2004). Hence, development of effective optimization methods for resource leveling, which is one of the main objectives of this thesis study, has both theoretical and practical relevance. The methods proposed for RLP could be categorized as the exact, heuristics, and meta-heuristics methods. RLP is NP-hard (non-deterministic polynomial-time hard) in the strong sense (Neumann, Schwindt, & Zimmermann, 2003) and as the problem size increases, the required problem solving time grows exponentially. Hence, exact methods can only solve problems including few activities. In a recent study of scheduling problems subject to general temporal constraints, instances up to 50 activities and five resources were solved to optimality (Rieck, Zimmermann, & Gather, 2012).

Within the relevant literature, numerous heuristic procedures have been proposed regarding RLP. Most of the heuristic methods used simple shifting heuristics with priority-rule techniques and very small size case examples were tested to validate the methods. Moreover, computational experiments were not implemented for performance evaluation in majority of heuristic studies. Few studies focused on evaluating the capabilities of the project management software in RLP (Iranagh & Sonmez, 2012; Son & Mattila, 2004).

Over the recent years, there has been an increasing interest in the adaptation of meta-heuristics in RLP. Genetic algorithms (GAs), artificial neural networks (ANN), particle swarm optimization (PSO), ant colony optimization (ACO) are among the sole meta-heuristic algorithms proposed for the RLP. Limited numbers of research studies integrated various optimization methods to the meta-heuristic algorithms, in order to employ the capabilities and overcome the shortcomings of each technique. Mainly, the early meta-heuristic methods were validated by one or two case examples including up to twenty activities.

While the majority of the meta-heuristics researches on resource leveling have focused on GAs, a sole GA may suffer from a rapid population convergence to local optima (Rudolph, 1994). In contrast, SA has fine tuning capability and good convergence property since its search is based on the cooling schedule (which specifies how the temperature is reduced as the search progresses) (Hajek, 1988). However, a sole SA has low search efficiency as it maintains one solution at a time. In recent years, skilled combinations of GAs with SA were proposed to achieve an efficient search algorithm for many optimization problems (Chen & Shahandashti, 2009; Hwang & He, 2006; Sonmez & Bettemir, 2012).

Within the recent years, beside the hybrid use of meta-heuristics, the recognition of the limitations of sole optimization methods has led to the development of new optimization strategies through combining multiple methods to provide a more efficient behavior and higher flexibility when dealing with real-world and large-scale problems (Blum & Roli, 2008). Memetic algorithms (MAs) were suggested within this context by hybridizing and combining existing algorithmic structures. MAs are extensions of evolutionary algorithms, and are composed of an evolutionary framework and a local search algorithm. Recent studies on MAs have demonstrated that they converge to high-quality solutions more efficiently than the sole evolutionary algorithms as they incorporate the individual learning as a separate process for local refinement (Nguyen, Ong, & Lim, 2009).

As a part of this thesis study, a memetic algorithm with simulated annealing method (MASA) was presented for solving resource leveling problems. The algorithm is composed of an evolutionary framework including a genetic algorithm (GA) with simulated annealing (SA), and a local search algorithm consisting of a shifting heuristic. The main objective of the proposed algorithm is to design an effective optimization strategy for the RLP by integrating complementary strengths of different optimization methods and incorporating the individual learning as a separate process. The proposed algorithm is applicable to resource leveling problems with all types of metrics as objective functions. A computational experiment was also executed for performance evaluation and comparison of MASA to other state-of-art algorithms. For this reason, the problem sets of J30, J60 and J120 were adopted from the project scheduling problems library, PSPLIB (Kolisch & Sprecher, 1997). The exact solutions of J30 set of the problems were obtained using mixed integer linear programing method, to have a benchmark for performance evaluation of MASA. Additionally, a Microsoft Excel interface was integrated into MASA to simplify problem input. Chapter three of the thesis explains the details of MASA.

1.1.2. Resource Constrained Discrete Time-Cost Trade-Off Study

There are extensive amount of researches on both RCPSP and DTCTP within the relevant literature. Nevertheless, very few studies focused simultaneously on these two problems. RCDTCTP has a very important role in planning and management of construction projects as there are resource constraints and project completion deadlines in the majority of real-life projects. Nonetheless, commonly used commercial project management software programs do not provide any options for the time-cost trade-off problem. Besides, they have very inadequate capabilities for solving the RCPSP (Bettemir & Sonmez, 2014; Hekimoglu, 2007; Lu, Lam, & Dai, 2008; Mellentien & Trautmann, 2001).

Due to the NP-hard nature of RCPSP and DTCTP, their main application has been on small size networks with exact methods. Hence, numerous heuristic and meta-heuristic methods were introduced in literature for optimal scheduling of projects under resource constraints or project completion deadlines. Nevertheless, majority of the researches for TCT problem have not considered resource constraints. Although within the literature an extensive amount of research have been concentrated on designing heuristics and meta-heuristics for the RCPSP and DTCTP, a limited number of them can be applied on real-life and large scale construction projects. Furthermore, the few proposed methods which are applicable on large problems usually require a considerable amount of computational time to achieve high quality solutions. In a most recent study, the constraint programming model of Menesi, Golzarpoor, and Hegazy (2013) achieved a solution with 6.39% deviation from the upper bound (best known solution) in 120 minutes. Hence, a significant gap among the literature and the requirements of real-life construction project management regarding the time-cost trade-off problem can be observed.

The final objective of this thesis is to design and develop a heuristic that can achieve high quality solutions in a short amount of computation time for the large-scale RCDTCTP. It is attempted to provide a fast method for optimal scheduling of real-life-size projects with project completion deadlines and resource constraints. For this purpose, a critical sequence crashing heuristic (CSCH) was introduced consisting two parts of scheduling and crashing. Backward-forward scheduling technique was used in the scheduling part for the resource constrained project scheduling problem. Afterwards, the critical sequences were defined and crashed in the second part. MASA was validated adopting large size problem instances from the literature and compared to other state-of-art methods for RCDTCT problem. A Microsoft Excel interface also was developed, in order to enable simplified data input/output and to improve using of the proposed CSCH in practice. Chapter five of this thesis is devoted to describe the details of CSCH.

1.2. Organization of the Thesis

The remaining chapters of this thesis are organized as follows. A brief introduction about RLP and RCDTCTP with their definitions, followed with the detailed review of literature in Chapter 2. In Chapter 3, specifics of MASA algorithm are explained for resource leveling of construction projects which can be applied for all types of leveling metrics. Chapter 4 presents the details of QHGA which is proposed for resource leveling of large-size construction projects. Chapter 5 is focused on the RCDTCT problem and MASA algorithm which is developed for solution of RCDTCTP in real-life-size construction projects. Finally, Chapter 6 includes the conclusions and the potential improvements for future studies.

CHAPTER 2

LITERATURE REVIEW

In this chapter, the principles, definitions and objectives of both resource leveling problem and resource constrained discrete time-cost trade-off problems are summarized. Additionally, a literature review is given on the methodologies and strategies which have been approached by researchers for dealing with each problem type.

2.1. Resource Leveling Problem (RLP)

Minimizing the total cost is one of the major objectives of the construction projects and having an efficient resource allocation could considerably influence the project cost. However, the Critical path method (CPM) which is commonly used for scheduling of construction projects, often cause undesirable fluctuations in resource utilization profile. These fluctuations are costly to be handled in projects because they require keeping some workers idle during low demand periods, or hiring and releasing the workers in short periods which can bring difficulties in attracting and keeping high-performance work teams (El-Rayes & Jun, 2009; Harris, 1978). Moreover, this situation makes disruption in the learning curve effects and subsequently lowers the productivity ratio (Stevens, 1990). Therefore, resource leveling is one of the crucial aspects which should be considered in project scheduling to have an effective resource allocation and optimized project cost. Resource leveling is to measure and minimize the aforementioned fluctuations in the resource profile based on a defined metric as the objective function.

2.1.1. Problem Definition

The aim of general RLP is to minimize the undesired fluctuations in the resource utilization profile with respect to an objective function while satisfying the precedence relations and using available floats. That is to say that objective of RLP is to schedule the non-critical activities in such a way that the fluctuations in the resource utilization profile are minimized, precedence relations are satisfied, and the project duration is remained unchanged.

Numerous resource leveling metrics have been proposed as the objective function to measure and minimize the fluctuations in the resource utilization profile. Followings are some of the most commonly used metrics for construction projects:

2.1.1.1 Sum of squares of daily resource requirement (SSRR)

The metric determines the sum of squares of daily resource requirements where the weight or cost of each resource type is defined. The mathematical formulation of objective function for the SSRR is as follows:

$$SSRR = \sum_{i=1}^{j} w_i \sum_{m=1}^{n} r_{im}{}^2 \tag{1.1}$$

where;

j is the number of different resource types,

w_i is the relative weight of the i^{th} resource type,

n is the project duration, and

r_{im} is the requirement of all activities on i^{th} resource type at the m^{th} day.

In order to better explain this objective function, the following examples is adopted from Yeniocak (2013). Figure 2.1 shows a sample resource usage profile for total duration of 10 days. Considering the squares of resource profile for each day by SSRR function, it is seen that the days with higher resource usage show stronger tendency to minimization. Hence, this metric has an effective capability of peak minimization than the other metrics. The right hand side histogram in Figure 2.1 represents the possible best resource profile since it has the lowest SSRR value (139). As it can be seen from the figure, SSRR tends to yield a rectangular-shaped resource usage curve.

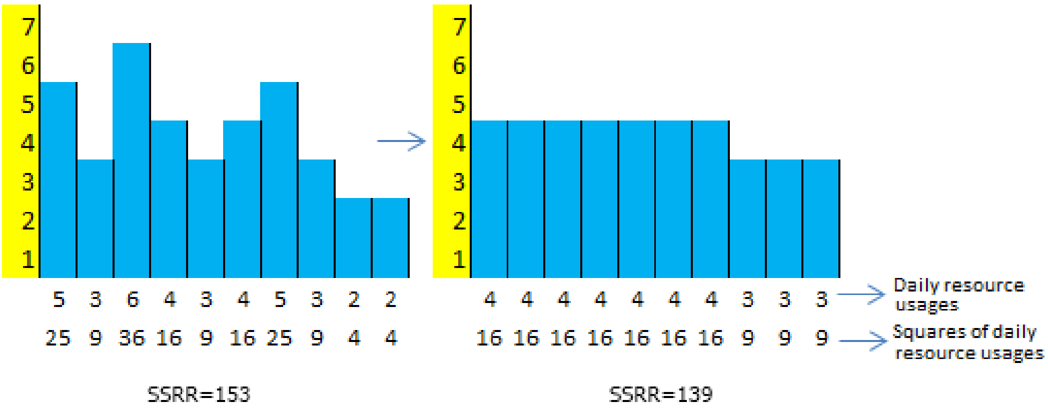


Figure 2.1. SSRR Values of a Sample Profile (Yeniocak, 2013)

2.1.1.2 Absolute differences between the resource requirement and the desired resource consumption (ADIF)

The metric defines the sum of absolute deviations between the resource requirement and the target resource level. The mathematical formulation of the objective function for the ADIF in which the target resource level is taken as the average resource consumption is as follows:

$$ADIF = \sum_{i=1}^{j} w_i \sum_{m=1}^{n} \left| U_i - R_{im} \right| \tag{1.2}$$

$$U_i = \left\lfloor \frac{\sum_{x=1}^{y} DM_{ix} \times DU_x}{n} \right\rfloor \tag{1.3}$$

where;

j is the number of different resource types;

w_i is the relative weight of the i^{th} resource type;

n is the project duration;

R_{im} is the requirement of all activities for resource i at the m^{th} day;

U_i represents the uniform level for the i^{th} resource type;

y is the number of activities;

DM_{ix} is the total demand of activity x for resource i;

DU_x is the duration of activity x; and

“$\lfloor \ldots \rceil$” notation is used for the function which rounds a decimal to the closest integer.

Figure 2.2 demonstrates the same resource profile of the previous section, this time considering the ADIF. Once more, the right hand side histogram exhibits the best possible solution compared to the left hand side profile, since it provides minimum ADIF value. Here also a rectangular-shape resource usage curve is tend to be made.

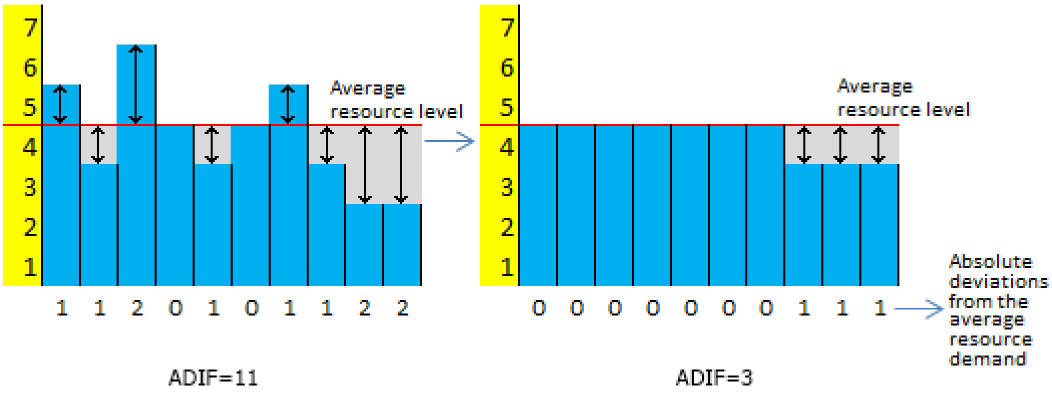


Figure 2.2. ADIF Values of a Sample Profile (Yeniocak, 2013)

2.1.1.3 Overloaded Resource Amounts (OVLD)

This metric aims to minimize the amount of resources that exceeds the desired resource requirement (Rieck et al., 2012). The formulation for the objective function of OVLD in which the target resource level is taken as the average resource consumption is as the following equations:

$$
\begin{aligned}
&OVLD = \sum_{i=1}^{j} w_i \sum_{m=1}^{n} \left(ovld_{im} \right) \\
&where; \\
&if \left(R_{im} - U_i \right) > 0 \rightarrow ovld_m = \left(R_{im} - U_i \right) \\
&else \qquad\qquad \rightarrow ovld_m = 0
\end{aligned}
\tag{1.4}
$$

where;

j is the number of different resource types;

w_i is the relative weight of the i^{th} resource type;

n is the project duration;

$ovld_{im}$ is the total overload amount for resource i at day m;

R_{im} is the requirement of all activities for resource i at the m^{th} day; and

U_i represents the uniform level for the i^{th} resource type.

Figure 2.3 demonstrates calculation of OVLD value for the same resource profile represented in the two previous sections.

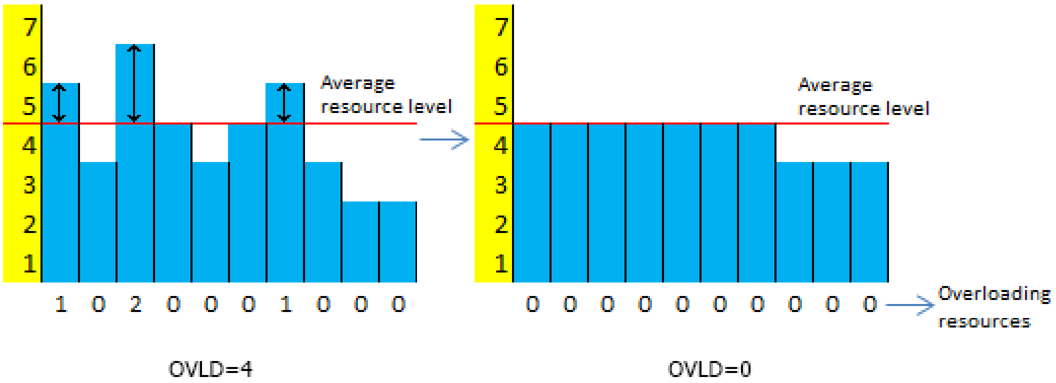


Figure 2.3. OVLD Values of a Sample Profile (Yeniocak, 2013)

The OVLD is very similar to the ADIF. The only difference is that in OVLD metric, the negative deviations from the target are not taken into account. Therefore, all resource bars which are greater than the target resource level are tended to approach toward the average resource level. Repetitively, it can be seen from the Figure 2.3 that like SSRR and ADIF, OVLD also forces generation of a flat, rectangular-shaped resource utilization curve.

2.1.1.4 Resource Idle Days (RID) and Maximum Resource Demand (MRD)

The RID metric has been introduced by El-Rayes and Jun (2009). This metric quantifies the total number of idle and nonproductive resource days during the entire project duration to directly measure and minimize the negative impact of resource fluctuations on construction productivity and cost. The mathematical formulation of the objective function for the RID is as follows:

$$RID = \sum_{i=1}^{j} w_i \sum_{m=1}^{n} \left[Min(Max(r_{i1}, r_{i2}, ..., r_{im}), Max(r_{im}, r_{im+1}, ..., r_{in})) - r_{im} \right] \qquad (1.5)$$

where;

j is the number of different resource types;

w_i is the relative weight of the i^{th} resource type;

n is the project duration;

r_{im} is the requirement of all activities on i^{th} resource type at the m^{th} day.

The resource usage curve obtained using RID might tend to involve high peak resource requirement since this metric does not consider the maximum resource demand (MRD). To overcome this shortcoming of RID, a combined metric of RID-MRD has been suggested by El-Rayes and Jun (2009) to simultaneously minimize the resource idle days and the maximum resource demands. Mathematical formulation of the objective function for RID-MRD is as follows;

$$RID - MRD = (W_1 * RID + W_2 * MRD) \qquad (1.6)$$

where;

MRD is the maximum resource demand during the entire project duration;

W_1 is the planner defined weight for the RID; and

W_2 is the planner defined weight for the MRD.

Figure 2.4 illustrates the calculation of RID and MRD values for the resource curve of the same example discussed in previous sections. The right hand side histogram of Figure 2.4 represents the achievable best resource curve that has the optimal RID-MRD value. As it can be realized from the figure, unlike other metrics, RID does not tend to utilize only a predefined rectangular-shaped graph.

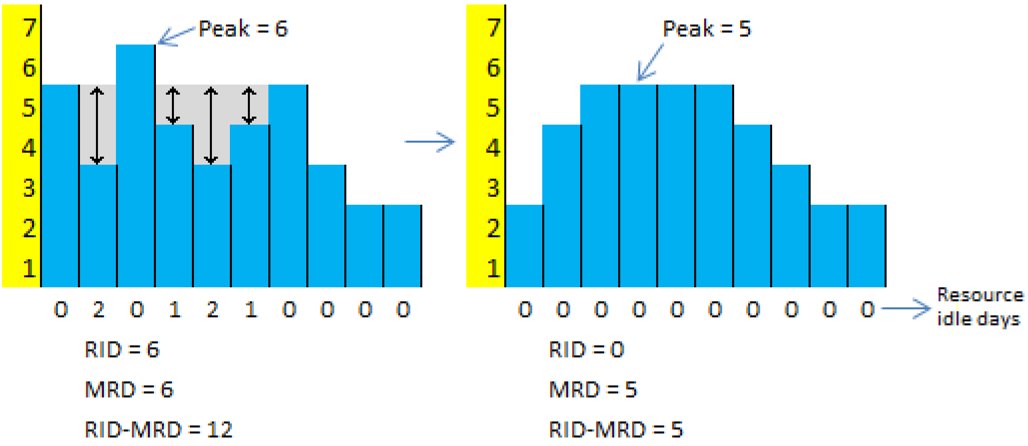


Figure 2.4. RID and MRD Values of a Sample Profile (Yeniocak, 2013)

2.1.2. RLP Literature

The existing methods of solving RLPs can be categorized into three main groups: the exact solution methods, heuristic methods and meta-heuristic algorithms. Exact methods based on dynamic programing (Bandelloni, Tucci, & Rinaldi, 1994), linear-integer programming (Easa, 1989; Mattila & Abraham, 1998; Rieck et al., 2012) and branch-and-bound (Gather, Zimmermann, & Bartels, 2010; Mutlu, 2010; Neumann & Zimmermann, 2000; Yeniocak, 2013) methods have been introduced by the researchers to find the exact solutions of RLPs. However, most of them mainly concentrated on solving the very small networks with a single resource due to the complexity of the RLP which requires significant amount of computational time. Hence, several studies focused on heuristic methods to overcome the problem of complexity in solving the RLPs. Heuristic method including priority-based procedure was first introduced by Burgess and Killebrew in 1962 and later on developed by some other researchers . Development in meta-heuristic algorithms in recent years, has lead several researchers to focus on different methods for the RLPs. Genetic algorithms (GAs), simulated annealing (SA), ant colony algorithms (ACA) and particle swarm optimization algorithms (PSO) are among these methods.

2.1.2.1 Exact Methods Literature

As the earliest contribution on exact methods Petrovic (1969) offered a multistage dynamic programming approach for solving the resource leveling problem. Later, Mason and Moodie (1971) developed a branch and bound algorithm for project scheduling problems that minimizes the total of resource leveling fluctuation cost and cost of delays in project completion time. In this algorithm any extension in project duration has been allowed but penalized according to a cost function. Moreover, a penalty function was applied if the resource demand by activities exceeded available resource levels. Mason and Moodie (1971) stated that according to the test results of the algorithm, the computation time not only depends on the structure of the project network, but is also notably related to the factors such as resource demands and number of activities with their durations.

Easa (1989) presented a linear integer model for optimal solving of RLPs, which minimizes the absolute deviations between the resource requirement and the average resource consumption (ADIF). The model was developed for optimization of single resource problems and tested on a sample network with four activities. Easa (1989) expressed that since a large number of variables and constraints are required in definition of the model, the application of it becomes practically difficult.

Another linear integer programming based model has been developed by Karshenas and Haber (1990) to minimize the total sum of costs related to all project resources and duration. To picture the performance of the model, the costs minimization outcome of two simple problems have been demonstrated. It has been stated that the schedules obtained from the model had not only an optimal duration but also an economically leveled resource usage profile Karshenas and Haber (1990) pointed out the necessity of utilizing a computer program to analyze the extensive data in the process of optimizing the cost of a real life project via the linear integer model.

Shah, Farid, and Baugh (1993) introduced a linear integer optimization model that determines minimum amount of resources required to complete a project. Moreover, Bandelloni et al. (1994) have proposed a non-serial dynamic programming model to reduce the absolute deviations from a required resource level. Later, Demeulemeester (1995) suggested a branch and bound algorithm for solving resource availability cost problem that determines the resource availability levels to minimize the sum of availability costs. Demeulemeester (1995) addressed that the computation time is an increasing function of the number of resource types. Subsequently , Younis and Saad (1996) presented a mathematical model for optimum resource leveling of networks with multiple resources.

Mattila and Abraham (1998) conducted a research on resource leveling of networks modeled with linear scheduling method, an area of research which was rarely studied previously. They presented an integer-linear programing based model, which minimizes the absolute deviations between the resource requirement and the average resource consumption. Linear scheduling method is generally applied for projects such as highways and pipelines construction, high-rise buildings, tunnels construction and etc. Within the scope of this research, the LINDO software package was employed to construct the model and the resource usage profile of a highway construction project was successfully leveled. Like other researchers, Mattila and Abraham (1998) have noted the difficulty of implementing the model on large-size projects due to complexity of problems having large number of variables and constraints.

One of the studies that had extensive contribution to RLP's literature has been published by Neumann and Zimmermann (2000) focusing on both heuristic and exact procedures to solve RLPs of networks with temporal constrains. In this study, minimization of resource fluctuation costs (resource investment problem), minimization of deviations from a predefined resource level and minimization of deviations of consecutive time periods are utilized as objective functions to solve RLP with and without resource limitations. In addition, net present value problem has been addressed via exact methods considering both limited and unlimited resources. The branch and bound and truncated branch and bound methods have been used by Neumann and Zimmermann (2000) in order to solve RLPs. The Branch and bound procedure was based on an enumeration of feasible start times of activities, and the truncated branch and bound procedure was equipped with a heuristic that refines the number of to be produced branches from a single node. According to the test results, most of the problems having up to 20 activities have been solved by Neumann and Zimmermann (2000) for optimality within 100 seconds. Moreover, within the relevant literature, for the first time, problems with 20 activities and 5 resources have been solved for optimal solutions

Nübel (2001) has proposed a depth-first branch and bound algorithm for solving resource renting problems with temporal constraints. The resource renting problem aims to minimize resource availability costs. Both time-independent and time-dependent renting costs have been considered in the study. The algorithm has been based on enumeration of a finite set of schedules that is proven to contain the exact solution. A computational study has been carried out over a randomly generated test set and results are addressed.

Vanhoucke, M. Demeulemeester, E. Herroelen (2001) have introduced a branch and bound algorithm for maximization of the net present value. One problem set from the resource constrained project scheduling problem (RCPSP) literature has been practiced to test the algorithm. It has been indicated that the instances with up to 30 numbers of activities and up to four numbers of resources have been solved optimally for the net present value problem. Afterwards, Son and Mattila (2004) have suggested a linear program binary variable model for RLPs, allowing the split of the activities. In this approach, the activities are permitted to be stopped during their execution period and get restarted later within their floats. Two example problems have been solved and it has been declared that the resource profiles allowing the split of activities are more practical regarding construction projects.

Mutlu (2010) has developed a branch and bound algorithm based on depth-first strategy for solving RLPs in his master thesis. Some problem instances up to 20 activities and 4 resources have been solved by the algorithm using different objective functions including SSRR, ADIF, RID and weighted combination of RID with MRD. One test set consisting of small-scale problems has been solved for RLP, and for the first time the objective function of minimization of resource idle days has been used for testing.

Recently Gather et al. (2010) have proposed a solution procedure to solve RLPs, combining the branch and bound method with the enumeration scheme subject to general temporal constraints. The proposed algorithm has been validated by a computational study using the well-known test sets of Kolisch, Schwindt, and Sprecher (1999) for instances with 10 and 20 numbers of activities. The instances with 20 activities have been solved for optimality for the first time. It has been declared that the algorithm outperformed the other methods known within the RLP literature. More recently, Rieck et al. (2012) have introduced a new mixed-integer linear programming procedure for the RLP subject to general temporal constraints scheduling. In this study, the SSRR and the OVLD metrics have been considered as the objective function. The algorithm has been modeled using CPLEX 12.1. The most comprehensive experimentation up to that time has been conducted in this study using problem sets of Kolisch et al. (1999). All instance problems having up to 30 activities and 5 resources have been solved for optimality. Additionally, the exact solutions of some instances having up to 50 numbers of activities have been determined for the first time within the RLP literature.

Finally Iranagh, Atan, and Sonmez (2013) have developed a mixed-integer linear model to solve RLPs that appoints weighted RID and MRD metrics as the objective function. The GAMS/CPLEX software has been employed to construct the model and it has been integrated into Microsoft Excel software in order to reach a simplified application. The performance of the model has been tested for leveling problems of Kolisch and Sprecher (1997) , which have up to 30 activities and 4 resources.

According to the relevant literature, only few studies have focused on solving the RLPs optimally. Neumann, Schwindt, and Zimmermann (2003) have shown that RLP is NP-hard in the strong sense, even if only one resource is considered. Hence, exact algorithms based on integer-linear programming, dynamic programming, and branch and bound methods can only solve problems that have few numbers of activities

2.1.2.2 Heuristic and Metaheuristic Methods

Due to the complexity of the RLP, which require significant amount of computational time for being solve, several studies focused on heuristics. Heuristic methods including priority-based procedure was firstly introduced by Burgess and Killebrew (1962) and subsequently developed by some other researchers. The algorithm presented by Burgess and Killebrew (1962) simply changes the start times of all non-critical activities one by one according to a priority list such as activity ID to find the best resource profile according to SSRR objective function value. The Burgess and Killebrew heuristic can be applied to a different objective functions and priority rules. Following Burgess and Killebrew (1962) study, Galbreath (1965) has also employed priority based shifting techniques for solving RLPs. Later, Woodworth and Willie (1975) once more have proposed a priority ruled heuristic to solve RLPs in multi-project and multi-resource scheduling.

Harris (1990) has presented a heuristic named as Packing Method (PACK) for solving RLPs in construction projects by minimization of the moment of resource histogram. The method has been approached to have the final distribution of rectangular shape in a way that the moment of the resource profile is minimized. Harris (1990) declared that the performance of the PACK is more capable over previously developed methods because of the fact that it is clear, logical and computationally efficient. Following Harris (1990), some other researchers have also referred to the PACK method. Martinez and Ioannou (1993) have introduced Modified Minimum Moment Method for RLPs in construction projects in order to improve PACK method. In a more recent study, Hiyassat (2000) has suggested some other modifications on PACK method. In this modified method, the resource demands and free slacks of activities have been considered as selection factor to shift activities. It has been stated that the suggested modified approach achieves nearly as effective results as the traditional methods but requires comparably lower computational attempt. Performance of the developed method has been compared with the performance of the traditional method using several problem instances. Later, Hiyassat (2001) declared that the modification of the PACK method has also presented better results for projects with multiple resources.

In a latest attempt, Christodoulou, Ellinas, and Michaelidou-Kamenou (2010) have approached the minimum moment and packing methods through allowing the expansion and compression of the activities. They have noted that by changing resource utilization rates, and incorporating the daily resource limits, better resource usage profiles can be obtained. A method named as “The entropy-maximization method” introduced in this paper, used the theory of entropy to restate the minimum moment method for RLPs. The entropy-maximization problem has been defined in a way to determine the maximum amount of resources, which can be assigned to a specific activity in order to maximize its entropy without exceeding resource limits. Christodoulou et al. (2010) have validated the developed model by two numerical examples.

Through the following years, meta-heuristic algorithms have become popular for solving RLPs like other prevalent optimization problems. That was because of the improvement of these algorithms from one point of view and the necessity to overcome the drawbacks of exact and heuristic methods from another point of view. Artificial neural networks (ANN), genetic algorithms (GAs), simulated annealing (SA), ant colony algorithms (ACA) and particle swarm optimization algorithms (PSO) are among these methods. The first technique different from priority based methods, which has been suggested for solving RLPs was ANN (Savin, Alkass, & Fazio, 1996, 1997; Kartam & Tongthong, 1998). The neural networks (NN) presented by Savin et al. (1996) consists of a discrete-time Hopfield neural network block with a control block to adjust Lagrange multipliers to determine the weights of Hopfield network. The model has been verified using two problem instances with five activities and single resource. Savin et al. (1997) have introduced a new approach for the calculation of the weight-matrix of a NN for RL problems. Later, Kartam and Tongthong (1998) have also proposed a neural network model for resource leveling of construction projects which has taken the advantage of competition-based artificial neural networks over the Hopfield networks. An example problem with nine activities and single resource has been practiced by Kartam and Tongthong (1998) to validate the proposed model. In the process of validation, several problems having up to 100 activities have been employed and the obtained results were compared with results from other RLP methods within the literature and commercial scheduling software programs.

GAs which are inspired by the principles of natural evolution mechanisms, are the most popular meta-heuristic methods that have been employed for solving RLPs. One of the earliest applications of GA in resource leveling problems of construction projects has been performed by Chan, Chua, and Kannan (1996). This study indicated that despite other existing methods, the new model encompasses both resource leveling and limited resource allocation problems. Two case problems, having 11 activities, one with single resource and the other with two resources have been evaluated to demonstrate the performance of the algorithm.

Another GA based algorithm for resource optimization has been developed by Hegazy (1999) for simultaneously optimizing resource allocation and resource leveling. A double-moment approach has been introduced as a modification to resource leveling and employing random priorities have been suggested as an improvement to resource allocation by Hegazy (1999). In addition, the algorithm has been automated using Microsoft Project (MSP) software macro program and a case problem with 20 activities and six resources tested its performance. The required long processing time has been emphasized by Hegazy (1999) as one of the drawbacks of the developed algorithm.

As one of the first hybrid approaches for solving RLPs, Son and Skibniewski (1999) have combined a local optimizer method with SA. It has been noted that SA has empowered the algorithm to escape from local optimal results in many cases. The local optimizer encompassed four heuristic procedures each with different rules to define sequences for shifting activities. On the other hand, one SA model has been employed for searching from the best solution reached by any of the four heuristics in local optimizer. Son and Skibniewski (1999) have verified their algorithm through two single resource example projects, one with 11 and other with 13 activities. The results were reported using SSRR metric as the objective function.

Neumann and Zimmermann (1999) have published a study in which a new methodology has been introduced for resource optimization with temporal constraints. It this study, a polynomial priority-rule based metaheuristic has been presented for the NP-hard RLP and two generalizations of this method have been suggested for resource allocation problem with explicit resource constraints. It has been stated that a feasible solution of the resource leveling problem could be found for the first time in polynomial time although it is an NP–hard problem. Three deferent objective functions of minimization of the deviations from a desired or uniform resource level, minimization of maximum resource costs per period, and minimization of the variations in resource utilization profiles over the time have been explored by authors. Extensive sets of problems up to 500 activities and five resources have been employed and for the first time in RLP literature a detailed experimental performance analysis has been conducted. The results proved that the developed method provides reasonable solutions. Recently, Ballestín, Schwindt, and Zimmermann (2007) have developed a population-based iterated greedy technique considering the production planning. Iterated greedy is a stochastic search meta-heuristic method that generates solutions by iterating through a greedy heuristic using destruction and construction phases. It has been clarified by the authors that the production scheduling problem has been modeled like a resource leveling project scheduling problem, as the orders for final production represented the activities of a project and the variability in the resource utilization over the time has been minimized. Ballestín et al. (2007) have conducted an experimental performance analysis employing a set of temporal scheduling problems up to 1000 activities and five resources. The average computational time for the problems with 1000 activities and up to five resources has been reported as 459.7 seconds. FThe authors have claimed that the proposed iterated greedy method outperformed state-of-the-art RLP heuristics from the literature including the population based method of Neumann and Zimmermann (1999).

Leu, Yang, and Huang (2000) have proposed a GA based algorithm for solving RLPs. A decision support system (DSS) has been introduced by authors for enabling practitioners to involve in the process of optimization and choosing from several resource profiles. Two single resource and one multi resource (three resources) case problems with 11, 13 and 9 activities respectively have been implemented to the model. It has been declared that the developed model has been capable of adequately leveling of resources considering ADIF metric as the objective function. Another algorithm for tackling RLPs and based on GA has been suggested by Zheng, Ng, and Kumaraswamy (2003) that utilized minimum moment approach. In this study, adaptive weights have been applied for leveling multiple resources in order to balance the search pressure among different resource types. Therefore, dominance of any resource type throughout the search process has been avoided. A simple case problem with six activities and two resources has been adopted from the literature to illustrate the concept of the proposed methodology. Zheng et al. (2003) have claimed that the model presented an encouraging performance and is applicable on larger and complicated projects.

Senouci and Eldin (2004) have developed a GA based model for minimization of project total cost considering the precedence relations and multiple crew strategies. In the model formulation, minimizations of the both direct and indirect costs have been targeted. Furthermore, a quadratic penalty function has been involved to the objective function for transforming constrained resource scheduling problem to an unconstrained resource leveling problem. A single resource case example with 12 activities has been implemented to indicate the performance of the method. It has been stated by authors that the algorithm has reached optimal or near optimal results successfully and can be applied on large scale projects.

As the first application of ant colony optimization (ACO) for tackling resource leveling problems, XIONG and KUANG (2006) have presented a hybrid model incorporating serial schedule generation scheme with ACO technique. The ACO method has been developed by Dorigo, Maniezzo, and Colorni (1996), as a global search procedure for optimization of the combinatorial problems. The main idea of the ACO is to simulate the social behavior of an ant searching for the best path to find the food. XIONG and KUANG (2006) have conducted a single resource case example with 13 activities to test the capability and performance of their proposed model. It has been noted by the authors that the developed algorithm could find the global optimal result by scanning only a small portion of the total solution space. Recently, Geng, Weng, and Liu (2010) have employed a directional ACO approach for practicing resource leveling problems. The technique has been declared to be effective and efficient in preventing premature convergence or poor exploitation, as compared with GAs.

Particle swarm optimization (PSO) which has been developed by Kennedy and Eberhart (1995), is another metaheuristic approach that has been employed to find solution for RLPs. The PSO is inspired by the social behavior of a group of migrating birds or schooling fishes trying to find an unknown destination. Despite the GAs, the evolutionary procedure of the PSO does not include creating new birds from parents. Instead, the birds in the population only proceed their movement towards a destination. In PSO each bird makes its decisions based on cognitive aspects which is based on good solutions ever found by the particle itself, and social aspects that is the influence of good solutions found by other particles (Eberhart & Kennedy, 1995; Kennedy & Eberhart, 1995).

As the first PSO based study about RLPs, Qi, Wang, and Guo (2007) have introduced an improved PSO algorithm. In the proposed model, first the position of the particles is checked for feasibility of the schedule, then the PSO begins searching the global and the local bests until the stopping criteria is reached. A case study has been presented implementing a single resource problem with eight activities. The improved PSO model has been shown to be more capable in comparison with other traditional methods. Later, Pang, Shi, and You (2008) have presented another PSO based procedure for leveling of resources. It has been claimed that the probability for the PSO to premature converge has been avoided by using a construction factor. The performance of the algorithm has been tested using a single resource example problem with nine activities. The need for further study for leveling of projects with multiple resources has also been highlighted by the authors. Following to Pang et al. (2008), Guo, Li, and Ye (2009) have developed another PSO method that could be implemented to multiple projects with multiple resources. An analytical hierarchy process (AHP) has been adopted to define the relative weights of the resources. Two case problems with nine activities and three resources have been unraveled by both PSO and GA approached methods and the obtained results have been compared. It has been addressed by Guo et al. (2009) that the proposed PSO model has shown better performance than GA.

El-Rayes and Jun (2009) have presented two new metrics for resource leveling and developed a genetic algorithm based model to compare them with other existing objective functions for RLPs. These metrics are Release and Rehire (RRH), which quantifies amount of the resources that are temporarily released through low demand periods and then rehired later when the demand gets high, and Resource Idle Day (RID), that determines the total idle resources per time throughout the project. The new metrics have been claimed to be more practical since despite the existing metrics, they are not trying to fit the resource profile to a predefined rectangular shape. Rather, they aims to eliminate undesired fluctuations of resource utilization curve. El-Rayes and Jun (2009) have compared these new objective functions with traditional metrics including SSRR and ADIF, using a single resource example network consisting 20 activities.

Bettemir (2009) has compared performance of five different GA based meta-heuristic methods including the sole GA and hybrid GA with simulated annealing, variable neighborhood search and etc. for solving RLPs. Seven projects up to 13 activities have been adopted from the literature, and solved to verify the methods and study their performances. Bettemir (2009) has stated that for all of the test instances, best known solutions have been reached by all the algorithms, however, the hybrid GA with SA has obtained the most promising solutions in shortest time.

Roca, Pugnaghi, and Libert (2008), and Jun and El-Rayes (2011) have employed GA for the solution of resource leveling problem and resource constrained project scheduling problem at the same time. Roca et al. (2008) have published a benchmark set adopting and modifying the problem sets of project scheduling problems library (PSPLIB) (R. Kolisch & Sprecher, 1997), in order to analyze the performance of their proposed algorithm. Jun and El-Rayes (2011) have integrated their model in MSP software program to facilitate its application to construction projects. The example instance of Hegazy (1999) has been adopted by authors for illustrating the application of the model and its validation.

Doulabi, Seifi, and Shariat (2011) have proposed a hybrid GA to tackle RLPs, allowing the activity splitting. The algorithm has been incorporated with a local searching technique and a repair system. An extensive set of example networks up to 5000 activities and nine resources have been generated by the authors in order to verify the algorithm. Doulabi et al. (2011) have provided the optimal solutions for small instances using an existing mixed integer linear programming method from the literature. It has been noted that for large size networks with 5000 activities and up to nine resources, the proposed model could solve the problems in average CPU time of 14502 seconds and reached to an improved value of ADIF for at least 76% better than the early start schedule. Later, Alsayegh & Hariga (2012) also have considered activity splitting in dealing with resource leveling problems and presented a hybrid procedure, combining particle swarm optimization and SA methods to level resources. The minimization of total costs originated from variation of the resource usage and from the splitting non-critical activities, has been defined as the objective function by the authors. Alsayegh and Hariga (2012) have evaluated the cost and computation time performances of the proposed method using a set of benchmark problems.

In a most recent study Ponz-Tienda, Yepes, Pellicer, and Moreno-Flores (2013) have developed a hybrid genetic algorithm for tackling resource leveling problems.

The model has been called adaptive genetic algorithm (AGA) by the authors. Ponz-Tienda et al. (2013) have conducted an experiment analysis to validate MASA by adopting the problem sets of J30, J60 and J120 from the Project Scheduling Problem Library, PSPLIB (Kolisch & Sprecher, 1997). The SSRR metric has been used as the objective function for MASA, and the result have been studied comparing with the early start schedules. Moreover, a three-parameter Weibull distribution has been applied by the authors as a stopping condition for MASA as an estimation of the global optimum. Ponz-Tienda et al. (2013) have declared that the proposed AGA has shown promising performance in comparison to the existing common heuristic methods, especially for the set of problems with 120 activities. The problem sets and results of MASA have been used as a benchmark in the following chapters of this thesis for performance analyzing of the developed resource leveling algorithms.

Despite the importance of resource leveling, the commercial scheduling software products which has being used commonly in construction industry, have very limited capabilities in solving the RLPs. There are very limited studies which have concentrated on evaluating the capabilities of project management software for the RLPs. Furthermore, the majority of these researches have compared capabilities of existing software programs with each other for the resource constrained scheduling problem (Johnson, 1992; Kastor & Sirakoulis, 2009; Maroto & Tormos, 1994; Mellentien & Trautmann, 2001; Trautmann & Baumann, 2009). Son and Mattila (2004) have used a two single resource example problems consist of eleven activities to reveal the limitations of SureTrak Project Manager and Primavera Project Planner (P3). Iranagh and Sonmez (2012) have illustrated the poor capabilities of Microsoft Project 2010 in resource leveling by comparing it with the performance of a sole GA algorithm. Problem instances up to 20 activities have been adopted from the literature by the authors to make the comparison.

The state-of-art heuristic and metaheuristic studies regarding resource leveling problems are summarized in Table 2.1, in a chronological order. General remarks related to each study have also been stated. Overall, majority of the heuristic and meta-heuristic algorithms offered for solution of the RLPs of construction projects, has been evaluated using very small size problem instances up to 20 activities and few resources. Very few of the proposed methods can be applied to large-size problems in practice. Besides, a few methods that are capable of solving large-scale problems usually require a significant amount of computation time to achieve high quality solutions. In addition, the commonly used commercial project management software packages have very limited capabilities to provide quality solutions for RLPs. One of the main objectives of this thesis is to develop high-performance and high-speed methods for resource leveling of real-life-size construction projects.

Table 2.1. Heuristic and Meta-heuristic Algorithms for Resource Leveling Problems (1/7)

Year of Publication	**Author(s)**	**Developed Methods**	**Scheduling Problem**	**Case problem(s)**	**Remarks**
1962	Burgess and Killebrew	Priority based shifting Heuristic	RLP	11 activities, single resource	Simple shifting heuristics or priority-rule based methods for project scheduling problems subject to precedence constraints.
1965	Galbreath			-	
1975	Woodworth and Willie			-	
1990	Harris	Pack method	RLP	11 activities, single resource	Method of minimizing the moment of resource profile has been introduced.
1993	Martinez and Ioannou	Pack method	RLP	-	Modified minimum moment method has been presented.
1996, 1997	Savin, Alkas and Fazio	Neural networks	RLP	five activities, single resource	Using lagrange multipliers in order to determine the weights for Hopfield network.

Table 2.2. Heuristic and Meta-heuristic Algorithms for Resource Leveling Problems (2/7)

Year of Publication	**Author(s)**	**Developed Methods**	**Scheduling Problem**	**Case problem(s)**	**Remarks**
1996	Chan, Chua and Kannan	Genetic algorithm	RLP, RCPSP	11 activities, two resources	A general model to carry out RLP and RCPSP simultaneously
1998	Kartam and Tongthong	Neural networks	RLP	Up to 100 activities, single resource	Employing competition-based artificial neural networks beyond the Hopfield networks
1999	Hegazy	Genetic algorithm	RLP, RCPSP	20 activities, six resources	A double-moment approach has been introduced for RLP, and using random priorities have been suggested for RCPSP
1999	Son and Skibniewski	Simulated annealing	RLP	Up to 13 activities, single resource	A local optimizer method has been combined with simulated annealing.

Table 2.3. Heuristic and Meta-heuristic Algorithms for Resource Leveling Problems (3/7)

Year of Publication	Author(s)	Developed Methods	Scheduling Problem	Case problem(s)	Remarks
1999	Neumann and Zimmermann	Polynomial priority based method	RLP, RCPSP	Up to 500 activities, five resources	Networks with temporal constraints has been considered.
2000	Leu, Yang and Huang	Genetic algorithm	RLP	Up to 13 activities, three resources	A decision support system (DSS) has been introduced for enabling practitioners to involve in the process of optimization and choosing from several resource profiles.
2000	Hiyassat	Pack method	RLP	12 activities, single resource	The resource demands and free slacks of activities have been considered as selection factor to shift them.
2001	Hiyassat	Pack method	RLP	13 activities, two resources	Extending the minimum moment approach to multiple resource leveling.

Table 2.4. Heuristic and Meta-heuristic Algorithms for Resource Leveling Problems (4/7)

Year of Publication	Author(s)	Developed Methods	Scheduling Problem	Case problem(s)	Remarks
2003	Zheng, Ng and Kumaraswamy	Genetic algorithm	RLP	Up to six activities, two resources	Adaptive weights have been applied for leveling multiple resources in order to avoid dominance of any resource type throughout the search process.
2004	Senouci and Eldin	Genetic algorithm	RLP, RCPSP	12 activities, single resource	The minimization of project total cost considering the precedence relations and multiple crew strategies has been considered.
2006	Xiong and Kuang	Ant colony	RLP	13 activities, single resource	A hybrid model incorporating serial schedule generation scheme with ACO technique has been presented.
2007	Ballestin, Schwindt and Zimmermann	Iterated greedy algorithm	RLP	Up to 1000 activities, five resources	The average CPU time for temporal constrained networks with 1000 activities has been reported as 459.7 seconds.

Table 2.5. Heuristic and Meta-heuristic Algorithms for Resource Leveling Problems (5/7)

Year of Publication	**Author(s)**	**Developed Methods**	**Scheduling Problem**	**Case problem(s)**	**Remarks**
2007	Qi, Wang and Guo	Particle swarm optimization	RLP	8 activities, single resource	A PSO algorithm has been proposed incorporating feasibility control process for positions of the particles.
2008	Pang, Shi and You	Particle swarm optimization	RLP	9 activities, single resource	A construction factor has been used to avoid the premature converge.
2008	Roca, Pugnaghi and Libert	Genetic algorithm	RLP, RCPSP	Up to 120 activities, four resources	A two-stage process for tackling RLP and RCPSP simultaneously consists of obtaining non-dominated solutions, and then seeking to improve the solutions.
2009	Guo, Li and Ye	Particle swarm optimization	RLP	9 activities, three resources	An analytical hierarchy process (AHP) has been adopted to define the relative weights of the resources.

Table 2.6. Heuristic and Meta-heuristic Algorithms for Resource Leveling Problems (6/7)

Year of Publication	**Author(s)**	**Developed Methods**	**Scheduling Problem**	**Case problem(s)**	**Remarks**
2009	Bettemir	Genetic algorithm	RLP	Up to 13 activities, three resources	Five different GA based metaheuristic algorithms have been developed and tested in parts of the thesis related to RLP.
2009	El-Rayes and Jun	Genetic algorithm	RLP	20 activities, single resource	Two new metrics of RID and RRH have been introduced for RLPs.
2010	Christodoulou, Ellinas and Kamenou	Pack method	RLP	8 activities, single resource	The entropy-maximization method has been introduced
2010	Geng, Weng and Li	Ant colony	RLP	9 activities, single resource	A directional ACO approach has been developed aiming to prevent premature convergence.
2010	Doulabi, Seifi and Shariat	Genetic algorithm	RLP	Up to 5000 activities, nine resources	The average CPU time for networks with 5000 activities has been reported as 14502 seconds.

Table 2.7. Heuristic and Meta-heuristic Algorithms for Resource Leveling Problems (7/7)

Year of Publication	**Author(s)**	**Developed Methods**	**Scheduling Problem**	**Case problem(s)**	**Remarks**
2011	Jun and El-Rayes	Genetic algorithm	RLP, RCPSP	20 activities, six resources	RLP and RCPSP have been considered simultaneously.
2012	Alsayegh and Hariga	Ant colony	RLP (cost optimization)	Up to 14 activities, six resources	A hybrid model, combining PSO and SA methods for minimization of total costs originated from variation of the resource usage and from the splitting activities.
2012	Iranagh and Sonmez	Genetic algorithm	RLP	Up to 20 activities, single resource	Performance of Microsoft Project software in resource leveling has been analyzed using a sole GA model.
2013	Ponz-Tienda, Yepes, Pollicer and Moreno Flores	Genetic algorithm	RLP	Up to 120 activities, four resources	A hybrid GA model integrated with a three-parameter Weibull distribution as a stopping condition for the model to be an estimation of the global optimum.

2.2. Resource Constrained Discrete Time-Cost Trade-Off Problem (RCDTCTP)

As mentioned in previous sections, critical path method (CPM) is not capable of optimal scheduling of projects when there are resource constraints or project deadlines. Hence, extensive research efforts have focused on the resource-constrained project scheduling problem (RCPSP), and the time/cost tradeoff problem. The general RCPSP aims to achieve the minimum project duration that satisfies both the precedence and resource constraints. The time/cost tradeoff problem, whereas involves minimizing the total direct and indirect costs without exceeding the project deadline. Since in practice many resources (e.g., crews, equipment) are available in discrete units, numerous research have focused on the discrete version of the time/cost trade-off problem called the discrete time-cost trade-off problem (DTCTP). Simultaneous consideration of both RCPSP and DTCTP problems is called as the resource constrained discrete time-cost trade-off problem (RCDTCTP).

2.2.1. Problem Definition

The objective of resource constrained time-cost trade-off problem is to determine a time/cost/resource mode (option) and a start date for each activity in such a way that, the precedence and resource constraints are satisfied, and the total direct costs, indirect costs, and the delay penalties (liquidated damages) are minimized. In the discrete version of this problem the relation between the duration of activities and the committed resources is discrete.

2.2.2. RCDTCTP Literature

RCPSP and DTCTP are both crucial for planning and management of construction projects as there are resource constraints and project completion deadlines in the majority of the projects. However, even the most popular commercial project management software packages have very limited capabilities for solving the RCPSP (Bettemir & Sonmez, 2014; Hekimoglu, 2007; Lu et al., 2008; Mellentien & Trautmann, 2001) and do not provide any options for the time/cost trade-off problem (Menesi et al., 2013).

RCPSP and DTCTP are both NP-hard in the strong case (Blazewicz, Lenstra, & Kan, 1983; De, Dunne, Ghosh, & Wells, 1997), and exact methods can solve these problems for small to medium-size networks. Hence numerous heuristic and meta-heuristic methods were proposed for optimal scheduling of projects under resource constraints or project deadlines. Priority rule based scheduling heuristics (Hegazy, Shabeeb, Elbeltagi, & Cheema, 2000; Özdamar & Ulusoy, 1994; Tormos & Lova, 2001), and meta-heuristics, including genetic algorithms (Chan et al., 1996; P H Chen & Shahandashti, 2009; Hartmann, 1998; Hegazy, 1999; Kim & Ellis, 2008, 2010; Sonmez & Uysal, 2014), simulated annealing (Bouleimen & Lecocq, 2003; Lee & Kim, 1996; Valls, Ballestín, & Quintanilla, 2005), tabu search (Deblaere, Demeulemeester, & Herroelen, 2011), and particle swarm optimization (Chen, 2011; Lu et al., 2008; Wang & Qi, 2009) are among the methods proposed for the RCPSP. The methods proposed for the DTCTP include Siemens approximation method (Siemens, 1971), genetic algorithms (Fallah-Mehdipour, Haddad, Rezapour Tabari, & Mariño, 2012; Feng, Liu, & Burns, 1997; Kandil & El-Rayes, 2006; Sonmez & Bettemir, 2012; Zheng, Ng, & Kumaraswamy, 2005), ant colony optimization (Afshar, Ziaraty, Kaveh, & Sharifi, 2009; Ng & Zhang, 2008; Xiong & KUANG, 2008), particle swarm optimization (Bettemir, 2009; Fallah-Mehdipour et al., 2012; Yang, 2007), shuffled frog leaping (Elbeltagi, Hegazy, & Grierson, 2007), and tabu-search (Vanhoucke & Debels, 2007).

The majority of the research on the time/cost trade-off problem did not consider resource constraints and few studies focused on the resource constrained time-cost trade-off problem which combines the time/cost trade-off problem with the RCPSP. In an early attempt to integrate resource constraints with the time-cost trade-off problem Chua, Chan, and Govindan (1997) proposed a genetic algorithm (GA) based model. Leu and Yang (1999) presented a multi-criteria genetic algorithm for the resource constrained discrete time-cost trade-off problem (RCDTCTP). Ahn and Erenguc (1998) developed a multi-pass heuristic procedure for the resource constrained time-cost trade-off problem. Chen and Weng (2009) adopted a GA-based time-cost trade-off analysis for considering resource constrained scheduling along with time-cost trade-off. Wuliang and Chengen (2009) developed a GA for the RCDTCTP. Hegazy and Menesi (2012) presented a heuristic method which crashes the lowest-cost critical activities that are determined by the critical path method, and resolves any resource over allocation by imposing start-delay values to the activities to meet both project deadlines and resource limits. In a recent study, Menesi et al. (2013) proposed a constraint programming model for the RCDTCTP and implemented the model for large scale projects including up to 2000 activities.

Despite the large amount of concentrated research on designing heuristics and meta-heuristics for the RCPSP and DTCTP, very few of the proposed methods can be applied on real-life construction projects which typically encompass more than 300 activities (Liberatore, Pollack-Johnson, & Smith, 2001). Besides, the limited methods that are capable of solving large-scale problems usually require a significant amount of computation time to achieve high quality solutions. The parallel genetic algorithm of Kandil and El-Rayes (2006) required 136.5 hours on a single processor, and 19.7 hours over a cluster of 20 processor to obtain the Pareto front for a DTCTP including 720 activities. Meta-heuristics of Bettemir (2009) were able to achieve a two percent deviation from the optimal in 73 minutes for DTCTP instances including 630 activities. The heuristic of Hegazy and Menesi (2012) required 32 minutes for a RCDTCTP including 360 activities (Menesi et al., 2013). The constraint programming model of Menesi et al. (2013) achieved a solution with 6.39% deviation from the upper bound (best known solution) in 120 minutes. Hence, for the time-cost trade-off problem there is a significant gap between the literature and the requirements of real-life construction project management. In this thesis, a new heuristic method will be presented to achieve high quality solutions for the RCDTCTP in short amount of computational time.

CHAPTER 3

A MEMETIC ALGORITHM FOR THE RESOURCE LEVELING PROBLEM

As described in the previous chapter, there are several meta-heuristic algorithms proposed for solving the resource leveling problem (RLP). However, very few published studies have focused on incorporating the individual learning as a separate process for local refinement to design an effective algorithm for the RLP. Genetic algorithm is suitable for implementing multiple directional search in parallel architecture and can capture critical components of the past good solutions, however, sole GAs often lacks sufficient search intensification capability (Holland, 1975). Memetic algorithms (MAs) were proposed to combine strengths of hierarchical population search methods with the intensification capabilities of local search procedures (Moscato & Norman, 1992). MAs offer a new problem oriented algorithmic design perspective (Neri, Cotta, & Moscato, 2012).

In this chapter, details of a memetic algorithm with simulated annealing (MASA), which has been developed for tackling RLP, are described. MASA is designed to achieve an efficient optimization strategy for RLP, using any kind of known metrics as the objective function by combining complementary strengths of genetic algorithms, a shifting heuristic, and simulated annealing. The performance of this algorithm is compared with the performance of common leveling heuristics of two popular commercial project management software, and state-of-art leveling heuristic and meta-heuristics methods. The solutions for the known problem sets in literature are also obtained by MASA, using resource idle day and maximum resource demand (RID-MRD) objective function metrics for the first time in the literature, in order to offer benchmark solutions for these metrics.

For small instances up to 30 activities, mixed-integer linear models are presented for two leveling metrics including sum of squares of daily resource requirement (SSRR) and sum of absolute difference between daily resource requirement and average resource consumption (ADIF), to provide a basis for performance evaluation. The computational results validate effectiveness of the proposed algorithm and illustrate limitations of the popular commercial project management software for resource leveling.

3.1. Chromosome Representation of MASA

In a GA, candidate solutions to an optimization problem are represented by individuals. The solutions are encoded to GA by using chromosomes which are a string of parameters called genes. In MASA, the genes are composed of real numbers between 0 and 1, representing start time alternatives of non-critical activities. A gene value close to 0 corresponds to a start time alternative within the early start time, while a gene value close to 1 corresponds to a start time alternative within the late start time. The leveling example of Son and Skibniewski (1999) is used to illustrate the chromosome representation along with the encoding and decoding scheme designed for MASA. The case example includes six non-critical activities as shown in Figure 3.1. An arbitrary chromosome representation for the example is given in Figure 3.2.

MASA schedules the activities in the precedence feasible activities list in ascending activity ID. The initial precedence feasible activities list includes activities 1 and 4. Activity-1 has a smaller activity ID, hence this activity is scheduled first. Activity-1 has a duration (D) of 8 days, and a resource requirement (RR) of 2. In the initial schedule, early start time (ES) of Activity-1 is day 0, and the late start time (LS) of this activity is day 7. The start time alternatives for Activity-1 is eight, as this activity has a total float (TF) of seven days. Eight intervals are constructed between zero and one to determine the start time of Activity-1. Thus, the interval length is 0.125 (1/8). Since Activity-1 is the first non-critical activity the value of first gene is used to determine the start time alternative for this activity. The value 0.240 corresponds to the second interval and Activity-1 is scheduled to start at the second start time alternative. Hence, the scheduled start time (SS) of Activity-1 is determined as day 1 and the scheduled finish time (SF) of Activity-1 is determined as day 9. Once Activity-1 is scheduled, it is removed from the precedence feasible activities list, Activity-2 is added to the list, and early start times, late start times, and total floats of all unscheduled non-critical activities are updated.

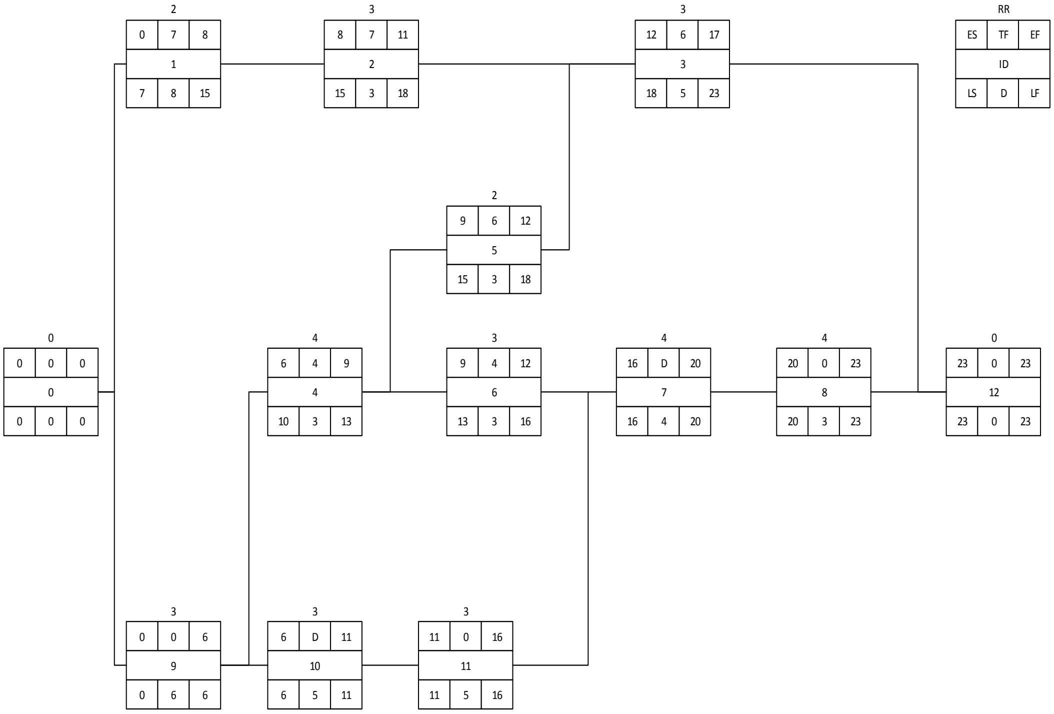


Figure 3.1. Example Network of Son and Skibniewski (1999)

0.240	0.631	0.719	0.853	0.402	0.363

Figure 3.2. Chromosome Representation of MASA for Example Problem

The procedure is continued with the next activity in the list until all of the activities are scheduled. The resulting schedule has an SSRR value of 985, as shown in Figure 3.3.

							Days																						
ID	ES	LS	SS	SF	D	RR	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	0	5	1	9	8	2		2	2	2	2	2	2	2	2														
2	9	15	13	16	3	3														3	3	3							
3	17	18	18	23	5	3																			3	3	3	3	3
4	6	10	10	13	3	4											4	4	4										
5	13	15	14	17	3	2															2	2	2						
6	13	13	13	16	3	3														3	3	3							
7	16	16	16	20	4	4																	4	4	4	4			
8	20	20	20	23	3	4																					4	4	4
9	0	0	0	6	6	3	3	3	3	3	3	3																	
10	6	6	6	11	5	3							3	3	3	3	3												
11	11	11	11	16	5	3												3	3	3	3	3							
			Daily RR				3	5	5	5	5	5	5	5	5	3	7	7	7	9	11	11	6	4	7	7	7	7	7
			Squared Daily RR				9	25	25	25	25	25	25	25	25	9	49	49	49	81	121	121	36	16	49	49	49	49	49
			SSRR	985																									

Figure 3.3. Decoding of the Chromosome Representation for Example Problem by MASA

3.2. Heuristic Improvement in MASA

A shifting heuristic is integrated to MASA for individual learning and local refinement. The shifting heuristic attempts to improve the resource histogram of a given schedule by searching the start time alternatives of non-critical activities one by one, in ascending activity ID order, without changing the start times of remaining activities. In order to illustrate the heuristic improvement, the shifting heuristic is applied to the schedule of Figure 3.3. Heuristic improvement for the first gene is explored first by evaluating the SSRR values of all possible start time alternatives for Activity-1, without changing the start times of remaining activities. Starting Activity-1 at days 0, 2, 3, 4, or 5 does not decrease the SSRR value hence Activity-1 is not shifted. However, starting Activity-2 at day 15 instead of day 13 improves the SSRR value to 961. Hence the start time of Activity-2 is shifted to day 15, and early start times, late start times, and total floats of all non-critical activities are updated. The procedure is applied to all remaining non-critical activities and the SSRR value is improved to 957 as shown in Figure 3.4. The start times of the improved schedule are encoded by using mid points of the corresponding intervals. For example, in the improved schedule Activity-2 has seven start time alternatives and the interval width for this activity is 0.143 (1/7). Hence, the latest start time alternative of Activity-2 corresponds to the interval 0.857-1.000, and the midpoint of the interval is 0.929. The chromosome representation of the improved schedule is given in Figure 3.5.

							Days																						
	ES	LS	SS	SF	D	R	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	0	7	1	9	8	2		2	2	2	2	2	2	2	2														
2	9	15	15	18	3	3																3	3	3					
3	18	18	18	23	5	3																			3	3	3	3	3
4	6	10	10	13	3	4											4	4	4										
5	13	15	13	16	3	2														2	2	2							
6	13	13	13	16	3	3														3	3	3							
7	16	16	16	20	4	4																	4	4	4	4			
8	20	20	20	23	3	4																					4	4	4
9	0	0	0	6	6	3	3	3	3	3	3	3																	
10	6	6	6	11	5	3							3	3	3	3	3												
11	11	11	11	16	5	3												3	3	3	3	3							
			Daily RR				3	5	5	5	5	5	5	5	5	3	7	7	7	8	8	11	7	7	7	7	7	7	7
			Squared Daily RR				9	25	25	25	25	25	25	25	25	9	49	49	49	64	64	121	49	49	49	49	49	49	49
			SSRR	957																									

Figure 3.4. Improved Schedule of Example Problem by Shifting Heuristic of MASA

0.188	0.929	0.500	0.900	0.165	0.500

Figure 3.5. Encoding of the Improved Schedule by MASA, for Example Problem

3.3. Crossover, Mutation, and Simulated Annealing in MASA

New individuals are introduced by using crossover and mutation operators. One point crossover is performed for problems including ten or less activities. For problems including more than ten activities, two point crossover is performed. The mutation operator of MASA changes a gene value of a selected chromosome with a random real number between 0 and 1. SA is integrated to MASA to perform mutations with an adaptive mutation rate based on a cooling schedule. MASA executes a mutation that leads to an individual with a worse fitness evaluation function value if:

$$r \leq e^{\frac{\left(f - f^{'}\right)}{f} * \frac{\mathrm{B}}{t}} \qquad (3.1)$$

where;

r is a random real number between 0 and 1;

f is the fitness value before mutation;

$f^{'}$ is the fitness value after the mutation;

B is the Boltzmann constant; and

t is the temperature.

The main purpose of the adaptive mutation rate strategy is to prevent premature convergence by controlling the search process more efficiently and to relax the parameter dependence of GA to some extent. At initial search stages mutations leading to a worse fitness value are allowed to avoid being trapped in certain solutions. At later stages, by decreasing the temperature based on a cooling schedule, fewer mutations leading to a worse fitness value are allowed for achieving fine tuning. MASA is evolved toward better solutions by elitist roulette wheel selection method. The flow chart of MASA is given in Figure 3.6.

3.4. Excel Interface of MASA

MASA was implemented using C# and compiled within Visual Studio 2010. A Microsoft Excel interface was integrated into MASA, in order to obtain a simplified tool for input the problems, and to enable data exchange with the commercial project management software programs. The input screen of the interface for the case example is shown in Figure 3.7 and its output screen is demonstrated in Figure 3.8.

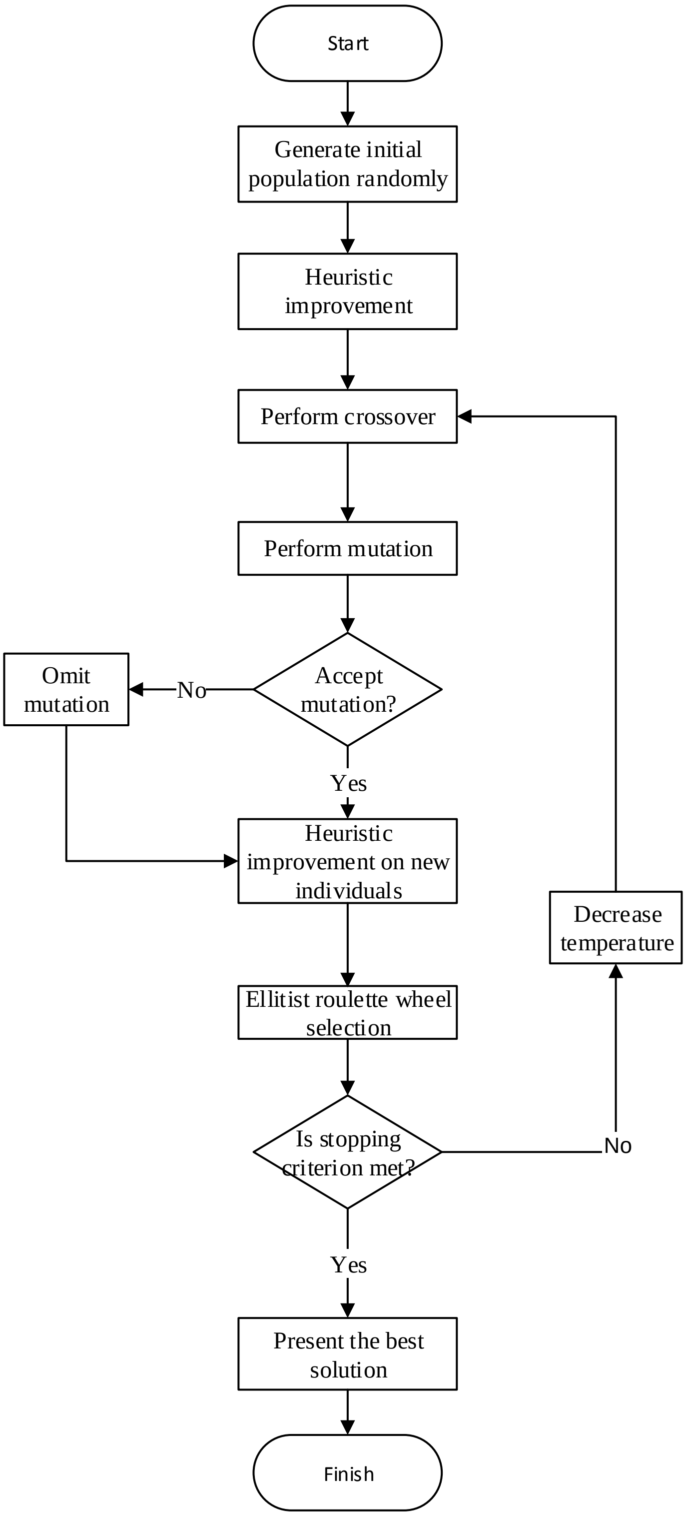
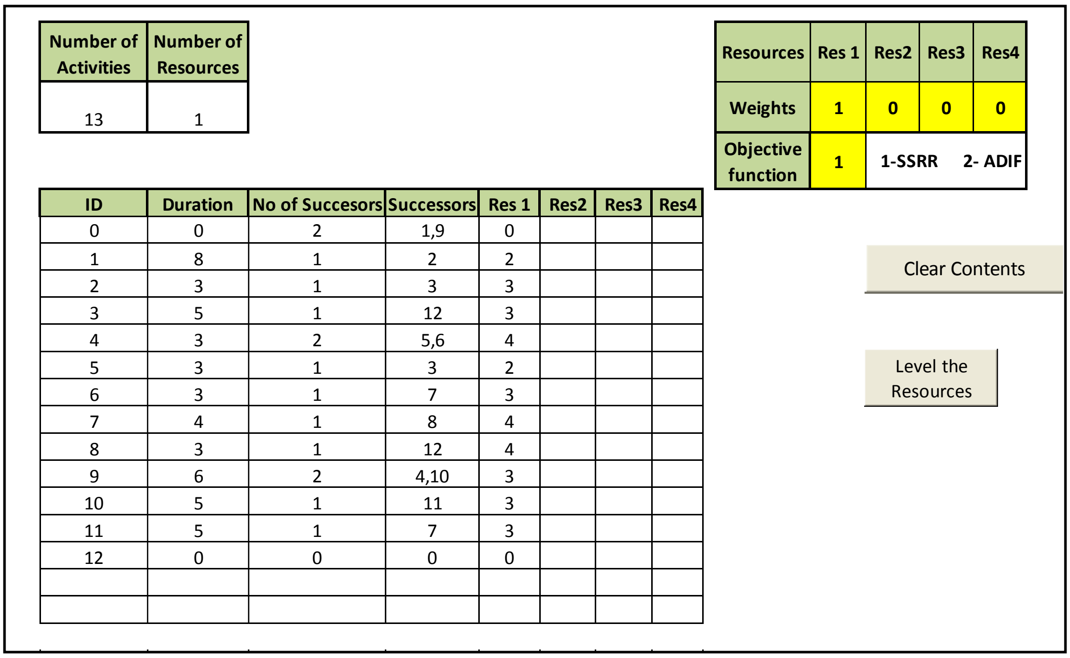


Figure 3.6. Flowchart of MASA

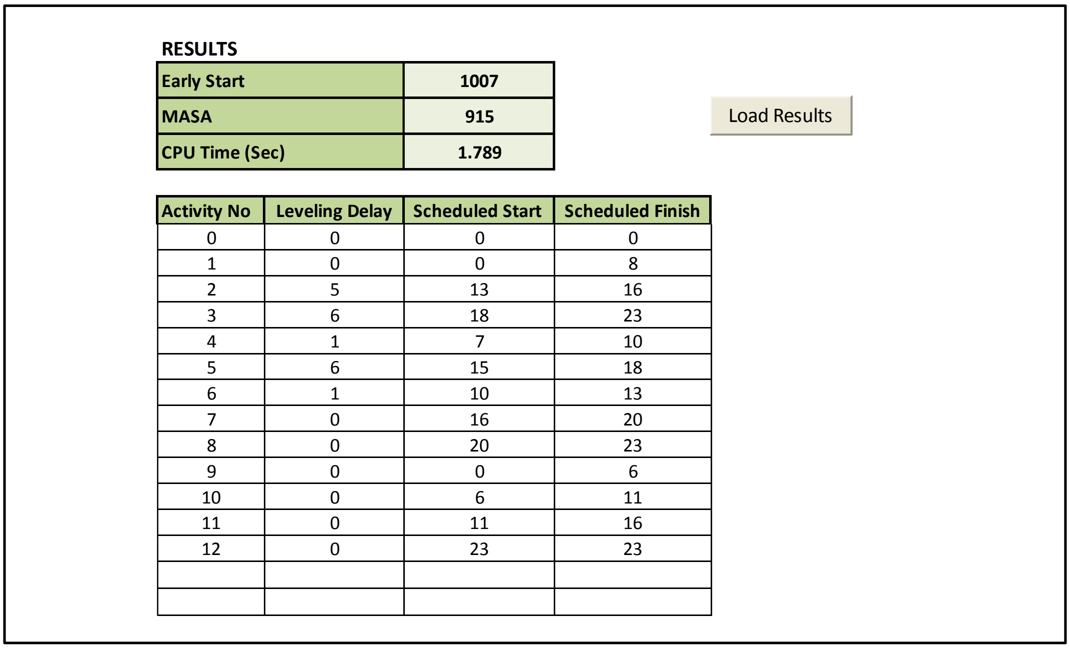


Number of Activities	Number of Resources
13	1

Resources	Res 1	Res2	Res3	Res4
Weights	1	0	0	0
Objective function	1	1-SSRR	2- ADIF	

ID	Duration	No of Succesors	Successors	Res 1	Res2	Res3	Res4
0	0	2	1,9	0			
1	8	1	2	2			
2	3	1	3	3			
3	5	1	12	3			
4	3	2	5,6	4			
5	3	1	3	2			
6	3	1	7	3			
7	4	1	8	4			
8	3	1	12	4			
9	6	2	4,10	3			
10	5	1	11	3			
11	5	1	7	3			
12	0	0	0	0			

Figure 3.7. Excel Interface Input Screen of MASA



RESULTS

Early Start	1007
MASA	915
CPU Time (Sec)	1.789

Activity No	Leveling Delay	Scheduled Start	Scheduled Finish
0	0	0	0
1	0	0	8
2	5	13	16
3	6	18	23
4	1	7	10
5	6	15	18
6	1	10	13
7	0	16	20
8	0	20	23
9	0	0	6
10	0	6	11
11	0	11	16
12	0	23	23

Figure 3.8. Excel Interface Input Screen of MASA

3.5. Mixed-integer Linear Models

Two mixed-integer linear models are presented in this section for minimizing ADIF and SSRR to evaluate performance of MASA. The models are an extension of the previous model presented by Iranagh et al. (2013) for the combined resource idle days and maximum resource demand metric.

3.5.1. Inputs of Models

The project is considered in which $I = \{1, 2, \ldots, I\}$ is the set of activities where $i = 1$ is the start activity and $i = \mathrm{I}$ is the finish activity of the project. Set $T = \{0, 1, \ldots, T\}$ represents the times (days) within the project duration that, $t = 0$ stands for the start day and $t = T$ refers to the last or finishing day of the project. Similarly, $R = \{1, 2, \ldots, R\}$ defines the set of resources, and finally, $N = \{0, 1, \ldots, N\}$ denotes the set of total daily demand for each resource by the activities.

Parameters of the model are as follows:

EST_i and LST_i represent the early start, and late start times of i^{th} activity;
d_i is the duration of i^{th} activity;
$r_{i,r}$ is the demand for resource r by i^{th} activity;
w_r is the weight of resource r;
D is the total project duration; and
pi, j is the relationship between activities i and j where:

$$pi, j : \begin{cases} 1 & \text{if activity } j \text{ should be finished before activity } i; \\ 0 & o/w. \end{cases}$$

The following variables are then defined for the model:
z_1 is the weighted sum of absolute deviations from the average resource demands (ADIF);
z_2 is the weighted sum squares of resource demands for all time periods (SSRR).

f_i is the start day of i^{th} activity;

$u_{t,r}$ is the daily demand of resource r at day t;

mxu_r is the maximum daily demand of the resource r;

$mx1u_{t,r}$ is the maximum daily demand for resource r before day t;

$mx2u_{t,r}$ is the maximum daily demand for resource r after day t;

$mnu_{t,r}$ is the smallest of of $mx1u_{t,r}$ or $mx2u_{t,r}$ for each day and for each resource; and finally

$\lambda_{n,t,r}$, $\varphi_{t,i}$ and $\sigma_{t,i}$ are defined as follows:

$$\lambda_{n,t,r} : \begin{cases} 1 & \text{if demand for resource } r \text{ at time (day) } t \text{ is equal to } n; \\ 0 & o/w. \end{cases}$$

$$\varphi_{t,i} : \begin{cases} 1 & \text{if activity } i \text{ is under progress at time (day) } t; \\ 0 & o/w. \end{cases}$$

$$\sigma_{t,i} : \begin{cases} 1 & \text{if activity } i \text{ has started at time (day) } t; \\ 0 & o/w. \end{cases}$$

3.5.2. Models Construction

The first model as shown in Eq.(3.2), minimizes the absolute deviation between the resource requirement and a targeted uniform resource level (ADIF). The objective of the second model is to minimize the sum of squares of resource requirements (SSRR) for all time periods as shown in Eq.(3.3).

$$\min z_1 = \sum_t \sum_r w_r \,| u_{t,r} - a_{t,r} | \tag{3.2}$$

$$\min z_2 = \sum_t \sum_r w_r u_{t,r}{}^2 \tag{3.3}$$

Since both of the metrics are not linear, the metrics are expressed in terms of the linear models. The ADIF leveling metric is expressed as a linear objective function in Eq.(3.4), and Eqs. (3.5) and (3.6) describe related constraints. The constraint given in Eq.(3.5) expresses the $u_{t,r}$ - $a_{t,r}$ term as difference of two non-negative integer variables as the absolute value function is not linear.

$$\min z_1 = \sum_t \sum_r w_r (x_{t,r} + y_{t,r}) \tag{3.4}$$

$$u_{t,r} - a_{t,r} = x_{t,r} - y_{t,r} \qquad \forall t \in T, \forall r \in R \tag{3.5}$$

$$x_{t,r}, y_{t,r} \in Z_0 \qquad \forall t \in T, \forall r \in R \tag{3.6}$$

Accordingly, the objective function given in Eq.(3.7) minimizes the weighted SSRR for all time periods, and Eqs. (3.8) and (3.9) describe related constraints to linearize the SSRR model. Eq.(3.8) determines the sum of resource requirements, and Eq.(3.9) determines the SSRR for all time periods. Eq.(3.10) ensures that the sum of resource requirement for resource *r* can take a unique value.

$$\min z_2 = \sum_t \sum_r w_r v_{t,r} \tag{3.7}$$

$$u_{t,r} = \sum_n n \lambda_{n,t,r} \qquad \forall t \in T, \forall r \in R \tag{3.8}$$

$$v_{t,r} = \sum_n n^2 \lambda_{n,t,r} \qquad \forall t \in T, \forall r \in R \tag{3.9}$$

$$\sum_n \lambda_{n,t,r} = 1 \qquad \forall t \in T, \forall r \in R \tag{3.10}$$

$$v_{t,r} \in Z_0 \qquad \forall t \in T, \forall r \in R \tag{3.11}$$

$$\lambda_{n,t,r} \in \{0,1\} \qquad \forall n \in N, \forall t \in T, \forall r \in R \tag{3.12}$$

3.5.3. Common Scheduling Constraints of the Models

The scheduling constraints which are common for both models, are as follow:

$$\sum_i r_{i,r} \varphi_{t,i} = u_{t,r} \qquad \forall t \in T, \forall r \in R \tag{3.13}$$

$$p_{i,j} f_i \geq p_{i,j} \left(f_j + d_j \right) \qquad \forall i, j \in I, i \neq j \tag{3.14}$$

$$\sum_{EST_i \leq t \leq LST_i} t \sigma_{t,i} = f_i \qquad \forall i \in I \tag{3.15}$$

$$\sum_{EST_i \le t \le LST_i} \sigma_{t,i} = 1 \qquad \forall i \in I \tag{3.16}$$

$$\varphi_{t,i} = \sum_{t=\max(EST_i, t-d_i+1)}^{\min(LST_i, t)} \sigma_{t,i} \qquad \forall t \in T, \forall i \in I, EST_i \le t \le LST_i + d_i - 1 \tag{3.17}$$

$$\varphi_{t,i} = 0 \qquad \forall t \in T, \forall i \in I, t < EST_i \tag{3.18}$$

$$\varphi_{t,i} = 0 \quad \forall t \in T, \forall i \in I, t > LST_i + d_i - 1 \tag{3.19}$$

$$f_1 = 0 \tag{3.20}$$

$$f_I \le D \tag{3.21}$$

$$\sigma_{0,1} = 1 \tag{3.22}$$

$$u_{t,r} \in Z_0 \quad \forall t \in T, \forall r \in R \tag{3.23}$$

$$f_i \in Z_0 \quad \forall i \in I \tag{3.24}$$

$$\varphi_{t,i} \in \{0,1\} \forall t \in T, \forall i \in I \tag{3.25}$$

$$\sigma_{t,i} \in \{0,1\} \forall t \in T, \forall i \in I \tag{3.26}$$

Eq.(3.13) defines the daily resource demand for resource type r and ensures that the activities use the resources only in days when they are active. Eq.(3.14) ensures the precedence relationships between the activities are satisfied. Eq.(3.15) determines activity start day. Eq.(3.16) ensures that activities can start only in a day between their early start and late start times. Eq.(3.17) determines the days that activities are active and ensures that the days that activities are active are consecutive. Eqs. (3.18) and (3.19) ensure that the activities are active only between early start and late finish days. First and last activities are dummy activities that identify the start and finish dates of the project. Eqs. (3.20) and (3.22) ensure that the first activity starts at day 0, and Eq.(3.21) ensures that all activities are completed before the dummy finish activity. Variables $u_{t,r}$ and f_i are non-negative integers, and $\varphi_{t,i}$ and $\sigma_{t,i}$ are binary variables.

3.6. Computational Experiments of MASA

In this section the performance of proposed MASA is compared with the performance of the state of the art heuristic and meta-heuristics methods. All of the tests were carried out on a computer with a 3.00 GHz Core 2 Duo Processor E8400 Intel CPU. A total of 1443 test instances including up to 120 activities and four resources, mainly from PSPLIB (Kolisch & Sprecher, 1997) were used in computational experiments. Performance analyzes of MASA using PSPLIB instances were conducted for both SSRR and RID-MRD objective function metrics.

3.6.1. Single Resource Case Examples

The majority of the leveling case examples in the literature includes a single resource and a few activities. Performance of MASA was evaluated initially for the two case examples presented with Son and Skibniewski (1999), and one case example presented with El-Rayes and Jun (2009). The stopping criterion for MASA was set as 50,000 schedules (Rainer Kolisch & Hartmann, 2006) for the single resource case examples. Optimal solutions of the case examples were obtained by using the models presented in Eq.(3.2), through Eq.(3.26). The targeted demand ($u_{t,r}$) was determined by rounding the average resource demand using the floor function. Results of MASA are given in Table 3.1. MASA was able to find the optimal result for all of the single case examples within 0.4 seconds.

Table 3.1. Results of MASA for Single Resource Case Examples

Source	No of Activities	Metric	Optimal	MASA	Time (Sec.)
Son and Skibniewski (1999)	13	SSRR	915	915	0.4
Son and Skibniewski (1999)	11	SSRR	6225	6225	0.2
El-Rayes and Jun (2009)	20	SSRR	3059	3059	0.3
El-Rayes and Jun (2009)	20	ADIF	90	90	0.3

3.6.2. Comparison of MASA with Microsoft Project and Primavera

Primavera and Microsoft Project are the most commonly used software for planning and management of construction projects (Liberatore et al., 2001). Resource leveling can be performed in Primavera and Microsoft Project by setting targets for the resource demands. Despite the importance of leveling in practice, very few studies in the literature evaluated the performance of project management software for the RLP. In this section the performance of MASA is compared with the performance of nine priority based leveling heuristics available in Microsoft Project 2010 and Primavera 6.7. The heuristics included Standard (STD) heuristics of Microsoft Project (MSP) 2010, and ID-Ascending (IDA), ID-Descending (IDD), Total Float-Ascending (TFA), Total Float-Descending (TFD), Early Start-Ascending (ESA), Early Start-Descending (ESD) Late Finish-Ascending (LFA), and Late Finish-Descending (LFD) heuristics of Primavera 6.7.

15 standard instances with 30 activities (J30), 15 standard instances with 60 activities (J60), and 15 standard instances with 120 activities (J120) were selected randomly from the project scheduling problem library (PSPLIB) of Kolisch and Sprecher (1997). All problem instances included four resource types. Details of the test instances are described in Kolisch and Sprecher (1997). In comparisons, ADIF leveling metric was used. The targeted demands for resources were determined by rounding the average resource demand for each resource using the floor function. The weights of all four resources were taken as equal. All of the selected J30 test instances were solved to optimality within a computation time limit of five hours by using the standard solver CPLEX and the model presented for ADIF in previous chapter.

The percent deviation from the upper bound (optimal or best known solution) is used to evaluate the performance of MASA. The percent deviation from the upper bound (PD) is calculated as Eq.(3.27).

$$\text{PD} = \frac{\textit{Solution - Upper Bound}}{\textit{Upper Bound}} \text{ x } 100 \tag{3.27}$$

where;

$Solution$ is the minimum objective function (SSRR) value obtained; and

$UpperBound$ is optimal or the best known solution for the problem.

Hence, in comparisons the average percentage deviation (APD) from the upper bounds for each problems set is used for performance evaluation. The stopping criterion for MASA was set as 500,000 schedules. The performance comparison results are presented in Table 3.2.

The APD of MASA from the optimal solutions was 0.5 for the J30 instances. Among the ten methods evaluated, MASA determined the best solution for 44 test instances. Total Float-Ascending heuristic obtained the best solution for the remaining instance. The average percentage deviations of MASA from the upper bounds were 0.0 and 0.1 for the J60 and J120 instances, respectively. The average CPU time for all the instances was 9.6 seconds. MASA produced very good results within reasonable computing time. The nine priority based leveling heuristics performed very poorly in comparison to MASA. Among the nine heuristics tested, Total Float-Ascending and Late Finish-Ascending methods performed relatively better. The average percentage deviations of these methods for all instances were 46.4 and 47.0 respectively, whereas the APD of MASA for all instances was 0.2. The performance gap between MASA and nine priority based leveling heuristics revealed the limitations of the commercial project management software for resource leveling.

Table 3.2. Comparison of MASA with Microsoft Project and Primavera

Instance Sets	MSP (2013)	Primavera P6 (8.4)								MASA	
	STD	IDA	IDD	TFA	TFD	ESA	ESD	LFA	LFD	NA	Time (Sec.)
J30 (15)	75.4	42.9	48.9	48.0	43.6	45.0	41.8	45.9	44.4	0.5	4.7
J60 (15)	89.3	59.1	62.0	49.4	54.0	61.3	52.4	46.3	57.5	0.0	8.3
J120 (15)	100.3	58.4	53.7	41.7	61.1	69.5	56.3	48.9	64.1	0.1	15.8
Average:	88.3	53.5	54.9	46.4	52.9	58.6	50.1	47.0	55.3	0.2	9.6

3.6.3. Performance Analyzes of MASA with SSRR Objective Function Using PSPLIB Instances

In a recent study Ponz-Tienda et al. (2013) presented an adaptive GA (AGA) for RLP. Ponz-Tienda et al. (2013) evaluated the performance of AGA for the SSRR metric by using 480 J30 instances, 480 J60 instances, and 480 J120 instances. In Table 3.3 the performance of MASA is compared with the performance of AGA with SSRR Objective Function Metric. The modified version of the well-known Burgess shifting heuristic (Burgess & Killebrew, 1962) is also included in the comparisons. The modified Burgess algorithm (Burgess2) executes the standard Burgess method for several randomly selected activity ID orders until a stopping criterion is met, and reports the best SSRR value achieved. The APD values given in Table 3.3 are the average percentage deviations from the current best solutions for the SSRR metric. In computational experiments the weights of all four resources were taken as equal. J30 test instances were solved within a computation time limit of five hours by using the standard solver CPLEX and the model presented for SSRR. Within the specified computation time limit 475 J30 instances were solved to optimality. In computational analysis, the result of MASA at the end of 500,000 schedules was reported. The CPU time of MASA for each problem was used as the stopping criterion for Burgess2 heuristic.

Table 3.3 presents the summary of the computational results. The complete results for all instances and optimal solutions for J30 instances are illustrated in Appendix A. The computational results indicate that with an APD of 0.2 for J30 instances, MASA was able to obtain high quality solutions which were either optimal or very close to the optimal. Out of 475 J30 instances with optimal solutions MASA was able to obtain the optimal for 232 instances. AGA with an APD of 0.7 was the second best method for J30 instances, and was able to determine the optimal for 76 instances. The computational experiments for AGA was performed on a computer with a 3.6 GHz Intel Core i7 processor. The average computing time of MASA for

J30 instances was reported as 15 seconds (Ponz-Tienda et al., 2013). For J30 instances the average CPU time of MASA on a computer with a 3.00 GHz Core 2 Duo Processor E8400 Intel CPU was 12.6 seconds. MASA was able to obtain better solutions compared to AGA within a shorter computing time. Among the three methods evaluated, Burgess2 ranked last for J30 instances, and achieved an APD of 3.6.

MASA obtained the best result for majority of J60 and J120 instances, and achieved an APD of 0.0 for J60, and 0.1 for J120 instances. AGA had an APD of 2.3 for J60, and 3.7 for J120 instances. The APD of Burgess2 for J60 and J120 instances were 3.1 and 2.1, respectively. MASA performed significantly better than MASA and Burgess2 for all instance sets. The average CPU time of MASA for all instances was 19.5 seconds. The computational experiment results for J30, J60 and J120 instances confirmed the effectiveness of MASA.

Table 3.3. Computational Results of MASA with SSRR Objective Function for PSPLIB Instances

Instance Sets	AGA (Ponz-Tienda et al., 2013)			Burgess2			MASA		
	APD (%)	No of Optimal	Time (S.)	APD (%)	No of Optimal	Time (S.)	APD (%)	No of Optimal	Time (S.)
J30 (480)	0.7	76	15	3.6	14	12.6	0.2	232	12.6
J60 (480)	2.3	NA	NA	3.1	NA	18.3	0.0	NA	18.3
J120 (480)	3.7	NA	NA	2.1	NA	27.6	0.1	NA	27.6
Average:	2.2			2.9		19.5	0.1		19.5

3.6.4. Performance Analyzes of MASA with RID-MRD Objective Function Metric Using PSPLIB Instances

El-Rayes and Jun (2009) showed that the combined metric RID-MRD, which minimizes the resource fluctuations and peak resource simultaneously, are capable of outperforming existing metrics in eliminating undesirable resource fluctuations and resource idle time. Despite the fact that the joint RID-MRD provides a metric of practical significance, there are very limited study in the literature focusing this metric. In this part the performance of MASA with the RID-MRD objective function was compared with the performance of aforementioned Burgess2 method over the J30, J60 and J120 problem sets, since there was not any available comparable other method in the literature. The results obtained by both MASA and Burgess2 are provided in Appendix B in order to offer a benchmark for the future studies. Table 3.4 presents the summary of the computational results for both the methods. Weights of all resource types together with the weights for both RID and MRD metrics were all defined as 1. According to the computational results, MASA outperforms Burgess2 by a huge margin with the overall APDs of 1.1 to 22.3. The stopping criteria for MASA was determined as the schedule number of 500,000. The computational time of each problem in MASA was used as the stopping criteria for the same problem.

Table 3.4. Computational Results of MASA with RID-MRD Objective Function Metric for PSPLIB Instances

Instance Sets	Burgess2		MASA	
	APD (%)	Time (Sec.)	APD (%)	Time (Sec.)
J30 (480)	23.0	13.9	0.8	13.9
J60 (480)	22.6	19.3	1.2	19.3
J120 (480)	21.4	29.5	1.2	29.5
Average:	22.3	20.9	1.1	20.9

CHAPTER 4

A QUASISTABLE SCHEDULE SEARCH HYBRID GENETIC ALGORITHM FOR RESOURCE LEVELING OF LARGE-SCALED CONSTRUCTION PROJECTS

Although vast amount of the research has been done on designing heuristics and meta-heuristics for the resource leveling problem (RLP), very few of the suggested methods is practical enough to implement to real-life-size construction projects. In addition, a few methods that are proficient of solving large-scale problems usually require a significant amount of computation time to achieve high quality solutions. The iterated greedy method of Ballestín et al. (2007) as one of the most capable and fast methods for large problems required 459.70 seconds to obtain a solution for RLP including 1000 activities and five resources. The analysis has been done on a personal computer with 1.4 GHz processor and 512 MB RAM and the stopping criteria has been defined as 1000 loops of iterations. Ballestín et al. (2007) has considered the networks with temporal constraints and modeled the production planning problem like a RLP and compared their proposed model with other methods for resource leveling of networks with temporal constraints.

Regarding to the RLP for critical path method (CPM) networks, the hybrid genetic algorithm of Doulabi et al. (2011) required an average period of 14502 seconds to find a solution for their generated instances including 5000 activities and up to nine resources. The memetic algorithm with simulated annealing (MASA) which is presented within this thesis in the previous chapter shown to be capable enough to unravel the medium size resource leveling problems including few numbers of resources within a reasonable computational time in comparison to the other state-of-art methods presented in the related literature. Another major contribution of MASA is to provide the benchmark solutions for the known problem sets of literature for all types of objective function metrics. Nevertheless, the necessity for a faster method to be able to solve RLPs of the real-life-size construction projects in practice cannot be ignored.

For this purpose, a hybrid genetic algorithm is developed for RLP as the second method in this study. The proposed algorithm which is called quasistable hybrid genetic algorithm (QHGA) limits the searching space only to quasistable schedules. QHGA is consisted of two priority based heuristics including constructive and local improvement modules incorporated with a genetic algorithm scheme that determines and modifies the priorities of the activities and their floats. The main objective of QHGA is to prepare a model that can achieve high quality solutions in a short period of computational time for the large-scale RLPs of construction projects. Both metrics of the sum of squares of daily resource requirement (SSRR) and total overloaded amount from average resource consumptions (OVLD) are applicable for QHGA as the objective function. This chapter is devoted to describe the details of QHGA.

4.1. Quasistable Schedules

Ballestín et al. (2007) and Neumann, Nübel, and Schwindt (2000) proved that the set of schedules with optimal squares of daily resource requirement (SSRR) and overloaded amount from the target resource consumptions OVLD objective value for RLP always contains at least a quasistable schedule. For a project consisting of $n+2$ activities $\{i = 0,1,...,n+1\}$, where activities 0 and $n+1$ represent start and finish milestones, according to Ballestín et al. (2007) and Neumann et al. (2000), a feasible schedule S is called quasistable if and only if for each activity $j \in V, j \neq 0$, one of the following conditions is met:

a) activity j starts at its earliest start time

b) activity j starts at its latest start time

c) there is an activity $i \in V$ such that $S_j = S_i + d_i$

d) there is an activity $i \in V$ such that $S_j = S_i - d_j$

where;

$V = \{0,1,...,n+1\}$ is the set of all activities, and

d_i and d_j is the duration of activities i and j.

Satisfying conditions (c) or (d) means that activity j is started immediately after finish time of activity i, or finished immediately before start time of activity i.

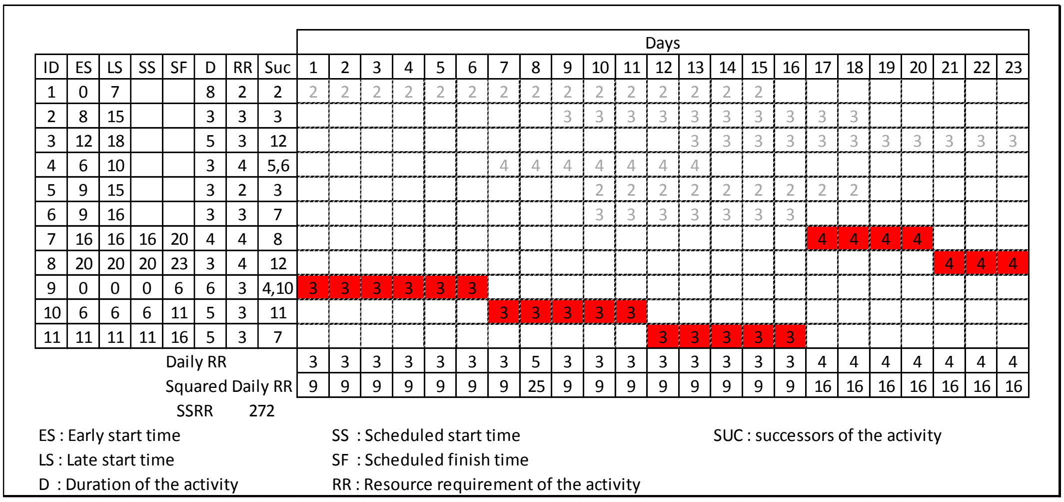
4.2. Chromosome Representation of QHGA

Like all genetic algorithm based models, in QHGA candidate solutions for the problem are represented by individuals named as chromosomes. In QHGA, a chromosome contains two strings of parameters called genes. The numbers of genes in each string is equal to the number of non-critical activities in the problem. The first string of genes represents the priorities of the non-critical activities for selection and the second string of genes denotes the start time alternatives of non-critical activities through the constructive module. The genes are composed of real numbers between 0 and 1, such that a gene value close to 1 in the first string of genes denotes the lowest priority for its corresponding activity to be select. Accordingly, a value close to 1 for the genes in the second string corresponds to a quasistable start time alternative close to the late start time or the late start time itself.

Here again, the leveling example of Son and Skibniewski (1999), shown in Figure 3.1, is used to illustrate the chromosome representation along with the encoding and decoding scheme designed for the constructive module of QHGA.

4.3. Constructive Module

The algorithm starts with generating the initial population by assigning random numbers between 0 and 1 as the gens values of each chromosome. Then for each chromosome the constructive module procedure starts with constructing the resource profile, first by scheduling the critical activities as they will be fixed in the profile for any condition. The resource usage sheet of the example problem for the critical activities is given in Figure 4.1.



								Days																						
ID	ES	LS	SS	SF	D	RR	Suc	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	0	7			8	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2								
2	8	15			3	3	3									3	3	3	3	3	3	3	3	3	3					
3	12	18			5	3	12													3	3	3	3	3	3	3	3	3	3	3
4	6	10			3	4	5,6							4	4	4	4	4	4	4										
5	9	15			3	2	3										2	2	2	2	2	2	2	2	2					
6	9	16			3	3	7										3	3	3	3	3	3	3							
7	16	16	16	20	4	4	8																	4	4	4	4			
8	20	20	20	23	3	4	12																					4	4	4
9	0	0	0	6	6	3	4,10	3	3	3	3	3	3																	
10	6	6	6	11	5	3	11							3	3	3	3	3												
11	11	11	11	16	5	3	7												3	3	3	3	3							
							Daily RR	3	3	3	3	3	3	3	5	3	3	3	3	3	3	3	3	4	4	4	4	4	4	4
							Squared Daily RR	9	9	9	9	9	9	9	25	9	9	9	9	9	9	9	9	16	16	16	16	16	16	16
						SSRR	272																							

ES : Early start time
LS : Late start time
D : Duration of the activity
SS : Scheduled start time
SF : Scheduled finish time
RR : Resource requirement of the activity
SUC : successors of the activity

Figure 4.1. The Resource Usage Sheet of the Example Problem for Critical Activities

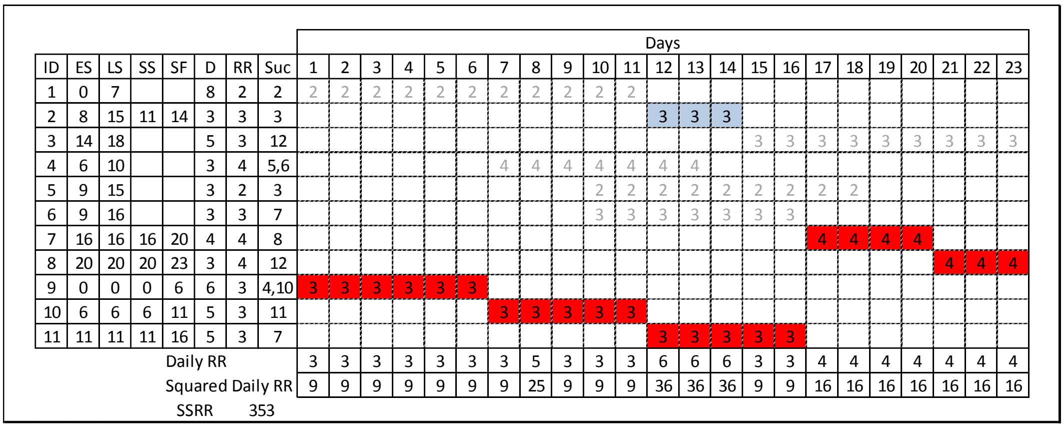
Then, a priority list is determined according to the first string gene values of the corresponding chromosome. A randomly generated chromosome for the example problem is shown in Figure 4.2.

A smaller gene value corresponds to a higher priority in chromosome decoding of QHGA. Therefore, according to the genes values of string-1, activity-2 has the highest and activity-6 has the lowest priorities for selection in constructive module. Consequently, the priority list of {2, 1, 4, 5, 3, 6}, is determined for the chromosome of Figure 4.2. Regarding to the priority list, Activity-2 is selected to be scheduled first. Activity-2 has duration (D) of 3 days, and a resource requirement (RR) of 3. In the initial schedule, early start time (ES) of Activity-2 is day 8, and the late start time (LS) of this activity is day 15. So, the start time alternatives for this activity are eight, from the day 8, up to the day 15. However, considering only the quasistable schedules, the choices as the possible start times reduces to the days 8, 11, 13 and 15. At this stage, while there are four options as the start time of Activity-2, four intervals with the length of 0.25 (1/4) are constructed between zero and one to determine the start time of it. Since the genes values of string-2 denote the start time alternatives of activities and the corresponding value of Activity-2 (0.48) is matched to second interval, the scheduled start time (SS) of Activity-2 is determined as day 11 and the scheduled finish time (SF) of Activity-2 is determined as day 14.

Acts:	1	2	3	4	5	6
String-1	0.25	0.10	0.73	0.31	0.56	0.95
String-2	0.35	0.48	0.65	0.55	0.70	0.85

Figure 4.2. Chromosome Representation of QHGA for Example Problem

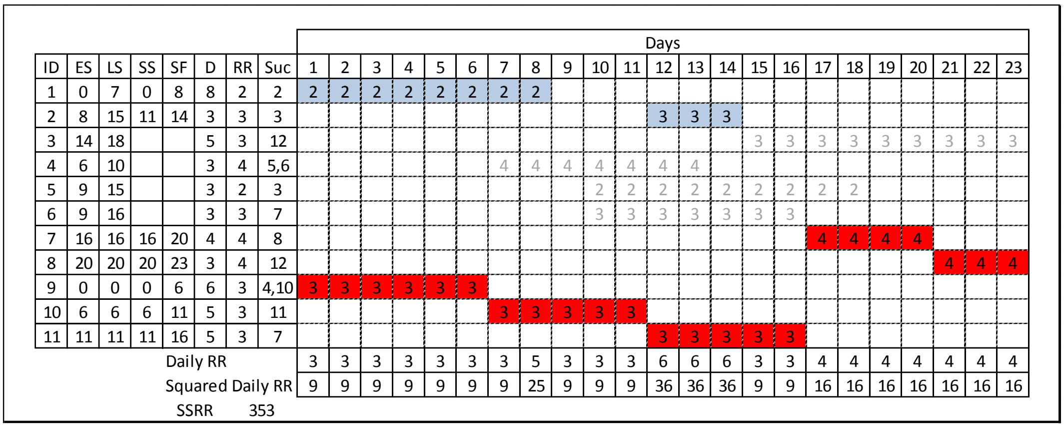
Once Activity-2 is scheduled, early start times, late start times, and total floats of all unscheduled non-critical activities are updated as shown in Figure 4.3. Then the procedure is continued with the next activity in the priority list until all of the activities are scheduled. Figure 4.4 to Figure 4.8, demonstrate the process of scheduling all the non-critical activities one by one.



								Days																						
ID	ES	LS	SS	SF	D	RR	Suc	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	0	7			8	2	2	2	2	2	2	2	2	2	2	2	2	2												
2	8	15	11	14	3	3	3												3	3	3									
3	14	18			5	3	12															3	3	3	3	3	3	3	3	3
4	6	10			3	4	5,6							4	4	4	4	4	4	4										
5	9	15			3	2	3										2	2	2	2	2	2	2	2	2					
6	9	16			3	3	7										3	3	3	3	3	3	3							
7	16	16	16	20	4	4	8																	4	4	4	4			
8	20	20	20	23	3	4	12																					4	4	4
9	0	0	0	6	6	3	4,10	3	3	3	3	3	3																	
10	6	6	6	11	5	3	11							3	3	3	3	3												
11	11	11	11	16	5	3	7												3	3	3	3	3							
				Daily RR				3	3	3	3	3	3	3	5	3	3	3	6	6	6	3	3	4	4	4	4	4	4	4
				Squared Daily RR				9	9	9	9	9	9	9	25	9	9	9	36	36	36	9	9	16	16	16	16	16	16	16
				SSRR		353																								

Figure 4.3. Resource Usage Sheet of the Example Problem after Scheduling of Activity-2

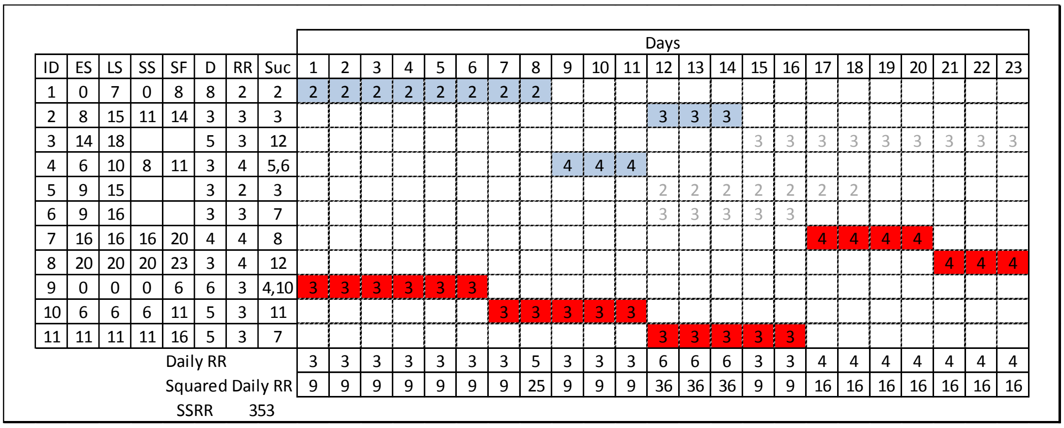
After scheduling of Activity-2, the start time options of Activity-1 for quasistable schedules are found to be as days 0 and 3. Then looking to its corresponding gene value of 0.35, the scheduled start and finish times of Activity-1 is defined as day 0 and eight respectively. Figure 4.4 shows the rescheduled resource usage sheet after placing the non-critical activities 2 and 1.



								Days																						
ID	ES	LS	SS	SF	D	RR	Suc	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	0	7	0	8	8	2	2	2	2	2	2	2	2	2	2															
2	8	15	11	14	3	3	3												3	3	3									
3	14	18			5	3	12															3	3	3	3	3	3	3	3	3
4	6	10			3	4	5,6							4	4	4	4	4	4	4										
5	9	15			3	2	3										2	2	2	2	2	2	2	2	2					
6	9	16			3	3	7										3	3	3	3	3	3	3							
7	16	16	16	20	4	4	8																	4	4	4	4			
8	20	20	20	23	3	4	12																					4	4	4
9	0	0	0	6	6	3	4,10	3	3	3	3	3	3																	
10	6	6	6	11	5	3	11							3	3	3	3	3												
11	11	11	11	16	5	3	7												3	3	3	3	3							
				Daily RR				3	3	3	3	3	3	3	5	3	3	3	6	6	6	3	3	4	4	4	4	4	4	4
				Squared Daily RR				9	9	9	9	9	9	9	25	9	9	9	36	36	36	9	9	16	16	16	16	16	16	16
				SSRR		353																								

Figure 4.4. Resource Usage Sheet of the Example Problem after Scheduling of Activity-1

For Activity-4, the start day choices for quasistable schedules are 6, 8 and 10. The gene value of 0.55 reflects the start time in second interval that is 8. Hence, the SS and SF for Activity-4 are selected as 8 and 11, which are shown in Figure 4.5.

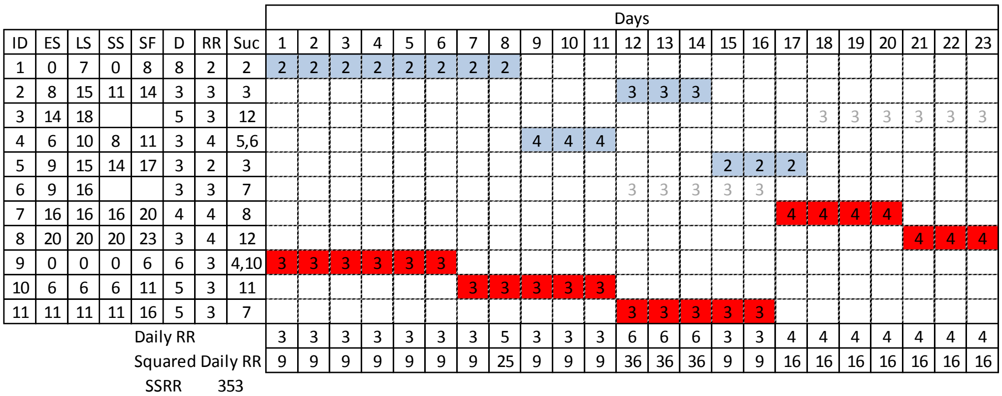


								Days																						
ID	ES	LS	SS	SF	D	RR	Suc	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	0	7	0	8	8	2	2	2	2	2	2	2	2	2	2															
2	8	15	11	14	3	3	3												3	3	3									
3	14	18			5	3	12															3	3	3	3	3	3	3	3	3
4	6	10	8	11	3	4	5,6									4	4	4												
5	9	15			3	2	3												2	2	2	2	2	2	2					
6	9	16			3	3	7												3	3	3	3	3							
7	16	16	16	20	4	4	8																	4	4	4	4			
8	20	20	20	23	3	4	12																					4	4	4
9	0	0	0	6	6	3	4,10	3	3	3	3	3	3																	
10	6	6	6	11	5	3	11							3	3	3	3	3												
11	11	11	11	16	5	3	7												3	3	3	3	3							
				Daily RR				3	3	3	3	3	3	3	5	3	3	3	6	6	6	3	3	4	4	4	4	4	4	4
				Squared Daily RR				9	9	9	9	9	9	9	25	9	9	9	36	36	36	9	9	16	16	16	16	16	16	16
				SSRR		353																								

Figure 4.5. Resource Usage Sheet of the Example Problem after Scheduling of Activity-4

Accordingly, the SS time of 14 is specified for Activity-5 from four alternatives of

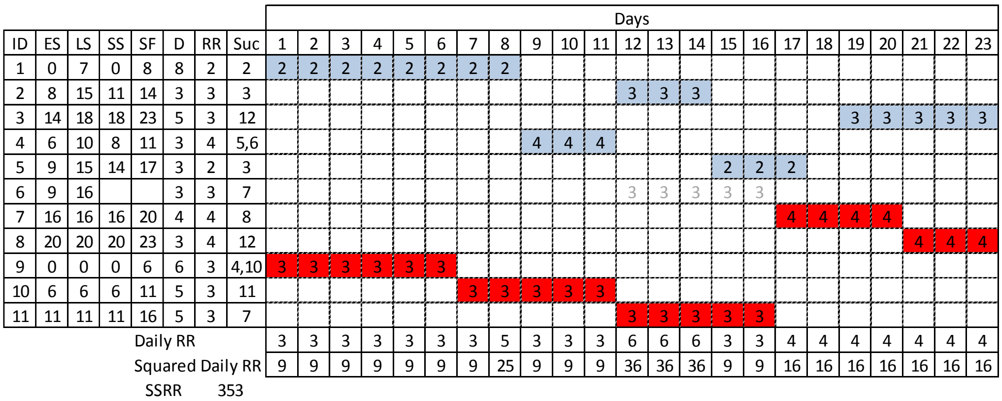
11, 13, 14 and 15, since the value of its gene is 0.70. The rescheduled resource usage sheet after placing the non-critical activities 2, 1, 4 and 5, is shown in Figure 4.6.



								Days																						
ID	ES	LS	SS	SF	D	RR	Suc	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	0	7	0	8	8	2	2	2	2	2	2	2	2	2	2															
2	8	15	11	14	3	3	3												3	3	3									
3	14	18			5	3	12																		3	3	3	3	3	3
4	6	10	8	11	3	4	5,6									4	4	4												
5	9	15	14	17	3	2	3															2	2	2						
6	9	16			3	3	7												3	3	3	3	3							
7	16	16	16	20	4	4	8																	4	4	4	4			
8	20	20	20	23	3	4	12																					4	4	4
9	0	0	0	6	6	3	4,10	3	3	3	3	3	3																	
10	6	6	6	11	5	3	11							3	3	3	3	3												
11	11	11	11	16	5	3	7												3	3	3	3	3							
				Daily RR				3	3	3	3	3	3	3	5	3	3	3	6	6	6	3	3	4	4	4	4	4	4	4
				Squared Daily RR				9	9	9	9	9	9	9	25	9	9	9	36	36	36	9	9	16	16	16	16	16	16	16
				SSRR		353																								

Figure 4.6. Resource Usage Sheet of the Example Problem after Scheduling of Activity-5

The next activity in the priority list is Activity-3 that has two options of days 17 and 18 as its start time, and the corresponding gene value of 0.65. As a result, its SS and SF times are determined as days 18 and 23 respectively that is demonstrated in Figure 4.7.

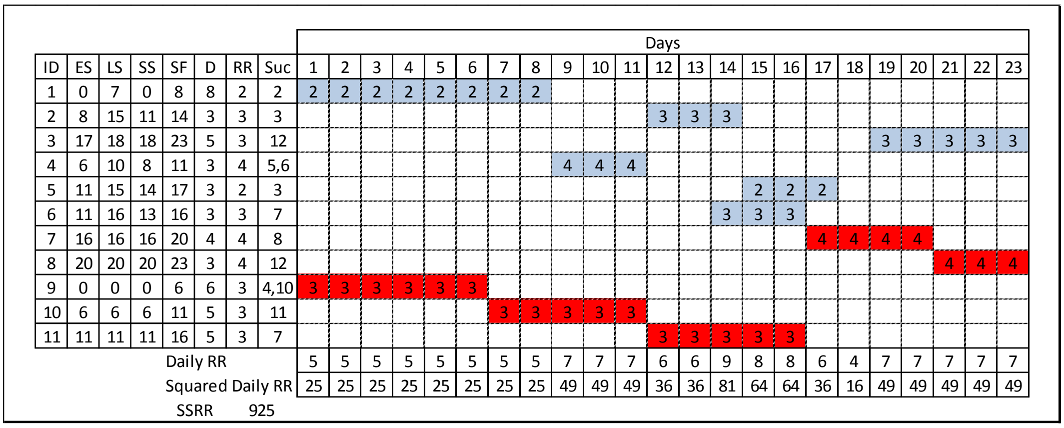


								Days																						
ID	ES	LS	SS	SF	D	RR	Suc	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	0	7	0	8	8	2	2	2	2	2	2	2	2	2	2															
2	8	15	11	14	3	3	3												3	3	3									
3	14	18	18	23	5	3	12																			3	3	3	3	3
4	6	10	8	11	3	4	5,6									4	4	4												
5	9	15	14	17	3	2	3															2	2	2						
6	9	16			3	3	7												3	3	3	3	3							
7	16	16	16	20	4	4	8																	4	4	4	4			
8	20	20	20	23	3	4	12																					4	4	4
9	0	0	0	6	6	3	4,10	3	3	3	3	3	3																	
10	6	6	6	11	5	3	11							3	3	3	3	3												
11	11	11	11	16	5	3	7												3	3	3	3	3							
				Daily RR				3	3	3	3	3	3	3	5	3	3	3	6	6	6	3	3	4	4	4	4	4	4	4
				Squared Daily RR				9	9	9	9	9	9	9	25	9	9	9	36	36	36	9	9	16	16	16	16	16	16	16
				SSRR		353																								

Figure 4.7. Resource Usage Sheet of the Example Problem after Scheduling of Activity-3

Finally, the last activity of the list (Activity-6) is selected and scheduled at the day 13, which is selected from the start time alternative days 11 and 13.

The complete constructed schedule that has an SSRR value of 925, is shown in Figure 4.8.



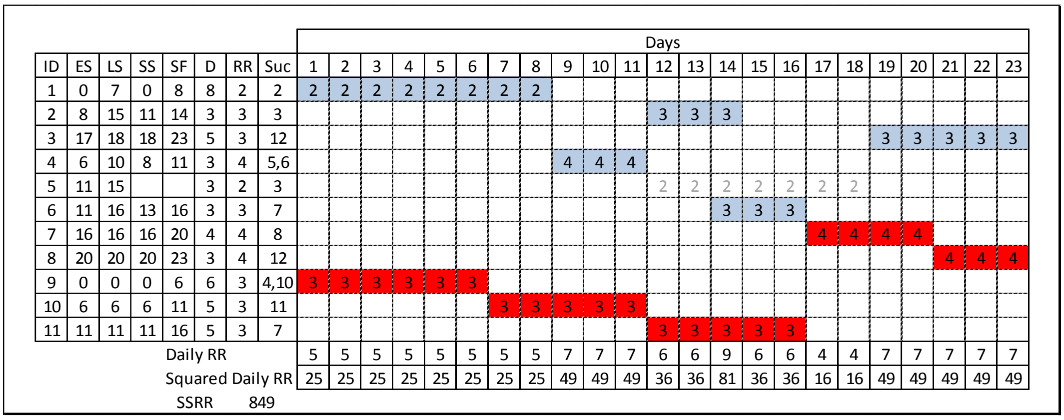
								Days																						
ID	ES	LS	SS	SF	D	RR	Suc	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	0	7	0	8	8	2	2	2	2	2	2	2	2	2	2															
2	8	15	11	14	3	3	3												3	3	3									
3	17	18	18	23	5	3	12																			3	3	3	3	3
4	6	10	8	11	3	4	5,6									4	4	4												
5	11	15	14	17	3	2	3															2	2	2						
6	11	16	13	16	3	3	7														3	3	3							
7	16	16	16	20	4	4	8																	4	4	4	4			
8	20	20	20	23	3	4	12																					4	4	4
9	0	0	0	6	6	3	4,10	3	3	3	3	3	3																	
10	6	6	6	11	5	3	11							3	3	3	3	3												
11	11	11	11	16	5	3	7												3	3	3	3	3							
				Daily RR				5	5	5	5	5	5	5	5	7	7	7	6	6	9	8	8	6	4	7	7	7	7	7
				Squared Daily RR				25	25	25	25	25	25	25	25	49	49	49	36	36	81	64	64	36	16	49	49	49	49	49
				SSRR			925																							

Figure 4.8. Resource Usage Sheet of the Example Problem after Scheduling of All Activities

4.4. Local Improvement Module

After scheduling all the activities and constructing a quasistable schedule resource profile, the obtained histogram is attempted to be improved through an iterative local improvement method. The procedure starts by randomly selection of all the noncritical activities, then, searching for the start time options of activities one by one, without changing the start times of remaining activities in a way that the schedules remain quasistable. When all the non-critical activities are checked for the possible improvement, a new priority list is generated and the process repeats until no improvement is made within a whole iteration.

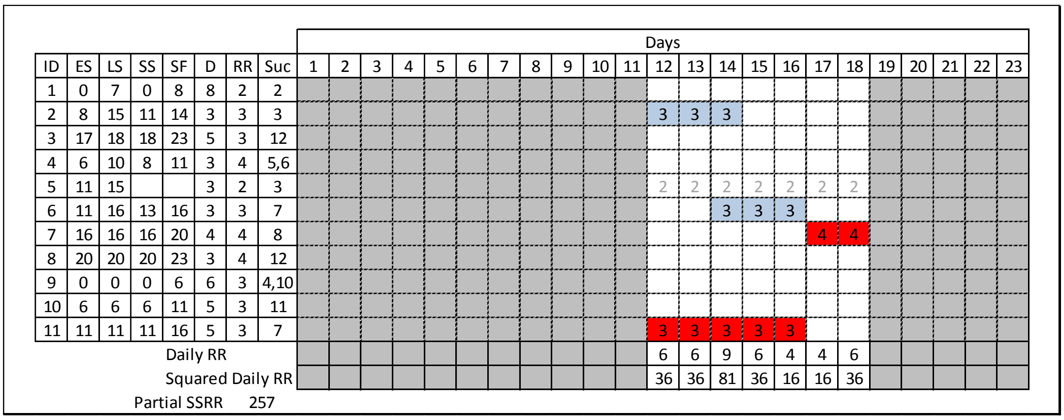
In order to illustrate the local improvement method, it is applied to the schedule of Figure 4.8, with the SSRR value of 925. Supposing that the first activity for improvement is selected as Activity-5, it is displaced from the resource usage profile. Then, all the possible start times for Activity-5 are defined, without changing the start times of remaining activities. Figure 4.9 shows the resource usage sheet for the schedule of Figure 4.8 after displacing Activity-5.



								Days																						
ID	ES	LS	SS	SF	D	RR	Suc	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	0	7	0	8	8	2	2	2	2	2	2	2	2	2	2															
2	8	15	11	14	3	3	3												3	3	3									
3	17	18	18	23	5	3	12																			3	3	3	3	3
4	6	10	8	11	3	4	5,6									4	4	4												
5	11	15			3	2	3												2	2	2	2	2	2	2					
6	11	16	13	16	3	3	7														3	3	3							
7	16	16	16	20	4	4	8																	4	4	4	4			
8	20	20	20	23	3	4	12																					4	4	4
9	0	0	0	6	6	3	4,10	3	3	3	3	3	3																	
10	6	6	6	11	5	3	11							3	3	3	3	3												
11	11	11	11	16	5	3	7												3	3	3	3	3							
Daily RR								5	5	5	5	5	5	5	5	7	7	7	6	6	9	6	6	4	4	7	7	7	7	7
Squared Daily RR								25	25	25	25	25	25	25	25	49	49	49	36	36	81	36	36	16	16	49	49	49	49	49
SSRR							849																							

Figure 4.9. Resource Usage Sheet of the Example Problem after Displacing Activity-5

Local improvement for the Activity-5 is explored by evaluating the SSRR values of all possible quasistable start time alternatives (11, 13, 14 and 15) for it, ensuring to keep the scheduled start times of other activities unchanged. Throughout the evaluation start time options for each activity, only the interval of the resource profile from earliest possible start time to latest possible finish time of that activity is taken to account. This helps to have a very fast calculation. Figure 4.10 shows the feasible interval of resource usage sheet for placement of the Activity-5, and the partial SSRR value for that region. Starting the Activity-5 on days 11, 13, 14 and 15 will result the partial SSRR value to be 353, 353, 333 and 325 respectively. Hence day 15 and 18 are selected as the SS and SF times of Activity-5.

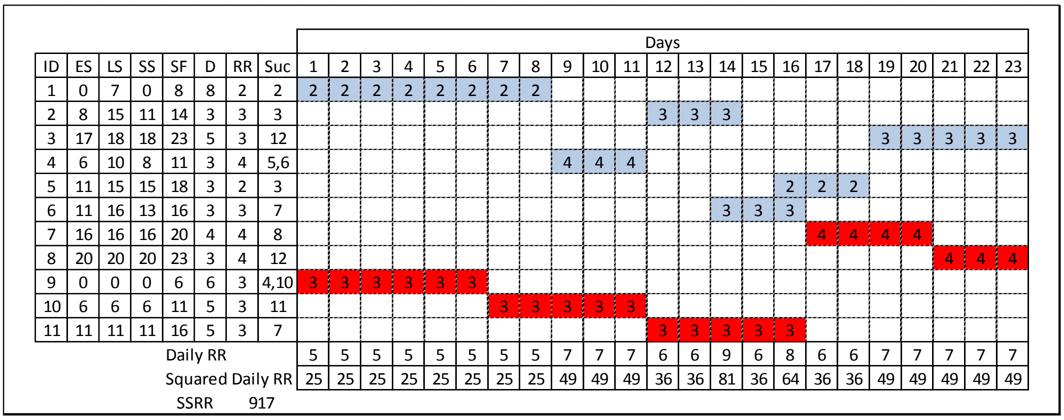


								Days																						
ID	ES	LS	SS	SF	D	RR	Suc	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	0	7	0	8	8	2	2																							
2	8	15	11	14	3	3	3												3	3	3									
3	17	18	18	23	5	3	12																							
4	6	10	8	11	3	4	5,6																							
5	11	15			3	2	3												2	2	2	2	2	2	2					
6	11	16	13	16	3	3	7														3	3	3							
7	16	16	16	20	4	4	8																	4	4					
8	20	20	20	23	3	4	12																							
9	0	0	0	6	6	3	4,10																							
10	6	6	6	11	5	3	11																							
11	11	11	11	16	5	3	7												3	3	3	3	3							
Daily RR																			6	6	9	6	4	4	6					
Squared Daily RR																			36	36	81	36	16	16	36					
Partial SSRR							257																							

Figure 4.10. The Feasible Interval of Resource Usage Sheet for Placement of the Activity-5

Once the scheduled start time for Activity-5 is determined, it is placed in the resource usage sheet and the improved SSRR value is recalculated as 917.

Figure 4.11 shows the improved resource usage sheet after placement of the Activity-5.



								Days																						
ID	ES	LS	SS	SF	D	RR	Suc	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	0	7	0	8	8	2	2	2	2	2	2	2	2	2	2															
2	8	15	11	14	3	3	3												3	3	3									
3	17	18	18	23	5	3	12																			3	3	3	3	3
4	6	10	8	11	3	4	5,6									4	4	4												
5	11	15	15	18	3	2	3																2	2	2					
6	11	16	13	16	3	3	7														3	3	3							
7	16	16	16	20	4	4	8																	4	4	4	4			
8	20	20	20	23	3	4	12																					4	4	4
9	0	0	0	6	6	3	4,10	3	3	3	3	3	3																	
10	6	6	6	11	5	3	11							3	3	3	3	3												
11	11	11	11	16	5	3	7												3	3	3	3	3							
				Daily RR				5	5	5	5	5	5	5	5	7	7	7	6	6	9	6	8	6	6	7	7	7	7	7
				Squared Daily RR				25	25	25	25	25	25	25	25	49	49	49	36	36	81	36	64	36	36	49	49	49	49	49
				SSRR		917																								

Figure 4.11. The Improved Resource Usage Sheet after Placement of the Activity-5

The procedure is repeated until all the non-critical activities in the priority list have been checked for improvement. Then a new priority list is generated and the process is repeated until no improvement is made within a whole cycle of iteration.

4.5. Crossover and Mutation in QHGA

In QHGA, an individual solution representing a chromosome is generated throughout a complete cycle of constructive and local improvement modules. The first population is created by randomly generated chromosomes values. Subsequently, the new individuals are introduced by using crossover and mutation operators. Like MASA, in QHGA also one point crossover is performed for problems including 10 or less non-critical activities, and two point crossover is done for problems including more than 10 non-critical activities. In order to attain the highest performance for QHGA, a tuning is done for its parameters including population size, mutation rate, and crossover rate with respect to the quality of the solutions. Combinations of the parameters are performed with three levels of low, medium and high. Based on the fine-tuning analyze, adequate set of parameter values for QHGA are determined as summarized in Table 4.1. The flowchart of MASA is given in Figure 4.12.

Table 4.1. Parameter Selection of QHGA

Parameter	Range of Parameters			Selected Value
	Low	Medium	High	
Population size	30	40	50	40
Crossover rate	0.20	0.30	0.40	0.30
Mutation rate	0.05	0.10	0.15	0.05

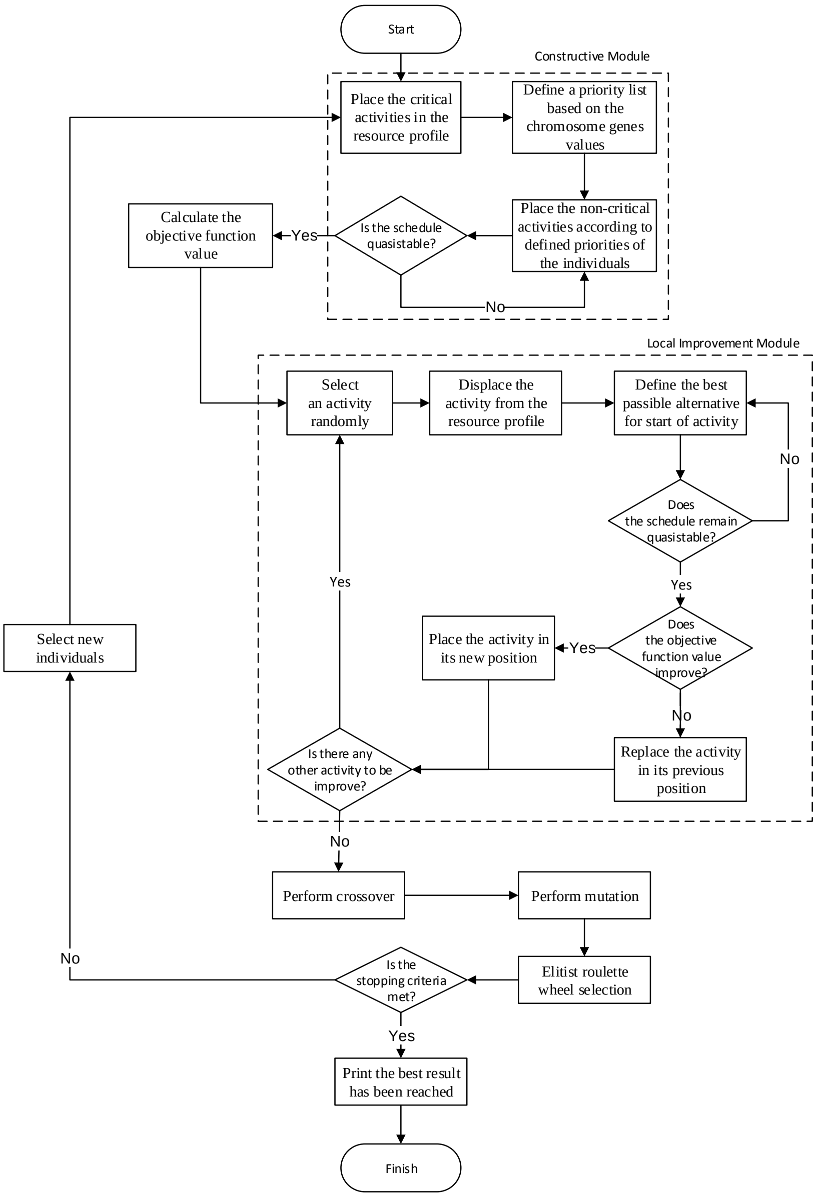
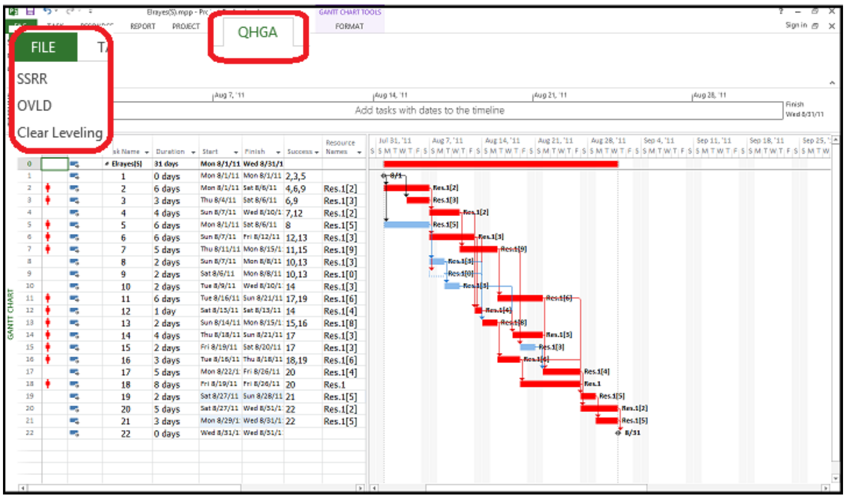


Figure 4.12. Flowchart of QHGA

4.6. Integration of QHGA to the Microsoft Project (2013)

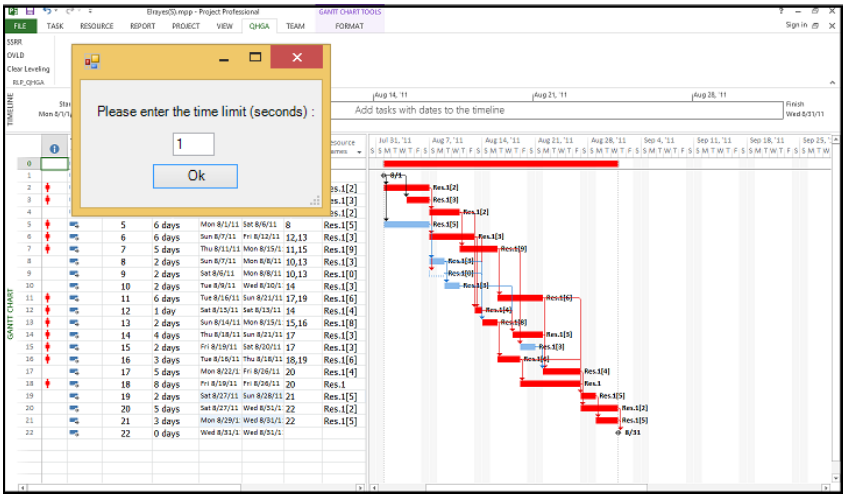
In order to have a more practical and facilitated application, QHGA is integrated to Microsoft Project Professional (MSP) version 2013, one of the commonly used software in construction projects scheduling. The integration is done using C# programing language within the Visual Studio 2013. QHGA ribbon of MSP 2013 is shown in Figure 4.13.



			sk Name	Duration	Start	Finish	Success	Resource Names
0			Elrayes(S)	31 days	Mon 8/1/11	Wed 8/31/1		
1			1	0 days	Mon 8/1/11	Mon 8/1/11	2,3,5	
2	◆		2	6 days	Mon 8/1/11	Sat 8/6/11	4,6,9	Res.1[2]
3	◆		3	3 days	Thu 8/4/11	Sat 8/6/11	6,9	Res.1[3]
4			4	4 days	Sun 8/7/11	Wed 8/10/1	7,12	Res.1[2]
5	◆		5	6 days	Mon 8/1/11	Sat 8/6/11	8	Res.1[5]
6	◆		6	6 days	Sun 8/7/11	Fri 8/12/11	12,13	Res.1[3]
7	◆		7	5 days	Thu 8/11/11	Mon 8/15/1	11,15	Res.1[9]
8			8	2 days	Sun 8/7/11	Mon 8/8/11	10,13	Res.1[3]
9			9	2 days	Sat 8/6/11	Mon 8/8/11	10,13	Res.1[0]
10			10	2 days	Tue 8/9/11	Wed 8/10/1	14	Res.1[3]
11	◆		11	6 days	Tue 8/16/11	Sun 8/21/11	17,19	Res.1[6]
12	◆		12	1 day	Sat 8/13/11	Sat 8/13/11	14	Res.1[4]
13	◆		13	2 days	Sun 8/14/11	Mon 8/15/1	15,16	Res.1[8]
14	◆		14	4 days	Thu 8/18/11	Sun 8/21/11	17	Res.1[3]
15	◆		15	2 days	Fri 8/19/11	Sat 8/20/11	17	Res.1[3]
16	◆		16	3 days	Tue 8/16/11	Thu 8/18/11	18,19	Res.1[6]
17			17	5 days	Mon 8/22/1	Fri 8/26/11	20	Res.1[4]
18	◆		18	8 days	Fri 8/19/11	Fri 8/26/11	20	Res.1
19			19	2 days	Sat 8/27/11	Sun 8/28/11	21	Res.1[5]
20			20	5 days	Sat 8/27/11	Wed 8/31/1	22	Res.1[2]
21			21	3 days	Mon 8/29/1	Wed 8/31/1	22	Res.1[5]
22			22	0 days	Wed 8/31/1	Wed 8/31/1		

Figure 4.13. QHGA Ribbon of MSP 2013

As shown in Figure 4.13, QHGA menu ribbon of MSP 2013 contains three buttons of “SSRR”, “OVLD” and “Clear Leveling”. The “SSRR” button implements QHGA which minimizes the sum of squares of daily resource requirement for the current project in MSP and applies the obtained solution to the schedule of that project. Likewise, the “OVLD” button performs the same procedure to minimize the overload amount of daily resource usage. Finally, the “Clear Leveling” button clears all the changes on the start times off current project in MSP, which have been applied by QHGA. In other words, it restores the schedule of the current project, as it was before the implementation of QHGA. Once the “SSRR” or “OVLD” buttons are clicked, the computation time limit is asked as it is shown in Figure 4.14. The computation time limit is a user defined parameter which is used as the stopping criteria for QHGA.



Please enter the time limit (seconds) :

1

Ok

2						es.1[2]
3						es.1[3]
4						es.1[2]
5	5	6 days	Mon 8/1/11	Sat 8/6/11	8	Res.1[5]
6	6	6 days	Sun 8/7/11	Fri 8/12/11	12,13	Res.1[3]
7	7	5 days	Thu 8/11/11	Mon 8/15/1	11,15	Res.1[9]
8	8	2 days	Sun 8/7/11	Mon 8/8/11	10,13	Res.1[3]
9	9	2 days	Sat 8/6/11	Mon 8/8/11	10,13	Res.1[0]
10	10	2 days	Tue 8/9/11	Wed 8/10/1	14	Res.1[3]
11	11	6 days	Tue 8/16/11	Sun 8/21/11	17,19	Res.1[6]
12	12	1 day	Sat 8/13/11	Sat 8/13/11	14	Res.1[4]
13	13	2 days	Sun 8/14/11	Mon 8/15/1	15,16	Res.1[8]
14	14	4 days	Thu 8/18/11	Sun 8/21/11	17	Res.1[3]
15	15	2 days	Fri 8/19/11	Sat 8/20/11	17	Res.1[3]
16	16	3 days	Tue 8/16/11	Thu 8/18/11	18,19	Res.1[6]
17	17	5 days	Mon 8/22/1	Fri 8/26/11	20	Res.1[4]
18	18	8 days	Fri 8/19/11	Fri 8/26/11	20	Res.1
19	19	2 days	Sat 8/27/11	Sun 8/28/11	21	Res.1[5]
20	20	5 days	Sat 8/27/11	Wed 8/31/1	22	Res.1[2]
21	21	3 days	Mon 8/29/1	Wed 8/31/1	22	Res.1[5]
22	22	0 days	Wed 8/31/1	Wed 8/31/1		

Figure 4.14. The Computation Time Limit Input Page of QHGA in MSP 2013

4.7. Computational Experiments

Comprehensive computational experiments are conducted to evaluate the performance of the proposed QHGA, using benchmark instances from the literature and a real construction project. QHGA is coded in C# and compiled within Visual Studio 2013 on a 64 bit platform. All of the tests are carried out on a computer with a 3.00 GHz Core 2 Duo Processor E8400 Intel CPU. Percent deviation from the upper bound (PD) is used to evaluate the performance of the different methods along with the CPU time. The formulation of the PD is described in the Eq.(3.27).

Three different experiments are conducted for performance evaluation of QHGA. First, the problem instances of PSPLIB (Kolisch & Sprecher, 1997), including the networks up to 120 activities and four resources are used with the SSRR objective function metric for comparison of the performance of QGHA with MASA and other state-of-art RLP methods. Then, instances up to 2000 activities are generated and implemented to compare QHGA with the commonly used commercial project scheduling software packages. Finally, a real case construction project is adopted to evaluate QHGA capability in real-life-size projects.

4.7.1. Performance Analyzes of QHGA Using PSPLIB Instances

In the previous chapter, the capability of MASA was revealed compared to the other state-of-art algorithms. In this section the performance of QHGA is compared with the performance of AGA, Burgess2 and MASA using the same instances from PSPLIB. Over again, all of the tests were carried out using SSRR objective function metric and, the weights of all four resources are taken to be equal as 1. The same CPU times of MASA here also was used as the stopping criterion for QHGA. The APD values given in Table 4.2 are the average percentage deviations from the current best solutions for all four algorithms. The optimal solutions obtained with the mixed integer linear model in Chapter 3, is used here again for J30 instance set as the current best value.

Summarized computational results are presented in Table 4.2. The complete results for all instances and optimal solutions for J30 instances are illustrated in Appendix A. The computational results show that with an APD of 0.1 for J30 instances, QHGA was able to obtain high quality solutions which were either optimal or very close to the optimal. QHGA was able to obtain the optimal solutions for 239 instances out of 475 J30 instances. The second best method is MASA with an APD of 0.2 for J30 instances, and was able to determine the optimal for 232 instances. AGA with an APD of 0.7 was the third best method for J30 instances, and was able to determine the optimal solutions for 76 instances. Finally, the Burgess2 is ranked the worst for J30 instances among the four methods with an APD of 3.6. QHGA obtained the best results almost for all of the J60 and J120 instances, with the achieved APD of 0.0. MASA is the second best method which achieved the APD of 0.6 and 1.4 for J60 and J120 instances respectively. AGA had an APD of 2.9 for J60, and 5.1 for J120 instances, and the APD of Burgess2 was 3.7 for J60, and 3.5 for J120 instance sets. QHGA shows significantly better performance than other methods for larger problems.

Table 4.2. Computational Results of QHGA for PSPLIB Instances

Instance Sets	AGA (Ponz-Tienda et al., 2013)			Burgess2			MASA			QHGA		
	APD (%)	No of Optimal	Time (Sec.)	APD (%)	No of Optimal	Time (Sec.)	APD (%)	No of Optimal	Time (Sec.)	APD (%)	No of Optimal	Time (Sec.)
J30 (480)	0.7	76	15	3.6	14	12.6	0.2	232	12.6	0.1	239	12.6
J60 (480)	2.9	NA	NA	3.7	NA	18.3	0.6	NA	18.3	0.0	NA	18.3
J120 (480)	5.1	NA	NA	3.5	NA	27.6	1.4	NA	27.6	0.0	NA	27.6
Average:	2.9			3.6		19.5	0.7		19.5	0.0		19.5

Paired t tests are performed to evaluate the significance of the difference between QHGA and MASA for each instance set containing 480 problems. The t values (t) for each test given in Table 4.3 are significantly greater than the critical t (t_c) value of 2.33 for $\alpha = 0.01$ with degrees of freedom ($df = 479$). The test results reveal that the average percentage deviations of QHGA from the upper bounds were significantly lower than the average percentage deviations of MASA for all the problems sets results. Overlay, the computational experiment results confirmed the effectiveness of QHGA. Analyses also show that, as the size of the problems grow, the performance of QHGA becomes meaningfully significant than MASA. This feature of QHGA enables it to be applicable in larger size problems in practice.

Table 4.3. Paired t-Test Statistics for QHGA

Method	J30	J60	J120
MASA	3.00	12.52	19.92

4.7.2. Comparison of QHGA with Microsoft Project and Primavera

Despite the importance of resource leveling in practice, Primavera and Microsoft Project which are two most commonly used software packages for planning and management of construction projects, have very limited capabilities in dealing with RLPs. In Chapter 3, weakness of the both software programs in solving a problem set up to 120 activities is revealed comparing to MASA. In this section, the performance of QHGA is compared with the performance of nine priority based leveling heuristics available in Microsoft Project 2010 and Primavera 6.7, using OVLD objective function metric for a problem set with large size problem instances up to 2000 activities. Here again, the heuristics included Standard (STD) heuristics of Microsoft Project (MSP) 2010, and ID-Ascending (IDA), ID-Descending (IDD), Total Float-Ascending (TFA), Total Float-Descending (TFD), Early Start-Ascending (ESA), Early Start-Descending (ESD) Late Finish-Ascending (LFA), and Late Finish-Descending (LFD) heuristics of Primavera P6 version 8.4.

In this section of the experiments, the case problem of El-Rayes and Jun (2009) is used as the basis for creating medium-size to large-size (100, 200, 500 , 1000 and 2000 activities) projects. The case problem of El-Rayes and Jun (2009) consists of 20 activities and a known optimal OVLD objective function value of 50 (Yeniocak, 2013). The case problems are created by copying the base project in a serial manner which helps to know the expected optimum solution for all of the cases. The optimal OVLD values of the problems are used for testing the quality of the solutions reached by QHGA and comparing it with the solutions obtained by Primavera and MSP 2013. In calculation of the OVLD values, the targeted demands for resources are determined by rounding the average resource demand for each resource using the floor function, and the weights of all four resources are taken as equal. The shorter computational time required either by Primavera or MSP 2013 for solution of any problem, is set as the stopping criterion for QHGA in solving the same problem. The performance comparison results are presented in Table 4.4.

According to the Table 4.4, QHGA could reach the overall APD of 0.8 from the optimal solutions of the instances. Among the ten evaluated methods, QHGA determined the optimal solutions for the instances up to 200 activities. Late Finish-Descending heuristic of Primavera obtained the second best solution with the APD of 10. It is shown that QHGA surpasses the best heuristic of primavera by a huge margin within the same periods of computational time. This performance gap between QHGA and the leveling heuristics of Primavera and MSP 2013, once more revealed the limitations of the commercial project management software for resource leveling.

Table 4.4. Comparison of QHGA with Microsoft Project and Primavera

No	No of Act.	Optimal OVLD	Primavera P6 (8.4)									Microsoft Project (2013)		QHGA	
			IDA	IDD	TFA	TFD	ESA	ESD	LFA	LFD	Time (S)	STND	Time (S)	NA	Time (S)
1	20	50	20.0	22.0	14.0	18.0	22.0	14.0	20.0	10.0	1	28.0	1	0.0	1
2	100	250	20.0	18.0	14.0	18.0	22.0	14.0	20.0	10.0	1	28.0	1	0.0	1
3	200	500	20.0	18.0	14.0	18.0	22.0	14.0	20.0	10.0	2	28.0	2	0.0	2
4	500	1250	20.0	18.0	14.0	18.0	22.0	14.0	20.0	10.0	3	28.0	9	0.5	3
5	1000	2500	20.0	17.9	14.0	18.0	22.0	14.0	20.0	10.0	4	28.0	50	2.5	4
6	2000	5000	20.0	18.0	14.0	18.0	22.0	14.0	20.0	10.0	9	28.0	315	1.9	9
APD (%)			20.0	18.6	14.0	18.0	22.0	14.0	20.0	10.0		28.0		0.8	

4.7.3. Evaluation of QHGA Using a Real Construction Project Case Study

The robustness of QHGA is revealed in the previous sections through the experiments are done using the large-size problem instances from the literature. In this section, performance of QHGA is studied using a real construction project, in order to do a more practical evaluation, and show the benefits of having an efficient resource leveling in the real-world construction. The project is a process plant project constructed by a Turkish contractor in Jordan. Total duration of construction works for this project was 672 working days and included 522 activities and 18 manpower resources. Total duration of this industrial plant project were 672 working days and included 522 activities and 18 resources. This number of resources includes only the direct manpower resources, since the effects of leveling for that sort of resources are meant to be demonstrated. All the precedence relation types of start to start (SS), start to finish (SF), and finish to finish (FF) are transformed to the finish to start (FS) in order to be appropriate for QHGA. This problem is then set up and solved using QHGA, the standard heuristic of MSP 2013, and all the aforementioned eight heuristics of Primavera P6 (R 8.4). The OVLD metric here also is adopted for calculation of objective function value. The comparisons are shown in Table 4.5. Because the MSP 2013 could not solve the problem in a five hours defined time period, it is excluded from the comparisons for this problem.

Table 4.5. Comparison of QHGA with Primavera, Using a Real Case Construction Project

Upper Bound	Primavera P6 (8.4)									QHGA	
	IDA	IDD	TFA	TFD	ESA	ESD	LFA	LFD	Time (S)	NA	Time (S)
63616	66708	66035	65548	66749	66759	66190	65487	66268	4	63616	4
PD (%)	4.9	3.8	3.0	4.9	4.9	4.0	2.9	4.2		0.0	

According to the experiments results shown in Table 4.5, QHGA could acquire better solution than all eight heuristics of Primavera within the same computational time. The LFA heuristic of Primavera as the best from all the eight heuristics has a percent deviation (PD) of 2.9 from the result reached by QHGA.

In order to better demonstrate the influences of having an effective resource leveling on the project cost, the impact of using both QHGA and Primavera are investigated over the case project. For this purpose, the individual peaks of the resources within the whole resource utilization curve for the early start (ES) schedule earned by CPM, and the leveled scheduled by both QHGA and Primavera are studied. Throughout the project execution, part of the skilled manpower were employed and transferred from Turkey. A total cost of 2500 United States Dollar (USD) is estimated as the indirect expenses of each manpower resource that is employed from Turkey. These expenses include the cost of travel, visa, working permit, and safety expenses. For a Jordanian worker this cost was estimated as $500 per worker. Hence as the resource requirement for each worker is increased, the indirect cost part that depends on the number of workers for each resource type is also increased.

Table 4.6 shows the list of resources along with the estimated indirect cost of manpower mobilization for each resource within the project.

The resource requirement peaks and the total indirect cost of manpower mobilization expenses for the early start schedule and for the leveled schedule by QHGA and all eight heuristic of Primavera are shown in Table 4.7. The total indirect cost of manpower mobilization expenses for the case construction project for early start schedule which is prepared with CPM is $1, 886,000. This amount is reduced to $1,645,500, when QHGA is implemented to level the resources. It means that, QHGA caused to reduce the project total cost by $240,500. With the resource leveling heuristics of Primavera, in the best case which is happened with the TFD heuristic, it could reduce the cos by $24,000. According to the experiments, QHGA was able to generate a more efficient resource usage profile than Primavera and had

$216,500 more saving. Moreover, since the weights of the resources are adjustable in QHGA, increasing the weights of the resources with larger amount of aforementioned manpower cost will definitely increase the amount of saving. As shown in Table 4.7, setting a larger weight of 5 for the Turkish workers has caused to reduce the cost to $1,445,000 and increase the amount saving to $431,000. These results of QHGA are obtained within the same computational time as Primavera, while, it has the option to run in larger computational time periods which indeed will find better alternative solutions.

Table 4.6. Required Resources of the Real Case Construction Project

No	Resource Name	Indirect Cost of Manpower Mobilization (US$)
1	Carpenter	500
2	Carpeting Helper	500
3	Cement Finisher	500
4	Electrician	2500
5	Electrician Helper	2500
6	Fabricated Item Installer	2500
7	Fabricated Item Installing Helper	2500
8	Fabricated Item Welder	2500
9	Fabricated Item Welding Helper	2500
10	Mechanical Installer	2500
11	Mechanical Installing Helper	2500
12	Pipe Fitter	2500
13	Pipe Fitting Helper	2500
14	Pipe Welder	2500
15	Pipe Welding Helper	2500
16	Reinforced Iron Worker	500
17	Structural Iron Worker	500
18	Structural Iron Welder	500

Table 4.7. Resource Requirement Peaks and the Total Indirect Cost of Manpower Mobilization Expenses for the Real Case Construction Project

N	Early Start	Primavera P6 (8.4)								QHGA	QHGA (Weighted OVLD)	
		IDA	IDD	TFA	TFD	ESA	ESD	LFA	LFD		Resource Weights	Peaks
1	213	142	178	165	167	158	167	151	150	72	1	76
2	56	35	48	43	41	43	44	41	37	21	1	24
3	50	32	42	40	35	36	38	38	32	15	1	26
4	70	99	112	114	116	135	101	165	102	108	5	115
5	52	61	78	77	71	101	71	105	63	95	5	73
6	39	43	45	36	39	45	44	34	40	25	5	25
7	32	32	25	24	26	24	28	35	26	20	5	22
8	66	71	78	58	72	76	80	59	70	42	5	42
9	32	27	27	31	27	29	29	29	24	26	5	19
10	83	106	89	67	90	94	83	77	91	38	5	42
11	51	59	51	43	51	52	51	47	49	23	5	26
12	45	41	55	47	39	69	43	44	41	50	5	40
13	26	24	24	26	20	37	25	26	24	29	5	19
14	87	74	86	83	61	122	78	82	82	93	5	66
15	60	50	57	56	45	86	54	54	54	64	5	43
16	156	102	131	119	117	118	121	110	108	49	1	53
17	71	69	71	71	69	60	71	67	71	60	1	63
18	11	10	10	11	10	8	11	9	11	9	1	8
Total (US$)	1,886,000	1,912,500	2,057,500	1,879,500	1,862,000	2,386,500	1,943,500	2,100,500	1,869,500	1,645,500		1,455,000

CHAPTER 5

A CRITICAL SEQUENCE CRASHING HEURISTIC FOR RESOURCE CONSTRAINED DISCRETE TIME-COST TRADE-OFF PROBLEM

Despite the importance of project deadlines and resource constraints in construction scheduling, very little success has been achieved in solving the resource constrained discrete time-cost trade-off problem (RCDTCTP), especially for large-scale projects. In this chapter a new heuristic method is designed and developed to achieve fast and high quality solutions for the large-scale RCDTCTP. The proposed heuristic consists of two parts. In the first part, backward-forward scheduling technique is adopted for the resource constrained project scheduling problem. The critical sequence including the activities which determine the project duration for a resource constrained schedule are crashed in the second part. The computational experiment results revealed that the new critical sequence crashing heuristic outperforms the state-of-art methods, both in terms of the solution quality and computation time. Solutions with a deviation of 0.25% from the upper bounds are achieved for a large-scale project including up to 2,000 activities within few seconds. The main contribution of the new heuristic to practitioners and researchers is that it provides a fast and effective method for optimal scheduling of real-life-size construction projects with project deadlines and resource constraints.

5.1. Resource Constrained Discrete Time-Cost Trade-off Problem

The objective of resource constrained time-cost trade-off problem is to determine a time/cost/resource mode (option) and a start date for each activity in such a way that, the precedence and resource constraints are satisfied, and the total direct costs, indirect costs, and the delay penalties (liquidated damages) are minimized. In the discrete version of this problem the relation between the duration of activities and the resources committed is discrete.

Different versions of the resource constrained time-cost trade-off problem have been studied in the literature. Chua et al. (1997) considered exceeding the resource constraints at an additional cost for optimizing the resourced constrained time-cost trade-off problem. Ahn and Erenguc (1998) minimized the sum of direct costs and the penalty costs for the resource constrained project scheduling problem in which the duration reduction (crashing) can be performed. Hegazy and Menesi (2012) and Menesi et al. (2013) focused on minimizing the sum direct and indirect costs and the penalties and incentives.

Few studies aimed to achieve the complete non-dominated set of the time/cost/resource modes and the start dates over the set of feasible project durations called the Pareto front, while considering the resource constraints. Leu and Yang (1999) obtained the non-dominated solutions that minimized the sum of direct and indirect costs for the RCDTCTP. Chen and Weng (2009) also focused on Pareto front optimization for the RCDTCTP and considered activity interruption. Wuliang and Chengen (2009) presented a multi-mode resource-constrained discrete time-cost tradeoff model to achieve the Pareto front for the RCDTCTP.

The majority of the RCDTCTP studies used problem instances, including up to 50 activities in computational experiments. Hegazy and Menesi (2012) reported the performance of a heuristic method for 360 activities. Menesi et al. (2013) used large size instances, including up to 2000 activities in computational experiments.

5.2. The Critical Sequence

In critical path method, the project duration is calculated by adding the durations of the activities on the longest path in the project network called the “critical path” which is determined by the precedence relations. When there are resource constraints, the critical path method is not sufficient to identify the sequence of activities that determine the project duration so-called critical sequence (Wiest, 1964) or critical chain (Goldratt, 1997). Wiest (1964) presented a procedure for calculation of floats in an early study to define and identify the critical sequence. Lu and Li (2003) proposed resource-activity critical-path method to calculate the floats and to determine the sequence of critical activities for resource-constrained scheduling. Lu et al. (2008) developed a simplified simulation-based scheduling system to provide valid floats and optimum schedules for the RCPSP.

The significance of critical sequence in the resource constrained scheduling is similar to the importance of critical path, on the critical path method. The precedence and resource feasible project duration can be shortened by crashing the activities that are on the critical sequence(s). Unlike the critical path method, in resource constrained project scheduling it is sometimes possible to shorten the project duration by crashing the activities that are not on the critical sequence (Wiest, 1964). However, an efficient heuristic method can be designed for the RCDTCTP by only considering crashing of the activities that are on the critical sequence(s) which is the main focus of this research.

5.3. Critical Sequence Crashing Heuristic (CSCH)

A novel heuristic method that is based on crashing of the critical sequence is designed and developed especially for large scale RCDTCTP. The heuristic method consists of two parts; backward-forward resource constrained scheduling, and critical sequence crashing.

5.3.1. Backward-Forward Resource Constrained Scheduling

The critical sequence crashing heuristic (CSCH) starts the search by using the normal (un-crashed) modes for the activities. Once the modes are selected the start dates of the activities and project completion can be determined by using a resource constrained project scheduling method. The resource constrained project scheduling method used to determine the start dates of activities has a very significant impact on the project duration and the critical sequence(s) (Herroelen & Leus, 2005). Backward-forward resource constrained scheduling method is integrated to the proposed critical sequence heuristic, to achieve an adequate and fast solution for the RCPSP.

Backward-forward scheduling method was proposed by Li and Willis (1993) to improve a feasible resource constrained schedule by increasing the resource utilization. Lova and Tormos (2001) developed a heuristic using backward-forward scheduling method for resource constrained multi-project scheduling problem, and showed that backward-forward scheduling improved the multi-project duration. In a recent study, the backward-forward scheduling method integrated hybrid genetic algorithm has outperformed the state-of-the-art methods for resource constrained multi-project scheduling problem (Sonmez & Uysal, 2014).

The backward-forward scheduling method performs resource constrained scheduling twice, by using the serial scheduling scheme (Kelley, 1963). The serial scheduling scheme sequentially schedules the activities (one by one) at their earliest precedence and resource feasible start time, according to a priority list. In backward-forward scheduling, first backward scheduling is executed in the reverse time direction then, forward scheduling is performed. An arbitrary project completion time is selected to start backward scheduling, since the exact duration of the feasible schedule is not known at the beginning. The resulting backward schedule is adjusted such that the project completion start is equal to time instant zero.

In the backward scheduling phase of the proposed CSCH, total floats of activities that are calculated by the critical path method are used to determine the priority list. The activity with the smallest float is backward scheduled first, and in case of a tie the activity with the larger activity number is selected. The total float priority enables the activities on the critical path to be resourced constrained scheduled first, and usually works well when the resource constraints are not tight. The forward scheduling is performed in the order of start times that are obtained in backward scheduling, and in case of a tie the activity with the larger activity number is selected.

5.3.2. Critical Sequence Crashing

The critical sequence(s) is identified for the schedule determined in backward-forward scheduling to start crashing. In the proposed heuristic method, the critical sequence is defined as the sequence of activities that determine the project duration for a precedence and resource feasible schedule. Hence, the critical sequence(s) is identified by tracking the sequence(s) of the activities that determine the project duration, by starting from the latest activity. Removal of local suboptimal results (Wiest, 1964) are not performed to identify the critical sequence in a short amount of computation time.

The crashing is performed only to the activities that are on the critical sequence(s). Among the activities that are on the critical sequence(s), the activity with the least daily crashing cost is crashed first, considering one activity crashing option at a time. In case of a tie, the activity with the least resource impact (least crashing resource difference) is selected. If the tie is not broken, the activity with the larger activity number is selected as the third criterion.

Backward-forward resource constrained scheduling is performed to determine the project duration once the activity to be crashed is determined. The project duration obtained by the latest mode selections is compared with the project duration obtained by the previous mode selections (in the first cycle previous mode selections includes the normal modes). Crashing is not executed and the selected activity is not crashed further, if the project duration of the latest mode selections is larger than the project duration of the previous mode selections. Finally, the crashing and backward-forward scheduling stages are executed until all of the activities in the latest critical sequence(s) are considered for crashing, and the solution with the minimum cost is reported. The flow chart of the critical sequence crashing heuristic method is illustrated in Figure 5.1.

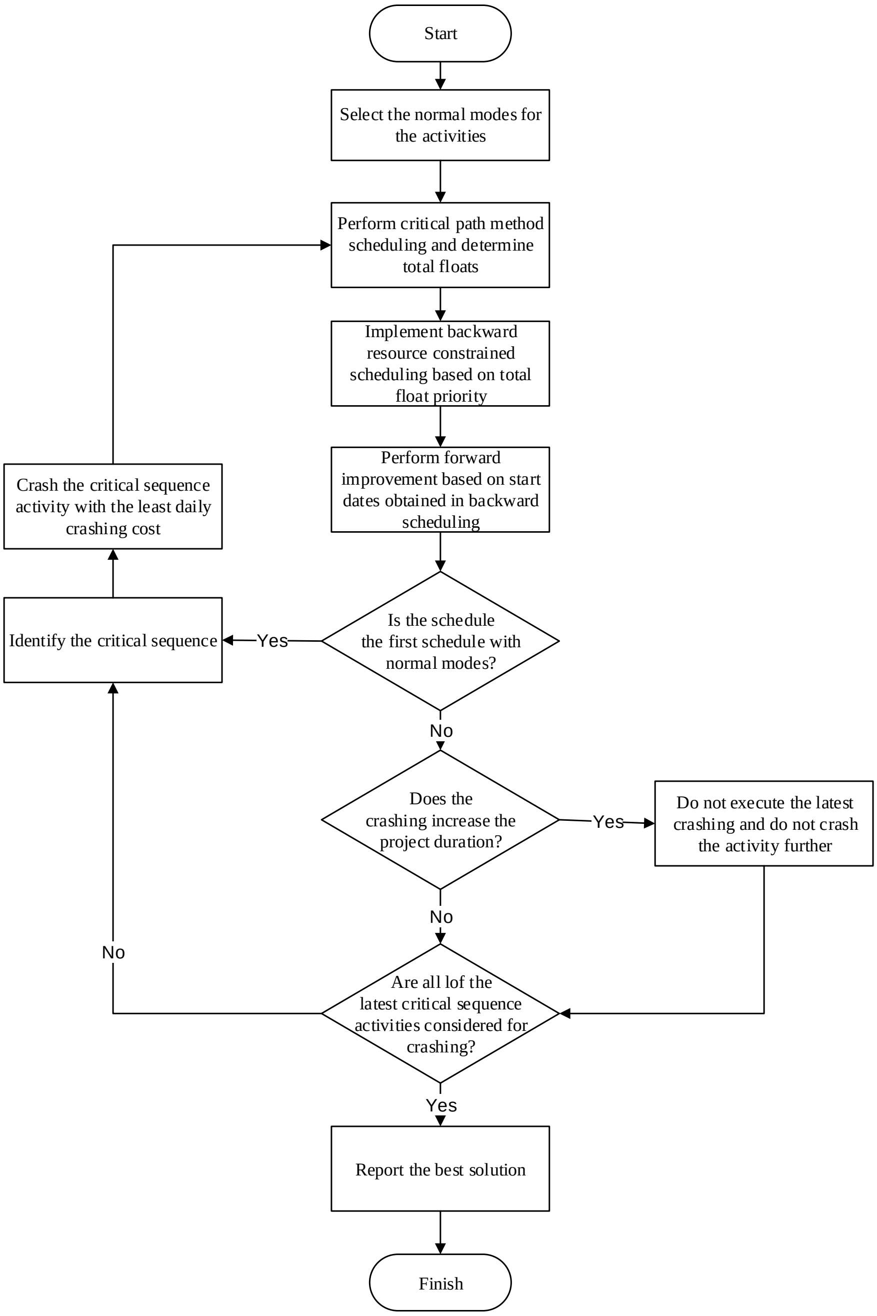


Figure 5.1. Flow Chart of Critical Sequence Crashing Heuristic

5.3.3. Case Example

A case example is presented in Figure 5.2, to illustrate the proposed CSCH. The deadline for the case example is 36 days, the indirect costs and the liquidated damages are defined as 2,500 \$/day and 5,000 \$/day respectively. The backward-forward resource constrained scheduling is initiated by selecting the normal modes for all of the activities.

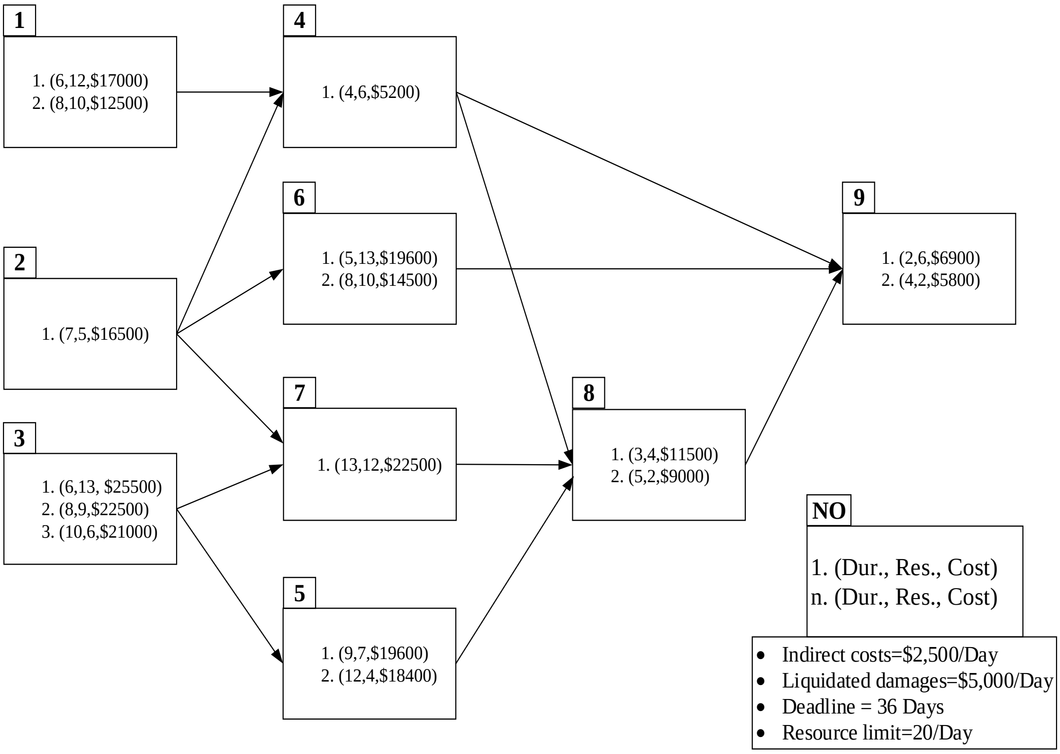


Figure 5.2. Network and Activity Modes of Case Example

Critical path method is performed to determine the floats of the activities as shown in Figure 5.3. The backward scheduling priority list is determined as <9, 8, 7, 3, 5, 2, 4, 1, 6> based on the total floats that are calculated according to the critical path method and by selecting the activity with the larger activity number in case of a tie.

The backward scheduling is performed according to the priority list by scheduling the activities in the reverse time direction (one by one) at their latest precedence and resource feasible finish time, using an arbitrary project completion time of 50 days, as shown in Figure 5.4.

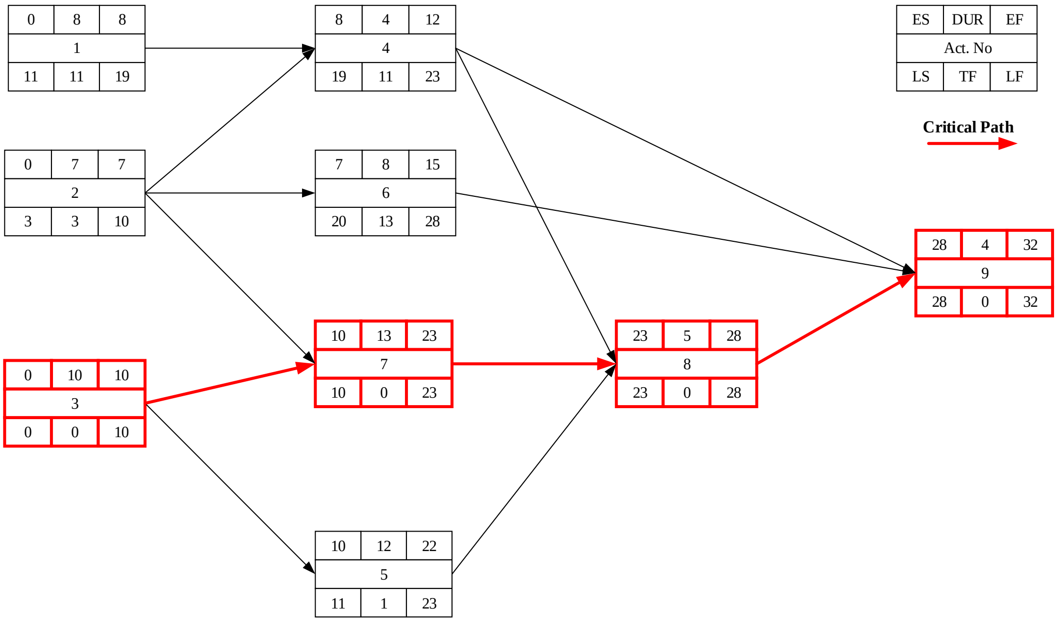


Figure 5.3. Critical Path Method Schedule for Case Example

The resulting backward schedule is adjusted such that the project start time is equal to day zero (Figure 5.5) and the project duration is obtained as 47 days.

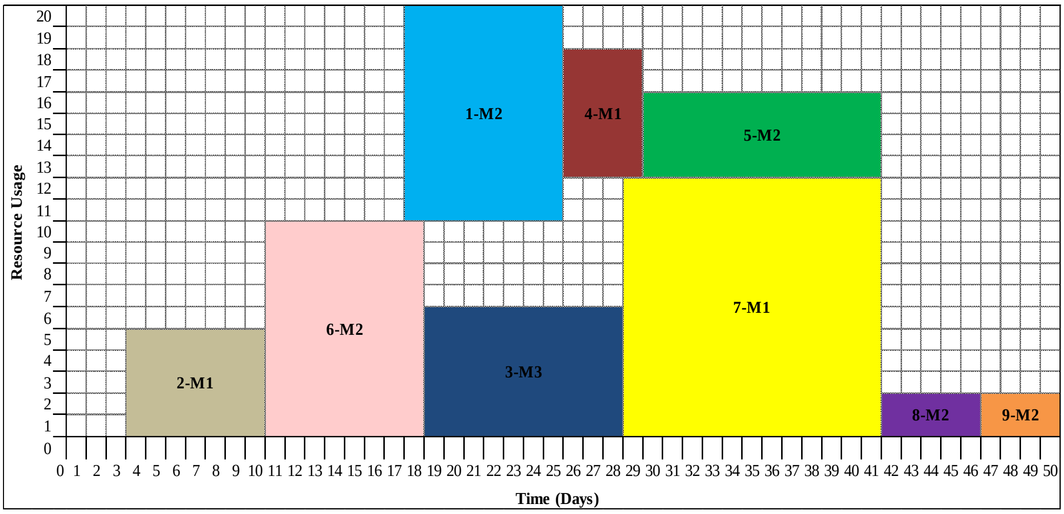


Figure 5.4. Resource Constrained Backward Schedule with Arbitrary Completion Time for Case Example

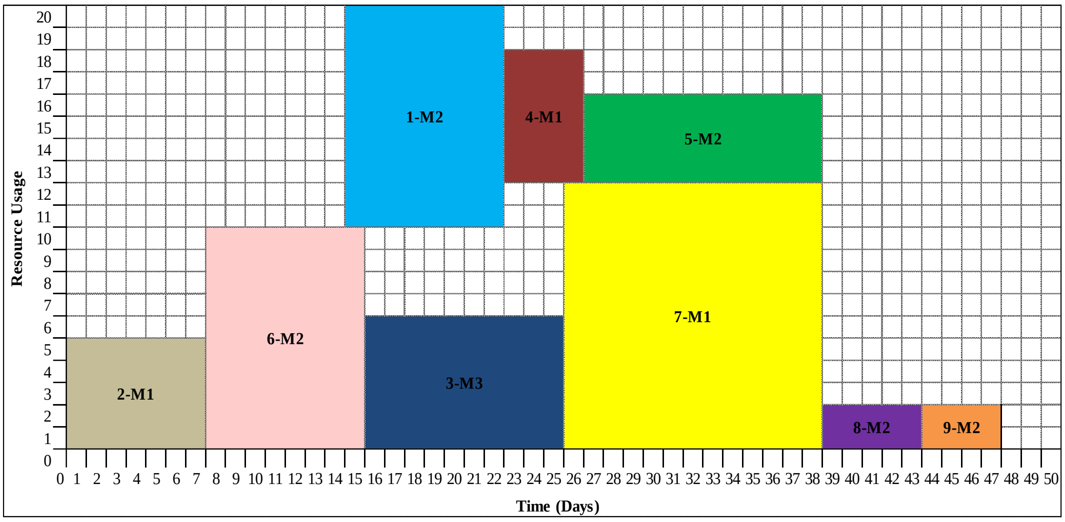


Figure 5.5. Resource Constrained Backward Schedule for Case Example

A priority list of <2, 6, 1, 3, 4, 7, 5, 8, 9> is obtained for the forward scheduling phase using the start times of the activities in the backward schedule of Figure 5.5. The forward scheduling is also performed by using the serial scheduling scheme, according to the priority list obtained from the backward scheduling phase, in order to improve the schedule obtained in the backward scheduling phase. The project duration of the resulting schedule (Schedule-1), has decreased to 40 days, at the end of forwards scheduling improvement as shown in Figure 5.6.

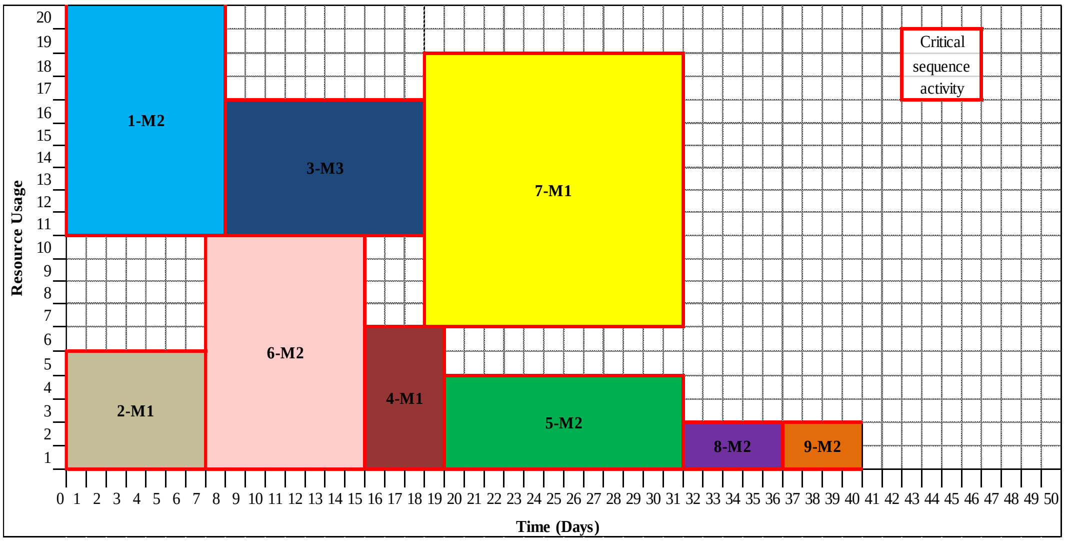


Figure 5.6. Critical Sequence for Schedule-1

In Schedule-1, all of the activities are identified to be on the critical sequence. The crashing options for the activities that are on a critical sequence in Schedule-1 are summarized in Table 5.1.

Table 5.1. Crashing Options for Activities on Critical Sequence in Schedule-1

Activity	Crashing option	Daily crashing cost ($/day)	Crashing resource difference
1	M2 to M1	2250	2
3	M3 to M2	750	3
5	M2 to M1	400	3
6	M2 to M1	1700	3
8	M2 to M1	1250	2
9	M2 to M1	550	4

Activity-5 is crashed first by changing the mode of this activity to Mode-1 (M-1), as this activity had the least daily crashing cost. The critical path method is performed to determine the floats of the activities for the new activity durations in which the mode of Activity-5 is changed to M-1. The next backward scheduling priority list is determined based on the revised floats obtained by the CPM. Backward scheduling and forward scheduling improvement are performed to obtain the next schedule (Schedule-2) as shown in Figure 5.7.

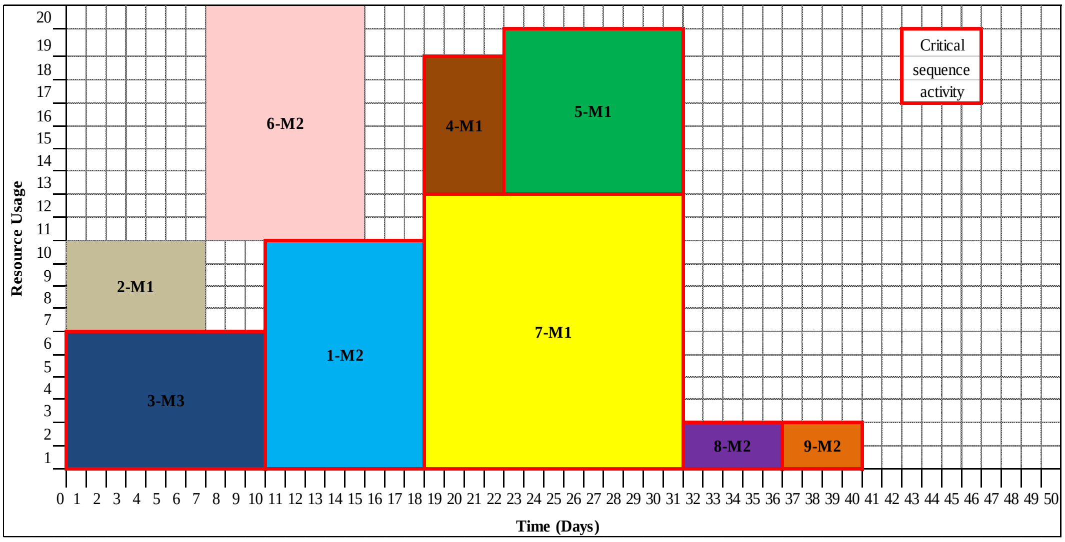


Figure 5.7. Critical Sequence for Schedule-2

The crashing options for the activities that are on critical sequence in Schedule-2 are given in Table 5.2.

Table 5.2. Crashing Options for Activities on Critical Sequence in Schedule-2

Activity	Crashing Option	Daily crashing cost ($/day)	Crashing resource difference
1	M2 to M1	2250	2
3	M3 to M2	750	3
8	M2 to M1	1250	2
9	M2 to M1	550	4

In Schedule-2, the Activity-2 and Activity-6 are not on the critical sequence. Hence, the next activity selected for crashing is Activity-9, and the procedure is repeated until all of the activities in the latest critical sequence(s) are considered for crashing. The proposed critical sequence crashing heuristic was able to achieve a minimum cost of $216,700 for the case example. The time/cost/resource modes and the start dates of the minimum cost solution are shown in Figure 5.8.

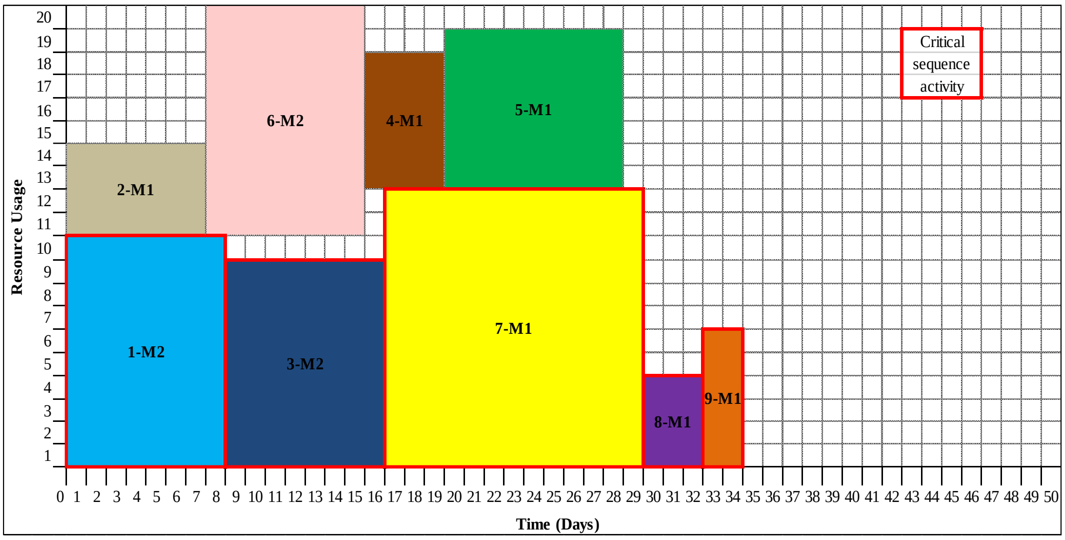
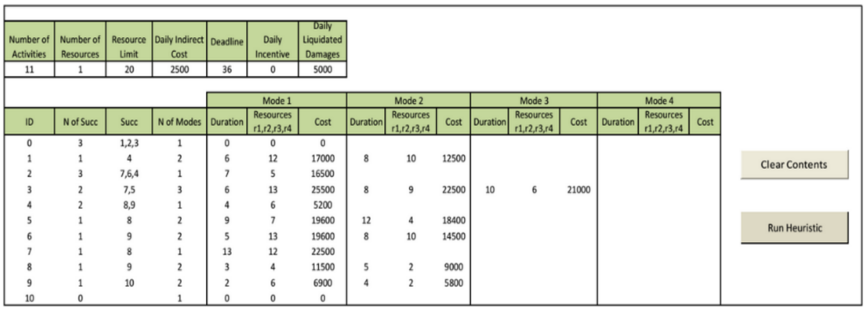


Figure 5.8. Minimum Cost Solution for the Case Example

5.3.4. Input / Output Interface

An input/output interface was developed in Microsoft Excel 2013 to enable simplified data input/output and to facilitate data exchange with the commercial project management software to enhance the use of the proposed critical sequence crashing heuristic in practice. The input screen of the interface, for the case example is illustrated in Figure 5.9. The heuristic requires a dummy start and a dummy finish activity. The successor information and time/cost/resource modes, the project deadline, the daily indirect cost, incentives, and the liquidated damages are entered in the input sheet of the interface.

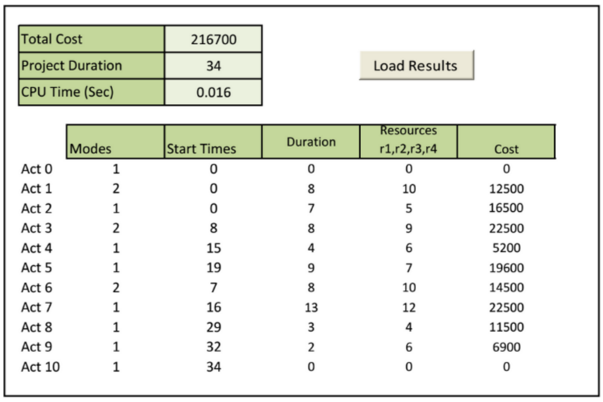


Number of Activities	Number of Resources	Resource Limit	Daily Indirect Cost	Deadline	Daily Incentive	Daily Liquidated Damages
11	1	20	2500	36	0	5000

				Mode 1			Mode 2			Mode 3			Mode 4		
ID	N of Succ	Succ	N of Modes	Duration	Resources r1,r2,r3,r4	Cost	Duration	Resources r1,r2,r3,r4	Cost	Duration	Resources r1,r2,r3,r4	Cost	Duration	Resources r1,r2,r3,r4	Cost
0	3	1,2,3	1	0	0	0									
1	1	4	2	6	12	17000	8	10	12500						
2	3	7,6,4	1	7	5	16500									
3	2	7,5	3	6	13	25500	8	9	22500	10	6	21000			
4	2	8,9	1	4	6	5200									
5	1	8	2	9	7	19600	12	4	18400						
6	1	9	2	5	13	19600	8	10	14500						
7	1	8	1	13	12	22500									
8	1	9	2	3	4	11500	5	2	9000						
9	1	10	2	2	6	6900	4	2	5800						
10	0		1	0	0	0									

Figure 5.9. Input Screen of the Input / Output Interface

Once the heuristic is executed, the time/cost/resource modes and the start dates of the activities, for the minimum cost solution that satisfies the resource constraints can be obtained in the output sheet as shown in Figure 5.10.



Total Cost	216700
Project Duration	34
CPU Time (Sec)	0.016

	Modes	Start Times	Duration	Resources r1,r2,r3,r4	Cost
Act 0	1	0	0	0	0
Act 1	2	0	8	10	12500
Act 2	1	0	7	5	16500
Act 3	2	8	8	9	22500
Act 4	1	15	4	6	5200
Act 5	1	19	9	7	19600
Act 6	2	7	8	10	14500
Act 7	1	16	13	12	22500
Act 8	1	29	3	4	11500
Act 9	1	32	2	6	6900
Act 10	1	34	0	0	0

Figure 5.10. Output Screen of the Input / Output Interface

5.4. Computational Experiments

Computational experiments are conducted to evaluate the performance of the proposed critical sequence crashing heuristic for the RCDTCTP, using benchmark instances. The proposed algorithm is coded in C# and compiled within Visual Studio 2013 on a 64 bit platform. All of the tests are carried out on a computer with an Intel Core i7-3.40 GHz CPU. Deviation from the upper bound (best known solution) is used to evaluate the performance of the different methods along with the CPU time. Deviation from the upper bound (PD) is calculated as Eq.(3.27) in which the Solution here is the minimum cost solution obtained that satisfies the resource constraints.

5.4.1. Small-scale Test Instances

The proposed heuristic is initially tested with the small-scale RCDTCTP test instances. The first test instance included a project, including nine activities with up to four modes and three resources (Leu & Yang, 1999). The problem is solved for the deadline of 64 days. CSCH obtained the best known solution of $7,400 in 0.03 seconds as shown in Table 5.3.

Table 5.3. Comparison of Results for Small Size Project-1 (Leu & Yang, 1999)

Source	Method	Solution *	PD (%)	CPU Time
Leu and Yang (1999)	Genetic algorithm	$7,400	0.00	NA
Hegazy and Menesi (2012)	Heuristic	$7,400	0.00	2 Sec
Menesi et al. (2013)	Constraint programming(CP)	$7,400	0.00	1 Sec
This study	CSCH	$7,400	0.00	0.03 Sec

* Solutions are for project deadline of 64 days
NA: Not available

The second small-scale RCDTCTP test instance, which is shown in Figure 5.11 consisted of a project, including ten activities up to four modes and a single resource

(Chen & Weng, 2009). The objective of the second problem was to determine the minimum total cost solution for an indirect expense of $2,200 per day, while considering a daily resource constraint of 30. The proposed heuristic achieved the best known solution of $244,000 in 0.02 seconds. The best solution was also obtained by the genetic algorithm of Chen and Weng (2009), which identified a Pareto front solution for the problem with an average processing time of 8 minutes.

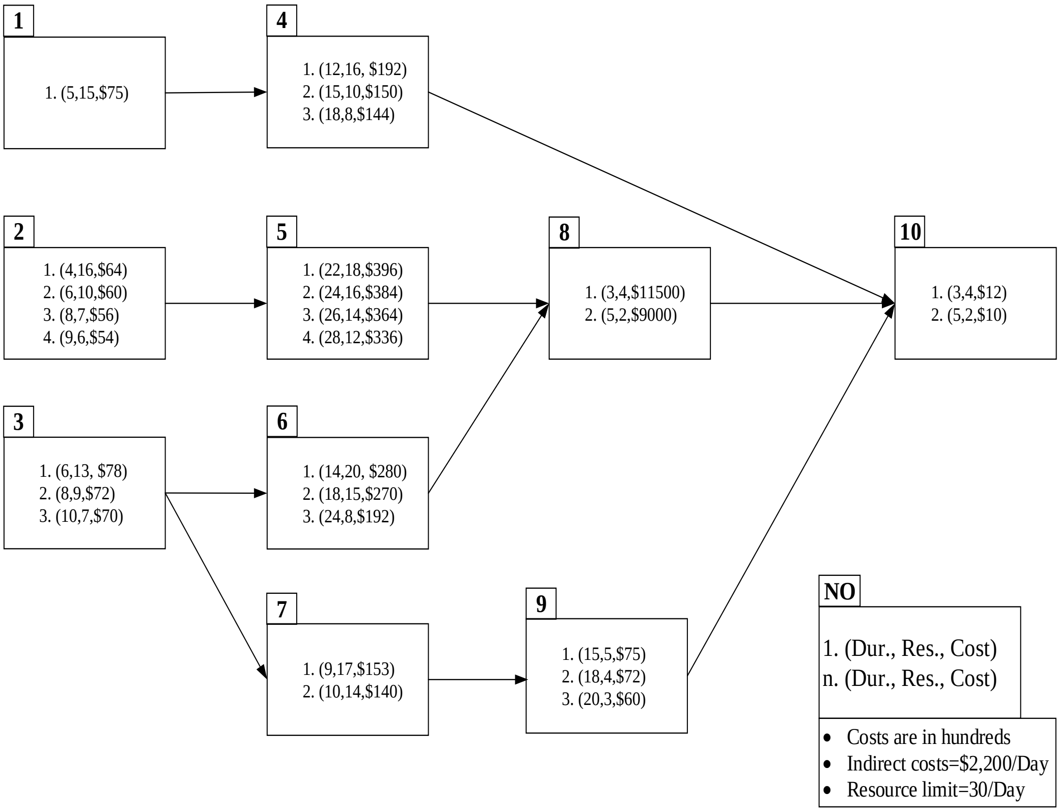


Figure 5.11. Small-Scale Test Instance-2 (Chen & Weng, 2009)

The constraint programming model presented by Menesi et al. (2013) also achieved the best known solution in one second. However, the heuristic of Hegazy and Menesi (2012) was able to obtain a solution of $245,900 in two seconds, which had a 0.78 % deviation from the upper bound. The performances of the four methods for the second problem are summarized in Table 5.4.

Table 5.4. Comparison of Results for Small Size Project-2 (Chen & Weng, 2009)

Source	Method	Solution	PD (%)	CPU Time
Chen and Weng (2009)	Genetic algorithm	$244,000	0.00	8 Min*
Hegazy and Menesi (2012)	Heuristic	$245,900	0.78	2 Sec
Menesi et al. (2013)	Constraint programming(CP)	$244,000	0.00	1 Sec
This study	CSCH	$244,000	0.00	0.02 Sec

* Average CPU time for pareto front optimization

5.4.2. Medium and Large-Scale Test Instances

Hegazy and Menesi (2012) and Menesi et al. (2013) created medium and large-scale test instances for the RCDTCTP by copying the test instance of Chen and Weng (2009) in serial several times. The test instances included 100, 300, 1,000 and 2,000 activities, and reflected the size of real-life construction projects.

Table 5.5 compares the performance of the proposed CSCH with the performance of the heuristic developed by Hegazy and Menesi (2012) and constraint programming model presented by Menesi et al. (2013). CSCH achieved a PD value of 0.22 and 0.24 for 100 and 300 activity problems in 0.03 and 0.20 seconds. The heuristic of Hegazy and Menesi (2012) obtained solutions with PD values of 0.78 for both of the problems in one and 21 minutes. The constraint programming model (Menesi et al., 2013) achieved a PD value of 2.34 in 15 seconds, and a PD value of 0.32 in 10 minutes, for the problem including 100 activities. For the problem, including 300 activities, the model was able to obtain a solution with a PD value of 5.90 in 15 seconds, and a solution with a PD value of 0.88 in 20 minutes. The proposed heuristic method achieved better solutions than the heuristic of Hegazy and Menesi (2012) and the constraint programming model (Menesi et al., 2013) at a significantly less computation time for medium-scale test instance, including 100 and 300 activities.

The performance of MASA was consistent for the large-scale test instances as shown in Table 5.6. The proposed heuristic achieved minimal deviations from the best known solutions with PD values of 0.24 and 0.25 for 1,000 and 2,000 activity problems in 4.64 and 33.34 seconds. For the problem, including 1,000 activities, the constraint programming model (Menesi et al., 2013) was able to obtain a solution with a PD value of 6.24 in 15 seconds, and a solution with a PD value of 4.18 in 120 minutes. The performance of the model for the problem, including 2,000 activities worsened and had the model obtained a solution with a PD value of 6.67 in 40 seconds, and a solution with a PD value of 6.39 in 120 minutes.

CSCH was able to determine a solution with a total cost of $48,919,400 for the project with 2,000 activities. The state-of-art methods could obtain a solution with a total cost of $51,916,400 for the same project. The proposed new heuristic enabled a potential cost saving in the amount of $2.997 Million by providing high quality solutions for the large size project. The proposed critical sequence crashing heuristic not only outperformed state-of-art methods, but was also able to achieve high quality solutions for the large-scale RCDTCTP within seconds for the first time.

Table 5.5. Comparison of Results for Medium Size Projects (Menesi et al., 2013)

	CP (Menesi et al., 2013)			Heuristic (Hegazy & Menesi, 2012)			CSCH (This study)		
Project Size	Solution	PD (%)	CPU Time	Solution	PD (%)	CPU Time	Solution	PD (%)	CPU Time
100 activities	$2.497,000	2.34	15 Sec	$2,459,000	0.78	1 Min	$2,445,400	0.22	0.03 Sec
	$2,452,900	0.53	5 Min						
	$2,447,900	0.32	10 Min						
300 activities	$7,751,700	5.90	15 Sec	$7,377,000	0.78	21 Min	$7,337,400	0.24	0.20 Sec
	$7,479,400	2.18	5 Min						
	$7,429,600	1.46	10 Min						
	$7,348,900	0.88	20 Min						

Table 5.6. Comparison of Results for Large Size Projects (Menesi et al., 2013)

	CP (Menesi et al., 2013)			CSCH (This study)		
Project Size	Solution	PD (%)	CPU Time	Solution	PD (%)	CPU Time
1000 activities	25,923,800	6.24	15 Sec	$24,459,400	0.24	4.64 Sec
		6.07	5 Min			
	25,571,700	4.80	20 Min			
	25,419,700	4.18	120 Min			
2000 activities	52,053,100	6.67	40 Sec	$48,919,400	0.25	33.35 Sec
	52,002,200	6.56	10 Min			
	51,969,200	6.50	30 Min			
	51,916,400	6.39	120 Min			

CHAPTER 6

CONCLUSIONS

Resource optimization is considered as one of the most crucial aspects of construction project scheduling for minimizing the project's cost. However, despite the importance of the resource optimization, very little success has been achieved in the studies that deal with the resource scheduling problems, especially for the large-scale projects. In addition, insufficiencies of the commonly used commercial project management software packages in coping with resource optimization have been repetitively mentioned in the literature. Resource leveling problem (RLP) and resource constraint discrete time-cost trade-off problem (RCDTCTP) are two of the important resource scheduling problems. RLP aims to minimize undesired fluctuations in resource utilization profiles and RCDTCTP determines the time/cost/resource options and start times of activities such that the precedence and resource constraints are satisfied and the total cost is minimized. Within the scope of this thesis, four optimization methods are developed, including a mixed-integer linear model for exact solutions, two different meta-heuristic algorithms for near optimal solutions of the RLPs, and one heuristic technique for the RCDTCTPs.

The first presented model is a mixed-integer linear model for solving the RLP to optimality, in which SSRR and ADIF metrics are used as the resource leveling objective functions. The model is implemented in GAMS/CPLEX solver environment. In order to provide a basis for performance evaluation of the proposed meta-heuristics, optimal solutions of J30 problem set of PSPLIB are obtained exercising the mixed-integer linear model.

For solving RLPs, a second model, a memetic algorithm with simulated annealing (MASA) is proposed. The optimization strategy of the MASA treats the individual learning as a separate process for local refinement. This method provides multiple contributions. First, it is adequately generic for solving the resource leveling problems incorporating any type of known objective function metrics. Second, it presents a novel optimization strategy which combines complementary searching strengths of the genetic algorithms, a shifting heuristic, and fine tuning abilities of the simulated annealing for the resource leveling problem under a memetic algorithm framework. Comparisons with the established commercial project management software and other state-of-art methods validated the effectiveness of the proposed approach. Third, it revealed the limitations of the popular commercial project management software for resource leveling. Finally, it provides solutions for well-known problem sets of PSPLIB using RID-MRD objective function metric for the first time in the literature which can be used as a benchmark for future studies. The computational experiments reveal that the optimization strategy of the MASA is able to obtain results of higher quality for the resource leveling problem compared to other existing methods.

To improve the effectiveness of MASA for the projects encompassing more activities and resources, the third method, a quasistable hybrid genetic algorithm (QHGA) is proposed. This method limits the searching space only to the solutions with quasi-stable schedules. QHGA is capable of minimizing the sum of squares of daily resource usage or total overloaded amount from average resource consumptions, for large-scale projects in a short computational time.

Three different experiment analyses are conducted to evaluate the performance of QHGA. First, the problem sets of PSPLIB with up to 120 activities and four resources are adopted to compare the performance of QHGA with other state-of-art methods within the relevant literature including MASA. The SSRR objective function metric is used through this comparisons. The QHGA obtained the best results in almost all of the instances, with the attained APD of zero. The results indicate that, as the size of the problems grow, the performance gap between QHGA and other methods increases. This distinguished feature of the QHGA enables practicing real-life large-size problems successfully.

The second type of experiments are conducted by generating problem instances up to 2000 activities from a known problem in the literature and comparing the quality of the solutions obtained by QHGA with the solutions provided by commercial project management software packages. Based on these experiments, the QHGA could surpass the heuristics of the commercial software programs by a huge margin within the same periods of computational time. This performance gap once again revealed the resource leveling limitations of the commercial project management software packages.

The performance of the QHGA is also evaluated using one real case construction project data. Within the same computational time of few seconds, QHGA achieved better solutions than all heuristics of commercial project management software packages. The impact of the individual resource peaks over the project's cost and the influence of employing the QHGA are also studied. The QHGA enabled significant indirect cost saving by adequate scheduling of the resource requirements. The computational experiments proved the robustness of QHGA compared to the existing methods and the commonly used commercial software programs.

The QHGA is also integrated to Microsoft Project in order to obtain a simplified application, and to improve Microsoft Project's capabilities in dealing with RLP. The performance gap between the QHGA and leveling heuristics of popular project management software reveals the potential for improving the heuristics of popular project management software for resource leveling. QHGA provides an efficient leveling alternative for practitioners which can be used along with other popular project management software for achieving optimal resource planning and management decision.

The final proposed algorithm within the context of this thesis is a critical sequence crashing heuristic which is designed and developed to achieve fast and high quality solutions for the large-scale RCDTCT problems. In the proposed heuristic, backward-forward scheduling technique is adopted and crashing of the activities on the critical sequence are considered to present an effective method for the resource constrained discrete time-cost trade-off problem. The computational tests reveal that the new heuristic is capable of finding competent results for small, medium, and large-scale projects with project deadlines and resource constraints, and outperformed other state-of-art methods with respect to both solution quality and computation time requirement. High quality solutions with minor deviations from the best known solutions are obtained within seconds for the large-scale resource constrained discrete time-cost trade-off problem, for the first time. The main contribution of the new heuristic is that it provides adequate solutions for the real-life-size projects within seconds, and enables significant savings during planning of construction projects with project deadlines and resource constraints.

Although the MASA could reach good results for instances with up to 120 activities and four resources within reasonable computing time, its computational time requirement to achieve an adequate solution notably increases for larger size problems. To improve the effectiveness of the MASA for the projects including larger number of activities and resources, utilization of parallel computing techniques appears to be a promising area for future research. The QHGA is also able to propose robust solutions for leveling of real-life-size problems in a very short computational time. However, its applicability is only limited to SSRR and OVLD objective functions of RLP, since the practicality of the quasistable schedule is not approved for other types of objective function metrics. Extension of the QHGA for other known metrics, particularly for RID as a more practical objective function, seems to be a very encouraging area for the future improvement of this algorithm. For the proposed critical sequence crashing heuristic, the large size problem instances are adopted to evaluate its capabilities in solving the RCDTCT problems. However, the instances do not fully reflect the complexity of the real-life construction projects, since they are series of small networks. Therefore, the performance of this heuristic might diminish for more complex problem instances. The quality of the solutions of the proposed heuristic can be improved by removal of local suboptimalities or by consideration of multipass methods during resource constrained scheduling, and by inclusion of activities that are not on the critical sequence in crashing, but these improvements will come at the expense of increased computational time.

REFERENCES

Afshar, A., Ziaraty, A. K., Kaveh, A., & Sharifi, F. (2009). Nondominated archiving multicolony ant algorithm in time–cost trade-off optimization. *Journal of Construction Engineering and Management, 135*(July), 668–674.

Ahn, T., & Erenguc, S. S. (1998). The resource constrained project scheduling problem with multiple crashable modes: A heuristic procedure. *European Journal of Operational Research*.

Alsayegh, H., & Hariga, M. (2012). Hybrid meta-heuristic methods for the multi-resource leveling problem with activity splitting. *Automation in Construction*, *27*, 89–98.

Ballestín, F., Schwindt, C., & Zimmermann, J. (2007). Resource leveling in make-to-order production: Modeling and heuristic solution method. *International Journal of Operation Research, 4*(1), 50–62.

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

Bettemir, Ö. (2009). *Optimization of time–cost–resource trade-off problems in project scheduling using meta-heuristic algorithms*. PhD thesis, Middle East Technical University, Ankara, Turkey.

Bettemir, Ö., & Sonmez, R. (2014). Hybrid genetic algorithm with simulated annealing for resource-constrained project scheduling. *Journal of Management in Engineering*, (10.1061/(ASCE)ME.1943-5479.0000323 , 04014082), 1–8.

Blazewicz, J., Lenstra, J. K., & Kan, A. H. G. R. (1983). Scheduling subject to resource constraints: Classification and complexity. *Discrete Applied Mathematics*, *5*, 11–24.

Blum, C., & Roli, A. (2008). Hybrid metaheuristics: An introduction. *Studies in Computational Intelligence*, *114*, 1–30.

Bouleimen, K., & Lecocq, H. (2003). A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. *European Journal of Operational Research*, *149*, 268–281.

Burgess, A. R., & Killebrew, J. B. (1962). Variation in activity level on a cyclical arrow diagram. *The Journal of Industrial Engineering*, *XIII*(2), 76–83.

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

Chen, P. H., & Shahandashti, S. M. (2009). Hybrid of genetic algorithm and simulated annealing for multiple project scheduling with multiple resource constraints. *Automation in Construction*.

Chen, P. H., & Weng, H. (2009). A two-phase GA model for resource-constrained project scheduling. *Automation in Construction*, *18*(4), 485–498.

Chen, R. M. (2011). Particle swarm optimization with justification and designed mechanisms for resource-constrained project scheduling problem. *Expert Systems with Applications*, *38*(6), 7102–7111.

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.

Chua, D. K. H., Chan, W. T., & Govindan, K. (1997). A time-cost trade-off model with resource consideration using genetic algorithm. *Civil Engineering Systems*, *14*(March 2015), 291–311.

De, P., Dunne, E. J., Ghosh, J. B., & Wells, C. E. (1997). Complexity of the discrete time-cost tradeoff problem for project networks. *Operations Research*, *45*(2), 302–306.

Deblaere, F., Demeulemeester, E., & Herroelen, W. (2011). RESCON: Educational project scheduling software. *Computer Applications in Engineering Education*, *19*, 327–336.

Demeulemeester, E. (1995). Minimizing resource availability costs in time-limited project networks. *Management Science*, *41*(10), 201–212.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *26*(1), 29–41.

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

Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*.

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

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

Fallah-Mehdipour, E., Haddad, O. bozorg, Rezapour Tabari, M. M., & Mariño, M. a. (2012). Extraction of decision alternatives in construction management projects: Application and adaptation of NSGA-II and MOPSO. *Expert Systems with Applications*, *39*(3), 2794–2803.

Feng, C.-W., Liu, L., & Burns, S. a. (1997). Using genetic algorithms to solve construction time-cost trade-off problems. *Journal of Computing in Civil Engineering*, *11*, 184–189.

Galbreath, R. (1965). Computer program for leveling resource usage. *Journal of the Construction Division*, *91*(1), 107–124.

Gather, T., Zimmermann, J., & Bartels, J. H. (2010). Exact methods for the resource levelling problem. *Journal of Scheduling*, *14*(6), 557–569.

Geng, J., Weng, L., & Liu, S. (2010). An improved ant colony optimization algorithm for nonlinear resource-leveling problems. *Computers & Mathematics with Applications*.

Goldratt, E. M. (1997). *Critical Chain*. *America* (Vol. 8).

Guo, Y., Li, N., & Ye, T. (2009). Multiple resources leveling in multiple projects scheduling problem using particle swarm optimization. In *5th International Conference on Natural Computation, ICNC 2009* (Vol. 3, pp. 260–264).

Hajek, B. (1988). Cooling schedules for optimal annealing. *Mathematics of Operations Research*.

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

Harris, R. (1978). *Precedence and arrow networks for construction*. New York: John Wiley and Sons.

Hartmann, S. (1998). A competitive genetic algorithm for resource-constrained project scheduling. *Naval Research Logistics*, *45*, 733–750.

Hegazy, T. (1999). Optimization of resource allocation and levelling using genetic algorithms. *Journal of Construction Engineering and Management*, *125*(June), 167–175.

Hegazy, T., & Menesi, W. (2012). Heuristic method for satisfying both deadlines and resource constraints. *Journal of Construction Engineering and Management*, *138*(June), 688–696.

Hegazy, T., Shabeeb, A. K., Elbeltagi, E., & Cheema, T. (2000). Algorithm for scheduling with multiskilled constrained resources. *Journal of Construction Engineering and Management*, *126*(December), 414–421.

Hekimoglu, O. (2007). *Comparison of the Resource Allocation Capabilities of Project Management Software Packages in Resource Constrained Project Scheduling Problems*. Master's thesis, Middle East Technical University, Ankara, Turkey.

Herroelen, W., & Leus, R. (2005). Project scheduling under uncertainty: Survey and research potentials. In *European Journal of Operational Research* (Vol. 165, pp. 289–306).

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

Hiyassat, M. A. (2001). Applying modified minimum moment method to multiple resource leveling. *Journal of Construction Engineering and Management*, *127*(3), 192–198.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. *Ann Arbor MI University of Michigan Press* (Vol. Ann Arbor).

Hwang, S. F., & He, R. S. (2006). Improving real-parameter genetic algorithm with simulated annealing for engineering problems. *Advances in Engineering Software*, *37*(6), 406–418.

Iranagh, M. A., Atan, T., & Sonmez, R. (2013). A mixed-integer linear model for optimization of resource idle days in project scheduling. In *Creative Construction Conference* (pp. 368–381). Budapest, Hungary.

Iranagh, M. A., & Sonmez, R. (2012). A genetic algorithm for resource leveling of construction projects. In *28th Annual ARCOM Conference* (pp. 1047–1055). Edinburgh, UK.

Johnson, R. V. (1992). Resource constrained scheduling capabilities of commercial project management software. *Project Management Institute*, *22*(4), 39–43.

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.

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

Karshenas, S., & Haber, D. (1990). Economic optimization of construction project scheduling. *Construction Management and Economics*, *8*(2), 135–146.

Kartam, N., & Tongthong, T. (1998). An artificial neural network for resource leveling problems. *Artificial Intelligence For Engineering Design Analysis And Manufacturing, 12*(03), 273–287.

Kastor, A., & Sirakoulis, K. (2009). The effectiveness of resource levelling tools for resource constraint project scheduling problem. *International Journal of Project Management, 27*(5), 493–500.

Kelley, J. E. J. (1963). The critical path method: Resources planning and scheduling. *Industrial Scheduling, 13*, 347–365.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks, 4*. doi:10.1109/ICNN.1995.488968

Kim, J.-L., & Ellis, R. D. (2008). Permutation-based elitist genetic algorithm for optimization of large-sized resource-constrained project scheduling. *Journal of Construction Engineering and Management, 134*(November), 904–913.

Kim, J.-L., & Ellis, R. D. (2010). Comparing schedule generation schemes in resource-constrained project scheduling using elitist genetic algorithm. *Journal of Construction Engineering and Management*, *136*(February), 160–169.

Kolisch, R., & Hartmann, S. (2006). Experimental investigation of heuristics for resource-constrained project scheduling: An update. *European Journal of Operational Research*, *174*(1), 23–37.

Kolisch, R., Schwindt, A., & Sprecher, A. (1999). Benchmark Instances for project scheduling problems. *Operations Research and Management Science*, *14*, 197–212.

Kolisch, R., & Sprecher, A. (1997). PSPLIB – A project scheduling problem library. *European Journal of Operational Research*, *96*, 205–216.

Lee, J.-K., & Kim, Y.-D. (1996). Search heuristics for resource constrained project scheduling. *Journal of the Operational Research Society, 47*(5), 678–689.

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

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

Li, R. K.-Y., & Willis, R. J. (1993). Resource constrained scheduling within fixed project durations. *Journal of the Operational Research Society*, *44*(1), 71–80. doi:10.1057/jors.1993.8

Liberatore, M. J., Pollack-Johnson, B., & Smith, C. A. (2001). Project management in construction: software use and research directions. *Journal of Construction Engineering and Management*, *127*(2), 101–107.

Lova, A., & Tormos, P. (2001). Analysis of scheduling schemes and heuristic rules performance in resource-constrained multiproject scheduling. *Annals of Operations Research*, *102*, 263–286.

Lu, M., Lam, H.-C., & Dai, F. (2008). Resource-constrained critical path analysis based on discrete event simulation and particle swarm optimization. *Automation in Construction*, *17*, 670–681.

Lu, M., & Li, H. (2003). Resource-activity critical-path method for construction planning. *Journal of Construction Engineering and Management*, *129*(4), 412–420.

Maroto, C., & Tormos, P. (1994). Project management and evaluation of software quality. *International Transaction in Operational Research*, *1*(2), 209–221.

Martinez, J., & Ioannou, P. (1993). Resource leveling based on the modified minimum moment heuristic. In *5th Int. Conf. Computing in Civ. and Build. Engrg* (pp. 287–294). ASCE, Reston, Va.

Mason, A. T., & Moodie, C. L. (1971). A branch and bound algorithm for minimizing cost in project scheduling. *Management Science*, *18*(4), 158–173.

Mattila, K. G., & Abraham, D. M. (1998). Resource leveling of linear schedules using integer linear programing. *Journal of Construction Engineering and Management*, *124*(3), 232–244.

Mellentien, C., & Trautmann, N. (2001). Resource allocation with project management software. *OR Spektrum*, *23*, 383–394.

Menesi, W., Golzarpoor, B., & Hegazy, T. (2013). Fast and near-optimum schedule optimization for large-scale projects. *Journal of Construction Engineering and Management*, *139*, 1117–1124.

Moscato, P., & Norman, M. G. (1992). A memetic approach for the traveling salesman problem implementation of a computational ecology for combinatorial optimization on message-passing systems. *Parallel Computing and Transputer Applications*, *28*(1), 177–186.

Mutlu, M. C. (2010). *A branch and bound algorithm for resource leveling problem*. Master's thesis, Middle East Technical University.

Neri, F., Cotta, C., & Moscato, P. (2012). *Handbook of memetic algorithms*. *Studies in computational intelligence*.

Neumann, K., Nübel, H., & Schwindt, C. (2000). Active and stable project scheduling. *Mathematical Methods of Operations Research*, *52*(3), 441–465.

Neumann, K., Schwindt, C., & Zimmermann, J. (2003). *Project Scheduling with Time Windows and Scarce Resources*. Berlin: Springer.

Neumann, K., & Zimmermann, J. (1999). Resource levelling for projects with schedule-dependent time windows. *European Journal of Operational Research*, *117*(3), 591–605.

Neumann, K., & Zimmermann, J. (2000). Procedures for resource leveling and net present value problems in project scheduling with general temporal and resource constraints. *European Journal of Operational Research*, *127*(2), 425–443.

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

Nguyen, Q. H., Ong, Y. S., & Lim, M. H. (2009). A probabilistic memetic framework. *IEEE Transactions on Evolutionary Computation*, *13*(3), 604–623.

Nübel, H. (2001). The resource renting problem subject to temporal constraints. *OR Spektrum*, *23*, 359–381.

Özdamar, L., & Ulusoy, G. (1994). A local constraint based analysis approach to project scheduling under general resource constraints. *European Journal of Operational Research*, *79*, 287–298.

Pang, N., Shi, Y., & You, Y. (2008). Resource leveling optimization of network schedule based on particle swarm optimization with constriction factor. In *Proceedings - 2008 International Conference on Advanced Computer Theory and Engineering, ICACTE 2008* (pp. 652–656).

Petrovic, R. (1969). On optimization of resource leveling in project plans. *In: Lombaers, H.J. (Ed.), Project Planning by Network Analysis: Pro- ceedings of the 2nd International Congress* (pp. 268–273). North-Hol land, Amsterdam.

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.

Qi, J. X., Wang, Q., & Guo, X. Z. (2007). Improved particle swarm optimization for resource leveling problem. *In Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, ICMLC 2007* (pp. 896–901). Hong Kong.

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

Roca, J., Pugnaghi, E., & Libert, G. (2008). Solving an extended resource leveling problem with multiobjective evolutionary algorithms. *International Journal of Computational Intelligence*, *4*(4), 289–300.

Rudolph, G. (1994). Convergence analysis of canonical genetic algorithms. *IEEE Transactions on Neural Networks*, *5*(1), 96–101.

Savin, D., Alkass, S., & Fazio, P. (1996). Construction resource leveling using neural networks. *Canadian Journal of Civil Engineering*, *23*(4), 917–925.

Savin, D., Alkass, S., & Fazio, P. (1997). A procedure for calculating the weight-matrix of a neural network for resource leveling. *Advances in Engineering Software*, *28*(5), 277–283.

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

Shah, K. A., Farid, F., & Baugh, J. W. (1993). Optimal resource leveling using integer linear programming. In *4th International Conference on Computing in Civil and Building Engineering* (pp. 501–508).

Siemens, N. (1971). A simple cpm time-cost tradeoff algorithm. *Management Science*.

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.

Sonmez, R., & Bettemir, Ö. H. (2012). A hybrid genetic algorithm for the discrete time-cost trade-off problem. *Expert Systems with Applications*, *39*, 11428–11434.

Sonmez, R., & Uysal, F. (2014). Backward-forward hybrid genetic algorithm for resource-constrained multiproject scheduling problem. *Journal of Computing in Civil Engineering*, (10.1061/(ASCE)CP.1943-5487.0000382, 04014072).

Stevens, J. D. (1990). *Techniques for Construction Network Scheduling*. New York: Mc Graw-Hill.

Tormos, P., & Lova, A. (2001). A competitive heuristic solution technique for resource-constrained project scheduling. *Annals of Operations Research*, *102*, 65–81.

Trautmann, N., & Baumann, P. (2009). Resource-allocation capabilities of commercial project management software: An experimental analysis. *2009 International Conference on Computers & Industrial Engineering*, 1143–1148.

Valls, V., Ballestín, F., & Quintanilla, S. (2005). Justification and RCPSP: A technique that pays. *European Journal of Operational Research*, *165*, 375–386.

Vanhoucke, M., & Debels, D. (2007). The discrete time/cost trade-off problem: Extensions and heuristic procedures. *Journal of Scheduling*, *10*, 311–326.

Vanhoucke, W., Demeulemeester, M., & Herroelen, E. (2001). On maximizing the net present value of a project under renewable resource constraints. *Management Science*, *47*(8), 1113–1121.

Wang, Q., & Qi, J. (2009). Improved particle swarm optimization for rcp scheduling problem (pp. 49–57). Berlin Heidelberg: Springer-Verlag.

Wiest, J. D. (1964). Some properties of schedules for large projects with limited resources. *Operations Research*.

Woodworth, M. W., & Willie, C. J. (1975). A heuristic algorithm for resource leveling in multi-project, multi resource scheduling. *Decision Sciences*, *6*(3), 525–540.

Wuliang, P., & Chengen, W. (2009). A multi-mode resource-constrained discrete time-cost tradeoff problem and its genetic algorithm based solution. *International Journal of Project Management, 27,* 600–609.

Xiong, Y., & Kuang, Y. P. (2006). Ant colony optimization algorithm for resource leveling problem of construction project. In *The CRIOCM 2006 International Symposium on "Advancement of Construction Management and Real Estate"* (pp. 384–391).

Xiong, Y., & Kuang, Y. P. (2008). Applying an ant colony optimization algorithm-based multiobjective approach for time–cost trade-off. *Journal of Construction Engineering and Management*, *134*(February), 153–156.

Yang, I.-T. (2007). Using elitist particle swarm optimization to facilitate bicriterion time-cost trade-off analysis. *Journal of Construction Engineering and Management*.

Yeniocak, H. (2013). *An efficient branch and bound algorithm for the resource leveling problem*. Master's thesis, Middle East Technical University.

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

Zheng, D. X. M., Ng, S. T., & Kumaraswamy, M. M. (2005). Applying pareto ranking and niche formation to genetic algorithm-based multiobjective time–cost optimization. *Journal of Construction Engineering and Management*.

Zheng, D. X. M., Ng, T., & Kumaraswamy, M. M. (2003). GA-based multiobjective technique for multi-resource leveling. *Construction Research 2003*, *120*, 29.

APPENDIX A

SSRR SOLUTIONS OBTAINED BY MASA AND QHGA FOR PSPLIB INSTANCES

All of the tests are carried out on a computer with a 3.00 GHz Core 2 Duo Processor E8400 Intel CPU. The stopping criteria for MASA is defined as 500,000 schedule and the computational time for each problem is set as the stopping criteria for that problem in QHGA. Weights of all the resources are taken 1. Optimal results of J30 instances are defined by the mixed integer-linear programing model presented in Chapter 3 within a time limit of five hours for each problem. 475 problems out of 480 could solve optimally except problems j3013_10, j3015_6, j3030_2, j3031_2, j3045_6.

Table A.1. J30 Instances' SSRR Solutions for RLP (1/6)

Instance	Optimal	Time (S)	MASA	QHGA	Instance	Optimal	Time (S)	MASA	QHGA
j301_1	7485	9.1	7485	7485	j303_1	8830	14.1	8830	8830
j301_2	8811	9.5	8821	8811	j303_2	7664	8.6	7664	7664
j301_3	6043	9.1	6055	6043	j303_3	6691	12.2	6705	6691
j301_4	11390	11.7	11398	11392	j303_4	8529	17.3	8529	8529
j301_5	5661	7.1	5709	5709	j303_5	9004	10.5	9004	9004
j301_6	6290	8.2	6290	6290	j303_6	5988	11.0	5988	5988
j301_7	6609	11.5	6609	6609	j303_7	6539	9.9	6539	6539
j301_8	8210	11.0	8210	8210	j303_8	5545	11.0	5545	5545
j301_9	9395	9.3	9395	9395	j303_9	7714	11.6	7714	7714
j301_10	6403	8.5	6427	6403	j303_10	8087	11.7	8119	8087
j302_1	6974	8.2	6974	6974	j304_1	6729	9.8	6729	6729
j302_2	7658	10.5	7658	7658	j304_2	7160	12.3	7232	7160
j302_3	6523	10.0	6523	6539	j304_3	6891	10.5	6905	6903
j302_4	6720	9.3	6730	6730	j304_4	10060	11.6	10060	10060
j302_5	7039	10.7	7039	7039	j304_5	7770	12.1	7774	7770
j302_6	6287	9.7	6287	6287	j304_6	9514	9.8	9544	9514
j302_7	7673	10.2	7759	7673	j304_7	7513	12.7	7513	7513
j302_8	6437	10.5	6437	6437	j304_8	7490	11.6	7536	7490
j302_9	12642	11.6	12642	12642	j304_9	8976	8.7	8976	8976
j302_10	8251	9.1	8251	8251	j304_10	7330	10.4	7330	7330

Table A.2. J30 Instances' SSRR Solutions for RLP (2/6)

Instance	Optimal	Time (S)	MASA	QHGA	Instance	Optimal	Time (S)	MASA	QHGA
j305_1	18234	11.3	18238	18286	j3010_1	50453	11.0	50563	50453
j305_2	26028	13.4	26058	26042	j3010_2	48117	13.2	48435	48147
j305_3	23838	13.2	23840	23838	j3010_3	46671	15.1	46757	46963
j305_4	32084	11.4	32084	32084	j3010_4	51089	13.5	51089	51131
j305_5	23999	13.6	24089	24013	j3010_5	52892	11.0	52918	52962
j305_6	29442	11.4	29442	29442	j3010_6	51976	11.0	51986	51986
j305_7	29246	11.0	29246	29350	j3010_7	50432	12.1	50518	50500
j305_8	25641	12.7	25823	25717	j3010_8	48686	13.1	48694	48702
j305_9	24267	9.9	24267	24365	j3010_9	41315	12.3	41639	41429
j305_10	21544	13.4	21614	21614	j3010_10	62934	10.1	62934	62934
j306_1	29204	14.0	29280	29204	j3011_1	38058	13.0	38072	38058
j306_2	24984	12.1	25062	24986	j3011_2	67003	15.4	67371	67177
j306_3	30164	11.7	30164	30216	j3011_3	33562	18.8	34514	33598
j306_4	37950	10.0	37950	37950	j3011_4	35977	15.2	35993	36059
j306_5	26963	14.4	27091	27003	j3011_5	36929	12.4	37165	37021
j306_6	25546	9.1	25546	25598	j3011_6	51603	12.0	51651	51603
j306_7	24536	10.6	24588	24554	j3011_7	71662	10.5	71662	71662
j306_8	31890	10.1	31890	31890	j3011_8	46970	15.5	47016	47002
j306_9	19699	11.7	19699	19729	j3011_9	29877	15.6	30073	29921
j306_10	34981	14.1	34981	35057	j3011_10	35661	10.4	35707	35771
j307_1	14698	12.9	14794	14850	j3012_1	50580	12.7	50580	50626
j307_2	28294	10.5	28312	28294	j3012_2	60157	12.5	60157	60285
j307_3	22713	10.2	22741	22713	j3012_3	47444	10.6	47520	47444
j307_4	22745	11.0	22777	22749	j3012_4	56508	15.6	56616	56508
j307_5	31411	10.7	31411	31411	j3012_5	53885	12.1	54083	53935
j307_6	22334	9.3	22344	22334	j3012_6	32255	13.3	32395	32255
j307_7	18886	12.3	19548	18900	j3012_7	53360	14.3	53360	53368
j307_8	22606	11.2	22606	22606	j3012_8	71284	9.7	71284	71284
j307_9	16744	12.7	16744	16744	j3012_9	58690	13.6	58946	58792
j307_10	26856	10.5	26856	26876	j3012_10	50031	14.1	50031	50063
j308_1	23759	11.1	23759	23759	j3013_1	83771	9.5	83789	83785
j308_2	23093	12.9	23093	23093	j3013_2	80492	9.0	80492	80492
j308_3	18785	13.0	18785	18801	j3013_3	79552	12.2	79612	79604
j308_4	15608	11.6	15794	15608	j3013_4	60778	12.7	61130	60892
j308_5	23901	14.1	23995	23959	j3013_5	81257	11.6	81257	81257
j308_6	24447	12.1	24475	24503	j3013_6	77827	12.0	78101	77827
j308_7	35994	10.6	35994	35996	j3013_7	69786	13.1	70012	69914
j308_8	18263	12.2	18303	18327	j3013_8	102449	13.7	102627	102897
j308_9	16314	10.1	16314	16362	j3013_9	58996	13.4	59020	59130
j308_10	14725	15.0	14839	14793	j3013_10	40819	12.9	41543	41001
j309_1	47813	13.9	47853	47951	j3014_1	62561	11.4	62561	62607
j309_2	52275	12.0	52331	52275	j3014_2	73683	12.3	74437	73981
j309_3	48416	12.7	48416	48422	j3014_3	74176	15.1	74858	74466
j309_4	45349	13.3	45409	45437	j3014_4	62980	11.7	63082	62980
j309_5	30487	13.0	30661	30581	j3014_5	64880	12.5	64880	65024
j309_6	28824	11.8	29124	29070	j3014_6	71714	9.8	71780	71940
j309_7	40190	11.1	40220	40214	j3014_7	82727	12.6	82835	82857
j309_8	55649	14.1	55691	55767	j3014_8	70197	13.6	70471	70391
j309_9	41604	10.2	41612	41604	j3014_9	60629	11.5	60731	60677
j309_10	42869	15.2	43043	42869	j3014_10	66017	14.6	66017	66459

Table A.3. J30 Instances' SSRR Solutions for RLP (3/6)

Instance	Optimal	Time (S)	MASA	QHGA	Instance	Optimal	Time (S)	MASA	QHGA
j3015_1	71948	12.5	72118	72210	j3020_1	10073	11.6	10073	10073
j3015_2	72820	12.6	72826	72820	j3020_2	8447	13.1	8447	8447
j3015_3	60271	12.3	60659	60427	j3020_3	9498	10.7	9498	9498
j3015_4	63267	12.7	63365	63463	j3020_4	8445	9.5	8445	8445
j3015_5	99961	14.8	100103	99969	j3020_5	6768	12.0	6768	6768
j3015_6	70906	16.3	71092	70984	j3020_6	6498	10.9	6498	6498
j3015_7	56064	12.3	56082	56064	j3020_7	5308	9.2	5344	5308
j3015_8	66809	11.9	66901	66907	j3020_8	9494	10.5	9494	9618
j3015_9	54707	13.5	54783	54945	j3020_9	6691	9.4	6691	6691
j3015_10	54142	16.1	54252	54366	j3020_10	8384	8.6	8384	8384
j3016_1	62333	13.4	62585	62437	j3021_1	18221	14.1	18221	18221
j3016_2	80184	12.8	80308	80300	j3021_2	22587	11.0	22627	22619
j3016_3	95492	10.0	95492	95680	j3021_3	19556	13.3	19590	19572
j3016_4	45994	12.2	46034	46112	j3021_4	20841	12.2	20885	20865
j3016_5	64875	13.5	64875	64879	j3021_5	21007	10.7	21007	21007
j3016_6	50837	13.1	50985	50891	j3021_6	26361	12.9	26395	26395
j3016_7	75590	9.9	75672	75702	j3021_7	18685	12.5	18743	18713
j3016_8	57693	11.5	57861	57857	j3021_8	24671	11.5	24671	24671
j3016_9	63056	11.5	63236	63304	j3021_9	30706	10.6	30706	30718
j3016_10	62878	13.1	63322	62952	j3021_10	20291	13.3	20291	20291
j3017_1	15609	10.4	15609	15609	j3022_1	21909	10.1	21909	21909
j3017_2	5998	12.0	5998	5998	j3022_2	23765	10.1	23765	23785
j3017_3	6024	11.8	6024	6024	j3022_3	15109	13.8	15149	15129
j3017_4	8453	9.1	8453	8453	j3022_4	29356	10.2	29356	29356
j3017_5	9458	8.1	9458	9458	j3022_5	19864	11.2	19864	19954
j3017_6	6781	12.1	6801	6781	j3022_6	30097	11.5	30097	30097
j3017_7	8692	10.7	8692	8692	j3022_7	24087	13.0	24191	24091
j3017_8	8667	10.7	8667	8667	j3022_8	27972	11.8	27972	27972
j3017_9	5590	9.0	5590	5590	j3022_9	15492	15.6	15498	15492
j3017_10	7234	13.6	7234	7234	j3022_10	29407	12.8	29407	29523
j3018_1	9339	10.1	9339	9339	j3023_1	19553	14.4	19565	19561
j3018_2	7613	10.4	7613	7613	j3023_2	16801	12.3	16819	16809
j3018_3	6151	11.1	6151	6151	j3023_3	20804	11.2	20804	20826
j3018_4	8950	13.5	8950	8950	j3023_4	24863	14.2	24869	24869
j3018_5	9770	10.7	9770	9770	j3023_5	18032	11.0	18032	18032
j3018_6	11970	10.9	11970	11970	j3023_6	20930	11.3	20930	20930
j3018_7	5770	10.0	5770	5770	j3023_7	17252	13.2	17276	17252
j3018_8	4435	10.6	4443	4435	j3023_8	30343	12.0	30399	30509
j3018_9	7781	9.7	7781	7781	j3023_9	30678	13.8	30678	30730
j3018_10	6947	9.4	6947	6947	j3023_10	14103	13.6	14143	14113
j3019_1	9348	8.7	9348	9348	j3024_1	16555	12.1	16555	16581
j3019_2	7421	11.7	7621	7421	j3024_2	21757	12.9	21757	21845
j3019_3	7572	15.6	7572	7572	j3024_3	29779	16.3	29779	29779
j3019_4	5614	8.8	5614	5614	j3024_4	20357	12.3	20357	20419
j3019_5	7466	10.4	7470	7466	j3024_5	17431	12.4	17431	17439
j3019_6	5587	9.4	5587	5587	j3024_6	26307	13.2	26629	26333
j3019_7	6501	11.0	6501	6501	j3024_7	16864	10.8	16864	16902
j3019_8	7170	11.7	7200	7170	j3024_8	26475	10.0	26475	26501
j3019_9	7280	8.9	7280	7280	j3024_9	23824	11.2	23824	23824
j3019_10	5902	9.5	5902	5902	j3024_10	23328	12.5	23418	23348

Table A.4. J30 Instances’ SSRR Solutions for RLP (4/6)

Instance	Optimal	Time (S)	MASA	QHGA
j3025_1	33258	15.2	33296	33326
j3025_2	37178	11.2	37178	37178
j3025_3	56409	11.9	56409	56417
j3025_4	37233	12.1	37233	37233
j3025_5	46468	11.6	46558	46472
j3025_6	46514	11.2	46514	46654
j3025_7	47268	14.8	47534	47482
j3025_8	61675	11.1	61689	61675
j3025_9	40714	12.7	40714	40722
j3025_10	26308	11.9	26564	26438
j3026_1	24558	13.9	24734	24644
j3026_2	41552	11.0	41608	41552
j3026_3	31763	14.1	31937	31983
j3026_4	58621	15.6	58665	58647
j3026_5	32750	15.2	33090	32806
j3026_6	49440	12.3	49440	49440
j3026_7	45950	13.6	45950	45966
j3026_8	26496	14.8	26662	26574
j3026_9	35725	10.8	35731	35773
j3026_10	46730	12.4	46730	46786
j3027_1	30164	11.0	30164	30176
j3027_2	47389	14.5	47433	47547
j3027_3	39617	14.3	39617	39617
j3027_4	28237	14.8	28307	28253
j3027_5	43552	12.8	43672	43552
j3027_6	46542	15.0	46542	46862
j3027_7	48316	11.9	48316	48316
j3027_8	35140	15.7	35272	35282
j3027_9	47002	14.0	47204	47156
j3027_10	37674	14.9	38110	37926
j3028_1	35816	16.2	36090	35818
j3028_2	42309	14.1	42449	42383
j3028_3	28855	10.7	28855	28883
j3028_4	45005	12.7	45035	45067
j3028_5	36291	17.3	36291	36497
j3028_6	37824	13.4	37834	37840
j3028_7	44459	12.3	44459	44461
j3028_8	40869	13.1	40869	40885
j3028_9	54348	15.6	54476	54348
j3028_10	44133	14.7	44183	44133
j3029_1	63623	15.3	63801	63749
j3029_2	71674	14.4	71882	71780
j3029_3	72010	11.8	72010	72010
j3029_4	78860	14.6	78904	78860
j3029_5	96056	13.7	96056	96056
j3029_6	107109	12.3	107109	107663
j3029_7	75801	11.2	75999	75801
j3029_8	97042	13.1	97042	97132
j3029_9	83828	14.1	83942	83912
j3029_10	48742	12.7	48762	48884
j3030_1	64349	11.6	64489	64349
j3030_2	74171	16.3	74471	74501
j3030_3	53982	13.5	54086	54124
j3030_4	77940	12.7	78016	77978
j3030_5	89370	14.2	89394	89870
j3030_6	79159	14.2	79357	79159
j3030_7	89007	16.9	89007	89007
j3030_8	63061	11.1	63061	63061
j3030_9	96471	12.5	96555	96661
j3030_10	101746	12.6	101746	101746
j3031_1	63601	11.7	63633	63627
j3031_2	66531	15.6	67097	66505
j3031_3	79138	15.1	79138	79190
j3031_4	47159	13.0	47363	47159
j3031_5	58775	12.6	58775	58775
j3031_6	49549	13.3	49603	49599
j3031_7	76382	15.5	76510	76382
j3031_8	54894	14.4	55108	54896
j3031_9	66820	12.2	67002	66924
j3031_10	86376	13.2	86384	86388
j3032_1	81773	15.5	81799	81773
j3032_2	64667	15.1	64739	64959
j3032_3	90724	14.7	90760	90776
j3032_4	69814	16.7	69820	69814
j3032_5	79767	14.2	79775	80047
j3032_6	54642	11.6	54642	55110
j3032_7	66411	9.7	66427	66411
j3032_8	65078	13.9	65088	65078
j3032_9	64065	15.9	64285	64277
j3032_10	66369	13.5	66395	66473
j3033_1	7474	13.0	7496	7474
j3033_2	6537	11.2	6537	6537
j3033_3	8532	9.4	8532	8532
j3033_4	8168	14.3	8168	8168
j3033_5	9520	9.0	9520	9520
j3033_6	8970	10.5	8970	8970
j3033_7	5594	11.2	5610	5594
j3033_8	7610	9.9	7610	7610
j3033_9	10191	11.6	10191	10191
j3033_10	7383	10.2	7383	7383
j3034_1	10351	12.8	10351	10351
j3034_2	7097	9.2	7097	7097
j3034_3	4945	12.4	4945	4945
j3034_4	9261	12.9	9261	9261
j3034_5	7805	11.5	7805	7805
j3034_6	6821	10.9	6821	6821
j3034_7	8387	10.9	8439	8387
j3034_8	10157	10.0	10157	10157
j3034_9	6624	11.8	6636	6624
j3034_10	5774	10.0	5774	5774

Table A.5. J30 Instances' SSRR Solutions for RLP (5/6)

Instance	Optimal	Time (S)	MASA	QHGA	Instance	Optimal	Time (S)	MASA	QHGA
j3035_1	7303	11.9	7303	7303	j3040_1	24388	12.1	24388	24502
j3035_2	7673	11.3	7673	7673	j3040_2	20893	13.2	20895	20895
j3035_3	6562	11.6	6562	6562	j3040_3	24774	13.0	24866	24812
j3035_4	8555	10.2	8555	8555	j3040_4	23762	13.3	23762	23778
j3035_5	5837	11.8	5837	5837	j3040_5	26089	14.3	26089	26131
j3035_6	6870	11.5	6870	6870	j3040_6	17791	13.6	17857	17791
j3035_7	6767	11.7	6767	6767	j3040_7	32911	11.9	32911	32923
j3035_8	6840	12.6	6840	6840	j3040_8	24298	14.8	24298	24342
j3035_9	7027	12.0	7027	7063	j3040_9	20173	15.1	20189	20173
j3035_10	7660	11.5	7660	7660	j3040_10	23011	12.7	23011	23011
j3036_1	9120	13.1	9120	9216	j3041_1	41878	13.1	41878	41898
j3036_2	6335	9.5	6335	6335	j3041_2	46743	13.0	46813	46743
j3036_3	5549	11.7	5549	5549	j3041_3	33799	14.4	33843	33799
j3036_4	8513	12.0	8513	8513	j3041_4	36187	12.1	36247	36263
j3036_5	8352	12.6	8352	8352	j3041_5	57344	15.5	57344	57344
j3036_6	10011	9.7	10011	10011	j3041_6	34786	15.3	34830	34846
j3036_7	8355	11.4	8355	8355	j3041_7	48238	14.3	48238	48268
j3036_8	5634	11.8	5638	5634	j3041_8	55525	13.8	55525	56115
j3036_9	9014	12.0	9014	9014	j3041_9	38295	16.2	38475	38479
j3036_10	8376	12.0	8376	8376	j3041_10	48872	14.8	48884	48904
j3037_1	28327	11.5	28489	28327	j3042_1	32919	14.0	32937	32919
j3037_2	20041	11.9	20041	20089	j3042_2	39389	12.0	39389	39449
j3037_3	25552	12.4	25992	25552	j3042_3	46472	13.7	46472	46472
j3037_4	22836	14.3	22860	22872	j3042_4	41363	11.0	41363	41363
j3037_5	25850	14.4	25850	25886	j3042_5	45071	13.0	45127	45175
j3037_6	19399	12.1	19399	19399	j3042_6	42706	14.2	42706	42760
j3037_7	29472	12.7	29712	29624	j3042_7	35236	15.3	35240	35366
j3037_8	19904	13.9	19904	19904	j3042_8	36278	15.7	36330	36296
j3037_9	13286	10.8	13314	13286	j3042_9	45660	14.8	45766	45660
j3037_10	17051	13.8	17051	17051	j3042_10	41411	17.3	41591	41411
j3038_1	14968	11.4	15004	14976	j3043_1	36688	13.2	36732	36688
j3038_2	18969	12.0	19069	19013	j3043_2	45233	11.2	45233	45247
j3038_3	24959	12.3	24959	25023	j3043_3	48680	13.9	48680	48688
j3038_4	26128	13.2	26128	26128	j3043_4	28252	15.5	28314	28252
j3038_5	26670	14.1	26688	26712	j3043_5	39083	15.0	39083	39083
j3038_6	21360	14.3	21400	21392	j3043_6	35950	13.8	35950	36028
j3038_7	18646	13.6	18646	18646	j3043_7	42665	13.5	42703	42665
j3038_8	25314	12.8	25314	25314	j3043_8	44551	15.1	44551	44647
j3038_9	22197	13.1	22197	22201	j3043_9	32647	13.8	32647	32647
j3038_10	24939	13.4	25221	24979	j3043_10	39433	14.5	39483	39457
j3039_1	22262	13.1	22412	22286	j3044_1	36856	12.7	36944	36902
j3039_2	23410	12.9	23594	23410	j3044_2	56040	14.0	56040	56096
j3039_3	18626	12.8	18626	18626	j3044_3	52913	13.4	53071	52985
j3039_4	22316	13.0	22316	22316	j3044_4	49379	14.2	49379	49439
j3039_5	17407	13.2	17421	17437	j3044_5	71141	15.4	71141	71249
j3039_6	25544	13.6	25544	25544	j3044_6	35598	14.3	35734	35616
j3039_7	12471	11.4	12479	12627	j3044_7	53111	12.5	53111	53169
j3039_8	22757	15.2	22757	22883	j3044_8	43622	13.1	43674	43622
j3039_9	21852	13.2	21856	21924	j3044_9	31433	15.2	31543	31457
j3039_10	18949	13.3	18965	18949	j3044_10	42977	17.2	43027	43003

Table A.6. J30 Instances' SSRR Solutions for RLP (6/6)

Instance	Optimal	Time (S)	MASA	QHGA	Instance	Optimal	Time (S)	MASA	QHGA
j3045_1	55837	13.2	55935	55837	j3047_1	49087	14.5	49131	49303
j3045_2	73215	15.7	73519	73469	j3047_2	75502	14.8	75668	75566
j3045_3	52268	14.3	52332	52268	j3047_3	75581	16.0	76375	75581
j3045_4	61275	15.0	61603	61275	j3047_4	60056	12.5	60056	60114
j3045_5	67350	14.8	67350	67350	j3047_5	55575	11.3	55575	55579
j3045_6	73343	17.7	73715	73383	j3047_6	71273	13.2	71273	71273
j3045_7	51961	17.5	51987	51961	j3047_7	81563	15.5	81631	81563
j3045_8	41772	15.8	42200	41772	j3047_8	65247	12.7	65289	65305
j3045_9	67231	13.4	67405	67605	j3047_9	66253	16.4	66339	66287
j3045_10	56931	14.5	56971	56931	j3047_10	75993	14.6	75993	75993
j3046_1	44776	14.2	44888	44776	j3048_1	55939	15.0	56025	56001
j3046_2	83631	16.5	83835	83631	j3048_2	50817	13.3	50817	50883
j3046_3	66198	16.0	66764	66280	j3048_3	69270	13.3	69270	69270
j3046_4	62624	15.3	62624	62624	j3048_4	73544	14.6	73544	73730
j3046_5	100839	14.9	100879	100839	j3048_5	65301	14.9	65301	65301
j3046_6	61326	14.2	61352	61384	j3048_6	53460	15.3	53562	53460
j3046_7	66713	12.7	66713	66713	j3048_7	71444	16.0	71698	71494
j3046_8	64746	14.2	64746	64764	j3048_8	67838	13.3	67908	67838
j3046_9	74110	12.5	74228	74110	j3048_9	91693	16.6	91771	91703
j3046_10	69040	12.2	69040	69040	j3048_10	79954	14.2	79954	79954

Table A.7. J60 Instances’ SSRR Solutions for RLP (1/4)

Instance	Time (S)	MASA	QHGA	Instance	Time (S)	MASA	QHGA	Instance	Time (S)	MASA	QHGA	Instance	Time (S)	MASA	QHGA
j601_1	17.3	14325	13959	j604_1	19.0	20190	19972	j607_1	19.0	50099	50153	j6010_1	21.1	95495	92229
j601_2	15.4	21987	22007	j604_2	14.2	18835	18379	j607_2	20.3	37480	36952	j6010_2	17.5	150776	150672
j601_3	15.6	20255	19981	j604_3	14.5	16735	16423	j607_3	16.4	63392	62190	j6010_3	19.6	122825	121539
j601_4	17.3	17263	17085	j604_4	15.1	14005	13959	j607_4	17.1	79100	79136	j6010_4	20.1	81454	80812
j601_5	15.4	14199	14007	j604_5	17.1	18515	18159	j607_5	17.4	48322	47336	j6010_5	20.8	125704	124938
j601_6	13.0	19545	19399	j604_6	15.8	10525	10425	j607_6	16.4	53026	52718	j6010_6	18.2	120705	120255
j601_7	14.7	20342	20130	j604_7	15.8	18477	18395	j607_7	21.8	58347	58299	j6010_7	18.7	149087	148325
j601_8	16.6	19874	19506	j604_8	14.9	12795	12557	j607_8	16.7	58470	58108	j6010_8	17.0	94717	94305
j601_9	17.0	17831	17251	j604_9	16.7	17869	17149	j607_9	12.8	85023	85059	j6010_9	19.5	133012	133476
j601_10	17.3	13873	13751	j604_10	16.6	14606	14112	j607_10	19.8	47722	46696	j6010_10	18.9	122733	122029
j602_1	15.3	15519	15329	j605_1	15.6	58824	58620	j608_1	16.5	56292	55570	j6011_1	19.1	143114	142370
j602_2	17.4	25801	25665	j605_2	18.9	67222	66792	j608_2	16.6	99184	99152	j6011_2	16.8	129672	129458
j602_3	17.4	25165	24751	j605_3	15.8	70336	69914	j608_3	19.0	53041	51743	j6011_3	20.0	128690	128130
j602_4	17.4	15337	15041	j605_4	14.4	51242	51158	j608_4	16.9	76949	76969	j6011_4	18.1	133135	132739
j602_5	13.7	18380	18300	j605_5	19.5	56428	55714	j608_5	20.3	52372	51788	j6011_5	17.9	133342	133132
j602_6	14.7	16659	16223	j605_6	16.4	56269	56077	j608_6	15.5	82326	82076	j6011_6	18.7	121346	121606
j602_7	13.1	18148	18036	j605_7	14.0	58750	58802	j608_7	16.6	73194	73288	j6011_7	19.2	139012	137992
j602_8	15.3	18801	18263	j605_8	16.6	48102	47298	j608_8	16.8	52907	52583	j6011_8	18.9	124845	124991
j602_9	15.5	16656	16526	j605_9	19.8	36396	35532	j608_9	15.6	56538	56302	j6011_9	16.9	133850	133602
j602_10	15.8	26681	26149	j605_10	17.5	66726	66472	j608_10	23.0	48183	47811	j6011_10	16.2	133379	133349
j603_1	14.6	15479	15373	j606_1	16.0	59621	59187	j609_1	16.5	137430	137054	j6012_1	16.8	148933	149329
j603_2	16.1	13884	13358	j606_2	17.2	48482	48188	j609_2	18.1	76870	76776	j6012_2	15.9	127436	126968
j603_3	21.0	13119	12991	j606_3	18.3	40415	40377	j609_3	17.1	93884	93976	j6012_3	19.5	120079	120237
j603_4	17.7	15119	14941	j606_4	17.0	67213	66923	j609_4	17.4	94719	94443	j6012_4	18.6	123484	123286
j603_5	18.3	16574	16308	j606_5	20.0	67198	66316	j609_5	15.2	136989	136639	j6012_5	17.4	150214	150040
j603_6	14.6	24924	24650	j606_6	14.9	66027	66021	j609_6	23.1	119298	118398	j6012_6	15.6	142277	142213
j603_7	13.9	18151	17867	j606_7	15.9	64928	64896	j609_7	19.5	142290	142366	j6012_7	18.9	119332	119022
j603_8	13.2	16496	16372	j606_8	18.1	60234	60326	j609_8	18.0	127556	127216	j6012_8	16.4	121964	121794
j603_9	15.2	17818	17804	j606_9	16.7	62423	62387	j609_9	20.8	116882	116180	j6012_9	17.0	152122	152002
j603_10	15.8	16295	16237	j606_10	18.4	48437	48349	j609_10	17.5	125055	124325	j6012_10	20.1	110029	109405

Table A.8. J60 Instances’ SSRR Solutions for RLP (2/4)

Instance	Time (S)	MASA	QHGA	Instance	Time (S)	MASA	QHGA	Instance	Time (S)	MASA	QHGA	Instance	Time (S)	MASA	QHGA
j6013_1	19.1	207894	207790	j6016_1	18.8	261742	261954	j6019_1	15.1	17411	17277	j6022_1	16.4	57027	56819
j6013_2	18.2	215945	214541	j6016_2	17.9	199490	199426	j6019_2	18.7	14551	14227	j6022_2	20.3	74388	74418
j6013_3	16.7	234444	235160	j6016_3	15.7	247205	247351	j6019_3	17.8	13628	13376	j6022_3	17.9	48096	47466
j6013_4	18.3	292794	292412	j6016_4	16.7	172924	172834	j6019_4	15.8	18227	18205	j6022_4	17.0	88529	88527
j6013_5	16.1	255190	255204	j6016_5	18.0	214041	213677	j6019_5	16.7	16363	16339	j6022_5	18.7	45309	44151
j6013_6	17.3	207103	207025	j6016_6	18.8	259325	259365	j6019_6	16.2	20033	20015	j6022_6	19.6	62190	61872
j6013_7	15.9	232895	233183	j6016_7	20.6	156052	155576	j6019_7	14.5	16383	16455	j6022_7	18.2	71771	71697
j6013_8	19.0	197658	198374	j6016_8	18.8	203418	203310	j6019_8	17.7	11840	11726	j6022_8	16.0	61155	60895
j6013_9	19.1	231751	231685	j6016_9	16.0	262815	262363	j6019_9	15.8	19374	19306	j6022_9	17.5	71263	71291
j6013_10	18.1	254298	254130	j6016_10	19.1	217991	217963	j6019_10	16.8	16035	15867	j6022_10	17.4	51647	51351
j6014_1	17.2	247484	248124	j6017_1	16.6	17476	17132	j6020_1	15.1	20333	20301	j6023_1	19.4	59014	58804
j6014_2	18.3	236618	236026	j6017_2	15.3	12870	12526	j6020_2	16.7	8382	8368	j6023_2	17.9	73206	73220
j6014_3	17.1	233249	234169	j6017_3	17.9	16599	16593	j6020_3	15.9	17977	17943	j6023_3	19.0	47661	47647
j6014_4	18.4	204578	204642	j6017_4	15.4	11919	11827	j6020_4	17.6	16435	16107	j6023_4	20.2	51629	51213
j6014_5	16.7	229742	230082	j6017_5	12.6	15990	15930	j6020_5	16.3	14280	14040	j6023_5	18.1	73044	72912
j6014_6	18.0	187186	186356	j6017_6	15.3	16921	16831	j6020_6	20.1	15338	15272	j6023_6	19.4	39091	38175
j6014_7	18.5	168978	169226	j6017_7	17.4	15209	15003	j6020_7	17.2	20326	20136	j6023_7	15.9	56232	56148
j6014_8	21.4	110580	109648	j6017_8	15.0	26672	26602	j6020_8	15.3	16031	15905	j6023_8	18.1	49602	48086
j6014_9	17.1	176535	175935	j6017_9	15.5	21127	20829	j6020_9	17.3	17596	17408	j6023_9	16.5	51569	51585
j6014_10	19.6	254052	254036	j6017_10	15.7	18655	18425	j6020_10	16.0	13815	13627	j6023_10	16.9	44900	44692
j6015_1	21.5	164226	164986	j6018_1	17.2	14925	14719	j6021_1	18.8	44582	44132	j6024_1	16.3	30761	30761
j6015_2	22.4	165348	164520	j6018_2	16.1	22911	22447	j6021_2	21.3	49672	48704	j6024_2	14.8	63683	63973
j6015_3	19.3	192146	192276	j6018_3	16.5	13860	13574	j6021_3	17.0	58986	58966	j6024_3	17.0	52358	52378
j6015_4	20.7	232204	230112	j6018_4	16.8	24753	24435	j6021_4	16.3	51397	51299	j6024_4	19.9	65029	64311
j6015_5	19.3	201854	201998	j6018_5	16.9	14711	14307	j6021_5	18.5	59267	59245	j6024_5	19.1	46727	46063
j6015_6	20.1	189376	190180	j6018_6	14.7	18922	18788	j6021_6	16.2	50169	50179	j6024_6	18.5	40299	39669
j6015_7	17.9	247739	247757	j6018_7	17.5	19106	18872	j6021_7	19.5	53894	53612	j6024_7	17.8	70918	70904
j6015_8	20.9	182326	182008	j6018_8	17.1	14365	13739	j6021_8	20.6	55094	54618	j6024_8	19.8	57088	56988
j6015_9	20.4	303652	303830	j6018_9	15.9	14056	14038	j6021_9	18.2	71577	71183	j6024_9	19.3	51131	50869
j6015_10	17.0	161981	161395	j6018_10	20.8	13058	12942	j6021_10	13.7	53132	53130	j6024_10	17.3	57549	57007

Table A.9. J60 Instances' SSRR Solutions for RLP (3/4)

Instance	Time (S)	MASA	QHGA	Instance	Time (S)	MASA	QHGA	Instance	Time (S)	MASA	QHGA	Instance	Time (S)	MASA	QHGA
j6025_1	18.1	97818	97308	j6028_1	22.7	101118	100344	j6031_1	18.1	184369	183445	j6034_1	15.5	20286	20144
j6025_2	18.7	137351	137779	j6028_2	17.1	104112	103906	j6031_2	20.0	229045	228955	j6034_2	15.5	18681	18681
j6025_3	22.5	97749	98045	j6028_3	18.5	99955	99611	j6031_3	18.4	187370	187464	j6034_3	14.7	18352	18086
j6025_4	20.0	107746	107574	j6028_4	21.1	96528	94792	j6031_4	18.4	157594	157834	j6034_4	17.7	15494	14260
j6025_5	16.4	100117	100247	j6028_5	19.0	115838	115642	j6031_5	19.3	177796	177310	j6034_5	17.3	13742	13616
j6025_6	20.0	132234	130718	j6028_6	21.8	110339	110195	j6031_6	20.4	316194	316914	j6034_6	17.4	25835	25753
j6025_7	17.1	95264	96130	j6028_7	19.3	102958	102934	j6031_7	19.9	163683	162661	j6034_7	18.0	19302	18944
j6025_8	16.3	142750	142876	j6028_8	16.9	114279	114125	j6031_8	20.0	201474	201856	j6034_8	14.2	13852	13708
j6025_9	18.0	86611	86209	j6028_9	19.7	130615	130113	j6031_9	22.7	186652	186066	j6034_9	16.7	17112	16930
j6025_10	20.5	112834	110610	j6028_10	20.2	172417	173515	j6031_10	16.6	197770	197626	j6034_10	19.5	20114	19852
j6026_1	20.2	97221	96821	j6029_1	16.9	198125	198813	j6032_1	19.0	192235	191865	j6035_1	17.8	18814	18680
j6026_2	16.9	94473	94401	j6029_2	22.2	141197	140643	j6032_2	26.8	133018	132178	j6035_2	17.1	17582	17446
j6026_3	19.3	117038	117054	j6029_3	19.6	238153	238185	j6032_3	22.5	222779	222329	j6035_3	18.3	15915	15905
j6026_4	17.3	111917	111131	j6029_4	20.5	212460	212070	j6032_4	16.3	200764	200670	j6035_4	16.1	19862	19812
j6026_5	16.6	114041	113791	j6029_5	20.1	171625	171651	j6032_5	20.6	199855	200171	j6035_5	16.8	19583	19569
j6026_6	19.1	116699	116499	j6029_6	21.8	274311	273793	j6032_6	23.0	143038	142674	j6035_6	17.4	16901	16851
j6026_7	18.6	80345	79685	j6029_7	19.3	178423	177565	j6032_7	20.6	176898	178282	j6035_7	16.2	17595	17555
j6026_8	21.9	92887	91995	j6029_8	19.5	197395	197279	j6032_8	19.4	144866	144070	j6035_8	17.3	20468	20198
j6026_9	17.8	175817	175531	j6029_9	19.4	173783	172825	j6032_9	20.6	215864	216634	j6035_9	16.3	20301	20129
j6026_10	21.1	92887	92221	j6029_10	19.3	235240	235396	j6032_10	20.4	185771	184745	j6035_10	16.0	16711	16689
j6027_1	23.5	86807	86427	j6030_1	18.8	170688	170714	j6033_1	19.0	17634	17510	j6036_1	14.5	17552	17538
j6027_2	18.5	78586	78792	j6030_2	18.3	193162	193110	j6033_2	20.8	14242	13940	j6036_2	16.0	18399	18361
j6027_3	19.4	104878	103630	j6030_3	21.6	216598	216400	j6033_3	16.1	13715	13489	j6036_3	17.3	14371	14347
j6027_4	16.3	112328	112406	j6030_4	19.9	175002	175208	j6033_4	17.2	16309	16175	j6036_4	18.0	13102	12940
j6027_5	20.2	104187	102679	j6030_5	20.3	255730	255200	j6033_5	20.0	17383	16649	j6036_5	13.8	18430	18242
j6027_6	17.8	140769	140271	j6030_6	18.2	174283	174655	j6033_6	15.6	16392	16344	j6036_6	16.8	19185	19127
j6027_7	21.1	109428	108238	j6030_7	21.5	235409	234991	j6033_7	15.7	20846	20820	j6036_7	16.0	17324	17344
j6027_8	21.8	96271	95391	j6030_8	17.8	212291	211483	j6033_8	16.3	17933	17865	j6036_8	16.0	14818	14712
j6027_9	19.5	107024	106920	j6030_9	24.5	176980	176736	j6033_9	20.3	15871	15113	j6036_9	18.3	13697	13613
j6027_10	16.3	139395	139205	j6030_10	21.3	229526	229748	j6033_10	15.8	14924	14890	j6036_10	17.0	10370	10336

Table A.10. J60 Instances' SSRR Solutions for RLP (4/4)

Instance	Time (S)	MASA	QHGA	Instance	Time (S)	MASA	QHGA	Instance	Time (S)	MASA	QHGA	Instance	Time (S)	MASA	QHGA
j6037_1	17.2	47188	47190	j6040_1	21.3	69655	69365	j6043_1	25.0	85726	85392	j6046_1	20.4	184216	183958
j6037_2	17.4	47772	47120	j6040_2	20.1	53249	52851	j6043_2	21.8	116853	114921	j6046_2	20.8	181979	181803
j6037_3	23.7	74267	73497	j6040_3	17.9	53748	53414	j6043_3	19.4	115465	115397	j6046_3	20.8	159274	157672
j6037_4	18.9	55103	55003	j6040_4	20.9	43536	42816	j6043_4	20.0	149366	149348	j6046_4	19.6	201584	201764
j6037_5	19.1	43878	42948	j6040_5	20.6	67554	67430	j6043_5	17.3	102759	102731	j6046_5	21.8	191656	191250
j6037_6	16.9	75981	76045	j6040_6	17.5	60253	60309	j6043_6	20.9	101783	101419	j6046_6	21.8	143881	143849
j6037_7	19.2	59794	59398	j6040_7	17.3	49836	49836	j6043_7	21.5	70466	69832	j6046_7	19.9	173969	173593
j6037_8	19.3	43286	42100	j6040_8	19.9	58915	58793	j6043_8	19.1	149784	149990	j6046_8	19.6	212630	212800
j6037_9	19.3	51212	50180	j6040_9	21.5	47292	46168	j6043_9	18.2	99435	99387	j6046_9	17.5	256064	256262
j6037_10	20.6	48301	48019	j6040_10	18.2	38859	38589	j6043_10	19.5	77654	77308	j6046_10	21.0	196327	196261
j6038_1	18.5	55463	55359	j6041_1	22.8	90698	90732	j6044_1	20.8	80244	79874	j6047_1	20.2	216079	215523
j6038_2	18.6	75636	75406	j6041_2	20.4	105763	105717	j6044_2	18.7	102317	102033	j6047_2	18.9	249799	250411
j6038_3	19.5	72681	72153	j6041_3	16.3	133858	133988	j6044_3	22.0	93112	92424	j6047_3	19.0	204673	204677
j6038_4	15.7	57347	57349	j6041_4	24.1	86386	85408	j6044_4	20.5	136237	135159	j6047_4	20.1	162471	163405
j6038_5	23.3	40663	40003	j6041_5	19.1	142275	141999	j6044_5	19.3	102139	101989	j6047_5	22.1	147819	147535
j6038_6	20.8	45226	44598	j6041_6	20.9	115995	115901	j6044_6	20.2	87844	87332	j6047_6	20.0	152519	152729
j6038_7	18.4	63490	63166	j6041_7	21.5	104517	104569	j6044_7	19.7	110274	109862	j6047_7	18.9	196317	197313
j6038_8	17.6	69816	69758	j6041_8	23.3	101028	100124	j6044_8	21.0	102188	102644	j6047_8	19.0	185570	185554
j6038_9	16.8	51599	51547	j6041_9	20.9	146191	146157	j6044_9	17.9	110715	110449	j6047_9	20.1	181168	181386
j6038_10	17.1	85656	85654	j6041_10	19.4	161841	161841	j6044_10	17.7	103744	103654	j6047_10	18.8	217103	217389
j6039_1	19.7	60098	60300	j6042_1	20.9	103715	102829	j6045_1	18.7	150821	150413	j6048_1	19.4	186002	185986
j6039_2	20.3	59088	58754	j6042_2	18.2	111958	111480	j6045_2	21.3	228255	228053	j6048_2	21.4	129259	128503
j6039_3	20.4	48017	47113	j6042_3	19.4	120339	120427	j6045_3	22.6	193614	193876	j6048_3	22.4	220708	220446
j6039_4	21.4	48751	48317	j6042_4	23.2	110493	110267	j6045_4	16.7	173870	173910	j6048_4	17.4	177944	177902
j6039_5	18.3	46545	46185	j6042_5	19.0	110865	110891	j6045_5	16.9	192157	192049	j6048_5	24.7	152912	151956
j6039_6	19.8	43635	43613	j6042_6	20.4	84158	84002	j6045_6	22.1	206795	206635	j6048_6	18.3	195215	194793
j6039_7	17.7	49163	48841	j6042_7	15.5	101538	101558	j6045_7	19.3	170972	170822	j6048_7	20.5	216827	216329
j6039_8	18.6	57606	57676	j6042_8	20.4	114225	114115	j6045_8	21.1	193928	193926	j6048_8	22.4	157680	156586
j6039_9	18.3	49621	49223	j6042_9	18.3	104106	104226	j6045_9	20.6	182486	182508	j6048_9	20.8	139114	138664
j6039_10	18.3	43683	43437	j6042_10	21.1	91374	90394	j6045_10	17.5	223249	223145	j6048_10	19.0	174406	174042

Table A.11. J120 Instances’ SSRR Solutions for RLP (1/4)

Instance	Time (S)	MASA	QHGA	Instance	Time (S)	MASA	QHGA	Instance	Time (S)	MASA	QHGA	Instance	Time (S)	MASA	QHGA
j1201_1	26.6	38844	38200	j1204_1	20.9	53594	53142	j1207_1	24.5	220072	219408	j12010_1	31.1	148673	146123
j1201_2	23.5	46570	45260	j1204_2	28.0	42876	40426	j1207_2	27.5	133142	128892	j12010_2	27.0	165498	160596
j1201_3	23.2	53173	52475	j1204_3	25.2	46267	45271	j1207_3	26.3	142146	136860	j12010_3	29.4	186723	183401
j1201_4	22.3	43847	43547	j1204_4	21.6	50535	49857	j1207_4	25.4	153546	150920	j12010_4	28.1	167539	165799
j1201_5	24.8	42933	41843	j1204_5	22.0	62618	60112	j1207_5	27.1	199682	199198	j12010_5	28.5	137572	135086
j1201_6	19.6	42472	41280	j1204_6	24.3	45999	44595	j1207_6	28.3	158028	155866	j12010_6	27.1	155321	151329
j1201_7	26.0	40356	38376	j1204_7	23.4	57271	56925	j1207_7	28.2	175595	172691	j12010_7	25.5	207820	206402
j1201_8	23.7	43556	42110	j1204_8	24.5	45354	43732	j1207_8	21.5	170234	169438	j12010_8	32.7	134468	131286
j1201_9	24.6	50941	50417	j1204_9	22.8	44270	43496	j1207_9	24.9	150483	148945	j12010_9	24.3	173664	172136
j1201_10	24.0	47315	46019	j1204_10	22.0	44035	42903	j1207_10	25.5	175663	175021	j12010_10	22.4	258297	257225
j1202_1	20.9	60373	58901	j1205_1	25.1	38943	37831	j1208_1	27.8	209392	207574	j12011_1	27.9	333929	330973
j1202_2	21.4	45789	44975	j1205_2	23.4	57442	57036	j1208_2	27.3	201567	200529	j12011_2	25.1	304478	303722
j1202_3	23.2	69641	68757	j1205_3	21.8	52371	51981	j1208_3	25.9	144508	142452	j12011_3	28.8	356797	355843
j1202_4	24.5	37795	37309	j1205_4	26.1	43449	42051	j1208_4	27.1	164893	162265	j12011_4	29.6	329992	329698
j1202_5	25.7	52574	51240	j1205_5	22.6	46176	45702	j1208_5	27.4	171879	170051	j12011_5	30.4	394485	391929
j1202_6	22.2	64570	63672	j1205_6	24.2	55727	54591	j1208_6	25.3	151506	151060	j12011_6	29.0	426668	427962
j1202_7	22.9	43115	42309	j1205_7	24.1	43164	42248	j1208_7	26.2	172105	171223	j12011_7	25.8	308142	305652
j1202_8	21.9	41425	39761	j1205_8	22.3	58203	56393	j1208_8	26.7	157444	157054	j12011_8	28.5	320769	310969
j1202_9	24.9	47406	46302	j1205_9	27.5	41288	39148	j1208_9	25.5	193913	192091	j12011_9	25.8	429292	429794
j1202_10	23.3	78395	77085	j1205_10	25.3	44168	43146	j1208_10	26.2	216119	215227	j12011_10	27.7	375953	373465
j1203_1	22.5	63166	61928	j1206_1	23.6	176824	174766	j1209_1	26.8	179707	178777	j12012_1	29.0	297815	294457
j1203_2	23.6	34750	34270	j1206_2	22.6	184174	182064	j1209_2	28.0	160920	158896	j12012_2	24.8	443376	442378
j1203_3	26.2	44058	42438	j1206_3	25.8	162922	160766	j1209_3	26.1	150417	149369	j12012_3	26.9	400456	400572
j1203_4	20.8	49279	48591	j1206_4	28.0	108329	107103	j1209_4	25.4	186620	185894	j12012_4	28.3	290095	286305
j1203_5	22.3	49509	48435	j1206_5	23.4	182625	181109	j1209_5	31.4	131536	129064	j12012_5	30.3	388378	385722
j1203_6	26.8	39705	38901	j1206_6	24.1	221624	220624	j1209_6	29.0	175357	173759	j12012_6	26.9	405084	404402
j1203_7	24.5	41285	38751	j1206_7	31.0	152743	150381	j1209_7	25.1	203841	203313	j12012_7	27.5	398263	398713
j1203_8	22.2	52609	51429	j1206_8	29.2	139366	135318	j1209_8	25.3	167339	165281	j12012_8	25.8	506922	505408
j1203_9	23.7	38242	37608	j1206_9	26.7	220763	220279	j1209_9	26.1	145951	142837	j12012_9	26.5	333985	332427
j1203_10	27.0	43052	41286	j1206_10	28.6	208329	202937	j1209_10	25.8	183859	181801	j12012_10	27.8	444787	443783

Table A.12. J120 Instances' SSRR Solutions for RLP (2/4)

Instance	Time (S)	MASA	QHGA	Instance	Time (S)	MASA	QHGA	Instance	Time (S)	MASA	QHGA	Instance	Time (S)	MASA	QHGA
j12013_1	34.8	315534	307674	j12016_1	25.2	720564	719660	j12019_1	27.6	653851	653485	j12022_1	23.3	54157	52927
j12013_2	24.9	425072	424512	j12016_2	28.8	786441	787807	j12019_2	27.8	834803	835149	j12022_2	27.6	52636	51136
j12013_3	31.8	296552	295104	j12016_3	30.8	745066	746472	j12019_3	25.2	816254	817524	j12022_3	23.4	66186	64958
j12013_4	28.2	397142	396990	j12016_4	28.5	688400	686804	j12019_4	29.2	584445	581567	j12022_4	23.4	52691	51371
j12013_5	25.7	377677	376057	j12016_5	29.6	602476	604190	j12019_5	28.7	691216	690926	j12022_5	25.8	51820	50126
j12013_6	27.1	339168	337282	j12016_6	27.2	829018	827682	j12019_6	26.9	728963	729551	j12022_6	24.7	47258	44542
j12013_7	31.9	336931	332117	j12016_7	29.2	620523	621011	j12019_7	29.4	636583	635423	j12022_7	29.5	32670	30384
j12013_8	28.2	359053	356163	j12016_8	25.6	668928	669806	j12019_8	29.6	614522	614074	j12022_8	25.3	58180	57678
j12013_9	25.3	371400	371340	j12016_9	28.3	645558	644410	j12019_9	25.9	700674	700366	j12022_9	25.0	57978	56390
j12013_10	26.3	437318	437252	j12016_10	30.9	613999	613309	j12019_10	28.6	652898	653508	j12022_10	21.0	49433	48425
j12014_1	27.4	470635	469263	j12017_1	29.2	738332	741012	j12020_1	29.0	649778	650166	j12023_1	27.3	53558	51746
j12014_2	28.2	483269	483511	j12017_2	25.7	776521	779109	j12020_2	30.8	562672	562278	j12023_2	29.8	37446	35300
j12014_3	27.4	310509	308835	j12017_3	24.6	685058	676540	j12020_3	27.0	894741	892891	j12023_3	26.0	38827	35577
j12014_4	27.3	344081	343227	j12017_4	29.0	626943	627675	j12020_4	28.1	573489	573097	j12023_4	27.2	34702	33578
j12014_5	27.8	295909	293365	j12017_5	28.5	561635	561283	j12020_5	23.9	657001	657485	j12023_5	25.4	39132	37436
j12014_6	28.8	348475	346535	j12017_6	24.3	798086	799748	j12020_6	26.0	535281	534603	j12023_6	26.6	46784	46480
j12014_7	28.7	411799	410867	j12017_7	31.7	676243	674551	j12020_7	26.7	542262	542476	j12023_7	26.5	35833	34665
j12014_8	31.0	368665	368293	j12017_8	25.2	713290	714830	j12020_8	32.2	560828	557914	j12023_8	25.8	35568	33814
j12014_9	29.7	304834	303422	j12017_9	27.2	628863	627591	j12020_9	27.7	771297	771691	j12023_9	27.4	41097	40301
j12014_10	26.4	359360	356288	j12017_10	29.3	751522	749656	j12020_10	28.1	783996	776070	j12023_10	27.0	47895	45417
j12015_1	26.5	392603	391233	j12018_1	32.2	663125	664517	j12021_1	26.4	47808	46824	j12024_1	24.6	44974	43342
j12015_2	26.1	482454	481962	j12018_2	33.7	543749	540197	j12021_2	24.2	46297	43499	j12024_2	24.7	37875	37031
j12015_3	27.6	347980	344060	j12018_3	25.6	804540	804690	j12021_3	28.7	54915	51889	j12024_3	24.8	54333	53167
j12015_4	26.7	426189	427161	j12018_4	26.4	705404	705998	j12021_4	28.4	50540	47794	j12024_4	26.6	44949	43515
j12015_5	27.5	361397	359807	j12018_5	28.7	748971	751151	j12021_5	25.7	38916	37486	j12024_5	23.9	41757	40445
j12015_6	30.6	350444	347920	j12018_6	31.4	464123	464423	j12021_6	26.1	43265	41601	j12024_6	25.2	50149	47791
j12015_7	24.6	364428	362896	j12018_7	28.5	645476	644524	j12021_7	22.8	50459	49895	j12024_7	27.8	46840	45444
j12015_8	35.1	290302	288376	j12018_8	27.8	605032	604562	j12021_8	30.3	35002	34290	j12024_8	26.4	43019	41639
j12015_9	32.1	339415	332997	j12018_9	26.2	742256	742610	j12021_9	24.1	44324	42824	j12024_9	22.2	45766	45116
j12015_10	28.8	424477	423663	j12018_10	28.1	764441	765097	j12021_10	23.4	41915	41411	j12024_10	25.0	56342	52952

Table A.13. J120 Instances’ SSRR Solutions for RLP (3/4)

Instance	Time (S)	MASA	QHGA	Instance	Time (S)	MASA	QHGA	Instance	Time (S)	MASA	QHGA	Instance	Time (S)	MASA	QHGA
j12025_1	23.6	49139	48513	j12028_1	28.3	187052	185556	j12031_1	28.7	343638	341978	j12034_1	24.5	404527	403095
j12025_2	27.6	38110	37656	j12028_2	30.5	126398	123714	j12031_2	27.3	403667	406449	j12034_2	29.2	354198	354156
j12025_3	26.8	46023	44057	j12028_3	29.1	133917	132401	j12031_3	25.8	352523	352961	j12034_3	30.2	413974	413936
j12025_4	29.5	38621	37647	j12028_4	29.3	147934	146900	j12031_4	32.5	271968	269800	j12034_4	29.5	339181	339499
j12025_5	26.1	42650	41470	j12028_5	28.8	117833	116215	j12031_5	30.2	343107	340725	j12034_5	29.9	326168	324850
j12025_6	24.9	48729	47297	j12028_6	29.2	142277	139443	j12031_6	30.8	347007	344299	j12034_6	30.1	401986	402666
j12025_7	25.4	44672	43412	j12028_7	29.1	191208	190480	j12031_7	32.4	320831	317653	j12034_7	32.0	282420	282688
j12025_8	22.6	62381	61489	j12028_8	26.2	153416	153426	j12031_8	29.2	351881	350635	j12034_8	26.6	431999	431943
j12025_9	25.0	41244	39420	j12028_9	27.3	141413	140207	j12031_9	29.6	346859	346623	j12034_9	26.2	330722	328050
j12025_10	24.4	40174	39836	j12028_10	30.9	133335	131669	j12031_10	28.4	373350	373134	j12034_10	30.7	384386	382540
j12026_1	29.0	179004	177698	j12029_1	30.1	149659	148191	j12032_1	30.8	321354	316698	j12035_1	27.1	341684	340686
j12026_2	28.8	137138	133136	j12029_2	27.6	163035	161385	j12032_2	30.2	298246	294866	j12035_2	32.4	314704	313406
j12026_3	28.7	165983	161111	j12029_3	25.5	235858	235216	j12032_3	30.4	346613	345555	j12035_3	25.9	438818	439198
j12026_4	30.5	149623	148563	j12029_4	24.5	185851	185469	j12032_4	31.3	260883	258567	j12035_4	30.7	309372	306612
j12026_5	25.1	178259	177039	j12029_5	29.7	171982	170432	j12032_5	30.0	324503	322795	j12035_5	28.6	330314	330756
j12026_6	33.2	121589	119739	j12029_6	26.8	166289	164519	j12032_6	28.8	294485	292101	j12035_6	27.3	339149	338523
j12026_7	25.2	159353	157335	j12029_7	28.7	156625	154953	j12032_7	30.8	348949	345991	j12035_7	30.4	326299	325377
j12026_8	29.8	134808	131498	j12029_8	25.1	158723	157237	j12032_8	26.3	370742	369724	j12035_8	31.1	345182	344456
j12026_9	35.1	145116	141538	j12029_9	28.3	131383	129297	j12032_9	27.2	361957	360511	j12035_9	28.6	364564	363054
j12026_10	32.9	130311	126311	j12029_10	27.9	129051	127337	j12032_10	28.6	346031	344309	j12035_10	27.3	378275	376695
j12027_1	24.9	206245	205165	j12030_1	29.7	159133	158963	j12033_1	29.2	257434	256206	j12036_1	31.0	580717	578661
j12027_2	26.5	152000	150024	j12030_2	31.0	128339	124569	j12033_2	28.5	330896	330480	j12036_2	28.7	640942	640094
j12027_3	28.8	163205	159363	j12030_3	30.5	153889	148609	j12033_3	27.2	348285	345707	j12036_3	28.9	622008	620918
j12027_4	25.7	153456	152438	j12030_4	26.2	173922	172854	j12033_4	29.9	324612	321734	j12036_4	29.3	588898	589598
j12027_5	26.2	125700	122736	j12030_5	25.7	190924	189270	j12033_5	32.2	329218	326472	j12036_5	30.2	588761	586807
j12027_6	28.3	203754	202798	j12030_6	25.1	200583	199537	j12033_6	32.8	236462	233946	j12036_6	31.7	545266	546816
j12027_7	28.9	137823	137299	j12030_7	28.1	190638	188484	j12033_7	29.4	389901	388879	j12036_7	32.0	593175	592057
j12027_8	31.3	193610	192056	j12030_8	25.5	217829	217447	j12033_8	28.7	378199	376907	j12036_8	26.1	424993	425629
j12027_9	28.5	151002	149648	j12030_9	28.4	175985	174949	j12033_9	30.0	325525	323391	j12036_9	31.1	465372	466136
j12027_10	30.3	156117	154059	j12030_10	26.2	165829	165087	j12033_10	28.0	367037	366163	j12036_10	29.6	637394	636864

Table A.14. J120 Instances’ SSRR Solutions for RLP (4/4)

Instance	Time (S)	MASA	QHGA	Instance	Time (S)	MASA	QHGA	Instance	Time (S)	MASA	QHGA	Instance	Time (S)	MASA	QHGA
j12037_1	29.8	502867	503707	j12040_1	26.1	561752	563352	j12043_1	26.4	44919	42981	j12046_1	32.9	154724	152100
j12037_2	27.8	600998	600232	j12040_2	29.3	716441	714451	j12043_2	29.0	37501	36219	j12046_2	33.0	160047	157953
j12037_3	31.4	496009	496237	j12040_3	28.7	597948	599004	j12043_3	25.0	41871	40935	j12046_3	29.4	143558	142040
j12037_4	30.9	671319	670459	j12040_4	34.2	532922	531320	j12043_4	26.6	50480	48952	j12046_4	27.2	176744	175332
j12037_5	34.9	598745	597353	j12040_5	30.2	508230	507382	j12043_5	25.9	46957	45953	j12046_5	28.3	135249	132541
j12037_6	33.1	650972	649014	j12040_6	29.1	665267	665081	j12043_6	23.5	55407	53395	j12046_6	27.8	150396	148908
j12037_7	27.4	686723	685293	j12040_7	29.0	584135	584803	j12043_7	29.7	47532	47086	j12046_7	30.1	128276	124102
j12037_8	34.1	518089	517593	j12040_8	29.8	538239	537657	j12043_8	28.5	40686	39022	j12046_8	28.3	169460	168144
j12037_9	27.6	695792	695998	j12040_9	34.9	521776	518784	j12043_9	27.1	48614	47452	j12046_9	28.2	153916	153358
j12037_10	26.5	621017	622075	j12040_10	30.6	655472	654102	j12043_10	28.0	36818	34342	j12046_10	30.5	159131	158769
j12038_1	31.7	565210	564742	j12041_1	27.5	43723	42861	j12044_1	26.4	35359	33857	j12047_1	30.5	141474	139608
j12038_2	28.8	603891	605127	j12041_2	26.5	52364	50886	j12044_2	27.3	41987	40129	j12047_2	30.0	150798	148934
j12038_3	32.7	599311	599671	j12041_3	29.8	46865	44711	j12044_3	27.8	43719	41661	j12047_3	29.3	168316	168452
j12038_4	34.3	549414	552200	j12041_4	23.5	44740	43522	j12044_4	25.7	53869	53309	j12047_4	30.1	138263	136659
j12038_5	30.4	620283	619977	j12041_5	28.9	39025	37999	j12044_5	26.3	53175	51791	j12047_5	30.1	142053	140217
j12038_6	31.1	684132	684066	j12041_6	24.6	41429	40969	j12044_6	27.6	42733	41223	j12047_6	29.3	152261	150807
j12038_7	29.2	684294	684276	j12041_7	24.8	42466	41344	j12044_7	26.7	53738	51018	j12047_7	26.7	194552	191592
j12038_8	30.8	586604	585880	j12041_8	28.7	40077	39313	j12044_8	26.5	34319	33459	j12047_8	25.7	211442	209144
j12038_9	39.7	485380	481858	j12041_9	25.8	42341	41897	j12044_9	24.6	30876	30214	j12047_9	29.8	150699	148341
j12038_10	30.9	710463	711731	j12041_10	29.8	39993	37643	j12044_10	27.1	59382	58700	j12047_10	29.5	162151	160427
j12039_1	30.0	531340	530974	j12042_1	24.6	55566	53854	j12045_1	26.9	34735	32937	j12048_1	28.0	132301	130295
j12039_2	31.4	508672	509192	j12042_2	30.3	32031	30693	j12045_2	24.9	46161	44539	j12048_2	26.3	172102	170132
j12039_3	30.6	588417	587901	j12042_3	26.2	39408	39130	j12045_3	25.4	41432	39960	j12048_3	30.1	159432	157252
j12039_4	25.7	645181	646921	j12042_4	26.8	41162	38132	j12045_4	26.1	37299	36063	j12048_4	30.6	169469	169317
j12039_5	30.5	430887	430769	j12042_5	28.2	52145	50279	j12045_5	29.5	43982	42142	j12048_5	28.2	156048	155266
j12039_6	30.1	549011	550779	j12042_6	24.1	50487	49499	j12045_6	31.6	44266	41190	j12048_6	27.9	160074	158828
j12039_7	30.6	654368	653854	j12042_7	28.5	35719	34721	j12045_7	26.6	41309	39829	j12048_7	29.1	153278	148618
j12039_8	29.5	500948	497458	j12042_8	26.7	44176	43378	j12045_8	26.9	40982	39838	j12048_8	29.0	134463	133663
j12039_9	27.0	727970	728094	j12042_9	25.7	45019	43785	j12045_9	29.2	34954	33112	j12048_9	29.3	136783	135719
j12039_10	30.4	499787	501169	j12042_10	26.8	52612	52470	j12045_10	25.8	42521	40999	j12048_10	28.6	130350	124912

APPENDIX B

RID-MRD SOLUTIONS OBTAINED BY MASA FOR PSPLIB INSTANCES

All of the tests are carried out on a computer with a 3.00 GHz Core 2 Duo Processor E8400 Intel CPU. The stopping criteria for is defined as 500,000 schedule. Weights of all the resources are taken 1. Weights of RID and MRD are both taken as 1.

Table B.1. J30 Instances' RID-MRD Solutions for RLP (1/5)

Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA
j301_1	235	13.2	92	j304_1	427	13.1	242
j301_2	371	13.3	238	j304_2	969	13.3	230
j301_3	475	12.9	288	j304_3	466	13.1	241
j301_4	805	13.5	315	j304_4	546	13.5	452
j301_5	263	12.2	91	j304_5	702	13.4	290
j301_6	506	12.6	281	j304_6	628	13.0	223
j301_7	848	13.2	469	j304_7	567	13.6	269
j301_8	440	13.2	245	j304_8	639	13.6	319
j301_9	619	13.2	370	j304_9	408	13.0	298
j301_10	438	12.9	161	j304_10	722	13.1	296
j302_1	422	12.7	229	j305_1	529	13.4	282
j302_2	278	13.4	181	j305_2	742	13.8	407
j302_3	470	13.2	279	j305_3	758	13.8	296
j302_4	433	13.1	192	j305_4	474	13.7	200
j302_5	406	13.3	234	j305_5	816	13.8	300
j302_6	573	13.0	278	j305_6	424	13.5	241
j302_7	426	13.0	110	j305_7	554	13.5	242
j302_8	647	13.1	99	j305_8	1008	13.7	417
j302_9	714	13.5	220	j305_9	443	13.1	292
j302_10	239	12.9	131	j305_10	707	13.8	211
j303_1	776	14.1	559	j306_1	446	13.9	212
j303_2	256	12.7	133	j306_2	1021	13.6	327
j303_3	779	13.5	256	j306_3	709	13.6	439
j303_4	1072	14.8	824	j306_4	687	13.4	436
j303_5	641	13.2	272	j306_5	1576	14.1	857
j303_6	507	13.2	252	j306_6	493	12.9	351
j303_7	415	13.0	120	j306_7	766	13.2	329
j303_8	622	13.2	340	j306_8	491	13.2	232
j303_9	609	13.4	243	j306_9	401	13.4	239
j303_10	417	13.4	200	j306_10	862	14.0	318

Table B.2. J30 Instances' RID-MRD Solutions for RLP (2/5)

Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA
j307_1	616	13.7	266	j3010_1	625	13.5	268	j3013_1	483	13.2	231	j3016_1	918	13.7	222
j307_2	489	13.2	266	j3010_2	1185	13.9	234	j3013_2	349	13.3	192	j3016_2	853	13.9	224
j307_3	489	13.1	217	j3010_3	633	14.6	244	j3013_3	596	13.9	215	j3016_3	391	13.3	214
j307_4	711	13.3	358	j3010_4	718	14.1	351	j3013_4	868	13.6	246	j3016_4	627	13.7	259
j307_5	815	13.5	391	j3010_5	711	13.6	173	j3013_5	751	13.6	226	j3016_5	643	14.1	233
j307_6	243	12.8	152	j3010_6	480	13.3	219	j3013_6	685	13.7	253	j3016_6	431	13.7	206
j307_7	1206	13.8	337	j3010_7	635	13.9	213	j3013_7	707	13.8	226	j3016_7	552	13.3	248
j307_8	664	13.5	269	j3010_8	638	14.0	335	j3013_8	698	14.0	280	j3016_8	749	13.4	224
j307_9	597	13.7	264	j3010_9	514	13.6	209	j3013_9	658	13.8	237	j3016_9	1022	13.5	280
j307_10	642	13.2	308	j3010_10	318	13.3	154	j3013_10	744	13.5	295	j3016_10	569	13.9	228
j308_1	431	13.2	315	j3011_1	910	13.9	365	j3014_1	514	13.3	235	j3017_1	582	13.2	292
j308_2	717	13.7	424	j3011_2	590	14.4	284	j3014_2	623	13.8	275	j3017_2	670	13.6	345
j308_3	455	13.8	202	j3011_3	1493	14.8	503	j3014_3	967	14.4	240	j3017_3	775	13.5	418
j308_4	757	13.3	308	j3011_4	1203	14.4	294	j3014_4	853	13.4	236	j3017_4	323	12.7	173
j308_5	613	13.8	355	j3011_5	702	13.8	251	j3014_5	682	13.5	179	j3017_5	341	13.0	239
j308_6	587	13.7	197	j3011_6	674	13.7	244	j3014_6	357	13.2	184	j3017_6	749	13.7	269
j308_7	396	13.2	163	j3011_7	515	13.6	199	j3014_7	746	13.8	230	j3017_7	486	13.3	299
j308_8	1230	13.5	214	j3011_8	1157	14.3	232	j3014_8	796	13.9	191	j3017_8	484	13.3	300
j308_9	330	13.2	158	j3011_9	1276	14.1	493	j3014_9	629	13.4	207	j3017_9	646	13.0	549
j308_10	1246	14.2	347	j3011_10	1199	13.3	241	j3014_10	1726	14.2	533	j3017_10	489	13.9	207
j309_1	726	13.9	282	j3012_1	955	13.8	326	j3015_1	1036	13.5	220	j3018_1	648	13.3	462
j309_2	763	13.6	228	j3012_2	523	13.7	246	j3015_2	802	13.9	209	j3018_2	313	13.4	136
j309_3	775	13.7	263	j3012_3	408	13.4	228	j3015_3	1064	13.5	249	j3018_3	410	13.4	243
j309_4	775	13.8	276	j3012_4	773	14.6	308	j3015_4	570	13.8	238	j3018_4	776	14.2	507
j309_5	1110	13.5	409	j3012_5	587	13.6	195	j3015_5	853	14.4	241	j3018_5	886	13.3	428
j309_6	520	13.5	321	j3012_6	770	13.8	258	j3015_6	1512	14.8	447	j3018_6	579	13.6	389
j309_7	468	13.5	205	j3012_7	985	14.3	226	j3015_7	577	13.6	282	j3018_7	679	13.3	367
j309_8	895	14.2	367	j3012_8	396	13.4	175	j3015_8	1111	13.6	320	j3018_8	501	14.0	305
j309_9	460	13.1	199	j3012_9	854	14.1	313	j3015_9	747	13.9	275	j3018_9	397	13.3	226
j309_10	1238	14.3	481	j3012_10	684	14.4	300	j3015_10	843	14.5	293	j3018_10	497	13.1	327

Table B.3. J30 Instances’ RID-MRD Solutions for RLP (3/5)

Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA
j3019_1	423	12.9	211	j3022_1	493	13.1	390	j3025_1	964	14.8	306	j3028_1	1558	14.5	443
j3019_2	697	13.8	401	j3022_2	526	13.2	319	j3025_2	956	13.7	302	j3028_2	1082	14.2	363
j3019_3	923	14.7	383	j3022_3	827	14.0	392	j3025_3	1166	14.0	294	j3028_3	340	13.6	212
j3019_4	427	12.9	210	j3022_4	413	13.6	325	j3025_4	1160	13.8	303	j3028_4	749	15.0	283
j3019_5	513	13.4	278	j3022_5	933	13.6	271	j3025_5	946	13.7	350	j3028_5	1551	15.3	416
j3019_6	586	13.0	253	j3022_6	855	13.7	349	j3025_6	681	13.7	311	j3028_6	1066	13.9	328
j3019_7	423	13.4	249	j3022_7	580	14.3	355	j3025_7	1186	14.8	584	j3028_7	616	14.1	365
j3019_8	575	13.7	264	j3022_8	871	13.7	206	j3025_8	908	13.6	196	j3028_8	1258	14.1	367
j3019_9	349	13.0	152	j3022_9	1168	14.7	680	j3025_9	1038	14.3	621	j3028_9	1141	14.6	299
j3019_10	675	13.1	317	j3022_10	677	14.2	343	j3025_10	761	13.4	310	j3028_10	909	14.3	378
j3020_1	1148	13.8	647	j3023_1	1164	14.3	667	j3026_1	1405	14.3	258	j3029_1	592	14.4	308
j3020_2	1030	14.6	463	j3023_2	956	13.6	381	j3026_2	416	13.6	233	j3029_2	772	14.4	306
j3020_3	296	13.5	148	j3023_3	716	13.7	255	j3026_3	1513	14.2	414	j3029_3	732	13.8	188
j3020_4	641	13.2	447	j3023_4	664	14.0	336	j3026_4	998	14.8	266	j3029_4	734	14.3	259
j3020_5	812	13.6	594	j3023_5	532	13.5	294	j3026_5	1119	14.5	366	j3029_5	1040	14.4	222
j3020_6	403	13.3	268	j3023_6	949	13.5	401	j3026_6	849	14.1	401	j3029_6	748	14.3	425
j3020_7	418	13.1	207	j3023_7	876	13.7	403	j3026_7	924	14.1	429	j3029_7	403	13.8	221
j3020_8	732	13.3	392	j3023_8	657	13.8	352	j3026_8	873	14.3	327	j3029_8	835	14.4	213
j3020_9	424	13.0	239	j3023_9	705	14.0	419	j3026_9	690	13.5	208	j3029_9	826	15.1	264
j3020_10	380	12.8	180	j3023_10	1078	14.4	406	j3026_10	681	14.2	236	j3029_10	1042	14.0	282
j3021_1	1269	14.1	581	j3024_1	613	13.9	279	j3027_1	814	13.5	232	j3030_1	577	13.8	227
j3021_2	585	13.5	372	j3024_2	826	13.7	275	j3027_2	630	14.6	430	j3030_2	1677	15.3	417
j3021_3	680	13.8	323	j3024_3	1063	14.8	390	j3027_3	712	14.2	223	j3030_3	915	14.3	332
j3021_4	624	13.7	311	j3024_4	1378	13.8	292	j3027_4	994	14.3	392	j3030_4	900	14.1	217
j3021_5	495	13.3	247	j3024_5	535	14.0	303	j3027_5	826	14.1	353	j3030_5	764	14.3	309
j3021_6	1269	13.9	369	j3024_6	1027	14.4	519	j3027_6	890	14.6	423	j3030_6	1232	14.8	373
j3021_7	1038	13.7	291	j3024_7	910	13.6	261	j3027_7	756	14.5	508	j3030_7	754	15.5	416
j3021_8	723	13.5	301	j3024_8	578	13.5	220	j3027_8	1163	15.3	453	j3030_8	470	13.9	210
j3021_9	665	13.4	209	j3024_9	567	13.9	369	j3027_9	894	14.7	281	j3030_9	577	14.5	242
j3021_10	1068	14.1	395	j3024_10	670	14.5	264	j3027_10	982	14.5	417	j3030_10	590	14.3	308

Table B.4. J30 Instances' RID-MRD Solutions for RLP (4/5)

Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA
j3031_1	847	14.1	238	j3034_1	1158	14.1	385	j3037_1	835	13.6	361	j3040_1	957	14.0	406
j3031_2	736	15.2	320	j3034_2	484	13.1	372	j3037_2	543	13.7	406	j3040_2	1195	14.4	411
j3031_3	1365	14.8	262	j3034_3	917	13.9	384	j3037_3	1030	13.8	557	j3040_3	1041	14.0	332
j3031_4	595	13.8	208	j3034_4	1159	14.1	748	j3037_4	824	14.2	359	j3040_4	896	14.4	456
j3031_5	765	14.0	404	j3034_5	795	13.9	444	j3037_5	809	14.4	473	j3040_5	852	14.3	320
j3031_6	854	13.9	300	j3034_6	719	13.9	408	j3037_6	955	13.7	742	j3040_6	1087	14.1	624
j3031_7	1336	14.7	502	j3034_7	592	13.9	437	j3037_7	663	14.1	389	j3040_7	630	14.2	450
j3031_8	1203	14.3	313	j3034_8	481	13.6	272	j3037_8	1050	14.4	284	j3040_8	1071	14.3	502
j3031_9	1086	13.8	345	j3034_9	942	13.7	432	j3037_9	827	13.5	374	j3040_9	791	14.5	416
j3031_10	888	14.4	301	j3034_10	465	13.5	342	j3037_10	1367	14.1	469	j3040_10	1157	14.0	555
j3032_1	871	14.7	327	j3035_1	577	13.8	361	j3038_1	647	14.0	345	j3041_1	1253	14.2	761
j3032_2	890	14.7	271	j3035_2	524	13.6	333	j3038_2	737	14.0	342	j3041_2	1038	14.2	523
j3032_3	819	14.6	344	j3035_3	656	13.5	459	j3038_3	676	14.0	301	j3041_3	899	14.2	276
j3032_4	1601	14.9	340	j3035_4	405	13.3	311	j3038_4	1028	14.4	546	j3041_4	772	14.0	364
j3032_5	609	14.7	198	j3035_5	609	13.8	261	j3038_5	1001	14.5	540	j3041_5	961	14.9	508
j3032_6	703	14.0	234	j3035_6	870	13.4	578	j3038_6	1203	14.5	657	j3041_6	1425	14.3	423
j3032_7	743	13.5	343	j3035_7	676	13.7	334	j3038_7	784	14.1	506	j3041_7	686	14.5	441
j3032_8	836	14.3	221	j3035_8	668	13.8	336	j3038_8	846	14.1	579	j3041_8	642	14.8	426
j3032_9	1186	15.1	365	j3035_9	898	13.8	597	j3038_9	847	14.1	493	j3041_9	1458	14.7	538
j3032_10	665	14.9	230	j3035_10	670	13.5	374	j3038_10	1574	14.4	688	j3041_10	1088	15.0	449
j3033_1	965	14.7	531	j3036_1	608	13.9	178	j3039_1	1084	14.2	538	j3042_1	835	14.2	285
j3033_2	527	13.7	369	j3036_2	370	13.2	264	j3039_2	1251	14.1	512	j3042_2	860	14.0	305
j3033_3	483	13.2	275	j3036_3	585	13.5	340	j3039_3	721	14.2	453	j3042_3	822	14.3	344
j3033_4	1292	14.3	539	j3036_4	1141	13.7	537	j3039_4	796	13.9	293	j3042_4	799	13.5	461
j3033_5	520	13.6	326	j3036_5	930	13.9	420	j3039_5	909	13.8	435	j3042_5	693	14.2	261
j3033_6	724	13.7	399	j3036_6	570	13.3	256	j3039_6	764	14.0	596	j3042_6	893	14.5	409
j3033_7	792	13.7	321	j3036_7	601	13.8	451	j3039_7	1235	13.2	596	j3042_7	2196	14.7	538
j3033_8	550	13.9	344	j3036_8	724	13.6	454	j3039_8	851	14.5	389	j3042_8	1109	14.6	505
j3033_9	942	13.9	576	j3036_9	573	13.7	204	j3039_9	855	14.1	540	j3042_9	1739	14.5	353
j3033_10	642	13.4	573	j3036_10	816	13.6	489	j3039_10	1110	13.8	669	j3042_10	1541	14.8	480

Table B.5. J30 Instances' RID-MRD Solutions for RLP (5/5)

Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA
j3043_1	1520	14.07	530	j3044_6	1168	14.19	516	j3046_1	1018	14.04	344	j3047_6	796	14.3	294
j3043_2	1184	13.74	460	j3044_7	659	13.87	458	j3046_2	1359	14.89	353	j3047_7	696	14.5	300
j3043_3	734	14.36	504	j3044_8	1034	13.98	527	j3046_3	1325	14.64	417	j3047_8	528	14.1	380
j3043_4	2024	14.25	505	j3044_9	1479	14.42	578	j3046_4	849	14.55	355	j3047_9	1209	14.7	578
j3043_5	785	14.51	431	j3044_10	1145	14.65	299	j3046_5	841	14.60	254	j3047_10	1012	14.5	361
j3043_6	763	14.06	400	j3045_1	682	13.88	259	j3046_6	1101	14.27	339	j3048_1	1236	14.1	438
j3043_7	977	14.18	303	j3045_2	1512	14.51	326	j3046_7	1048	13.82	361	j3048_2	1040	13.8	352
j3043_8	1243	14.43	559	j3045_3	1024	14.37	548	j3046_8	931	14.45	339	j3048_3	945	13.9	303
j3043_9	686	14.16	341	j3045_4	1406	14.61	350	j3046_9	941	13.75	242	j3048_4	679	14.3	257
j3043_10	1179	14.16	456	j3045_5	952	14.28	282	j3046_10	852	13.89	222	j3048_5	1031	14.5	264
j3044_1	869	13.87	267	j3045_6	1393	15.13	403	j3047_1	1430	14.2	475	j3048_6	824	14.4	328
j3044_2	1032	14.65	572	j3045_7	1806	14.94	368	j3047_2	1005	14.4	289	j3048_7	1359	14.4	330
j3044_3	1001	14.43	397	j3045_8	1361	14.21	530	j3047_3	811	14.6	363	j3048_8	1028	14.1	454
j3044_4	1197	14.66	883	j3045_9	723	14.01	263	j3047_4	654	13.7	283	j3048_9	1098	14.7	356
j3044_5	892	15.39	393	j3045_10	741	14.27	343	j3047_5	890	13.6	337	j3048_10	1276	14.2	415

Table B.6. J60 Instances' RID-MRD Solutions for RLP (1/4)

Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA
j601_1	1109	19.0	352	j604_1	1141	18.9	485	j607_1	1585	18.9	671	j6010_1	1563	19.4	574
j601_2	1063	18.1	388	j604_2	650	17.6	229	j607_2	1246	19.4	585	j6010_2	947	19.1	339
j601_3	1155	17.9	297	j604_3	709	17.9	189	j607_3	1182	18.9	570	j6010_3	2142	19.8	488
j601_4	1078	18.6	486	j604_4	1320	18.6	394	j607_4	1120	19.1	437	j6010_4	1921	19.4	541
j601_5	1129	18.2	474	j604_5	1196	18.4	370	j607_5	1665	18.3	705	j6010_5	1590	19.9	517
j601_6	976	17.4	255	j604_6	1273	17.9	440	j607_6	1339	18.4	412	j6010_6	1157	18.8	466
j601_7	800	18.2	346	j604_7	851	18.1	280	j607_7	1454	19.8	646	j6010_7	1178	19.2	523
j601_8	902	18.3	223	j604_8	799	18.3	199	j607_8	1053	18.4	369	j6010_8	1289	18.4	488
j601_9	1468	18.8	559	j604_9	1166	18.3	357	j607_9	744	17.5	381	j6010_9	981	19.6	440
j601_10	1369	18.6	551	j604_10	1353	18.3	652	j607_10	1245	19.1	575	j6010_10	1305	19.1	440
j602_1	829	17.9	327	j605_1	604	17.9	374	j608_1	1353	18.8	531	j6011_1	1929	19.8	662
j602_2	1059	18.4	511	j605_2	1375	19.2	540	j608_2	1217	19.0	406	j6011_2	1089	19.1	447
j602_3	1231	18.9	576	j605_3	1106	19.0	514	j608_3	2620	18.9	650	j6011_3	1580	20.3	564
j602_4	1233	18.3	455	j605_4	1067	19.4	308	j608_4	1205	18.6	359	j6011_4	1475	19.2	409
j602_5	1002	17.7	278	j605_5	2572	19.2	829	j608_5	1519	19.3	552	j6011_5	1345	19.2	366
j602_6	775	17.6	293	j605_6	1087	18.8	364	j608_6	1037	18.5	357	j6011_6	1268	19.3	421
j602_7	516	17.6	192	j605_7	1093	18.3	300	j608_7	1041	18.7	366	j6011_7	1271	19.8	424
j602_8	823	18.0	346	j605_8	1476	19.4	390	j608_8	916	18.4	435	j6011_8	1284	19.6	366
j602_9	703	18.0	198	j605_9	1615	19.9	640	j608_9	890	18.0	387	j6011_9	1000	18.8	431
j602_10	809	18.6	279	j605_10	1174	19.5	452	j608_10	1685	20.0	591	j6011_10	1005	19.1	407
j603_1	478	17.9	226	j606_1	1138	19.1	349	j609_1	702	18.8	379	j6012_1	1110	19.9	386
j603_2	979	18.1	336	j606_2	1253	19.0	340	j609_2	1587	18.7	591	j6012_2	1143	18.5	332
j603_3	1678	19.2	573	j606_3	1484	19.0	386	j609_3	1646	19.4	418	j6012_3	1521	19.9	620
j603_4	1317	18.4	523	j606_4	1078	18.9	289	j609_4	946	18.9	403	j6012_4	1406	19.8	420
j603_5	1155	18.9	421	j606_5	1371	20.5	700	j609_5	925	18.3	425	j6012_5	1133	19.0	335
j603_6	685	18.2	378	j606_6	695	18.4	295	j609_6	1660	20.6	688	j6012_6	991	18.5	363
j603_7	883	17.7	381	j606_7	1036	18.5	306	j609_7	965	19.5	417	j6012_7	1109	19.2	350
j603_8	704	17.4	218	j606_8	1366	18.9	514	j609_8	929	19.0	330	j6012_8	1092	18.4	371
j603_9	771	17.8	500	j606_9	1123	18.5	390	j609_9	1707	19.5	647	j6012_9	1137	18.9	437
j603_10	972	18.1	252	j606_10	1414	18.9	440	j609_10	1399	19.1	463	j6012_10	2067	19.4	574

Table B.7. J60 Instances’ RID-MRD Solutions for RLP (2/4)

Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA
j6013_1	897	19.3	433	j6016_1	1115	19.7	351	j6019_1	963	18.2	534	j6022_1	1510	18.4	500
j6013_2	2158	19.0	437	j6016_2	994	19.4	367	j6019_2	1283	19.1	606	j6022_2	2006	19.6	644
j6013_3	952	18.9	384	j6016_3	954	18.5	373	j6019_3	1481	18.6	573	j6022_3	1263	19.1	414
j6013_4	916	20.4	348	j6016_4	1105	18.6	354	j6019_4	916	17.8	379	j6022_4	898	20.1	437
j6013_5	1154	19.4	316	j6016_5	933	19.0	435	j6019_5	1322	18.4	566	j6022_5	1728	19.2	623
j6013_6	842	18.9	382	j6016_6	1055	19.5	460	j6019_6	1353	18.4	665	j6022_6	2169	20.2	603
j6013_7	885	19.3	345	j6016_7	2181	19.2	471	j6019_7	1218	17.7	480	j6022_7	1316	20.1	632
j6013_8	1126	20.0	371	j6016_8	996	19.5	355	j6019_8	1691	18.4	539	j6022_8	1270	18.9	545
j6013_9	850	19.8	449	j6016_9	1624	18.7	363	j6019_9	1170	18.2	446	j6022_9	1344	20.0	420
j6013_10	1654	19.4	365	j6016_10	1337	20.4	470	j6019_10	1200	18.3	413	j6022_10	1514	19.3	381
j6014_1	941	20.0	370	j6017_1	1008	18.2	458	j6020_1	934	18.2	372	j6023_1	1288	19.7	491
j6014_2	1234	20.0	438	j6017_2	982	17.7	480	j6020_2	1283	17.8	799	j6023_2	2024	20.3	739
j6014_3	964	19.3	376	j6017_3	1046	18.5	540	j6020_3	746	18.2	394	j6023_3	1899	19.4	827
j6014_4	1103	20.3	383	j6017_4	1886	17.8	609	j6020_4	1298	18.7	399	j6023_4	2595	19.4	662
j6014_5	1603	19.0	458	j6017_5	782	17.1	266	j6020_5	1221	18.2	458	j6023_5	1064	19.4	372
j6014_6	1370	19.0	396	j6017_6	1215	17.5	493	j6020_6	1183	19.2	789	j6023_6	2121	19.9	824
j6014_7	1427	20.0	539	j6017_7	1442	18.5	463	j6020_7	997	18.2	431	j6023_7	1096	18.4	454
j6014_8	1837	19.4	543	j6017_8	773	17.9	240	j6020_8	1326	17.9	405	j6023_8	1798	19.8	502
j6014_9	870	18.9	403	j6017_9	1469	17.9	645	j6020_9	1345	18.6	406	j6023_9	1508	18.9	490
j6014_10	989	20.2	400	j6017_10	893	18.2	361	j6020_10	1056	17.9	462	j6023_10	1338	18.4	449
j6015_1	2251	19.9	484	j6018_1	2039	18.2	649	j6021_1	2144	19.0	781	j6024_1	1524	19.4	661
j6015_2	1569	19.9	485	j6018_2	963	18.1	342	j6021_2	2960	19.2	734	j6024_2	1032	19.3	282
j6015_3	1121	19.7	397	j6018_3	1375	18.1	441	j6021_3	1247	18.5	464	j6024_3	1319	18.9	476
j6015_4	987	20.6	445	j6018_4	1396	18.4	582	j6021_4	1440	18.3	452	j6024_4	2271	20.6	772
j6015_5	1151	20.3	405	j6018_5	1500	18.2	460	j6021_5	1330	18.9	513	j6024_5	1206	19.2	703
j6015_6	1779	19.9	516	j6018_6	921	17.8	313	j6021_6	1309	18.4	582	j6024_6	1490	19.2	576
j6015_7	1300	19.3	385	j6018_7	1962	18.4	922	j6021_7	1642	19.4	558	j6024_7	1093	19.4	433
j6015_8	1299	19.8	428	j6018_8	1731	19.1	689	j6021_8	2347	19.5	549	j6024_8	1786	19.7	538
j6015_9	1089	20.0	415	j6018_9	957	18.0	344	j6021_9	1521	19.1	449	j6024_9	2243	19.9	651
j6015_10	1162	18.8	413	j6018_10	1813	19.5	742	j6021_10	1084	17.6	519	j6024_10	1844	18.9	797

Table B.8. J60 Instances' RID-MRD Solutions for RLP (3/4)

Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA
j6025_1	1788	20.2	510	j6028_1	3016	20.2	599	j6031_1	1030	19.8	473	j6034_1	1206	19.0	494
j6025_2	1058	20.4	519	j6028_2	1340	18.8	501	j6031_2	1364	20.9	562	j6034_2	1144	18.8	583
j6025_3	2155	20.1	658	j6028_3	1992	18.9	650	j6031_3	950	19.9	366	j6034_3	1229	18.7	619
j6025_4	1458	20.3	477	j6028_4	1977	19.6	574	j6031_4	1214	20.0	499	j6034_4	1255	19.2	748
j6025_5	1387	19.2	462	j6028_5	1710	19.1	547	j6031_5	892	20.2	352	j6034_5	1374	19.0	730
j6025_6	2267	20.7	558	j6028_6	1517	19.9	604	j6031_6	2024	21.2	580	j6034_6	1963	19.8	655
j6025_7	2026	18.7	474	j6028_7	1114	19.3	453	j6031_7	1489	20.4	538	j6034_7	1966	19.4	567
j6025_8	1135	18.8	409	j6028_8	1390	18.8	705	j6031_8	1486	20.5	484	j6034_8	1176	18.3	609
j6025_9	1568	18.7	574	j6028_9	1498	19.4	437	j6031_9	1421	21.2	497	j6034_9	1374	19.0	549
j6025_10	1492	19.6	638	j6028_10	1299	20.2	648	j6031_10	903	19.7	326	j6034_10	1770	20.1	544
j6026_1	2063	19.2	627	j6029_1	1355	18.7	443	j6032_1	822	19.8	390	j6035_1	1476	19.5	602
j6026_2	1607	18.5	686	j6029_2	3306	19.7	478	j6032_2	2423	21.8	730	j6035_2	1372	19.4	834
j6026_3	1953	20.5	575	j6029_3	1569	19.5	472	j6032_3	1879	21.4	544	j6035_3	1531	19.6	557
j6026_4	1698	18.6	442	j6029_4	1337	19.8	391	j6032_4	1190	19.5	377	j6035_4	1328	18.8	638
j6026_5	1063	18.6	542	j6029_5	2311	19.5	617	j6032_5	1531	20.8	495	j6035_5	1354	19.3	728
j6026_6	1855	19.2	568	j6029_6	1570	20.7	461	j6032_6	3560	21.0	709	j6035_6	1359	19.3	668
j6026_7	1809	18.9	639	j6029_7	1207	19.1	571	j6032_7	1398	20.5	500	j6035_7	1060	18.9	569
j6026_8	1619	19.8	918	j6029_8	1066	19.3	513	j6032_8	2157	19.3	609	j6035_8	1381	19.3	608
j6026_9	1236	19.2	460	j6029_9	1751	19.2	400	j6032_9	1943	20.1	484	j6035_9	1140	18.8	671
j6026_10	1705	19.4	767	j6029_10	1440	19.5	430	j6032_10	1440	20.6	425	j6035_10	940	18.7	488
j6027_1	3273	20.1	750	j6030_1	1168	19.0	487	j6033_1	2147	19.6	1071	j6036_1	1156	18.2	542
j6027_2	1771	18.6	678	j6030_2	1058	19.2	366	j6033_2	1854	19.8	869	j6036_2	1430	17.9	715
j6027_3	2353	19.0	604	j6030_3	1653	20.4	453	j6033_3	1593	18.7	549	j6036_3	1535	18.9	728
j6027_4	1158	18.6	450	j6030_4	1319	19.4	549	j6033_4	1227	19.2	646	j6036_4	1488	19.1	634
j6027_5	1525	19.5	707	j6030_5	1901	20.6	503	j6033_5	1668	20.0	720	j6036_5	1048	18.2	444
j6027_6	1170	19.3	401	j6030_6	1267	18.9	388	j6033_6	1072	18.9	495	j6036_6	1179	19.1	483
j6027_7	3782	19.7	535	j6030_7	1512	20.4	550	j6033_7	1152	19.2	438	j6036_7	1092	19.1	415
j6027_8	1508	19.7	639	j6030_8	891	20.1	467	j6033_8	948	19.0	403	j6036_8	1084	18.4	458
j6027_9	1855	19.3	757	j6030_9	2246	21.5	558	j6033_9	1808	20.0	939	j6036_9	1240	19.0	704
j6027_10	1200	18.9	365	j6030_10	1266	21.1	403	j6033_10	1553	18.8	575	j6036_10	1186	18.9	738

Table B.9. J60 Instances' RID-MRD Solutions for RLP (4/4)

Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA
j6037_1	1751	19.1	562	j6040_1	1946	20.4	934	j6043_1	2428	21.4	1202	j6046_1	1916	19.3	550
j6037_2	1124	19.5	583	j6040_2	2480	20.0	569	j6043_2	1979	21.4	637	j6046_2	1445	19.6	524
j6037_3	2529	21.2	875	j6040_3	1805	19.6	613	j6043_3	1701	21.2	880	j6046_3	1870	19.5	533
j6037_4	1478	19.6	784	j6040_4	1646	20.1	574	j6043_4	1166	21.2	641	j6046_4	1376	19.4	511
j6037_5	2665	19.4	715	j6040_5	2720	20.7	1294	j6043_5	1450	19.8	601	j6046_5	1635	19.8	645
j6037_6	1451	19.6	614	j6040_6	1497	19.0	796	j6043_6	2195	20.6	548	j6046_6	1724	19.5	643
j6037_7	1583	20.1	602	j6040_7	1406	19.2	620	j6043_7	3131	20.0	664	j6046_7	2177	19.1	532
j6037_8	1692	19.9	619	j6040_8	1712	19.9	576	j6043_8	2577	20.1	778	j6046_8	1358	19.5	530
j6037_9	2013	20.0	744	j6040_9	2006	20.2	933	j6043_9	1801	19.1	848	j6046_9	757	19.0	385
j6037_10	2047	20.3	568	j6040_10	1466	19.3	725	j6043_10	1780	19.3	715	j6046_10	2378	20.1	596
j6038_1	1155	19.9	597	j6041_1	2121	20.9	718	j6044_1	2170	19.5	632	j6047_1	2642	19.5	566
j6038_2	1710	20.3	674	j6041_2	2280	20.2	552	j6044_2	1770	19.3	558	j6047_2	1792	19.9	483
j6038_3	1813	20.1	766	j6041_3	1231	19.0	482	j6044_3	2125	19.7	743	j6047_3	1542	19.9	514
j6038_4	1461	18.5	473	j6041_4	1951	20.5	1072	j6044_4	2056	19.8	601	j6047_4	2061	19.3	668
j6038_5	2745	20.7	996	j6041_5	1761	20.3	583	j6044_5	2330	18.9	789	j6047_5	2100	20.7	578
j6038_6	1654	20.0	771	j6041_6	2400	20.5	901	j6044_6	2347	19.2	647	j6047_6	1604	19.5	552
j6038_7	2027	19.9	701	j6041_7	2144	20.6	772	j6044_7	1087	19.0	538	j6047_7	1451	19.6	530
j6038_8	1334	19.7	692	j6041_8	2507	20.7	851	j6044_8	1817	19.4	590	j6047_8	1376	19.3	533
j6038_9	1535	19.0	660	j6041_9	1985	20.3	997	j6044_9	1461	19.1	514	j6047_9	1932	19.6	440
j6038_10	1493	19.5	606	j6041_10	1329	20.6	575	j6044_10	1592	18.8	545	j6047_10	1336	19.6	492
j6039_1	2212	20.1	654	j6042_1	1977	20.2	593	j6045_1	1711	18.9	569	j6048_1	1022	19.3	422
j6039_2	2166	20.4	907	j6042_2	1299	19.5	498	j6045_2	2122	19.9	451	j6048_2	1318	19.8	566
j6039_3	2038	20.4	580	j6042_3	1544	19.8	557	j6045_3	1487	20.0	610	j6048_3	1245	21.1	455
j6039_4	2299	19.8	1086	j6042_4	3042	21.3	967	j6045_4	1357	18.5	440	j6048_4	1412	18.9	476
j6039_5	1559	19.5	654	j6042_5	1788	20.0	694	j6045_5	868	18.7	402	j6048_5	4531	21.2	838
j6039_6	2520	19.4	642	j6042_6	1703	20.4	799	j6045_6	1684	20.3	624	j6048_6	1112	20.2	558
j6039_7	1025	19.5	563	j6042_7	1183	18.9	469	j6045_7	1798	19.0	570	j6048_7	2144	20.8	521
j6039_8	1769	19.4	726	j6042_8	2204	20.5	1026	j6045_8	1639	19.8	531	j6048_8	2290	21.0	531
j6039_9	1736	19.6	572	j6042_9	1755	19.9	714	j6045_9	1477	19.5	549	j6048_9	1121	20.5	604
j6039_10	1836	19.3	563	j6042_10	2294	20.3	807	j6045_10	1018	18.8	420	j6048_10	1867	20.0	454

Table B.10. J120 Instances' RID-MRD Solutions for RLP (1/4)

Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA
j1201_1	2237	28.8	582	j1204_1	1387	26.4	527	j1207_1	1670	29.5	520	j12010_1	3069	31.3	674
j1201_2	2053	27.2	598	j1204_2	2714	28.3	730	j1207_2	2097	29.6	591	j12010_2	1772	30.1	666
j1201_3	1661	27.6	454	j1204_3	2029	28.2	544	j1207_3	1493	29.1	583	j12010_3	2516	30.5	619
j1201_4	1803	27.5	402	j1204_4	1583	27.3	565	j1207_4	2127	28.2	726	j12010_4	2626	29.9	640
j1201_5	2185	27.8	720	j1204_5	1413	28.2	439	j1207_5	2046	29.3	646	j12010_5	1938	29.1	666
j1201_6	975	26.6	264	j1204_6	1017	28.7	387	j1207_6	2799	29.3	764	j12010_6	2504	28.6	516
j1201_7	2521	28.1	584	j1204_7	1823	26.9	553	j1207_7	2169	29.2	697	j12010_7	2050	29.3	699
j1201_8	1640	26.9	557	j1204_8	1519	27.5	494	j1207_8	1612	28.4	540	j12010_8	2943	29.9	861
j1201_9	1920	27.5	489	j1204_9	1838	27.0	535	j1207_9	1343	28.8	686	j12010_9	1631	28.4	594
j1201_10	1759	27.4	435	j1204_10	1157	26.5	372	j1207_10	2616	29.4	607	j12010_10	1283	28.2	469
j1202_1	1517	26.8	445	j1205_1	1435	27.7	434	j1208_1	2068	31.9	596	j12011_1	2554	29.4	753
j1202_2	1215	27.0	432	j1205_2	1872	27.6	457	j1208_2	2248	30.4	583	j12011_2	2153	29.2	599
j1202_3	1216	28.0	396	j1205_3	1303	28.3	520	j1208_3	2687	29.8	568	j12011_3	1770	30.4	674
j1202_4	1785	27.2	642	j1205_4	1691	28.9	664	j1208_4	1865	29.3	577	j12011_4	2608	30.8	725
j1202_5	2213	28.1	628	j1205_5	995	28.0	384	j1208_5	2060	30.4	539	j12011_5	2212	31.7	842
j1202_6	1491	27.7	667	j1205_6	1361	28.3	480	j1208_6	1957	28.4	551	j12011_6	1890	30.0	551
j1202_7	1853	27.1	672	j1205_7	1257	28.8	520	j1208_7	2121	28.7	510	j12011_7	2353	29.2	719
j1202_8	1299	26.7	405	j1205_8	1157	28.3	391	j1208_8	1788	28.7	601	j12011_8	2105	30.7	774
j1202_9	2065	28.0	589	j1205_9	2082	28.9	675	j1208_9	1488	28.8	493	j12011_9	1538	30.2	583
j1202_10	1381	27.7	513	j1205_10	1790	28.5	725	j1208_10	1620	29.2	608	j12011_10	2309	29.6	760
j1203_1	1834	27.0	416	j1206_1	2476	29.0	551	j1209_1	2250	29.0	813	j12012_1	2569	29.9	630
j1203_2	2288	27.4	539	j1206_2	1804	29.7	547	j1209_2	2142	29.6	685	j12012_2	1707	28.9	548
j1203_3	1603	27.7	456	j1206_3	1834	29.6	541	j1209_3	2303	29.5	553	j12012_3	2058	30.4	611
j1203_4	1395	26.7	391	j1206_4	2655	29.7	796	j1209_4	1771	29.7	505	j12012_4	1987	30.8	663
j1203_5	1724	26.6	516	j1206_5	1627	29.2	515	j1209_5	2618	30.2	713	j12012_5	2535	31.5	689
j1203_6	1805	27.5	713	j1206_6	1852	29.6	560	j1209_6	2450	30.7	847	j12012_6	2631	29.8	509
j1203_7	1940	27.2	671	j1206_7	1969	30.5	684	j1209_7	1618	29.6	589	j12012_7	2034	30.3	633
j1203_8	1807	27.2	576	j1206_8	1959	29.6	846	j1209_8	1770	29.8	733	j12012_8	2160	30.4	569
j1203_9	1537	27.2	488	j1206_9	1847	29.2	557	j1209_9	2703	29.6	794	j12012_9	1650	30.3	576
j1203_10	1997	28.0	805	j1206_10	2355	29.9	529	j1209_10	1661	30.4	610	j12012_10	2176	31.3	720

Table B.11. J120 Instances' RID-MRD Solutions for RLP (2/4)

Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA
j12013_1	2800	31.8	823	j12016_1	1688	29.9	520	j12019_1	1383	31.6	622	j12022_1	1974	28.3	692
j12013_2	1314	30.0	520	j12016_2	1568	30.6	594	j12019_2	2164	30.5	566	j12022_2	2737	28.9	774
j12013_3	2823	32.6	672	j12016_3	2226	31.4	623	j12019_3	1990	28.9	591	j12022_3	1893	28.2	603
j12013_4	1923	30.6	602	j12016_4	1676	30.7	589	j12019_4	2737	29.6	582	j12022_4	1440	28.5	488
j12013_5	1401	29.3	515	j12016_5	1441	30.8	646	j12019_5	1943	29.6	677	j12022_5	1972	29.4	674
j12013_6	1907	29.7	684	j12016_6	1511	30.7	610	j12019_6	1638	29.5	637	j12022_6	1590	28.6	596
j12013_7	2610	30.9	647	j12016_7	1466	31.4	612	j12019_7	2345	29.8	667	j12022_7	3847	29.5	1095
j12013_8	2074	30.8	667	j12016_8	1468	29.7	574	j12019_8	1959	29.8	672	j12022_8	1540	28.6	647
j12013_9	1489	29.2	568	j12016_9	1694	31.0	668	j12019_9	1211	30.0	582	j12022_9	2516	28.6	679
j12013_10	1776	30.2	609	j12016_10	1701	31.2	635	j12019_10	1864	31.4	683	j12022_10	1930	28.0	617
j12014_1	1345	31.0	614	j12017_1	1689	31.4	654	j12020_1	1738	31.0	641	j12023_1	1887	29.2	746
j12014_2	1705	32.2	616	j12017_2	1712	29.6	566	j12020_2	1862	30.9	650	j12023_2	2153	29.7	828
j12014_3	2093	30.4	576	j12017_3	1394	29.5	590	j12020_3	1855	30.9	621	j12023_3	2376	28.6	562
j12014_4	1601	30.1	634	j12017_4	1760	31.5	651	j12020_4	1611	30.7	538	j12023_4	2350	28.9	895
j12014_5	2034	30.2	669	j12017_5	1942	30.9	614	j12020_5	1199	29.6	531	j12023_5	2074	29.0	807
j12014_6	2007	30.3	830	j12017_6	1379	29.7	560	j12020_6	1571	29.1	570	j12023_6	2136	29.0	669
j12014_7	2203	31.0	575	j12017_7	2224	30.8	713	j12020_7	1483	29.6	583	j12023_7	1499	28.2	567
j12014_8	4028	31.4	686	j12017_8	1451	29.8	532	j12020_8	3346	31.2	691	j12023_8	3059	28.0	664
j12014_9	2647	31.1	681	j12017_9	1656	30.0	625	j12020_9	1464	31.1	560	j12023_9	2859	28.8	1025
j12014_10	1745	30.1	599	j12017_10	1329	31.2	572	j12020_10	1446	31.3	557	j12023_10	1847	29.7	469
j12015_1	1889	31.3	611	j12018_1	2403	32.3	780	j12021_1	2408	28.5	652	j12024_1	2518	28.2	757
j12015_2	1493	30.1	527	j12018_2	3536	32.5	737	j12021_2	1836	28.0	651	j12024_2	1985	27.9	684
j12015_3	2062	30.2	626	j12018_3	1696	29.8	596	j12021_3	2828	29.6	1250	j12024_3	2363	28.9	481
j12015_4	1668	30.8	521	j12018_4	1405	30.2	534	j12021_4	2737	29.0	904	j12024_4	1859	28.8	644
j12015_5	1987	30.5	592	j12018_5	1774	30.6	730	j12021_5	2398	28.7	966	j12024_5	2615	28.0	652
j12015_6	2266	30.5	717	j12018_6	3874	30.2	817	j12021_6	2042	28.3	643	j12024_6	2218	29.0	800
j12015_7	1790	29.3	552	j12018_7	1689	30.6	652	j12021_7	1422	28.0	462	j12024_7	2806	29.6	645
j12015_8	2747	31.1	905	j12018_8	1988	30.4	599	j12021_8	2079	29.6	738	j12024_8	2601	28.6	706
j12015_9	1953	31.4	859	j12018_9	1329	30.5	564	j12021_9	2077	28.0	646	j12024_9	1677	27.4	595
j12015_10	1870	30.6	657	j12018_10	1837	31.0	572	j12021_10	1433	28.4	612	j12024_10	3333	28.1	794

Table B.12. J120 Instances' RID-MRD Solutions for RLP (3/4)

Instances	$ES_{\text{RID-MRD}}$	Time (S)	MASA	Instances	$ES_{\text{RID-MRD}}$	Time (S)	MASA	Instances	$ES_{\text{RID-MRD}}$	Time (S)	MASA	Instances	$ES_{\text{RID-MRD}}$	Time (S)	MASA
j12025_1	1538	28.3	621	j12028_1	2736	30.8	709	j12031_1	2262	30.5	795	j12034_1	1577	28.4	598
j12025_2	1959	28.6	767	j12028_2	3245	30.4	1127	j12031_2	2441	30.2	779	j12034_2	2746	29.9	671
j12025_3	2352	28.9	636	j12028_3	2391	30.1	992	j12031_3	2134	29.6	652	j12034_3	2138	30.4	907
j12025_4	2694	29.7	974	j12028_4	3165	31.1	838	j12031_4	3966	31.2	773	j12034_4	2225	30.0	722
j12025_5	2428	28.5	756	j12028_5	2644	30.0	977	j12031_5	1934	31.3	678	j12034_5	3030	29.9	833
j12025_6	2190	29.1	648	j12028_6	2106	30.1	802	j12031_6	2144	31.9	815	j12034_6	2632	29.9	827
j12025_7	2247	28.6	611	j12028_7	3422	30.4	855	j12031_7	2389	31.7	824	j12034_7	2882	30.2	1005
j12025_8	1896	28.1	852	j12028_8	2036	30.2	803	j12031_8	2255	31.1	743	j12034_8	1671	29.4	741
j12025_9	2357	28.2	649	j12028_9	2390	29.3	680	j12031_9	1876	30.0	689	j12034_9	2114	28.8	611
j12025_10	1798	27.8	538	j12028_10	4179	30.8	878	j12031_10	2344	30.0	854	j12034_10	2863	29.9	789
j12026_1	2373	31.0	757	j12029_1	3233	30.5	1128	j12032_1	2737	30.3	707	j12035_1	2462	29.2	735
j12026_2	3890	29.8	1131	j12029_2	3312	30.0	1101	j12032_2	3273	31.3	794	j12035_2	3687	31.0	914
j12026_3	2399	30.8	714	j12029_3	2266	30.7	692	j12032_3	2516	31.3	755	j12035_3	1879	29.8	681
j12026_4	2749	30.5	978	j12029_4	1609	29.5	505	j12032_4	3313	30.4	1170	j12035_4	3031	29.9	853
j12026_5	3168	29.1	780	j12029_5	3201	31.1	824	j12032_5	3103	30.7	716	j12035_5	2403	29.3	852
j12026_6	3610	31.7	901	j12029_6	2053	30.3	738	j12032_6	2193	30.3	636	j12035_6	2100	28.8	738
j12026_7	2362	29.1	750	j12029_7	2662	30.1	981	j12032_7	2793	31.2	981	j12035_7	2533	29.9	865
j12026_8	3781	30.3	1284	j12029_8	2287	29.3	917	j12032_8	2585	29.4	580	j12035_8	2246	30.4	860
j12026_9	3361	31.4	1161	j12029_9	3202	29.1	1025	j12032_9	2188	29.8	923	j12035_9	2328	29.7	895
j12026_10	4539	31.3	1395	j12029_10	3159	29.2	755	j12032_10	3409	31.3	789	j12035_10	1927	29.4	826
j12027_1	1686	29.5	710	j12030_1	2175	30.0	708	j12033_1	2712	30.6	798	j12036_1	2626	30.4	703
j12027_2	3243	29.6	740	j12030_2	2821	31.6	964	j12033_2	3321	30.4	846	j12036_2	1812	29.5	632
j12027_3	3725	30.7	850	j12030_3	3108	30.5	931	j12033_3	2331	30.3	633	j12036_3	1965	29.9	636
j12027_4	1570	29.9	685	j12030_4	2376	30.0	649	j12033_4	2585	30.2	886	j12036_4	2917	29.6	865
j12027_5	2002	30.2	659	j12030_5	3084	29.1	680	j12033_5	3823	30.2	820	j12036_5	2262	30.0	793
j12027_6	2777	30.4	597	j12030_6	1981	29.3	604	j12033_6	2991	30.1	928	j12036_6	2142	30.1	840
j12027_7	2921	30.1	1026	j12030_7	1726	31.0	721	j12033_7	3418	29.6	636	j12036_7	3040	30.8	925
j12027_8	3878	31.2	799	j12030_8	2622	30.0	665	j12033_8	2913	29.6	754	j12036_8	1893	28.3	731
j12027_9	2572	31.0	723	j12030_9	2915	30.2	924	j12033_9	3642	29.7	908	j12036_9	2299	29.5	754
j12027_10	2902	31.3	1047	j12030_10	2214	29.3	860	j12033_10	1805	29.3	827	j12036_10	2047	30.2	794

Table B.13. J120 Instances' RID-MRD Solutions for RLP (4/4)

Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA	Instances	ES RID-MRD	Time (S)	MASA
j12037_1	2321	29.4	688	j12040_1	3029	28.5	557	j12043_1	2203	28.0	717	j12046_1	3189	31.3	1211
j12037_2	1938	29.3	644	j12040_2	1978	30.2	689	j12043_2	3439	28.1	1319	j12046_2	3851	31.5	1302
j12037_3	3865	30.0	779	j12040_3	2083	29.4	740	j12043_3	2347	27.6	928	j12046_3	2718	30.5	1112
j12037_4	2283	30.6	743	j12040_4	2463	30.9	858	j12043_4	1814	28.1	844	j12046_4	2494	29.2	775
j12037_5	3479	31.1	866	j12040_5	3163	29.6	724	j12043_5	2389	27.7	1166	j12046_5	2513	29.1	1029
j12037_6	2121	31.3	880	j12040_6	1920	30.0	695	j12043_6	2921	27.3	1180	j12046_6	3313	28.7	849
j12037_7	3989	29.9	637	j12040_7	2132	29.6	702	j12043_7	2312	28.9	1231	j12046_7	3566	29.6	1235
j12037_8	2216	30.4	744	j12040_8	2289	29.6	663	j12043_8	2715	28.0	925	j12046_8	2481	29.3	688
j12037_9	1537	29.2	608	j12040_9	3373	30.9	764	j12043_9	2678	28.0	1050	j12046_9	2829	29.2	878
j12037_10	2402	29.1	556	j12040_10	1767	30.5	690	j12043_10	2982	27.9	1089	j12046_10	2630	30.1	1020
j12038_1	2338	30.2	799	j12041_1	2529	28.4	723	j12044_1	2688	27.6	853	j12047_1	4428	30.5	939
j12038_2	2647	29.6	819	j12041_2	2513	27.8	869	j12044_2	2314	28.0	621	j12047_2	4293	30.4	1386
j12038_3	4520	30.5	730	j12041_3	2897	28.5	1046	j12044_3	2296	27.9	1120	j12047_3	3009	31.7	938
j12038_4	5796	30.6	802	j12041_4	2360	26.8	968	j12044_4	1751	27.9	553	j12047_4	2505	30.4	878
j12038_5	3306	30.2	780	j12041_5	3149	28.2	1040	j12044_5	2094	28.2	960	j12047_5	2678	30.3	964
j12038_6	2284	30.6	797	j12041_6	2772	27.3	1033	j12044_6	2609	28.2	934	j12047_6	3060	30.7	1128
j12038_7	2272	29.7	663	j12041_7	2062	27.3	654	j12044_7	3127	28.0	1010	j12047_7	2535	30.8	857
j12038_8	2324	30.1	762	j12041_8	3573	28.6	1174	j12044_8	2210	27.8	991	j12047_8	3041	29.7	829
j12038_9	3601	31.7	1258	j12041_9	2046	27.5	746	j12044_9	2739	27.4	751	j12047_9	3137	30.5	1031
j12038_10	3072	30.5	632	j12041_10	3276	28.3	1225	j12044_10	2183	28.4	878	j12047_10	3483	30.6	1047
j12039_1	3274	29.7	963	j12042_1	2051	27.6	810	j12045_1	2577	27.9	1045	j12048_1	2918	29.9	779
j12039_2	2696	30.0	938	j12042_2	2630	28.5	1109	j12045_2	2266	27.6	732	j12048_2	3339	30.4	945
j12039_3	2598	29.8	871	j12042_3	2618	27.7	1021	j12045_3	2292	28.5	1072	j12048_3	3164	30.8	1359
j12039_4	2680	28.9	621	j12042_4	2747	27.9	903	j12045_4	2899	28.1	1139	j12048_4	2926	30.8	1119
j12039_5	2619	29.2	800	j12042_5	2308	28.2	815	j12045_5	2854	28.8	1183	j12048_5	3372	30.6	850
j12039_6	1663	29.8	779	j12042_6	2139	27.4	961	j12045_6	2998	29.6	1218	j12048_6	2548	30.1	761
j12039_7	2914	30.2	845	j12042_7	3015	27.9	786	j12045_7	2734	28.1	1197	j12048_7	2845	30.6	773
j12039_8	2578	29.1	618	j12042_8	2540	28.1	938	j12045_8	2270	28.3	1013	j12048_8	3544	30.3	908
j12039_9	2201	29.3	675	j12042_9	2445	27.7	1174	j12045_9	2743	28.9	995	j12048_9	3730	30.5	1028
j12039_10	3349	29.6	866	j12042_10	2582	28.5	935	j12045_10	2009	28.2	656	j12048_10	4307	30.1	958

CURRICULUM VITAE

Mahdi Abbasi Iranagh
Date of birth: Oct. 12, 1975
Place of birth: Tabriz, Iran
PhD Candidate
Civil Engineering Dep.
Middle East Technical University, Ankara, Turkey
Email: mairanagh@gmail.com

PROFILE

Highly self-motivated PhD candidate with demonstrated research expertise resource optimization of construction projects.

Rich experience in modeling optimization algorithms using such as genetic algorithms, particle swarm optimization, etc.

Computer skills: C++, C#; Microsoft Project, Primavera; Microsoft Office, etc.

Eight years of experience in construction of residential and industrial projects.

EDUCATION

Middle East Technical University, Ankara, Turkey Feb 2011-July 2015 (expected)
Doctor of Philosophy (PhD), Civil Engineering,
Construction Engineering and Management
GPA = 3.79 / 4

Shahid Beheshti University, Tehran, Iran Sep 2002 - Sep 2006
Master's degree, Project & Construction Management
GPA = 15.68 / 20

Islamic Azad University, Tabriz, Iran Sep 1995 - Oct 1999
Bachelor's degree, Civil Engineering
GPA = 15.59 / 20

WORK EXPERIENCE

Research Assistant Mar 2014 - Present
Civil Engineering Department, Construction Engineering and Management Division
Middle East Technical University, Ankara, Turkey
Research assistant in the project titled “An Efficient Memetic Algorithm for the Solving Discrete Time-Cost Trade-Off Problem for Construction Projects“

Research Assistant Oct 2011 – Mar 2014
Civil Engineering Department, Construction Engineering and Management Division
Middle East Technical University, Ankara, Turkey
Research assistant in the project titled “Development of High Performance Exact and Meta-Heuristic Algorithms for Resource Leveling Problem in Construction Projects“

- Evaluating the performance of construction management commercial software programs (MS Project and Primavera) in resource optimization.
- Developing a high performance meta-heuristic algorithm for resource leveling problem in construction projects.
- Integrating the developed algorithm with the MS Project software to enhance its performance in resource leveling.

Supervisor (Inspector) Engineer Jan 2009 - Apr 2010
Tehran Mohaseb Consulting Engineers, Tehran, Iran
Construction Project of 1000-unit El-Goli Residential Complex , Tabriz, Iran.
Responsible as the inspector engineer for construction of blocks E and B.

Site Supervisor Oct 2004 - Sep 2008
Fathi Contractorship, Tabriz, Iran
Construction Project of 233-unit Mehr Residential Complex , Tabriz, Iran.
Responsible as the site supervisor and representative of Fathi Contractorship, the subcontractor of structural and brick-works of the project.

Site Supervisor Nov 2001 - Mar 2003
Beton Bastar Engineering Company, Tabriz, Iran
Ghaed Bassir Petrochemical Plant project, Golpayegan, Iran.
Responsible as the construction site supervisor for the Compounding unit construction.

Site Engineer Aug 2000 - Nov 2001
Nobar Charitable Society, Tabriz, Iran.
Construction Project of 96-unit Residential Complex, Tabriz, Iran.
Responsible as construction site engineer.

Technical Office Engineer Jul 1999 - Aug 2000
Tabriz Islamic Azad University, Tabriz, Iran.
Technical Office of Tabriz Islamic Azad University.
Responsible as Technical Office Engineer.

PUBLICATIONS

A Hybrid Genetic Algorithm for Resource Leveling of Large Scale Construction Projects (in preparation).

A Critical Sequence Crashing Heuristic for Resource Constrained Discrete Time-Cost Trade-Off Problem. Submitted for publishing in “Journal of Construction Engineering and Management” (under second review).

A Memetic Algorithm Approach for the Resource Leveling Problem. Submitted for publishing in “Applied Soft Computing” (under second review).

A Mixed-Integer Linear Model for Optimization of Resource Idle Days In Project Scheduling. Proceedings Creative Construction Conference 2013 July 6–9, 2013, Budapest, Hungary, 368–381.

A Genetic Algorithm for Resource Leveling of Construction Projects.

Proceedings 28th Annual ARCOM Conference, 3-5 September 2012, Edinburgh, UK. Association of Researchers in Construction Management, 1047–54.
September 3, 2012

MS Project Paket Programlarının Kaynak Dengeleme Problemi Çözümündeki Performansı. 2. Proje ve Yapım Yönetimi Kongresi, 13 – 16 Eylül 2012 İzmir Yüksek Teknoloji Enstitüsü, Urla-İzmir
September 13, 2012

CERTIFICATES

The engineering occupational license for supervision and execution (Rank 2) from "Ministry of Road and Urban Development of Iran (East Azarbaijan Organization of Construction Engineering Council)"

PROFESSIONAL MEMBERSHIPS

Member of East Azarbaijan Organization of Construction Engineering Council, I.R. Iran