

**A LINEAR PROGRAMMING BASED METHOD FOR THE
RESOURCE CONSTRAINED MULTI-PROJECT SCHEDULING
PROBLEM WITH WEIGHTED EARLINESS/TARDINESS COSTS**

by

MEHMET BERKE PAMAY

Submitted to the Graduate School of Engineering and Natural Sciences

in partial fulfillment of

the requirements for the degree of

Master of Science

Sabancı University

Spring 2011

A LINEAR PROGRAMMING BASED METHOD FOR THE RESOURCE
CONSTRAINED MULTI-PROJECT SCHEDULING PROBLEM WITH
WEIGHTED EARLINESS/TARDINESS COSTS

APPROVED BY

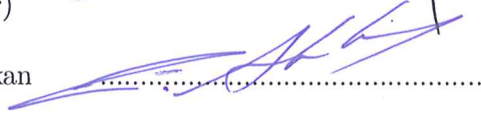
Assist. Prof. Kerem Bülbül
(Thesis Co-Supervisor)



Prof. Gündüz Ulusoy
(Thesis Co-Supervisor)



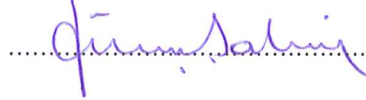
Assoc. Prof. Can Akkan



Prof. Ümit Bilge



Assoc. Prof. Güvenç Şahin



DATE OF APPROVAL: 21.10.2011

©Mehmet Berke Pamay 2011

All Rights Reserved

to my beloved ones

Acknowledgments

This thesis would have not been accomplished without the love and support of many very special people.

First, I want to express my deepest gratitude to my thesis advisor Assoc. Prof. Kerem Blbl for his remarkable patience and invaluable support. I feel extremely fortunate to have him as my advisor. I wish to express my deep acknowledgments to Prof. Gndz Ulusoy, who was not only my second advisor but also a great mentor.

My special thanks goes to Bihter Abdullahođulları for giving me the courage to follow my heart and for inspiring me to work hard. Soner Beyhan deserves a special mention, who has been a precious friend for more than a decade.

I am indebted to all my friends from Sabanci University for their motivation and endless friendship. Many special thanks go to Halil Ően, Őetin A. Suyabatmaz, Taner L. Tunç, Mahir U. Yldrm, E. Arda ŐiŐbot and all those others who directly and indirectly helped me.

Above all, I would like to thank my family for always being there for me. My parents and my brother supported me and showed great love, patience and support at all times.

A LINEAR PROGRAMMING BASED METHOD FOR THE RESOURCE
CONSTRAINED MULTI-PROJECT SCHEDULING PROBLEM WITH
WEIGHTED EARLINESS/TARDINESS COSTS

Mehmet Berke Pamay

Industrial Engineering, Master of Science Thesis, 2011

Thesis Supervisors:

Assist. Prof. Kerem Bülbül

Prof. Gündüz Ulusoy

Keywords: multi-project scheduling, weighted earliness/tardiness, dynamic
scheduling

Abstract

This study addresses the Resource Constrained Multi Project Scheduling Problem with Weighted Earliness Tardiness Costs (RCMPSPWET). In multi-project environments, the project portfolio of a company does often change dramatically in time. In this dynamic context, the arrival of a new project requires quoting a due date while keeping the disruptions to the existing plans and schedules to a minimum. The suggested solution method is an adaptation of the well known shifting bottleneck (SB) heuristic in the job shop literature. Initially, a base schedule is obtained by relaxing all resource capacities and solving the resulting model as a linear program (LP). The SB heuristic then resolves the resource conflicts present in the optimal solution of this resource relaxation iteratively by solving a set of single-resource weighted earliness tardiness scheduling subproblems with precedence constraints. The unit earliness and tardiness costs in the subproblems are estimated by drawing upon tools from LP sensitivity analysis recently proposed by Bülbül and Kaminsky [1] for a general job shop scheduling problem. The subproblems in the SB heuristic are a generalization of the NP-hard single machine weighted earliness tardiness problem, and a neighborhood search based algorithm is applied to these for the efficiency of the overall SB algorithm. The solution of a subproblem introduces new precedence relationships based on the concept of resource flows. These new precedence constraints are incorporated into the LP mentioned above and ensure that the capacity of the resource under consideration is observed. These steps are repeated until all resource conflicts are removed. The order in which the resource conflicts are resolved is a major determinant of the final solution quality, and therefore, a systematic tree search strategy is implemented for resolving the resource conflicts in different orders. A local search algorithm for the original problem is also adopted to benchmark the results.

BELİRLİ KAYNAK KISITLI ÇOKLU PROJE ORTAMINDA
AĞIRLIKLANDIRILMIŞ ERKENLİK GEÇLİK PROBLEMİ İÇİN DARBOĞAZ
ÖTELEME TEMELLİ ÇÖZÜM YÖNTEMİ YAKLAŞIMI

Mehmet Berke Pamay

Endüstri Mühendisliği, Yüksek Lisans Tezi, 2011

Tez Danışmanları:

Yar. Doç. Kerem Bülbül

Prof. Gündüz Ulusoy

Anahtar Kelimeler: çoklu proje çizelgeleme, erkenlik geçlik, devingen çizelgeleme

Özet

Çoklu proje ortamlarında firmaların proje havuzlarında zamana bağlı olarak ciddi değişimler olmaktadır. Bu devingen yapı içerisinde, yeni bir projenin havuza dahil olması, mevcut çizelge üzerindeki projelere en az etkiyi yapacak şekilde yeni gelen proje için bir tamamlanma zamanı belirlenmesini zorunlu kılar. Tanımlanan bu problem için geliştirilmiş olan çözüm yöntemi işlik çizelgeleme teknik yazınında sıklıkla kullanılan darboğaz öteleme çözüm yaklaşımının proje çizelgeleme ortamına uyarlanmış halidir. Çözüm yaklaşımın kapsamında, tüm kaynak kısıtlarının gevşetilmesi sonucu oluşan doğrusal model çözümlere öncül bir çizelge elde edilir. Sonraki adımlarda mevcut çizelgedeki kaynak aşımaları her kaynak tipi için çözümlenen erkenlik geçlik yan problemlerinin sonuçlarına bağlı olarak ortadan kaldırılır. Yan problem çözümünde kullanılan erkenlik geçlik katsayıları doğrusal model üzerine uygulanan duyarlılık analizi sonucu elde edilmektedir [1]. Tanıtılan yan problemin çözümü için yöre tarama tabanlı sezgisel bir yöntem geliştirilmiştir. Her yan problem çözümü ana probleme eklenen öncüllük kısıtları yardımıyla, sonraki adımlarda ele alınan kaynak tipi için kapasite aşımı olmasını önler. Bu şekilde modelin doğrusal yapısı korunurken, diğer kaynak tipleri için duyarlılık analizinin uygulanabilmesi sağlanır. Bu süreç tüm kaynak tipleri etkin hale gelinceye kadar tekrarlanmaktadır. Son olarak, kaynak tiplerinin etkinleştirilme sırasının çözüm yaklaşımının performansı için önemli bir etken olduğundan, ağaç yapısında tarama ve tekrar çizelgeleme yaklaşımları çözüm yöntemie dahil edilmiştir. Son olarak, sezgisel yerel tarama yöntemi çözüm niceliğinin sınanması amacıyla geliştirilmiştir.

Table of Contents

Abstract	vi
Özet	vii
1 INTRODUCTION AND MOTIVATION	1
1.1 Contributions	7
1.2 Outline	8
2 LITERATURE SURVEY AND PROBLEM DESCRIPTION	9
2.1 Literature Survey	9
2.1.1 Resource Constrained Project Scheduling Problem	9
2.1.2 Dynamic Resource Constrained Project Scheduling Problem	9
2.1.3 Resource Constrained Project Scheduling Problem with Weighted Earliness/Tardiness Costs	10
2.1.4 Machine Scheduling	13
2.2 Problem Environment	14
2.2.1 Resources	14
2.2.2 Network Structure	15
2.3 Problem Formulation	15
2.3.1 Sets and Indices	16
2.3.2 Parameters	17
2.3.3 Decision Variables	18
2.3.4 Mathematical Model	18
3 SOLUTION APPROACH	20
3.1 Resource Unconstrained Problem	23
3.2 Sensitivity Analysis	24
3.3 Single Resource Subproblem	26
3.3.1 A Heuristic Approach for the Single Resource Subproblem	28
3.4 Constraint Propagation Algorithm	28
3.5 Tree Search	35
3.6 Rescheduling	38
4 AN ITERATED LOCAL SEARCH APPROACH FOR RCPSP- WET	41
4.1 Activity List Presentation	42

4.2	Initial Population Generation	43
4.3	List-Positional Neighborhood Search	43
4.4	Optimal Timing Neighborhood Search	44
4.5	LP-Based Optimal Timing	45
4.6	Parameter Finetuning	45
5	COMPUTATIONAL STUDY	47
5.1	Experimental Data	47
5.1.1	Project Pool Generation	48
5.1.2	Total Number of Activities	49
5.1.3	Project Combinations	50
5.1.4	Due Date Generation	50
5.1.5	Due Date Distribution	51
5.1.6	Due Date Tightness	52
5.1.7	Resource Conditions	53
5.1.8	Completion Time Factor	54
5.1.9	Instance Naming Conventions	55
5.2	Results	55
6	CONCLUSION AND FUTURE WORK	61
Appendix		
A	LP CONSTRAINT GENERATION ALGORITHM	67
B	SETTINGS FOR DATA SET PROJECT POOL GENERATION	69
C	RESULTS	71

List of Figures

1.1	Decision making process in project management as presented in [2]	2
2.1	Resource usage profile examples as presented in [3]	14
3.1	Flow chart of the solution approach.	22
3.2	Comparison of the exact solution and the heuristic approach for the single resource subproblem.	29
3.3	Activity-on-node network of the numerical example.	31
3.4	Gantt chart for the resource unconstrained solution.	32
3.5	Activity-on-node network with resource arcs of resource type 1.	33
3.6	Gantt chart for the solution with resource arcs of resource type 1.	34
3.7	Activity-on-node network with resource arcs of resource types 1 and 2.	34
3.8	Gantt chart for the solution with all resources active.	35
3.9	Gantt chart for the optimal solution.	36
3.10	Tree search example.	37
4.1	Flow of the local search heuristic.	41
4.2	Parameter finetuning results.	46
5.1	Schedule of distributed due date windows.	52
5.2	Schedule of clustered due date windows.	53
5.3	Instance naming conventions.	56

List of Tables

3.1	Numerical example data for resource flow networks.	31
3.2	Solution of the resource unconstrained problem.	32
3.3	Solution for the network with resource arcs of resource type 1.	33
3.4	Solution for all resources active network.	34
3.5	Optimal solution of the problem.	35
4.1	Parameter selection settings	45
5.1	Dataset parameter settings.	48
5.2	Summary of results.	57
5.3	Summary of results.	58
5.4	Effect of due date tightness on solution quality.	59
5.5	Effects of due date distribution on solution quality.	60
A.1	Parameters used in the algorithm.	67
B.1	Settings for project pool generation.	69
B.2	Instances	70
C.1	Results for instances with 20, 30, and 40 activities.	72
C.2	Results for instances with 20, 30, and 40 activities-cont.	73
C.3	Results for instances with 20, 30, and 40 activities-cont.	74
C.4	Results for instances with 20, 30, and 40 activities-cont.	75
C.5	Results for instances with 20, 30, and 40 activities-cont.	76
C.6	Results for instances with 50, 100, 150, and 200 activities.	77
C.7	Results for instances with 50, 100, 150, and 200 activities-cont.	78
C.8	Results for instances with 50, 100, 150, and 200 activities-cont.	79
C.9	Results for instances with 50, 100, 150, and 200 activities-cont.	80
C.10	Results for instances with 50, 100, 150, and 200 activities-cont.	81
C.11	Results for instances with 50, 100, 150, and 200 activities-cont.	82
C.12	Results for instances with 50, 100, 150, and 200 activities-cont.	83
C.13	Results for instances with 50, 100, 150, and 200 activities-cont.	84
C.14	Results for instances with 50, 100, 150, and 200 activities-cont.	85
C.15	Results for instances with 50, 100, 150, and 200 activities-cont.	86

CHAPTER 1

INTRODUCTION AND MOTIVATION

Building a skyscraper in a metropolitan, preparing a term assignment at school or organizing a concert for rising interest in global climate change involve various tasks to be completed in a systematic order to reach the final target. All these tasks have to be accomplished while considering limitations of resources such as cash, skilled workers, concert hall etc. as well as predefined precedence relations between individual steps. This statement perfectly matches with the definition made by Kurtulus and Davis [4] for a resource constrained project scheduling problem. Project management approach can be applied to any of these processes to improve efficiency as a decision tool. Moreover, Browning et al. [5] state that such a wide range of applications make projects a common structure for organizing works.

Demeulemeester et al. [2] emphasize that decisions in project management can be classified under three categories: strategic, tactical and operational. Strategic decisions focus on the financing of projects, resource allocation and investment strategies, whereas project portfolio evaluation, capacity planning and due date quotation belong to tactical decisions. Operational decisions, on the other hand, involve actions to be taken to generate a schedule. Allocation of resources and timetabling of activities are considered at the operational level and play an essential role for efficient realizations of projects. Figure1.1 presents different decision levels of project management.

Competitive business dynamics force companies to manage multiple projects simultaneously. Internal company activities like maintenance or R&D, external activities performed for customers in order to create a profit like in construction or

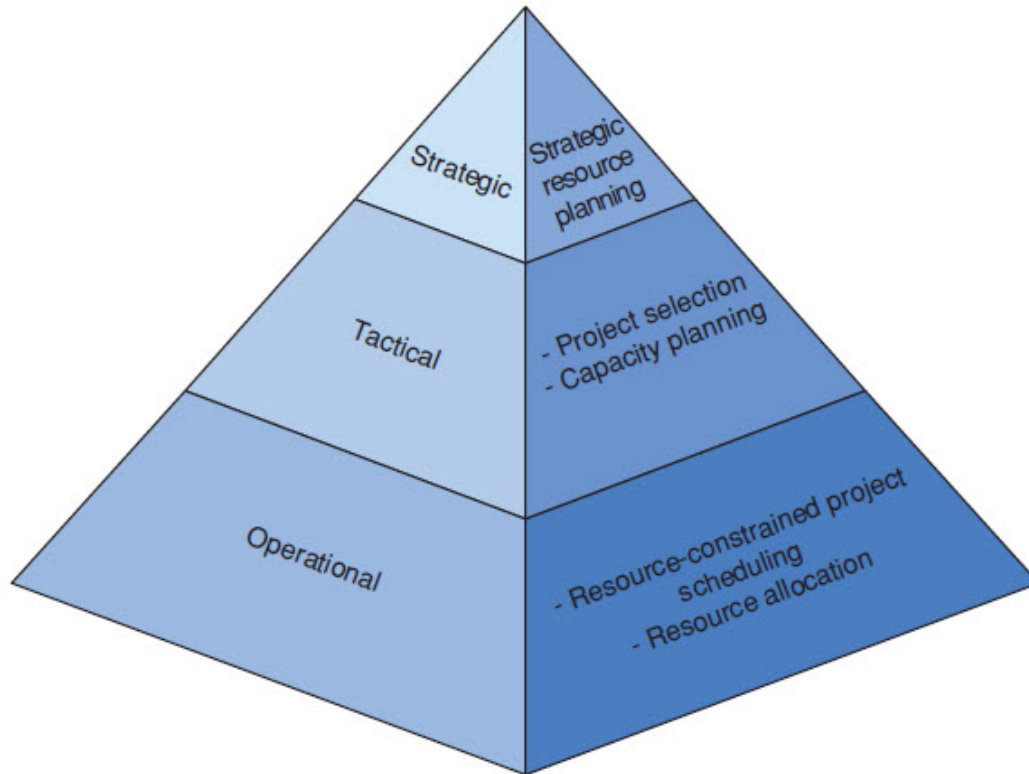


Figure 1.1: Decision making process in project management as presented in [2]

software development industries involve various examples of multi-project management applications. Payne [6], reports that up to 90% of the value of all projects occur in the multi-project context. Typically, multiple projects share common resource pools and the capacities of these resources are not sufficient to satisfy the demand of all project activities at the same time. Therefore, managers have to take critical decisions including portfolio selection, resource allocation and scheduling of multiple projects to remain competitive in the market. The Resource Constrained Multi-Project Scheduling Problem (*RCMPSP*), which focuses on scheduling of multiple projects while using available resource profiles and satisfying the precedence constraints to optimize the desired objective value, draws attention in the literature as well as in real life applications.

Various mathematical models are developed to support managers while generating optimal schedules with respect to selected performance measures. Most of these models concentrate on static characteristics, where schedules are based on the data

available before the solution procedure and the effects of unexpected events in the multi-project environment such as disruptions in projects, arrival of new projects and modifications in resource availability are not considered. Herbots et. al. [7] points out that static approaches are less realistic and a revision of the existing schedule might be required especially when dealing with external projects. The main reason behind the dynamic nature of external projects lies in the complex network of business relations between companies. Cooperation with other organizations and subcontractor companies is a very common strategy in multi-project management. Companies take over responsibility at certain steps of different project networks and foreseeing the total project load in the future becomes almost impossible. As a consequence, the project portfolio of companies changes dramatically with time. Therefore, models dealing with dynamic multi-project environments become more critical to provide realistic decision instruments.

Although selecting dynamic models is a critical step for realistic implementations in *RCMPSP*, additional model characteristics have to be determined to clarify the problem structure. First, deterministic and stochastic approaches differ in solution procedures as well as in decision parameters. The arrival times of new projects and the activity durations can be represented with random variables within a stochastic formulation. As a common approach in OR literature, the solutions obtained in stochastic models cover mostly optimization of expected objective values. At this point, the lack of data to fit random variables will be an obstacle on the way to model the problem with a stochastic approach. Moreover, the expected values are only realized in the long run. If the business environment is changing too rapidly, this would not be appropriate for our purposes. Another option might involve using dynamic scenarios modeled in a deterministic way where solutions are obtained for each modification in the problem data. Here, each modification corresponds to a new scenario or an unexpected event. To clarify, the solution process starts with an initial project portfolio and a baseline schedule. At a certain point of time a new project arrives and the existing schedule is revised. This case represents a single turn in this dynamic environment. Multiple disruptions or events, i.e. the arrival of

multiple projects at different points in time, can be simulated and the sensitivity of performance measures can be compared within a deterministic approach. In other words, specific solutions are offered for each modification. With this capability, deterministic dynamic models develop case based action plans for managers and provide required information about how to deal with an unexpected event at each turn. Therefore, the model presented in this paper is based on deterministic problem data with the main focus on creating effective decision tools for dynamic multi-project environments.

Selecting the appropriate performance measure is essential to reflect the reality. There are different performance measures in the literature with their own strengths. The project completion time is a quite popular performance measure showing the effective usage of resources as well as the responsiveness of a company. From the point of view of each individual company in a supply chain, minimizing the makespan of projects is a useful objective to utilize the common resources and narrow the timespan of the network. However, dynamic decision processes involve progressive schedule generation steps. Therefore, starting times of activities as well as resource allocation decisions in the schedule can change dramatically while minimizing the makespan for modified data sets. Most of the companies cannot handle these deviations, since additional costs might be involved for allocating a resource to another activity. Moreover, minimizing the makespan for the project network of each individual company at the same time is impossible. Therefore, focusing on deviations in different schedules can absorb the negative effects of dynamic events. At this point, one should keep in mind that the comparison between schedules for a newly arriving project cannot be performed, since no baseline schedule for this project exists. A due date has to be quoted in order to calculate the deviation from a promised deadline, and not from an existing schedule for this project. Yang and Sum [8] state that a negotiation procedure between the project owner and the supplier company is mostly adapted in the decision process to handle this problem. The project owner wants to complete his own project as soon as possible and offers an increased payment for an earlier completion time to motivate the supplier. The supplier company,

on the other hand, tries to place the new project at the very end of his schedule to eliminate any late delivery costs for existing agreements. A common decision is made upon the trade off between the increase in the expected revenue of this new project and the associated costs for not delivering existing projects on time. The decision process can be extended if rejecting the offer, which presents the newly arrived project in this case, is an option for the supplier company.

To be more specific about the economic sanctions of late deliveries, selecting appropriate time restrictions for each project as well as for each activity and defining respective costs for violating these deadlines is indeed a common objective for such complex business networks. Moreover, loss of customer goodwill and reputation have a similar effect and have to be taken into account. Although, finishing a project earlier is not penalized within the content of such business agreements, the costs related to storage and opportunity costs have to be considered during scheduling process. As a result, punishing both earliness and tardiness, directly or indirectly, force the companies to schedule all the activities on time or as close as possible to their due dates. Additionally, quoting additional due dates for new projects to maximize the revenue makes the problem even more difficult. In general, our problem is denoted as Resource Constrained (Multi-) Project Scheduling Problem with Weighted Earliness Tardiness (*RC(M)PSPWET*) in the literature and defined as follows: In a single (multi-) project network with a certain number of available renewable resource types, a processing time, a due date, resource profiles and associated tardiness and earliness costs are assigned to each activity. The objective is to create a schedule without violating precedence and resource capacity constraints in which the weighted sum of earliness and tardiness values are minimized. In addition to this base model, a cost parameter for the completion time(s) of the new project(s), representing the loss in revenue offered by the project owner, is included. As a result, the objective of our modified model becomes minimizing the weighted sum of earliness/tardiness costs and the cost associated with the new project's completion time.

Within the context of this problem, the due dates and associated penalties are

important parameters defining the characteristics of an instance. An applicable due date selection procedure is to convert the planned completion times into due dates. In other words, a baseline schedule, which is accepted by the subcontractor as well as by the project owner, is generated, and associated costs are defined if the new schedule deviates from the baseline plan. This approach can be applied to our deterministic model easily, since each disruption, as explained earlier, provides a baseline schedule and can be converted into due dates for a new event occurrence. With this approach, the dynamic problem can be simulated for multiple disruptions. The variations in revenue and deviations in schedules can be observed for multiple project arrivals at different points in time. Another strategy might involve defining some critical progress levels and penalties only for certain milestones of projects. Defining milestones is actually equivalent to adjusting activity based cost parameters. Moreover, higher penalties for project completion times can be selected in order to reflect different priority levels from a managerial point of view. With this strategy, the problem can be simplified from an activity due date based to a project based level. The deviation of each activity is not considered and number of decision parameters are decreased.

For any of these options, the following step is balancing tardiness penalty values. An important factor for these penalties is the tightness of due dates. A project with strict due dates has a greater possibility of becoming tardy so the penalty values for a unit time will be lower than those under loose due dates where the subcontractor company has a wider timespan to complete the project on time. However, it should also be remembered that the total realized penalty for an activity does also depend on the tardiness/earliness value. For later stages of this research, the relationship between delivery times and due date quotation penalties per unit time has to be examined in detail. The cost parameters have to be determined in a way such that a trade-off between deviation from the baseline schedule and the due date of the new project exists. Otherwise, the problem can be solved trivially by scheduling the new project at the very early stages or at the very end of schedule depending on the relative weights of the cost values.

The effect of robustness of the baseline schedule on the solution procedure is also important. Considering the possibility of an unexpected event occurrence and leveling the resource capacities in order to eliminate negative effects of arriving project can extend the scope of *RCMPSPWET* problem. Setting on project due dates while keeping resource usage under a threshold level, can improve the objective value and the solution time. A supplier creating the baseline schedule for his current project portfolio and utilizing resources under the available capacity can use slack resource capacities to schedule the new project without causing deviations in the new schedule. However, this might cause an increase in the makespan of the baseline schedule. At this point, the effects of underutilization of resources and the robustness of the baseline schedule on the dynamic scheduling procedure has to be questioned. The sensitivity of the objective function with respect to different resource utilization levels shall be studied as well.

In this thesis, different solution methods are presented for *RCMPSPWET* problem. The dynamics of the problem is analyzed with respect to due date tightness, cost parameter definition strategies and resource usage level limitations. The main goal is to devise with solution methods that provide quick and near optimal solutions for this problem. The reason behind it lies on developing decision support tools for managers, who have to take these critical decisions for their project networks frequently to survive in competitive markets. At this point, quick solution methods can make rescheduling, time and cost feasible in comparison with repair heuristics, which incorporate myopic approaches in most of the cases.

1.1 Contributions

The primary purpose of the present study is to develop an effective linear programming (LP) based method for *RCMPSPWET*. The following list shows the contributions of this study:

- An LP-based approach is implemented to solve the original problem. The procedure is based on a resource decomposition extension of the well known

shifting bottleneck heuristic in the job shop literature.

- A local search heuristic is adopted to benchmark the results obtained with the LP-based method.
- A hybrid approach, combining both methods, is developed to improve the solution quality.
- A unique data set investigating the effects of due date tightness, due date distribution, completion time factor of newly arriving project and number of active resource types on the solution approach will be generated.

1.2 Outline

The thesis is organized as follows: In Chapter 2, the related work in the literature is presented and an integer programming formulation of *RCMPSPWET*, as an exact solution formulation, is introduced. Chapter 3 focuses on the LP-based solution procedure and provides detailed information about individual steps of the proposed approach. A heuristic approach for the original problem is presented in Chapter 4. Introduction of data sets and evaluation of results are made in Chapter 5. Finally, a conclusion including the comments about the study and possible extensions for a future work is in Chapter 6.

CHAPTER 2

LITERATURE SURVEY AND PROBLEM DESCRIPTION

2.1 Literature Survey

The related work in the project scheduling literature does not completely cover all the aspects mentioned in previous chapter. Therefore, the literature survey includes articles from different disciplines. In this section, the research papers in resource constrained project scheduling literature will be presented first. Then, a closer look will be taken to the weighted earliness/tardiness extension of the problem. The relevant machine scheduling literature will be also mentioned to give an insight about applicable solution strategies.

2.1.1 Resource Constrained Project Scheduling Problem

The *RCPSP* covers various models, solution algorithms and extensions studied in the literature for different classes of project scheduling problems. Therefore, it is quite difficult to mention every single extension. However, several survey papers have been published since 1990s to summarize recent developments in the *RCPSP* literature. Interested readers may refer to [9], [10], [11] and [12]. In this section, related articles to our study will be presented.

2.1.2 Dynamic Resource Constrained Project Scheduling Problem

As it is also mentioned in Chapter 1, the Dynamic Resource Constrained Project Scheduling Problem (*DRCPSP*) deals with stochastic as well as deterministic models depending on the modeling approach to be implemented. Herroelen and Leus [9]

classifies the related work on *DRCPSP* under four categories: Reactive scheduling, stochastic rescheduling, fuzzy project scheduling and proactive scheduling. Note that our approach is within the scope of first category. Therefore, related work in the literature will be mentioned only for reactive scheduling processes. However, readers may also refer to a very recent review paper about the stochastic project scheduling by Ashtiani et al. [13].

The models focusing on reactive scheduling try to model any unexpected event within a deterministic approach. The concept is based on the presence of a baseline schedule similar to our approach, and minimizing the effects of the unexpected event is the main objective. As stated earlier, two different options can be used to accomplish that. First, the baseline schedule can be repaired. Second, a full rescheduling process can be adopted to solve the problem. Artigues et al. [14] study the case, where a new activity has to be inserted in the baseline schedule. The objective is to minimize the maximum lateness in a multi-mode multi-project setting. The multi-project environment is transformed to a resource flow network setting and dominant insertion cuts are used to generate the new schedule. El Sakkout and Wallace [15], on the other hand, proposed a solution method for minimizing the weighted absolute difference between the starting time in the baseline schedule and in the modified schedule for each activity. The weighted absolute differences correspond to earliness/tardiness concepts with symmetric costs if the finishing times in the baseline schedule are converted into due dates. They propose a repair based heuristic approach to solve this problem.

2.1.3 Resource Constrained Project Scheduling Problem with Weighted Earliness/Tardiness Costs

The existing work on *RCPSPWET* addresses the single project version of our problem. To the best of our knowledge, no research has been conducted on a problem environment with multiple projects. Moreover, the concept of a baseline schedule is also not included in most of the studies. Neumann et al. [16] mentions an original schedule subject to change as a result of unexpected events. The limited work in

the literature includes some exact solution approaches as well as heuristic methods for the problem.

An exact solution procedure for the resource unconstrained version of the problem is suggested by Vanhoucke et al. [17]. The objective function is composed of the weighted sum of earliness/tardiness values. This approach is based on a recursive search algorithm and consists of two main steps. First, a schedule is generated by scheduling activities at their due dates or later while considering precedence relations only. As a result, no right shift in the schedule can decrease the objective value. Therefore, in the second step of the algorithm the set of activities are selected, for which a backward shift can decrease the objective value. This is mainly done by implementing a recursive search. Two different studies are built upon contributions of this article. First, Vanhoucke et al. [18] extend the model with resource capacity constraints. With the exact solution algorithm for the resource unconstrained version on hand, they developed a branch and bound algorithm. This systematic search method is based on solving the resource conflicts in a resource unconstrained solution. Precedence relations are added between activities in process during a resource conflict period. Each conflict corresponds to a new node in the search tree and feasible solutions are obtained if all conflicts are resolved. A further extension of the resource constrained model is mentioned in [19]. In this study, for each activity, various due date options are offered. Each option differs in tightness and unit cost value of the due date. That means, if an earlier due date is selected for an activity, the unit earliness and tardiness cost values are lower than those for a loose due date. The objective is to select an appropriate due date option for each activity and generate a schedule such that the weighted sum of earliness tardiness values is minimized. A double branch and bound algorithm is developed to solve this problem. First, the resource unconstrained model is solved with the convex due date cost profiles. These profiles are obtained by converting the combination of different due date cost functions for each activity into a convex envelope, which is the highest convex function fitting below the cost profile. Thereby, a single due date is selected for each activity. However, unit earliness or tardiness costs might change

according to the convex envelope profile. The solution is a lower bound for the actual due date profile and the first branch and bound is applied while considering the distance between the convex envelope and the original due date profile for each activity completion time. The optimal solution is obtained after applying a second branch and bound in order to cope with the resource conflicts as in [18].

One of the most recent studies is by Ballestin et al. [20]. The authors develop an iterated local search algorithm for the problem. A population of feasible solutions is generated and local search procedures are applied to improve the objective function value. One important thing to mention about this work is the representation of a schedule. Activity lists and a schedule generation scheme are used to generate corresponding schedules. The activities are scheduled iteratively with respect to a parameter called simulated due date, which is the completion time of an activity in a randomly generated precedence feasible but resource unconstrained schedule. Note that, simulated due dates are selected instead of the original due date values in the problem data in order to create diversity in the population. Four different local search procedures are then applied to existing schedules. At this stage, the activity lists are not changed; instead, schedules are modified in order to obtain improved solutions for a particular activity list in the population. To expand the search space, activity lists are perturbed. The sequence of activities in the list as well as the simulated due dates are updated according to five different perturbation procedures.

Another list-based heuristic approach is proposed by Nanobe et al. [21]. This work covers a variety of project scheduling problems with convex cost functions including the weighted earliness/tardiness problem. The idea behind the solution procedure relies on keeping event lists to obtain schedules. Each activity consists of a start- and an end-event, where positions of events in a list define priority relations. Each list can be represented on an event-on-node network presentation, and the dual problem can be solved as a minimum cost network flow problem. It should be noted that event lists have to be resource and precedence feasible. This is done by controlling the total resource demand of activities which are allowed to be processed

simultaneously. If this is the case, the list is modified and made feasible by changing the positions of events. A neighborhood is defined by moving events in the list backward or forward. Furthermore, an iterated local search is applied to the solution with the best objective value.

2.1.4 Machine Scheduling

Weighted tardiness as well as weighted earliness/tardiness problems have been studied thoroughly in machine scheduling literature. Both of these objectives are modeled in single machine, job shop and flow shop scheduling environments. It should be noted that the single machine problem is also considered in many job shop scheduling solution approaches as a subproblem. This is mainly done by decomposing the main problem into single machine subproblems. In such decomposition algorithms, the first problem to be solved is the sequencing problem. The processing sequence at each machine is obtained by solving the sequencing problem. Once a sequence is determined, the operation starting times are then obtained by solving the second subproblem, referred to as the optimal timing problem, which is solved for the given operation sequence on each machine.

Avci and Storer [22], Brandimarte et al. [23] study job shop scheduling problems. The first article studies the weighted earliness/tardiness problem, whereas Brandimarte et al. study the weighted tardiness problem. They propose neighborhood search algorithms by modeling the optimal timing subproblem as a maximum cost network flow problem. The neighborhood definition is based on changing directions of disjunctive arcs in the network, which corresponds to changing the sequence of operations processed on the same machine. Both studies define specific properties of neighborhood structures for finding improving moves. Thereby, the required number of iterations is decreased and the solution quality is improved.

2.2 Problem Environment

2.2.1 Resources

In [10], four different types of resources are emphasized: renewable, nonrenewable, doubly constrained and partially renewable resources. Nonrenewable resources have limited capacities over the project horizon without any restrictions for the consumption within each period. Renewable resources, on the other hand, are limited on a time period basis. That means, there is a certain capacity which is renewed at the beginning of each time period. Doubly constrained resources are limited on a time period as well as on a project horizon basis. Finally, partially renewable resources are similar to renewable resources, but they are limited within a specific range of the time horizon instead of each period of time. For instance, the available capacity of a partially renewable resource type cannot exceed 5 units in 3 consecutive periods.

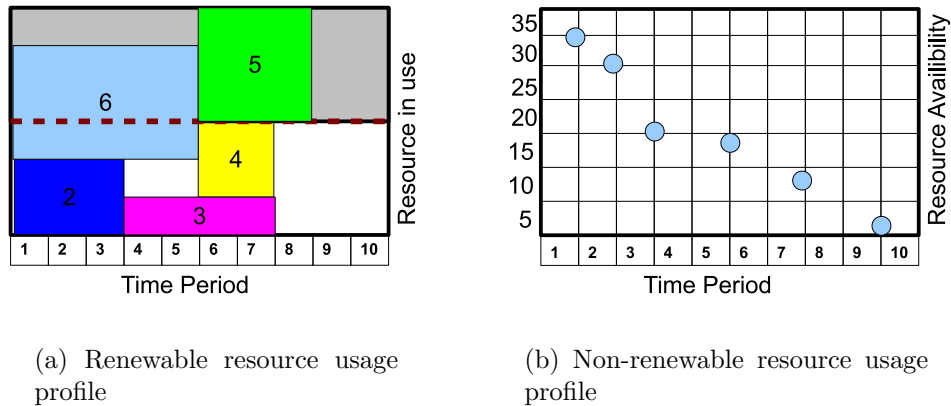


Figure 2.1: Resource usage profile examples as presented in [3]

Most of the work in the literature use renewable and nonrenewable resources in their models. The remaining two types are rarely implemented to reflect specific resource characteristics. In our study, we will only consider renewable resources. The reason why nonrenewable resources are not included is that our problem is modeled in a single mode setting, which will be further explained in the problem formulation section.

2.2.2 Network Structure

The *RCPSP* scheduling process includes determining starting times of activities with respect to available resource capacities and the temporal relations between activities. Precedence relations, which are used to define these temporal relations, are mostly imposed due to the technological reasons. Four different types of precedence relations are mentioned in [24] : start to start (S-S), start to finish (S-F), finish to start (F-S) and finish to finish (F-F). All these precedence relation types correspond to specific time lags between starting times or finishing times of two activities. A minimum time lag F-F precedence relation, for instance, dictates that a predefined amount of time has to elapse between the finishing time of an activity and the finishing time of its successor.

The network structure also varies with respect to the selected presentation scheme. Activity-on-node and activity-on-arc network structures are commonly used for project scheduling problems. In activity-on-node networks, each node corresponds to an activity and directed arcs stand for precedence relations. In activity-on-arc networks, on the other hand, activities are defined by event nodes. For each activity, there is a start and an end node. The completion of the activity is presented by an arc between these two nodes. Precedence relations are also presented by arcs between events depending on the type of the precedence relation. In our problem environment, activity-on-node networks with F-S zero time lag type precedence relations will be used to describe the problem data. Note that all the instances in this study are starting with dummy start projects and ending with dummy finish projects.

2.3 Problem Formulation

The model presented in this study covers a single mode *RCMSP*. As it is proven by Blazewicz et al. [25], *RCPSP* is NP-Hard. Although our problem is mainly defined as a *RCMSPWET* and extends *RCPSP* with a nonregular objective function [18], it does not only include the sum of weighted earliness/tardiness costs. The considered

scenario is based on an existing project portfolio with a baseline schedule on hand, where a new project arrives on top of a baseline schedule. The objective is first to quote a due date for the new project and also generate a new schedule while minimizing the total weighted deviation of finishing times. Therefore, the problem becomes a weighted earliness-tardiness problem with an additional cost component for the quoted due date of the new project.

The problem is a single mode problem. That means, each activity can only be executed according to a single recipe where the resource requirements and activity durations are fixed. It is important to mention that we consider only renewable resources. Since there is only a single way to execute each activity, it does not make any sense to check whether the problem is feasible with respect to nonrenewable resources. It is straightforward to see that the nonrenewable resource feasibility is sustained when the sum of requirements is no more than the available capacity of a particular nonrenewable resource type. Moreover, once an activity is started, it has to be completed without interruption, in other words, preemption is not allowed.

The work done by Pritsker et al. [26], is one of the earliest mixed integer (*MIP*) models developed for the *RCPSP*. An extension for the multi-mode version of the problem is mentioned in Sperenza et al. [27]. These two models differ in their definitions of 0-1 decision variables as well in the structure of their objective functions. Although our problem setting is not similar to any of these formulations, the model presented below is based on the models studied in these works.

2.3.1 Sets and Indices

T = set of time periods

I = set of all projects in the baseline schedule

I^* = set of all projects including the arriving project

$h = |I|$

$h + 1$ = index of the arriving project

J_i = set of all activities of project i

P_i = set of precedence relations between all activities $j \in J_i$ of project i

R = set of all renewable resources

2.3.2 Parameters

W_{rt} = available amount of renewable resource r in period t

ES_{ij} = earliest start time of activity j of project i

LS_{ij} = latest start time of activity j of project i

d_{ij} = due date of activity j of project i

p_{ij} = processing time of activity j of project j

w_{ijr} = renewable resource requirement of activity j of project i of type r per unit time

e_{ij} = earliness penalty of activity j of project i per unit time

t_{ij} = lateness penalty of activity j of project i per unit time

K = due date penalty for the arriving project per unit time

The parameters presented above are required to define an instance. For each activity, p_{ij} , w_{ijr} values define the single execution mode. However, there are additional parameters for activities depending on their status in the problem. First, for activities in the baseline schedule, a due date and corresponding earliness and tardiness values must be present in the problem data. A completion time factor standing for the cost associated with the completion time of the arriving project has to be also included. Note that e_{ij} , t_{ij} , d_{ij} and K are not part of the original problem data in the experimental study. They have to be determined according to the prescheduling procedure, which will be detailed in Chapter 5. Finally, the available capacities of renewable resources are required. Note that the earliest and latest start times of activities can be calculated for a given time horizon T by using the conventional forward and backward pass algorithms of the critical path method. A proper time horizon can be selected by summing up the processing times of all activities and adding it to the maximum due date value. Although, a smaller T value can improve the performance of an exact solution algorithm and decrease the

number of variables in the model, it is not possible to narrow the theoretical time horizon, since our objective function is nonregular. Therefore, an increase in the makespan of the whole multi-project network might indeed decrease the objective function value.

2.3.3 Decision Variables

The decision variables listed below have to be determined within the solution procedure in order to obtain a schedule. The 0-1 decision variable, x_{ijt} , is defined for each activity in the multi-project network including the dummy start and finish activities. For the activities in the baseline schedule; a finishing time, earliness and tardiness values have to be determined. For the arriving project, a due date has to be quoted. As stated in Chapter 1, this is done by selecting the finishing time of the dummy finish activity of the arriving project as the quoted due date.

$$x_{ijt} = \begin{cases} 1 & \text{if activity } j \text{ of project } i \text{ starts at time period } t, \\ 0 & \text{otherwise.} \end{cases}$$

f_{ij} = finishing time of activity j of project i

d_{h+1} = due date of arriving project

E_{ij} = earliness of activity j of project i

T_{ij} = tardiness of activity j of project i

2.3.4 Mathematical Model

Model RCMPSPWET :

$$\min \sum_{i \in I} \sum_{j \in J_i} (e_{ij} \cdot E_{ij} + t_{ij} \cdot T_{ij}) + K \cdot d_{h+1} \quad (2.1)$$

$$f_{il} - f_{ik} \geq p_{il} \quad \forall i \in I^*, \forall (k, l) \in P_i \quad (2.2)$$

$$f_{ij} = \sum_{t=ES_{ij}}^{LS_{ij}} x_{ijt} \cdot t + p_{ij} \quad \forall i \in I^*, \forall j \in J_i \quad (2.3)$$

$$E_{ij} \geq d_{ij} - f_{ij} \quad \forall i \in I, \forall j \in J_i \quad (2.4)$$

$$T_{ij} \geq f_{ij} - d_{ij} \quad \forall i \in I, \forall j \in J_i \quad (2.5)$$

$$d_{h+1} \geq f_{h+1j} \quad \forall j \in J_{h+1} \quad (2.6)$$

$$\sum_{i \in I^*} \sum_{j \in J_i} \sum_{\theta=\max\{ES_{ij}, t-p_{ij}+1\}}^t x_{ij\theta} \cdot w_{ijr} \leq W_{rt} \quad \forall r \in R, \forall t \in T \quad (2.7)$$

$$\sum_{t=LS_{ij}}^{ES_{ij}} x_{ijt} = 1 \quad \forall i \in I^*, \forall j \in J_i \quad (2.8)$$

$$x_{ijt} \in \{0, 1\} \quad \forall i \in I^*, \forall j \in J_i, \forall t \in ES_{ij}, \dots, LS_{ij} \quad (2.9)$$

$$d_{h+1}, f_{h+1j} \geq 0 \quad \forall j \in J_{h+1} \quad (2.10)$$

$$E_{ij}, T_{ij}, f_{ij} \geq 0 \quad \forall i \in I, \forall j \in J_i \quad (2.11)$$

The model introduced above is a minimization problem, where the objective function consists of weighted sum of earliness and tardiness values of the activities in the baseline schedule and the completion time cost of the new project. Constraint (2.2) defines the precedence relationships for each project. The finishing times of the activities are given in constraint (2.3). Constraints (2.4) and (2.5) determine the earliness and tardiness values, respectively. The quoted due date value, in other words the completion time of the arriving project is set by constraint (2.6). The renewable resource capacities for each time period are given in constraint (2.7). Finally, constraint (2.8) is included to ensure that each activity is assigned once and constraints (2.9), (2.10), and (2.11) define the domains of the decision variables.

CHAPTER 3

SOLUTION APPROACH

Our solution approach is mainly an extension of the machine-based decomposition heuristic studied by Bülbül and Kaminsky [1]. Therefore, a closer look will be taken at the original article, before presenting the general flow of our algorithm.

A shifting bottleneck heuristic for a large class of job shop scheduling problems is presented by the authors. We note that, although the suggested approach is applicable to a wide range of problems, these problems are limited to those for which their optimal timing subproblem can be modeled as an LP. The objective function studied in this article consists of two components: an intermediate holding cost component and a cost component which is a function of completion times. The objective function involves weighted costs of earliness and tardiness values and intermediate holding costs. Additionally, a weighted total makespan term is also a part of the objective function. As stated before, the solution strategy can be defined as a modification of the shifting bottleneck heuristic, which was originally developed by Adams et al. [28]. As a decomposition based heuristic, the subproblems of the bottleneck single machines in the job shop are solved iteratively. The disjunctive graph representation is used to give a better understanding of the solution approach. Two different types of arcs are adopted in order to define the relations between the operations. The first type corresponds to the precedence relations between the operations of the jobs. These arcs are fixed and given in the problem data. The second type of arcs are called disjunctive arcs. They represent the sequencing decisions made between the operations to be processed on the same machine. By fixing these arcs, machine capacity violations are prevented. Thereby, by solving

the single machine subproblems iteratively, disjunctive arcs of machines are added to the network. The key steps of the algorithm are given below:

- At the initial state, no decisions are made about the sequences of jobs on the machines in the job shop, i.e. no disjunctive arcs in the network are fixed. Therefore, machine capacity constraints are ignored.
- At each iteration, single machine subproblems are solved for each unscheduled machine. The bottleneck machine is determined according to the objective values of the single machine subproblems. The disjunctive arcs for the bottleneck machine are added to the network. Thereby, the machine capacity constraints of this particular machine is no longer violated. Rescheduling of already scheduled machines is also applied as an option to improve the schedule.
- These steps are repeated until all the machines are scheduled. In other words, the disjunctive arcs for each machine are fixed, such that no capacity constraints can be violated.

The single machine subproblem is not solved according to the original problem data, but instead a sensitivity analysis is applied to reflect the changes after adding corresponding disjunctive arcs for already scheduled machines. That means, the costs for moving the operations according to the current state of the schedule are estimated. This mainly done by determining the change in the objective function value, if the finishing time of an operation is increased or decreased by a representative time unit. An approximation for this estimation is obtained by implementing a single iteration of the dual simplex method. Note that a partial tree search is also included to process different sequences of scheduling the machines. Since our approach is an extension, no further detail will be provided about the job shop study.

After introducing the inspiring work in the literature, the implementation for the *RCMPSPWET* will be presented. The general flow of the algorithm is given in Figure 3.1. Note that the tree search procedure is not illustrated in this figure. First of all, conflicts of different resource types are solved. This step corresponds

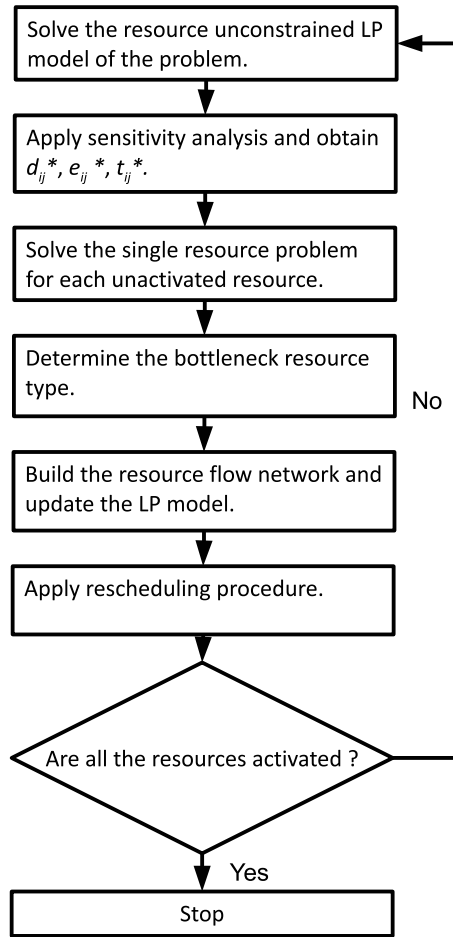


Figure 3.1: Flow chart of the solution approach.

to fixing disjunctive arcs of the bottleneck machines in the original paper. At the beginning of the solution method, all resource constraints are relaxed. Sensitivity analysis is applied to estimate the costs for moving activities earlier or later by a representative time unit in the resource unconstrained schedule. A single resource subproblem, which will be introduced in the upcoming sections, is solved for each resource type. According to the objective function values of the single resource subproblems, the bottleneck resource type is determined. For this particular resource type, a constraint propagation algorithm is used to avoid further resource conflicts. As it is stated in the previous paragraph, the solution approach can be applied to problems whose optimal timing subproblem can be modeled as an LP. Therefore, the constraint propagation algorithm is used to generate resource capacity preserving LP models for activated resource types. Once the associated constraints are added

to the model, the subproblem data is updated and new bottleneck resource type is determined. This process is repeated until all the resource types in the problem are activated. Just like in the original solution procedure, tree search and rescheduling strategies are adapted to our approach.

3.1 Resource Unconstrained Problem

As the first step of the solution procedure, the model of the resource unconstrained problem is presented below. As stated in chapter 2, the original model is an MIP model. The resource capacities are controlled by introducing 0-1 decision variable x_{ijt} , which is set to 1, if activity j of project i starts at time t , and is set to 0, otherwise. However, the resource unconstrained version can be modeled as an LP.

Resource Unconstrained LP Model :

$$\min \sum_{i \in I} \sum_{j \in J_i} (e_{ij} \cdot E_{ij} + t_{ij} \cdot T_{ij}) + K \cdot d_{h+1} \quad (3.1)$$

$$f_{lj} - f_{kj} \geq p_{lj} \quad \forall i \in I^*, \forall (k, l) \in P_i \quad (3.2)$$

$$E_{ij} \geq d_{ij} - f_{ij} \quad \forall i \in I, \forall j \in J_i \quad (3.3)$$

$$T_{ij} \geq f_{ij} - d_{ij} \quad \forall i \in I, \forall j \in J_i \quad (3.4)$$

$$d_{h+1} \geq f_{h+1j} \quad \forall j \in J_{h+1} \quad (3.5)$$

$$d_{h+1}, f_{h+1j} \geq 0 \quad \forall j \in J_{h+1} \quad (3.6)$$

$$E_{ij}, T_{ij}, f_{ij} \geq 0 \quad \forall i \in I, \forall j \in J_i \quad (3.7)$$

The objective function(3.1) consists of the weighted sum of earliness/tardiness values of the activities in the baseline schedule and an additional term stands for the completion time of the newly arriving project, just like in the original model. Constraint (3.2) introduces the precedence relations between the activities. Earliness and tardiness values are determined in constraints (3.3) and (3.4), respectively. Completion time of the arriving project is determined in constraint (3.5). Finally,

the lower bound values for the decision variables are set in constraints (3.6) and (3.7).

3.2 Sensitivity Analysis

Manipulating the original data is an important part of our solution approach. This new data has to reflect the current situation of the problem, in other words, the effects of already activated resource types. On the other hand, it should provide the appropriate costs for making any changes in this schedule. This is mainly done by performing a sensitivity analysis in our study.

At a certain step of the algorithm, once the resource unconstrained LP problem is solved, an objective function value for the original problem is obtained. The solution is probably infeasible, since not all resource capacity constraints are active. The main purpose of the sensitivity analysis is to determine the cost of shifting each activity later or earlier in time by a representative time unit of δ . Thereby, the modified earliness and tardiness costs for each activity can be estimated by assuming that the completion time in the solution is a due date. It should be noted that these modified values are obtained for all activities in the problem including the ones of the newly arriving project. As a result, the objective function of the subproblem, which will be introduced in the upcoming sections, consists of the weighted sum of modified earliness/tardiness values only.

There are two different approaches to estimate the modified cost values. Both of them are based on the LP solution obtained at the beginning of each iteration. In general, constraints (3.8) and (3.9) are added to the model in order to determine the change in the objective function value of the problem. The first constraint represents the case of shifting the activity by δ time units later and the second one stands for moving the activity δ time units earlier. Remember that these constraints are added to the model one at a time for each activity and, one of the aforementioned options involves adding the constraints for each activity, building the new model and solving it. This process has to be repeated two times for each activity. At each time, the estimated cost value is calculated by dividing the difference between the objective

function value of the LP at the beginning of iteration and the objective function value of the constraint added LP by δ . By doing this the modified LP is solved to optimality for each activity and the complete procedure of the sensitivity analysis is applied. However, considering the number of activities and the number of iterations in the solution procedure, this approach will be computationally costly.

$$f_{ij} \geq d_{ij}^* + \delta \quad (3.8)$$

or

$$f_{ij} \leq d_{ij}^* - \delta \quad (3.9)$$

An alternative way of performing the sensitivity analysis is using an approximation for modified cost values. This is mainly done by applying a single iteration of dual simplex method. Once the optimal solution for the LP model is obtained, adding one more constraint, either (3.8) or (3.9), makes the primal model infeasible. The dual model, on the other hand, is still feasible. Therefore, the dual simplex method can be applied to repair the infeasibility of the primal model. By performing one iteration of the dual simplex method, the actual earliness/tardiness costs can be approximated. It is important to mention that all these calculations are made by using the data available in the LP optimal solution at the beginning of each iteration. In general, earliness and tardiness values can be calculated according to the expressions given below:

$$e_{ij}^* = \frac{\bar{c}_t}{\bar{A}_{jt}} = - \max_{k \neq j | \bar{A}_{jk} > 0} \frac{\bar{c}_k}{-\bar{A}_{jk}} \quad (3.10)$$

and

$$t_{ij}^* = -\frac{\bar{c}_t}{\bar{A}_{jt}} = - \max_{k \neq j | \bar{A}_{jk} < 0} \frac{\bar{c}_k}{\bar{A}_{jk}} \quad (3.11)$$

As stated above, these expressions are obtained by a single implicit dual simplex iteration, where \bar{c}_t is the dual cost of the entering variable t and $\bar{\mathbf{A}} = (\bar{\mathbf{A}}_{\mathcal{B}} \bar{\mathbf{A}}_{\mathcal{N}}) = (\mathbf{A}_{\mathcal{B}}^{-1} \mathbf{A}_{\mathcal{B}} \mathbf{A}_{\mathcal{B}}^{-1} \mathbf{A}_{\mathcal{N}}) = (\mathbf{I} \bar{\mathbf{A}}_{\mathcal{N}})$, where $\mathbf{A}_{\mathcal{B}}$ is the basis matrix. Note that \bar{A}_{jk} is the

element of the matrix \bar{A} with the row index j and the column index k . We simply apply a ratio test to the LP model and calculate the change in the objective function explicitly. Readers may refer to [1] for more detailed information about the approach.

3.3 Single Resource Subproblem

The data obtained in the sensitivity analysis is used in the single resource subproblem. In general, for each activity including the activities of the newly arriving project, a modified due date value, which is equal to the completion time of the activity in the LP model and earliness/tardiness values are defined. As stated before, these values are the costs for moving an activity earlier or later in time. In the single resource subproblem, a schedule has to be generated while considering the capacity constraints of the candidate resource type only. Once a schedule is obtained, the network will be modified with the constraint propagation algorithm, which will be explained later, such that no resource infeasibilities of the activated resource occur, in the subsequent iterations.

One way of solving the single resource subproblem is developing the associated model and solving the problem with the exact solution method. This model is represented below:

Model for the Single Resource Subproblem for a resource type k :

$$\min \sum_{i \in I^*} \sum_{j \in J_i} (e_{ij}^* \cdot E_{ij} + t_{ij}^* \cdot T_{ij}) \quad (3.12)$$

$$(2.3), (2.8), (2.9) \text{ and } (2.11)$$

$$f_{il} - f_{ik} \geq p_{il} \quad \forall i \in I^*, \forall (k, l) \in P_i^* \quad (3.13)$$

$$E_{ij} \geq d_{ij}^* - f_{ij} \quad \forall i \in I^*, \forall j \in J_i \quad (3.14)$$

$$T_{ij} \geq f_{ij} - d_{ij}^* \quad \forall i \in I^*, \forall j \in J_i \quad (3.15)$$

$$\sum_{i \in I^*} \sum_{j \in J_i} \sum_{\theta = \max\{ES_{ij}, t - p_{ij} + 1\}}^t x_{ij\theta} \cdot w_{ijk} \leq W_{kt} \quad \forall t \in T \quad (3.16)$$

$$f_{h+1j}, E_{h+1j}, T_{h+1j} \geq 0 \quad \forall j \in J_{h+1} \quad (3.17)$$

The model is actually the original *RCPSPWET* model in a multi-project environment. That means, the objective function consists of the weighted sum of earliness and tardiness values only. Moreover, the model is a MIP model because of the 0-1 decision variable x_{ijt} , which is exactly the same variable presented in the model for the original problem in Chapter 2. Note that constraints (2.3), (2.8), (2.9) and (2.11) are taken from this model. Constraint (3.13) stands for precedence relations of projects in the baseline schedule and the arriving project. The original precedence set P_i for a project i is replaced by the set P_i^* , which includes the original precedence relations as well as the arcs added in the network for activated resource types by the constraint propagation algorithm. The earliness and tardiness values according to the modified due date d_{ij}^* are imposed in constraints (3.14) and (3.15). Constraint (3.16) is added to the model in order to control the resource capacity of the activated type k . Constraint (3.17) define the range of the decision variables.

Initial experiments conducted with very small instances consisting of 20 activities show that implementing the exact model for the single resource subproblem into the solution procedure is computationally expensive. Even for these small instances, the required CPU times are longer than the amount of time required for the exact solution method of the original problem. There are actually two main reasons behind that. The first reason is that the model for the single resource subproblem is a MIP model just like the one for the original problem. The number of constraints are decreased by selecting only one type of resource. However, additional constraints are also added by defining earliness and tardiness values for the newly arriving project. In general, the size of this model is slightly smaller in comparison with the original

model. The second disadvantage of the model is that the single resource subproblem has to be solved for several times in the solution procedure. For example, in an instance with five resource types for a given activation sequence of resources, the subproblem has to be solved for five times. It should also be noted that this number increases dramatically as tree search and rescheduling steps are included. Therefore, solving the single resource subproblem with the exact model is computationally prohibitive. An alternative approach is suggested in the upcoming subsection.

3.3.1 A Heuristic Approach for the Single Resource Subproblem

The heuristic approach proposed for the single resource subproblem is a local search algorithm. A generalization of the same method is developed for the original problem in order to benchmark the results. Therefore, the detailed structure and steps of the procedure are presented in Chapter 4 including the differences between implementations for the subproblem and the original problem.

The performance of the heuristic approach is initially tested by comparing the solution quality of the algorithm with that of the exact MIP model. An initial experiment is conducted with 10 instances in the data set with 20 activities each. For each solution of the single resource subproblem, the results obtained by both methods are compared. The results are shown in Figure 3.2. As it can be also seen in the figure, the heuristic approach performs well for a total of 1169 single resource subproblem results.

3.4 Constraint Propagation Algorithm

Once the single resource subproblem is solved at a particular step of the solution algorithm, the structure of the network has to be modified in order to reflect the schedule obtained in the subproblem and avoid resource infeasibilities of the activated resource type. In job shop scheduling, the original model is an LP model and the solution of the subproblem is introduced by adding additional precedence arcs to the network. Therefore, the structure of the model does not change. For the *RCPSP* extension, a similar approach has to be implemented in order to conserve the LP

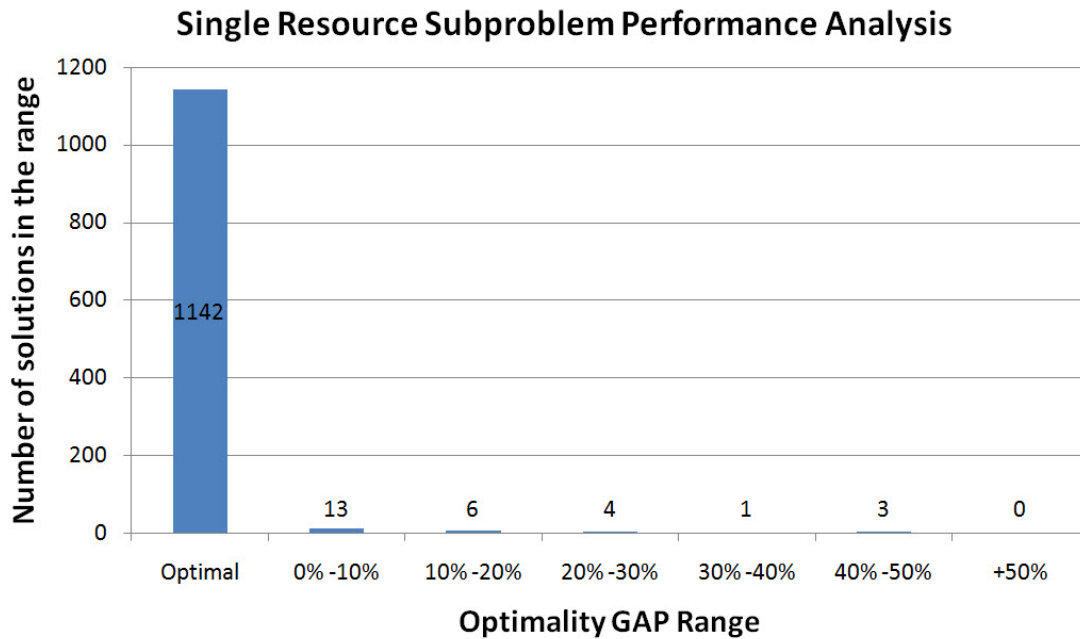


Figure 3.2: Comparison of the exact solution and the heuristic approach for the single resource subproblem.

characteristics of the resource unconstrained model while preserving feasibility of the already activated resource types. One way of doing this is adding additional precedence constraints to the model. There are several studies in the literature focusing on this property of the *RCPSP* networks. Artigues et al. [14] used resource flow arcs to represent a schedule. Vanhoucke et al. [17] added additional precedence arcs to the original problem for the activities in process during a resource conflict. Another study is performed by Policella et al. [29]. In this study, the main purpose is to robustify a solution by converting the associated schedule into a resource feasible resource unconstrained model. The partial order schedule as defined by the authors, is an activity network, such that any possible temporal solution is also a resource-consistent assignment. That means only precedence relations are used to attain resource feasibility. This is mainly done by defining flows for each unit of resource from the dummy start activity of the project to the dummy finish activity for a given schedule. Each activity can only transfer resource to activities, which start after its finishing time. If no precedence arc already exists in the network between two activities transferring a resource unit, an arc is added to the network.

In this study, we use a similar concept of partial order schedules. For a given solution of the single resource subproblem, we will convert this schedule into a network consisting of temporal arcs, such that no resource infeasibility can occur for this particular resource. In order to do that, an LP constraint propagation algorithm is proposed. The parameters used in the algorithm and the pseudocode of the algorithm are presented in Appendix A. The flow of the algorithm is as follows:

- First, each activity has a demand for the considered resource type equal to its resource requirement. Additionally, a source node with the available capacity equal to the total capacity of the resource and a sink node with the demand equal to the total capacity are introduced.
- All the activities in the schedule are sorted according to their starting times. The algorithm selects the activity with the minimum starting time and satisfies its demand by transferring the resource from already processed activities, for which the finishing time is less than or equal to the starting time of the activity in process. Note that, at the beginning of the algorithm only the source node is in the processed activity list.
- There are two options while satisfying the demand of an activity. The resource transfer can be done over existing precedence relations or by adding additional arcs to the network. In our algorithm, we first transfer resources to an activity from its predecessors in the processed activity list, that means from activities, which are connected to the activity in process with an arc. If the resource amount offered by those activities is not enough, resource is transferred from other activities in the processed list. At this point, the candidates are selected according to the maximum available resource capacity. Thereby, the number of additionally propagated arcs is minimized.
- If the resource demand is satisfied, the activity is added to the processed list with the available capacity equal to its requirement.
- This process is repeated until all the activities, including the dummy sink

node, are added to the processed activity list.

Although the LP constraint propagation algorithm is able to provide LP models without any resource conflicts, it should be noted that there are various possibilities while adding constraints to the network. Each resource transfer decision is made among various candidates in the processed activity list. Therefore, this approach may narrow the search space for the original problem. In general, adding resource arcs progressively might affect the optimal timing of activities in the final solution. In order to visualize these effects, a single project *RCPSPWET* numerical example is provided below.

Table 3.1: Numerical example data for resource flow networks.

Activity	p	d	ec	tc	$r1$	$r2$	Successors
1	0	0	2	0	0	0	2,3,6
2	3	4	2	2	1	2	5
3	4	5	2	2	3	1	4
4	2	8	3	3	2	2	7
5	3	10	4	1	3	0	7
6	5	5	3	2	1	3	7
7	0	9	6	4	0	0	

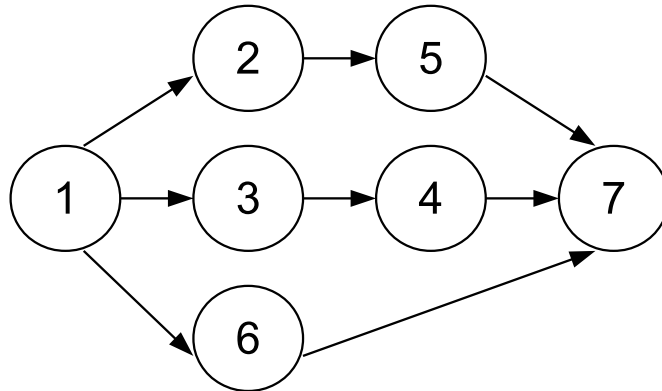


Figure 3.3: Activity-on-node network of the numerical example.

The problem data, including the processing times, due dates, earliness and tardiness costs, resource requirements, and successor activities are summarized in Table 3.1. The activity-on-node network presentation is also given in Figure 3.3. Our

approach will be similar to the original solution method in this report. That means, we will solve the resource unconstrained version of the problem first. Then associated resource flow arcs will be added to the network and the modified LP will be solved again. The same process will be repeated for the second resource type and the increase in the objective function value will be observed.

Table 3.2: Solution of the resource unconstrained problem.

Activity	1	2	3	4	5	6	7	OFV=4
f_j	0	4	5	8	9	5	9	
E_j	0	0	0	0	1	0	0	
T_j	0	0	0	0	0	0	0	

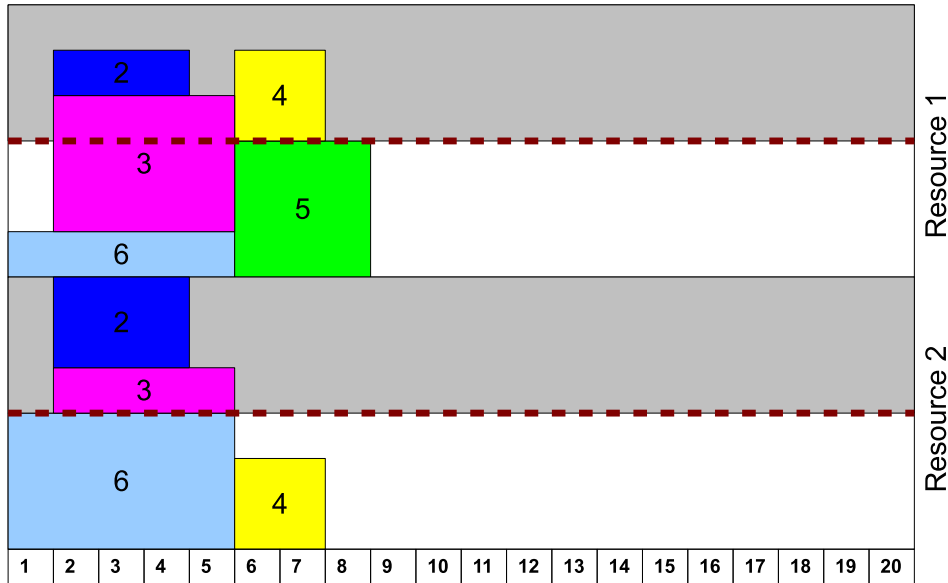


Figure 3.4: Gantt chart for the resource unconstrained solution.

The finishing times and the earliness/tardiness values for the resource unconstrained version of the problem are given in Table 3.2. Since there are no resource constraints in the problem, the only priority relations while scheduling the activities are the precedence relations. The Gantt chart of the solution is given in Figure 3.4. As it can be also seen on the Gantt chart, the resources capacity constraints are violated. Activities 2, 3 and 6 consume more than the available capacity of resource types 1 and 2, whereas activities 4 and 5 cause a resource conflict for resource type 1. In order to resolve these conflicts additional precedence arcs will be added. As

the first step, we will only consider the resource type 1. An arc is added between activities 3 and 6 to solve the first conflict and an arc between activities 4 and 5 will prevent capacity violations of those activities. The resulting network is given in Figure 3.5 with the additional arcs colored in red. Note that there are alternative arc combinations while processing the schedule. For example, it is also possible to add arcs from activity 3 to activities 2 and 6. The critical point is finding predecessor activities, which can provide enough resource for its successors.

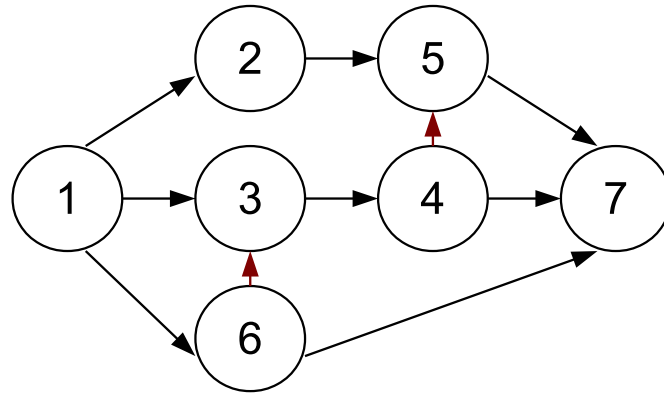


Figure 3.5: Activity-on-node network with resource arcs of resource type 1.

Table 3.3: Solution for the network with resource arcs of resource type 1.

Activity	1	2	3	4	5	6	7	OFV=41
f_j	0	4	9	11	14	5	14	
E_j	0	0	0	3	0	0	0	
T_j	0	0	4	0	4	0	5	

The modified LP model is solved again and the solution is presented in Table 3.3. The corresponding Gantt chart is given in Figure 3.6. As it can be seen, with the additional arcs, no conflicts occur for resource type 1. However, activities 2 and 6 cause a resource conflict for resource type 2. It should be also mentioned that new precedence relations cause an increase in the objective function value from 4 to 41. At this point, an arc from activity 6 to activity 2 is added to solve the resource conflict. The final structure of the network is given in Figure 3.7.

The final solution of the numerical example is provided in Table 3.4, and the

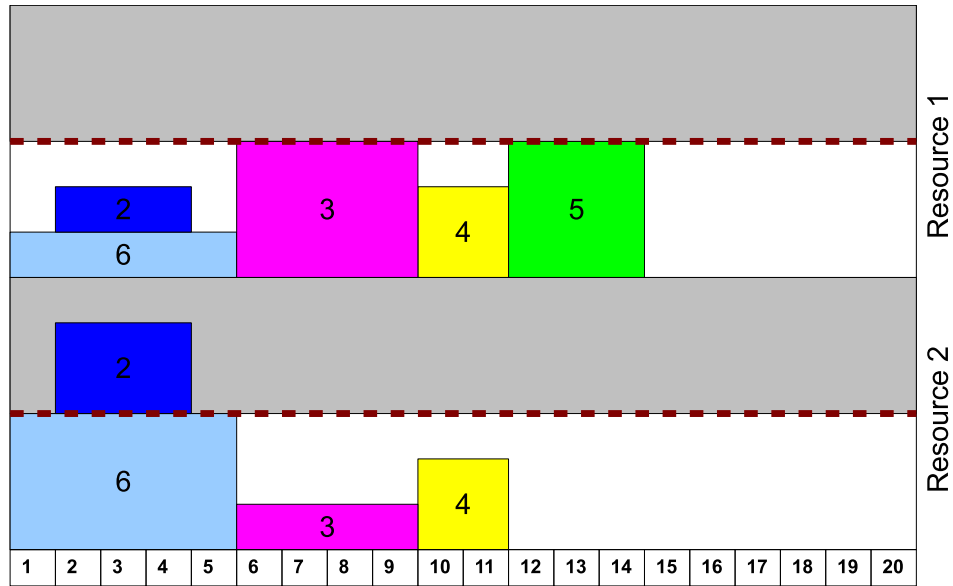


Figure 3.6: Gantt chart for the solution with resource arcs of resource type 1.

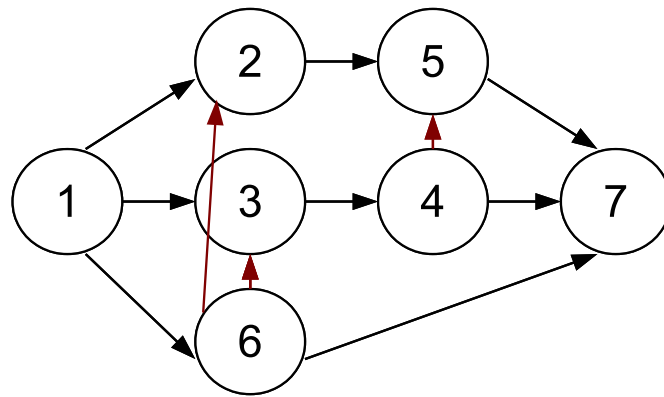


Figure 3.7: Activity-on-node network with resource arcs of resource types 1 and 2.

Table 3.4: Solution for all resources active network.

Activity	1	2	3	4	5	6	7	OFV=76
f_j	0	14	9	11	17	5	17	
E_j	0	0	0	0	0	0	0	
T_j	0	10	4	3	7	0	8	

Gantt chart in Figure 3.8. The objective function value is increased to 76. The optimal solution of the same problem is given in Table 3.5 and Figure 3.9. In this particular example, adding constraints to solve the resource conflicts makes it possible to solve the problem as an LP without any resource conflicts. However, the

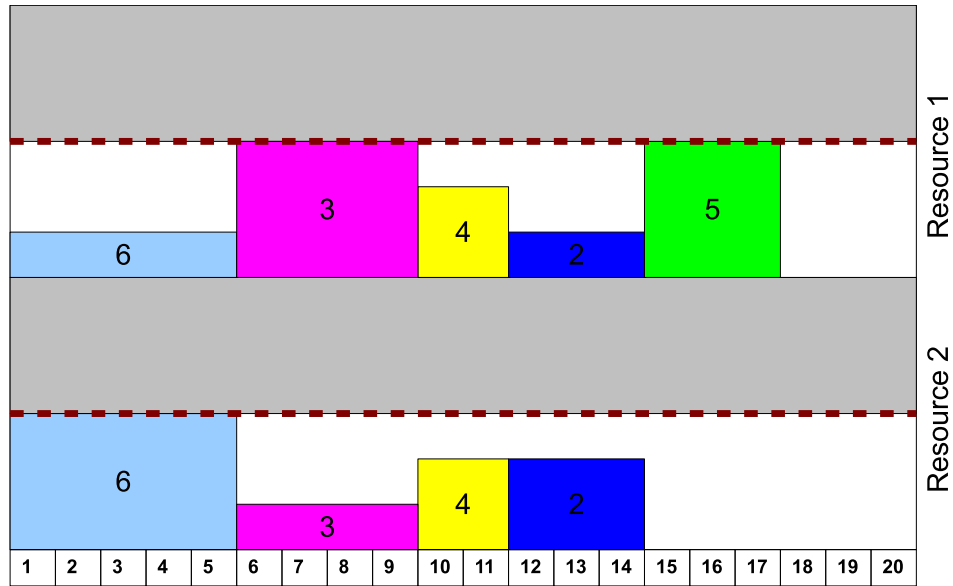


Figure 3.8: Gantt chart for the solution with all resources active.

objective function value is increased at the same time, since the solution space is limited by modifying the network structure.

Table 3.5: Optimal solution of the problem.

Activity	1	2	3	4	5	6	7	OFV=66
f_j	0	3	7	9	17	14	17	
E_j	0	1	0	0	0	0	0	
T_j	0	0	2	1	7	9	8	

3.5 Tree Search

In order obtain solutions for different resource activation sequences, a tree search procedure is implemented in the solution approach. An example tree for an instance with four resource types is given in Figure 3.10.

In general, the tree search starts with an LP model without any resource related constraints. As stated before, sensitivity analysis is then applied to obtain the required data for the single resource subproblems. At this point, the decision of the activated resource is made upon the subproblem objective function values of child nodes. As it can be seen on the first level of the tree, activation of resource types 1, 2, 3 and 4 is evaluated and the corresponding nodes are generated. In this particular

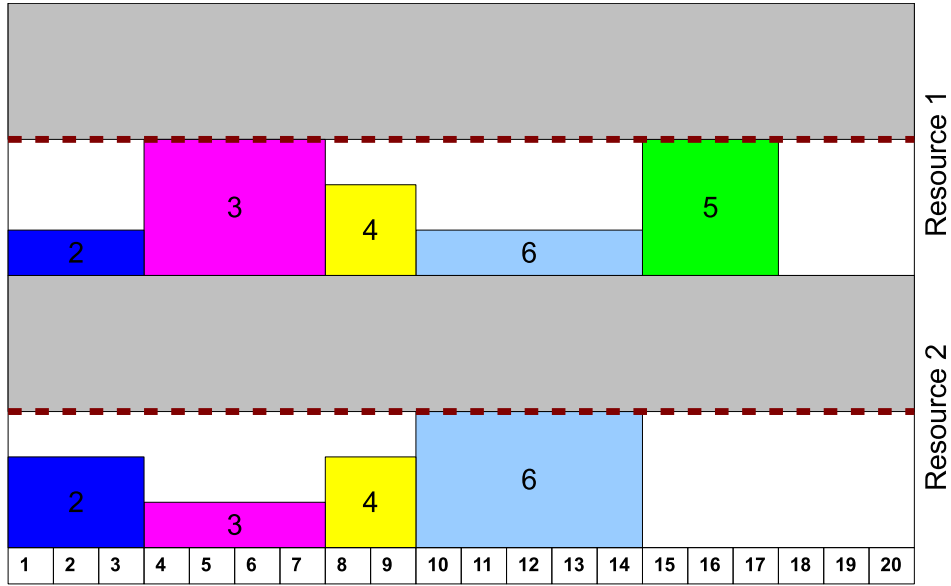


Figure 3.9: Gantt chart for the optimal solution.

example, the objective function value for the subproblem with the activated resource type 1 is greater than those of the other nodes. Therefore, this node is selected as the new node to be processed. Once the resource type 1 is selected, the LP constraint generation algorithm is activated and related arcs are added to the network. Since the LP model is modified, the sensitivity analysis is reapplied and modified due dates and costs are obtained. At the second level of the tree, 3 different child nodes are generated for activating resources 2, 3 and 4 after resource type 1. Subproblem objective function values are evaluated and the node $\{1, 3, -, -\}$ is selected. At the third level of the tree, there are actually 2 different child nodes, which yield to feasible solutions for the original problem. After the arcs for resource type 3 are added and the network is updated, the end nodes are separately generated by first solving the subproblem for resource type 2 and then for type 4. These steps are then repeated in the reverse order. These two child nodes at the end are $\{1, 3, 2, 4\}$ and $\{1, 3, 4, 2\}$. It should be noted before repeating the reverse order for the final two resource types, the corresponding arcs have to be removed and the initial state of parent node $\{1, 3, -, -\}$ has to be rebuilt. Once the end nodes are generated, backtracking is applied to process other sequences. As showed in this example, a possible backtracking node might be $\{1, 4, -, -\}$. Deactivated resource arcs have

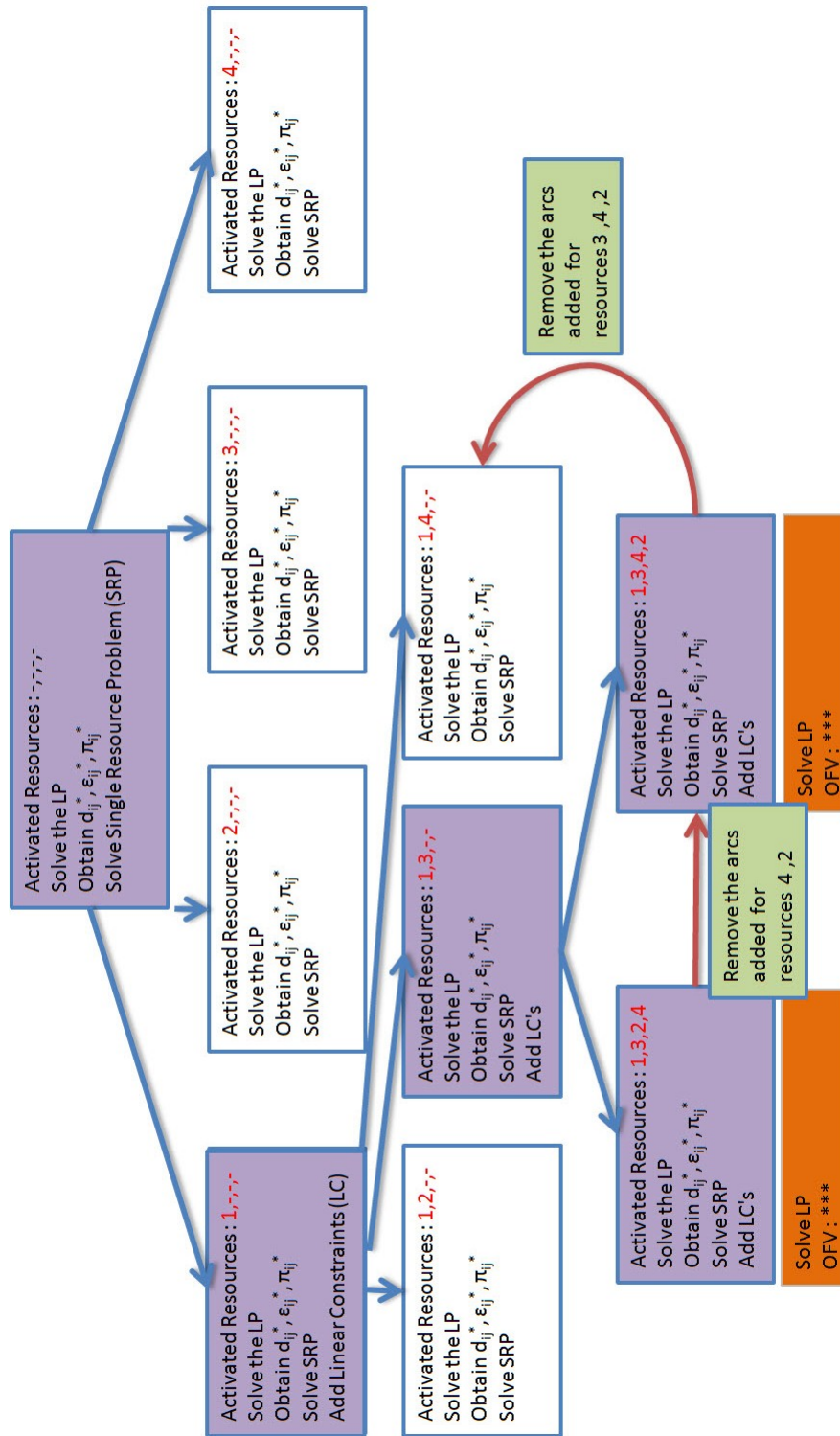


Figure 3.10: Tree search example.

to be removed from the network again. Therefore, in order to move from node $\{1, 3, 4, 2\}$ to node $\{1, 4, -, -\}$ the arcs added for resources 2, 4 and 3 have to be

deleted. At this node, the sensitivity analysis is then performed and new child nodes are generated accordingly.

The maximum number of resource types in an instance is in general less than the number of machines in the benchmark instances in the job shop scheduling literature. As it will be explained in the experimental study chapter, two different resource type settings are selected in our study: 2 and 5. For an instance with two different resource types, there are only two different resource activation sequences either $\{1, 2\}$ or $\{2, 1\}$. Therefore, the size of the tree is relatively small. For an instance with five resource types, on the other hand, the number of nodes in the tree is increasing but it does not reach the dramatic level in job shop scheduling instances. Therefore, no extra fathoming rule will be included into the tree search approach. However, in order to decrease the required computational effort, the maximum number of nodes generated at a certain level of the tree is limited to 3. That means, according to our example, at the first level of tree, among nodes $\{1, -, -, -\}$, $\{2, -, -, -\}$, $\{3, -, -, -\}$ and $\{4, -, -, -\}$ child nodes will be generated only for three nodes with the highest single resource subproblem objective function values. Apart from that, a rescheduling strategy is also implemented to improve the solution at each node in the tree, which will be explained in the next subsection.

3.6 Rescheduling

The sequence in which the resources are activated, is an important decision variable for the performance of the solution method. As stated in the previous subsection, tree search is one of the tools used to explore the solution space more thoroughly and obtain schedules for different resource activation sequences. Another tool which can be incorporated into our approach is the rescheduling step. In general, a classical rescheduling step in the SB heuristic in the job shop literature includes removing the arcs for one of the already scheduled machines and re-solving the subproblem of this particular machine while considering the modified state of the network. The solution of the subproblem yields the new arcs to be added to the network. Rescheduling can be repeated at various steps of the algorithm and also for a given

number of rescheduling cycles depending on the problem structure. Demirkol et al. [30] reported in their study that rescheduling can improve the performance of SB algorithms significantly. The rescheduling process in our solution approach focuses on arcs added to the LP for each resource type. That means, at a particular rescheduling step we will remove the arcs which are only used for the selected resource type. It should be noted that an arc can transfer resources for more than one resource type. At this point, these arcs have to be kept in the network in order to avoid resource conflicts. Moreover, rescheduling can be repeated for more than one cycle. The main steps of the rescheduling are summarized below:

- At a given node, the arcs corresponding to the first resource type in the sequence are removed and the LP model is solved.
- Sensitivity analysis is applied and the single resource subproblem is solved with the data obtained in the sensitivity analysis.
- The arcs are then added according to the solution of the single resource subproblem and the problem for the modified network is solved.
- These steps are repeated for the remaining active resource types in the node and the objective function values of the LP model are obtained for each resource type.
- The resource types are sorted according to the objective values of the LP model. At this point, a single cycle of rescheduling is completed and this sorted vector is the new sequence for the second cycle of rescheduling process.
- Rescheduling is repeated unless the predefined number of the rescheduling cycles is performed or an already processed rescheduling sequence is obtained.

Initial experiments show that the structure of the resource flow network affects the performance of the rescheduling process. The resource flow arcs are mostly used by more than one resource type. That means, each arc in the network transfers more than one resource type. These arcs might be present in the problem or they can be

added while activating resources at the earlier stages of the solution algorithm. As a result, it is observed that the arcs in the network are completely preserved although the arcs corresponding to a single resource type are deactivated. In other words, there is no arc in the network, which is only responsible for transferring the resource type to be rescheduled. Therefore, the solution of the initial LP as well as the single resource subproblem remains the same after deactivation. That means the vector at the end of the rescheduling cycle consists of a single objective value for all resource types, which are rescheduled. In such a case the rescheduling process is terminated in order to prevent further unnecessary CPU requirements.

CHAPTER 4

AN ITERATED LOCAL SEARCH APPROACH FOR RCPSPWET

Heuristic procedures are developed for RCPSPWET in single project environments as it is also mentioned in Chapter 2. List-based heuristics studied in [20] and [21] perform well by means of solution quality as well as computation times. Moreover, neighborhoods can easily be defined for schedules presented by lists and schedule generation procedures are quite simple and fast. Therefore, a population based local search procedure is suggested to solve the single resource subproblem and the original problem. The general flow of the solution algorithm is presented in Figure 4.1. Note that the detailed information about the structure of the heuristic developed for the single resource subproblem is also presented in this chapter.

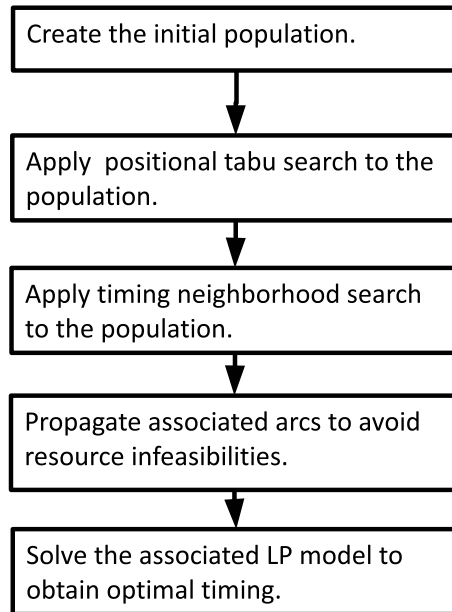


Figure 4.1: Flow of the local search heuristic.

The heuristic method starts by generating an initial population. Three different improving steps are applied to this initial population iteratively in order to improve the activity lists. These steps replace the sequencing and optimal timing procedures, which are commonly used in the machine scheduling literature for weighted earliness tardiness problems. First, a list-position based neighborhood search is performed. Thereby, the sequencing in each activity list is improved. An optimal timing based neighborhood search is then applied to move chains of activities earlier in time. Finally, as it is also used in the LP-based method, for all the resource types in an instance, associated arcs are added to the network and the corresponding LP model is solved.

4.1 Activity List Presentation

Activity-lists are used to present a schedule in the population. Each activity is assigned to a position in the list. In a feasible activity list, each activity has to be positioned after its predecessors and before its successors. If an activity-list is precedence feasible, a schedule is generated by scheduling each activity in the list to start at its locally optimal position. Locally optimal position means scheduling an activity to a position with the minimum cost with respect to its due date and according to the current state of the schedule. There is a small difference between the locally optimal scheduling scheme of the subproblem and the original problem. As it is stated before, each activity in the subproblem including activities of the newly arriving project has a modified due date and earliness/tardiness costs. Therefore, each activity in the subproblem can be scheduled locally optimal with respect to its due date. In the original problem, on the other hand, the activities of the newly arriving project have to be scheduled as early as possible since their contribution to the objective function is regular and depends on the completion time factor K . Therefore, in the original problem if an activity belongs to the newly arriving project, it is scheduled to the earliest resource feasible position.

4.2 Initial Population Generation

An initial population is generated to apply the neighborhood search procedures. Each member of the population is actually a precedence feasible activity list. In order to expand the search space, different dispatching rules and totally random activity lists are generated. At this point, there is again a small difference between the implementations of both problem types. For the single resource subproblem, earliest due date, most number of successors and minimum latest starting time rules are used to form an initial population. This is mainly done by selecting the activity with best value according to the dispatching rule among the precedence feasible candidates. Furthermore, roulette wheel versions of these dispatching rules are also used to increase the size of the population. To each candidate activity, a probability is assigned proportional to the fitness values and a random number is generated to select an activity. Finally, totally random selection among candidate activities is also adopted. For the original problem, on the other hand, the earliest due date rule cannot be used to generate activity-lists, because no due date value is defined for the activities of the arriving project.

4.3 List-Positional Neighborhood Search

Once the members of the initial population are generated, the first neighborhood search procedure starts. This process is applied to each member of the initial population separately and if an improvement is observed, the activity list is replaced and the search for better schedules continues with the new activity list. This is mainly done by calculating the contribution of each activity to the objective function in an activity-list. The neighborhood is defined as selecting the activity with the highest cost value and placing it to an earlier position while preserving precedence feasibility. Thereby, the activity can be scheduled at earlier stages of the schedule generation process, so there is a greater chance to schedule the activity closer to its due date. For a selected activity, there are several candidate positions for insertion in the list. These candidate positions lie in the range between the predecessor with the latest

position and the current position of the activity. In order to make a move, the activity is removed from its current position and inserted to each possible spot in the list. For each position, the objective function value is determined by using the locally optimal scheduling scheme. If the objective value can be improved, this change is applied to the activity-list. If it does not improve, the activity with the second highest cost contribution is selected and the procedure is repeated until a limitation number of nonimproving steps is reached. If no improvement can be observed after reaching this threshold level, the best nonimproving move is applied and the move is added to a tabu list to record forbidden moves. In general, the neighborhood search of an activity list continues until a predefined number of moves is reached or no improvement can be observed. It should be noted that no tabu list extension is applied for the single resource subproblem, since satisfactory solution performance is obtained with the basic neighborhood structure as reported in Chapter 3. Moreover, the cost contribution of the activities of the newly arriving project in the original problem is determined by selecting the cost contribution of the project, which is equal to the completion time of the project times K , for activities on the critical path of the project and 0 for the remaining ones.

4.4 Optimal Timing Neighborhood Search

The activity lists can also be improved by applying a timing process. The locally optimal scheduler places an activity to its own optimal position without taking other activities into account. Therefore, the total objective function value can be reduced by moving a single activity earlier or later in time. This can be done by modifying the due dates of activities, such that the locally optimal positions of the activities are changed for the same sequence. First, the activity chains in a schedule are determined. A chain of activities consists of activities which are connected with precedence arcs and the finishing time of the successor activity is equal to the starting time of its predecessor. Once the chains in the schedule are determined, the total cost contribution of each chain is calculated. The chain with the maximum cost is selected and the due date of the first activity in this chain is decreased by a

single time unit. This due date value is used while scheduling the activity locally optimally, but the objective function is still calculated with the original problem data. By decreasing the due date of an activity, other members of the chain can move earlier in time and the objective function value can be improved. If this is the case, the activity list is replaced. If it is not, then the chain with second highest cost value is selected and the procedure is repeated. The search is terminated if either the number of nonimproving steps or the number of total steps reaches the threshold value.

4.5 LP-Based Optimal Timing

As a final improvement step, some of the activity lists in the initial population, which is a decision parameter in the heuristic method, are converted to LP models by adding additional arcs in the network, such that no resource infeasibilities can occur. The LP model is solved in order to obtain an improved solution for the modified structure of the network.

4.6 Parameter Finetuning

In order to select the best-performing parameter setting, a finetuning procedure is applied. 20 different instances with 200 activities are tested. Six different parameters are adjusted: maximum number of steps for positional neighborhood search, maximum number of steps for timing neighborhood search, two different parameters for the maximum number of nonimproving steps of these neighborhoods, the size of the tabu list and the number of LP-based search repetitions. Different values selected for each setting and the results are presented in Table 4.1 and Figure 4.2.

Table 4.1: Parameter selection settings

Method	# of Positional Neighborhood Search Steps	# of Timing Neighborhood Search Steps	# of Nonimproving Steps for Positional Neighborhood	# of Nonimproving Steps for Timing Neighborhood	# of LP-Based Search Steps	Size of Tabu List
Setting 1	20	30	10	20	5	5
Setting 2	50	50	20	40	5	5
Setting 3	100	100	30	70	10	5
Setting 4	200	200	100	150	10	5

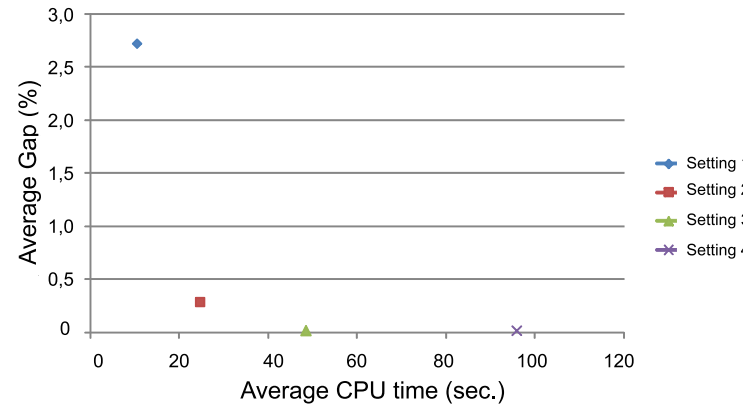


Figure 4.2: Parameter finetuning results.

Figure 4.2 shows the average GAP between the best solution and the solution found by each setting and the average CPU time for each setting. As it can be seen clearly, setting 3 performs well by means of solution quality and CPU times. Therefore, setting 3 is selected for the single resource subproblem and for the original problem solution procedures.

CHAPTER 5

COMPUTATIONAL STUDY

All the solution approaches have been coded in Visual C#. IBM ILOG CPLEX Optimization Studio ver. 12.1 is selected as the optimization software for solving the LP models. A data set of 800 unique instances are generated to test the performance of the suggested methods. The experiments were conducted on an HP Compaq dx 7400 Microtower with a 2.33 GHz Intel Core 2 Quad CPU Q8200 processor and 3.46 GB of RAM.

5.1 Experimental Data

As stated before, related work in the literature focuses on the single project version of *RCPSPWET*. Moreover, the existing benchmark instances in these works do not always investigate the effects of different problem parameters on the performance of the proposed solution approaches. Therefore, a new data set is generated. Each unique instance represents a new scenario corresponding to a critical decision which has to be taken by a project manager as stated in the previous chapters. That means, each instance of the problem set consists of a group of projects present in a baseline schedule with activity based due dates, earliness, and tardiness values. A newly arriving project is also included with a completion time factor K . The parameter settings for the whole data set are given in the table below. The idea behind adopting each of these parameters will be detailed in the upcoming subsections.

Table 5.1: Dataset parameter settings.

Total # of Activities	20, 40, 50, 100, 150, or 200
Due Date Distribution	Clustered or Distributed
Due Data Tightness	Tight or Loose
# of Resource Types	2 or 5
Completion Time Factor	High or Low

5.1.1 Project Pool Generation

Before focusing on the details of the parameters for generating data sets, a closer look will be taken at the project pool generation method. Our problem is a multi-project scheduling problem. Therefore, each instance in the data set consists of a group of projects. For this reason, a project pool is generated first, which will be later used to create multi-project instances. Various project generation software are present in the literature. ProGen is developed by Kolisch [31] for *RCPSP* and for the multi-mode extension. ProGen/max developed by Schwindt [32] is an upgraded version of ProGen for minimal and maximal time lag extensions of generalized precedence relations. Both pieces of software are popular in the literature and used by many researchers. A more recent project generator, called RanGen, is developed by Vanhoucke et al. [33]. Although no one can claim that any of these generators is definitely superior against others, RanGen enables selecting predefined complexity measures for generated networks, which is important for differentiating instances. Apart from that, the other adjustable parameters are similar in all software. As a result, RanGen is selected as the project generation software in our study.

Four parameters have to be adjusted in RanGen to obtain different project networks. The first setting is the order strength (*OS*), which is defined as the number of precedence relations including the transitive ones but not including the arcs connecting the dummy start and end activity divided by the max number of precedence relations ($n(n-1)/2$), where n denotes the number of non-dummy activities in the network [34]. As stated before, RanGen is able to generate unique networks with the prespecified *OS* values. Three different *OS* values (0.25, 0.50 and 0.75) are selected. For each project, 5 types of renewable resources are also defined. Two

different resource usage related parameters are included, which are introduced in equations (5.1) and (5.2). The first resource related measure is the resource density (RU). This parameter is used to define the minimum number of resource types used by an activity in the network. This parameter is preferred against another resource related measure referred to as the resource factor (RF) introduced by Alvarez et al. [35], because RF might yield networks in which some of the activities are not using any resources at all. Another resource measure, the resource-constrainedness (RC), is defined as the ratio between the available capacity of a resource type and the average usage of activities (\bar{w}_r) for this particular resource. The RU and RC values are selected as 4 and 5, 0.25 and 0.75, respectively. Last but not least, the number of the activities in a project is an input data in the software. To reach the required number of activities for each *RCMPSPWET* instance, projects with 5, 10 ,20 and 30 activities are generated.

$$RU_i = \sum_{r=1}^R \begin{cases} 1 & \text{if } w_{ir} > 0 , \\ 0 & \text{otherwise.} \end{cases} \quad (5.1)$$

$$RC_i = \frac{\bar{w}_r}{W_{rt}}; \quad (5.2)$$

All parameter settings are summarized in Appendix B. Each activity number category, except the one with 5 activities, consists of 50 different projects. The reason for a total of 32 projects with 5 activities is that the generator is not able to generate 50 unique networks with the specified *OS* values due to the small number nodes in the network.

5.1.2 Total Number of Activities

The total number of activities in an instance is a quite important measure to define the size as well as the difficulty of the problem. As it can be seen in Table 5.1 the activity numbers are selected within a range of 20 to 200 activities. It should be noted that this number does not include the dummy start and finish projects of each instance and their corresponding activities as well as the dummy start and finish

activities of each project. Note that we solve instances with up to 200 activities while the maximum number of activities considered in the literature is 100([20], [18], [16]).

5.1.3 Project Combinations

For each setting of the total number of activities, different scenario schemes are generated. That means, different combinations of projects are suggested to reach the total number of activities in an instance. For example, in order to generate a 30-activity instance, a combination of three projects with 10 activities each, is selected as one of the combinations. In this scenario, for two of these three projects, due dates, earliness and tardiness costs are generated and the third project is defined as the newly arriving one, so a completion time factor is determined. Another combination is selected as a project portfolio consisting of 6 projects with 5 activities each. Just like in the previous combination, one of these projects is the arriving one. For each setting of the total number of activities, up to three different combinations are selected. These combinations differ in the total number of projects in an instance. For each combination, five different master instances (*MI*) are generated. These master instances provide the information about which projects in the pool are added to the project portfolio. This is accomplished by selecting projects from the pool with the desired number of activities randomly. Master instances are then used to create unique instances by adding the data about due dates, earliness and tardiness costs and the completion time factors depending on the values of remaining parameters. All the project combination schemes are given in Appendix B.

5.1.4 Due Date Generation

As stated before, the way the due dates are generated in this study is based on a real baseline schedule. In other words, all projects in an instance, except for the new arrival, have an associated existing schedule constructed by a scheduling routine described next. In [20], [18] and [16], on the other hand, the data sets are generated while considering the critical paths and earliest start time values of the activities in

the network.

The method used to schedule the activities is quite important for the effective utilization of the resources. Therefore, makespan minimization is the objective of the pre-scheduling process. In other words, it will be appropriate to schedule the activities as early as possible. There are many heuristic approaches in the literature developed for makespan minimization. We decided to use a scheduling scheme with an effective dispatching rule in order to generate schedules with good objective values within reasonable computation times. In his review paper about the performance of different dispatching rules for makespan minimization, Kolisch [36] states that the Latest Starting Time (*LST*) rule shows the best performance. Therefore, the *LST* rule is applied to existing projects in order to generate the baseline schedule. The rule simply selects the activity with the minimum latest starting time among the ones whose predecessors are already scheduled. The latest starting time values are calculated by using a backward pass heuristic, as stated in the previous chapters of this report.

5.1.5 Due Date Distribution

The distribution of the due dates in the time horizon is important for the flexibility while scheduling. That means, if the due dates of a project are spread throughout the entire planning horizon, activities can be moved forward or backward in time while scheduling the arriving project. A clustered due date distribution, on the other hand, forces the projects to stay close to their active time intervals in the baseline schedule. The difference between these two settings are visualized in Figures 5.1 and 5.2. In the first case, all projects are active in most of the schedule timeline. The second case presents a schedule, where the total makespan is divided into intervals in which limited number of projects are active.

In order to obtain schedules with these two different characteristics, the basic schedule generation scheme with the *LST* rule is modified. For a distributed due date generation method, the progress levels of the projects are controlled while scheduling activities iteratively. That means at a certain step of scheduling, the

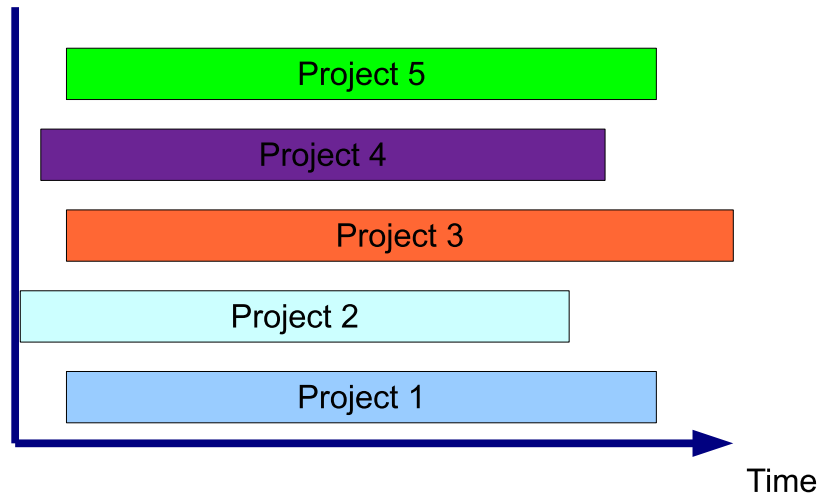


Figure 5.1: Schedule of distributed due date windows.

activity with the lowest LST value is selected among the activities of the project with the minimum progress level. With this approach, the projects are kept active along the entire baseline schedule timeline. The clustered due date distribution is obtained, by selecting a project randomly and scheduling the activities of this particular project first according to LST rule. The process continues by selecting another unscheduled project randomly and scheduling the activities of this project. Thus, the selected project is scheduled as early as possible after it is started. In order to observe the effects of this parameter setting, distributed and clustered due dates are only applied to project combinations with a higher number of total projects. If this is not the case, only distributed due dates are generated. The details are provided in Appendix B.

5.1.6 Due Date Tightness

An additional due date factor is implemented to adjust the tightness of the due dates. Tight due dates values are closer to the starting time of the schedule and offer less flexibility for meeting the due date. Loose due dates, on the other hand, allow the activities to move in time without effecting successor activities or causing any extra cost. In other words, there is a higher possibility to meet loose due

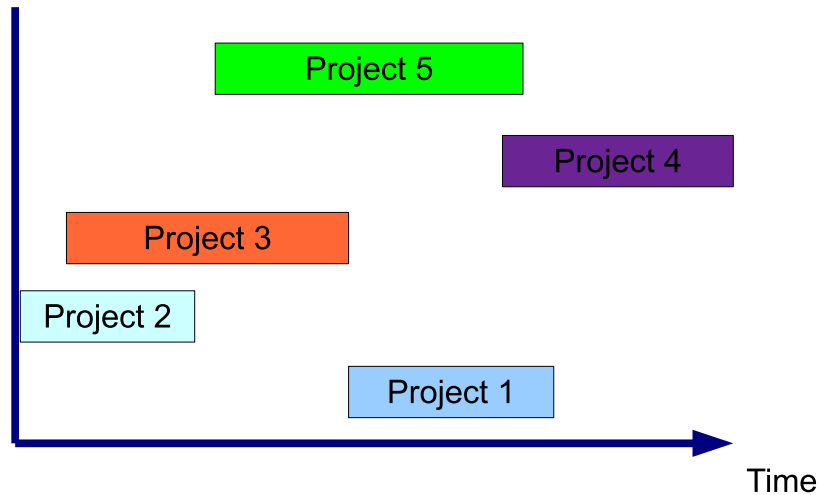


Figure 5.2: Schedule of clustered due date windows.

dates than the tight ones. It should be noted that the due date tightness in the related articles in the literature is adjusted by manipulating the due date value with a tightness factor. Our approach, on the other hand is built upon a different approach. In order to reflect these tightness and looseness aspects, we change the available resource capacities in the baseline schedule. In this way, we increase the total makespan of the baseline schedule and offer an additional resource capacity to schedule the newly arriving project. Thereby, it is expected that the number activities scheduled on time will increase and for the same master instance a lower objective value can be obtained. A level of 80% is selected to adjust the capacities for the loose due date setting and 100% resource availability in the baseline schedule corresponds to the case of tight due dates. For each unique instance loose and tight due dates settings are present in the data set.

5.1.7 Resource Conditions

As it is mentioned in the previous chapters, the number of active resources is important for the performance of our solution method for two reasons. First, our tree search process aims to compare the objective values for different release sequences of resource types. Therefore, the number of nodes in the tree increases dramatically

as the number of resource types increases. Secondly, it is straightforward to state that the number of resource flow constraints in the model increases as the number of resource types increases. In our data set, we selected two different resource type settings. We created instances with either two or five resource types. Actually, all the instances are created with five resource types, but if the instance has two active resource types, the last three resource types are not processed.

5.1.8 Completion Time Factor

Finally, the completion time factor value is changed for different instances in the dataset. As one of the contributions of our study, the effects of the completion time factor K will be studied. As stated before, a tradeoff between the earliness/tardiness costs and the completion time related cost must exist in order to generate a reasonable problem setting. Otherwise, scheduling the newly arriving project in the beginning or at the end of the schedule, depending on dominant cost values, might yield good solutions in most of the instances. Therefore, we implemented another pre-scheduling step, just like that in the due date generation process, to obtain completion time factors. For two due date distribution settings, different approaches are used. For the distributed due date setting, we generate a new schedule with the newly arriving project by using the *LST* rule and calculated the earliness/tardiness costs for the projects in the baseline schedule. For this value, we obtained the completion time factor by dividing the total earliness/tardiness cost by the completion time of newly arriving project. For the clustered due date distribution, we used the sequence of projects selected while generating due dates. In this sequence vector, we inserted the newly arriving project to each possible position and scheduled projects accordingly again with the *LST* rule. Hereby, we obtain different cost values for the same instance depending on the position of the new project in the sequence. Therefore, we take the average value of these cost values and completion times. The completion time factor is then calculated by taking the ratio of these two average values. Thereby, we generated completion factors specific to the problem data instead of selecting the same factor for all instances.

These completion time factor values are modified in order to obtain different parameter settings. They are multiplied by two different constants. These constants are selected as 0.5 and 1. These values represent that the contribution of the newly arriving project in the original problem to the objective function value is expected to be the half or as much as the contribution of existing projects, respectively. As a result, two different completion factor settings, a high (with the constant value 1) and a low value (with the constant value 0.5) are determined for each instance.

5.1.9 Instance Naming Conventions

Depending on different parameters introduced above a seven digit coding scheme is used to define an instance. In general, each instance is presented in this form: $A1_2.3.4_5.6_7$. The first position stands for the total number of projects in the instance. Digit 2 represents the combination scheme number for a given total number of activities. Digit 3 is the master instance number. Digit 4 is set to 0 for the distributed due date settings and to 1 for diverse due date settings. A 0 in digit 5 stands for tight and a 1 is for loose due date values. An instance with 2 active resource types is presented with a 2 in the sixth digit and 5 is used for an instance with 5 resource types. Finally, the last digit is equal to the completion time factor calculated for each unique instance.

An example is provided to make the naming procedure clear in fig 5.3.

5.2 Results

Experimental analysis is performed with four different solution methods: exact solution procedure (MIP), LP-based method (LP), local search heuristic (LS) and a hybrid method (H). The hybrid method is developed in a way such that the end nodes in the search tree of the LP-based algorithm are used as the initial population of the LS procedure. Each node is converted to an activity list according to the starting times in the schedule. For each of those activity lists, two unique members of the population are generated by selecting the due date values as the original due date value or the finishing time in the schedule. The same neighborhood procedures

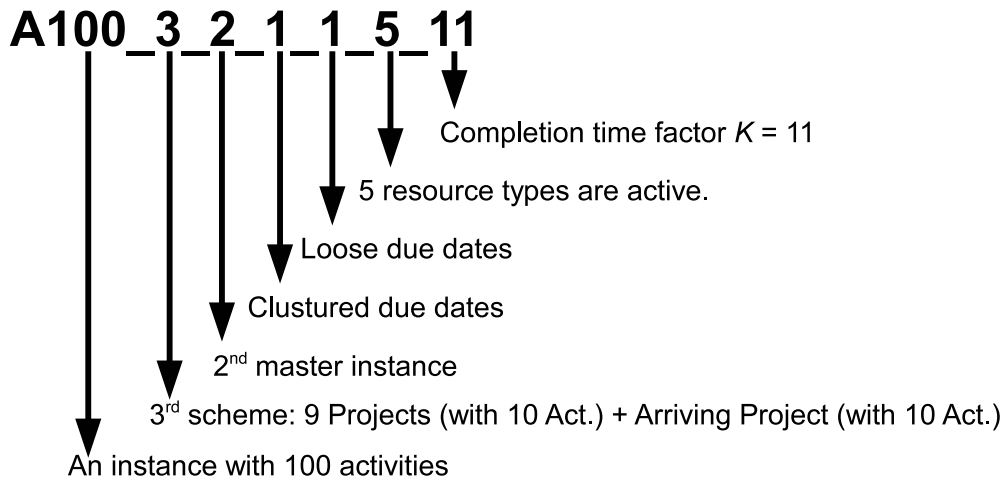


Figure 5.3: Instance naming conventions.

are then applied to improve the solutions obtained in the LP-based method.

All 200 instances with 20, 30, and 40 activities are solved with these four methods. The exact solution procedure is not applied to instances with 50, 100, 150, and 200 activities because of long computational times. For the remaining 600 instances, solutions of LP, H, and LS are provided. All the results of our experiments are given in Appendix C.

The experiments with MIP are performed for a given time limit. This time limit is selected as one hour for instances with 20 and 30 activities and as two hours for 40 activity-instances. The summary of results are given in Table 5.2. As it can be seen on the table, the LP method performs worst among all four methods. An average gap of 84.30% is calculated with respect to the best solution values. It should be noted that for some of the instances MIP is not able to find the optimal solution. Therefore, the average gap for the exact solution algorithm is not equal to 0%, but instead there is an average gap of 4.38%. The local search method, on the other hand, is within 7.83% of gap on average for 200 instances. The hybrid method improves the solutions obtained by LP and reduces the average gap from 84.30% to 40.09%. However, the limited initial population cannot reach the performance of the original LS structure. Combining the results obtained by LS and H (LS + H), in other words generating an initial population consisting of activity lists of the

LS method and the end nodes of the SB tree, results with an average gap value of 5.94% with respect to the best known values. For instances with 50, 100, 150, and 200 activities, LS + H performs best. Combining the results of LS with H improves the average gap to 1.23%. For LP and H, the average gaps are at 157.10% and 102.34%, respectively.

Table 5.2: Summary of results.

Instance Sets	Avg. gap from best solution value (%)				
	MIP	LP	LS	H	LS + H
A20_, A30_, and A40_	4.38	84.30	7.83	40.09	5.94
A50_, A100_, A150_, and A200_		157.10	1.23	102.34	0.00

Out of 200 instances, LP method is able to find the same objective value with MIP for 2 instances and for a single instance the LP method finds a better solution in comparison with MIP. LS, on the other hand, is able to find the same objective value with MIP for 53 instances. For 28 instances, LS can outperform the exact solution method. Considering the whole data set with 800 instances, LP can improve the solutions obtained by LS for 28 instances and find the same objective value for 9 instances. The number of improved solutions are increased from 28 to 96 by implementing the hybrid approach. For 18 instances, the hybrid approach can come up with the same objective value with LP.

A detailed investigation of results is made by comparing average optimality gaps and computational times in Table 5.3. At this point, the instances are categorized according to the number of activities and the number of active resource types. The reason for that is that the computational times change dramatically for different number of active resource types. This statement is valid for the instance sets. As it can be seen in the table, the average CPU times are also increasing as the number of activities increases. The exact solution method is not always able to terminate the process by either solving the problem to optimality or ending the procedure while reaching the time limit. This is the case if the system is in out of memory (OoM) status. The number of instances, for which the solution procedure cannot be completed because of OoM status are presented in parenthesis. The solver

terminates the solution because of OoM status for 25 instances with 2 resource types and for 4 instances with 5 resource types.

Table 5.3: Summary of results.

Act.	Res.	# of Instances	Avg. CPU Time (sec.)				Avg. gap from best solution value (%)				
			MIP	LP	LS	H	MIP	LP	LS	H	LS+H
20	2	20	74.23	1.44	3.11	1.23	0.00	21.17	2.53	7.24	2.09
20	5	20	476.13	13.87	6.05	20.6	0.00	63.23	5.21	20.17	0.15
30	2	40	1240.58 (9)	2.53	4.71	1.93	0.10	94.28	9.78	54.76	8.74
30	5	40	4137.68	34.27	8.91	36.44	1.55	112.18	5.05	53.23	3.24
40	2	40	790.29 (16)	3.22	5.63	2.51	12.40	63.76	9.87	35.61	8.88
40	5	40	5120.50 (4)	43.11	11.99	50.05	7.87	109.09	10.60	43.13	7.74
50	2	60		2.93	7.06	2.64		82.98	1.39	63.01	0.00
50	5	60		29.44	13.57	43.29		91.03	4.77	62.10	0.00
100	2	80		6.38	14.35	6.11		129.27	0.49	93.23	0.00
100	5	80		88.02	26.47	111.19		178.93	2.20	84.72	0.00
150	2	80		10.7	24.36	10.5		143.14	0.47	106.98	0.00
150	5	80		140.81	46.26	184.6		191.13	1.20	114.22	0.00
200	2	80		12.81	34.96	13.62		168.63	0.04	145.43	0.00
200	5	80		142.42	71.68	232.3		236.67	0.17	129.16	0.00

The computation times of LP range from 1.44 seconds to 12.81 seconds for instances with 2 resource types and from 13.87 to 142.42 seconds for instances with 5 resource types. LS is slower than LP for instances with 2 resource types and is able to generate solutions in 3.11 to 34.96 seconds. However, LP completes the solution procedure in 6.05 to 71.68 seconds for the instances with 5 resource types. The hybrid method requires computational times about the sum of CPU times required by the LP and LS methods, which is reasonable since both solution approaches are applied iteratively. The gap between LP and MIP increases as the number of activities increase. Moreover, the performance of LP is also better for instances with 2 resource types. Clearly, more arcs are added to the network in order to avoid resource infeasibilities for increasing number of activities and resource types. Therefore, LP's performance does depend on these parameters. This can be also observed in comparison between the LP and LS. The average gap difference is lower for instances with 2 active resource types in any activity number category. LS, on the other hand, performs better with 2 resource types for instances with 20 and 40 activities with respect to the instances with 5 resource types. However, with 30 activities, increasing number of resources have a positive effect on the performance of the solution method. The hybrid method shows a similar performance trend with LP. This makes sense, since the initial population is generated by this method. However, it should be noted that applying hybrid approach improves the schedules obtained by the LP for all the instance sets. Moreover, combining the results of the

LS and H improves the average gap of LS up to 5.06%.

Table 5.4: Effect of due date tightness on solution quality.

Act.	Tightness	# of Instances	Avg. gap from best solution value (%)				
			MIP	LP	LS	H	LS+H
20	0	20	0.00	52.02	4.22	20.68	1.35
20	1	20	0.00	32.37	3.52	6.72	0.88
30	0	40	0.62	115.30	6.59	56.16	5.51
30	1	40	1.04	91.15	8.25	51.82	6.47
40	0	40	14.41	103.53	12.59	57.13	10.66
40	1	40	5.85	69.32	7.88	21.61	5.96
50	0	60		108.51	1.01	85.12	0.00
50	1	60		65.50	5.14	40.00	0.00
100	0	80		151.29	1.06	106.22	0.00
100	1	80		156.91	1.63	71.73	0.00
150	0	80		144.76	1.45	104.94	0.00
150	1	80		189.50	0.22	116.25	0.00
200	0	80		214.22	0.00	167.66	0.00
200	1	80		191.08	0.21	106.93	0.00

A further data analysis is performed in order to investigate the effects of due date tightness. The results are given in Table 5.4. Tight due dates have a negative effect on the solution performance of LP. Only for instances with 100 and 150 activities, this is not the case. The LS method, on the other hand, can obtain better solutions for 20, 40 and 150 activities with loose due dates in comparison with the tight due date setting for the same number of activities in the instance. The hybrid method can reach better average gap values for all the activity number categories with loose due dates except the one with 150 activities in comparison with the tight setting. That means, H is able to improve the solutions obtained by LP considerably, such that the higher gap value for instances with loose due dates and 100 activities is improved significantly (from 156.91% to 71.73%).

Finally, a comparison is made for clustered and distributed due dates. The results are given in Table 5.5. Since different due date distribution parameters are selected for instances with 50, 100, 150, and 200 activities, no MIP results are present for this analysis. Both methods, LP and H, show better performances, if the due dates are not distributed. The only exception is for instances with 150 activities. Moreover,

Table 5.5: Effects of due date distribution on solution quality.

Act.	Distribution	# of Instances	Avg. gap from best solution value (%)			
			LP	LS	H	LS+H
50	0	80	43.52	4.29	24.31	0.00
50	1	40	173.99	0.65	139.06	0.00
100	0	120	119.95	1.68	60.13	0.00
100	1	40	256.56	0.37	175.51	0.00
150	0	120	172.11	0.16	111.14	0.00
150	1	40	152.22	2.87	108.97	0.00
200	0	120	195.27	0.14	127.78	0.00
200	1	40	224.81	0.00	165.86	0.00

expanding the initial population of LS has a greater effect on solution quality, if the due dates are distributed. This is not the case only for instances with 150 activities.

As a conclusion, the combination of the LS and H performs best among the four methods tested in the experimental study. The MIP is limited by the computation time requirements. The LP does not obtain good solutions in comparison with the other methods. LP, LS and H obtain solutions in reasonable computational times even for instances with 200 activities. Due date tightness and due dates distribution effect the performance of solution approaches. However, these effects are not very distinctive.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this study, an LP-based approach is adopted to solve the resource constrained multi-project scheduling problem with weighted earliness tardiness costs. The main purpose of the work is to model the dynamic project scheduling environments. A project arrives on top of an existing project portfolio and a due date has to be quoted for the new project while keeping the costs related to changes in the schedule at minimum. The objective function consists of the weighted earliness tardiness costs of the existing projects and a completion time related term for the newly arriving project. The solution approach is a generalization of the shifting bottleneck approach from the machine scheduling literature. Resources are activated iteratively and at each activation step a single resource subproblem is solved with the data obtained by applying a sensitivity analysis to the existing resource feasible schedule. The project network is updated by adding arcs to avoid any resource conflicts for activated resources. A local search heuristic is also developed to benchmark the results. In order to analyze the performance and behavior of the proposed method, a new multi-project data set is created by controlling the due date tightness and the due date distribution, the number of resource types, the completion time factor and the total number of activities in an instance. A series of computational experiments are carried out. The performance of the LP-based method, local search approach and a hybrid method are tested. The hybrid method is based on creating an initial population for the local search algorithm with the schedules obtained in the LP-based method. Moreover, exact solutions for small instances are provided by solving the mathematical model. The results show that the LP-based

approach does not perform well for the *RCPSPWET*. The reason behind that is mostly that the activities are using multiple resources. Therefore, activating arcs iteratively for each resource type prevents generating good solutions for the original problem. The heuristic approach, on the other hand, performs well in terms of solution quality and CPU times. By applying a hybrid approach, the performance of the LP-based method is improved considerably. However, the hybrid approach fails to reach the performance of the original local search approach. Combining the best results obtained by the local search procedure and the hybrid method show the best performance among all the tested solution methods in this study.

There are several extension possibilities which can be studied in the future.

- Precedence relations between projects can also be included considering that in practice some projects need to precede others because of technological reasons especially in R&D environments.
- Arrival of multiple projects at a time or at different points in time can be studied to study different dynamic scenarios.
- Multiple-mode extension of the problem can be adopted to define multiple execution options for the activities.

The proposed work is as far as known the first study on the multi-project version of *RCPSPWET*. Moreover, the resource decomposition approach is also a unique approach for resource constrained project scheduling problems. Considering new research possibilities and limited work on the problem, resource constrained multi-project scheduling problem with weighted earliness tardiness costs is a rich topic for further research activities. Moreover, the fact that most of the companies have to manage their project portfolio in dynamic environments offers a wide range of implementation options in the business context.

Bibliography

- [1] K. Bülbül and P. Kaminsky, “A linear programming based general method for job shop scheduling,” 2010.
- [2] E. Demeulemeester, F. Debleare, J. Herbots, O. Lambrechts, and S. V. de Vonder, “A multi-level approach to project management under uncertainty,” *Tijdschrift voor Economie en Management*, vol. LII, no. 3, pp. 391–409, 2007.
- [3] A. Can, “Multi-project scheduling with 2-stage decomposition,” Master’s thesis, Sabancı University, 2010.
- [4] R. Kurtulus and E. W. Davis, “Multi-project scheduling: categorization of heuristic rules performance,” *Management Science*, vol. 28, pp. 161–172, 1982.
- [5] T. Browning and A. Yassine, “Resource-constrained multi-project scheduling: Priority rule performance revisited,” *International Journal of Production Economics*, vol. 126, no. 2, pp. 212 – 228, 2010.
- [6] J. H. Payne, “Management of multiple simultaneous projects: a state-of-the-art review,” *International Journal of Project Management*, vol. 13, pp. 163–168, 1995.
- [7] J. Herbots, W. Herroelen, and R. Leus, “Dynamic order acceptance and capacity planning on a single bottleneck resource,” *Naval Research Logistics*, vol. 54, no. 8, pp. 874–889, 2007.

- [8] K. K. Yang and C. Sum, “An evaluation of due date, resource allocation, project release, and activity scheduling rules in a multi-project environment,” *European Journal of Operational Research*, vol. 13, pp. 139–154, 2007.
- [9] W. Herroelen and R. Leus, “Project scheduling under uncertainty: Survey and research potentials,” *European Journal Of Operational Research*, vol. 165, pp. 289–306, 2005.
- [10] R. Kolisch and R. Padman, “An integrated survey of deterministic project scheduling,” *Omega*, vol. 29, pp. 249–272, 2001.
- [11] L. Özdamar and G. Ulusoy, “A survey on the resource-constrained project scheduling problem,” *IIE Transactions*, vol. 27, pp. 574–586, 1995.
- [12] S. Hartmann and D. Biskron, “A survey of variants and extensions of the resource-constrained project scheduling problem,” *European Journal of Operational Research*, vol. 207, no. 1, pp. 1 – 14, 2010.
- [13] B. Ashtiani, R. Leus, and M. Aryanezhad, “New competitive results for the stochastic resource-constrained project scheduling problem: exploring the benefits of pre-processing,” *Journal of Scheduling*, 2011.
- [14] C. Artigues and F. Roubellat, “A polynomial activity insertion algorithm in a multi-resource schedule with cumulative constraints and multiple modes,” *European Journal of Operational Research*, vol. 127, no. 2, pp. 297 – 316, 2000.
- [15] H. E. Sakkout and M. Wallace, “Probe backtrack search for minimal perturbation in dynamic scheduling,” *Constraints*, vol. 5, no. 4, pp. 359–388, 2000.
- [16] K. Neumann, C. Schwindt, and J. Zimmermann, *Project scheduling with time windows and scarce resources*, 2nd ed., 2003.
- [17] M. Vanhoucke, E. Demeulemeester, and W. Herroelen, “An exact procedure for the unconstrained weighted earliness tardiness project scheduling problem,” *Research Report 9907, Department of Applied Economics, Katholieke Universiteit Leuven*, no. 9907, 1999.

- [18] ———, “An exact procedure for the resource-constrained weighted earliness tardiness project scheduling problem,” *Annals of Operations Research*, vol. 102, pp. 179–196, 2001.
- [19] M. Vanhoucke, “Optimal due date assignment in project scheduling,” *Working Paper, Ghent University and Vleric Luevent Gent Management School*, no. 159, 2002.
- [20] F. Ballestin and N. Trautman, “An iterated-local-search heuristic for the resource-constrained weighted earliness-tardiness project scheduling problem,” *International Journal of Production Research*, vol. 46, pp. 6231–6249, 2008.
- [21] K. Nonobe and T. Ibaraki, *Perspectives in Modern Project Scheduling*. International Series in Operations Research & Management Science, 2006, vol. 92, ch. 9.
- [22] S. Avci and H. R. Storer, “Compact local search neighborhoods for generalized scheduling problems,” 2004.
- [23] P. Brandimarte and M. Maiocco, “Job shop scheduling with a non-regular objective: a comparison of neighbourhood structures based on a sequencing/timing decomposition,” *International Journal of Production Research*, vol. 37, pp. 1697–1715, 1999.
- [24] S. E. Elmaghraby and J. Kamburowski, “The analysis of activity networks under generalized precedence relations (gprs),” *Management Science*, vol. 38, pp. 1245–1263, 1992.
- [25] J. Blazewicz, J. Lenstra, and A. Rinnooy, “Scheduling subject to resource constraints - classification and complexity,” *Discrete Applied Mathematics*, vol. 5, pp. 11–24, 1983.
- [26] L. J. Pritsker, A. A. B. Walters, and P. M. Wolfe, “Multiproject scheduling with limited resources: a zero one programming approach,” *Management Science*, vol. 16, pp. 93–108, 1969.

- [27] M. Speranza and C. Vercellis, “Hierarchical models for multi-project planning and scheduling,” *European Journal of Operational Research*, vol. 64, no. 2, pp. 312 – 325, 1993.
- [28] J. Adams, E. Balas, and D. Zawack, “The shifting bottleneck procedure for job shop scheduling,” *Management Science*, vol. 34, no. 3, pp. 391–401, 1988.
- [29] N. Policella, A. Cesta, A. Oddi, and S. F. Smith, “From precedence constraint posting to partial order schedules a csp approach to robust scheduling,” *AI Communications*, vol. 20, no. 3, pp. 163–180, 2007.
- [30] E. Demirkol, S. Mehta, and R. Uzsoy, “A computational study of shifting bottleneck procedures for shop scheduling problems,” *Journal of Heuristics*, vol. 3, pp. 111–137, 1997.
- [31] R. Kolisch, A. Sprecher, and A. Drexl, “Characterization and generation of a general class of resource-constrained project scheduling problems,” *Management Science*, vol. 41, pp. 1693–1703, 1995.
- [32] C. Schwindt, “Generation of resource-constrained project scheduling problems subject to temporal constraints,” *Technical Report WIOR*, vol. 543, 1998.
- [33] M. Vanhoucke, E. Demeulemeester, and W. Herroelen, “Rangen: A random network generator for activity-on-the-node networks,” Tech. Rep., 2003.
- [34] A. Mastor, “An experimental investigation and comparative evaluation of production line balancing techniques,” *Management Science*, vol. 16, no. 11, pp. 728–746, 1970.
- [35] R. Alvarez-Valdes and J. M. Tamarit, *Heuristic algorithms for resource constrained project scheduling: A review and empirical analysis*, 1989.
- [36] R. Kolisch, “Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation,” *European Journal of Operational Research*, vol. 90, no. 2, pp. 320 – 333, 1996.

Appendix A

LP CONSTRAINT GENERATION ALGORITHM

Table A.1: Parameters used in the algorithm.

I	Set of all projects
P_i	Set of activities of project i
$(0, 0)$	Index of the dummy start activity
$(f, 0)$	Index of the dummy finish activity
$ListPro$	Set of processed activities by the algorithm
$ListUnpro$	Set of unprocessed activities by the algorithm
R_k	Total capacity of resource k
r_{ijk}	Requirement of activity j of project i from resource k
e_{ijk}	The remaining/available capacity of activity j of project i for resource k
e_{f0k}	The remaining/available capacity of the dummy finish activity
e_{00k}	The remaining/available capacity of dummy start activity
s_{ij}	Starting time of activity j of project i in existing solution
f_{ij}	Finishing time of activity j of project i in existing solution
$Pred_{kl}$	Set of all predecessor activities of activity l of project k

input : A feasible solution for the single resource subproblem

output: An LP, preserving the resource feasibility of a particular resource type k

```

1 begin
2   Initialization Let  $e_{00k} = R_k$  and  $ListPro = (0, 0)$  Let  $e_{f0k} = R_k$  and
    $ListUnpro = (f, 0)$  foreach Project  $i \in I$  do
3     foreach Activity  $j \in P_i$  do
4        $e_{ijk} = r_{ijk}$   $ListUnpro = ListUnpro \cup (i, j)$ 
5     end
6   end
7   Processing while  $ListUnpro \neq \emptyset$  do
8     Find  $(i, j)$  such that  $\min_{(i,j) \in ListUnpro} (s_{ij})$  foreach  $(u, v) \in Pred_{ij}$  do
9       if  $(u, v) \in ListPro$  then
10        if  $e_{uvk} \geq e_{ijk}$  then
11           $e_{uvk} = e_{uvk} - e_{ijk}$   $ListPro = ListPro \cup (i, j)$ 
           $ListUnpro = ListUnpro \setminus (i, j)$   $e_{ijk} = r_{ijk}$  break
12        end
13        if  $e_{uvk} < e_{ijk}$  then
14           $e_{ijk} = e_{ijk} - e_{uvk}$   $ListPro = ListPro \setminus (u, v)$ 
15        end
16      end
17    end
18    if  $e_{ijk} > 0$  then
19      Find  $(u, v) \max_{(u,v) \in ListPro} (e_{uvk})$  such that  $f_{uv} \leq s_{ij}$  if  $e_{uvk} \geq e_{ijk}$  then
20         $e_{uvk} = e_{uvk} - e_{ijk}$   $ListPro = ListPro \cup (i, j)$ 
         $ListUnpro = ListUnpro \setminus (i, j)$   $Pred_{ij} = Pred_{ij} \cup (u, v)$   $e_{ijk} = r_{ijk}$ 
21      end
22      if  $e_{uvk} < e_{ijk}$  then
23         $e_{ijk} = e_{ijk} - e_{uvk}$   $ListPro = ListPro \setminus (u, v)$   $Pred_{ij} = Pred_{ij} \cup (u, v)$ 
24      end
25    end
26  end
27 end

```

Algorithm 1: LP Constraint Generation Algorithm.

Appendix B

SETTINGS FOR DATA SET PROJECT POOL GENERATION

Table B.1: Settings for project pool generation.

# of Activities	<i>OS</i>	<i>RU</i>	<i>RC</i>	# of Unique Projects
5	0.25	4	0.25	3
	0.50	4	0.50	10
	0.75	5	0.50	9
	0.50	5	0.25	10
10	0.25	4	0.25	10
	0.5	5	0.25	10
	0.75	5	0.25	10
	0.75	4	0.50	10
	0.25	5	0.50	10
20	0.25	5	0.25	10
	0.50	5	0.25	10
	0.75	5	0.25	10
	0.25	4	0.50	10
	0.75	4	0.50	10
30	0.25	5	0.25	10
	0.50	5	0.25	10
	0.75	5	0.25	10
	0.25	4	0.50	10
	0.75	4	0.50	10

Table B.2: Instances

# of Activities	Combinations	ID	# of MI	Due Date Distribution	Due Date	# of Resource Types	K	E/T Cost Values	# of Instances
20	(5 A, x 3P + 5A x 1P)	A20.1	5	Distributed	Loose or Tight	2 or 5	High or Low	Random (0,10)	40
30	(10 A x 2P + 10 A x 1P)	A30.1	5	Distributed	Loose or Tight	2 or 5	High or Low	Random (0,10)	40
30	(5 A x 5P + 5 A x 1P)	A30.2	5	Distributed	Loose or Tight	2 or 5	High or Low	Random (0,10)	40
40	(10 A x 3P + 10 A x 1P)	A40.1	5	Distributed	Loose or Tight	2 or 5	High or Low	Random (0,10)	40
40	(10 A x 3P 5A x 1P + 5 A, x 1P)	A40.2	5	Distributed	Loose or Tight	2 or 5	High or Low	Random (0,10)	40
50	(10 A x 4P + 10 A x 1P)	A50.1	5	Distributed	Loose or Tight	2 or 5	High or Low	Random (0,10)	40
50	(5A x 8P + 10 A x 1P)	A50.2	5	Clustered or Distributed	Loose or Tight	2 or 5	High or Low	Random (0,10)	80
100	(30 A x 3P + 10 A x 1P)	A100.1	5	Distributed	Loose or Tight	2 or 5	High or Low	Random (0,10)	40
100	(10 A x 2P 20A x 3P + 20 A x 1P)	A100.2	5	Distributed	Loose or Tight	2 or 5	High or Low	Random (0,10)	40
100	(10 A x 9P + 10 A x 1P)	A100.3	5	Clustered or Distributed	Loose or Tight	2 or 5	High or Low	Random (0,10)	80
150	(30 A x 4P + 30 A x 1P)	A150.1	5	Distributed	Loose or Tight	2 or 5	High or Low	Random (0,10)	40
150	(20 A x 6P + 30 A x 1P)	A150.2	5	Distributed	Loose or Tight	2 or 5	High or Low	Random (0,10)	40
150	(10 A x 10P 30 A x 1P + 20 A x 1P)	A150.3	5	Clustered or Distributed	Loose or Tight	2 or 5	High or Low	Random (0,10)	80
200	(30 A x 6P + 20 A x 1P)	A200.1	5	Distributed	Loose or Tight	2 or 5	High or Low	Random (0,10)	40
200	(20 A x 8P 10A x 1P + 30 A x 1P)	A200.2	5	Distributed	Loose or Tight	2 or 5	High or Low	Random (0,10)	40
200	(5 A x 10P 10A x 12P + 30 A x 1P)	A200.3	5	Clustered or Distributed	Loose or Tight	2 or 5	High or Low	Random (0,10)	80

Appendix C

RESULTS

The results of the experimental analysis are provided below. Four different methods are tested: the exact solution method MIP, the LP-Based method LP, the local search approach LS and the hybrid approach H. The instances with 20, 30, and 40 activities are solved by all these four methods. MIP is not applied to remaining data sets. The objective function values and CPU times are given for each method. CPU times are given in seconds. Optimality gap is also provided for MIP, which is the gap reported by CPLEX after termination. Comparisons between solution methods are also performed. For each method, the gap is calculated by comparing the result obtained with this method and the best solution value found for a particular instance. That means if a solution method obtained the best solution value among these four methods, a gap of 0.00% stands for the gap value. Furthermore, an additional column (LS + H) is added to present the combined performance of the original LS with the hybrid approach. For this case, the initial population consists of randomly generated activity lists and the end nodes of the tree in LP method.

Table C.1: Results for instances with 20, 30, and 40 activities.

Dataset	MIP				LP				LS				H				H+LS	
	OFV	Opt. gap	CPU Time	Gap	OFV	CPU Time	Gap	OFV	CPU Time	Gap	OFV	CPU Time	Gap	OFV	CPU Time	Gap	OFV	Gap
A20.1.1.0.0.2.16	715	0	61.11	0.00	726	1.91	1.54	752	3.31	5.17	726	1.53	1.54	382	1.53	1.54	726	1.54
A20.1.1.0.0.2.2.8	376	0	35.34	0.00	563	1.91	49.73	376	3.31	0.00	382	1.52	1.60	0.00	1.52	1.60	382	1.60
A20.1.1.0.0.5.4	260	0	202.11	0.00	430	18.42	65.38	260	6.25	0.00	298	23.80	14.62	0.00	23.80	14.62	298	14.62
A20.1.1.0.0.5.8	444	0	94.17	0.00	555	17.36	25.00	450	6.14	1.35	444	26.33	0.00	0.00	26.33	0.00	444	0.00
A20.1.1.0.1.2.10	336	0	62.03	0.00	433	1.84	28.87	336	3.17	0.00	346	1.47	2.98	0.00	1.47	2.98	346	2.98
A20.1.1.0.1.2.5	205	0	23.17	0.00	224	1.84	9.27	206	3.14	0.49	217	1.48	5.85	0.00	1.48	5.85	217	5.85
A20.1.1.0.1.5.10	468	0	488.97	0.00	529	13.03	13.03	511	6.36	9.19	468	19.70	0.00	0.00	19.70	0.00	468	0.00
A20.1.1.0.1.5.5	325	0	77.06	0.00	385	18.63	18.46	325	6.16	0.00	333	25.59	2.46	0.00	25.59	2.46	333	2.46
A20.1.2.0.0.2.10	407	0	298.44	0.00	475	1.31	16.71	475	3.22	16.71	475	1.23	16.71	0.00	1.23	16.71	475	16.71
A20.1.2.0.0.2.5	285	0	145.98	0.00	375	1.33	31.58	285	3.19	0.00	285	1.25	0.00	0.00	1.25	0.00	285	0.00
A20.1.2.0.0.5.10	820	0	1791.52	0.00	1282	11.84	56.34	820	6.34	0.00	967	18.88	17.93	0.00	18.88	17.93	967	17.93
A20.1.2.0.0.5.5	410	0	180.13	0.00	1132	10.81	176.10	410	6.34	0.00	732	19.73	78.54	0.00	19.73	78.54	732	78.54
A20.1.2.0.1.2.4	284	0	35.94	0.00	372	1.33	30.99	284	3.22	0.00	372	1.19	30.99	0.00	1.19	30.99	372	30.99
A20.1.2.0.1.2.8	509	0	142.88	0.00	594	1.34	16.70	524	3.20	2.95	524	1.19	2.95	0.00	1.19	2.95	524	2.95
A20.1.2.0.1.5.4	316	0	244.39	0.00	993	11.89	214.24	316	6.55	0.00	420	18.89	32.91	0.00	18.89	32.91	420	32.91
A20.1.2.0.1.5.8	632	0	700.23	0.00	1257	11.50	98.89	632	6.42	0.00	696	18.41	10.13	0.00	18.41	10.13	696	10.13
A20.1.3.0.0.2.2	74	0	539.50	0.00	142	1.81	91.89	74	2.97	0.00	75	1.47	1.35	0.00	1.47	1.35	75	1.35
A20.1.3.0.0.2.4	139	0	13.95	0.00	165	1.84	18.71	146	2.98	5.04	148	1.48	6.47	0.00	1.48	6.47	148	6.47
A20.1.3.0.0.5.4	220	0	304.31	0.00	637	21.09	189.55	220	5.92	0.00	478	26.81	117.27	0.00	26.81	117.27	478	117.27
A20.1.3.0.0.5.8	436	0	193.09	0.00	822	21.16	88.53	440	5.75	0.92	721	26.53	65.37	0.00	26.53	65.37	721	65.37
A20.1.3.0.1.2.2	48	0	3.58	0.00	50	0.83	4.17	48	2.86	0.00	48	1.06	0.00	0.00	1.06	0.00	48	0.00
A20.1.3.0.1.2.4	92	0	2.66	0.00	100	1.78	8.70	92	2.80	0.00	92	1.34	0.00	0.00	1.34	0.00	92	0.00
A20.1.3.0.1.5.4	169	0	19.63	0.00	208	14.91	23.08	169	5.91	0.00	169	20.48	0.00	0.00	20.48	0.00	169	0.00
A20.1.3.0.1.5.8	268	0	42.23	0.00	280	14.64	4.48	268	5.77	0.00	268	20.75	0.00	0.00	20.75	0.00	268	0.00
A20.1.4.0.0.2.10	333	0	36.41	0.00	338	1.33	1.50	336	3.14	0.90	338	1.14	1.50	0.00	1.14	1.50	338	1.50
A20.1.4.0.0.2.5	215	0	19.45	0.00	268	1.33	24.65	215	3.20	0.00	268	1.14	24.65	0.00	1.14	24.65	268	24.65
A20.1.4.0.0.5.4	156	0	76.36	0.00	187	11.72	19.87	185	5.94	18.59	158	18.61	1.28	0.00	18.61	1.28	158	1.28
A20.1.4.0.0.5.8	214	0	241.52	0.00	214	11.89	0.00	289	5.92	35.05	214	18.27	0.00	0.00	18.27	0.00	214	0.00
A20.1.4.0.1.2.2	46	0	5.64	0.00	52	1.33	13.04	46	3.08	0.00	48	1.06	4.35	0.00	1.06	4.35	48	4.35
A20.1.4.0.1.2.4	77	0	3.45	0.00	84	1.31	9.09	88	3.06	14.29	84	1.06	9.09	0.00	1.06	9.09	84	9.09
A20.1.4.0.1.5.1	17	0	5.00	0.00	22	6.06	29.41	22	5.58	29.41	17	13.53	0.00	0.00	13.53	0.00	17	13.53
A20.1.4.0.1.5.2	33	0	4.08	0.00	36	6.06	9.09	36	5.59	9.09	33	13.63	0.00	0.00	13.63	0.00	33	13.63
A20.1.5.0.0.2.4	156	0	7.13	0.00	200	1.31	28.21	156	3.11	0.00	168	1.16	7.69	0.00	1.16	7.69	168	7.69
A20.1.5.0.0.2.8	304	0	17.73	0.00	376	1.33	23.68	304	3.06	0.00	371	1.13	22.04	0.00	1.13	22.04	371	22.04
A20.1.5.0.0.5.10	427	0	3164.67	0.00	599	15.84	40.28	430	5.83	0.70	436	20.27	2.11	0.00	20.27	2.11	436	2.11
A20.1.5.0.0.5.5	215	0	58.08	0.00	411	14.95	91.16	215	5.98	0.00	286	20.36	33.02	0.00	20.36	33.02	286	33.02
A20.1.5.0.1.2.12	262	0	8.91	0.00	262	0.92	0.00	262	3.08	0.00	262	0.88	0.00	0.00	0.88	0.00	262	0.00
A20.1.5.0.1.2.6	140	0	21.38	0.00	160	1.47	14.29	147	3.00	5.00	147	0.88	5.00	0.00	0.88	5.00	147	5.00
A20.1.5.0.1.5.2	61	0	40.89	0.00	105	13.53	72.13	61	6.16	0.00	70	20.05	14.75	0.00	20.05	14.75	70	14.75
A20.1.5.0.1.5.4	115	0	1594.06	0.00	149	12.09	29.57	115	6.03	0.00	130	21.45	13.04	0.00	21.45	13.04	130	13.04

Table C.2: Results for instances with 20, 30, and 40 activities-cont.

Dataset	MIP				LP				LS				H				H+LS	
	OFV	Opt. gap	CPU Time	Gap	OFV	CPU Time	Gap	OFV	CPU Time	Gap	OFV	CPU Time	Gap	OFV	CPU Time	Gap	OFV	Gap
A30.1.1.0.0.2.14	1428	17	OoM	0.00	4123	2.80	188.73	1728	5.80	21.01	3309	2.31	131.72	21.01	3309	2.31	131.72	21.01
A30.1.1.0.0.2.28	2912	29	3600.51	0.00	4894	2.84	68.06	3086	5.81	5.98	4580	2.31	57.28	5.98	4580	2.31	57.28	5.98
A30.1.1.0.0.5.11	1540	48	3631.46	0.00	4832	48.83	182.60	1662	11.27	7.92	3500	49.02	127.27	7.92	3500	49.02	127.27	7.92
A30.1.1.0.0.5.22	3058	56	3646.18	0.00	5328	50.20	74.23	3191	10.97	4.35	4264	46.63	39.44	4.35	4264	46.63	39.44	4.35
A30.1.1.0.1.2.11	1122	13	3600.35	0.00	4520	2.81	302.85	1532	5.88	36.54	3876	2.33	245.45	36.54	3876	2.33	245.45	36.54
A30.1.1.0.1.2.22	2244	19	3600.37	0.00	4973	2.75	121.61	2599	5.95	15.82	4344	2.39	93.58	15.82	4344	2.39	93.58	15.82
A30.1.1.0.1.5.13	2083	57	3600.60	1.12	5264	56.00	155.53	2060	11.44	0.00	4452	48.23	116.12	0.00	4452	48.23	116.12	0.00
A30.1.1.0.1.5.26	4194	65	3632.59	8.46	6115	56.25	58.13	3867	11.09	0.00	5163	47.17	33.51	0.00	5163	47.17	33.51	0.00
A30.1.2.0.0.2.10	448	11	OoM	0.67	1038	2.52	133.26	445	4.52	0.00	892	1.92	100.45	0.00	892	1.92	100.45	0.00
A30.1.2.0.0.2.20	843	19	OoM	0.00	1304	2.47	54.69	875	4.48	3.80	1158	1.97	37.37	3.80	1158	1.97	37.37	3.80
A30.1.2.0.0.5.18	864	28	7200.42	0.00	1315	44.30	52.20	864	8.00	0.00	1042	44.66	20.60	0.00	1042	44.66	20.60	0.00
A30.1.2.0.0.5.9	420	0	5218.52	0.00	1201	44.22	185.95	432	8.09	2.86	745	41.89	77.38	2.86	745	41.89	77.38	2.86
A30.1.2.0.1.2.10	424	0	3250.17	0.00	1025	2.50	141.75	445	4.50	4.95	759	1.86	79.01	4.95	759	1.86	79.01	4.95
A30.1.2.0.1.2.20	829	20	OoM	0.00	1272	2.48	53.44	863	4.45	4.10	1069	1.91	28.95	4.10	1069	1.91	28.95	4.10
A30.1.2.0.1.5.10	480	0	665.33	0.00	1187	44.70	147.29	480	8.27	0.00	993	42.75	106.88	0.00	993	42.75	106.88	0.00
A30.1.2.0.1.5.20	1082	36	7200.31	12.71	1527	45.27	59.06	960	8.17	0.00	1173	45.08	22.19	0.00	1173	45.08	22.19	0.00
A30.1.3.0.0.2.12	684	0	152.53	0.00	1807	2.86	164.18	684	5.14	0.00	1801	2.25	163.30	0.00	1801	2.25	163.30	0.00
A30.1.3.0.0.2.24	1352	0	5069.97	0.00	2332	2.92	72.49	1368	5.03	1.18	2228	2.28	64.79	1.18	2228	2.28	64.79	1.18
A30.1.3.0.0.5.16	1045	21	7200.39	5.45	1783	46.17	79.92	991	9.05	0.00	1367	43.28	37.94	0.00	1367	43.28	37.94	0.00
A30.1.3.0.0.5.8	488	0	1022.45	0.00	1189	38.16	143.65	503	9.23	3.07	822	40.16	68.44	3.07	822	40.16	68.44	3.07
A30.1.3.0.1.2.15	855	0	731.27	0.00	2298	2.77	168.77	870	5.09	1.75	1960	2.23	129.24	1.75	1960	2.23	129.24	1.75
A30.1.3.0.1.2.30	1770	17	OoM	3.39	2838	2.80	65.77	1712	5.08	0.00	2532	2.17	47.90	0.00	2532	2.17	47.90	0.00
A30.1.3.0.1.5.11	683	9	9376.31	0.00	1841	53.69	169.55	727	9.27	6.44	1525	47.38	123.28	6.44	1525	47.38	123.28	6.44
A30.1.3.0.1.5.22	1306	3	7200.38	0.00	2691	45.25	106.05	1398	9.27	7.04	2368	43.95	81.32	7.04	2368	43.95	81.32	7.04
A30.1.4.0.0.2.10	475	0	768.52	0.00	1212	2.67	155.16	650	5.39	36.84	753	2.09	58.53	36.84	753	2.09	58.53	36.84
A30.1.4.0.0.2.5	288	0	422.89	0.00	650	2.66	125.69	325	5.36	12.85	409	2.03	42.01	12.85	409	2.03	42.01	12.85
A30.1.4.0.0.5.10	840	54	7282.06	15.07	1549	54.80	112.19	730	9.72	0.00	930	48.64	27.40	0.00	930	48.64	27.40	0.00
A30.1.4.0.0.5.5	365	0	6514.95	0.00	1370	51.08	275.34	365	9.72	0.00	519	49.81	42.19	0.00	519	49.81	42.19	0.00
A30.1.4.0.1.2.4	251	0	682.97	0.00	763	2.53	203.98	260	5.11	3.59	275	1.95	9.56	3.59	275	1.95	9.56	3.59
A30.1.4.0.1.2.8	359	0	601.84	0.00	1104	2.55	207.52	383	5.06	6.69	443	1.97	23.40	6.69	443	1.97	23.40	6.69
A30.1.4.0.1.5.12	984	53	7200.42	14.55	1482	43.64	72.53	859	9.48	0.00	1000	45.11	16.41	0.00	1000	45.11	16.41	0.00
A30.1.4.0.1.5.6	383	17	7200.55	0.00	1326	46.92	246.21	438	9.63	14.36	755	44.77	97.13	14.36	755	44.77	97.13	14.36
A30.1.5.0.0.2.11	913	17	OoM	0.00	3315	2.70	263.09	1207	4.98	32.20	2629	2.19	187.95	32.20	2629	2.19	187.95	32.20
A30.1.5.0.0.2.22	1870	24	OoM	0.00	3757	2.67	100.91	2109	5.11	12.78	3179	2.27	70.00	12.78	3179	2.27	70.00	12.78
A30.1.5.0.0.5.18	1836	32	7200.50	2.00	5023	56.06	179.06	1800	9.39	0.00	2819	50.48	56.61	0.00	2819	50.48	56.61	0.00
A30.1.5.0.0.5.9	900	11	7200.39	0.00	4138	56.33	359.78	900	9.38	0.00	3000	49.69	233.33	0.00	3000	49.69	233.33	0.00
A30.1.5.0.1.2.14	1176	22	OoM	0.00	2574	2.72	118.88	1388	4.91	18.03	1913	2.25	62.67	18.03	1913	2.25	62.67	18.03
A30.1.5.0.1.2.7	581	0	2636.77	0.00	1838	2.63	216.35	814	4.92	40.10	1378	2.11	137.18	40.10	1378	2.11	137.18	40.10
A30.1.5.0.1.5.12	1200	14	7200.33	0.00	5198	52.67	333.17	1200	9.47	0.00	4540	46.47	278.33	0.00	4540	46.47	278.33	0.00
A30.1.5.0.1.5.24	2400	33	7373.52	0.00	6338	57.72	164.08	2400	9.50	0.00	5212	49.64	117.17	0.00	5212	49.64	117.17	0.00

Table C.3: Results for instances with 20, 30, and 40 activities-cont.

Dataset	MIP			LP			LS			H			H+LS		
	OFV	Opt. gap	CPU Time	Gap	OFV	CPU Time	Gap	OFV	CPU Time	Gap	OFV	CPU Time	Gap	OFV	Gap
A30.2.1.0.0.2.14	400	0	209.36	0.00	418	1.83	4.50	410	3.98	2.50	418	1.44	4.50	1.99	1.50
A30.2.1.0.0.2.7	251	0	167.72	0.00	271	1.83	7.97	256	4.02	1.99	271	1.41	7.97	2.52	1.99
A30.2.1.0.0.5.3	149	0	63.03	0.00	357	18.22	139.60	149	8.64	0.00	190	26.47	27.52	0.00	0.00
A30.2.1.0.0.5.6	273	0	603.56	0.00	425	16.42	55.68	286	8.59	4.76	316	24.11	15.75	4.76	4.76
A30.2.1.0.1.2.4	84	0	4.38	0.00	94	1.70	11.90	84	3.73	0.00	84	1.23	0.00	0.00	0.00
A30.2.1.0.1.2.8	168	0	4.39	0.00	179	1.67	6.55	168	3.70	0.00	168	1.22	0.00	0.00	0.00
A30.2.1.0.1.5.12	282	0	49.00	0.00	357	16.64	26.60	292	8.13	3.55	316	25.45	12.06	3.55	3.55
A30.2.1.0.1.5.6	150	0	19.13	0.00	222	16.52	48.00	174	8.06	16.00	160	25.28	6.67	6.67	6.67
A30.2.2.0.0.2.4	238	0	101.72	0.00	504	2.80	111.76	254	4.72	6.72	316	2.05	32.77	6.72	6.72
A30.2.2.0.0.2.8	470	0	375.66	0.00	918	1.88	95.32	490	4.81	4.26	822	1.67	74.89	4.26	4.26
A30.2.2.0.0.5.16	1152	9	7200.39	1.59	1665	18.45	46.83	1134	8.69	0.00	1485	26.72	30.95	0.00	0.00
A30.2.2.0.0.5.8	576	0	272.66	0.00	1350	19.75	134.38	576	8.59	0.00	1044	27.50	81.25	0.00	0.00
A30.2.2.0.1.2.18	812	0	706.98	0.00	1061	2.80	30.67	1023	4.72	25.99	1061	1.92	30.67	25.99	25.99
A30.2.2.0.1.2.9	506	0	1202.63	0.00	737	2.80	45.65	576	4.84	13.83	696	1.97	37.55	13.83	13.83
A30.2.2.0.1.5.10	810	10	7200.34	1.25	1054	15.92	31.75	800	9.03	0.00	960	24.38	20.00	0.00	0.00
A30.2.2.0.1.5.20	1266	16	7200.44	0.00	1434	15.56	13.27	1423	9.19	12.40	1280	24.06	1.11	1.11	1.11
A30.2.3.0.0.2.3	120	0	75.50	0.00	348	2.52	190.00	120	3.83	0.00	191	1.83	59.17	0.00	0.00
A30.2.3.0.0.2.6	240	0	150.39	0.00	315	2.55	31.25	240	3.80	0.00	277	1.80	15.42	0.00	0.00
A30.2.3.0.0.5.3	123	0	90.16	0.00	302	22.72	145.53	123	6.64	0.00	141	27.36	14.63	0.00	0.00
A30.2.3.0.0.5.6	246	0	61.61	0.00	356	22.89	44.72	246	6.61	0.00	287	27.88	16.67	0.00	0.00
A30.2.3.0.1.2.4	116	0	29.69	0.00	124	2.45	6.90	130	3.73	12.07	124	1.70	6.90	6.90	6.90
A30.2.3.0.1.2.8	205	0	35.34	0.00	212	2.48	3.41	230	3.75	12.20	212	1.70	3.41	3.41	3.41
A30.2.3.0.1.5.18	436	0	48.80	0.00	481	17.91	10.32	460	6.50	5.50	446	24.72	2.29	2.29	2.29
A30.2.3.0.1.5.9	238	0	58.92	0.00	283	18.34	18.91	262	6.56	10.08	248	24.69	4.20	4.20	4.20
A30.2.4.0.0.2.8	423	0	1299.70	0.00	669	2.67	13.20	799	4.72	35.19	669	1.95	13.20	13.20	13.20
A30.2.4.0.0.2.8	423	0	1121.52	0.00	517	2.69	22.22	445	4.72	5.20	517	2.00	22.22	5.20	5.20
A30.2.4.0.0.5.18	975	23	7200.44	0.00	1650	15.95	69.23	1188	9.44	21.85	1172	26.83	20.21	20.21	20.21
A30.2.4.0.0.5.9	594	0	3668.13	0.00	1174	15.03	97.64	594	9.56	0.00	920	25.23	54.88	0.00	0.00
A30.2.4.0.1.2.18	900	15	Om	0.00	925	2.73	2.78	977	4.55	8.56	925	2.02	2.78	2.78	2.78
A30.2.4.0.1.2.9	540	1	7200.44	0.00	807	2.67	49.44	540	4.56	0.00	572	2.03	5.93	0.00	0.00
A30.2.4.0.1.5.16	803	25	7200.36	0.00	1263	14.50	57.29	987	9.17	22.91	879	25.02	9.46	9.46	9.46
A30.2.4.0.1.5.8	537	26	7200.34	0.00	1014	13.72	88.83	584	9.20	8.75	818	24.91	52.33	8.75	8.75
A30.2.5.0.0.2.4	184	0	64.70	0.00	443	2.70	140.76	184	4.48	0.00	319	1.88	73.37	0.00	0.00
A30.2.5.0.0.2.8	366	0	352.34	0.00	519	2.69	41.80	368	4.82	0.55	442	1.88	20.77	0.55	0.55
A30.2.5.0.0.5.14	390	0	393.75	0.00	802	25.23	105.64	463	8.72	18.72	414	34.31	6.15	6.15	6.15
A30.2.5.0.0.5.7	236	0	109.52	0.00	573	28.44	142.80	276	8.53	16.95	260	29.22	10.17	10.17	10.17
A30.2.5.0.1.2.18	367	0	37.48	0.00	400	2.08	8.99	367	4.36	0.00	373	1.44	1.63	0.00	0.00
A30.2.5.0.1.2.9	223	0	199.80	0.00	265	2.11	18.83	232	4.33	4.04	238	1.47	6.73	4.04	4.04
A30.2.5.0.1.5.4	103	0	69.56	0.00	148	23.20	43.69	109	8.00	5.83	123	28.86	19.42	5.83	5.83
A30.2.5.0.1.5.8	193	0	29.33	0.00	212	23.00	9.84	210	8.09	8.81	194	29.91	0.52	0.52	0.52

Table C.4: Results for instances with 20, 30, and 40 activities-cont.

Dataset	MIP				LP				LS				H				H+LS		
	OFV	Opt. gap	CPU Time	Gap	OFV	CPU Time	Gap	OFV	CPU Time	Gap	OFV	CPU Time	Gap	OFV	CPU Time	Gap	OFV	Gap	
A40.1.1.0.0.2.13	767	0	926.20	0.00	1951	3.20	154.37	967	5.83	26.08	1580	2.89	106.00	26.08	106.00	26.08	26.08	19.56	
A40.1.1.0.0.2.26	1534	15	OoM	0.00	3445	3.31	124.58	1854	5.70	19.56	2713	2.78	76.86	19.56	76.86	19.56	76.86	19.56	
A40.1.1.0.0.5.11	1056	29	8549.42	0.00	3080	43.55	191.67	1266	13.44	19.89	2074	50.34	96.40	19.89	96.40	19.89	96.40	19.89	
A40.1.1.0.0.5.22	2090	30	7200.59	0.00	3859	54.59	84.64	2267	13.38	8.47	3469	59.27	65.98	8.47	65.98	8.47	65.98	8.47	
A40.1.1.0.1.2.13	810	36	OoM	0.00	1652	3.27	103.95	833	6.08	2.84	1364	2.75	68.40	2.84	68.40	2.84	68.40	2.84	
A40.1.1.0.1.2.26	1638	44	OoM	0.68	2192	3.28	34.73	1627	5.94	0.00	1803	2.78	10.82	0.00	10.82	0.00	10.82	0.00	
A40.1.1.0.1.5.12	1128	41	7508.31	0.00	2626	38.48	132.80	1186	13.41	5.14	1658	47.42	46.99	5.14	46.99	5.14	46.99	5.14	
A40.1.1.0.1.5.6	570	19	7200.47	0.00	868	31.80	634	52.28	13.47	11.23	868	41.39	52.28	11.23	52.28	11.23	52.28	11.23	
A40.1.2.0.0.2.18	1350	31	OoM	0.00	1713	3.48	26.89	1570	5.59	16.30	1713	2.78	26.89	16.30	26.89	16.30	26.89	16.30	
A40.1.2.0.0.2.9	703	15	OoM	0.00	1398	3.41	98.86	911	5.63	29.59	1398	2.73	98.86	29.59	98.86	29.59	98.86	29.59	
A40.1.2.0.0.5.4	676	52	10140.14	39.67	1221	46.22	152.27	484	12.86	0.00	1184	54.50	144.63	0.00	144.63	0.00	144.63	0.00	
A40.1.2.0.0.5.8	1277	62	14567.44	31.92	1928	52.05	99.17	968	12.61	0.00	1699	54.91	75.52	0.00	75.52	0.00	75.52	0.00	
A40.1.2.0.1.2.16	712	14	OoM	0.00	1108	3.27	55.62	734	5.27	3.09	778	2.33	9.27	3.09	9.27	3.09	9.27	3.09	
A40.1.2.0.1.2.8	365	0	1666.00	0.00	559	3.20	53.15	412	5.31	12.88	455	2.30	24.66	12.88	24.66	12.88	24.66	12.88	
A40.1.2.0.1.5.12	1497	60	12976.02	6.85	1821	53.69	29.98	1401	12.66	0.00	1595	57.09	13.85	0.00	13.85	0.00	13.85	0.00	
A40.1.2.0.1.5.6	1076	58	7200.66	51.98	1545	51.67	118.22	708	12.75	0.00	1156	57.94	63.28	0.00	63.28	0.00	63.28	0.00	
A40.1.3.0.0.2.11	693	0	29.56	0.00	1813	3.77	161.62	693	5.53	0.00	1399	2.81	101.88	0.00	101.88	0.00	101.88	0.00	
A40.1.3.0.0.2.22	1383	0	175.50	0.00	2352	3.94	70.07	1386	5.58	0.22	2297	2.94	66.09	0.22	66.09	0.22	66.09	0.22	
A40.1.3.0.0.5.15	1001	0	877.36	0.00	2739	55.48	173.63	1020	10.56	1.90	1983	60.41	98.10	1.90	98.10	1.90	98.10	1.90	
A40.1.3.0.0.5.30	1991	7	7200.52	0.00	4252	54.97	113.56	2040	10.42	2.46	3404	59.58	70.97	2.46	70.97	2.46	70.97	2.46	
A40.1.3.0.1.2.15	1005	0	1036.14	0.00	2077	3.63	106.67	1005	5.56	0.00	1471	2.83	46.37	0.00	46.37	0.00	46.37	0.00	
A40.1.3.0.1.2.30	2040	17	OoM	1.49	2749	3.53	36.77	2010	5.56	0.00	2449	2.73	21.84	0.00	21.84	0.00	21.84	0.00	
A40.1.3.0.1.5.13	977	16	7200.47	0.00	2557	52.63	161.72	979	10.53	0.20	1463	56.39	49.74	0.20	49.74	0.20	49.74	0.20	
A40.1.3.0.1.5.26	1862	23	7200.42	0.00	3314	49.95	77.98	1915	10.52	2.85	2634	55.61	41.46	2.85	41.46	2.85	41.46	2.85	
A40.1.4.0.0.2.12	651	7	OoM	0.00	925	3.28	42.09	807	5.45	23.96	885	2.47	35.94	23.96	35.94	23.96	35.94	23.96	
A40.1.4.0.0.2.6	383	0	1629.28	0.00	661	3.33	72.58	410	5.44	7.05	633	2.50	65.27	7.05	65.27	7.05	65.27	7.05	
A40.1.4.0.0.5.11	824	0	3185.28	0.00	2023	39.78	145.51	858	11.22	4.13	1322	49.81	60.44	4.13	60.44	4.13	60.44	4.13	
A40.1.4.0.0.5.22	1733	20	7200.59	3.28	2450	45.64	44.82	1678	11.19	0.00	2064	52.39	23.00	0.00	23.00	0.00	23.00	0.00	
A40.1.4.0.1.2.16	731	0	212.08	0.00	896	3.31	22.57	768	5.20	5.06	808	2.36	10.53	5.06	10.53	5.06	10.53	5.06	
A40.1.4.0.1.2.8	381	0	243.53	0.00	562	3.27	47.51	384	5.20	0.79	464	2.36	21.78	0.79	21.78	0.79	21.78	0.79	
A40.1.4.0.1.5.12	714	4	7200.44	0.00	1156	43.91	61.90	827	11.16	15.83	862	45.94	20.73	15.83	20.73	15.83	20.73	15.83	
A40.1.4.0.1.5.24	1277	4	7200.39	0.00	1690	47.19	32.34	1414	11.05	10.73	1370	49.08	7.28	10.73	7.28	7.28	10.73	7.28	
A40.1.5.0.0.1.2.12	1288	40	OoM	0.00	3302	3.16	156.37	1320	7.06	2.48	2827	3.17	119.49	2.48	119.49	2.48	119.49	2.48	
A40.1.5.0.0.2.24	2662	53	OoM	0.00	3734	3.20	40.27	2763	7.05	3.79	3247	3.00	21.98	3.79	21.98	3.79	21.98	3.79	
A40.1.5.0.0.5.18	2607	55	7200.70	4.07	4723	52.59	88.54	2505	13.81	0.00	3038	58.72	21.28	0.00	21.28	0.00	21.28	0.00	
A40.1.5.0.0.5.9	1251	55	OoM	0.00	4384	73.53	250.44	1326	13.80	6.00	2363	70.28	88.89	6.00	88.89	6.00	88.89	6.00	
A40.1.5.0.1.2.10	1113	48	OoM	0.00	1810	3.08	62.62	1210	6.91	8.72	1627	2.84	46.18	8.72	46.18	8.72	46.18	8.72	
A40.1.5.0.1.2.20	1988	56	OoM	36.82	2701	3.17	85.89	1453	6.77	0.00	2119	2.89	45.84	0.00	45.84	0.00	45.84	0.00	
A40.1.5.0.1.5.18	2305	70	7200.61	55.85	2914	74.95	97.03	1479	12.92	0.00	1497	73.64	1.22	0.00	1.22	0.00	1.22	0.00	
A40.1.5.0.1.5.9	1233	73	OoM	13.54	2206	74.61	103.13	1124	13.39	3.50	1086	72.33	0.00	3.50	0.00	0.00	0.00	3.50	0.00

Table C.5: Results for instances with 20, 30, and 40 activities-cont.

Dataset	MIP			LP			LS			H			H+LS		
	OFV	Opt. gap	CPU Time	Gap	OFV	CPU Time	Gap	OFV	CPU Time	Gap	OFV	CPU Time	Gap	OFV	Gap
A40.2.1.0.0.2.14	4533	93	OoM	456.88	1221	3.55	50.00	814	6.33	0.00	916	2.78	12.53	0.00	0.00
A40.2.1.0.0.2.28	894	33	OoM		1417	3.48	58.50	1262	6.23	41.16	1112	2.73	24.38	24.38	
A40.2.1.0.0.5.12	1458	71	OoM	6.58	2692	50.16	96.78	1368	13.53	0.00	1773	63.73	29.61	0.00	0.00
A40.2.1.0.0.5.6	881	75	OoM	28.80	2098	48.64	206.73	684	13.61	0.00	1355	60.45	98.10	0.00	0.00
A40.2.1.0.1.2.18	485	0	130.09	0.00	862	3.50	77.73	519	5.61	7.01	616	2.58	27.01	7.01	
A40.2.1.0.1.2.9	316	0	110.09	0.00	586	3.41	85.44	324	5.63	2.53	339	2.47	7.28	2.53	
A40.2.1.0.1.5.18	1809	71	7200.45	53.96	1955	46.67	66.38	1326	13.69	12.85	1175	54.58	0.00	0.00	0.00
A40.2.1.0.1.5.9	1079	70	7200.53	12.98	1775	47.67	85.86	1060	13.81	10.99	955	58.94	0.00	0.00	0.00
A40.2.2.0.0.2.11	426	0	3948.84	0.00	918	2.42	115.49	595	5.98	39.67	669	2.08	57.04	39.67	
A40.2.2.0.0.2.22	805	32	7200.00	0.00	1116	2.42	38.63	962	5.98	19.50	931	2.08	15.65	15.65	
A40.2.2.0.0.5.12	296	0	2077.28	0.00	454	27.56	53.38	333	11.95	12.50	419	37.66	41.55	12.50	
A40.2.2.0.0.5.6	157	0	782.36	0.00	316	27.75	101.27	158	11.77	0.64	281	37.97	78.98	0.64	
A40.2.2.0.1.2.12	246	0	26.61	0.00	406	3.41	65.04	258	5.27	4.88	254	2.58	3.25	3.25	
A40.2.2.0.1.2.24	462	0	82.14	0.00	661	3.30	43.07	486	5.27	5.19	470	2.56	1.73	1.73	
A40.2.2.0.1.5.12	252	0	231.84	0.00	469	24.44	86.11	252	10.89	0.00	297	37.36	17.86	0.00	
A40.2.2.0.1.5.6	132	0	253.81	0.00	412	20.27	212.12	132	10.88	0.00	132	34.64	0.00	0.00	
A40.2.3.0.0.2.10	241	0	317.55	0.00	426	3.58	76.76	241	5.20	0.00	381	2.67	58.09	0.00	0.00
A40.2.3.0.0.2.20	407	0	431.72	0.00	630	3.56	54.79	514	5.28	26.29	575	2.64	41.28	26.29	
A40.2.3.0.0.5.12	283	0	632.02	0.00	794	31.20	180.57	312	11.05	10.25	436	42.06	54.06	10.25	
A40.2.3.0.0.5.6	198	0	388.52	0.00	583	39.56	194.44	210	11.41	6.06	309	41.58	56.06	6.06	
A40.2.3.0.1.2.14	200	0	17.73	0.00	201	1.70	0.50	200	5.03	0.00	201	1.42	0.50	0.00	
A40.2.3.0.1.2.7	102	0	16.42	0.00	117	1.66	14.71	102	5.00	0.00	105	1.41	2.94	0.00	
A40.2.3.0.1.5.2	36	0	211.06	0.00	68	33.50	88.89	46	11.42	27.78	38	43.53	5.56	5.56	
A40.2.3.0.1.5.4	67	0	225.58	0.00	92	33.27	37.31	92	11.11	37.31	87	43.23	29.85	29.85	
A40.2.4.0.0.2.12	244	0	362.52	0.00	480	3.20	96.72	327	5.13	34.02	293	2.38	20.08	20.08	
A40.2.4.0.0.2.6	155	0	304.63	0.00	356	3.25	129.68	183	5.13	18.06	253	2.41	63.23	18.06	
A40.2.4.0.0.5.12	550	52	7200.39	3.19	715	29.92	34.15	567	9.61	6.38	533	37.16	0.00	0.00	
A40.2.4.0.0.5.24	583	9	7200.47	0.00	956	38.36	63.98	894	9.70	53.34	683	41.44	17.15	17.15	
A40.2.4.0.1.2.18	236	0	71.31	0.00	300	3.08	27.12	254	4.86	7.63	264	2.09	11.86	7.63	
A40.2.4.0.1.2.9	128	0	80.53	0.00	151	3.09	17.97	137	4.86	7.03	147	2.14	14.84	7.03	
A40.2.4.0.1.5.12	203	0	124.09	0.00	409	28.16	101.48	281	9.72	38.42	307	40.83	51.23	38.42	
A40.2.4.0.1.5.6	125	0	142.91	0.00	331	28.06	164.80	213	10.05	70.40	199	30.22	59.20	59.20	
A40.2.5.0.0.2.12	449	16	OoM	0.00	547	3.45	21.83	500	5.78	11.36	534	2.55	18.93	11.36	
A40.2.5.0.0.2.6	335	39	OoM	0.00	354	3.52	5.67	362	5.84	8.06	388	2.52	15.82	8.06	
A40.2.5.0.0.5.14	858	50	10438.58	2.14	1595	42.80	89.88	840	13.06	0.00	1072	50.09	27.62	0.00	0.00
A40.2.5.0.0.5.7	475	0	2493.34	0.00	1330	24.39	180.00	686	13.14	44.42	905	36.25	90.53	44.42	
A40.2.5.0.1.2.18	382	0	17.33	0.00	400	3.06	4.71	382	5.11	0.00	386	1.95	1.05	0.00	
A40.2.5.0.1.2.9	202	0	66.98	0.00	220	3.00	8.91	202	5.11	0.00	206	1.92	1.92	0.00	
A40.2.5.0.1.5.10	323	0	243.38	0.00	381	30.14	17.96	323	12.09	0.00	358	41.17	10.84	0.00	0.00
A40.2.5.0.1.5.20	592	0	281.69	0.00	1123	34.44	89.70	593	12.17	0.17	680	42.05	14.86	0.17	0.17

Table C.6: Results for instances with 50, 100, 150, and 200 activities.

Dataset	LP			LS			H			H+LS
	OFV	CPU	Gap	OFV	CPU	Gap	OFV	CPU	Gap	Gap
A50.1.1.0.0.2.11	1886	2.63	114.32	880	7.56	0.00	1869	2.66	112.39	0.00
A50.1.1.0.0.2.22	2559	4.02	45.40	1760	7.36	0.00	1961	3.53	11.42	0.00
A50.1.1.0.0.5.18	1672	53.00	22.76	1362	12.47	0.00	1369	60.28	0.51	0.00
A50.1.1.0.0.5.9	1106	36.91	37.22	806	12.78	0.00	990	49.27	22.83	0.00
A50.1.1.0.1.2.12	1302	3.97	35.91	958	7.39	0.00	1144	3.39	19.42	0.00
A50.1.1.0.1.2.24	1741	2.67	21.66	1507	7.27	5.31	1431	3.42	0.00	0.00
A50.1.1.0.1.5.14	1434	45.20	33.52	1074	12.66	0.00	1138	50.64	5.96	0.00
A50.1.1.0.1.5.28	1754	33.80	29.16	1634	12.27	20.32	1358	33.66	0.00	0.00
A50.1.2.0.0.2.12	1417	3.73	56.06	908	6.91	0.00	1360	3.30	49.78	0.00
A50.1.2.0.0.2.24	2216	3.91	25.06	1772	6.83	0.00	1896	3.36	7.00	0.00
A50.1.2.0.0.5.13	1819	54.58	91.07	952	12.59	0.00	1027	64.45	7.88	0.00
A50.1.2.0.0.5.26	1863	51.27	24.78	1494	12.55	0.07	1493	53.28	0.00	0.00
A50.1.2.0.1.2.11	461	3.48	19.43	397	6.44	2.85	386	2.92	0.00	0.00
A50.1.2.0.1.2.22	917	3.56	63.75	655	6.36	16.96	560	2.91	0.00	0.00
A50.1.2.0.1.5.18	932	28.56	46.54	636	11.25	0.00	652	45.09	2.52	0.00
A50.1.2.0.1.5.9	645	32.67	81.69	355	11.22	0.00	371	41.88	4.51	0.00
A50.1.3.0.0.2.12	1622	4.31	113.70	759	6.48	0.00	1552	3.45	104.48	0.00
A50.1.3.0.0.2.24	2146	4.03	46.28	1467	6.48	0.00	1862	3.33	26.93	0.00
A50.1.3.0.0.5.16	1531	16.83	61.16	950	10.83	0.00	1351	35.11	42.21	0.00
A50.1.3.0.0.5.8	1243	48.59	160.04	478	10.94	0.00	1069	48.38	123.64	0.00
A50.1.3.0.1.2.13	1192	3.83	77.65	671	6.14	0.00	810	3.03	20.72	0.00
A50.1.3.0.1.2.26	1490	4.00	38.22	1078	6.09	0.00	1106	3.20	2.60	0.00
A50.1.3.0.1.5.10	499	36.47	7.08	526	10.59	12.88	466	46.28	0.00	0.00
A50.1.3.0.1.5.20	742	41.84	15.22	778	10.72	20.81	644	42.92	0.00	0.00
A50.1.4.0.0.2.11	2033	4.53	146.42	825	7.08	0.00	2021	3.80	144.97	0.00
A50.1.4.0.0.2.22	3909	3.55	136.91	1650	7.06	0.00	3390	3.47	105.45	0.00
A50.1.4.0.0.5.11	3090	77.11	183.75	1089	14.00	0.00	2997	77.14	175.21	0.00
A50.1.4.0.0.5.22	4404	60.42	109.91	2098	14.13	0.00	3630	65.11	73.02	0.00
A50.1.4.0.1.2.13	2245	3.50	100.27	1121	7.11	0.00	1727	3.36	54.06	0.00
A50.1.4.0.1.2.26	2617	4.36	28.47	2037	7.13	0.00	2370	3.63	16.35	0.00
A50.1.4.0.1.5.12	2877	70.50	142.17	1188	13.84	0.00	2352	74.36	97.98	0.00
A50.1.4.0.1.5.24	3679	62.70	57.22	2340	14.22	0.00	2808	74.38	20.00	0.00
A50.1.5.0.0.2.10	980	2.64	50.77	650	6.73	0.00	980	2.48	50.77	0.00
A50.1.5.0.0.2.20	1399	4.17	8.96	1284	6.72	0.00	1338	3.25	4.21	0.00
A50.1.5.0.0.5.16	1359	36.69	12.41	1241	12.92	2.65	1209	49.14	0.00	0.00
A50.1.5.0.0.5.8	1119	36.59	64.08	682	12.91	0.00	865	47.89	26.83	0.00
A50.1.5.0.1.2.14	438	3.69	24.79	398	6.16	13.39	351	2.61	0.00	0.00
A50.1.5.0.1.2.7	322	3.67	37.61	234	6.19	0.00	280	2.63	19.66	0.00
A50.1.5.0.1.5.11	620	29.41	83.43	338	12.20	0.00	388	36.77	14.79	0.00
A50.1.5.0.1.5.22	838	30.69	55.76	538	12.19	0.00	600	40.84	11.52	0.00
A50.2.1.0.0.2.10	789	1.92	16.03	680	8.06	0.00	789	2.17	16.03	0.00
A50.2.1.0.0.2.20	959	1.94	3.90	923	7.77	0.00	959	2.25	3.90	0.00
A50.2.1.0.0.5.12	862	31.55	1.77	847	13.31	0.00	893	43.53	5.43	0.00
A50.2.1.0.0.5.24	1063	27.86	0.00	1159	13.27	9.03	1063	42.59	0.00	0.00
A50.2.1.0.1.2.11	593	1.89	9.01	544	7.36	0.00	578	2.00	6.25	0.00
A50.2.1.0.1.2.22	944	1.89	8.01	906	7.39	3.66	874	2.06	0.00	0.00
A50.2.1.0.1.5.12	512	17.23	0.00	648	12.56	26.56	512	34.63	0.00	0.00
A50.2.1.0.1.5.24	1022	26.63	25.71	981	12.80	20.66	813	41.66	0.00	0.00
A50.2.1.1.0.2.2	1116	1.97	580.49	164	7.45	0.00	891	2.11	443.29	0.00
A50.2.1.1.0.2.4	1212	2.00	269.51	328	7.44	0.00	730	2.14	122.56	0.00
A50.2.1.1.0.5.2	800	31.83	381.93	166	12.69	0.00	642	42.28	286.75	0.00
A50.2.1.1.0.5.4	952	30.88	186.75	332	12.80	0.00	908	48.94	173.49	0.00
A50.2.1.1.1.2.2	88	1.94	10.00	90	7.03	12.50	80	1.95	0.00	0.00
A50.2.1.1.1.2.4	377	1.95	124.40	168	7.06	0.00	377	1.98	124.40	0.00
A50.2.1.1.1.5.2	243	30.31	182.56	86	12.33	0.00	243	44.88	182.56	0.00
A50.2.1.1.1.5.4	637	34.23	270.35	172	12.23	0.00	204	45.70	18.60	0.00
A50.2.2.0.0.2.16	1069	4.25	28.95	862	7.27	3.98	829	3.34	0.00	0.00
A50.2.2.0.0.2.8	877	4.25	90.65	460	7.34	0.00	686	3.30	49.13	0.00
A50.2.2.0.0.5.14	1584	24.88	11.71	1585	15.38	11.78	1418	41.88	0.00	0.00
A50.2.2.0.0.5.28	1920	24.09	5.73	1965	15.36	8.20	1816	42.53	0.00	0.00

Table C.7: Results for instances with 50, 100, 150, and 200 activities-cont.

Dataset	LP			LS			H			H+LS
	OFV	CPU	Gap	OFV	CPU	Gap	OFV	CPU	Gap	Gap
A50.2.2.0.1.2.18	377	4.19	1.62	378	6.31	1.89	371	3.08	0.00	0.00
A50.2.2.0.1.2.9	213	4.13	12.70	189	6.33	0.00	195	2.95	3.17	0.00
A50.2.2.0.1.5.18	1295	24.20	69.06	766	14.97	0.00	964	41.63	25.85	0.00
A50.2.2.0.1.5.9	734	24.22	31.31	559	15.19	0.00	734	41.22	31.31	0.00
A50.2.2.1.0.2.2	360	4.20	181.25	128	6.72	0.00	360	3.28	181.25	0.00
A50.2.2.1.0.2.4	462	4.27	95.76	236	6.67	0.00	449	3.33	90.25	0.00
A50.2.2.1.0.5.2	1241	17.92	490.95	210	15.56	0.00	1196	36.17	469.52	0.00
A50.2.2.1.0.5.4	1618	24.89	285.24	420	15.30	0.00	1618	41.31	285.24	0.00
A50.2.2.1.1.2.2	44	3.34	10.00	40	6.13	0.00	44	2.23	10.00	0.00
A50.2.2.1.1.2.4	84	3.31	5.00	80	6.16	0.00	84	2.25	5.00	0.00
A50.2.2.1.1.5.2	214	17.80	22.99	174	14.42	0.00	214	38.42	22.99	0.00
A50.2.2.1.1.5.4	1001	17.94	207.06	326	14.41	0.00	1001	37.95	207.06	0.00
A50.2.3.0.0.2.12	1623	2.13	55.76	1042	8.53	0.00	1311	2.61	25.82	0.00
A50.2.3.0.0.2.24	2210	2.13	47.53	1498	8.47	0.00	2122	2.61	41.66	0.00
A50.2.3.0.0.5.10	1508	21.95	25.67	1200	14.84	0.00	1450	40.83	20.83	0.00
A50.2.3.0.0.5.20	1874	21.16	0.21	2190	14.92	17.11	1870	39.84	0.00	0.00
A50.2.3.0.1.2.11	1482	2.08	19.32	1242	8.56	0.00	1482	2.48	19.32	0.00
A50.2.3.0.1.2.22	2077	2.11	0.00	2105	8.44	1.35	2113	2.50	1.73	1.35
A50.2.3.0.1.5.13	941	25.00	0.00	1340	14.78	42.40	941	40.73	0.00	0.00
A50.2.3.0.1.5.26	1663	22.36	20.16	2161	14.45	56.14	1384	37.77	0.00	0.00
A50.2.3.1.0.2.2	808	2.06	270.64	218	8.08	0.00	567	2.58	160.09	0.00
A50.2.3.1.0.2.4	1630	2.05	273.85	436	8.19	0.00	1618	2.56	271.10	0.00
A50.2.3.1.0.5.2	433	24.72	88.26	230	14.17	0.00	417	40.30	81.30	0.00
A50.2.3.1.0.5.4	1032	25.50	124.35	460	14.30	0.00	1010	40.81	119.57	0.00
A50.2.3.1.1.2.1	515	2.02	340.17	117	7.83	0.00	513	2.33	338.46	0.00
A50.2.3.1.1.2.2	949	1.98	305.56	234	7.83	0.00	949	2.38	305.56	0.00
A50.2.3.1.1.5.2	902	25.55	313.76	218	13.25	0.00	539	39.50	147.25	0.00
A50.2.3.1.1.5.4	1311	23.80	200.69	436	13.09	0.00	677	38.45	55.28	0.00
A50.2.4.0.0.2.14	868	2.80	16.98	742	6.52	0.00	759	2.38	2.29	0.00
A50.2.4.0.0.2.7	623	3.33	67.92	371	6.53	0.00	623	2.56	67.92	0.00
A50.2.4.0.0.5.16	1826	17.39	13.98	1728	16.52	7.87	1602	36.78	0.00	0.00
A50.2.4.0.0.5.8	1610	17.59	86.34	864	16.73	0.00	1370	37.41	58.56	0.00
A50.2.4.0.1.2.10	210	0.77	0.00	210	6.13	0.00	210	0.92	0.00	0.00
A50.2.4.0.1.2.5	105	0.78	0.00	105	6.14	0.00	105	0.94	0.00	0.00
A50.2.4.0.1.5.12	1357	16.83	5.85	1282	16.28	0.00	1347	36.73	5.07	0.00
A50.2.4.0.1.5.24	1887	16.55	14.57	1747	16.09	6.07	1647	36.45	0.00	0.00
A50.2.4.1.0.2.1	53	1.97	1.92	52	6.47	0.00	53	1.94	1.92	0.00
A50.2.4.1.0.2.2	296	1.98	190.20	102	6.47	0.00	259	1.91	153.92	0.00
A50.2.4.1.0.5.2	750	17.03	257.14	210	15.25	0.00	696	36.16	231.43	0.00
A50.2.4.1.0.5.4	989	24.23	135.48	420	15.23	0.00	844	43.06	100.95	0.00
A50.2.4.1.1.2.1	21	0.22	0.00	21	5.70	0.00	21	0.70	0.00	0.00
A50.2.4.1.1.2.2	42	0.20	0.00	42	5.73	0.00	42	0.69	0.00	0.00
A50.2.4.1.1.5.2	990	24.69	338.05	226	14.52	0.00	928	41.08	310.62	0.00
A50.2.4.1.1.5.4	1151	15.16	185.61	403	14.33	0.00	1016	34.69	152.11	0.00
A50.2.5.0.0.2.11	1935	4.78	117.17	891	8.23	0.00	1851	3.89	107.74	0.00
A50.2.5.0.0.2.22	2111	2.88	19.94	1760	8.27	0.00	2111	2.83	19.94	0.00
A50.2.5.0.0.5.10	1382	15.92	74.49	792	14.19	0.00	1069	35.03	34.97	0.00
A50.2.5.0.0.5.20	1775	15.91	12.91	1572	14.17	0.00	1775	34.23	12.91	0.00
A50.2.5.0.1.2.12	563	2.16	17.54	576	7.45	20.25	479	2.11	0.00	0.00
A50.2.5.0.1.2.24	1167	3.22	22.33	965	7.42	1.15	954	2.64	0.00	0.00
A50.2.5.0.1.5.11	563	16.30	10.83	508	13.16	0.00	534	33.66	5.12	0.00
A50.2.5.0.1.5.22	906	22.36	4.98	947	13.22	9.73	863	37.88	0.00	0.00
A50.2.5.1.0.2.2	579	2.98	271.15	156	7.56	0.00	423	2.83	171.15	0.00
A50.2.5.1.0.2.4	828	2.98	165.38	312	7.56	0.00	795	2.81	154.81	0.00
A50.2.5.1.0.5.1	81	17.13	3.85	78	13.47	0.00	81	34.44	3.85	0.00
A50.2.5.1.0.5.2	231	17.41	48.08	156	13.16	0.00	231	33.92	48.08	0.00
A50.2.5.1.1.2.2	168	4.38	29.23	130	6.86	0.00	162	3.09	24.62	0.00
A50.2.5.1.1.2.4	510	2.25	107.32	246	6.84	0.00	510	2.13	107.32	0.00
A50.2.5.1.1.5.1	45	15.59	2.27	47	12.14	6.82	44	32.64	0.00	0.00
A50.2.5.1.1.5.2	90	15.80	2.27	94	12.11	6.82	88	32.56	0.00	0.00

Table C.8: Results for instances with 50, 100, 150, and 200 activities-cont.

Dataset	LP			LS			H			H+LS
	OFV	CPU	Gap	OFV	CPU	Gap	OFV	CPU	Gap	Gap
A100_1.1.0.0.2.11	3589	8.16	47.82	2428	14.11	0.00	3218	7.67	32.54	0.00
A100_1.1.0.0.2.22	3985	7.61	8.61	3692	13.70	0.63	3669	7.33	0.00	0.00
A100_1.1.0.0.5.14	7441	194.17	97.32	3771	27.55	0.00	4178	185.59	10.79	0.00
A100_1.1.0.0.5.28	7991	178.70	74.78	5286	27.56	15.62	4572	163.42	0.00	0.00
A100_1.1.0.1.2.14	1253	7.23	64.87	760	11.52	0.00	907	6.42	19.34	0.00
A100_1.1.0.1.2.7	854	7.44	88.94	466	11.50	3.10	452	6.55	0.00	0.00
A100_1.1.0.1.5.14	1917	151.00	119.84	992	23.66	13.76	872	137.50	0.00	0.00
A100_1.1.0.1.5.28	2421	150.41	65.48	1734	23.67	18.52	1463	138.09	0.00	0.00
A100_1.2.0.0.2.14	10693	9.64	247.85	3074	18.61	0.00	10693	8.28	247.85	0.00
A100_1.2.0.0.2.7	4029	9.27	156.30	1572	18.59	0.00	4029	8.17	156.30	0.00
A100_1.2.0.0.5.14	13925	179.97	293.14	3542	37.38	0.00	8836	206.34	149.46	0.00
A100_1.2.0.0.5.28	15986	196.98	125.66	7084	36.83	0.00	10933	191.63	54.33	0.00
A100_1.2.0.1.2.10	7976	9.20	200.41	2655	18.72	0.00	7976	7.73	200.41	0.00
A100_1.2.0.1.2.20	8919	9.73	84.09	4845	18.52	0.00	8586	7.94	77.21	0.00
A100_1.2.0.1.5.16	12046	132.50	181.19	4284	36.19	0.00	6751	159.38	57.59	0.00
A100_1.2.0.1.5.8	13120	127.97	506.28	2164	36.36	0.00	6024	154.25	178.37	0.00
A100_1.3.0.0.2.14	4598	7.98	79.96	2555	13.17	0.00	3546	7.14	38.79	0.00
A100_1.3.0.0.2.28	6083	7.23	42.63	4265	12.97	0.00	4572	6.95	7.20	0.00
A100_1.3.0.0.5.15	7275	122.95	144.29	2978	22.61	0.00	4773	132.14	60.28	0.00
A100_1.3.0.0.5.30	7418	106.25	69.67	4372	22.39	0.00	5063	125.61	15.81	0.00
A100_1.3.0.1.2.14	3553	7.06	96.84	1805	12.64	0.00	2502	6.73	38.61	0.00
A100_1.3.0.1.2.7	3213	7.22	189.98	1108	12.56	0.00	1847	6.84	66.70	0.00
A100_1.3.0.1.5.13	4712	116.78	175.72	1792	23.14	4.86	1709	137.17	0.00	0.00
A100_1.3.0.1.5.26	4303	108.56	88.89	3176	22.84	39.42	2278	127.42	0.00	0.00
A100_1.4.0.0.2.14	9191	8.08	210.40	2961	14.56	0.00	7896	7.34	166.67	0.00
A100_1.4.0.0.2.7	3924	8.13	143.27	1613	14.72	0.00	3779	7.42	134.28	0.00
A100_1.4.0.0.5.14	12059	173.83	151.28	4799	25.53	0.00	9151	175.23	69.85	0.00
A100_1.4.0.0.5.28	12871	182.98	88.48	6829	25.50	0.00	8209	184.20	34.85	0.00
A100_1.4.0.1.2.14	4171	7.17	78.10	2342	14.30	0.00	2537	6.73	8.33	0.00
A100_1.4.0.1.2.7	3595	7.17	168.08	1341	14.19	0.00	2546	6.88	89.86	0.00
A100_1.4.0.1.5.14	5785	154.17	112.61	2721	27.00	0.00	2934	179.23	7.83	0.00
A100_1.4.0.1.5.7	6255	149.91	299.17	1567	27.05	0.00	2269	173.64	44.80	0.00
A100_1.5.0.0.2.10	8679	8.38	225.06	2670	14.97	0.00	8679	7.39	225.06	0.00
A100_1.5.0.0.2.20	9898	8.42	124.34	4412	14.73	0.00	9877	7.66	123.87	0.00
A100_1.5.0.0.5.13	11110	184.45	200.60	3696	25.00	0.00	7331	178.45	98.35	0.00
A100_1.5.0.0.5.26	12393	157.77	104.78	6052	24.88	0.00	8199	171.23	35.48	0.00
A100_1.5.0.1.2.10	7278	8.11	175.89	2638	14.97	0.00	5300	7.48	100.91	0.00
A100_1.5.0.1.2.20	6888	8.70	57.12	4384	14.89	0.00	6342	7.77	44.66	0.00
A100_1.5.0.1.5.12	7153	150.27	150.81	2899	25.78	1.65	2852	163.33	0.00	0.00
A100_1.5.0.1.5.6	5913	118.97	287.48	1526	26.11	0.00	2615	143.19	71.36	0.00
A100_2.1.0.0.2.11	9275	9.36	71.82	5398	17.41	0.00	8131	8.02	50.63	0.00
A100_2.1.0.0.2.22	11014	9.45	57.43	6996	17.42	0.00	10959	8.22	56.65	0.00
A100_2.1.0.0.5.14	8413	95.78	57.75	5333	29.36	0.00	8229	106.59	54.30	0.00
A100_2.1.0.0.5.28	10856	97.30	47.34	7368	28.98	0.00	8970	119.61	21.74	0.00
A100_2.1.0.1.2.14	8742	9.70	72.22	5076	17.30	0.00	8213	7.89	61.80	0.00
A100_2.1.0.1.2.7	4789	5.55	18.36	4046	17.20	0.00	4789	6.08	18.36	0.00
A100_2.1.0.1.5.11	8290	80.97	147.68	3347	28.81	0.00	7674	103.64	129.28	0.00
A100_2.1.0.1.5.22	10615	80.25	106.08	5151	28.78	0.00	9099	109.31	76.65	0.00
A100_2.2.0.0.2.10	10367	7.75	122.23	4665	19.58	0.00	8177	7.81	75.28	0.00
A100_2.2.0.0.2.20	11022	7.75	83.36	6011	19.53	0.00	7576	7.88	26.04	0.00
A100_2.2.0.0.5.14	5949	77.66	50.34	3957	31.28	0.00	4818	101.91	21.76	0.00
A100_2.2.0.0.5.28	7664	77.86	34.86	5683	30.94	0.00	5871	99.75	3.31	0.00
A100_2.2.0.1.2.13	7708	7.25	199.69	2572	17.77	0.00	4445	7.30	72.82	0.00
A100_2.2.0.1.2.26	8706	7.59	109.33	4159	17.55	0.00	5298	7.70	27.39	0.00
A100_2.2.0.1.5.13	6591	75.36	110.44	3132	29.31	0.00	4507	98.16	43.90	0.00
A100_2.2.0.1.5.26	7735	82.02	80.18	5029	29.58	17.14	4293	110.28	0.00	0.00
A100_2.3.0.0.2.13	3621	7.44	62.23	2232	13.41	0.00	2831	7.00	26.84	0.00
A100_2.3.0.0.2.26	4357	7.64	24.27	3506	13.25	0.00	3845	7.05	9.67	0.00
A100_2.3.0.0.5.15	4114	153.52	51.03	2724	23.81	0.00	2876	154.45	5.58	0.00
A100_2.3.0.0.5.30	4346	156.47	24.78	4337	23.61	24.52	3483	154.95	0.00	0.00

Table C.9: Results for instances with 50, 100, 150, and 200 activities-cont.

Dataset	LP			LS			H			H+LS
	OFV	CPU	Gap	OFV	CPU	Gap	OFV	CPU	Gap	Gap
A100.2.3.0.1.2.18	1399	6.91	34.00	1044	11.95	0.00	1123	6.13	7.57	0.00
A100.2.3.0.1.2.9	923	7.33	76.82	522	11.94	0.00	736	6.22	41.00	0.00
A100.2.3.0.1.5.15	1323	103.95	45.23	911	22.05	0.00	976	123.72	7.14	0.00
A100.2.3.0.1.5.30	1967	101.16	20.23	1800	21.95	10.02	1636	106.69	0.00	0.00
A100.2.4.0.0.2.12	3366	7.08	71.73	1960	12.88	0.00	2715	6.59	38.52	0.00
A100.2.4.0.0.2.24	3867	7.41	43.81	2689	12.69	0.00	3259	6.75	21.20	0.00
A100.2.4.0.0.5.12	4498	120.23	75.84	2558	25.05	0.00	3327	139.73	30.06	0.00
A100.2.4.0.0.5.6	3475	144.14	82.41	1905	25.00	0.00	2549	157.92	33.81	0.00
A100.2.4.0.1.2.10	1187	7.11	148.85	477	11.20	0.00	639	6.27	33.96	0.00
A100.2.4.0.1.2.5	749	7.13	212.08	240	11.27	0.00	482	6.20	100.83	0.00
A100.2.4.0.1.5.18	1666	144.11	79.72	927	22.48	0.00	982	133.47	5.93	0.00
A100.2.4.0.1.5.9	1365	124.95	182.02	484	22.20	0.00	766	144.92	58.26	0.00
A100.2.5.0.0.2.16	7204	5.11	240.29	2117	14.89	0.00	6532	5.38	208.55	0.00
A100.2.5.0.0.2.8	5109	5.03	297.59	1285	15.02	0.00	4594	5.44	257.51	0.00
A100.2.5.0.0.5.13	9212	100.80	291.00	2356	25.59	0.00	7347	126.61	211.84	0.00
A100.2.5.0.0.5.26	10385	133.44	187.12	3617	25.42	0.00	7514	141.05	107.74	0.00
A100.2.5.0.1.2.10	3522	5.02	213.62	1123	14.72	0.00	2407	5.30	114.34	0.00
A100.2.5.0.1.2.5	2403	4.97	319.37	573	14.70	0.00	1626	5.28	183.77	0.00
A100.2.5.0.1.5.16	6517	107.69	258.87	1816	24.75	0.00	4306	141.50	137.11	0.00
A100.2.5.0.1.5.8	5211	113.28	468.89	916	24.80	0.00	3599	139.66	292.90	0.00
A100.3.1.0.0.2.11	3508	4.09	83.19	1915	17.00	0.00	3409	5.14	78.02	0.00
A100.3.1.0.0.2.22	3954	4.16	9.65	3606	17.39	0.00	3779	5.27	4.80	0.00
A100.3.1.0.0.5.11	3530	44.95	89.58	1862	27.84	0.00	3607	79.36	93.72	0.00
A100.3.1.0.0.5.22	4167	42.86	22.49	3402	28.09	0.00	4255	77.13	25.07	0.00
A100.3.1.0.1.2.12	1501	3.83	40.02	1072	14.84	0.00	1334	4.75	24.44	0.00
A100.3.1.0.1.2.24	2005	3.73	18.08	1698	14.97	0.00	1772	4.73	4.36	0.00
A100.3.1.0.1.5.12	1906	41.03	22.81	1568	26.50	1.03	1552	73.05	0.00	0.00
A100.3.1.0.1.5.24	2661	41.78	34.12	2204	25.67	11.09	1984	74.63	0.00	0.00
A100.3.1.1.0.2.2	2681	4.19	119.93	1219	15.00	0.00	2656	4.69	117.88	0.00
A100.3.1.1.0.2.4	2843	4.16	87.66	1515	15.38	0.00	2822	4.80	86.27	0.00
A100.3.1.1.0.5.2	2928	47.06	81.53	1755	25.88	8.80	1613	78.92	0.00	0.00
A100.3.1.1.0.5.4	3208	46.58	69.29	2005	25.98	5.80	1895	79.39	0.00	0.00
A100.3.1.1.1.2.3	1476	3.97	196.98	497	14.42	0.00	1194	4.55	140.24	0.00
A100.3.1.1.1.2.6	1841	3.94	90.19	968	14.33	0.00	1790	4.56	84.92	0.00
A100.3.1.1.1.5.2	716	44.13	96.70	364	23.70	0.00	554	77.48	52.20	0.00
A100.3.1.1.1.5.4	1735	42.08	144.37	710	23.70	0.00	1148	76.17	61.69	0.00
A100.3.2.0.0.2.14	5665	5.42	119.06	2586	17.78	0.00	4667	5.81	80.47	0.00
A100.3.2.0.0.2.28	7276	10.09	45.29	5008	17.33	0.00	5511	8.28	10.04	0.00
A100.3.2.0.0.5.12	11690	57.69	357.71	2554	34.20	0.00	7887	91.63	208.81	0.00
A100.3.2.0.0.5.6	13332	64.33	935.09	1288	34.27	0.00	7981	97.36	519.64	0.00
A100.3.2.0.1.2.14	3560	8.72	64.81	2160	15.72	0.00	2539	7.38	17.55	0.00
A100.3.2.0.1.2.7	3492	8.59	140.00	1455	15.95	0.00	1753	7.59	20.48	0.00
A100.3.2.0.1.5.14	7904	51.45	145.47	3220	33.63	0.00	4215	88.70	30.90	0.00
A100.3.2.0.1.5.7	5146	52.56	216.09	1628	33.52	0.00	3733	83.13	129.30	0.00
A100.3.2.1.0.2.3	3261	5.55	441.69	602	15.72	0.00	3063	5.55	408.80	0.00
A100.3.2.1.0.2.6	6361	5.58	464.42	1127	15.72	0.00	6406	5.48	468.41	0.00
A100.3.2.1.0.5.4	9560	52.84	355.02	2101	31.25	0.00	4619	90.02	119.85	0.00
A100.3.2.1.0.5.8	14561	56.80	394.43	2945	31.38	0.00	10140	93.73	244.31	0.00
A100.3.2.1.1.2.3	1371	5.53	113.22	643	14.78	0.00	1995	5.25	210.26	0.00
A100.3.2.1.1.2.6	2307	5.55	85.15	1246	14.69	0.00	1687	5.27	35.39	0.00
A100.3.2.1.1.5.3	7423	53.88	935.29	717	31.00	0.00	3551	86.09	395.26	0.00
A100.3.2.1.1.5.6	9867	86.25	593.88	1422	30.89	0.00	6543	106.06	360.13	0.00
A100.3.3.0.0.2.15	2338	3.67	31.35	1780	12.20	0.00	2159	4.42	21.29	0.00
A100.3.3.0.0.2.30	2808	3.72	11.43	3228	12.70	28.10	2520	4.38	0.00	0.00
A100.3.3.0.0.5.11	2229	57.59	3.58	2152	22.30	0.00	2229	88.95	3.58	0.00
A100.3.3.0.0.5.22	2772	72.23	1.95	2762	22.27	1.58	2719	96.64	0.00	0.00
A100.3.3.0.1.2.11	773	3.31	64.82	469	10.91	0.00	686	3.95	46.27	0.00
A100.3.3.0.1.2.22	1068	3.30	21.50	883	10.98	0.46	879	4.02	0.00	0.00
A100.3.3.0.1.5.12	593	39.58	45.34	412	19.19	0.98	408	67.66	0.00	0.00
A100.3.3.0.1.5.24	836	38.17	17.09	724	19.25	1.40	714	65.70	0.00	0.00

Table C.10: Results for instances with 50, 100, 150, and 200 activities-cont.

Dataset	LP			LS			H			H+LS
	OFV	CPU	Gap	OFV	CPU	Gap	OFV	CPU	Gap	Gap
A100.3.3.1.0.2.2	1769	3.80	94.18	911	11.77	0.00	1769	4.31	94.18	0.00
A100.3.3.1.0.2.4	1831	3.80	66.00	1103	11.98	0.00	1828	4.27	65.73	0.00
A100.3.3.1.0.5.3	1613	57.61	24.46	1296	20.30	0.00	1613	83.16	24.46	0.00
A100.3.3.1.0.5.6	1839	61.02	16.32	1581	20.30	0.00	1831	86.19	15.81	0.00
A100.3.3.1.1.2.3	292	3.86	111.59	138	10.63	0.00	292	3.97	111.59	0.00
A100.3.3.1.1.2.6	409	3.84	51.48	270	10.78	0.00	409	3.88	51.48	0.00
A100.3.3.1.1.5.3	1039	47.38	724.60	126	18.23	0.00	409	70.00	224.60	0.00
A100.3.3.1.1.5.6	1221	45.95	384.52	252	18.22	0.00	597	72.56	136.90	0.00
A100.3.4.0.0.2.14	3719	4.19	101.46	1846	15.09	0.00	2897	4.88	56.93	0.00
A100.3.4.0.0.2.28	4114	4.16	77.17	2322	15.28	0.00	3035	4.92	30.71	0.00
A100.3.4.0.0.5.11	2649	32.77	48.32	1786	25.67	0.00	2069	68.88	15.85	0.00
A100.3.4.0.0.5.22	4510	34.41	63.47	2759	25.47	0.00	2983	66.36	8.12	0.00
A100.3.4.0.1.2.15	3771	4.13	117.60	1733	15.30	0.00	3266	4.98	88.46	0.00
A100.3.4.0.1.2.30	4117	4.13	78.46	2307	15.13	0.00	3758	4.98	62.90	0.00
A100.3.4.0.1.5.12	4026	33.67	80.46	2231	25.67	0.00	3114	65.20	39.58	0.00
A100.3.4.0.1.5.24	3791	33.75	47.68	2567	25.78	0.00	3124	65.59	21.70	0.00
A100.3.4.1.0.2.1	1901	4.00	967.98	178	13.53	0.00	1828	4.86	926.97	0.00
A100.3.4.1.0.2.2	2031	3.98	470.51	356	13.50	0.00	1958	4.88	450.00	0.00
A100.3.4.1.0.5.2	2968	31.11	729.05	358	23.19	0.00	2263	64.56	532.12	0.00
A100.3.4.1.0.5.4	3131	31.20	337.29	716	23.30	0.00	3131	64.36	337.29	0.00
A100.3.4.1.1.2.2	3352	4.14	263.16	923	12.95	0.00	1476	4.81	59.91	0.00
A100.3.4.1.1.2.4	3550	4.14	246.34	1025	12.92	0.00	1562	4.78	52.39	0.00
A100.3.4.1.1.5.2	2322	30.56	545.00	360	22.91	0.00	2094	63.77	481.67	0.00
A100.3.4.1.1.5.4	2779	30.78	285.97	720	23.11	0.00	2504	63.50	247.78	0.00
A100.3.5.0.0.2.11	1367	7.98	43.14	955	13.53	0.00	1414	6.72	48.06	0.00
A100.3.5.0.0.2.22	2183	7.95	89.33	1153	13.63	0.00	1901	6.64	64.87	0.00
A100.3.5.0.0.5.18	3980	38.27	103.16	1959	29.78	0.00	2753	90.14	40.53	0.00
A100.3.5.0.0.5.9	2886	45.30	80.49	1599	29.88	0.00	2555	94.89	59.79	0.00
A100.3.5.0.1.2.13	556	7.39	59.31	366	11.16	4.87	349	5.88	0.00	0.00
A100.3.5.0.1.2.26	923	7.55	31.86	717	11.20	2.43	700	5.88	0.00	0.00
A100.3.5.0.1.5.13	3498	46.08	58.78	2203	29.27	0.00	2492	84.64	13.12	0.00
A100.3.5.0.1.5.26	4303	44.50	63.24	2636	28.98	0.00	2765	80.63	4.89	0.00
A100.3.5.1.0.2.2	1550	7.91	110.03	738	12.42	0.00	1404	6.33	90.24	0.00
A100.3.5.1.0.2.4	1486	7.98	55.11	958	12.61	0.00	1486	6.42	55.11	0.00
A100.3.5.1.0.5.2	3764	42.03	171.77	1385	26.95	0.00	2193	78.33	58.34	0.00
A100.3.5.1.0.5.4	3102	44.28	115.87	1437	26.91	0.00	3005	81.88	109.12	0.00
A100.3.5.1.1.2.2	120	5.02	71.43	70	10.48	0.00	115	4.48	64.29	0.00
A100.3.5.1.1.2.4	164	5.00	17.14	140	10.31	0.00	159	4.48	13.57	0.00
A100.3.5.1.1.5.2	3977	33.97	75.20	2270	26.23	0.00	2545	68.86	12.11	0.00
A100.3.5.1.1.5.4	4049	33.83	57.55	2570	26.25	0.00	3064	69.09	19.22	0.00
A150.1.1.0.0.2.14	22034	11.98	186.27	7697	29.83	0.00	18309	12.14	137.87	0.00
A150.1.1.0.0.2.28	24104	12.36	129.02	10525	29.75	0.00	19250	12.44	82.90	0.00
A150.1.1.0.0.5.14	55146	197.73	459.01	9865	84.53	0.00	30300	260.86	207.15	0.00
A150.1.1.0.0.5.7	30582	184.02	234.08	9154	84.58	0.00	18886	259.75	106.31	0.00
A150.1.1.0.1.2.11	12024	12.31	368.04	2569	27.84	0.00	7205	12.17	180.46	0.00
A150.1.1.0.1.2.22	14759	12.44	204.81	4842	28.03	0.00	10492	12.25	116.69	0.00
A150.1.1.0.1.5.10	36842	165.44	284.49	9582	85.13	0.00	21113	276.08	120.34	0.00
A150.1.1.0.1.5.5	24655	187.98	239.09	7271	84.86	0.00	13375	268.66	83.95	0.00
A150.1.2.0.0.2.16	39044	13.25	143.92	16007	35.09	0.00	35791	13.52	123.60	0.00
A150.1.2.0.0.2.8	14096	12.75	8.51	12991	34.89	0.00	13759	13.16	5.91	0.00
A150.1.2.0.0.5.10	63684	290.66	360.18	13839	60.86	0.00	42406	298.48	206.42	0.00
A150.1.2.0.0.5.20	65071	251.08	258.74	18139	60.77	0.00	46619	267.34	157.01	0.00
A150.1.2.0.1.2.14	26581	13.41	73.74	15299	34.81	0.00	26855	13.58	75.53	0.00
A150.1.2.0.1.2.28	35400	14.17	71.21	20676	34.44	0.00	34606	13.64	67.37	0.00
A150.1.2.0.1.5.10	46911	323.28	331.92	10861	61.83	0.00	25603	330.67	135.73	0.00
A150.1.2.0.1.5.5	36023	262.50	317.37	8631	61.31	0.00	13963	300.20	61.78	0.00
A150.1.3.0.0.2.18	30584	12.41	243.22	8911	34.61	0.00	25843	13.03	190.01	0.00
A150.1.3.0.0.2.9	26481	12.55	305.71	6527	34.47	0.00	21643	12.84	231.59	0.00
A150.1.3.0.0.5.12	40992	254.98	321.73	9720	60.77	0.00	24840	278.84	155.56	0.00
A150.1.3.0.0.5.24	39532	248.86	204.73	12973	61.00	0.00	26320	282.34	102.88	0.00

Table C.11: Results for instances with 50, 100, 150, and 200 activities-cont.

Dataset	LP			LS			H			H+LS	
	OFV	CPU	Gap	OFV	CPU	Gap	OFV	CPU	Gap	Gap	Gap
A150.1.3.0.1.2.12	15767	12.19	330.56	3662	31.55	0.00	8560	12.61	133.75	0.00	0.00
A150.1.3.0.1.2.6	12867	12.02	572.26	1914	31.06	0.00	6937	12.28	262.43	0.00	0.00
A150.1.3.0.1.5.16	24099	241.45	345.12	5414	58.27	0.00	10517	279.81	94.26	0.00	0.00
A150.1.3.0.1.5.8	20432	235.09	528.10	3253	58.59	0.00	9815	275.67	201.72	0.00	0.00
A150.1.4.0.0.2.11	12671	11.64	145.94	5152	25.28	0.00	10971	11.47	112.95	0.00	0.00
A150.1.4.0.0.2.22	15284	11.80	108.74	7322	25.28	0.00	13577	11.52	85.43	0.00	0.00
A150.1.4.0.0.5.13	21706	250.44	206.63	7079	46.89	0.00	14703	266.72	107.70	0.00	0.00
A150.1.4.0.0.5.26	21887	269.94	132.22	9425	46.86	0.00	16015	278.75	69.92	0.00	0.00
A150.1.4.0.1.2.12	4823	12.58	142.48	1989	21.19	0.00	4146	11.19	108.45	0.00	0.00
A150.1.4.0.1.2.6	4222	12.14	292.74	1075	21.30	0.00	3023	10.92	181.21	0.00	0.00
A150.1.4.0.1.5.10	5524	211.95	160.94	2117	39.67	0.00	3320	247.20	56.83	0.00	0.00
A150.1.4.0.1.5.20	8170	215.52	110.84	3875	39.69	0.00	4271	239.03	10.22	0.00	0.00
A150.1.5.0.0.2.13	20477	13.52	110.89	9710	31.41	0.00	18255	12.92	88.00	0.00	0.00
A150.1.5.0.0.2.26	22563	13.66	60.77	14034	31.41	0.00	20442	12.64	45.66	0.00	0.00
A150.1.5.0.0.5.14	31000	231.81	160.24	11912	54.27	0.00	20945	255.94	75.83	0.00	0.00
A150.1.5.0.0.5.7	25791	216.16	152.19	10227	54.05	0.00	18754	252.20	83.38	0.00	0.00
A150.1.5.0.1.2.18	22983	13.88	70.26	13499	32.19	0.00	22736	13.16	68.43	0.00	0.00
A150.1.5.0.1.2.9	20076	13.47	88.74	10637	32.05	0.00	20070	12.89	88.68	0.00	0.00
A150.1.5.0.1.5.14	21998	220.66	216.61	6948	52.25	0.00	14445	244.22	107.90	0.00	0.00
A150.1.5.0.1.5.28	22827	163.52	94.04	11764	52.16	0.00	16146	228.00	37.25	0.00	0.00
A150.2.1.0.0.2.18	29646	14.02	92.78	15378	35.75	0.00	29088	13.63	89.15	0.00	0.00
A150.2.1.0.0.2.9	18527	15.91	48.13	12507	35.95	0.00	19248	14.48	53.90	0.00	0.00
A150.2.1.0.0.5.10	38537	261.34	145.44	15701	66.58	0.00	25777	279.22	64.17	0.00	0.00
A150.2.1.0.0.5.20	64473	239.42	234.21	19291	66.88	0.00	36428	267.33	88.83	0.00	0.00
A150.2.1.0.1.2.10	22607	13.09	33.73	16905	35.92	0.00	20371	13.23	20.50	0.00	0.00
A150.2.1.0.1.2.5	20065	15.66	31.23	15290	35.69	0.00	19328	14.20	26.41	0.00	0.00
A150.2.1.0.1.5.18	46957	270.42	259.47	13063	65.91	0.00	28521	290.64	118.33	0.00	0.00
A150.2.1.0.1.5.9	30476	263.59	209.12	9859	66.28	0.00	15393	281.17	56.13	0.00	0.00
A150.2.2.0.0.2.11	17951	12.77	384.25	3707	28.55	0.00	17061	12.47	360.24	0.00	0.00
A150.2.2.0.0.2.22	19631	12.45	205.07	6435	28.38	0.00	19391	12.13	201.34	0.00	0.00
A150.2.2.0.0.5.18	22339	126.69	203.02	7372	46.69	0.00	16806	174.66	127.97	0.00	0.00
A150.2.2.0.0.5.9	19696	115.72	275.73	5242	46.73	0.00	15454	164.64	194.81	0.00	0.00
A150.2.2.0.1.2.12	9752	12.16	207.44	3172	25.97	0.00	8915	11.66	181.05	0.00	0.00
A150.2.2.0.1.2.24	11562	11.86	92.19	6016	25.64	0.00	9350	12.45	55.42	0.00	0.00
A150.2.2.0.1.5.10	7358	108.19	215.12	2335	43.06	0.00	6139	158.09	162.91	0.00	0.00
A150.2.2.0.1.5.5	4310	123.95	253.28	1220	42.47	0.00	3669	171.77	200.74	0.00	0.00
A150.2.3.0.0.2.16	7452	11.64	99.57	3734	22.48	0.00	6435	11.36	72.34	0.00	0.00
A150.2.3.0.0.2.8	6827	11.44	169.42	2534	22.42	0.00	5478	11.17	116.18	0.00	0.00
A150.2.3.0.0.5.13	8047	233.77	104.24	3940	40.61	0.00	5155	254.63	30.84	0.00	0.00
A150.2.3.0.0.5.26	10161	212.23	70.80	6007	40.34	0.97	5949	254.19	0.00	0.00	0.00
A150.2.3.0.1.2.14	1647	10.77	38.40	1190	18.42	0.00	1681	10.03	41.26	0.00	0.00
A150.2.3.0.1.2.7	1213	10.95	103.87	595	18.45	0.00	1106	10.08	85.88	0.00	0.00
A150.2.3.0.1.5.12	2101	212.42	103.59	1032	33.94	0.00	1152	225.06	11.63	0.00	0.00
A150.2.3.0.1.5.24	2692	215.64	50.14	2054	34.22	14.56	1793	230.31	0.00	0.00	0.00
A150.2.4.0.0.2.14	13901	11.81	125.37	6168	28.94	0.00	12777	11.59	107.15	0.00	0.00
A150.2.4.0.0.2.7	13105	11.36	167.18	4905	28.89	0.00	12555	11.36	155.96	0.00	0.00
A150.2.4.0.0.5.10	12722	202.52	161.39	4867	46.80	0.00	11186	206.64	129.83	0.00	0.00
A150.2.4.0.0.5.20	13830	186.52	104.89	6750	46.67	0.00	11295	217.86	67.33	0.00	0.00
A150.2.4.0.1.2.14	6477	10.91	275.70	1724	23.30	0.00	4664	11.22	170.53	0.00	0.00
A150.2.4.0.1.2.28	8024	11.00	135.58	3406	23.31	0.00	5466	10.92	60.48	0.00	0.00
A150.2.4.0.1.5.18	7243	162.36	232.40	2179	40.11	0.00	5158	222.50	136.71	0.00	0.00
A150.2.4.0.1.5.9	5457	166.83	396.54	1099	40.41	0.00	4012	190.58	265.06	0.00	0.00
A150.2.5.0.0.2.10	11000	8.38	251.21	3132	25.48	0.00	10992	9.09	250.96	0.00	0.00
A150.2.5.0.0.2.20	12030	9.23	159.66	4633	25.56	0.00	12372	9.77	167.04	0.00	0.00
A150.2.5.0.0.5.12	19718	110.84	285.27	5118	70.28	0.00	15733	174.33	207.41	0.00	0.00
A150.2.5.0.0.5.24	21679	112.53	172.69	7950	69.83	0.00	16216	180.34	103.97	0.00	0.00
A150.2.5.0.1.2.12	5970	11.91	330.12	1388	21.78	0.00	4655	10.91	235.37	0.00	0.00
A150.2.5.0.1.2.6	1624	12.45	103.25	799	22.42	0.00	1772	10.88	121.78	0.00	0.00
A150.2.5.0.1.5.16	8898	95.30	146.28	3613	57.33	0.00	6743	160.75	86.63	0.00	0.00
A150.2.5.0.1.5.8	9369	107.97	394.93	1893	57.66	0.00	5670	172.83	199.52	0.00	0.00

Table C.12: Results for instances with 50, 100, 150, and 200 activities-cont.

Dataset	LP			LS			H			H+LS
	OFV	CPU	Gap	OFV	CPU	Gap	OFV	CPU	Gap	Gap
A150.3.1.0.0.2.12	7909	7.94	83.63	4307	23.17	0.00	7850	8.63	82.26	0.00
A150.3.1.0.0.2.6	3647	8.11	1.84	3581	23.08	0.00	3635	8.61	1.51	0.00
A150.3.1.0.0.5.14	7504	83.39	60.17	4685	37.80	0.00	7108	125.45	51.72	0.00
A150.3.1.0.0.5.7	7020	84.97	83.24	3831	37.67	0.00	6566	126.28	71.39	0.00
A150.3.1.0.1.2.14	3042	7.02	78.94	1700	18.69	0.00	2346	7.63	38.00	0.00
A150.3.1.0.1.2.28	3874	7.08	32.13	2932	18.98	0.00	3237	7.58	10.40	0.00
A150.3.1.0.1.5.12	2707	66.19	79.75	1506	32.86	0.00	2271	114.09	50.80	0.00
A150.3.1.0.1.5.24	3482	71.19	24.54	2796	33.11	0.00	2812	125.03	0.57	0.00
A150.3.1.1.0.2.10	5453	11.50	40.25	3888	21.55	0.00	5467	9.80	40.61	0.00
A150.3.1.1.0.2.5	5225	8.30	64.67	3173	21.52	0.00	4853	8.25	52.95	0.00
A150.3.1.1.0.5.4	3331	73.77	25.79	2648	35.86	0.00	2915	118.41	10.08	0.00
A150.3.1.1.0.5.8	5087	83.13	48.14	3434	36.05	0.00	5089	132.69	48.19	0.00
A150.3.1.1.1.2.10	2866	7.02	138.44	1202	18.89	0.00	2699	7.22	124.54	0.00
A150.3.1.1.1.2.5	2381	7.00	270.87	642	18.59	0.00	2374	7.06	269.78	0.00
A150.3.1.1.1.5.4	1120	67.02	60.92	696	32.95	0.00	1028	113.47	47.70	0.00
A150.3.1.1.1.5.8	3157	86.86	175.00	1148	32.81	0.00	2628	127.61	128.92	0.00
A150.3.2.0.0.2.16	15067	8.53	254.60	4249	29.55	0.00	15475	9.98	264.20	0.00
A150.3.2.0.0.2.8	13838	8.61	510.95	2265	29.66	0.00	13085	10.09	477.70	0.00
A150.3.2.0.0.5.12	15843	79.95	310.23	3862	45.92	0.00	14595	131.05	277.91	0.00
A150.3.2.0.0.5.6	11534	53.06	385.85	2374	45.81	0.00	11534	111.06	385.85	0.00
A150.3.2.0.1.2.18	10680	6.61	123.90	4770	27.58	0.00	10419	8.81	118.43	0.00
A150.3.2.0.1.2.9	9042	6.53	277.69	2394	27.45	0.00	9809	8.70	309.73	0.00
A150.3.2.0.1.5.16	11898	52.03	180.88	4236	43.20	0.00	10200	110.97	140.79	0.00
A150.3.2.0.1.5.8	10188	88.72	379.66	2124	43.16	0.00	9495	140.78	347.03	0.00
A150.3.2.1.0.2.3	6285	8.28	126.00	2781	25.05	0.00	3702	8.86	33.12	0.00
A150.3.2.1.0.2.6	6985	8.20	98.66	3516	24.73	0.00	4431	9.08	26.02	0.00
A150.3.2.1.0.5.3	8893	50.95	142.38	3669	38.97	0.00	8887	107.48	142.22	0.00
A150.3.2.1.0.5.6	9609	53.11	113.11	4509	39.13	0.00	9624	108.94	113.44	0.00
A150.3.2.1.1.2.4	5981	13.69	462.12	1064	24.81	0.00	1870	11.59	75.75	0.00
A150.3.2.1.1.2.8	13286	8.39	524.34	2128	24.64	0.00	12998	9.30	510.81	0.00
A150.3.2.1.1.5.3	8220	52.73	911.07	813	38.03	0.00	7996	104.95	883.52	0.00
A150.3.2.1.1.5.6	10726	74.08	559.66	1626	38.09	0.00	8727	126.13	436.72	0.00
A150.3.3.0.0.2.13	3302	7.97	43.63	2299	18.78	0.00	3058	7.89	33.01	0.00
A150.3.3.0.0.2.26	5597	8.02	59.55	3508	18.92	0.00	5448	7.98	55.30	0.00
A150.3.3.0.0.5.10	3907	106.86	80.13	2169	33.97	0.00	2867	157.58	32.18	0.00
A150.3.3.0.0.5.20	6482	114.52	108.49	3109	34.14	0.00	5275	163.03	69.67	0.00
A150.3.3.0.1.2.13	1231	5.56	29.72	949	17.30	0.00	1231	6.41	29.72	0.00
A150.3.3.0.1.2.26	2035	9.16	15.17	1767	17.34	0.00	2035	8.23	15.17	0.00
A150.3.3.0.1.5.10	1380	94.95	81.82	759	31.67	0.00	976	150.31	28.59	0.00
A150.3.3.0.1.5.20	1942	94.63	31.13	1530	31.59	3.31	1481	148.53	0.00	0.00
A150.3.3.1.0.2.2	1555	5.86	8.29	1741	17.06	21.24	1436	6.56	0.00	0.00
A150.3.3.1.0.2.4	1810	5.86	9.04	1927	17.27	16.08	1660	6.58	0.00	0.00
A150.3.3.1.0.5.2	1773	95.95	34.52	1342	31.20	1.82	1318	145.95	0.00	0.00
A150.3.3.1.0.5.4	2011	95.58	31.96	1524	31.58	0.00	1980	146.94	29.92	0.00
A150.3.3.1.1.2.3	739	5.67	242.13	216	15.64	0.00	649	6.34	200.46	0.00
A150.3.3.1.1.2.6	901	5.59	108.56	432	15.59	0.00	730	6.27	68.98	0.00
A150.3.3.1.1.5.4	414	81.36	36.18	304	29.03	0.00	331	113.66	8.88	0.00
A150.3.3.1.1.5.8	1627	114.92	167.60	608	28.95	0.00	1023	159.14	68.26	0.00
A150.3.4.0.0.2.12	5346	11.98	167.30	2000	19.98	0.00	5346	10.67	167.30	0.00
A150.3.4.0.0.2.24	6593	11.91	89.67	3476	20.14	0.00	6435	10.84	85.13	0.00
A150.3.4.0.0.5.10	10956	62.70	360.14	2381	45.17	0.00	9064	132.19	280.68	0.00
A150.3.4.0.0.5.20	12061	74.86	204.49	3961	45.55	0.00	10496	130.06	164.98	0.00
A150.3.4.0.1.2.13	1724	11.33	37.15	1257	17.55	0.00	1704	9.98	35.56	0.00
A150.3.4.0.1.2.26	2671	7.73	23.94	2155	17.69	0.00	2479	8.03	15.03	0.00
A150.3.4.0.1.5.10	2453	63.83	84.44	1330	37.78	0.00	1973	125.83	48.35	0.00
A150.3.4.0.1.5.20	4067	65.13	75.23	2321	38.00	0.00	2544	122.70	9.61	0.00
A150.3.4.1.0.2.3	2337	12.03	42.07	1645	18.81	0.00	2220	10.41	34.95	0.00
A150.3.4.1.0.2.6	2682	12.13	34.37	1996	18.63	0.00	2553	10.59	27.91	0.00
A150.3.4.1.0.5.4	2679	76.61	58.61	2654	39.61	57.13	1689	126.02	0.00	0.00
A150.3.4.1.0.5.8	8791	83.33	168.84	3270	39.48	0.00	8791	131.27	168.84	0.00

Table C.13: Results for instances with 50, 100, 150, and 200 activities-cont.

Dataset	LP			LS			H			H+LS
	OFV	CPU	Gap	OFV	CPU	Gap	OFV	CPU	Gap	Gap
A150.3.4.1.1.2.3	1278	7.92	235.43	381	16.39	0.00	781	8.19	104.99	0.00
A150.3.4.1.1.2.6	1261	7.73	99.21	633	16.41	0.00	995	8.22	57.19	0.00
A150.3.4.1.1.5.10	4609	66.72	138.81	1930	33.97	0.00	3919	116.30	103.06	0.00
A150.3.4.1.1.5.5	1255	71.14	17.84	1065	34.19	0.00	1193	119.89	12.02	0.00
A150.3.5.0.0.2.11	5311	11.66	48.43	3578	21.17	0.00	4604	11.25	28.68	0.00
A150.3.5.0.0.2.22	5778	11.72	20.95	4777	21.25	0.00	5136	11.06	7.52	0.00
A150.3.5.0.0.5.14	7906	136.34	60.40	4929	43.31	0.00	6488	163.50	31.63	0.00
A150.3.5.0.0.5.28	7786	102.80	14.15	6821	43.47	0.00	7384	162.97	8.25	0.00
A150.3.5.0.1.2.12	1287	10.91	84.91	696	17.17	0.00	1051	9.95	51.01	0.00
A150.3.5.0.1.2.24	1834	11.03	31.75	1392	17.70	0.00	1530	10.00	9.91	0.00
A150.3.5.0.1.5.18	2722	71.48	59.55	1706	34.75	0.00	2131	113.20	24.91	0.00
A150.3.5.0.1.5.9	1879	68.16	123.96	839	34.66	0.00	1523	112.63	81.53	0.00
A150.3.5.1.0.2.2	3092	11.64	78.83	1729	19.11	0.00	2613	10.78	51.13	0.00
A150.3.5.1.0.2.4	3375	12.00	59.27	2119	19.22	0.00	2640	10.64	24.59	0.00
A150.3.5.1.0.5.3	3354	56.36	1.88	3896	37.59	18.35	3292	114.22	0.00	0.00
A150.3.5.1.0.5.6	6617	56.09	53.53	4310	37.45	0.00	6431	106.44	49.21	0.00
A150.3.5.1.1.2.3	308	11.84	68.31	183	15.63	0.00	281	9.31	53.55	0.00
A150.3.5.1.1.2.6	470	12.00	12.44	418	15.95	0.00	441	9.48	5.50	0.00
A150.3.5.1.1.5.3	1314	80.75	318.47	314	30.77	0.00	914	112.83	191.08	0.00
A150.3.5.1.1.5.6	2390	57.50	301.01	596	30.64	0.00	1513	99.88	153.86	0.00
A200.1.1.0.0.2.11	11910	15.33	65.81	7183	31.25	0.00	8534	15.11	18.81	0.00
A200.1.1.0.0.2.22	19912	15.56	115.33	9247	31.47	0.00	18715	15.34	102.39	0.00
A200.1.1.0.0.5.10	28462	138.17	108.85	13628	100.44	0.00	22970	248.66	68.55	0.00
A200.1.1.0.0.5.20	33274	136.39	97.03	16888	100.11	0.00	24570	245.69	45.49	0.00
A200.1.1.0.1.2.12	6505	14.25	194.34	2210	26.64	0.00	5259	13.45	137.96	0.00
A200.1.1.0.1.2.6	1950	14.89	66.81	1169	26.70	0.00	1634	13.78	39.78	0.00
A200.1.1.0.1.5.10	25398	136.81	99.15	12753	97.78	0.00	20127	256.97	57.82	0.00
A200.1.1.0.1.5.5	21416	174.13	90.38	11249	97.78	0.00	17625	246.88	56.68	0.00
A200.1.2.0.0.2.16	35096	17.05	160.22	13487	43.02	0.00	34272	17.36	154.11	0.00
A200.1.2.0.0.2.8	32259	16.72	208.55	10455	41.28	0.00	29750	17.17	184.55	0.00
A200.1.2.0.0.5.12	46059	268.77	320.71	10948	68.38	0.00	33025	323.39	201.65	0.00
A200.1.2.0.0.5.24	47130	282.94	199.28	15748	68.09	0.00	36271	318.00	130.32	0.00
A200.1.2.0.1.2.10	26953	16.80	178.38	9682	39.27	0.00	17729	17.13	83.11	0.00
A200.1.2.0.1.2.5	22695	16.84	195.24	7687	39.81	0.00	13235	16.95	72.17	0.00
A200.1.2.0.1.5.13	28519	226.67	223.31	8821	66.77	0.00	13111	305.08	48.63	0.00
A200.1.2.0.1.5.26	31928	277.31	125.42	14164	66.97	0.00	15964	337.08	12.71	0.00
A200.1.3.0.0.2.14	16484	16.25	53.73	10723	30.45	0.00	15977	14.41	49.00	0.00
A200.1.3.0.0.2.7	12471	16.16	31.88	9456	30.61	0.00	12055	14.55	27.49	0.00
A200.1.3.0.0.5.10	16577	268.06	85.24	8949	51.23	0.00	9009	305.55	0.67	0.00
A200.1.3.0.0.5.20	20299	259.11	93.34	10499	51.45	0.00	17254	305.45	64.34	0.00
A200.1.3.0.1.2.11	5820	16.30	262.17	1607	26.09	0.00	3056	14.11	90.17	0.00
A200.1.3.0.1.2.22	6238	16.48	107.66	3004	25.80	0.00	3902	13.64	29.89	0.00
A200.1.3.0.1.5.12	3560	272.48	107.10	1719	44.94	0.00	3061	274.23	78.07	0.00
A200.1.3.0.1.5.24	3976	204.28	56.29	2863	45.00	12.54	2544	241.27	0.00	0.00
A200.1.4.0.0.2.13	15258	15.75	72.29	8856	32.66	0.00	14477	14.73	63.47	0.00
A200.1.4.0.0.2.26	16159	15.64	42.02	11378	32.38	0.00	15231	14.91	33.86	0.00
A200.1.4.0.0.5.14	24732	156.53	209.69	7986	58.67	0.00	15509	237.09	94.20	0.00
A200.1.4.0.0.5.7	18612	148.08	143.52	7643	58.94	0.00	8809	228.88	15.26	0.00
A200.1.4.0.1.2.10	1789	15.67	51.48	1181	23.89	0.00	1671	13.28	41.49	0.00
A200.1.4.0.1.2.20	2593	10.44	29.59	2069	23.77	3.40	2001	10.61	0.00	0.00
A200.1.4.0.1.5.16	4692	147.08	167.50	1754	44.27	0.00	2100	200.58	19.73	0.00
A200.1.4.0.1.5.8	3830	140.69	294.85	970	43.89	0.00	1568	205.11	61.65	0.00
A200.1.5.0.0.2.14	26406	20.14	67.64	15752	42.52	0.00	26600	17.81	68.87	0.00
A200.1.5.0.0.2.7	24678	20.03	95.07	12651	42.50	0.00	24654	17.91	94.88	0.00
A200.1.5.0.0.5.16	24460	255.23	61.57	15139	73.08	0.00	19946	326.14	31.75	0.00
A200.1.5.0.0.5.8	22698	226.64	93.45	11733	73.03	0.00	18357	336.19	56.46	0.00
A200.1.5.0.1.2.12	20498	12.72	59.37	12862	42.30	0.00	20498	14.16	59.37	0.00
A200.1.5.0.1.2.24	29927	21.53	68.41	17770	42.47	0.00	25342	18.23	42.61	0.00
A200.1.5.0.1.5.10	22432	290.36	86.90	12002	71.39	0.00	18169	331.47	51.38	0.00
A200.1.5.0.1.5.20	21149	285.34	31.84	16042	71.02	0.00	18697	324.25	16.55	0.00

Table C.14: Results for instances with 50, 100, 150, and 200 activities-cont.

Dataset	LP			LS			H			H+LS
	OFV	CPU	Gap	OFV	CPU	Gap	OFV	CPU	Gap	Gap
A200.2.1.0.0.2.10	17830	17.30	150.99	7104	40.41	0.00	17837	15.75	151.08	0.00
A200.2.1.0.0.2.20	20932	16.67	116.91	9650	40.08	0.00	20225	15.98	109.59	0.00
A200.2.1.0.0.5.12	17702	115.98	86.67	9483	63.09	0.00	13655	195.25	43.99	0.00
A200.2.1.0.0.5.24	21184	113.45	69.87	12471	63.17	0.00	14906	198.27	19.53	0.00
A200.2.1.0.1.2.16	4810	13.95	73.08	2779	28.95	0.00	4304	13.55	54.88	0.00
A200.2.1.0.1.2.8	3301	13.22	106.44	1599	29.19	0.00	2927	12.98	83.05	0.00
A200.2.1.0.1.5.12	7138	114.03	263.63	1963	48.16	0.00	4417	197.81	125.01	0.00
A200.2.1.0.1.5.6	5206	92.92	425.33	991	48.28	0.00	3367	207.33	239.76	0.00
A200.2.2.0.0.2.14	9887	15.63	97.35	5010	28.06	0.00	9579	14.80	91.20	0.00
A200.2.2.0.0.2.28	10637	15.70	65.30	6435	27.94	0.00	9486	15.08	47.41	0.00
A200.2.2.0.0.5.16	46325	79.72	139.06	19378	96.30	0.00	41544	189.38	114.39	0.00
A200.2.2.0.0.5.8	27562	114.53	61.84	17030	96.14	0.00	25612	215.59	50.39	0.00
A200.2.2.0.1.2.10	4128	14.13	251.92	1173	25.75	0.00	3143	14.44	167.95	0.00
A200.2.2.0.1.2.20	4830	14.47	129.67	2103	25.48	0.00	3872	14.20	84.12	0.00
A200.2.2.0.1.5.12	19009	72.44	220.02	5940	86.53	0.00	14405	179.86	142.51	0.00
A200.2.2.0.1.5.6	7727	68.78	90.51	4056	86.77	0.00	7717	183.09	90.26	0.00
A200.2.3.0.0.2.14	20288	8.89	144.35	8303	33.88	0.00	20706	10.61	149.38	0.00
A200.2.3.0.0.2.28	21954	8.95	86.45	11775	33.95	0.00	21954	10.53	86.45	0.00
A200.2.3.0.0.5.12	18305	157.70	161.09	7011	56.36	0.00	15700	215.75	123.93	0.00
A200.2.3.0.0.5.24	20210	141.97	102.12	9999	56.06	0.00	17512	213.95	75.14	0.00
A200.2.3.0.1.2.10	4483	8.17	90.20	2357	29.23	0.00	4397	9.73	86.55	0.00
A200.2.3.0.1.2.20	5504	7.83	35.00	4077	28.83	0.00	5549	9.98	36.10	0.00
A200.2.3.0.1.5.10	5601	120.59	118.96	2558	49.73	0.00	4316	198.58	68.73	0.00
A200.2.3.0.1.5.5	5425	119.14	293.12	1380	49.81	0.00	4072	191.19	195.07	0.00
A200.2.4.0.0.2.13	11495	14.45	174.21	4192	30.34	0.00	10359	13.98	147.11	0.00
A200.2.4.0.0.2.26	13501	11.05	126.56	5959	30.41	0.00	12108	11.69	103.19	0.00
A200.2.4.0.0.5.11	26086	75.05	153.51	10290	97.56	0.00	24419	188.34	137.31	0.00
A200.2.4.0.0.5.22	28742	72.39	105.87	13961	96.25	0.00	25437	187.17	82.20	0.00
A200.2.4.0.1.2.10	3740	9.44	317.88	895	25.42	0.00	2858	9.95	219.33	0.00
A200.2.4.0.1.2.20	4391	9.38	163.09	1669	25.53	0.00	3079	10.19	84.48	0.00
A200.2.4.0.1.5.16	10588	79.48	158.75	4092	75.52	0.00	7162	180.28	75.02	0.00
A200.2.4.0.1.5.8	9142	70.47	219.65	2860	75.84	0.00	5896	173.73	106.15	0.00
A200.2.5.0.0.2.12	21516	17.39	168.31	8019	41.06	0.00	20020	16.59	149.66	0.00
A200.2.5.0.0.2.6	10064	11.28	42.75	7050	41.28	0.00	8875	13.28	25.89	0.00
A200.2.5.0.0.5.12	37669	217.72	283.40	9825	77.66	0.00	23471	309.39	138.89	0.00
A200.2.5.0.0.5.24	40359	228.44	217.94	12694	77.31	0.00	24717	312.88	94.71	0.00
A200.2.5.0.1.2.10	6691	16.16	137.61	2816	35.20	0.00	6460	15.67	129.40	0.00
A200.2.5.0.1.2.20	7795	16.83	79.40	4345	35.22	0.00	7593	16.08	74.75	0.00
A200.2.5.0.1.5.14	21261	219.47	435.81	3968	67.36	0.00	5582	308.94	40.68	0.00
A200.2.5.0.1.5.28	25328	216.59	251.73	7277	66.38	1.06	7201	304.00	0.00	0.00
A200.3.1.0.0.2.10	8687	9.25	251.27	2473	32.81	0.00	8458	11.06	242.01	0.00
A200.3.1.0.0.2.5	3022	9.08	64.42	1838	32.58	0.00	2533	10.97	37.81	0.00
A200.3.1.0.0.5.16	17855	108.09	288.07	4601	82.13	0.00	15432	211.05	235.41	0.00
A200.3.1.0.0.5.8	18789	110.02	431.36	3536	82.06	0.00	14841	218.98	319.71	0.00
A200.3.1.0.1.2.11	5318	8.38	254.77	1499	29.03	0.00	4805	10.59	220.55	0.00
A200.3.1.0.1.2.22	6523	8.41	120.37	2960	28.92	0.00	6000	10.72	102.70	0.00
A200.3.1.0.1.5.10	10003	109.84	362.25	2164	70.59	0.00	5363	214.08	147.83	0.00
A200.3.1.0.1.5.5	7158	99.84	459.66	1279	70.73	0.00	3384	218.78	164.58	0.00
A200.3.1.1.0.2.3	4971	8.80	110.19	2365	29.38	0.00	4312	10.42	82.33	0.00
A200.3.1.1.0.2.6	5682	8.75	38.75	4095	29.50	0.00	5135	10.27	25.40	0.00
A200.3.1.1.0.5.4	5812	112.61	140.26	2419	75.41	0.00	5986	210.91	147.46	0.00
A200.3.1.1.0.5.8	6484	113.09	109.23	3099	75.00	0.00	6350	210.38	104.90	0.00
A200.3.1.1.1.2.3	1371	8.13	79.45	764	26.11	0.00	1783	9.88	133.38	0.00
A200.3.1.1.1.2.6	1791	8.28	52.82	1172	26.25	0.00	2367	9.98	101.96	0.00
A200.3.1.1.1.5.3	4616	113.86	420.99	886	60.75	0.00	2659	202.97	200.11	0.00
A200.3.1.1.1.5.6	3993	109.19	180.01	1426	61.09	0.00	2956	203.02	107.29	0.00
A200.3.2.0.0.2.14	5169	15.42	67.17	3092	28.53	0.00	5117	15.38	65.49	0.00
A200.3.2.0.0.2.28	6172	12.56	46.53	4212	28.44	0.00	5981	14.25	42.00	0.00
A200.3.2.0.0.5.14	20751	154.48	266.30	5665	87.88	0.00	17199	276.67	203.60	0.00
A200.3.2.0.0.5.7	18576	160.70	323.53	4386	88.16	0.00	11950	276.86	172.46	0.00

Table C.15: Results for instances with 50, 100, 150, and 200 activities-cont.

Dataset	LP			LS			H			H+LS
	OFV	CPU	Gap	OFV	CPU	Gap	OFV	CPU	Gap	Gap
A200.3.2.0.1.2.10	2168	14.45	202.79	716	24.92	0.00	1719	15.11	140.08	0.00
A200.3.2.0.1.2.20	2804	14.44	96.63	1426	25.09	0.00	1952	14.36	36.89	0.00
A200.3.2.0.1.5.14	13640	154.50	346.48	3055	78.41	0.00	8445	263.02	176.43	0.00
A200.3.2.0.1.5.28	13865	171.94	147.02	5613	78.45	0.00	9706	262.19	72.92	0.00
A200.3.2.1.0.2.3	3537	15.61	17.27	3016	26.00	0.00	3499	15.42	16.01	0.00
A200.3.2.1.0.2.6	3953	15.45	18.89	3325	26.27	0.00	3870	14.56	16.39	0.00
A200.3.2.1.0.5.4	13831	132.94	151.70	5495	76.28	0.00	12955	239.77	135.76	0.00
A200.3.2.1.0.5.8	14442	139.92	132.07	6223	77.61	0.00	13321	241.20	114.06	0.00
A200.3.2.1.1.2.3	882	10.81	292.00	225	23.14	0.00	836	12.20	271.56	0.00
A200.3.2.1.1.2.6	737	14.52	67.12	441	23.05	0.00	874	14.95	98.19	0.00
A200.3.2.1.1.5.3	8795	145.20	293.86	2233	66.19	0.00	5889	258.48	163.73	0.00
A200.3.2.1.1.5.6	9787	145.88	267.79	2661	65.67	0.00	7496	253.34	181.70	0.00
A200.3.3.0.0.2.12	47127	14.33	739.16	5616	60.38	0.00	46067	17.63	720.28	0.00
A200.3.3.0.0.2.6	24405	14.08	699.12	3054	60.55	0.00	22955	17.36	651.64	0.00
A200.3.3.0.0.5.12	50422	98.05	706.62	6251	87.30	0.00	45213	196.66	623.29	0.00
A200.3.3.0.0.5.6	18357	98.89	395.20	3707	87.39	0.00	13442	201.88	262.61	0.00
A200.3.3.0.1.2.12	41874	14.52	505.55	6915	59.47	0.00	36538	17.89	428.39	0.00
A200.3.3.0.1.2.6	21226	14.67	391.00	4323	59.59	0.00	20438	17.83	372.77	0.00
A200.3.3.0.1.5.12	49474	89.45	381.78	10269	86.81	0.00	46742	189.33	355.18	0.00
A200.3.3.0.1.5.6	19745	92.09	149.46	7915	86.78	0.00	11206	188.92	41.58	0.00
A200.3.3.1.0.2.3	20761	13.66	439.53	3848	50.20	0.00	20224	16.39	425.57	0.00
A200.3.3.1.0.2.6	21512	13.81	87.84	11452	50.75	0.00	21290	16.09	85.91	0.00
A200.3.3.1.0.5.3	20535	92.22	383.52	4247	71.27	0.00	16519	184.64	288.96	0.00
A200.3.3.1.0.5.6	21987	88.77	296.66	5543	71.78	0.00	17947	182.53	223.78	0.00
A200.3.3.1.1.2.3	15620	13.69	157.63	6063	50.27	0.00	13871	15.94	128.78	0.00
A200.3.3.1.1.2.6	17758	13.66	140.33	7389	50.11	0.00	16140	15.89	118.43	0.00
A200.3.3.1.1.5.3	24340	121.11	316.71	5841	73.59	0.00	20014	203.58	242.65	0.00
A200.3.3.1.1.5.6	28788	112.13	105.80	13988	74.66	0.00	23713	193.78	69.52	0.00
A200.3.4.0.0.2.10	27613	10.86	512.67	4507	62.63	0.00	25447	15.72	464.61	0.00
A200.3.4.0.0.2.5	19411	10.63	598.99	2777	62.59	0.00	17825	15.27	541.88	0.00
A200.3.4.0.0.5.16	47365	104.47	583.18	6933	103.09	0.00	25775	233.25	271.77	0.00
A200.3.4.0.0.5.8	41127	98.98	951.04	3913	103.13	0.00	21872	244.22	458.96	0.00
A200.3.4.0.1.2.16	17324	10.33	119.76	7883	56.22	0.00	15607	15.16	97.98	0.00
A200.3.4.0.1.2.8	14217	9.91	216.99	4485	56.58	0.00	14215	14.91	216.95	0.00
A200.3.4.0.1.5.16	41804	96.09	363.25	9024	92.13	0.00	20851	226.23	131.06	0.00
A200.3.4.0.1.5.8	40980	102.13	375.85	8612	92.94	0.00	17887	221.73	107.70	0.00
A200.3.4.1.0.2.2	19425	10.38	541.94	3026	49.20	0.00	17671	14.14	483.97	0.00
A200.3.4.1.0.2.4	23381	10.38	526.50	3732	49.80	0.00	23381	14.13	526.50	0.00
A200.3.4.1.0.5.3	40865	108.25	502.02	6788	88.14	0.00	29732	224.33	338.01	0.00
A200.3.4.1.0.5.6	54399	111.63	310.62	13248	89.34	0.00	36179	225.67	173.09	0.00
A200.3.4.1.1.2.3	18367	9.41	201.35	6095	43.73	0.00	18367	13.19	201.35	0.00
A200.3.4.1.1.2.6	21237	9.61	180.13	7581	44.05	0.00	21237	13.05	180.13	0.00
A200.3.4.1.1.5.2	30145	100.00	511.96	4926	80.30	0.00	15309	215.55	210.78	0.00
A200.3.4.1.1.5.4	31688	104.23	452.63	5734	81.19	0.00	15765	220.63	174.94	0.00
A200.3.5.0.0.2.14	5602	8.73	89.07	2963	25.84	0.00	5516	10.11	86.16	0.00
A200.3.5.0.0.2.7	3589	8.64	73.63	2067	25.86	0.00	3547	9.88	71.60	0.00
A200.3.5.0.0.5.10	9419	108.33	304.60	2328	57.73	0.00	9261	200.77	297.81	0.00
A200.3.5.0.0.5.20	10707	149.86	167.81	3998	57.69	0.00	10614	234.11	165.48	0.00
A200.3.5.0.1.2.14	1049	7.55	15.27	910	21.13	0.00	915	8.42	0.55	0.00
A200.3.5.0.1.2.28	1980	7.53	8.79	1820	21.13	0.00	1825	8.53	0.27	0.00
A200.3.5.0.1.5.14	2234	119.27	70.14	1313	43.06	0.00	1735	192.02	32.14	0.00
A200.3.5.0.1.5.7	970	92.28	46.53	662	43.16	0.00	669	173.84	1.06	0.00
A200.3.5.1.0.2.1	1435	8.16	306.52	353	22.75	0.00	1435	9.34	306.52	0.00
A200.3.5.1.0.2.2	2703	7.97	459.63	483	22.88	0.00	2492	9.27	415.94	0.00
A200.3.5.1.0.5.1	2823	134.42	61.96	1743	50.56	0.00	2757	210.30	58.18	0.00
A200.3.5.1.0.5.2	3815	140.53	63.24	2337	50.53	0.00	3790	208.98	62.17	0.00
A200.3.5.1.1.2.1	70	7.44	7.69	65	20.03	0.00	65	6.95	0.00	0.00
A200.3.5.1.1.2.2	140	7.44	7.69	130	19.78	0.00	130	6.86	0.00	0.00
A200.3.5.1.1.5.1	444	80.84	367.37	95	38.77	0.00	108	151.95	13.68	0.00
A200.3.5.1.1.5.2	552	81.48	190.53	190	38.86	0.00	200	149.09	5.26	0.00