MINIMIZING THE TOTAL COMPLETION TIME IN A TWO STAGE FLOW SHOP WITH A SINGLE SETUP SERVER

A THESİS

SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL ENGINEERING

AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Muhammet KOLAY

July, 2012

ABSTRACT

MINIMIZING THE TOTAL COMPLETION TIME IN A TWO STAGE FLOW SHOP WITH A SINGLE SETUP SERVER

Muhammet KOLAY

M.S. in Industrial Engineering

Supervisor: Prof.Dr.Ülkü Gürler

Supervisor: Assoc. Prof.Dr.Mehmet Rüştü Taner

July, 2012

In this thesis, we study a two stage flow shop problem with a single server. All jobs are available for processing at time zero. Processing of a job is preceded by a sequence independent setup operation on both machines. The setup and processing times of all jobs on the two machines are given. All setups are performed by the same server who can perform one setup at a time. Setups cannot be performed simultaneously with job processing on the same machine. Once the setup is completed for a job, processing can automatically progress without any further need for the server. Setup for a job may start on the second machine before that job finishes its processing on the first machine. Preemption of setup or processing operations is not allowed. A job is completed when it finishes processing on the second machine. The objective is to schedule the setup and processing operations on the two machines in such a way that the total completion time is minimized. This problem is known to be strongly NP-hard [3]. We propose a new mixed integer programming formulation for small-sized instances and a Variable Neighborhood Search (VNS) mechanism for larger problems. We also develop several lower bounds to help assess the quality of heuristic solutions on large instances for which optimum solutions are not available. Experimental results indicate that the proposed heuristic provides reasonably effective solutions in a variety of instances and it is very efficient in terms of computational requirements.

Keywords: machine scheduling, flow shop, single setup server, total completion time, variable neighborhood search.

ÖZET

TEK SUNUCULU İKİ AŞAMALI SERİ AKIŞDA TOPLAM TAMAMLANMA ZAMANINI EN AZLAMAK

Muhammet Kolay

Endüstri Mühendisliği, Yüksek Lisans

Tez Yöneticisi: Prof. Dr. Ülkü Gürler

Tez Yöneticisi: Doç. Dr. Mehmet Rüştü Taner

Temmuz 2012

Bu tez kapsamında; tek sunuculu iki aşamalı seri akış problemi çalışılmıştır. Tüm işler sıfır zamanında işlem görmek üzere hazırdırlar. Her iki makine üzerinde, bir iş için işleme başlamadan önce sıradan bağımsız olarak hazırlık işlemleri yapılmaktadır. Bütün işler için, işlem süreleri ve hazırlık süreleri verilmiştir. Tüm hazırlık işlemleri, tek seferde sadece bir iş için hazırlık yapabilen bir sunucu tarafından yapılmaktadır. Aynı makine üzerinde bir işe ait işlem operasyonu devam ederken aynı anda hazırlık işlemi yapılamamaktadır. Bir işin hazırlığı tamamlandığı zaman, işlem operasyonu sunucuya ihtiyaç duyulmadan otomatik olarak yapılabilmektedir. Bir işin ikinci makinede hazırlanmasına, aynı işin birinci makinedeki işlemi devam ederken başlanabilmektedir. Hazırlık sürelerinin ve işlem sürelerinin bölünerek yapılmasına müsaade edilmemektedir. Bir iş, ikinci makinedeki işlemi bittiğinde tamamlanmış olarak kabul edilir. Problemin amacı verilen tüm işleri iki makine üzerinde toplam tamamlanma zamanlarını en azlayacak şekilde çizelgelemektir. Bu problem NP-zor bir problemdir. Küçük boyutlu problemleri çözmek için karışık tam sayılı programlama modeli, büyük problemleri çözebilmek için ise sezgisel Değişken Komşu Arama mekanizması önerilmiştir. Ayrıca, sezgisel algoritmaların performansını en iyi sonuçların elde edilemediği büyük problemlerde değerlendirebilmek amacıyla, alt sınırlar geliştirilmiştir. Yapılan deneylerin sonucunda; önerilen sezgisel algoritmaların farklı örnek çeşitlerinde etkili sonuçlar verdiği ve hesaplanabilirlik açısından çok etkili oldukları görülmüştür.

Anahtar sözcükler: makine çizelgeleme, seri akış, tek hazırlık sunucusu, toplam tamamlanma zamanı, değişken komşu arama.

Acknowledgement

Foremost, I would like to express my deepest and most sincere gratitude to my advisor Assoc. Prof. Dr. Mehmet Rüştü Taner for his invaluable guidance, encouragement and motivation during my graduate study. I could not have imagined having better mentor for my M.S study.

I would like to express my sincere thanks to Professor Ulku Gurler for accepting to serve as one of my supervisors in the last few months of my study and for her insightful comments that helped to improve the readability of this thesis.

I am also grateful to Assoc. Prof. Dr. Oya Ekin Karaşan and Asst. Prof. Dr. Sinan Gürel for accepting to read and review this thesis and for their invaluable suggestions.

I am indebted to Selva Şelfun for her incredible support, patience and understanding throughout my M.S study.

Many thanks to Betül Kolay, Ahmet Kolay, Kamil Kolay and Murat Kolay for their moral support and help during my graduate study. I am lucky to have them as a part of my family.

Last but not the least; I would like to thank to my lovely family. The constant inspiration and guidance kept me focused and motivated. I am grateful to my dad Mustafa Kolay for giving me the life I ever dreamed. I can't express my gratitude for my mom Niğmet Kolay in words, whose unconditional love has been my greatest strength. The constant love and supports of my brother Alper Kolay and my sister İrem Tuana Kolay are sincerely acknowledged.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In many manufacturing facilities and assembly lines, each job has to go through a series of operations. Mostly, these operations have to be done in the same order for all jobs which implies that, each job gets to be processed on every machine in the same order. Flow shop problems aim to schedule a given set of jobs on a number of machines in such a system so as to optimize a given criterion. Most researchers assume that, setup times can be included in the processing times or ignored entirely. Even though this assumption may be valid for some cases, there are still many applications where an explicit consideration of setup times is necessary. Some example cases in which setup times may significantly affect the operational planning activities are those involving tool change, die change, clean up, loading and unloading operations. .

Additionally in real life, most systems work semi-automatically, and this requires involvement of a server before or after processing operations. This may be the case in certain flow shop applications also. In these applications, setup operation for a job may be carried out before that job finishes its previous processing and arrives at the current machine. To avoid high labor expenses, a single server may be in charge of performing setups on multiple machines.

The main goal of our problem is to minimize the total completion time of all jobs. A flow shop environment is a good design for producing similar items in large quantities to stock. But usually, there tends to be little space along the production

line for work=in=process storage. This suggests that minimizing the total completion time is a particularly relevant objective in such a setting.

Differently from classical two stage flow shop, in our problem immediately before processing an operation, machine has to be prepared by the server for the operation, which is called setup. During setup, both the machine and the server will be occupied. All setups should be done by the single server who can perform at most one setup at a time. Server is responsible only for the setup operation, once the setup is completed, process is executed automatically without the server. A job is completed when the processing on the second machine is finished. The goal of the problem is to schedule the given set of jobs in order to minimize the total completion time.

Although there appeared some studies in the literature on parallel machine scheduling problems with a single server, the flow shop versions of these problems received much less attention from the research community. Lim et al. [1] consider single server in a two machine flow shop for the makespan objective. Although the classical two machine flow shop problem with the makespan objective is polynomially solvable [2], the version of the problem with a single setup server is known to be unary NP-Hard [3]. Lim et al. propose a number of heuristics to gain near optimal solutions for the problem with the makespan objective.

This thesis focuses on the total completion time objective in a two machine flow shop with a single setup server. Chapter 2 reviews the literature on scheduling problems with a single server. After presenting a formal definition of the problem at hand, a mixed integer programming model is proposed in Chapter 3 to solve small instances to optimality. Then, Chapter 4 presents the proposed heuristic algorithms to solve larger instances that cannot be solved via exact methods due to the curse of dimensionality. We also develop several lower bounds in this same chapter to help assess the quality of heuristic solutions on instances for which optimum solutions cannot be obtained. Chapter 5 presents the results of our computational study through which the proposed algorithms and lower bounds are tested. Finally, Chapter 6 concludes the thesis with a summary of the major findings and directions for possible future studies.

Chapter 2

Literature Review

In many manufacturing and assembly facilities each job has to undergo a series of operations. Often, these operations have to be done on all jobs in the same order implying that the jobs have to follow the same route. The machines are then assumed to be set up in series and the environment is referred to as a flow shop [4].

After publication of Johnson's [2] classical paper, in which he proved that the two stage flow-shop problem with the makespan objective can be solved in polynomial time, many researchers focused on flow-shop problems. However, makespan is still the only objective function for which the two stage flow shop problem can be solved in polynomial time. For the classical flow-shop problem, Garey [5] shows that the problem is NP-hard with the total completion time as the objective criterion.

There are some studies in the literature that have an explicit consideration of setups. Setup times are classified as separable and non-separable. When setup times are separable from the processing times, they could be anticipatory (or detached). In the case of anticipatory (detached) setups, the setup of the next job can start as soon as a machine becomes free to process the job since the shop floor control system can identify the next job in the sequence. In such a situation, the idle time of a machine can be used to complete the setup of a job on a specific machine. In other situations, setup times are non-anticipatory (or attached) and the setup

operations can start only when the job arrives at a machine as the setup is attached to the job [6]. Setup times are also classified as sequence dependent and sequence independent. Setups are called sequence dependent, when the setup times change as a function of the job order. There is extensive literature on scheduling with setup times excellent reviews of which can be found in [6], [7] and [8].

In our problem, setup times are anticipatory and sequence independent. Additionally, a single setup server is responsible to carry out the setup operation. There are a number of studies on scheduling problems with a single server. Most of them consider parallel machine and flow shop problems. Although our study focuses on a two machine flow shop, we also present a summary of some major findings for the parallel machine environment with a single server.

Koulamas [9] studies two parallel machines with a single server to minimize the machine idle time resulting from the unavailability of the server. He shows that this problem is NP-hard in the strong sense and proposes an efficient beam search heuristic.

In the study of Kravchenko and Werner [10] they consider the problem of scheduling jobs on parallel machines with a single server to minimize a makespan objective. They present a pseudo polynomial algorithm for the case of two machines when all setup times are equal to one. They also show that the more general problem with an arbitrary number of machines is unary NP-hard.

Hall et al. [11] present complexity results for the special cases of the parallel machine scheduling with a single server for different objectives. For each problem considered, they provide either a polynomial or pseudo-polynomial-time algorithm, or a proof of binary or unary NP-hardness. Brucker et al. [12] also derive new complexity results for the same problem in addition to [10] and [11].

Abdekhodaee and Wirth [13] study scheduling parallel machines with a single server with the makespan minimization problem for the special cases of equal processing and equal setup times. They show that both of the special cases are NP-

hard in the ordinary sense, and they construct a tight lower bound and two highly effective O(*n logn*) heuristics.

In the work of Glass et al. [14], scheduling for parallel dedicated machines with a single server is studied. The machines are dedicated in the sense that each machine processes its own set of pre-assigned jobs. They show that for the objective of makespan, this problem is NP-hard in the strong sense even when the setup times or the processing times are all equal. The authors propose a simple greedy algorithm which creates a schedule with a makespan that is at most twice the optimal value. Additionally, for the two machine case, their improved heuristic guarantees a tighter worst-case ratio of 3/2.

Kravchenko and Werner [15] consider the scheduling on *m* identical parallel machines with a single server for the objective that minimizes the sum of the completion times in the case of unit setup times and arbitrary processing times. Since the problem is NP-hard, they propose a heuristic algorithm with an absolute error bounded by the product of the number of short jobs (with processing times less than m−1 and m−2).

Wang and Cheng [16] study the scheduling on parallel several identical machines with a common server to minimize the total weighted job completion times. They propose an approximation algorithm and analyze the worst case performance of the algorithm.

Abdekhodaee and Wirth [17] consider two parallel machines with a single server with the makespan objective. In their work, an integer programming formulation is developed to solve problems with up to 12 jobs. Also they show that two special cases which are short processing times and equal length jobs can be solved in polynomial time. Heuristics are presented for the general case. Results of the heuristics are compared with defined simple lower bounds over a range of problems. The same authors consider this same problem again in another paper [18] and this time they propose the use of Gilmore Gomory algorithm, a greedy heuristic and a genetic algorithm to solve the general case of the problem.

In regard to the flow shop problems, Yoshida and Hitomi [19] show that the two-stage flow-shop problem can be solved in polynomial time when there is sufficiently many servers. Brucker [4] proves that the version of this problem with a single server is NP-hard in the strong sense. He also presents new complexity results for the special cases of flow-shop scheduling problems with a single server.

Cheng et al. [20] study the problem of scheduling *n* jobs in a one-operator two-machine flow-shop to minimize the makespan. In their problem, besides setup operations, the so called dismounting operations are also considered. After the machine finishes processing a job, the operator needs to perform a dismounting operation by removing that job from the machine before setting up the machine for another job. Furthermore, they assume that the setup server moves between the two machines according to same cyclic pattern. They analyze the problem under the two cases of separable and non-separable setup and removal times. Problems in both of these cases are proved to be NP-hard in the strong sense. Some heuristics are proposed for their solution and their worst-case error bounds are analyzed.

Glass et al. [14] show that the two-stage flow-shop problem with a single server is NP-hard in the strong sense for the makespan objective. Also in the same paper, a no-wait constraint, which forces a job completed in the first stage to be sent immediately to the second stage, is added to the original problem. They reduce the resulting problem to the Gilmore-Gomory traveling salesman problem and solve it in polynomial time.

In the work of Cheng et al. [21], both processing and setup operations are performed by the single server. Additionally a machine dependent setup time is needed whenever the server switches from one machine to the other. They observe that; scheduling single server in a two machine flow-shop problem with a given job sequence can be reduced to a single machine batching problem for many regular performance criteria. Additionally, they solve the problem with agreeable processing times in O(*nlogn*) time for the maximum lateness and total completion time objectives.

Lim et al. [1] study minimizing makespan in a two-machine flow shop with a single server for the special case where all processing times are constant. First, they show that some special cases of this problem are polynomial-time solvable. Then, they present an IP formulation and check effectiveness of the proposed lower bounds against optimal solutions of small instances. Several heuristics are proposed to solve the problem including simulated annealing, tabu search, genetic algorithms, GRASP, and other hybrids. The results on small instances are compared with the optimal solutions, whereas results on large instances are compared to a lower bound. In addition, they remove the constant processing time assumption and consider the general problem also. The same heuristics and lower bounds are applicable with modifications to the general problem. Their proposed heuristics produce close to optimum results in the computational experiments.

For the objective of minimizing the total completion time, Ling-Huey-Su et al. [22] study the single-server two-machine flow-shop problem with a no-wait consideration. Since the problem is NP-hard in the strong sense they propose some heuristics, identify some polynomial solvable special cases, establish some optimality properties for the general case and propose a branch & bound algorithm for its solution. They observe that both their heuristic and exact methods perform better that their existing counterparts.

Chapter 3

Problem Definition

This chapter consists of the detailed formal definition of the problem. We first define our problem and provide an illustrative example. After defining the relevant parameters and variables, we propose a mixed integer programming model that will be instrumental in obtaining optimum solutions for smaller problem instances. Also a special case that can be solved in polynomial time is presented in this chapter.

3.1 Problem Definition

We are given two machines and a set of jobs $j \in N$, each job has two operations which have to be processed on the first and second machines in that order. All jobs are available for processing on the first machine at time zero. Also, before processing a job on machine *i*, setup operation must be done by the server, during which time both the machine and the server will be occupied for "s_{ij}"units. No other job can be processed on that machine while it is under setup.

Since there is only one server in our problem, at most one setup can be carried out on only one machine at any given time. This single server is responsible only for the setup operations. That is, once the setup is completed, job processing takes place automatically without the server. At each time after setup operation, the processing operation must be carried out possibly after some idle time. Setup times and processing times are separable in the sense that a setup for a job on

machine 2 can be performed while that job is being processed on machine 1. Preemption is not allowed for both processing and setup. For example; the processing of any job started at time t on one of the machines will be completed at time $t + p_{ij}$ on the same machine. The goal of the problem is to obtain a schedule which gives the minimum total completion time.

Figure 3.1 illustrates an example of the described problem which considers a problem instance with four jobs. Table 3.1 gives the processing and setup times on machines 1 and 2. The optimum solution to this instance is shown on a Gantt chart in Figure 3.1. The shaded areas represent setups, while the unshaded areas represent processing times. Mark that, there is no overlap of setup between the machines since our problem has only one server. Also, unlike many other flow-shop problems, the optimal solution may be a non-permutation schedule where the processing order of jobs is not the same on machines 1 and 2. Moreover, two or more consecutive setup operations can be done on the same machine, which means that server doesn't have to make setup operations alternately between machines. We can clearly see these characteristics on the given example.

Table 3.1: Data for Given Example

Jobs		1	2	3	4
p	**M1**	3	5	1	2
	M2	18	20	4	10
S	**M1**	2	1	4	1
	M2	5	2	16	3

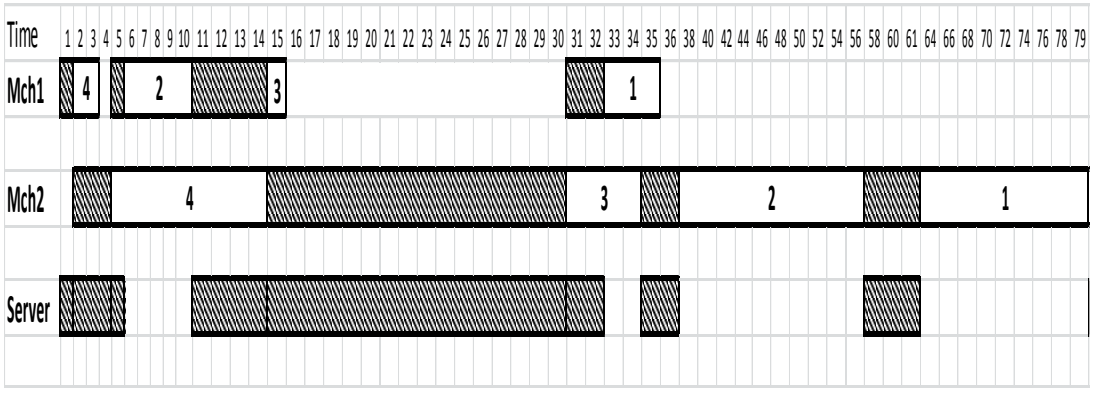

Figure 3.1: Optimal Solution for the Given Example

3.2 The Mixed Integer Programming Model

Lim et al. [1] present a MIP formulation for the same problem with the makespan objective. In their model, they use binary variables to ensure the precedence relations between setup and processing operations. We construct our MIP formulation by using binary variables that keep track of which job will be processed in which order on machines 1 and 2. First we define the parameters and variables of the problem. Then we present a mixed integer programming model.

3.2.1 Parameters

M_i : Represents i^{th} machine, where $i = 1,2$.

N : Number of jobs. $j = \{1, \dots, \mathrm{N}\}$

M : A very large number.

p_{ij} = Process time of job j on machine i .

s_{ij} = Setup time of job j on machine i .

3.2.2 Variables

We use the following two discrete decision variables to keep starting time of the process and setup operations.

X_{ik} = Starting time of processing of the k th job on machine i

W_{ik} = Starting time of setup of the k th job on machine i

C_l = Completion time of the l th job

To decide, in which positions on machines 1 and 2 job $j \in \mathrm{N}$ should be done, we use the following binary decision variable:

$$y_{klj} = \begin{cases} 1, & \text{if job } j \in \mathrm{N} \text{ processed as the } k^{th} \text{ job on } M_1 \text{ and } l^{th} \text{ job on } M_2 \\ 0, & \text{otherwise} \end{cases}$$

Since the server can carry out at most one setup operation at a time, we use the following binary decision variable to coordinate the setups on the two machines.

$$r_{kl} = \begin{cases} 1, & \text{if the } k^{th} \text{ job on } M_1 \text{ starts its setup before the } l^{th} \text{ job on } M_2 \\ 0, & \text{otherwise} \end{cases}$$

3.2.3 MIP Model

The proposed formulation is as follows,

$$Min \sum_{l} C_l \tag{1}$$

s.t.

$$\sum_{k}\sum_{l} y_{klj} = 1 \qquad \forall\, j \in N \tag{2}$$

$$\sum_{l}\sum_{j} y_{klj} = 1 \qquad \forall\, k \in N \tag{3}$$

$$\sum_{k}\sum_{j} y_{klj} = 1 \qquad \forall\, l \in N \tag{4}$$

$$X_{1k} + \sum_{j} y_{klj}\ p_{1j} \le X_{2l} + M\left(1 - \sum_{j} y_{klj}\right) \quad \forall\, k \in N, \qquad \forall\, l \in N \tag{5}$$

$$W_{1k} + \sum_{l}\sum_{j} y_{klj}\ s_{1j} \le X_{1k} \qquad \forall\, k \in N \tag{6}$$

$$W_{2l} + \sum_{k}\sum_{j} y_{klj}\ s_{2j} \le X_{2l} \qquad \forall\, l \in N \tag{7}$$

$$X_{1k} + \sum_{l}\sum_{j} y_{klj}\ p_{1j} \le W_{1\,(k+1)} \qquad \forall\, k \in N \tag{8}$$

$$X_{2l} + \sum_{k}\sum_{j} y_{klj}\ p_{2j} \le W_{2\,(l+1)} \qquad \forall\, l \in N \tag{9}$$

$$W_{1k} + \sum_{d}\sum_{j} y_{kdj}\, s_{1j} \le W_{2l} + M\,(1 - r_{kl}) \quad \forall\, k \in N, \forall\, l \in N \tag{10}$$

$$W_{2l} + \sum_{d}\sum_{j} y_{dlj}\, s_{2j} \le W_{1k} + M\ r_{kl} \qquad \forall\, k \in N, \forall\, l \in N \tag{11}$$

$$C_l \ge X_{2l} + \sum_{k}\sum_{j} y_{klj}\ p_{2j} \qquad \forall\, l \in N \tag{12}$$

$$X_{ik} \ge 0 \qquad \forall\, i \in M, \qquad \forall\, k \in N \tag{13}$$

$$W_{ik} \ge 0 \qquad \forall\, i \in M, \qquad \forall\, k \in N \tag{14}$$

$$C_l \ge 0 \qquad \forall\, l \in N \tag{15}$$

$$y_{klj} \in \{0,1\} \quad \forall\, k \in N, \qquad \forall\, l \in N, \qquad \forall\, j \in N \tag{16}$$

$$r_{kl} \in \{0,1\} \quad \forall\, k \in N, \qquad \forall\, l \in N \tag{17}$$

The objective function (1) is the minimization of the total completion time.

Constraint sets (2), (3), (4) guarantee that only one job can be processed as the k^{th} job on machine 1 and the l^{th} job on machine 2 and each job can be assigned to only one position on machines 1 and 2.

Constraint set (5) ensures that processing of a job can start on machine 2, only after its processing is finished on machine 1.

Constraint sets (6) and (7) indicate that setup of the job must be finished before starting its process on the same machine.

Constraint sets (8) and (9) ensure that setup of the next job can start after finishing process of the previous job on the same machine.

Constraint sets (10) and (11) prevent setup operations on machine 1 and machine 2 to take place at the same time.

Constraint set (12) computes the completion time of the jobs.

Constraint sets (13), (14), (15), (16) and (17) give the non-negativity and binary restrictions.

3.3 Polynomial-Time Solvable Case

Although minimizing total completion time in a two stage flow shop problem with a single server is NP-hard in the strong sense [3], it is possible to solve a special case of the problem in polynomial time. This special case is the following;

Propositon 1. If all the setup times are equal and $\mathrm{p_{ij}} \leq \mathrm{s_{ij}} = \mathrm{s\ for\ i} = 1, 2$ and $\mathrm{j} \in \mathrm{N}$ then an optimal permutation schedule exists in which the server serves the jobs in ascending order of their processing times on the second machine.

Proof: When all setup times are equal and greater than all the processing times, processing operations can be performed while setup is in progress on the other machine. Hence, the processing operations on the first machine do not affect the solution. Processing operations on the second machine will be done immediately after the respective setup operations. Thus our problem for this special case can be seen as a single machine scheduling problem on the second machine. Since SPT

(Shortest Processing Time) gives optimal schedule for total completion time objective on a single machine, sorting jobs in ascending order of their processing times on the second machine gives the optimal schedule for our problem. Figure 3.2. illustrates this case.

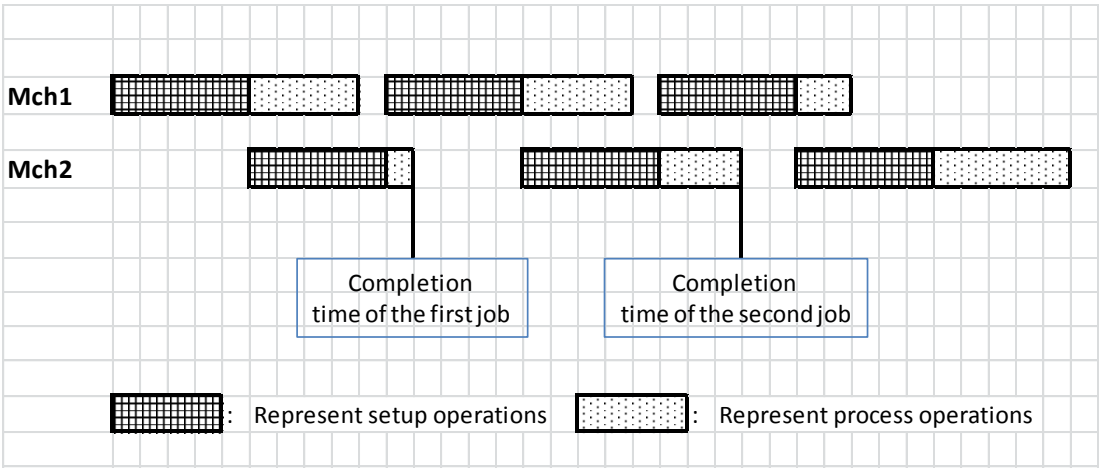

Figure 3.2: Example of an Optimal Schedule for the Special Case.

Chapter 4

Lower Bounds and Heuristic Solution Mechanisms

In this chapter we first present lower bounds to help assess the quality of heuristic solutions on large instances. Then, to aid in solving larger instances a greedy constructive heuristic is proposed to get an initial feasible solution. Later, we introduce representation and validation of the solutions. Lastly, we describe an already existing Variable Neighborhood Search (VNS) Algorithm, discovered search methods and implementation of VNS to our problem.

4.1 Lower Bounds

In this section we develop several lower bounds to help assess the performance of our heuristics in those instances for which exact solutions cannot be obtained in reasonable computational times.

4.1.1 Lower Bound 1

If we relax the capacity of the first machine by assuming that all the jobs are available for processing on the second machine as early as necessary, then we get a relaxed problem which is a single machine scheduling problem where $P_j = s_{2j} + p_{2j}$ for all *j*. Since SPT (Shortest Processing Time) rule gives the optimal solution for

minimizing the total completion time in a single machine, lower bound 1 can be obtained by computing:

$$LB_1 = \sum C_j^2 + N \times min\{s_{1j}\}$$

where $\sum C_j^2$ is the optimal total completion time of the problem defined on machine 2 and obtained by the SPT rule based on the total setup time and processing time of each job. Also, $N \times min\{s_{1j}\}$ is added, because it corresponds to the smallest unavoidable idle time on machine 2.

Since this lower bound mainly considers second machine setup times and processing times, it is expected to gıve good bounds for the instances where setup operations and processing operations on the second machine are much longer than those on the first machine.

To illustrate this lower bound, consider the following toy problem.

Table 4.1: Data for Toy Problem

j	**1**	**2**	**3**	**4**
$\boldsymbol{s_{1j}}$	2	4	7	4
$\boldsymbol{p_{1j}}$	3	5	4	1
$\boldsymbol{s_{2j}}$	5	2	10	1
$\boldsymbol{p_{2j}}$	6	4	4	3

We compute $s_{2j} + p_{2j}$

$\boldsymbol{j}$	**1**	**2**	**3**	**4**
$\boldsymbol{s_{2j} + p_{2j}}$	11	6	14	4

By applying the SPT rule, we obtain the job sequence {4,2,1,3}. So,

$$\sum C_j^2 = (4) + (4+6) + (4+6+11) + (4+6+11+14) = 70$$

Thus, $$LB_1 = 70 + 4 \times 2 = 78$$

4.1.2 Lower Bound 2

No two setups can be carried out simultaneously on the single server. The best case is that the server is continuously busy performing setups until the beginning of the processing of the last job on machine 2. Hence lower bound 2 can be obtained by relaxing the problem to a single machine where $P_{\mathrm{j}} = \mathrm{s}_{1\mathrm{j}} + \mathrm{s}_{2\mathrm{j}}$ for all j. Since completion time is defined as the finishing time of process on second machine, lower bound 2 can be defined as:

$$LB_2 = \sum C_j^{setups} + \sum_{j \in N} p_{2j}$$

where $\sum C_j^{setups}$ is the optimal total completion time of the relaxed single machine problem and obtained by the SPT rule after merging the first machine setup time and the second machine setup time of each job. Also, $\sum_{j \in N} p_{2j}$ is added, because job is completed when process ends on the second machine.

Since we mainly consider setup times while computing this lower bound, it is expected to produce good bounds for the instances in which, setup times are generally greater than processing times on both machines. We apply this lower bound on the toy problem.

We compute $\mathrm{s}_{1\mathrm{j}} + \mathrm{s}_{2\mathrm{j}}$

j	1	2	3	4
$s_{1j} + s_{2j}$	7	6	17	5

By applying the SPT rule, we obtain the job sequence {4,2,1,3}. So,

$$\sum C_j^{setups} = (5) + (5+6) + (5+6+7) + (5+6+7+17) = 69$$

$$\sum_{j \in N} p_{2j} = 6 + 4 + 4 + 3 = 17$$

Thus, $$LB_2 = 69 + 17 = 86$$

4.1.3 Lower Bound 3

The best case on machine 1 is when there is no idle time. Hence, we relax the capacity of the second machine. Then we get a relaxed problem which is a single machine scheduling problem where $P_{\mathrm{j}} = \mathrm{s}_{1\mathrm{j}} + \mathrm{p}_{1\mathrm{j}}$ for all *j*. Lower bound 3 can be obtained by computing:

$$LB_3 = \sum C_j^1 + PIT_2$$

where $\sum C_j^1$ is the optimal total completion time of the problem defined on machine 1 and obtained by the SPT rule based on the total setup time and processing time of each job on machine 1 and PIT_2 is the possible idle times that can occur on the second machine. There are two possible cases that job can be setup on the second machine. In the first case ($s_{2j} < p_{1j}$) and s_{2j} starts as soon as s_{1j} finishes on the first machine. Idle time doesn't occur on the second machine while p_{2j} starts as soon as p_{1j} finishes on the first machine. Differently from first case, in the second case ($s_{2j} > p_{1j}$) which causes ($s_{2j} - p_{1j}$) unit idle time on the second machine because p_{2j} can be started on the second machine after s_{2j} is completed. Figure 4.1. illustrates these cases in which, it can be seen that idle time occurs at least $(s_{2j} - p_{1j})^+$ time units for each job. Total completion time objective grows cumulatively hence any idle time for job j affects also all the jobs which are processed after$j$. Therefore, we consider small idle times early and large idle times late as much as possible on the schedule. So, jobs are sorted in ascending order of $(s_{2j} - p_{1j})^+$ values and multiplied with $N - ord(j)$ where $ord(j)$ gives the order of job *j* in ascending orders.

This lower bound mainly considers setup times and processing times on the first machine. Hence for the instances that have larger processing and setup operations on the first machine, it is expected to give good bounds. However, effect of the idle times that may be experienced on the second machine is also considered as much as possible. Thus, this lower bound is expected to work well for the instances in which average of setup times and average of processing times are near each other on machines 1 and 2.

Consider the toy problem to illustrate LB 3:

We compute $s_{1j} + p_{1j}$ and $(s_{2j} - p_{1j})^+$

j	1	2	3	4
$s_{1j} + p_{1j}$	5	9	11	5
$(s_{2j} - p_{1j})^+$	2	0	6	0

By applying the SPT rule, we obtain the job sequence {4, 1, 2, 3}. So,

$$\sum C_j^2 = (5) + (5+5) + (5+5+9) + (5+5+9+11) = 64$$

Ascending order of $(s_{2j} - p_{1j})^+$ is $\{2,4,1,3\}$. So,

$$PIT_2 = (4 \times 0) + (3 \times 0) + (2 \times 2) + (1 \times 6) = 10$$

Thus, $LB_3 = 64 + 10 = 74$

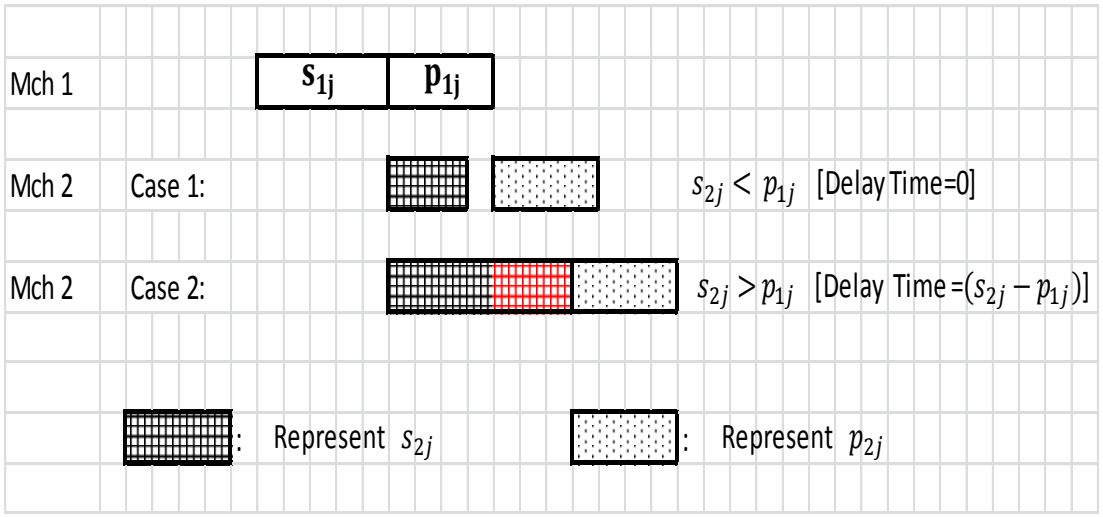

Figure 4.1: Possible Idle Times for the LB 3

4.1.4 Lower Bound 4 and Lower Bound 5

These lower bounds are improved versions of lower bound 3. First we transform the problem in to the classical flow shop problem (i.e., one without setups) with the total completion time objective in such a way that the optimum objective function value of the transformed problem will always be smaller than that of the original problem. In order to make this transformation, we redefine the processing time of each job j on machine 1 as P'_{1j} where $P'_{1j} = s_{1j} + p_{1j}$. We then completely

disregard the setup operations on the second machine and consider only the processing times p_{2j} for $j = 1, .., \mathrm{N}$. Now the problem is transformed into the classical problem any lower bound for which will also be a lower bound for our original problem.

After this transformation, we use the lower bound developed by Han Hoogeveen et al. [23]. They formulate the classical two stage flow shop problem as an integer program (IP) using positional completion time variables. They show that solving the linear programming relaxation of their IP model gives a very strong lower bound. We use both their integer programming model and its LP relaxation to obtain lower bounds for our problem. Model is as follows.

$$Min \sum_{i=1}^{n} n\, a_i\, x_{i1} + \sum_{i=1}^{n}\sum_{j=1}^{n}(n+1-j) b_i\, x_{ij} + \sum_{j=2}^{n}(n-j+1)\, I_j \qquad (1)$$

s.t.

$$\sum_{j=1}^{n} x_{ij} \geq 1, \quad for\ i = 1, \ldots\ldots, n \qquad (2)$$

$$\sum_{i=1}^{n} x_{ij} \leq 1, \quad for\ j = 1, \ldots\ldots, n \qquad (3)$$

$$\sum_{i=1}^{n}\sum_{j=1}^{k-1} b_i x_{ij} - \sum_{i=1}^{n}\sum_{j=2}^{k} a_i x_{ij} + \sum_{j=2}^{k} I_j \geq 0, \quad for\ k = 2, \ldots, n \qquad (4)$$

$$x_{ij} \in \{0,1\} \quad for\ i = 1, \ldots., n\ and\ j = 1, \ldots., n \qquad (5)$$

$$I_j \geq 0, \quad for\ j = 2, \ldots., n \qquad (6)$$

where a_i represents the first machine process, b_i represents the second machine process, I_j is the slack variable and x_{ij} is the binary variable, which takes on a value of 1 if job i is assigned to position j, and it is 0 otherwise. Constraint (1) is the objective function which tries to minimize total completion time of the jobs. Constraint sets (2) and (3) ensure that only one job can be assigned to one position. Constraint (4) satisfies the machine capacity conditions. Finally, constraint sets (5)

and (6) give the non-negativity and binary restrictions. By solving this IP model, lower bound 4 can be obtained.

Lower bound 5 is obtained by solving the version of the problem that relaxes the binary constraints to $x_{ij} \geq 0$ and $x_{ij} \leq 1$. Hence for the instances up to 35 jobs lower bound 5 is not computed as it can never be better than lower bound 4. However, it is not possible to obtain lower bound 4 for the N=50 and N=100 instances in reasonable computational times. Thus, lower bound 5 is computed instead for these instances.

Lower bounds 4 and 5 ignore the single setup server and setup operations on the second machine. They rather focus on only the job processing where processing times on the first machine are adjusted to include also the respective setup times. Since effects of the single setup server and setup times on the second machine are ignored, it is expected to give good bounds for the instances where processing times are generally greater than setup times on both machines.

4.2 Heuristic Solution Mechanisms

In this section we propose a constructive method to obtain an initial solution followed by two versions of a VNS mechanism that are intended to improve this starting solution.

4.2.1 Greedy Constructive Heuristic

This heuristic aims to produce a full schedule by inserting one job at a time, into a partial schedule in a greedy manner. It enforces a permutation schedule in the sense that jobs are processed in the same order on machines 1 and 2. In each step, each unscheduled job is considered as a candidate for the next position. The candidate that increases objective function by the minimum value is selected from the set.

Define t_1 as the starting time of the next setup on first machine and t_2 as the finishing time of the last process on second machine for a given partial schedule. The details of the algorithm are as presented in the following.

Algorithm (Greedy Constructive Heuristic)

Step 1. *(Initialization)*

Select the job with the maximum total completion time on the second machine as the first job in the schedule.

Place $k = argmin_{j \in N} \{ s_{1j} + max \{s_{2j} , p_{1j}\} + p_{2j}\}$

$set\ t_1 \leftarrow\ s_{1k} + max \{s_{2k} , p_{1k}\}$

$set\ t_2 \leftarrow\ t_1 + p_{2k}$

$set\ N \leftarrow N/\{k\}$

Step 2. *(Analysis of Remaining Jobs to be scheduled)*

For each job that has not yet been scheduled do the following: Consider that job as the next one in the partial sequence and compute the total completion time of all the jobs scheduled until that point.

Step 3. *(Selection of the Next Job in Partial Schedule)*

Of all jobs analyzed in Step 2, select the job with the smallest total completion time as the next one in the partial sequence.

Place $k = argmin_{j \in N} \{max\{t_1 + s_{1j} + p_{1j} , max\{t_1 + s_{1j} ,\ t_2\} + s_{2j}\} + p_{2j}\}$

$set\ t_1 \leftarrow\ max\{t_1 + s_{1k} + p_{1k} , max\{t_1 + s_{1k} ,\ t_2\} + s_{2k}\}$

$set\ t_2 \leftarrow\ t_1 + p_{2k}$

$set\ N \leftarrow N/\{k\}$

Step 4. *(Stopping Criterion)*

If $N = \emptyset$, *STOP. Else, Go to Step 2.*

4.2.2 Representation and Validation of Solutions

We represent a solution as an integer array of length $2 \times$ N, where N is the number of jobs. Each entry of the array indicates setup of a job on machines 1 and 2. For any job *j*, its setup on the first machine is represented as *j* while that on the second machine is represented as $j +$ N. Since processing has to take place as soon as possible on either machine, we are able to construct the full schedule when the

setup sequence on the two machines is known. To illustrate this, consider the following example where N=4 and setup order is [1,2,6,3,5,4,7,8]. Figure 4.2 shows a Gantt chart of the solution corresponding to this order.

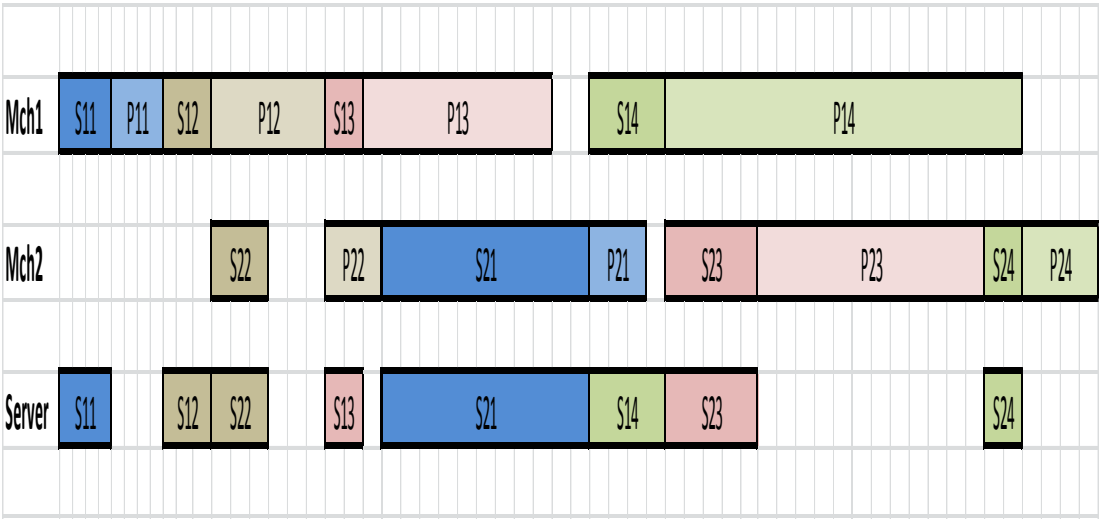

Figure 4.2: Solution Representation for the Given Example

Note that for each job, first machine operation must be done before second machine operation. Thus, in our solution representation, j must precede $j + \mathrm{N}$ for all $j < \mathrm{N}$, for the corresponding solution to be feasible.

4.2.3 Variable Neighborhood Search (VNS)

Large set of tools are (Branch & Bound, IP Formulation etc.) available to solve defined problem exactly, but they may not suffice to solve large instances due to complexity issues. Hence, we present heuristic algorithms to solve larger instances. A common heuristic approach is local search which starts with an initial solution and attempts to improve its objective function value by a sequence of local changes within a given search neighborhood. It usually continues with its search until a local optimum is reached.

In this study, we use a version of this common technique known as Variable Neighborhood Search (VNS) which was originally proposed by Hansen [24]. Instead of stopping at a local optimum point with respect to a given neighborhood, VNS

searches different neighborhoods of the current incumbent solution. It jumps from this solution to a new one if and only if an improvement has been seen. Hence, it tries to find better local optimum points at each time. If there is not any improvement on the objective function value, then VNS algorithm terminates with the best solution at hand.

A finite set of neighborhood structures denoted with N_k $(k = 1, \dots, k_{max})$ and the set of solutions in the k^{th} neighborhood of x are denoted with $N_k(x)$ are defined. VNS algorithm starts with initialization step in which, the set of neighborhood structuresN_k, k = 1,…., k_{max} , that will be used in the search are selected and initial solution x is found. Additionally stopping condition should be chosen in this step.

Search mechanism starts from first neighborhood structure, namely $k = 1$. Later a point x' at random from the k^{th} neighborhood of x $\left(x' \in N_k(x)\right)$ is generated. Some local search method is applied to x' and local optimum x'' is obtained. At this point a decision is made to move or not to this new point. If this local optimum is better than the incumbent, the move decision is made and search is continued with the very beginning of the neighborhood structures, i.e.$N_1(k \leftarrow 1)$. Otherwise, the following neighborhood structure is applied. This routine continues until the stopping condition is met.

Greedy constructive heuristic always generates a permutation schedule as an initial solution but the optimum solution may be a non-permutation schedule. Search methods are defined for the aim of also generating non-permutation schedules from a given initial permutation schedule. At this point, we decide to use VNS mechanism for defining a strategy to make crossing operations of search methods between each other with the intent of obtaining better objective values.

4.2.4 Neighborhood Structures

Before describing VNS algorithms which are adapted for our problem, four search methods are presented as neighborhood structures of the VNS algorithm.

As we mentioned before, the solution is represented as an integer array of length $2x\mathrm{N}$, and each entry of the array indicates a setup on either machine 1 or on machine 2. Search strategies are based on change of entries in this array. The crucial point is that, new solution must be feasible in the sense that no job has a setup on the second machine before having one on the first. We use the following neighborhood search structures.

1. Insert Search:

The idea of this method is searching through different orders obtained by inserting jobs to each other's positions, i.e. changing machine 1 order.

Method starts for the first setup operation on machine 1, i.e. the first entry in solution array which satisfies $j < \mathrm{N}$ condition and tries to insert setup operation of job j instead of following setup operations on machine 1 respectively. Inserting one setup operation to a new position in machine 1, affects the second machine setup order. Because of this, after each insert trial for setup operation in machine 1, method searches for possible positions in machine 2 for the subject job. This is required to generate feasible and new solutions. Inserting one job to a new position in machine 1 and searching for possible positions in machine 2 is called a trial. These trials are carried out for each job and machine 1 position pairs in a systematic way. For one job all possible positions in machine 1 are searched then search continues with the next job trials.

If there is an improvement in the objective function value during these trials, meaning that one of the new orders has a lower total completion time, then that new order is set as the main order and method starts from the beginning for the subject order. Otherwise, method keeps going until all possible trials are applied and cannot get any improvement on the objective value.

Since insert method searches for possible positions in machine 2 after inserting one job to machine 1, it allows to scan schedules where the solution is non-permutation. The Insert search method can be summarized as follows.

Algorithm (Insert Search Method)

Step 1. *(Initialization)*

Algorithm starts with a given initial order.

Step 2. *(Selection of the first job)*

First setup operation on the first machine is selected.

Step 3. *(Deciding insert or not)*

Selected setup operation is attempted to be inserted in each one of the other setup operations positions on the first machine except its own position.

After trying to insert one setup operation instead of another setup operation on the first machine, algorithm places second machine setup operation of the selected job to possible locations: from new position of first machine setup operation in the solution array to 2N. This is a single insert trial for the selected job.

If there is an improvement in the objective function value during this insert trial, insert the selected job, update the order and go to Step 1.

Otherwise apply next insert trial for the selected job by choosing next candidate setup operation on the first machine. After trying all candidate jobs, if there is not any improvement in the objective function, then go to Step 4.

Step 4. *(Selection of the next job)*

If there is not any improvement on the objective function value after applying all Insert trials for the selected job, then next setup operation on the first machine is selected. And go to Step 3.

Step 5. *(Stopping Criterion)*

When all setup operations in the first machine are selected, then STOP.

For a better understanding, following demonstration is given. Through the demonstration, TCT_k represents the total completion time for order *k* and X_i implies that job X is processed in machine *i*. According to our solution representation $X_2=X_1+N$. Same terminology will be used through the other search method's demonstrations.

Given order:

TCT_0 -	A_1	A_2	B_1	B_2	C_1	C_2

Best Bound =TCT_0

Method starts with inserting A_1 to the position of B_1.

A_2	B_1	**A_1**	B_2	C_1	C_2

Inserting A_1 to the position of B_1 violates feasibility.(A_2 must come after A_1) The algorithm continues to search possible positions for A_2.

TCT_1 -	B_1	A_1	**A_2**	B_2	C_1	C_2

TCT_2 -	B_1	A_1	B_2	**A_2**	C_1	C_2

TCT_3 -	B_1	A_1	B_2	C_1	**A_2**	C_2

TCT_4 -	B_1	A_1	B_2	C_1	C_2	**A_2**

If TCT_i > Best Bound for *i* = 1,2,3,4, algorithm moves to inserting A_1 to the position of C_1.

A_2	B_1	B_2	C_1	**A_1**	C_2

Inserting A_1 to the position of C_1 violates feasibility.(A_2 must come after A_1) The algorithm continues to search possible positions for A_2.

TCT_5 -	B_1	B_2	C_1	A_1	**A_2**	C_2

TCT_6 -	B_1	B_2	C_1	A_1	C_2	**A_2**

If TCT_i > bestBound for *i* = 5,6, algorithm moves to inserting B_1 to the position of A_1

B_1	A_1	A_2	B_2	C_1	C_2

Inserting B_1 to the position of A_1 does not violate feasibility. As the algorithm continues to search possible positions for B_2, this order will be one of the choices.

TCT_7 -	B_1	**B_2**	A_1	A_2	C_1	C_2

TCT_8 -	B_1	A_1	**B_2**	A_2	C_1	C_2

.

.

.

TCT_9 -	B_1	A_1	A_2	C_1	C_2	**B_2**

The algorithm continues in this manner until all possible trials are carried out or a new order with a better total completion time is found. Let's say order with TCT_9 has a better bound i.e TCT_9 < Best Bound, then the algorithm starts from the beginning for order 9 and Best Bound is equalized to TCT_9.

B_1	A_1	A_2	C_1	C_2	B_2

B_1 will be inserted to the position of A_1

A_1	**B_1**	A_2	C_1	C_2	B_2

Inserting B_1 to the position of A_1 does not violate feasibility. As the algorithm continues to search possible positions for B_2, this order will be one of the choices.

TCT_{10} -	A_1	B_1	**B_2**	A_2	C_1	C_2

TCT_{11}-	A_1	B_1	A_2	**B_2**	C_1	C_2

.

.

.

TCT_{12} -	A_1	B_1	A_2	C_1	C_2	**B_2**

2. Pairwise Interchange Search:

Basic principal of pairwise interchange method is switching two jobs entirely both in machine one and machine two orders. The method searches for new orders according to the order of machine 1. Method starts with the first setup operation pairs on machines 1 and 2, i.e. the first and second entries in solution array which satisfy $j < \mathrm{N}$ condition. After the two jobs are chosen to be interchanged, then the corresponding second machine orders are also switched. This is done in order to preserve feasibility.

In order to avoid searching same orders, each job is tried to be switched respectively with the jobs that comes after it. The procedure continues until a new order with a better solution is obtained or all allowed interchanging operations are applied. In former case, method starts from the beginning for with the new order as the initial order. Otherwise method terminates with the current best solution and the order. The pairwise interchange search method can be summarized as follows.

Algorithm (Pairwise Interchange Search Method)

Step 1. *(Initialization)*

Algorithm starts with a given initial order.

Step 2. *(Selection of the first job)*

First setup operation on the first machine is selected.

Step 3. *(Deciding switch or not)*

Selected setup operation is switched with following setup operation on the first machine, and second machine setup operations of the affected jobs are also switched simultaneously. This is a single pairwise interchange trial.

If there is an improvement on the objective function value for the trial, switch the selected job pairs, update the order and go to Step 1.

Else, apply next pairwise interchange trial for the selected setup operation by choosing next setup operation on the first machine.

Step 4. *(Selection of the next job)*

If there is not any improvement in the objective function value after applying all pairwise interchange trials for the selected job, then the next setup operation on the first machine is selected. And go to Step 3.

Step 5. *(Stopping Criterion)*

When all setup operations in the first machine are selected, then STOP.

Similar to insert method, for a better understanding, following demonstration is given.

Given order:

TCT_0 -	A_1	A_2	B_1	B_2	C_1	C_2

Best Solution =TCT_0

Method starts with switching job A with job B.

B_1	A_2	**A_1**	B_2	C_1	C_2	- Switching jobs in machine 1 order.

B_1	**B_2**	A_1	**A_2**	C_1	C_2	- Switching jobs in machine 2 order.

This is the new feasible order with objective value TCT_1 to be checked with current best solution. If TCT_1 > Best Solution, then job A is switched with job C.

C_1	A_2	B_1	B_2	**A_1**	C_2	- Switching jobs in machine 1 order.

C_1	**C_2**	B_1	B_2	A_1	**A_2**	- Switching jobs in machine 2 order.

This is the new feasible order with objective value TCT_2 to be checked with current best solution. If TCT_2 > Best Solution, then job B is switched with job C not job A.

A_1	A_2	**C_1**	B_2	**B_1**	C_2	- Switching jobs in machine 1 order.

A_1	A_2	C_1	**C_2**	B_1	**B_2**	- Switching jobs in machine 2 order.

This is the new feasible order with objective value TCT_3 to be checked with current best bound. If TCT_3 > Best Solution, then algorithm terminates, since there is no job

other than C that comes after B and there is no job that comes after C. However if TCT_3 < Best Solution, then the algorithm starts all over again for this new order 3. First switch will be between A and C.

3. Second Machine Move Search:

According to our problem definition, each job must be setup in machine 1 before setup in machine 2. However there is no constraint for non-permutation, setup operations on the second machine may be done in any order after from its first machine setup operation.

Second machine move method starts with a given feasible order. Therefore, using the fact above, moving each second machine setup operation to the right starting from its own position to end of order does not violates feasibility but generates new feasible solutions for our problem. The second machine move search method can be summarized as follows.

Algorithm (Second Machine Move Search Method)

Step 1. *(Initialization)*

Algorithm starts with a given initial order.

Step 2. *(Selection of the first job)*

First setup operation on the second machine is selected.

Step 3. *(Shifting selected setup operations to the right)*

Selected setup operation is shifted right at each trial until to the position 2N in the solution array.

If there is an improvement in the objective function value for the trial; shift selected setup operation to this position, update order and go to Step 1.

Step 4. *(Selection of the next job)*

If there is not any improvement in the objective function value at the end of any of the trials for the selected job, then next setup operation on the second machine is selected. And go to Step 3.

Step 5. *(Stopping Criterion)*

When all setup operations in the second machine are selected, then STOP.

Method starts with first setup operation on machine 2 i.e., the first entry in the solution array that satisfies $j > \mathrm{N}$. Then it continues for all the second machine setup operations. This procedure continues until all allowed combinations are searched or a new order with a better objective value is found. If a better solution is found, method starts all over again from this new order.

Given order:

TCT_0 -	A_1	A_2	B_1	B_2	C_1	C_2

Best Solution =TCT_0

-Method starts with moving A_2 to the right by one unit.

TCT_1 -	A_1	B_1	**A_2**	B_2	C_1	C_2

This is a new feasible order with objective value TCT_1 to be checked with current best solution. If TCT_i > Best Solution for i = 2, 3, 4, then job A_2 continues to move to the right until 2N.

TCT_2 -	A_1	B_1	B_2	**A_2**	C_1	C_2

TCT_3 -	A_1	B_1	B_2	C_1	**A_2**	C_2

TCT_4 -	A_1	B_1	B_2	C_1	C_2	**A_2**

If a better solution is not found, method moves to second setup in machine 2.

-Moving B_2 to the right by one unit.

TCT_5 -	A_1	A_2	B_1	C_1	**B_2**	C_2

The algorithm continues in this manner until all combinations are searched or a better solution is found. Let's say TCT_5 < Best Solution, then order 5 is the new main order and the method starts from the beginning for order 5 by moving A_2 to the right.

4. First Machine Move Search:

First machine move method uses the same logic with the second machine move method. Method starts with first setup operation in machine 1 according to the given solution i.e., first entry in the solution array that satisfies $j < \mathrm{N}$. For each job all possible positions are searched from its position in solution array to zero index. Then it continues for all the first machine setup operations. This routine continues until all allowed combinations are searched or a new order with a better objective value is found. If a better solution is found, method starts all over again from this new order. The first machine move search method can be summarized as follows.

Algorithm (First Machine Move Search Method)

Step 1. *(Initialization)*

Algorithm starts with a given initial order.

Step 2. *(Selection of the first job)*

Second setup operation on the first machine is selected.

Step 3. *(Shifting selected setup operations to the left)*

Selected setup operation is shifted left at each trial until to the (position 0) in the solution array.

If there is an improvement in the objective function value for the trial; shift the selected setup operation to this position, update the order and go to Step 1.

Step 4. *(Selection of the next job)*

If there is not any improvement in the objective function value at the end of all trials for the selected job, then next setup operation on the first machine is selected. And go to Step 3.

Step 5. *(Stopping Criterion)*

When all setup operations in the first machine are considered, then STOP.

Given order:

TCT_0 -	A_1	A_2	B_1	B_2	C_1	C_2

Best Solution =TCT_0

-Method would try to move A_1 to the left. However since it is the first job of the order, method will move to second job in machine 1 which is B_1.

TCT_1 -	A_1	**B_1**	A_2	B_2	C_1	C_2

This is a new feasible order with objective value TCT_1 to be checked with current best solution. If TCT_1 > Best Solution, then job B_1 continues to move to the left until index 0.

TCT_2 -	**B_1**	A_1	A_2	B_2	C_1	C_2

If a better solution is not found, method moves to the third job processed in machine 1, C_1.

-Moving C_1 to the left by one unit.

TCT_3 -	A_1	A_2	B_1	**C_1**	B_2	C_2

TCT_4 -	A_1	A_2	**C_1**	B_1	B_2	C_2

Method continues to move C_1 until a better solution is found or C_1 is tried at each position from its original position to 0. Let's say order 4 is a better solution. Then it becomes the main order and method starts all over again for order 4, by again trying to move A_1 to the left then by moving second job C_1 to the left unit by unit.

4.2.5 VNS 1 Algorithm

The VNS 1 algorithm takes in the initial solution obtained from the constructive heuristic and attempts to improve it by searching the Insert, Pairwise Interchange, Second Machine Move and First Machine Move neighborhood structures. When the local optimum at a given neighborhood results in an improvement in the objective function value of the initial solution fed into that neighborhood in the current iteration, the search mechanism goes back to the Insert neighborhood and starts a new iteration with the current best solution found up to that point. VNS 1 algorithm terminates only, if there isn't any improvement on the objective value at the end of last search method.

Initial experimentation shows that the order in which different neighborhoods are searched has an effect on solution quality. Nonetheless, there is no evidence that indicates that any particular order is better than the other. Thus we make a judgment call and first search the insert and pairwise interchange neighborhoods that are of larger sizes than the other two. Then second machine move and first machine move neighborhoods are searched respectively. Since we have no clear direction on whether insert should precede or succeed interchange, we try both resulting in cases 1 and 2, respectively. Hence we compute solution of both cases and accept solution which has the minimum objective value. During this thesis, VNS algorithms are described on case 1 in which insert method is applied first. For the second case, everything is the same except that order of insert search and pairwise search is exchanged. The details of the algorithm are as presented in the following.

Algorithm (Variable Neighborhood Search 1)

Step 1. *(Initialization)*

Apply Greedy Constructive Heuristic and find an initial order.

Best Solution = Initial Solution

Best Order = Initial Order.

Step 2. *(Insert Search Method)*

Execute insert search method for the best order.

If Insert Solution $<$ Best Solution

Best Solution = Insert Solution

Best Order = Insert Order

Go to the Step 3.

Step 3. *(Pairwise interchange Search Method)*

Execute pairwise interchange search method for best order.

If Pairwise Solution < Best Solution

Best Solution = Pairwise Solution

Best Order = Pairwise Order

Go to the Step 2.

Else, Go to the Step 4.

Step 4. *(Second Machine Move Search Method)*

Execute second machine move search method for best order.

If Second Machine Move Solution < Best Solution

Best Solution = Second Machine Move Solution

Best Order = Second Machine Move Order

Go to the Step 2.

Else, Go to the Step 5.

Step 5. *(First Machine Move Search Method)*

Execute first machine move search method for best order.

If First Machine Move Solution < Best Solution

Best Solution = First Machine Move Solution

Best Order = First Machine Move Order

Go to the Step 2.

Else, STOP and RETURN Best Solution and Best Order.

A flow chart is given in Figure 4.3 to illustrate VNS 1 Algorithm.

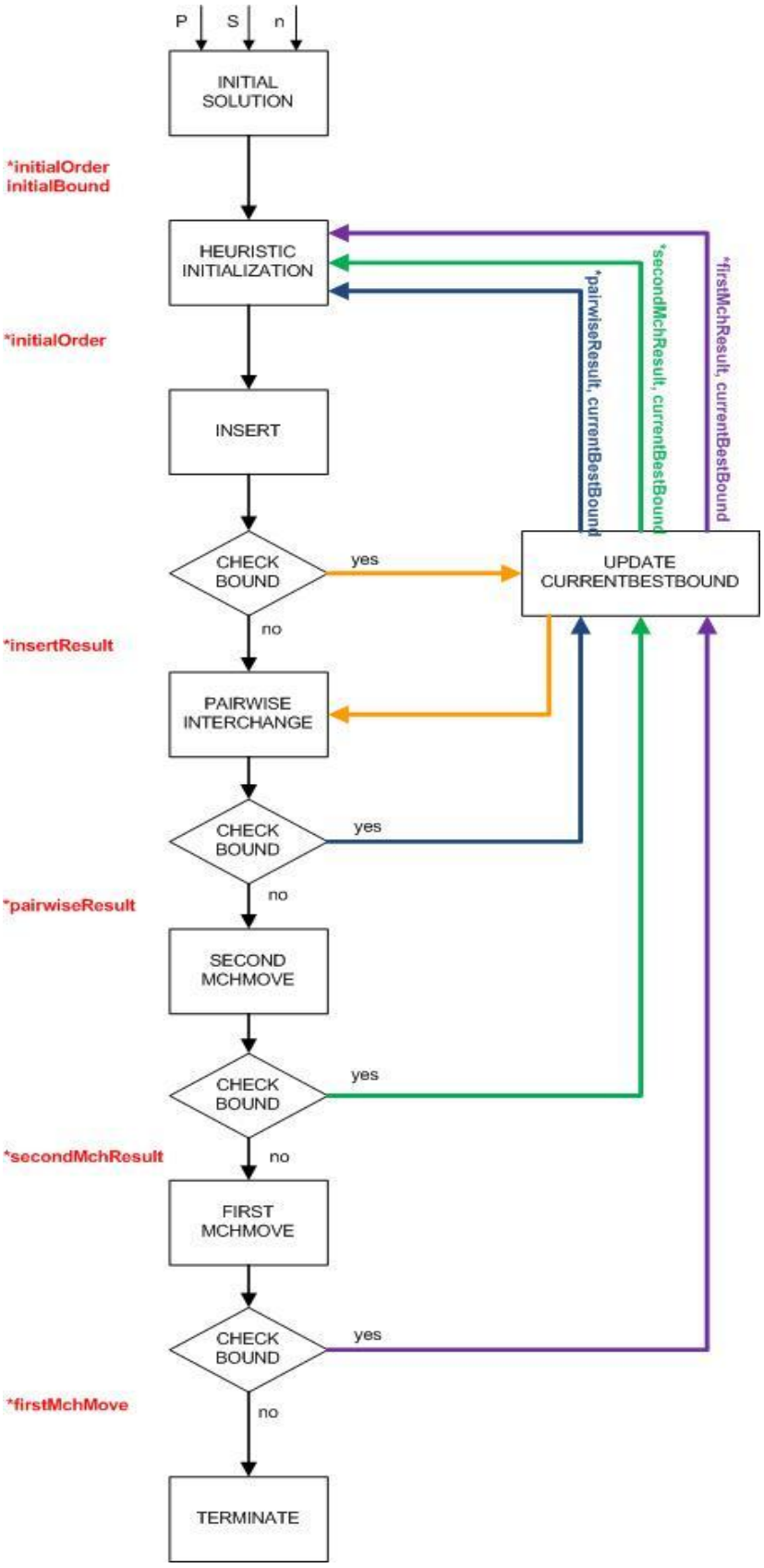

Figure 4.3: Flow Chart of VNS 1 Algorithm

4.2.6 VNS 2 Algorithm

The motivation behind the VNS 2 algorithm is that solving VNS 1 may still be computationally expensive for larger problems. In VNS-1 algorithm, search methods work recursively in the sense that the search begins a new each time local optimum solution is found. For any search method, if a better order is found then search mechanism starts from scratch for this order. At this point, we decide to limit number of search trials of each candidate job in search methods. Since, each job should have a chance to be tried for all possible positions; we limit number of search trials with the number of jobs.

For a better understanding a flow chart is presented for insert and pairwise interchange search methods in Figure 4.4 and another flow chart is presented for first machine move and second machine move methods in Figure 4.5. Working principles of the insert and pairwise interchange search methods are similar. Hence they are explained through the same flow chart. This is also the case for First Machine Move and Second Machine Move methods hence they are also explained on the same flow chart. As clearly seen on flow charts, additionally to VNS 1 algorithm, trial check step is added to VNS 2 algorithm for each search method.

Aside from this, everything is the same as in VNS 1 algorithm.

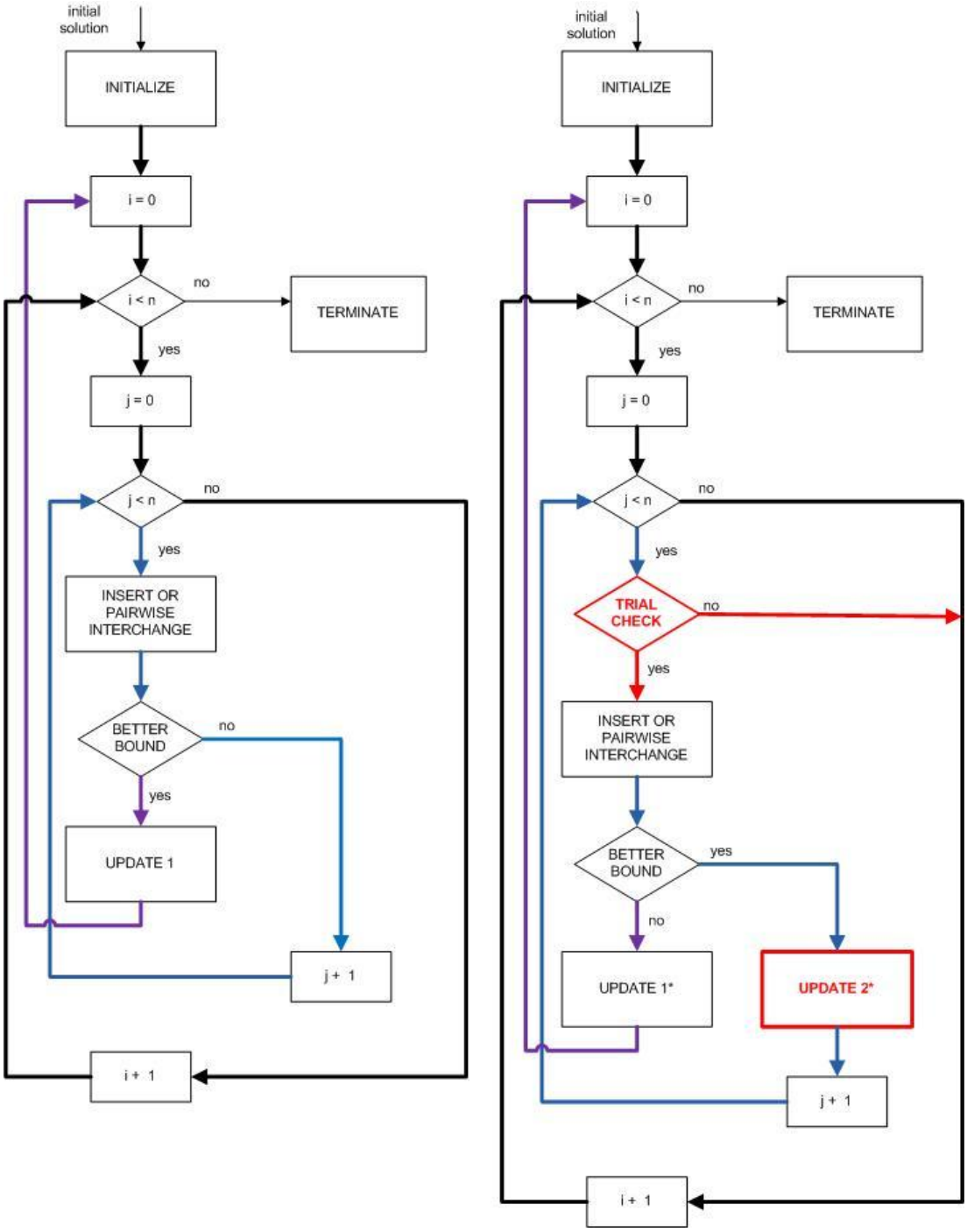

Figure 4.4: Flow Chart of Insert/Pairwise Interchange Methods for VNS 1 and VNS 2

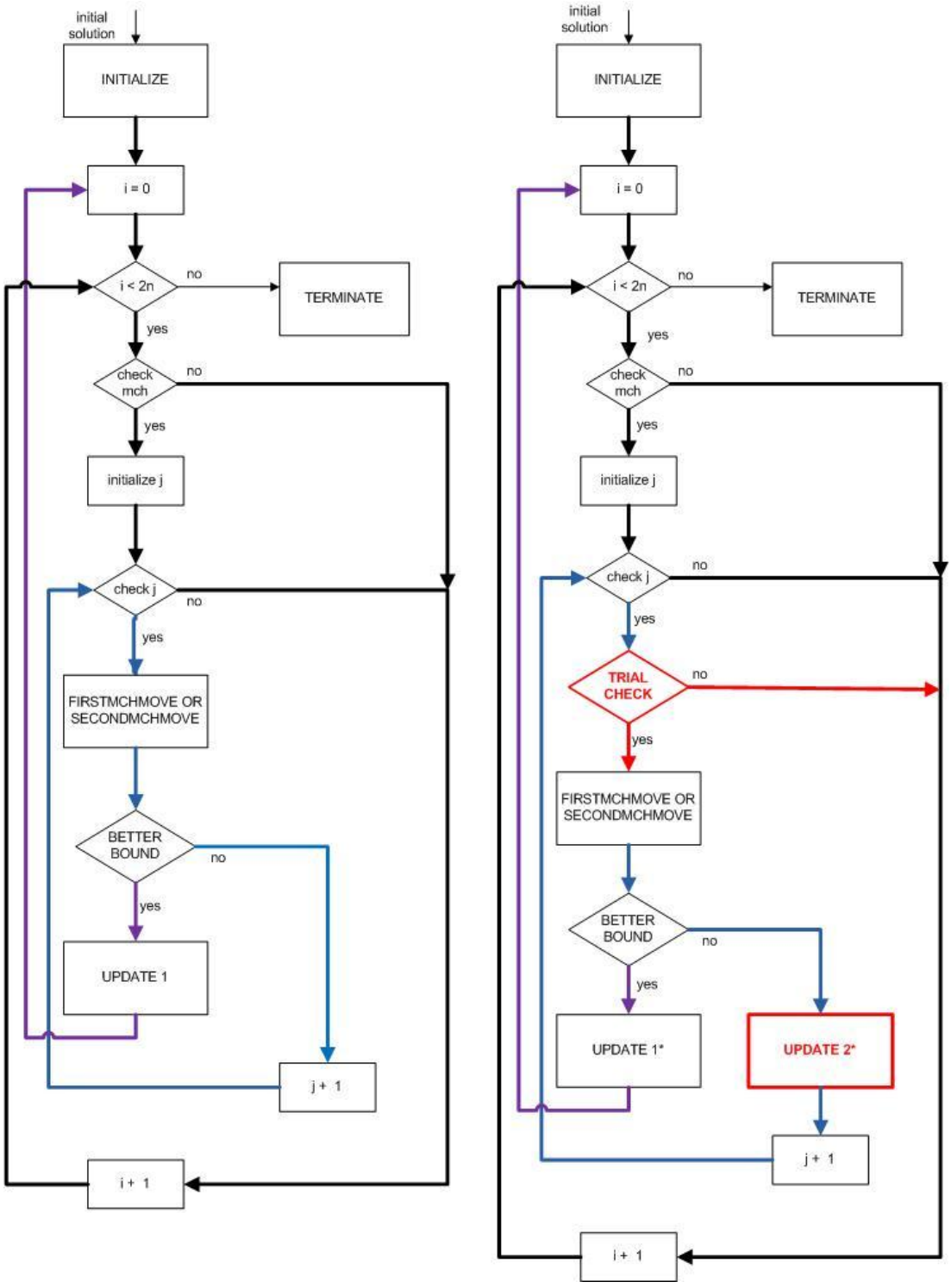

Figure 4.5: Flow Chart of First Machine Move/Second Machine Move Methods for VNS 1 and VNS 2

Chapter 5

Computational Results

In this section, we first define the experimental design and then report and discuss our computational results.

5.1 Computational Setup

Our control parameters are the number of jobs, and the processing and setup times of the jobs.

Since minimizing total completion time in a two machine flow shop with a single server is NP-hard in the strong sense [3], it is not possible to solve it optimally except for small instances. In fact, we are able to solve only those instances with up to 10 jobs with our mixed integer programming. Thus, to be able to assess the quality of the lower bounds and that of the heuristic solutions against optimal solutions we first generate small instances with N=5, N= 8 and N=10. On the other hand, to observe effectiveness and applicability of heuristic algorithms for more practical larger instances we also generate medium and large sized instances with N=15, N=20, N=30, N=35, N=50 and N=100. The better of the VNS 1 and VNS 2 solutions is taken as the heuristic solution in these experiments. An exception is made for N=100 where only VNS 2 is applied to these instances.

As for the processing times (p) and setup times (s), we focus on their relative magnitudes. To describe this relation, we define R = ($\bar{s}/\bar{p}$) as the ratio of the average setup time to the average processing time. To decide which R values should be used, we follow Lim et al. [1], who study this problem with the makespan objective, and use nine different ratios R, which are 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50 and 100 respectively. The processing and setup times are generated from uniform distributions with ranges of (0,100) and (0,100R), respectively for R=0.01, 0.05, 0.1, 0.5 and 1. As for R=5, 10, 50 and 100 setup times are uniform between (0,100) and processing time is uniform between (0,100R), respectively. For example, if we choose N=10 and R=0.5, then each of the 10 instances has processing times as uniformly distributed from 1 to 100 and setup times as uniformly distributed from 1 to 50.

For each N and R combination except when N=10, 10 random instances are generated for all. In those combinations with N=10, however, only five random instances are generated due to the long computational times required in obtaining optimum solutions.

The MIP model and lower bound 4 is solved with GUROBI Optimizer 5.0 using Python 2.7.2 as an interface, and all heuristics and lower bounds 1,2,3 are programmed and implemented in C++ programming language using Microsoft Visual Studio 2008. All experiments are performed on an Intel (R) Core (TM) i5-2430M CPU processor with a clock speed of 2.40GHz and 4 GB RAM running under Windows 7 Enterprise.

5.2 Experimental Results

The results of the computational experiments are summarized in this section. The software package for the proposed model and algorithm output various results to the user. These are optimal objective value of the model/algorithm and the running time of the model/algorithm which corresponds to the speed of model/algorithm.

We compare the performance of the lower bounds and heuristic algorithms according to these outputs.

Detailed results of the all experiments are presented in Appendix.

5.2.1 Comparison of Lower Bounds

In order to assess the efficiency of the proposed lower bounds, we solved our instances optimally. When we tried to determine problem size, for which we can obtain optimum solutions, we observed that the problem could be solved up to 8 jobs for all of the *R* values and up to 10 jobs for *R*=0.01, 0.05, 0.1, 0.5, 1 . Although these sizes are small, this is not a surprising finding especially when the work on this problem with the makespan objective is considered. For example, Lim et al. [1], report that they can solve the problem optimally only for up to 10 jobs with the makespan objective also.

In the Table 5.1, we present average results of the 10 instances for each *R* and N=5, 8, 10. Shaded values in the table show the best lower bound performances. Figures 5.1, 5.2 and 5.3 also presents these results visually for N=5, 8 and 10. The percentage deviation is calculated as (OPT-LB)/OPT $\times$ 100 %, where OPT is the $\sum Cj$ of optimal solution found by solving MIP model. The LP relaxation of our MIP model is also taken as a lower bound in Table 5.1. Since however those lower bounds 1 and 3 always outperform it, we no longer use it as a lower bound in the remainder of the thesis.

We observe that some of the lower bounds are closer to the optimal solution for specific *R* values. First one of them is lower bound 2, which performs better when *R*=5, 10, 50, 100. According to the lower bound 2, a relaxed problem, which assumes that setup server is continuously busy performing setup operations, is defined. So, setup times are considered mainly while computing this lower bound. Since lower bound 2 primarily considers setup times, it is predictable to get close to optimal results for the data type where setup operations are much longer than process operations. The other one is lower bound 4, which gives better lower bound for *R*=0.01, 0.05, 0.1 values. While solving integer programming formulation for the

lower bound 4, only second machine setup times are not considered according to the defined problem. Processing times on both machines are mainly considered and setup times on the first machine are added to processing times on the same machine. Similarly, for the data type where process operations are much longer than setup operations, lower bound 4 works better since it is the most process based lower bound. Finally, lower bound 3 gives better results than other lower bounds for *R*=0.5, 1 values. Due to the fact that lower bound 3 tries to consider idle times as much as possible, for the case where setup times and process times are near each other, it is comprehensible to get best performance from lower bound 3, because there will be more idle time on both machines and the setup server in this range of *R* values. Also, it is observed that lower bound 4 which totally disregards the setups on machine 2 is very unlikely to perform well for the R=5, 10, 50, 100 values. Hence, it is not computed in later experiments for these R values.

Table 5.1: Results for Comparison of Lower Bounds with Optimal Solutions

Data Type		LB1	LB2	LB3	LB4	LP Relaxed of MIP	Opt. Soln.	Percentage Deviation of				
								Opt. &LB1	Opt. &LB2	Opt. &LB3	Opt. &LB4	Opt.& LP Relx. of MIP
N=5	R=0,01	583,5	279,8	812,9	890,6	578,5	892,7	34,64	68,66	8,94	0,24	35,20
	R=0,05	722,2	355,8	836,9	919,2	714,7	931,1	22,44	61,79	10,12	1,28	23,24
	R=0,1	587,7	367,9	849,8	891,2	574,2	911,3	35,51	59,63	6,75	2,21	36,99
	R=0,5	844,5	781,3	1087,4	1073,8	804,0	1193,3	29,23	34,53	8,87	10,01	32,62
	R=1	1357,6	1505,5	1615,2	1533,3	1301,1	1734,9	21,75	13,22	6,90	11,62	25,00
	R=5	796,9	1298,3	1200,8	813,9	686,4	1317,1	39,50	1,43	8,83	38,21	47,89
	R=10	765,8	1392	1299,1	741,6	712,3	1395,6	45,13	0,26	6,91	46,86	48,96
	R=50	689,2	1260,6	1161,8	583,7	620,2	1261,4	45,36	0,06	7,90	53,73	50,83
	R=100	563,2	1092,9	1019,7	549,5	500,2	1092,9	48,47	0,00	6,70	49,72	54,23
N=8	R=0,01	1441,8	493,3	1570,8	1837,8	1433,8	1854,6	22,26	73,40	15,30	0,91	22,69
	R=0,05	1784,5	635,4	1742,6	2019,6	1776,5	2073,9	13,95	69,36	15,97	2,62	14,34
	R=0,1	1571,2	730,1	1997,3	2111,3	1561,6	2176,1	27,80	66,45	8,22	2,98	28,24
	R=0,5	2362,6	1921,9	2720,6	2747,4	2296,2	3016,6	21,68	36,29	9,81	8,92	23,88
	R=1	2814,7	3142,7	3415	3096,1	2760,3	3872,5	27,32	18,85	11,81	20,05	28,72
	R=5	1915	3287,5	3099,5	2000,8	1813,4	3325,4	42,41	1,14	6,79	39,83	45,47
	R=10	1559,8	2925,8	2651,8	1543,7	1467,0	2933,8	46,83	0,27	9,61	47,38	50,00
	R=50	1454	3085,6	2740,6	1453,4	1391,6	3087,7	52,91	0,07	11,24	52,93	54,93
	R=100	1322,4	2942,4	2681,5	1509,5	1244,0	2942,4	55,06	0,00	8,87	48,70	57,72
N=10	R=0,01	2163,8	637,4	2516	2775,8	2153,8	2796,2	22,62	77,20	10,02	0,73	22,97
	R=0,05	2165	768	2531	2792,6	2153,0	2853,6	24,13	73,09	11,31	2,14	24,55
	R=0,1	2508,8	1049,4	2471,8	2705,8	2492,8	2874	12,71	63,49	13,99	5,85	13,26
	R=0,5	3633,8	2989	4143	4189,6	3559,8	4625	21,43	35,37	10,42	9,41	23,03
	R=1	4351,6	4395,2	4810,2	4537	4299,6	5832,8	25,39	24,65	17,53	22,22	26,29

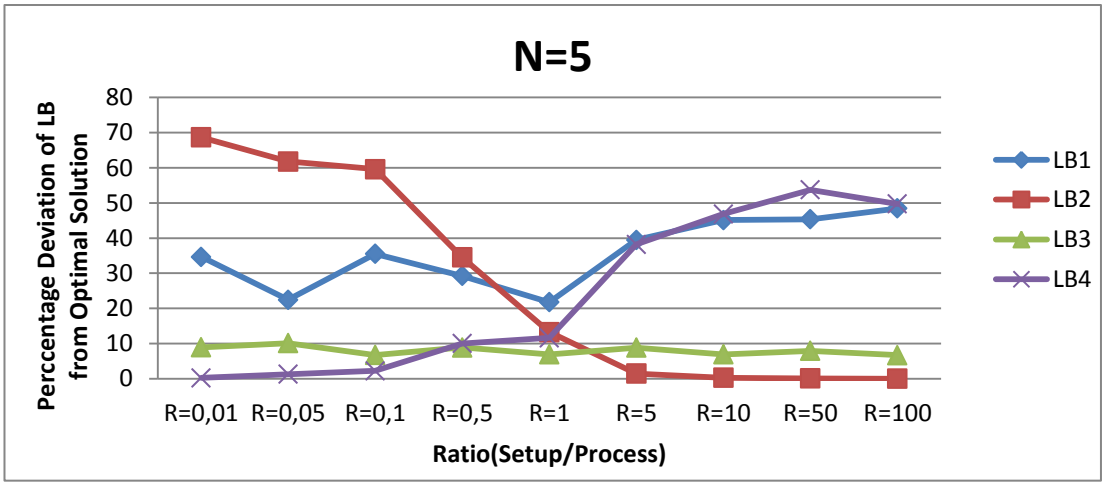

Figure 5.1: Percentage Deviation of Lower Bounds from Optimal Solutions for N=5.

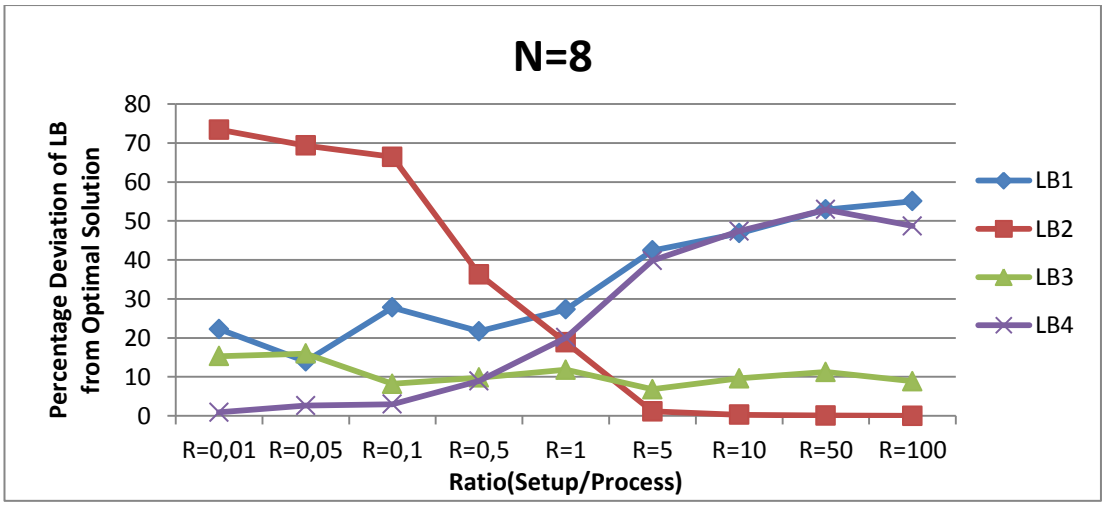

Figure 5.2: Percentage Deviation of Lower Bounds from Optimal Solutions for N=8.

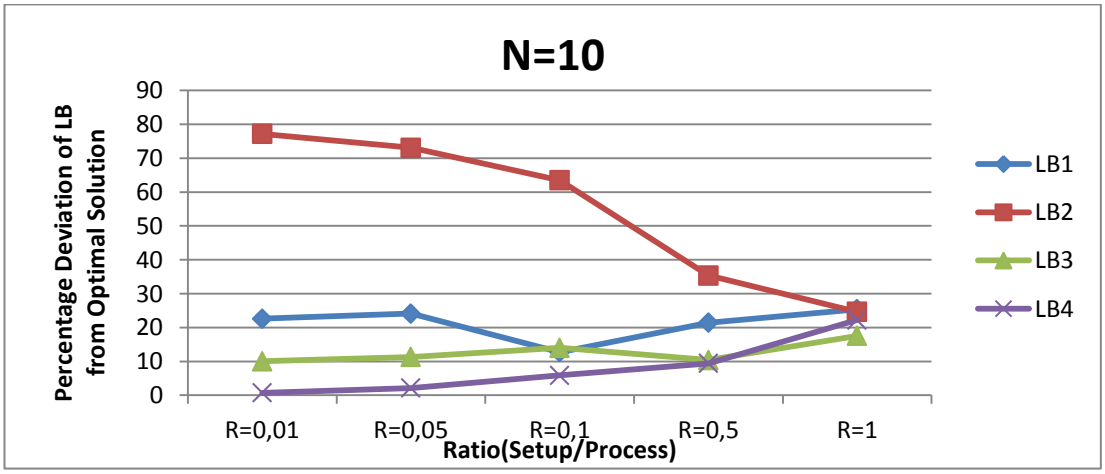

Figure 5.3: Percentage Deviation of Lower Bounds from Optimal Solutions for N=10.

Another aspect that we compare lower bounds between each other is their running times. Even though running times of the lower bound 1, lower bound 2, lower bound 3 and lower bound 5 is negligible for all the experiments, lower bound 4 has a remarkable running time for job sizes of greater than 20, since it is an integer programming model. Figure 5.4 shows average running time of the lower bound 4 according to job size.

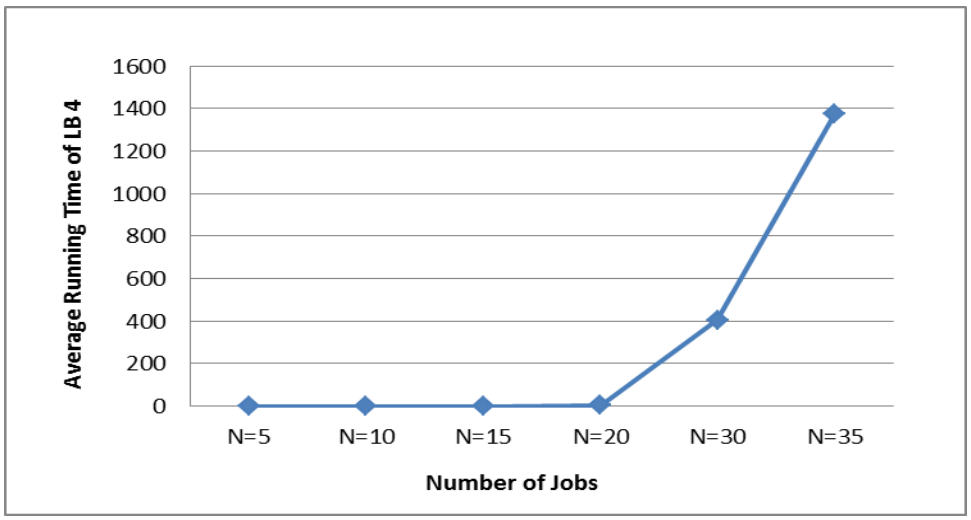

Figure 5.4: Average Running Time of LB 4 according to N.

5.2.2 Comparison of Best Lower Bound with Optimal Solution

The maximum of all lower bounds is used for performance assessment. For the same experiment given above, we compared our best lower bound with the optimal solution. The results of the experiment are given Table 5.2.

Deviations from lower bounds vary depending on the *R* and N values. The lower bounds are tight when *R* is small or large, but loose when *R* is in the range [0.5,1]. The reason is that when setup times and processing times are close to each other, there is more avoidable idle time on both machine and the setup server. Additionally, for the same *R* values, lower bounds get worse as N gets larger. Since objective function of our problem is minimizing total completion time, as N gets larger the gap between best lower bound and optimal solution grows. To illustrate these results, Figure 5.5 presents the deviations between best lower bounds and optimum results for different *R* and N values.

Table 5.2: Results for Comparison of Best Lower Bound with Optimal Solution

Data Type		Best LB	Optimal Solution		Perc.Dev. of LB from Opt.Soln
			Obj.Val	Time(sec.)	
N=5	R=0.01	890.6	892.7	1.44	0.24
	R=0.05	919.2	931.1	1.19	1.28
	R=0.1	891.3	911.3	1.76	2.19
	R=0.5	1128.7	1193.3	3.73	5.41
	R=1	1628.3	1734.9	3.67	6.14
	R=5	1299	1317.1	16.90	1.37
	R=10	1392	1395.6	32.76	0.26
	R=50	1260.6	1261.4	38.23	0.06
	R=100	1092.9	1092.9	21.10	0.00
N=8	R=0.01	1837.8	1854.6	18.15	0.91
	R=0.05	2021.7	2073.9	22.17	2.52
	R=0.1	2111.3	2176.1	64.659	2.98
	R=0.5	2766.6	3016.6	161.488	8.29
	R=1	3516.7	3872.5	269.597	9.19
	R=5	3288.4	3325.4	1708.81	1.11
	R=10	2925.8	2933.8	1825.21	0.27
	R=50	3085.6	3087.7	7072.42	0.07
	R=100	2942.4	2942.4	4797.64	0.00
N=10	R=0.01	2775.8	2796.2	304.87	0.73
	R=0.05	2792.6	2853.6	1183.52	2.14
	R=0.1	2705.8	2874	339.76	5.85
	R=0.5	4262.8	4625	2832.71	7.83
	R=1	5213.8	5832.8	8779.21	10.61

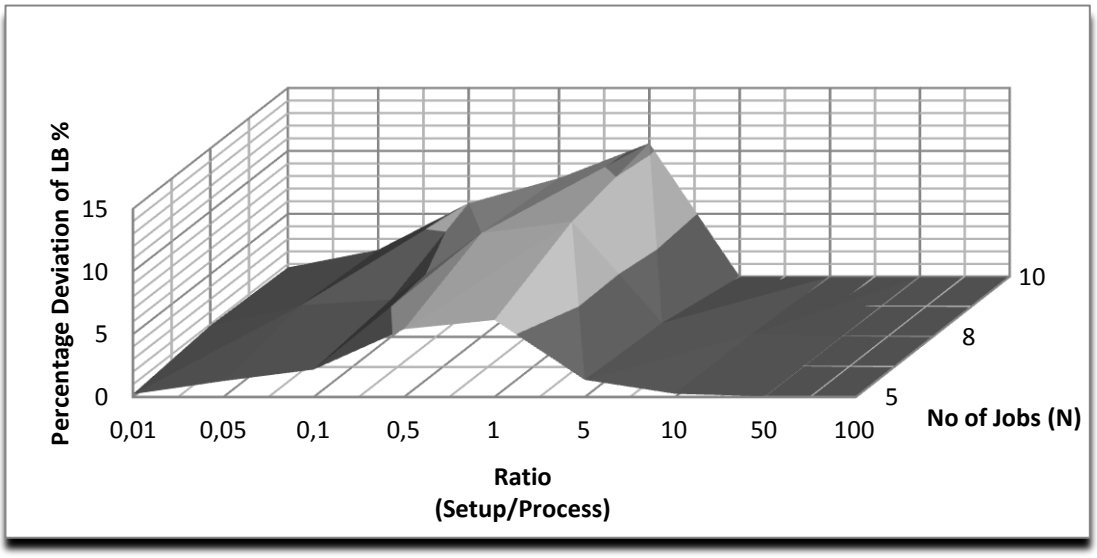

Figure 5.5: Percentage Deviation of LB from Optimal Solutions according to N and R

Another criterion based on which we compare the performance of the best lower bound with the optimal solution is the running time. Average running time of the optimal solutions is presented according to the number of jobs and ratios of setup/process in Figure 5.6. Despite the fact that running time of the best lower bound is negligible for the given experiment, running time of the optimal solution grows significantly as the number of job gets larger. Therefore, while evaluating the performance of heuristic algorithms for the larger sized instances, all solutions are compared with the best lower bound.

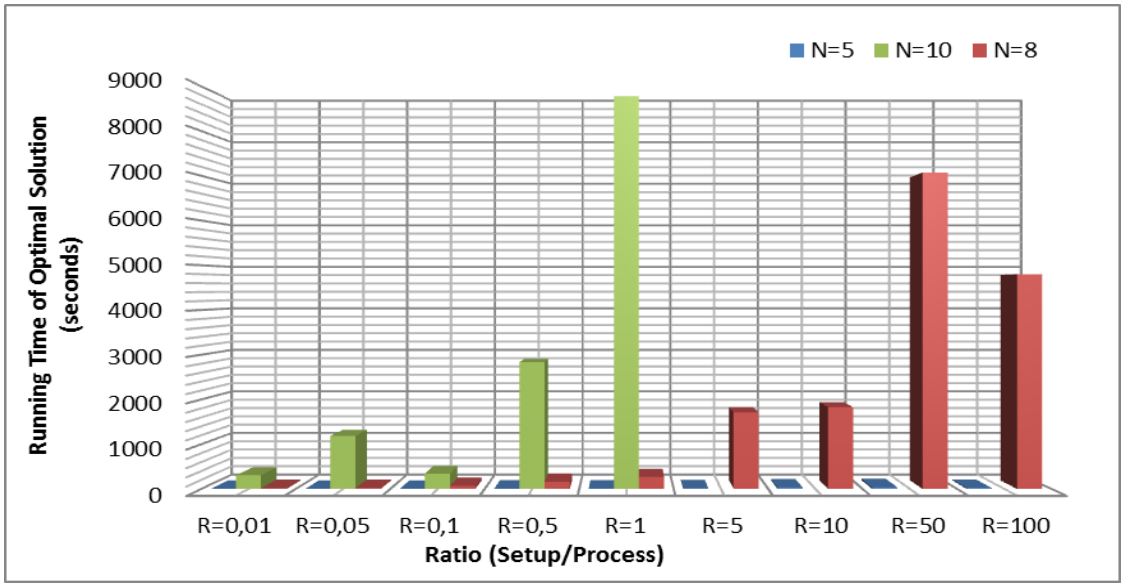

Figure 5.6: Running Time of the Optimal Solutions according to *R* and N

5.2.3 Comparison of Heuristic Algorithms with Optimum Solutions

For small instances we compare performance of the proposed heuristic algorithms with optimum solutions. As we mentioned before, the smaller of the VNS 1 and VNS 2 solutions is taken as the solution of the heuristic algorithm. Average results of the instances for each *R* and N values are presented in Table 5.3.

When we analyze experimental results, first we notice that percentage deviation of heuristic from optimal solutions grows as N gets larger. As a result of Variable Neighborhood Search structure, our heuristic algorithms search for local optimal points and switch between different neighborhood structures when they

are unable to find an improvement in one. Naturally for the small sized problems, the solution space is also small. Hence, probability to finding a global optimal point while searching for local optimal points is also high for small sized problems. As we can see clearly in Figure 4.10, when N=5, our heuristic algorithm solves most of the instances optimally. On the contrary, when problem size gets larger, probability of finding global optimal is low because solution space also gets larger. Figure 5.7 also illustrates this situation for N=10.

Table 5.3: Results for Comparison of Heuristic Algorithms with Optimal Solutions

Data Type		Heuristic	Optimal Solutions		Perc.Dev.of Heuristic from Opt.Soln.
		Obj.Val	Obj.Val	Time(sec.)	
N=5	R=0.01	893.7	892.7	1.44	0.11
	R=0.05	937.9	931.1	1.18	0.73
	R=0.1	912.2	911.3	1.76	0.10
	R=0.5	1193.3	1193.3	3.73	0.00
	R=1	1734.9	1734.9	3.67	0.00
	R=5	1317.1	1317.1	16.89	0.00
	R=10	1395.6	1395.6	32.75	0.00
	R=50	1261.4	1261.4	38.22	0.00
	R=100	1092.9	1092.9	21.10	0.00
N=8	R=0.01	1865.3	1854.6	18.15	0.58
	R=0.05	2083.1	2073.9	22.17	0.44
	R=0.1	2192.3	2176.1	64.65	0.74
	R=0.5	3029.4	3016.6	161.48	0.42
	R=1	3912.7	3872.5	269.59	1.04
	R=5	3325.4	3325.4	1708.81	0.00
	R=10	2933.8	2933.8	1825.21	0.00
	R=50	3087.7	3087.7	7072.42	0.00
	R=100	2942.4	2942.4	4797.64	0.00
N=10	R=0.01	2825	2796.2	304.87	1.03
	R=0.05	2881.2	2853.6	1183.52	0.97
	R=0.1	2893.6	2874	339.76	0.68
	R=0.5	4691.8	4625	2832.71	1.44
	R=1	5926.2	5832.8	8779.21	1.60

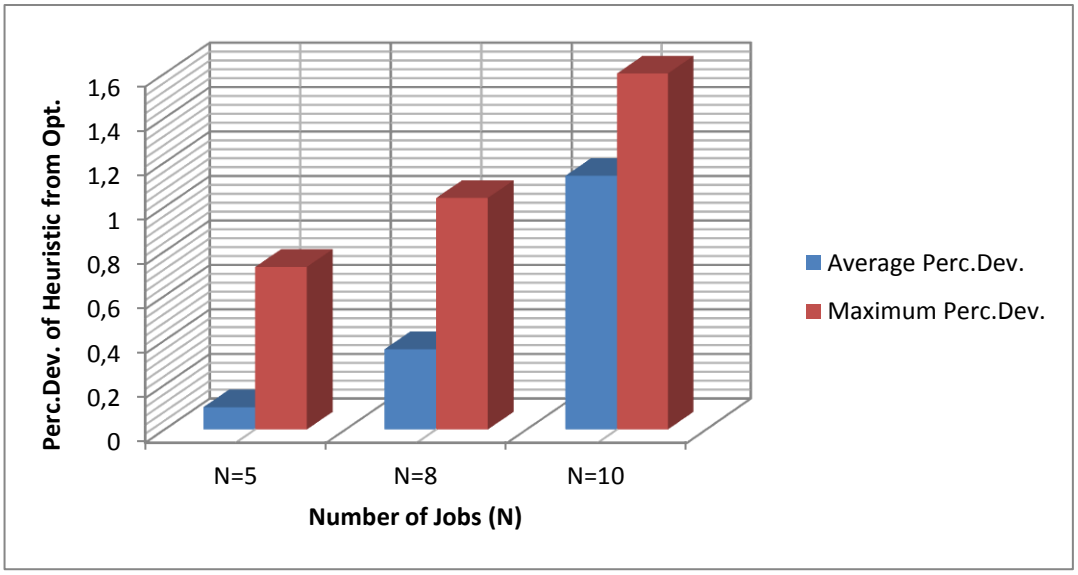

Figure 5.7: Maximum and Average Percentage Deviation of Heuristic from Optimal

When we assess the performance as a function of the ratio of setup and processing times, we see that when *R* is small or large, the heuristic algorithm gives near optimal solutions. This is because; when *R* is too small, process times are larger than setup times, hence setup operation on machine 2 may be done while process operation continues on machine 1. This situation, decreases idle times due to setup server and increases performance of the heuristic algorithms. In the same manner, when *R* is too large, setup times are larger than process times, process operation on machine 2 may be done while setup operation continues on machine 1. This situation decreases machine idle time and increases performance of the heuristic algorithms.

5.2.4 Comparison of Heuristic Algorithm with the Best Lower Bound

We compare performance of the heuristic algorithms with the best lower bound for problems with more than 10 jobs. Tables 5.4, 5.5 present the average results of the 10 instances for different N and R values. The percentage deviation is calculated by (Heuristic-LB)/LB $\times$ 100 %.

Table 5.4: Results for Comparison of Heuristic Algorithms with Best LB – 1/2

Data Type		Heuristic		Best Lower Bound		Perc.Dev. of H&LB
		Obj.Val	Time(sec.)	Obj.Val	Time(sec.)	
N=5	R=0.01	893.70	0.00	890.60	0.00	0.35
	R=0.05	937.90	0.00	919.20	0.00	2.03
	R=0.1	912.20	0.00	891.30	0.00	2.34
	R=0.5	1193.30	0.00	1128.70	0.00	5.72
	R=1	1734.90	0.00	1628.30	0.00	6.55
	R=5	1317.10	0.00	1299.00	0.00	1.39
	R=10	1395.60	0.00	1392.00	0.00	0.26
	R=50	1261.40	0.00	1260.60	0.00	0.06
	R=100	1092.90	0.00	1092.90	0.00	0.00
N=10	R=0.01	2925.60	0.00	2889.30	0.00	1.26
	R=0.05	3009.20	0.00	2882.50	0.00	4.40
	R=0.1	3072.40	0.00	2941.50	0.00	4.45
	R=0.5	4512.30	0.00	4113.30	0.00	9.70
	R=1	6264.30	0.00	5566.00	0.00	12.55
	R=5	4691.50	0.00	4643.70	0.00	1.03
	R=10	4464.60	0.00	4446.30	0.00	0.41
	R=50	4518.10	0.00	4517.50	0.00	0.01
	R=100	4817.10	0.00	4817.10	0.00	0.00
N=15	R=0.01	5567.60	0.00	5487.00	0.00	1.47
	R=0.05	6297.30	0.00	6097.90	0.00	3.27
	R=0.1	6134.30	0.00	5835.70	0.00	5.12
	R=0.5	9160.70	0.00	8466.00	0.00	8.21
	R=1	12826.90	0.00	11448.60	0.00	12.04
	R=5	9494.30	0.00	9343.90	0.00	1.61
	R=10	9494.80	0.00	9474.40	0.00	0.22
	R=50	10294.90	0.00	10294.00	0.00	0.01
	R=100	9539.70	0.00	9539.70	0.00	0.00
N=20	R=0.01	10224.10	2.58	10029.20	4.53	1.94
	R=0.05	10442.80	4.93	10150.20	14.01	2.88
	R=0.1	10526.00	3.18	10109.00	5.12	4.13
	R=0.5	15415.90	2.16	13900.50	0.09	10.90
	R=1	21803.00	3.71	19380.80	0.00	12.50
	R=5	16331.00	0.56	16139.90	0.00	1.18
	R=10	16627.70	0.31	16568.90	0.00	0.35
	R=50	16370.70	0.14	16368.70	0.00	0.01
	R=100	16519.50	0.15	16517.80	0.00	0.01

Table 5.5: Results for Comparison of Heuristic Algorithms with Best LB – 2/2

Data Type		Heuristic		Best Lower Bound		Perc.Dev. of H&LB
		Obj.Val	Time(sec.)	Obj.Val	Time(sec.)	
N=30	R=0.01	20233.30	18.53	19762.20	538.26	2.38
	R=0.05	21662.90	9.99	20875.60	601.18	3.77
	R=0.1	23647.10	8.86	22362.80	841.46	5.74
	R=0.5	33334.40	6.05	29882.20	0.36	11.55
	R=1	46675.40	8.93	41442.20	0.00	12.63
	R=5	37398.70	5.34	36978.00	0.00	1.14
	R=10	35885.10	12.76	35780.30	0.00	0.29
	R=50	37564.90	0.68	37564.90	0.00	0.00
	R=100	36332.00	0.47	36332.00	0.00	0.00
N=35	R=0.01	27895.10	16.74	27001.50	3160.29	3.31
	R=0.05	29694.00	15.25	28362.50	1366.08	4.69
	R=0.1	29979.00	11.24	28207.80	2353.95	6.28
	R=0.5	46036.40	15.97	41929.70	0.62	9.79
	R=1	62928.60	10.03	55277.20	0.00	13.84
	R=5	51771.30	14.12	51236.50	0.00	1.04
	R=10	50562.80	6.76	50349.90	0.00	0.42
	R=50	48533.30	0.99	48523.40	0.00	0.02
	R=100	51699.60	0.75	51699.60	0.00	0.00
N=50	R=0.01	53753.20	16.42	51650.50	0.00	4.07
	R=0.05	57286.90	32.01	54526.00	0.00	5.06
	R=0.1	62691.70	19.44	58228.10	0.00	7.67
	R=0.5	86474.40	14.90	77762.00	0.00	11.20
	R=1	124317.00	24.60	109587.00	0.00	13.44
	R=5	99440.70	16.04	98195.70	0.00	1.27
	R=10	99401.40	5.15	98971.60	0.00	0.43
	R=50	98804.80	1.44	98796.90	0.00	0.01
	R=100	97358.40	1.02	97358.40	0.00	0.00
N=100	R=0.01	212591.80	76.01	204295.00	0.00	4.06
	R=0.05	219715.40	77.11	208179.60	0.00	5.54
	R=0.1	231620.60	82.91	216014.90	0.00	7.22
	R=0.5	339381.20	83.53	305772.40	0.00	10.99
	R=1	491823.90	117.79	436446.30	0.00	12.69
	R=5	412288.20	51.89	408884.00	0.00	0.83
	R=10	400471.20	32.00	399532.50	0.00	0.23
	R=50	401187.40	20.19	401115.10	0.00	0.02
	R=100	400724.70	10.19	400724.70	0.00	0.00

When the results are examined, increment on the percentage deviation depending on N is observed first. Since, both the lower bound and heuristic move away from

optimal solution when N gets larger, this result is an expected one. In Figure 5.8, average and maximum percentage deviation for each N is given.

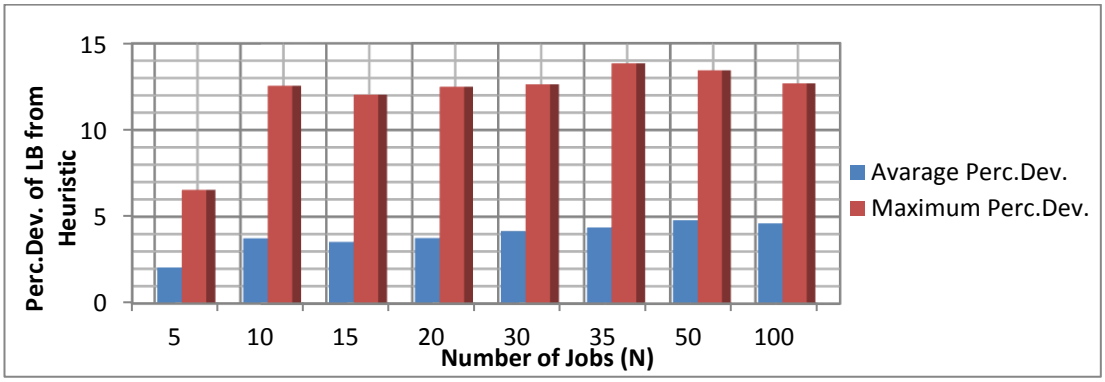

Figure 5.8: Maximum and Average Percentage Deviation of Best Lower Bound from Heuristic according to N.

Another important point seen in Figure 5.8 is the large difference between average and maximum percentage deviations for different N values. This result originates from the effect of *R* (setup/process ratio). As we mentioned before, for the small and large *R* values both the heuristic and lower bound gives near optimal solutions. However, for the interval *R*=[0,5-1] the problem is more difficult. Hence the performance of the heuristic in comparison to the best lower bound gets worse in this interval. Figure 5.9 illustrates the performance of the heuristic as a function of *R*. It is observed that worst performance of the heuristic is seen for *R*=0.5 and R=1.

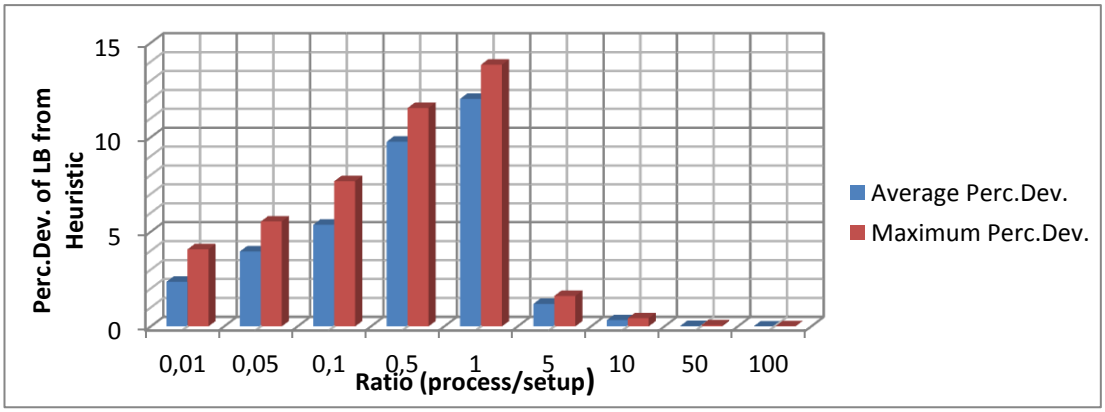

Figure 5.9: Maximum and Average Percentage Deviation of Best Lower Bound from Heuristic according to R.

Running times of the best lower bound and heuristic are negligible for up to 15 jobs. When we analyze the running times of the heuristic and the best lower bound for larger instances, in Figure 5.10 we observe that for *R*=0.01, 0.05 and 0.1 values, running times of the best lower bound are very large relative to the running times of the heuristic. This is because, lower bound 4 is the tightest one in this interval and it is an integer programming based bound. However, for the instances with N=50 and N=100 jobs, running times of the best lower bound are also negligible since lower bound 5, which is an LP relaxation of the IP model, is computed instead of lower bound 4 for these instances.

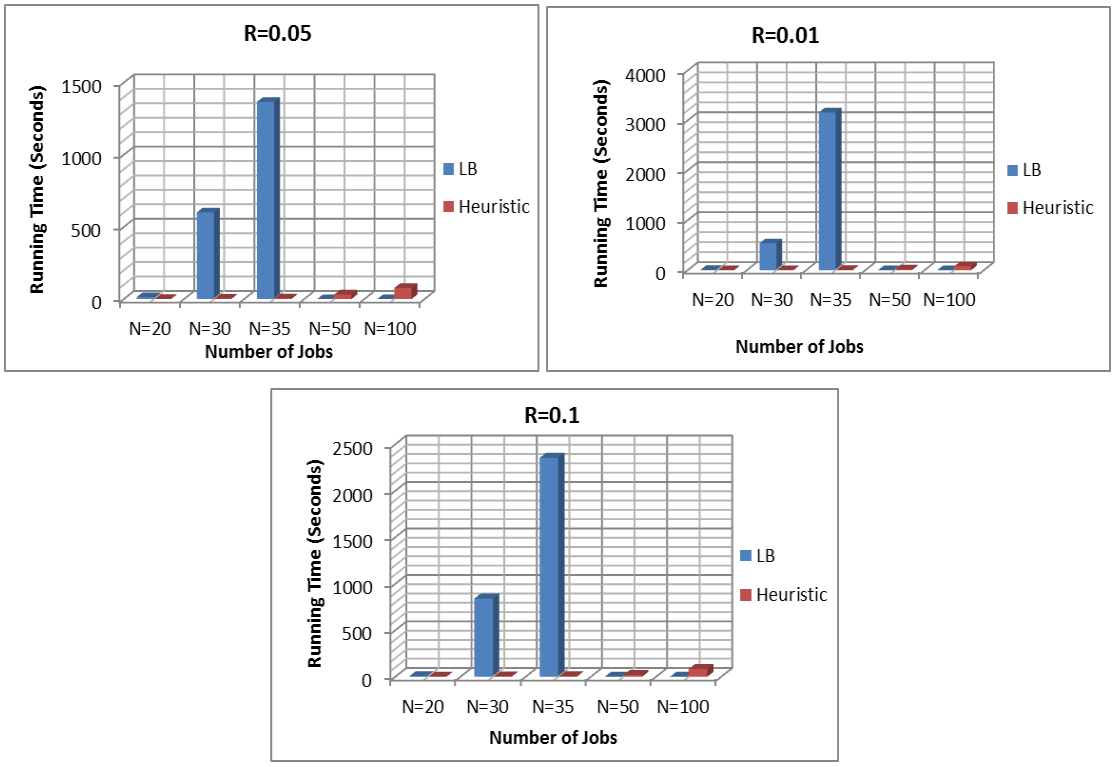

Figure 5.10: Comparison of Running Times of LB and Heuristic for *R*=0.01, *R*=0.05 and *R*=0.1

5.2.5 Comparison of Heuristic Algorithms (VNS 1 vs VNS 2)

As we mentioned before, two VNS algorithms are defined. The only difference between them is that, the number of search trials for each job is limited to N for each search method in VNS 2, where it is unlimited in VNS 1. In Table 5.6, 5.7, percentage deviation of VNS-1 and VNS-2 from the best lower bound is presented. There is not so much difference between results for VNS-1 and VNS-2. This is

because it is possible to reach the same local optimal points with different search orders among the neighborhood structures. However, we observe that for some instances, VNS 2 gives better solution than VNS 1. Since VNS 2 searches orders which may not be searched by VNS 1, this situation is possible. VNS 2 is forced to generate different orders according to limited number of search trials. Figure 5.11 visually presents a comparison of VNS 1 with VNS-2. It can be seen clearly that, average percentage gap of best lower bound for VNS 1 and VNS 2 is the same at most points.

Moreover, VNS 2 beats VNS 1 in terms of the running times. It is an expected result since VNS 2 has much fewer search trials than VNS 1. Figure 5.12 shows the average running times of the algorithms as a function of N.

Table 5.6: Comparison of VNS 1 and VNS 2 according to Best Lower Bound – 1/2

Data Type		VNS-1		VNS-2		Best Lower Bound	Perc.Dev.of	
		Obj.Val	Time (sec.)	Obj.Val	Time (sec.)		VNS1 from LB	VNS2 from LB
N=5	R=0.01	893.70	0.01	893.70	0.00	890.60	0.35	0.35
	R=0.05	937.90	0.00	937.90	0.00	919.20	2.03	2.03
	R=0.1	912.20	0.00	912.20	0.00	891.30	2.34	2.34
	R=0.5	1193.30	0.00	1193.30	0.00	1128.70	5.72	5.72
	R=1	1734.90	0.00	1734.90	0.00	1628.30	6.55	6.55
	R=5	1317.10	0.00	1317.10	0.00	1299.00	1.39	1.39
	R=10	1395.60	0.00	1395.60	0.00	1392.00	0.26	0.26
	R=50	1261.40	0.00	1261.40	0.00	1260.60	0.06	0.06
	R=100	1092.90	0.00	1092.90	0.00	1092.90	0.00	0.00
N=10	R=0,01	2927.40	0.04	2932.30	0.03	2889.30	1.32	1.49
	R=0,05	3011.60	0.03	3014.90	0.02	2882.50	4.48	4.59
	R=0,1	3076.00	0.03	3074.00	0.02	2941.50	4.57	4.50
	R=0,5	4530.30	0.03	4518.80	0.02	4113.30	10.14	9.86
	R=1	6268.20	0.03	6280.90	0.02	5566.00	12.62	12.84
	R=5	4691.50	0.02	4691.50	0.01	4643.70	1.03	1.03
	R=10	4464.60	0.01	4464.60	0.01	4446.30	0.41	0.41
	R=50	4518.10	0.01	4518.10	0.01	4517.50	0.01	0.01
	R=100	4817.10	0.01	4817.10	0.01	4817.10	0.00	0.00
N=15	R=0,01	5611.60	0.20	5592.10	0.11	5487.00	2.27	1.92
	R=0,05	6313.20	0.15	6305.60	0.08	6097.90	3.53	3.41
	R=0,1	6149.10	0.21	6154.60	0.08	5835.70	5.37	5.46
	R=0,5	9184.30	0.09	9184.90	0.08	8466.00	8.48	8.49
	R=1	12851.10	0.12	12874.70	0.08	11448.60	12.25	12.46
	R=5	9494.30	0.07	9494.30	0.04	9343.90	1.61	1.61
	R=10	9494.80	0.04	9494.80	0.03	9474.40	0.22	0.22
	R=50	10294.90	0.03	10294.90	0.02	10294.00	0.01	0.01
	R=100	9539.70	0.02	9539.70	0.02	9539.70	0.00	0.00
N=20	R=0,01	10263.10	0.50	10255.70	0.30	10029.20	2.33	2.26
	R=0,05	10504.60	0.62	10461.60	0.25	10150.20	3.49	3.07
	R=0,1	10585.70	0.46	10543.30	0.21	10109.00	4.72	4.30
	R=0,5	15453.60	0.45	15469.50	0.19	13900.50	11.17	11.29
	R=1	21892.40	0.45	21896.60	0.22	19380.80	12.96	12.98
	R=5	16331.00	0.29	16332.40	0.11	16139.90	1.18	1.19
	R=10	16627.70	0.15	16627.70	0.09	16568.90	0.35	0.35
	R=50	16370.70	0.06	16370.70	0.06	16368.70	0.01	0.01
	R=100	16519.50	0.06	16519.50	0.06	16517.80	0.01	0.01

Table 5.7: Comparison of VNS 1 and VNS 2 according to Best Lower Bound – 2/2

Data Type		VNS-1		VNS-2		Best Lower Bound	Perc.Dev.of	
		Obj.Val	Time (sec.)	Obj.Val	Time (sec.)		VNS1 from LB	VNS2 from LB
N=30	R=0.01	20284.50	3.80	20297.30	1.01	19762.20	2.64	2.71
	R=0.05	21724.60	3.66	21750.70	0.96	20875.60	4.07	4.19
	R=0.1	23694.10	3.14	23757.40	0.73	22362.80	5.95	6.24
	R=0.5	33523.40	2.00	33469.00	0.83	29882.20	12.19	12.00
	R=1	46896.30	2.68	46895.40	0.72	41442.20	13.16	13.16
	R=5	37399.80	1.97	37413.50	0.41	36978.00	1.14	1.18
	R=10	35885.10	1.03	35885.10	0.39	35780.30	0.29	0.29
	R=50	37564.90	0.29	37564.90	0.23	37564.90	0.00	0.00
	R=100	36332.00	0.20	36332.00	0.20	36332.00	0.00	0.00
N=35	R=0.01	27977.80	7.34	27950.60	2.00	27001.50	3.62	3.51
	R=0.05	29747.50	6.51	29821.00	1.58	28362.50	4.88	5.14
	R=0.1	30017.20	5.25	30058.30	1.53	28207.80	6.41	6.56
	R=0.5	46139.10	5.99	46195.10	1.56	41929.70	10.04	10.17
	R=1	63259.70	4.99	63202.60	1.42	55277.20	14.44	14.34
	R=5	51772.30	5.51	51772.20	0.83	51236.50	1.05	1.05
	R=10	50562.80	1.73	50562.80	0.62	50349.90	0.42	0.42
	R=50	48533.30	0.00	48533.30	0.37	48523.40	0.02	0.02
	R=100	51699.60	0.00	51699.60	0.32	51699.60	0.00	0.00
N=50	R=0.01	54002.92	30.52	53839.00	8.06	51650.50	4.55	4.24
	R=0.05	57435.65	45.65	57446.20	6.61	54526.00	5.34	5.36
	R=0.1	62885.58	34.38	62981.20	5.25	58228.10	8.00	8.16
	R=0.5	86666.30	21.20	86745.30	5.39	77762.00	11.45	11.55
	R=1	124923.40	34.10	124923.10	7.08	109587.00	13.99	13.99
	R=5	99493.83	16.33	99516.70	4.24	98195.70	1.32	1.35
	R=10	99418.23	14.73	99404.70	2.86	98971.60	0.45	0.44
	R=50	98807.10	2.30	98804.80	1.44	98796.90	0.01	0.01
	R=100	97359.45	1.05	97358.40	1.02	97358.40	0.00	0.00

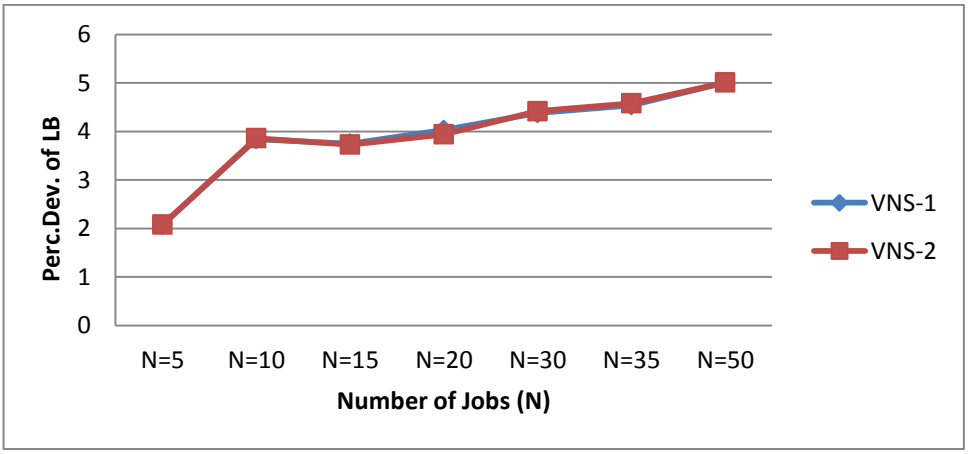

Figure 5.11: Comparison of VNS 1 and VNS 2 according to N

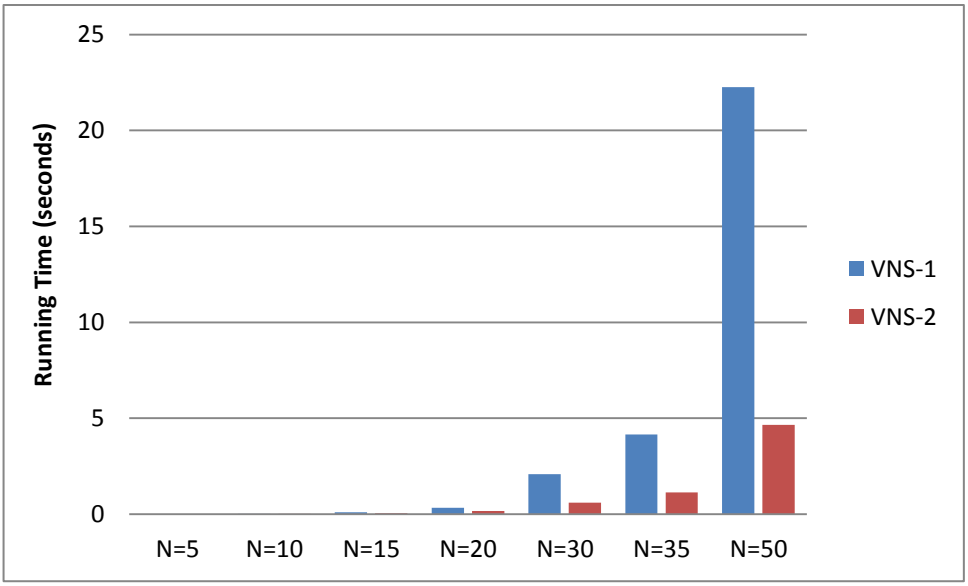

Figure 5.12: Average Running Times of the VNS 1 and VNS 2 according to N.

5.2.6 General Discussion

First we observe that, it is not possible to get optimal solutions for instances with more than 10 jobs with our MIP model. Hence, we develop heuristic algorithms to solve larger instances in a reasonable time. Even though we can solve large problems with more than 100 jobs, we observe that the effectiveness of the heuristic deteriorates as does the instance size. This is an expected result due to the summative nature of the objective function.

For specific *R* values some of the lower bounds outperform others. One of them is lower bound 4, which gives better results for $R \leq 0.1$ and the other one is lower bound 2 which gives better results for$R \geq 5$. This stems from the fact that, lower bound 4 primarily considers process times and lower bound 2 primarily considers setup times.

The solutions of the instances indicate that, heuristic algorithm gives near optimal solutions for the intervals $R \leq 0.1$ and $R \geq 5$. However, for $R=0.5$ and $R=1$, performance of the heuristic deteriorates. This result arises partly from the poor performance of the best lower bound in these *R* values. This observation can also be seen in the results of the small sized instances for which we have optimum solutions. In these instances, the performance of the heuristic is not bad in

comparison to the optimum, however, that of the best lower bound is very loose when there is more avoidable idle times as is the case when R=0.5 and R=1 for which setup times and processing times are close to each other.

Chapter 6

Conclusion and Future Research

In this thesis, we considered a two stage flow shop problem with a single server with the objective of minimizing total completion time. The problem is known to be NP-hard in the strong sense [3]. We formulated the problem as an MIP, proposed several lower bounds, and developed a constructive greedy mechanism and two versions of a Variable Neighborhood Search as heuristics. We also identified a special case of the problem that can be solved in polynomial time.

The proposed heuristic algorithms are tested via computational experimentation considering different N (number of jobs) and $R = (\bar{s}/\bar{p})$ (average ratio of setup times to process times) values. Results show that in performance of both the lower bounds and the heuristic algorithms get worse as the problem size increases in terms of number of jobs. This is due to the fact that the objective function considers a sum over all jobs and there are simply more terms with deviations to add up in larger instances. Also, solutions of the instances indicate that, for small R values lower bound 4 and for large R values lower bound 2 outperforms other lower bounds and gives near optimal solutions.

To test the performance of the heuristics, their results are compared with optimal solutions in small instances with lower bounds in larger ones. Results show that, especially for small and large R values VNS gives near optimal solutions. For

the instances in which process times and setup times are near each other, performance of the heuristic deteriorates as there is more avoidable idle time on both machines and the server in such cases.

As a future research, it may be worthwhile to seek better lower bounds and more effective heuristics for problems with *R* values in the range of 0.5 and 1.0. Also, since this problem is studied with the makespan and total completion time objectives so far, consideration of other objective functions is an obvious direction for future research. Another possible but challenging direction is to study a multiple machine version of the problem in which there are m machines in the flow shop setting.

Bibliography

[1] Lim A, Rodrigues B, Wang C. Two-machine flow shop problems with a single server. Journal of Scheduling 2006, 9: 515-543.

[2] Johnson SM. Optimal two and three stages production schedules with setup times included. Nav Res Logistics 1954, Quart 1: 61-68.

[3] Brucker P, Knust S, Wang G. Complexity results for flow-shop problems with a single server. European Journal of Operational Research 2005, 165: 398-407.

[4] Pinedo ML, Scheduling: Theory Algorithms and Systems. Springer Science, New York, 2008.

[5] Garey MR, Johnson DS, Sethi R. The complexity of flow shop and job shop scheduling. Mathematics of Operations Research 1976, 1: 117-129.

[6] Cheng TCE, Gupta JND, Wang G. A review of flow shop scheduling research with setup times. Production and Operations Management 2000, 9: 262-282.

[7] Allahverdi A, Ng CT, Cheng TCE, Kovalyov MY. A survey of scheduling problems with setup times or costs. European Journal of Operational Research 2008, 187: 985-1032.

[8] Allahverdi A, Gupta JND, Aldowaisan T. A review of scheduling research involving setup considerations. Omega, International Journal Management Science 1999, 27: 219-239.

[9] Koulamas CP. Scheduling two parallel semiautomatic machines to minimize machine interference. Computers and Operations Research 1996, Volume 23: 945-956.

[10] Kravchenko SA, Werner F. Parallel machine scheduling problems with a single server. Mathematical Computational Modelling 1997, 26: 1-11.

[11] Hall N, Potts C, Sriskandarajah. Parallel machine scheduling with a common server. Discrete Applied Mathematics 2000, 102: 223-243.

[12] Brucker P, Dhaenens-Flipo C, Knust S, Kravchenko SA, Werner F. Complexity results for parallel machine problems with a single server. Journal of Scheduling 2002, 5: 429-457.

[13] Abdekhodaee AH, Wirth A, Gan HS. Equal processing and equal setup time cases of scheduling parallel machines with a single server. Computers and Operations Research 2004, 31: 1867-1889.

[14] Glass CA, Shafransky YM, Strusevich VA. Scheduling for parallel dedicated machines with a single server. Naval Research Logistics 2000, 47: 304-328.

[15] Kravchenko SA, Werner F. A heuristic algorithm for minimizing mean flow time with unit setups. Information Processing Letters 2001, 6: 291-296.

[16] Cheng TCE, Wang G. An approximation algorithm for parallel machine scheduling with a common server. Journal of the Operational Research Society 2001, 52: 234-237.

[17] Abdekhodaee AH, Wirth A. Scheduling parallel machines with a single server: some solvable cases and heuristics. Computers and Operations Research 2002, 29: 295-315.

[18] Abdekhodaee AH, Wirth A, Gan HS. Scheduling two parallel machines with a single server: the general case. Computers and Operations Research 2006, 33: 994-1009.

[19] Yoshida T, Hitomi K. Optimal two stage production scheduling with setup times separated. AIIE Transactions 1979, 11: 261-263.

[20] Cheng TCE, Wang G, Sriskandarajah. One operator-two machine flow shop scheduling with setup and dismounting times. Computers and Operations Research 1999, 26: 715-730.

[21] Cheng, T.C.E., Kovalyov, M.Y.,2003, Scheduling a single server in a two-machine flow shop. Computing 2003, 70: 167-180.

[22] Su LH, Lee YY. The two machine flowshop no-wait scheduling problem with a single server to minimize the total completion time. Computers and Operations Research 2000, 35: 2952-2963.

[23] Hoogeveen H, Norden L, Velde S. Lower bounds for minimizing total completion time in a two machine flow shop. Journal of Scheduling 2006, 9: 559-568.

[24] Hansen P, Mladenovic N. Variable Neighborhood Search. Computers and Operations Research 1997, 11: 1097-1100.

Appendix A

Computational Results

Table A.1: Computational Results for N=5 – 1/3

N=5		Optimal Soln.		LB 1	LB 2	LB 3	LB 4	Best Lower Bound	VNS 1		VNS 2		Min of VNS		Perc.Dev.of		
Data Type	No	Obj. Val.	Time (sec.)						Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Opt.Soln. from LB	Opt.Soln. from VNS	Best LB from VNS
R=0,01	1	908	0.61	620	301	799	905	905	908	0.00	908	0.00	908	0.00	0.33	0.00	0.33
	2	985	0.86	733	347	798	981	981	985	0.02	985	0.00	985	0.00	0.41	0.00	0.41
	3	1327	1.74	514	275	1316	1327	1327	1327	0.00	1327	0.00	1327	0.00	0.00	0.00	0.00
	4	662	1.91	632	299	512	656	656	662	0.00	662	0.00	662	0.00	0.91	0.00	0.91
	5	947	2.48	630	292	911	946	946	949	0.02	949	0.00	949	0.00	0.11	0.21	0.32
	6	766	0.42	597	291	668	765	765	770	0.00	770	0.00	770	0.00	0.13	0.52	0.65
	7	730	0.72	594	273	568	726	726	730	0.00	730	0.00	730	0.00	0.55	0.00	0.55
	8	798	1.27	582	258	781	797	797	802	0.02	802	0.00	802	0.00	0.13	0.50	0.63
	9	899	2.02	439	224	871	898	898	899	0.02	899	0.00	899	0.00	0.11	0.00	0.11
	10	905	2.40	494	238	905	905	905	905	0.00	905	0.00	905	0.00	0.00	0.00	0.00
R=0,05	1	1061	0.33	931	405	966	1054	1054	1061	0.00	1061	0.00	1061	0.00	0.66	0.00	0.66
	2	955	0.70	685	315	901	942	942	964	0.00	964	0.00	964	0.00	1.36	0.94	2.34
	3	895	1.24	521	257	895	895	895	895	0.02	895	0.00	895	0.00	0.00	0.00	0.00
	4	944	1.53	892	432	737	918	918	944	0.00	944	0.00	944	0.00	2.75	0.00	2.83
	5	798	1.98	569	310	747	783	783	798	0.00	798	0.00	798	0.00	1.88	0.00	1.92
	6	791	0.32	687	351	747	775	775	791	0.00	791	0.00	791	0.00	2.02	0.00	2.06
	7	1062	0.89	883	407	810	1049	1049	1079	0.00	1079	0.00	1079	0.00	1.22	1.60	2.86
	8	1067	1.26	823	417	972	1061	1061	1093	0.02	1093	0.00	1093	0.00	0.56	2.44	3.02
	9	1115	1.64	769	403	1006	1106	1106	1115	0.00	1115	0.00	1115	0.00	0.81	0.00	0.81
	10	623	1.98	462	261	588	609	609	639	0.00	639	0.00	639	0.00	2.25	2.57	4.93
R=0,1	1	1023	0.75	724	406	918	983	983	1024	0.00	1024	0.02	1024	0.02	3.91	0.10	4.17
	2	1198	1.42	581	376	1193	1194	1194	1198	0.00	1198	0.00	1198	0.00	0.33	0.00	0.34
	3	1023	1.80	741	493	977	1016	1016	1023	0.00	1023	0.00	1023	0.00	0.68	0.00	0.69
	4	811	2.66	474	306	790	789	790	811	0.00	811	0.00	811	0.00	2.59	0.00	2.66
	5	912	3.11	672	441	773	882	882	912	0.00	912	0.00	912	0.00	3.29	0.00	3.40
	6	857	0.42	562	319	857	857	857	857	0.00	857	0.00	857	0.00	0.00	0.00	0.00
	7	886	1.24	464	364	850	869	869	886	0.00	886	0.00	886	0.00	1.92	0.00	1.96
	8	974	1.58	859	425	802	927	927	974	0.00	974	0.00	974	0.00	4.83	0.00	5.07
	9	726	1.98	539	336	635	692	692	734	0.00	734	0.00	734	0.00	4.68	1.10	6.07
	10	703	2.64	261	213	703	703	703	703	0.00	703	0.00	703	0.00	0.00	0.00	0.00

Table A.2: Computational Results for N=5 – 2/3

N=5		Optimal Soln.		LB 1	LB 2	LB 3	LB 4	Best Lower Bound	VNS 1		VNS 2		Min of VNS		Perc.Dev.of		
Data Type	No	Obj. Val.	Time (sec.)						Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Opt.Soln. from LB	Opt.Soln. from VNS	Best LB from VNS
R=0,5	1	1155	0.70	835	692	1090	1087	1090	1155	0.00	1155	0.00	1155	0.00	5.63	0.00	5.96
	2	1204	3.35	857	870	1083	1082	1083	1204	0.00	1204	0.00	1204	0.00	10.05	0.00	11.17
	3	1353	4.83	1010	982	1275	1259	1275	1353	0.00	1353	0.00	1353	0.00	5.76	0.00	6.12
	4	1190	6.20	732	741	1165	1165	1165	1190	0.00	1190	0.00	1190	0.00	2.10	0.00	2.15
	5	1237	7.24	643	551	1231	1201	1231	1237	0.02	1237	0.00	1237	0.00	0.49	0.00	0.49
	6	1138	1.41	813	794	1047	1018	1047	1138	0.00	1138	0.00	1138	0.00	8.00	0.00	8.69
	7	1140	2.25	431	629	1140	1140	1140	1140	0.00	1140	0.00	1140	0.00	0.00	0.00	0.00
	8	1473	2.48	1473	970	1228	1224	1473	1473	0.00	1473	0.00	1473	0.00	0.00	0.00	0.00
	9	894	3.15	821	698	653	626	821	894	0.00	894	0.00	894	0.00	8.17	0.00	8.89
	10	1149	5.73	830	886	962	936	962	1149	0.00	1149	0.00	1149	0.00	16.28	0.00	19.44
R=1	1	1845	0.70	1688	1645	1685	1644	1688	1845	0.00	1845	0.02	1845	0.02	8.51	0.00	9.30
	2	1904	3.50	1311	1702	1738	1667	1738	1904	0.00	1904	0.00	1904	0.00	8.72	0.00	9.55
	3	1460	4.35	1077	1216	1406	1388	1406	1460	0.00	1460	0.00	1460	0.00	3.70	0.00	3.84
	4	1215	5.13	1034	1012	981	910	1034	1215	0.00	1215	0.00	1215	0.00	14.90	0.00	17.50
	5	1577	6.71	1245	1469	1422	1260	1469	1577	0.00	1577	0.00	1577	0.00	6.85	0.00	7.35
	6	1969	0.71	1612	1843	1856	1785	1856	1969	0.02	1969	0.00	1969	0.00	5.74	0.00	6.09
	7	1963	2.64	1445	1603	1836	1584	1836	1963	0.00	1963	0.00	1963	0.00	6.47	0.00	6.92
	8	1608	3.09	1284	1429	1584	1545	1584	1608	0.00	1608	0.00	1608	0.00	1.49	0.00	1.52
	9	1801	3.99	1665	1657	1637	1543	1665	1801	0.00	1801	0.00	1801	0.00	7.55	0.00	8.17
	10	2007	5.92	1215	1479	2007	2007	2007	2007	0.00	2007	0.00	2007	0.00	0.00	0.00	0.00
R=5	1	1498	3.91	812	1498	1431	1270	1498	1498	0.00	1498	0.00	1498	0.00	0.00	0.00	0.00
	2	864	8.14	525	864	803	611	864	864	0.00	864	0.00	864	0.00	0.00	0.00	0.00
	3	1530	15.41	881	1530	1392	834	1530	1530	0.00	1530	0.00	1530	0.00	0.00	0.00	0.00
	4	1010	18.46	607	996	849	550	996	1010	0.00	1010	0.00	1010	0.00	1.39	0.00	1.41
	5	915	23.18	441	864	810	755	864	915	0.00	915	0.02	915	0.02	5.57	0.00	5.90
	6	2015	7.58	1277	2015	1855	1201	2015	2015	0.00	2015	0.00	2015	0.00	0.00	0.00	0.00
	7	1691	16.44	1239	1691	1645	938	1691	1691	0.00	1691	0.00	1691	0.00	0.00	0.00	0.00
	8	1333	19.93	818	1256	1114	613	1256	1333	0.02	1333	0.00	1333	0.00	5.78	0.00	6.13
	9	1074	23.68	629	1034	1041	713	1041	1074	0.00	1074	0.00	1074	0.00	3.07	0.00	3.17
	10	1241	32.23	740	1235	1068	654	1235	1241	0.00	1241	0.00	1241	0.00	0.48	0.00	0.49

Table A.3: Computational Results for N=5 – 3/3

N=5		Optimal Soln.		LB 1	LB 2	LB 3	LB 4	Best Lower Bound	VNS 1		VNS 2		Min of VNS		Perc.Dev.of		
Data Type	No	Obj. Val.	Time (sec.)						Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Opt.Soln. from LB	Opt.Soln. from VNS	Best LB from VNS
R=10	1	1568	16.88	760	1568	1458	978	1568	1568	0.00	1568	0.00	1568	0.00	0.00	0.00	0.00
	2	1284	23.63	769	1284	1257	649	1284	1284	0.00	1284	0.00	1284	0.00	0.00	0.00	0.00
	3	1318	38.43	534	1318	1028	694	1318	1318	0.00	1318	0.00	1318	0.00	0.00	0.00	0.00
	4	1611	48.72	916	1611	1608	913	1611	1611	0.00	1611	0.00	1611	0.00	0.00	0.00	0.00
	5	1929	66.99	1204	1929	1853	856	1929	1929	0.00	1929	0.00	1929	0.00	0.00	0.00	0.00
	6	1086	6.20	571	1066	1064	670	1066	1086	0.00	1086	0.00	1086	0.00	1.84	0.00	1.88
	7	1964	20.43	1294	1964	1795	800	1964	1964	0.02	1964	0.00	1964	0.00	0.00	0.00	0.00
	8	947	28.45	316	947	815	677	947	947	0.00	947	0.00	947	0.00	0.00	0.00	0.00
	9	1472	35.68	866	1472	1456	758	1472	1472	0.00	1472	0.00	1472	0.00	0.00	0.00	0.00
	10	777	42.15	428	761	657	421	761	777	0.00	777	0.02	777	0.02	2.06	0.00	2.10
R=50	1	1490	11.13	722	1490	1357	767	1490	1490	0.00	1490	0.00	1490	0.00	0.00	0.00	0.00
	2	1434	24.74	678	1434	1357	858	1434	1434	0.00	1434	0.00	1434	0.00	0.00	0.00	0.00
	3	1525	56.01	731	1525	1294	699	1525	1525	0.00	1525	0.00	1525	0.00	0.00	0.00	0.00
	4	1197	65.13	743	1197	1140	464	1197	1197	0.00	1197	0.00	1197	0.00	0.00	0.00	0.00
	5	1514	73.70	899	1514	1490	764	1514	1514	0.00	1514	0.00	1514	0.00	0.00	0.00	0.00
	6	848	7.57	652	848	792	235	848	848	0.02	848	0.00	848	0.00	0.00	0.00	0.00
	7	1227	16.76	709	1227	1220	647	1227	1227	0.00	1227	0.00	1227	0.00	0.00	0.00	0.00
	8	1413	36.53	603	1413	1159	648	1413	1413	0.00	1413	0.00	1413	0.00	0.00	0.00	0.00
	9	972	42.11	570	972	892	368	972	972	0.00	972	0.00	972	0.00	0.00	0.00	0.00
	10	994	48.61	585	986	917	387	986	994	0.00	994	0.00	994	0.00	0.80	0.00	0.81
R=100	1	1134	6.83	375	1134	1047	817	1134	1134	0.00	1134	0.00	1134	0.00	0.00	0.00	0.00
	2	1254	13.37	605	1254	1151	581	1254	1254	0.00	1254	0.00	1254	0.00	0.00	0.00	0.00
	3	665	17.54	364	665	659	360	665	665	0.00	665	0.00	665	0.00	0.00	0.00	0.00
	4	929	21.24	729	929	887	198	929	929	0.00	929	0.00	929	0.00	0.00	0.00	0.00
	5	1267	25.97	576	1267	1242	806	1267	1267	0.00	1267	0.02	1267	0.02	0.00	0.00	0.00
	6	1071	9.02	349	1071	879	630	1071	1071	0.00	1071	0.00	1071	0.00	0.00	0.00	0.00
	7	934	13.15	677	934	927	290	934	934	0.00	934	0.00	934	0.00	0.00	0.00	0.00
	8	1245	24.93	605	1245	1077	542	1245	1245	0.00	1245	0.00	1245	0.00	0.00	0.00	0.00
	9	968	35.10	546	968	875	409	968	968	0.00	968	0.00	968	0.00	0.00	0.00	0.00
	10	1462	43.87	806	1462	1453	862	1462	1462	0.00	1462	0.00	1462	0.00	0.00	0.00	0.00

Table A.4: Computational Results for N=8 – 1/3

N=8		Optimal Soln.		LB 1	LB 2	LB 3	LB 4	Best Lower Bound	VNS 1		VNS 2		Min of VNS		Perc.Dev.of		
Data Type	No	Obj. Val.	Time (sec.)						Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec,)	Opt.Soln. from LB	Opt.Soln. from VNS	Best LB from VNS
R=0,01	1	1902	16.36	1259	452	1753	1880	1880	1902	0.05	1902	0.03	1902	0.03	1.16	0.00	1.17
	2	2145	3.10	2003	614	1391	2117	2117	2166	0.00	2166	0.02	2166	0.02	1.31	0.98	2.31
	3	2072	11.51	1363	498	2001	2067	2067	2088	0.03	2088	0.02	2088	0.02	0.24	0.77	1.02
	4	1851	26.15	1130	439	1727	1843	1843	1855	0.05	1855	0.02	1855	0.02	0.43	0.22	0.65
	5	1539	13.77	1010	420	1372	1533	1533	1539	0.05	1539	0.02	1539	0.02	0.39	0.00	0.39
	6	1762	18.48	1410	502	1526	1750	1750	1762	0.02	1762	0.02	1762	0.02	0.68	0.00	0.69
	7	1454	19.99	1221	440	1149	1433	1433	1513	0.02	1513	0.00	1513	0.00	1.44	4.06	5.58
	8	2010	7.04	1846	579	1487	1982	1982	2017	0.02	2024	0.02	2017	0.02	1.39	0.35	1.77
	9	1772	13.47	1573	462	1469	1751	1751	1772	0.00	1772	0.00	1772	0.00	1.19	0.00	1.20
	10	2039	51.64	1603	527	1833	2022	2022	2039	0.02	2048	0.02	2039	0.02	0.83	0.00	0.84
R=0,05	1	2412	6.43	2070	684	2167	2348	2348	2418	0.02	2418	0.02	2418	0.02	2.65	0.25	2.98
	2	1620	4.35	1342	623	1284	1570	1570	1620	0.02	1620	0.02	1620	0.02	3.09	0.00	3.18
	3	1876	4.34	1801	644	1163	1786	1801	1884	0.02	1884	0.02	1884	0.02	4.00	0.43	4.61
	4	1786	15.90	1122	516	1697	1768	1768	1809	0.02	1809	0.00	1809	0.00	1.01	1.29	2.32
	5	1609	3.60	1548	548	1213	1542	1548	1643	0.02	1624	0.02	1624	0.02	3.79	0.93	4.91
	6	2111	54.09	1811	676	1648	2061	2061	2111	0.00	2111	0.00	2111	0.00	2.37	0.00	2.43
	7	2329	20.15	2185	720	1935	2238	2238	2344	0.03	2344	0.02	2344	0.02	3.91	0.64	4.74
	8	1941	89.11	1273	545	1874	1925	1925	1941	0.00	1941	0.02	1941	0.02	0.82	0.00	0.83
	9	3156	18.06	2864	791	2777	3100	3100	3164	0.05	3202	0.02	3164	0.05	1.77	0.25	2.06
	10	1899	5.74	1829	607	1668	1858	1858	1916	0.02	1916	0.00	1916	0.00	2.16	0.90	3.12
R=0,1	1	2088	4.06	1877	808	1709	1942	1942	2088	0.02	2088	0.02	2088	0.02	6.99	0.00	7.52
	2	1946	8.80	1695	761	1732	1847	1847	2009	0.02	2009	0.02	2009	0.02	5.09	3.24	8.77
	3	2045	13.51	1594	750	1840	1953	1953	2068	0.02	2066	0.02	2066	0.02	4.50	1.03	5.79
	4	2974	84.74	1432	711	2941	2968	2968	2974	0.03	2980	0.02	2974	0.03	0.20	0.00	0.20
	5	2069	3.20	1776	788	2010	2022	2022	2069	0.02	2089	0.02	2069	0.02	2.27	0.00	2.32
	6	2401	170.68	1325	671	2333	2384	2384	2409	0.03	2424	0.00	2409	0.03	0.71	0.33	1.05
	7	1834	58.11	1466	666	1556	1742	1742	1847	0.02	1847	0.02	1847	0.02	5.02	0.71	6.03
	8	2109	105.44	1469	777	1802	2043	2043	2115	0.02	2115	0.02	2115	0.02	3.13	0.28	3.52
	9	2101	157.43	1173	599	1962	2068	2068	2153	0.03	2152	0.02	2152	0.02	1.57	2.43	4.06
	10	2194	40.62	1905	770	2088	2144	2144	2194	0.00	2194	0.02	2194	0.02	2.28	0.00	2.33

Table A.5: Computational Results for N=8 – 2/3

N=8		Optimal Soln.		LB 1	LB 2	LB 3	LB 4	Best Lower Bound	VNS 1		VNS 2		Min of VNS		Perc.Dev.of		
Data Type	No	Obj. Val.	Time (sec.)						Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec,)	Opt.Soln. from LB	Opt.Soln. from VNS	Best LB from VNS
R=0,5	1	2959	9.25	2689	1921	2389	2600	2689	2959	0.02	2959	0.02	2959	0.02	9.12	0.00	10.04
	2	3206	153.41	2132	1997	3070	3051	3070	3208	0.02	3208	0.00	3208	0.00	4.24	0.06	4.50
	3	3523	64.26	2935	2716	2985	2970	2985	3523	0.02	3523	0.02	3523	0.02	15.27	0.00	18.02
	4	2701	17.99	2294	1659	2515	2629	2629	2705	0.00	2705	0.02	2705	0.02	2.67	0.15	2.89
	5	2893	143.93	2084	1669	2628	2632	2632	2895	0.02	2955	0.02	2895	0.02	9.02	0.07	9.99
	6	3085	343.67	2272	1915	2866	2839	2866	3085	0.02	3085	0.00	3085	0.00	7.10	0.00	7.64
	7	2557	363.10	1886	1308	2172	2145	2172	2634	0.00	2634	0.05	2634	0.05	15.06	3.01	21.27
	8	2739	101.83	2297	1725	2531	2552	2552	2739	0.02	2739	0.00	2739	0.00	6.83	0.00	7.33
	9	3461	55.12	3000	2472	3168	3153	3168	3504	0.02	3504	0.00	3504	0.00	8.47	1.24	10.61
	10	3042	362.32	2037	1837	2882	2903	2903	3042	0.03	3042	0.02	3042	0.02	4.57	0.00	4.79
R=1	1	4020	240.88	2937	2828	3553	3473	3553	4070	0.02	4070	0.02	4070	0.02	11.62	1.24	14.55
	2	3979	370.50	2596	3440	3513	3308	3513	3979	0.00	3979	0.00	3979	0.00	11.71	0.00	13.27
	3	3587	154.94	2418	2632	3182	3150	3182	3587	0.02	3587	0.02	3587	0.02	11.29	0.00	12.73
	4	3449	120.17	2836	2802	2930	2464	2930	3589	0.02	3589	0.00	3589	0.00	15.05	4.06	22.49
	5	4240	252.53	2975	3517	3973	3626	3973	4247	0.02	4247	0.00	4247	0.00	6.30	0.17	6.90
	6	3866	219.11	2913	3231	3525	2840	3525	4031	0.00	4031	0.02	4031	0.02	8.82	4.27	14.35
	7	2581	554.94	1741	2438	2231	1779	2438	2581	0.02	2581	0.00	2581	0.00	5.54	0.00	5.87
	8	4616	360.08	2740	3616	4394	4014	4394	4706	0.02	4656	0.00	4656	0.00	4.81	0.87	5.96
	9	4473	420.07	3227	3895	3803	3418	3895	4480	0.00	4473	0.02	4473	0.02	12.92	0.00	14.84
	10	3914	2.75	3764	3028	3046	2889	3764	3914	0.00	3914	0.00	3914	0.00	3.83	0.00	3.99
R=5	1	2998	1709.03	1622	2949	2503	1602	2949	2998	0.02	2998	0.00	2998	0.00	1.63	0.00	1.66
	2	3076	6481.33	1574	2969	2752	1798	2969	3076	0.00	3076	0.02	3076	0.02	3.48	0.00	3.60
	3	3589	1004.67	2136	3589	3545	2228	3589	3589	0.02	3589	0.00	3589	0.00	0.00	0.00	0.00
	4	3353	2718.79	2444	3319	3141	1442	3319	3353	0.00	3353	0.00	3353	0.00	1.01	0.00	1.02
	5	2174	745.01	986	2172	2037	1753	2172	2174	0.00	2174	0.00	2174	0.00	0.09	0.00	0.09
	6	3541	334.49	2195	3541	3328	2162	3541	3541	0.02	3541	0.02	3541	0.02	0.00	0.00	0.00
	7	2879	1734.97	2041	2755	2764	1546	2764	2879	0.00	2879	0.00	2879	0.00	3.99	0.00	4.16
	8	4087	667.91	2555	4087	3919	2174	4087	4087	0.00	4087	0.00	4087	0.00	0.00	0.00	0.00
	9	3137	727.52	1528	3074	2765	1961	3074	3137	0.02	3137	0.02	3137	0.02	2.01	0.00	2.05
	10	4420	964.39	2069	4420	4241	3342	4420	4420	0.00	4420	0.00	4420	0.00	0.00	0.00	0.00

Table A.6: Computational Results for N=8 – 3/3

N=8		Optimal Soln.		LB 1	LB 2	LB 3	LB 4	Best Lower Bound	VNS 1		VNS 2		Min of VNS		Perc.Dev.of		
Data Type	No	Obj. Val.	Time (sec.)						Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec,)	Opt.Soln. from LB	Opt.Soln. from VNS	Best LB from VNS
R=10	1	1995	3125.98	1336	1988	1759	787	1988	1995	0.00	1995	0.00	1995	0.00	0.35	0.00	0.35
	2	2939	1632.06	1446	2939	2754	1778	2939	2939	0.02	2939	0.00	2939	0.00	0.00	0.00	0.00
	3	3727	3427.90	1757	3727	3249	1952	3727	3727	0.00	3727	0.02	3727	0.02	0.00	0.00	0.00
	4	3219	1412.89	1586	3219	3018	1890	3219	3219	0.00	3219	0.00	3219	0.00	0.00	0.00	0.00
	5	3224	2471.31	1989	3224	2975	1418	3224	3224	0.02	3224	0.00	3224	0.00	0.00	0.00	0.00
	6	3168	2661.83	1464	3119	2688	1608	3119	3168	0.00	3168	0.00	3168	0.00	1.55	0.00	1.57
	7	2934	1545.35	1569	2934	2591	1585	2934	2934	0.00	2934	0.02	2934	0.02	0.00	0.00	0.00
	8	2018	1286.12	1032	2018	1859	1203	2018	2018	0.00	2018	0.00	2018	0.00	0.00	0.00	0.00
	9	2860	498.22	1511	2836	2593	1581	2836	2860	0.00	2860	0.00	2860	0.00	0.84	0.00	0.85
	10	3254	190.46	1908	3254	3032	1635	3254	3254	0.02	3254	0.02	3254	0.02	0.00	0.00	0.00
R=50	1	3076	12028.26	1588	3076	2736	1389	3076	3076	0.02	3076	0.00	3076	0.00	0.00	0.00	0.00
	2	2932	6191.56	1334	2925	2645	1429	2925	2932	0.00	2932	0.00	2932	0.00	0.24	0.00	0.24
	3	1974	5168.34	1105	1967	1805	910	1967	1974	0.00	1974	0.00	1974	0.00	0.35	0.00	0.36
	4	2969	6449.84	1320	2969	2549	1342	2969	2969	0.00	2969	0.02	2969	0.02	0.00	0.00	0.00
	5	3963	5355.76	1842	3963	3463	1777	3963	3963	0.02	3963	0.00	3963	0.00	0.00	0.00	0.00
	6	3089	9765.21	1126	3089	2502	1545	3089	3089	0.00	3089	0.02	3089	0.02	0.00	0.00	0.00
	7	3426	9732.92	1400	3419	3305	2078	3419	3426	0.00	3426	0.00	3426	0.00	0.20	0.00	0.20
	8	3122	5011.04	1742	3122	2976	1381	3122	3122	0.00	3122	0.00	3122	0.00	0.00	0.00	0.00
	9	3297	3727.42	2047	3297	2920	1047	3297	3297	0.02	3297	0.02	3297	0.02	0.00	0.00	0.00
	10	3029	7293.89	1036	3029	2505	1636	3029	3029	0.00	3029	0.00	3029	0.00	0.00	0.00	0.00
R=100	1	2102	5904.38	1031	2102	1828	925	2102	2102	0.00	2102	0.02	2102	0.02	0.00	0.00	0.00
	2	2508	4669.72	571	2508	1929	1438	2508	2508	0.00	2508	0.02	2508	0.02	0.00	0.00	0.00
	3	2937	12016.16	1226	2937	2447	1349	2937	2937	0.02	2937	0.00	2937	0.00	0.00	0.00	0.00
	4	3084	4867.58	1125	3084	2911	1874	3084	3084	0.00	3084	0.02	3084	0.02	0.00	0.00	0.00
	5	3825	5264.62	2171	3825	3688	1725	3825	3825	0.00	3825	0.00	3825	0.00	0.00	0.00	0.00
	6	2805	4696.54	1234	2805	2622	1492	2805	2805	0.00	2805	0.00	2805	0.00	0.00	0.00	0.00
	7	2894	5101.39	1420	2894	2658	1326	2894	2894	0.02	2894	0.02	2894	0.02	0.00	0.00	0.00
	8	2817	1887.54	1282	2817	2497	1439	2817	2817	0.00	2817	0.00	2817	0.00	0.00	0.00	0.00
	9	4012	1879.01	2061	4012	3950	2185	4012	4012	0.00	4012	0.00	4012	0.00	0.00	0.00	0.00
	10	2440	1689.51	1103	2440	2285	1342	2440	2440	0.00	2440	0.02	2440	0.02	0.00	0.00	0.00

Table A.7: Computational Results for N=10 – 1/3

N=10		Optimal Soln.		LB 1	LB 2	LB 3	LB 4	Best Lower Bound	VNS 1		VNS 2		Min of VNS		Perc.Dev.of		
Data Type	No	Obj. Val.	Time (sec.)						Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec,)	Opt.Soln. from LB	Opt.Soln. from VNS	Best LB from VNS
R=0,01	1	2826	89.61	2327	680	2315	2789	2789	2925	0.09	2959	0.06	2925	0.09	1.31	3.50	4.88
	2	2020	378.05	1495	510	1769	2009	2009	2020	0.05	2034	0.05	2020	0.05	0.54	0.00	0.55
	3	2964	615.55	2202	643	2770	2941	2941	2980	0.03	2999	0.05	2980	0.03	0.78	0.54	1.33
	4	3594	150.23	2913	768	3313	3578	3578	3600	0.02	3595	0.02	3595	0.02	0.45	0.03	0.48
	5	2577	290.91	1882	586	2413	2562	2562	2607	0.03	2605	0.02	2605	0.02	0.58	1.09	1.68
	6			2777	749	2740	3225	3225	3256	0.03	3256	0.03	3256	0.03			0.96
	7			2091	611	3142	3351	3351	3364	0.08	3359	0.02	3359	0.02			0.24
	8			1919	598	2771	2788	2788	2795	0.03	2795	0.03	2795	0.03			0.25
	9			2474	683	2860	3016	3016	3054	0.05	3048	0.03	3048	0.03			1.06
	10			1741	550	2473	2634	2634	2673	0.03	2673	0.02	2673	0.02			1.48
R=0,05	1	2919	4998.23	1635	654	2790	2894	2894	2959	0.05	2959	0.02	2959	0.02	0.86	1.37	2.25
	2	2722	658.50	1659	688	2606	2715	2715	2725	0.03	2722	0.03	2722	0.03	0.26	0.00	0.26
	3	2158	91.15	1824	619	1978	2108	2108	2215	0.00	2215	0.02	2215	0.02	2.32	2.64	5.08
	4	3609	53.35	3241	998	2961	3489	3489	3609	0.02	3649	0.02	3609	0.02	3.33	0.00	3.44
	5	2860	116.41	2466	881	2320	2757	2757	2922	0.03	2901	0.02	2901	0.02	3.60	1.43	5.22
	6			1826	714	2110	2266	2266	2300	0.03	2300	0.02	2300	0.02			1.50
	7			3003	889	2342	3023	3023	3168	0.03	3180	0.02	3168	0.03			4.80
	8			2490	818	2750	2940	2940	3009	0.03	3010	0.05	3009	0.03			2.35
	9			3346	1003	2925	3544	3544	3753	0.03	3753	0.03	3753	0.03			5.90
	10			3089	994	2811	2241	3089	3456	0.02	3460	0.03	3456	0.02			11.88
R=0,1	1	2673	166.94	2203	924	2160	2541	2541	2701	0.05	2717	0.02	2701	0.05	4.94	1.05	6.30
	2	3208	101.42	2859	1170	2778	3033	3033	3228	0.05	3208	0.02	3208	0.02	5.46	0.00	5.77
	3	3565	63.34	3258	1213	3055	3335	3335	3579	0.03	3579	0.02	3579	0.02	6.45	0.39	7.32
	4	2863	1133.76	2383	1050	2587	2674	2674	2883	0.03	2883	0.02	2883	0.02	6.60	0.70	7.82
	5	2061	233.36	1841	890	1779	1946	1946	2097	0.02	2097	0.02	2097	0.02	5.58	1.75	7.76
	6			2094	966	3176	3209	3209	3254	0.02	3254	0.02	3254	0.02			1.40
	7			2053	1080	3380	3460	3460	3541	0.02	3541	0.05	3541	0.05			2.34
	8			1973	839	2708	2770	2770	2824	0.08	2824	0.03	2824	0.03			1.95
	9			1966	1015	3179	3238	3238	3342	0.05	3326	0.02	3326	0.02			2.72
	10			2897	1137	3037	3209	3209	3311	0.02	3311	0.02	3311	0.02			3.18

Table A.8: Computational Results for N=10 – 2/3

N=10		Optimal Soln.		LB 1	LB 2	LB 3	LB 4	Best Lower Bound	VNS 1		VNS 2		Min of VNS		Perc.Dev.of		
Data Type	No	Obj. Val.	Time (sec.)						Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec,)	Opt.Soln. from LB	Opt.Soln. from VNS	Best LB from VNS
R=0,5	1	4440	1066.82	3702	3002	3947	3864	3947	4639	0.03	4639	0.03	4639	0.03	11.10	4.48	17.53
	2	4104	160.49	3705	2580	3114	3441	3705	4104	0.05	4104	0.00	4104	0.00	9.72	0.00	10.77
	3	4775	8752.50	2891	2402	4686	4676	4686	4775	0.03	4775	0.02	4775	0.02	1.86	0.00	1.90
	4	4631	3517.81	3534	3361	4178	4169	4178	4729	0.02	4679	0.02	4679	0.02	9.78	1.04	11.99
	5	5175	665.97	4337	3600	4790	4798	4798	5262	0.02	5262	0.03	5262	0.03	7.29	1.68	9.67
	6			3607	3087	3512	3457	3607	4409	0.02	4279	0.02	4279	0.02			18.63
	7			2486	2634	3656	3479	3656	3993	0.03	4053	0.02	3993	0.03			9.22
	8			2672	2486	3818	3853	3853	4294	0.03	4294	0.02	4294	0.02			11.45
	9			3230	2772	4042	3973	4042	4375	0.02	4375	0.02	4375	0.02			8.24
	10			2852	2741	4659	4661	4661	4723	0.02	4728	0.02	4723	0.02			1.33
R=1	1	5087	969.97	4531	3735	4155	4083	4531	5133	0.03	5133	0.03	5133	0.03	10.93	0.90	13.29
	2	5731	15296.14	4300	4453	4967	4870	4967	5764	0.02	5835	0.02	5764	0.02	13.33	0.58	16.05
	3	6979	7510.88	5132	5315	6441	5884	6441	7131	0.03	7092	0.02	7092	0.02	7.71	1.62	10.11
	4	5112	12318.94	4341	4048	4016	3624	4341	5210	0.02	5210	0.02	5210	0.02	15.08	1.92	20.02
	5	6255	7800.12	3454	4425	5789	4224	5789	6432	0.03	6432	0.00	6432	0.00	7.45	2.83	11.11
	6			4245	4862	5057	4794	5057	5705	0.03	5710	0.02	5705	0.03			12.81
	7			3785	4298	5174	4975	5174	5882	0.06	5972	0.03	5882	0.06			13.68
	8			5077	6510	6584	6206	6584	7316	0.02	7316	0.02	7316	0.02			11.12
	9			4912	6387	6908	6697	6908	7599	0.02	7599	0.02	7599	0.02			10.00
	10			4766	5793	5868	4684	5868	6510	0.02	6510	0.02	6510	0.02			10.94
R=5	1			2314	3483	3035		3483	3542	0.03	3542	0.02	3542	0.02			1.69
	2			2609	4862	4299		4862	4913	0.02	4913	0.02	4913	0.02			1.05
	3			2636	4672	3974		4672	4672	0.02	4672	0.02	4672	0.02			0.00
	4			3577	6352	5954		6352	6352	0.02	6352	0.02	6352	0.02			0.00
	5			2712	5067	3921		5067	5217	0.02	5217	0.02	5217	0.02			2.96
	6			2308	4102	3638		4102	4238	0.02	4238	0.00	4238	0.00			3.32
	7			2017	3411	3181		3411	3482	0.02	3482	0.00	3482	0.00			2.08
	8			3688	5490	4846		5490	5490	0.03	5490	0.02	5490	0.02			0.00
	9			2960	4549	4180		4549	4549	0.02	4549	0.02	4549	0.02			0.00
	10			2801	4449	3739		4449	4460	0.02	4460	0.02	4460	0.02			0.25

Table A.9: Computational Results for N=10 – 3/3

N=10		Optimal Soln.		LB 1	LB 2	LB 3	LB 4	Best Lower Bound	VNS 1		VNS 2		Min of VNS		Perc.Dev.of		
Data Type	No	Obj. Val.	Time (sec.)						Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec,)	Opt.Soln. from LB	Opt.Soln. from VNS	Best LB from VNS
R=10	1			2739	4537	4450		4537	4547	0.02	4547	0.02	4547	0.02			0.22
	2			1730	3228	2738		3228	3274	0.00	3274	0.00	3274	0.00			1.43
	3			2071	3697	3286		3697	3772	0.02	3772	0.02	3772	0.02			2.03
	4			2574	4832	4257		4832	4832	0.02	4832	0.00	4832	0.00			0.00
	5			2347	4350	3731		4350	4395	0.02	4395	0.02	4395	0.02			1.03
	6			3210	5611	5223		5611	5611	0.00	5611	0.00	5611	0.00			0.00
	7			2290	3723	3443		3723	3723	0.02	3723	0.02	3723	0.02			0.00
	8			2890	5478	4708		5478	5478	0.02	5478	0.02	5478	0.02			0.00
	9			3018	6037	5495		6037	6037	0.00	6037	0.00	6037	0.00			0.00
	10			1287	2970	2656		2970	2977	0.02	2977	0.02	2977	0.02			0.24
R=50	1			2410	4720	4496		4720	4720	0.00	4720	0.00	4720	0.00			0.00
	2			1221	2960	2668		2960	2960	0.00	2960	0.02	2960	0.02			0.00
	3			2257	4531	3850		4531	4531	0.02	4531	0.00	4531	0.00			0.00
	4			2312	4506	4026		4506	4506	0.00	4506	0.02	4506	0.02			0.00
	5			2419	4929	4197		4929	4929	0.02	4929	0.00	4929	0.00			0.00
	6			1971	4188	3857		4188	4188	0.02	4188	0.02	4188	0.02			0.00
	7			1831	4727	4106		4727	4733	0.00	4733	0.00	4733	0.00			0.13
	8			2965	5885	5523		5885	5885	0.00	5885	0.02	5885	0.02			0.00
	9			2247	4163	3557		4163	4163	0.02	4163	0.02	4163	0.02			0.00
	10			2869	4566	4262		4566	4566	0.00	4566	0.02	4566	0.02			0.00
R=100	1			2305	4811	4362		4811	4811	0.02	4811	0.00	4811	0.00			0.00
	2			2454	4464	3928		4464	4464	0.00	4464	0.02	4464	0.02			0.00
	3			2655	6103	5395		6103	6103	0.00	6103	0.00	6103	0.00			0.00
	4			2203	4858	4574		4858	4858	0.02	4858	0.02	4858	0.02			0.00
	5			2442	4110	3477		4110	4110	0.02	4110	0.00	4110	0.00			0.00
	6			1777	4396	3982		4396	4396	0.00	4396	0.02	4396	0.02			0.00
	7			1956	4715	4416		4715	4715	0.02	4715	0.00	4715	0.00			0.00
	8			2864	4401	4069		4401	4401	0.02	4401	0.02	4401	0.02			0.00
	9			2528	5817	5106		5817	5817	0.00	5817	0.00	5817	0.00			0.00
	10			2094	4496	4172		4496	4496	0.02	4496	0.02	4496	0.02			0.00

Table A.10: Computational Results for N=15 – 1/3

N=15		LB 1	LB 2	LB 3	LB 4	Best Lower Bound	VNS 1		VNS 2		Min of VNS		Perc.Dev. of LB from VNS
Data Type	No						Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec,)	
R=0,01	1	3584	877	5107	5443	5443	5524	0.22	5451	0.13	5451	0.13	0.15
	2	4333	1010	4194	5121	5121	5350	0.17	5243	0.12	5243	0.12	2.38
	3	3851	875	4097	4399	4399	4496	0.16	4496	0.11	4496	0.11	2.21
	4	3223	884	5079	5228	5228	5233	0.23	5233	0.08	5233	0.08	0.10
	5	3675	919	4523	4955	4955	5009	0.20	5145	0.13	5009	0.20	1.09
	6	4492	1060	3796	4818	4818	4989	0.23	4923	0.14	4923	0.14	2.18
	7	5117	1060	4392	5775	5775	5957	0.30	6024	0.09	5957	0.30	3.15
	8	4259	976	5666	5989	5989	6028	0.23	6070	0.11	6028	0.23	0.65
	9	6144	1183	6270	6734	6734	6797	0.16	6797	0.14	6797	0.14	0.94
	10	5628	1152	5496	6408	6408	6733	0.13	6539	0.09	6539	0.09	2.04
R=0,05	1	5924	1517	5638	6488	6488	6733	0.13	6733	0.06	6733	0.06	3.78
	2	4092	1358	5922	6145	6145	6322	0.16	6329	0.06	6322	0.16	2.88
	3	5802	1623	4647	6022	6022	6389	0.11	6389	0.09	6389	0.09	6.09
	4	5074	1392	5733	5889	5889	6087	0.19	6025	0.13	6025	0.13	2.31
	5	5509	1411	6145	6683	6683	6951	0.19	6900	0.06	6900	0.06	3.25
	6	5036	1351	4101	5135	5135	5509	0.27	5510	0.05	5509	0.27	7.28
	7	5148	1568	6356	6825	6825	7009	0.09	7006	0.06	7006	0.06	2.65
	8	4849	1231	5101	5543	5543	5712	0.16	5787	0.08	5712	0.16	3.05
	9	3501	1188	5181	5415	5415	5529	0.14	5487	0.11	5487	0.11	1.33
	10	4820	1376	6765	6834	6834	6891	0.09	6890	0.08	6890	0.08	0.82
R=0,1	1	4636	1706	5846	6020	6020	6167	0.16	6150	0.09	6150	0.09	2.16
	2	5139	1714	4161	4865	5139	5485	0.16	5490	0.06	5485	0.16	6.73
	3	2754	1342	4817	4902	4902	5064	0.28	5071	0.11	5064	0.28	3.30
	4	4648	1879	2914	4224	4648	4857	0.19	4855	0.09	4855	0.09	4.45
	5	5504	1801	6277	6652	6652	6905	0.25	6918	0.09	6905	0.25	3.80
	6	4436	1750	7345	7392	7392	7423	0.31	7542	0.05	7423	0.31	0.42
	7	5186	1786	4368	5265	5265	5893	0.16	5902	0.06	5893	0.16	11.93
	8	6450	1859	5047	6493	6493	7173	0.09	7044	0.08	7044	0.08	8.49
	9	4202	1696	5755	5924	5924	6112	0.28	6129	0.11	6112	0.28	3.17
	10	5922	1956	5052	5714	5922	6412	0.20	6445	0.08	6412	0.20	8.27

Table A.11: Computational Results for N=15 – 2/3

N=15		LB 1	LB 2	LB 3	LB 4	Best Lower Bound	VNS 1		VNS 2		Min of VNS		Perc.Dev. of LB from VNS
Data Type	No						Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec,)	
R=0,5	1	6656	4602	7555	7655	7655	8501	0.14	8491	0.11	8491	0.11	10.92
	2	7367	6213	9266	9303	9303	9617	0.08	9617	0.05	9617	0.05	3.38
	3	8412	5970	8968	9101	9101	9744	0.08	9817	0.05	9744	0.08	7.07
	4	4479	4124	8953	8954	8954	8992	0.08	8992	0.05	8992	0.05	0.42
	5	6745	6148	8518	8679	8679	9804	0.09	9786	0.08	9786	0.08	12.75
	6	7066	6403	7436	6988	7436	8794	0.09	8963	0.09	8794	0.09	18.26
	7	9172	7463	7393	6945	9172	10003	0.09	9933	0.13	9933	0.13	8.30
	8	5416	4562	9665	9623	9665	9899	0.08	9867	0.09	9867	0.09	2.09
	9	6840	4889	6603	6579	6840	8153	0.06	8047	0.08	8047	0.08	17.65
	10	7855	4887	6256	6343	7855	8336	0.08	8336	0.05	8336	0.05	6.12
R=1	1	7739	9499	10442	9776	10442	11205	0.13	11159	0.11	11159	0.11	6.87
	2	12053	11262	11386	8878	12053	13252	0.08	13252	0.05	13252	0.05	9.95
	3	7724	7543	8746	8376	8746	9915	0.08	10035	0.06	9915	0.08	13.37
	4	9681	10920	10336	9328	10920	12800	0.17	12903	0.09	12800	0.17	17.22
	5	11694	12203	10812	8877	12203	13455	0.14	13337	0.05	13337	0.05	9.29
	6	9403	11163	13209	12740	13209	14308	0.09	14308	0.06	14308	0.06	8.32
	7	10578	10374	9932	8980	10578	12540	0.22	12593	0.09	12540	0.22	18.55
	8	11891	11198	10414	9075	11891	13550	0.13	13472	0.14	13472	0.14	13.30
	9	9729	12228	11031	10217	12228	13433	0.08	13433	0.08	13433	0.08	9.85
	10	11973	12216	10899	9398	12216	14053	0.08	14255	0.06	14053	0.08	15.04
R=5	1	5881	9377	8727		9377	9525	0.11	9525	0.05	9525	0.05	1.58
	2	5948	9512	8234		9512	9514	0.11	9514	0.06	9514	0.06	0.02
	3	6269	10413	8773		10413	10571	0.06	10571	0.05	10571	0.05	1.52
	4	6322	10565	9768		10565	10565	0.06	10565	0.03	10565	0.03	0.00
	5	5754	9033	8144		9033	9303	0.03	9303	0.05	9303	0.05	2.99
	6	3759	7471	6923		7471	7817	0.06	7817	0.06	7817	0.06	4.63
	7	4883	7019	6436		7019	7293	0.13	7293	0.05	7293	0.05	3.90
	8	4963	9142	7969		9142	9289	0.05	9289	0.02	9289	0.02	1.61
	9	4846	9099	8075		9099	9258	0.03	9258	0.05	9258	0.05	1.75
	10	7047	11808	10569		11808	11808	0.09	11808	0.03	11808	0.03	0.00

Table A.12: Computational Results for N=15 – 3/3

N=15		LB 1	LB 2	LB 3	LB 4	Best Lower Bound	VNS 1		VNS 2		Min of VNS		Perc.Dev. of LB from VNS
Data Type	No						Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec,)	
R=10	1	4036	10671	9261		10671	10704	0.02	10704	0.03	10704	0.03	0.31
	2	5625	10026	9163		10026	10026	0.05	10026	0.03	10026	0.03	0.00
	3	5620	9597	8963		9597	9597	0.03	9597	0.02	9597	0.02	0.00
	4	5992	9447	8949		9447	9460	0.02	9460	0.03	9460	0.03	0.14
	5	4393	8602	7423		8602	8602	0.05	8602	0.02	8602	0.02	0.00
	6	4664	9775	8585		9775	9876	0.05	9876	0.03	9876	0.03	1.03
	7	6706	10747	9717		10747	10747	0.06	10747	0.02	10747	0.02	0.00
	8	4605	8550	7651		8550	8560	0.03	8560	0.03	8560	0.03	0.12
	9	2825	7464	6477		7464	7491	0.11	7491	0.05	7491	0.05	0.36
	10	4810	9865	8933		9865	9885	0.02	9885	0.02	9885	0.02	0.20
R=50	1	5957	9995	8885		9995	9995	0.05	9995	0.03	9995	0.03	0.00
	2	5177	11182	9941		11182	11182	0.02	11182	0.03	11182	0.03	0.00
	3	5824	12044	11319		12044	12044	0.03	12044	0.02	12044	0.02	0.00
	4	4402	10653	9681		10653	10653	0.03	10653	0.03	10653	0.03	0.00
	5	3265	8154	7320		8154	8154	0.02	8154	0.02	8154	0.02	0.00
	6	5443	10356	9283		10356	10356	0.03	10356	0.03	10356	0.03	0.00
	7	4007	10443	9037		10443	10443	0.02	10443	0.02	10443	0.02	0.00
	8	3841	8980	7447		8980	8989	0.03	8989	0.03	8989	0.03	0.10
	9	4777	9592	8171		9592	9592	0.03	9592	0.02	9592	0.02	0.00
	10	5318	11541	10916		11541	11541	0.02	11541	0.03	11541	0.03	0.00
R=100	1	4706	10173	9030		10173	10173	0.03	10173	0.02	10173	0.02	0.00
	2	4769	10143	9162		10143	10143	0.02	10143	0.03	10143	0.03	0.00
	3	3254	8606	7619		8606	8606	0.03	8606	0.03	8606	0.03	0.00
	4	4618	9308	8218		9308	9308	0.02	9308	0.02	9308	0.02	0.00
	5	4788	9971	8541		9971	9971	0.03	9971	0.03	9971	0.03	0.00
	6	3501	9020	8391		9020	9020	0.02	9020	0.02	9020	0.02	0.00
	7	4702	9826	8886		9826	9826	0.03	9826	0.03	9826	0.03	0.00
	8	4316	10242	9462		10242	10242	0.03	10242	0.02	10242	0.02	0.00
	9	3452	8142	7163		8142	8142	0.02	8142	0.03	8142	0.03	0.00
	10	4593	9966	8759		9966	9966	0.03	9966	0.02	9966	0.02	0.00

Table A.13: Computational Results for N=20 – 1/3

N=20		LB 1	LB 2	LB 3	LB 4		Best Lower Bound		VNS 1		VNS 2		Min of VNS		Perc.Dev. of LB from VNS
Data Type	No				Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	
R=0,01	1	6998	1280	9921	10001	1.24	10001	1.24	10053	0.41	10109	0.22	10053	0.41	0.52
	2	6217	1287	8721	9256	3.10	9256	3.10	9499	1.22	9510	0.19	9499	1.22	2.63
	3	7888	1386	7450	8754	0.40	8754	0.40	8997	0.52	8896	0.56	8896	0.56	1.62
	4	8436	1499	9591	10231	10.81	10231	10.81	10536	0.33	10453	0.30	10453	0.30	2.17
	5	9249	1612	9483	10548	8.75	10548	8.75	10751	0.73	10794	0.31	10751	0.73	1.92
	6	7226	1375	11329	11744	3.58	11744	3.58	11898	0.34	11848	0.39	11848	0.39	0.89
	7	8353	1450	8922	10146	12.19	10146	12.19	10461	0.36	10457	0.31	10457	0.31	3.07
	8	8328	1530	6780	8912	2.95	8912	2.95	9251	0.27	9222	0.23	9222	0.23	3.48
	9	5937	1258	8024	8761	1.02	8761	1.02	9110	0.27	8987	0.31	8987	0.31	2.58
	10	7696	1479	11382	11939	1.30	11939	1.30	12075	0.56	12281	0.20	12075	0.56	1.14
R=0,05	1	9129	2182	6672	8738	0.28	9129	0.00	9443	0.14	9423	0.13	9423	0.13	3.22
	2	5843	1848	9373	9706	26.14	9706	26.14	9945	1.17	10030	0.19	9945	1.17	2.46
	3	9521	2310	10056	10355	1.15	10355	1.15	10731	0.64	10758	0.25	10731	0.64	3.63
	4	6430	1874	9214	9884	2.87	9884	2.87	10161	1.05	10005	0.19	10005	0.19	1.22
	5	7010	1949	9078	9737	37.75	9737	37.75	10201	0.42	10009	0.38	10009	0.38	2.79
	6	7629	2085	11156	11262	0.52	11262	0.52	11358	0.39	11361	0.19	11358	0.39	0.85
	7	8847	1959	8268	9999	32.40	9999	32.40	10651	0.75	10640	0.34	10640	0.34	6.41
	8	10133	2286	10602	10993	0.45	10993	0.45	11355	0.39	11217	0.30	11217	0.30	2.04
	9	7026	2073	8258	8659	0.88	8659	0.88	9074	0.58	8973	0.33	8973	0.33	3.63
	10	9966	2232	10470	11778	37.89	11778	37.89	12127	0.66	12200	0.27	12127	0.66	2.96
R=0,1	1	9132	3150	12003	12147	0.29	12147	0.29	12300	0.16	12296	0.13	12296	0.13	1.23
	2	6959	2630	9665	9955	1.10	9955	1.10	10184	0.38	10145	0.11	10145	0.11	1.91
	3	9165	2965	9488	10161	2.62	10161	2.62	10765	0.42	10849	0.17	10765	0.42	5.94
	4	6944	2773	9345	9467	1.23	9467	1.23	9914	0.48	9765	0.25	9765	0.25	3.15
	5	9132	2956	7754	9483	25.79	9483	25.79	10452	0.23	10443	0.20	10443	0.20	10.12
	6	6800	2322	8183	8685	4.80	8685	4.80	9258	0.86	9278	0.17	9258	0.86	6.60
	7	7969	3142	9756	10352	12.25	10352	12.25	10934	0.62	10731	0.23	10731	0.23	3.66
	8	7868	2792	10684	11021	1.18	11021	1.18	11487	0.39	11409	0.25	11409	0.25	3.52
	9	7968	3051	8534	8656	1.08	8656	1.08	8958	0.41	9027	0.23	8958	0.41	3.49
	10	7988	2971	10632	11163	0.82	11163	0.82	11605	0.62	11490	0.39	11490	0.39	2.93

Table A.14: Computational Results for N=20 – 2/3

N=20		LB 1	LB 2	LB 3	LB 4		Best Lower Bound		VNS 1		VNS 2		Min of VNS		Perc.Dev. of LB from VNS
Data Type	No				Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	
R=0,5	1	11834	9105	14247	14119	0.21	14247	0.00	15368	0.55	15261	0.25	15261	0.25	7.12
	2	11621	8907	16905	16982	0.30	16982	0.30	17353	0.17	17331	0.13	17331	0.13	2.06
	3	12762	10317	14119	14087	0.38	14119	0.00	15736	0.22	15736	0.11	15736	0.11	11.45
	4	12915	9611	11902	11809	0.53	12915	0.00	14285	0.28	14335	0.11	14285	0.28	10.61
	5	10950	9503	13693	13798	0.30	13798	0.30	15218	0.39	15453	0.14	15218	0.39	10.29
	6	13193	9197	14082	13776	0.46	14082	0.00	16386	0.45	16504	0.14	16386	0.45	16.36
	7	13360	9437	12476	12329	1.48	13360	0.00	15884	0.44	15884	0.28	15884	0.28	18.89
	8	12861	8891	15714	15892	0.30	15892	0.30	16815	0.98	16948	0.20	16815	0.98	5.81
	9	10562	8866	11546	11087	0.29	11546	0.00	13255	0.50	13158	0.19	13158	0.19	13.96
	10	11120	9070	12064	11672	0.30	12064	0.00	14236	0.53	14085	0.34	14085	0.34	16.75
R=1	1	18090	17007	18208	16673	0.17	18208	0.00	21646	0.23	21640	0.14	21640	0.14	18.85
	2	15743	21547	22566	21085	0.20	22566	0.00	25366	0.36	25249	0.17	25249	0.17	11.89
	3	15291	14458	15401	14147	0.29	15401	0.00	17880	0.28	18230	0.16	17880	0.28	16.10
	4	17178	17404	19860	18705	0.20	19860	0.00	21299	0.28	21803	0.23	21299	0.28	7.25
	5	18927	19438	20158	19114	0.28	20158	0.00	22703	0.19	22674	0.19	22674	0.19	12.48
	6	22043	20470	19172	17154	0.43	22043	0.00	25153	0.56	25021	0.37	25021	0.37	13.51
	7	19396	18027	18752	16512	0.30	19396	0.00	22484	0.70	22060	0.22	22060	0.22	13.73
	8	16180	15560	21108	20739	0.22	21108	0.00	22094	0.19	22124	0.23	22094	0.19	4.67
	9	16059	16227	15667	14556	0.40	16227	0.00	19368	0.86	19182	0.34	19182	0.34	18.21
	10	12959	16442	18841	18434	0.28	18841	0.00	20931	0.87	20983	0.11	20931	0.87	11.09
R=5	1	8671	13496	12614			13496	0.00	13796	0.19	13796	0.11	13796	0.11	2.22
	2	9129	16467	14370			16467	0.00	16644	0.19	16644	0.11	16644	0.11	1.07
	3	8778	18273	17110			18273	0.00	18422	0.27	18422	0.09	18422	0.09	0.82
	4	9742	16110	13422			16110	0.00	16124	0.36	16128	0.13	16124	0.36	0.09
	5	12874	20800	19421			20800	0.00	20819	0.22	20819	0.13	20819	0.13	0.09
	6	9759	17344	15611			17344	0.00	17474	0.06	17474	0.06	17474	0.06	0.75
	7	10121	18943	17407			18943	0.00	18943	0.50	18943	0.11	18943	0.11	0.00
	8	8910	13409	11865			13409	0.00	13972	0.47	13982	0.13	13972	0.47	4.20
	9	8080	12447	10801			12447	0.00	12637	0.48	12637	0.13	12637	0.13	1.53
	10	8314	14110	12950			14110	0.00	14479	0.13	14479	0.11	14479	0.11	2.62

Table A.15: Computational Results for N=20 – 3/3

N=20		LB 1	LB 2	LB 3	LB 4		Best Lower Bound		VNS 1		VNS 2		Min of VNS		Perc.Dev. of LB from VNS
Data Type	No				Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	
R=10	1	7363	15013	13596			15013	0.00	15013	0.22	15013	0.08	15013	0.08	0.00
	2	6052	16255	13415			16255	0.00	16404	0.11	16404	0.12	16404	0.12	0.92
	3	8426	16073	15024			16073	0.00	16073	0.09	16073	0.11	16073	0.11	0.00
	4	8756	15892	13776			15892	0.00	15892	0.08	15892	0.06	15892	0.06	0.00
	5	10217	17519	15148			17519	0.00	17536	0.17	17536	0.13	17536	0.13	0.10
	6	9498	19436	16981			19436	0.00	19436	0.16	19436	0.06	19436	0.06	0.00
	7	8151	16332	15634			16332	0.00	16467	0.11	16467	0.06	16467	0.06	0.83
	8	9451	16294	14246			16294	0.00	16294	0.19	16294	0.08	16294	0.08	0.00
	9	7357	16543	13461			16543	0.00	16712	0.16	16712	0.08	16712	0.08	1.02
	10	9694	16332	14343			16332	0.00	16450	0.19	16450	0.12	16450	0.12	0.72
R=50	1	7180	13518	12633			13518	0.00	13518	0.05	13518	0.06	13518	0.06	0.00
	2	5824	12771	11441			12771	0.00	12771	0.05	12771	0.06	12771	0.06	0.00
	3	8136	16881	14087			16881	0.00	16884	0.05	16884	0.05	16884	0.05	0.02
	4	6699	17150	15755			17150	0.00	17150	0.06	17150	0.05	17150	0.05	0.00
	5	7065	17496	15134			17496	0.00	17496	0.06	17496	0.05	17496	0.05	0.00
	6	7598	18553	16433			18553	0.00	18553	0.11	18553	0.08	18553	0.08	0.00
	7	8317	14923	14412			14923	0.00	14940	0.06	14940	0.06	14940	0.06	0.11
	8	7475	17960	15457			17960	0.00	17960	0.06	17960	0.06	17960	0.06	0.00
	9	8942	20019	18256			20019	0.00	20019	0.05	20019	0.08	20019	0.08	0.00
	10	7212	14416	13795			14416	0.00	14416	0.05	14416	0.05	14416	0.05	0.00
R=100	1	5468	14001	11894			14001	0.00	14001	0.06	14001	0.05	14001	0.05	0.00
	2	6323	18370	15927			18370	0.00	18370	0.05	18370	0.05	18370	0.05	0.00
	3	6399	15310	13000			15310	0.00	15310	0.06	15310	0.06	15310	0.06	0.00
	4	6414	17339	14464			17339	0.00	17339	0.06	17339	0.05	17339	0.05	0.00
	5	8670	14287	12945			14287	0.00	14287	0.06	14287	0.06	14287	0.06	0.00
	6	7598	18553	16433			18553	0.00	18553	0.11	18553	0.08	18553	0.08	0.00
	7	8317	14923	14412			14923	0.00	14940	0.05	14940	0.06	14940	0.06	0.11
	8	7475	17960	15457			17960	0.00	17960	0.05	17960	0.05	17960	0.05	0.00
	9	8942	20019	18256			20019	0.00	20019	0.05	20019	0.05	20019	0.05	0.00
	10	7212	14416	13795			14416	0.00	14416	0.06	14416	0.06	14416	0.06	0.00

Table A.16: Computational Results for N=30 – 1/3

N=30		LB 1	LB 2	LB 3	LB 4		Best Lower Bound		VNS 1		VNS 2		Min of VNS		Perc.Dev. of LB from VNS
Data Type	No				Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec,)	
R=0,01	1	13387	2260	18338	19183	607.14	19183	607.14	19369	4.40	19558	0.86	19369	4.40	0.97
	2	16223	2376	19242	20434	13.68	20434	13.68	20798	4.34	20812	0.67	20798	4.34	1.78
	3	14325	2285	17076	18222	50.25	18222	50.25	18768	4.29	18864	0.61	18768	4.29	3.00
	4	17820	2468	18305	19884	23.33	19884	23.33	20699	2.93	20474	1.26	20474	1.26	2.97
	5	17215	2550	23349	24047	2640.91	24047	2640.91	24631	3.93	24642	0.67	24631	3.93	2.43
	6	15613	2414	18636	19086	436.96	19086	436.96	19435	3.84	19446	0.97	19435	3.84	1.83
	7	17467	2465	17473	19068	646.7	19068	646.7	19666	1.84	19717	0.64	19666	1.84	3.14
	8	11807	2117	16807	17981	43.4	17981	43.4	18507	4.60	18574	0.92	18507	4.60	2.93
	9	15163	2413	15115	16924	683.17	16924	683.17	17478	3.70	17679	1.06	17478	3.70	3.27
	10	19884	2637	20808	22793	237.03	22793	237.03	23494	4.10	23207	2.43	23207	2.43	1.82
R=0,05	1	14700	4012	19937	21083	26.24	21083	26.24	21602	6.52	21641	1.12	21602	6.52	2.46
	2	17090	3703	20845	21226	8.6	21226	8.6	21692	2.96	21877	0.84	21692	2.96	2.20
	3	21183	3958	21767	22757	15.91	22757	15.91	23462	2.23	23577	0.83	23462	2.23	3.10
	4	15694	3606	19523	20195	15.4	20195	15.4	20667	5.13	20684	0.64	20667	5.13	2.34
	5	14522	3697	19997	20350	17.53	20350	17.53	20776	7.52	20816	0.66	20776	7.52	2.09
	6	16454	3743	18375	19596	2838.37	19596	2838.37	20983	2.57	20491	1.72	20491	1.72	4.57
	7	19066	3649	15246	19515	318.08	19515	318.08	21060	2.01	21157	0.86	21060	2.01	7.92
	8	17826	3958	19403	20586	175.48	20586	175.48	21394	3.65	21721	1.08	21394	3.65	3.92
	9	19979	3979	21684	23134	535.44	23134	535.44	24118	1.65	23993	0.83	23993	0.83	3.71
	10	17382	3689	18646	20314	2060.79	20314	2060.79	21492	2.37	21550	1.00	21492	2.37	5.80
R=0,1	1	17455	5722	20824	21112	20.81	21112	20.81	21851	2.50	22099	0.47	21851	2.50	3.50
	2	18425	5496	20667	21721	56.71	21721	56.71	22910	3.42	22726	0.89	22726	0.89	4.63
	3	20564	5636	19780	21862	6412.25	21862	6412.25	23555	1.70	23667	0.83	23555	1.70	7.74
	4	19525	5302	18415	20191	326.27	20191	326.27	22172	3.70	22346	0.64	22172	3.70	9.81
	5	19567	5681	21681	23528	428.2	23528	428.2	24829	2.12	24865	0.62	24829	2.12	5.53
	6	14522	5583	20866	21210	4.16	21210	4.16	21660	8.75	22140	0.70	21660	8.75	2.12
	7	20172	6283	23035	23502	95.57	23502	95.57	24644	1.90	24621	0.61	24621	0.61	4.76
	8	21432	6102	19280	21778	501.32	21778	501.32	23929	1.31	23788	0.87	23788	0.87	9.23
	9	26304	6072	15992	24331	7.48	26304	0	26978	1.65	27031	0.59	26978	1.65	2.56
	10	21622	6133	20594	22420	569.34	22420	569.34	24413	4.35	24291	1.06	24291	1.06	8.35

Table A.17: Computational Results for N=30 – 2/3

N=30		LB 1	LB 2	LB 3	LB 4		Best Lower Bound		VNS 1		VNS 2		Min of VNS		Perc.Dev. of LB from VNS
Data Type	No				Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec,)	
R=0,5	1	31727	19358	27342	27595	7.2	31727	0	34027	2.14	33941	0.59	33941	0.59	6.98
	2	28530	20202	26848	26835	88.16	28530	0	33287	1.00	32679	0.83	32679	0.83	14.54
	3	28440	18467	22638	22149	49.74	28440	0	30582	0.87	30759	0.80	30582	0.87	7.53
	4	25471	21988	32403	32159	1.05	32403	0	34948	1.78	34758	0.89	34758	0.89	7.27
	5	25230	19553	26198	25726	2.66	26198	0	30417	2.73	30046	1.00	30046	1.00	14.69
	6	26793	21559	28562	28520	4.46	28562	0	33793	1.25	34681	1.48	33793	1.25	18.31
	7	32879	20860	29411	29344	4.81	32879	0	36205	2.31	35784	0.76	35784	0.76	8.84
	8	29413	18547	27859	28698	213.35	29413	0	34578	1.86	34479	0.80	34479	0.80	17.22
	9	30150	21597	25464	26224	33.76	30150	0	33504	3.54	33389	0.70	33389	0.70	10.74
	10	25379	21336	30500	30520	3.6	30520	3.6	33893	2.53	34174	0.44	33893	2.53	11.05
R=1	1	35393	34080	35937	34146	1.78	35937	0	42606	3.00	41916	0.61	41916	0.61	16.64
	2	44748	43268	43138	40266	0.45	44748	0	51211	3.82	52212	0.59	51211	3.82	14.44
	3	32535	35540	40887	38168	0.3	40887	0	44914	1.93	45032	0.61	44914	1.93	9.85
	4	37688	40223	39656	35355	0.1	40223	0	46786	2.37	46943	0.86	46786	2.37	16.32
	5	34387	32188	35947	34098	0.35	35947	0	40530	3.43	40857	0.86	40530	3.43	12.75
	6	36011	38851	40917	39195	0.39	40917	0	46714	1.58	46215	1.05	46215	1.05	12.95
	7	32036	33509	41247	40138	0.42	41247	0	47721	3.67	47202	0.89	47202	0.89	14.44
	8	40334	39462	41486	37510	0.64	41486	0	47679	0.73	47178	0.44	47178	0.44	13.72
	9	45692	41664	41765	40353	1.17	45692	0	50929	1.79	51059	0.66	50929	1.79	11.46
	10	34267	38813	47338	46068	0.37	47338	0	49873	4.45	50340	0.67	49873	4.45	5.36
R=5	1	23553	38622	35148			38622	0	39096	3.50	39096	0.37	39096	0.37	1.23
	2	19082	32304	29101			32304	0	32971	0.72	33010	0.41	32971	0.72	2.06
	3	21526	40562	35994			40562	0	41245	2.75	41245	0.44	41245	0.44	1.68
	4	22230	37949	33612			37949	0	38089	4.29	38089	0.80	38089	0.80	0.37
	5	21124	37996	35270			37996	0	38391	3.15	38488	0.19	38391	3.15	1.04
	6	19893	37357	32785			37357	0	37418	1.75	37418	0.48	37418	0.48	0.16
	7	21571	36395	31920			36395	0	36704	0.28	36704	0.20	36704	0.20	0.85
	8	19908	34392	30514			34392	0	34858	0.20	34858	0.19	34858	0.19	1.35
	9	18615	36645	31899			36645	0	37435	2.33	37424	0.42	37424	0.42	2.13
	10	20621	37558	31336			37558	0	37791	0.75	37803	0.59	37791	0.75	0.62

Table A.18: Computational Results for N=30 – 3/3

N=30		LB 1	LB 2	LB 3	LB 4		Best Lower Bound		VNS 1		VNS 2		Min of VNS		Perc.Dev. of LB from VNS
Data Type	No				Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec,)	
R=10	1	23572	39528	36574			39528	0	39705	0.98	39705	0.62	39705	0.62	0.45
	2	16688	30254	26854			30254	0	30422	0.80	30422	0.41	30422	0.41	0.56
	3	16380	31188	28707			31188	0	31407	0.92	31407	0.38	31407	0.38	0.70
	4	14607	30483	26543			30483	0	30604	1.05	30604	0.42	30604	0.42	0.40
	5	18692	34977	31089			34977	0	35145	1.59	35145	0.19	35145	0.19	0.48
	6	18040	40133	35434			40133	0	40133	1.05	40133	0.42	40133	0.42	0.00
	7	21975	39018	35778			39018	0	39053	1.14	39053	0.39	39053	0.39	0.09
	8	20605	36388	33632			36388	0	36388	0.58	36388	0.41	36388	0.41	0.00
	9	17918	36152	32153			36152	0	36214	1.01	36214	0.27	36214	0.27	0.17
	10	16683	39682	34681			39682	0	39780	1.23	39780	0.41	39780	0.41	0.25
R=50	1	18846	36764	31831			36764	0	36764	0.20	36764	0.17	36764	0.17	0.00
	2	15244	34297	29983			34297	0	34297	0.44	34297	0.23	34297	0.23	0.00
	3	15632	37807	31485			37807	0	37807	0.28	37807	0.41	37807	0.41	0.00
	4	16569	35671	31579			35671	0	35671	0.19	35671	0.20	35671	0.20	0.00
	5	20155	42178	37257			42178	0	42178	0.25	42178	0.25	42178	0.25	0.00
	6	19024	37109	32228			37109	0	37109	0.39	37109	0.17	37109	0.17	0.00
	7	19066	40128	35625			40128	0	40128	0.25	40128	0.20	40128	0.20	0.00
	8	18910	39005	35480			39005	0	39005	0.23	39005	0.23	39005	0.23	0.00
	9	14791	36344	32617			36344	0	36344	0.22	36344	0.20	36344	0.20	0.00
	10	13149	36346	31509			36346	0	36346	0.44	36346	0.19	36346	0.19	0.00
R=100	1	17609	37144	33376			37144	0	37144	0.17	37144	0.20	37144	0.20	0.00
	2	14207	35499	32204			35499	0	35499	0.19	35499	0.23	35499	0.23	0.00
	3	13454	33725	30127			33725	0	33725	0.25	33725	0.16	33725	0.16	0.00
	4	18277	37747	33944			37747	0	37747	0.16	37747	0.19	37747	0.19	0.00
	5	22539	44687	39629			44687	0	44687	0.17	44687	0.20	44687	0.20	0.00
	6	12221	34345	29588			34345	0	34345	0.22	34345	0.23	34345	0.23	0.00
	7	20893	41191	38223			41191	0	41191	0.23	41191	0.17	41191	0.17	0.00
	8	14975	33012	29098			33012	0	33012	0.19	33012	0.20	33012	0.20	0.00
	9	14502	36102	32206			36102	0	36102	0.19	36102	0.22	36102	0.22	0.00
	10	14685	29868	27732			29868	0	29868	0.22	29868	0.20	29868	0.20	0.00

Table A.19: Computational Results for N=35 – 1/3

N=35		LB 1	LB 2	LB 3	LB 4		Best Lower Bound		VNS 1		VNS 2		Min of VNS		Perc.Dev. of LB from VNS
Data Type	No				Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec,)	
R=0,01	1	23380	2933	20640	25650	2689.77	25650	2689.77	27122	5.21	27199	2.01	27122	5.21	5.74
	2	29444	3353	21504	29864	2031.01	29864	2031.01	31462	6.47	31598	2.57	31462	6.47	5.35
	3	18561	2824	21437	24183	3242.2	24183	3242.2	24878	5.43	24879	2.40	24878	5.43	2.87
	4	19694	2881	26304	27227	5164.18	27227	5164.18	27922	14.03	27706	3.12	27706	3.12	1.76
	5	24837	3216	28617	30281	7749.5	30281	7749.5	30882	8.74	30900	1.73	30882	8.74	1.98
	6	26206	3272	22270	28393	192.84	28393	192.84	29626	9.42	29392	1.86	29392	1.86	3.52
	7	22705	3014	23569	25693	134.63	25693	134.63	26604	6.96	26281	1.03	26281	1.03	2.29
	8	20101	2946	18586	22846	2248.38	22846	2248.38	23804	9.44	24064	1.14	23804	9.44	4.19
	9	23683	3143	23602	27033	142.63	27033	142.63	27992	2.56	27938	2.01	27938	2.01	3.35
	10	24101	3171	27273	28845	8007.8	28845	8007.8	29486	5.18	29549	2.09	29486	5.18	2.22
R=0,05	1	21481	4706	24659	27092	5130.51	27092	5130.51	28443	10.67	28700	1.50	28443	10.67	4.99
	2	27704	4726	23286	28363	402.34	28363	402.34	30539	2.37	30811	0.97	30539	2.37	7.67
	3	24800	4807	26616	28516	101.91	28516	101.91	29996	6.27	30059	2.50	29996	6.27	5.19
	4	21709	4626	26424	28090	1542.92	28090	1542.92	29384	5.77	29114	1.37	29114	1.37	3.65
	5	26465	4679	29357	31715	100.23	31715	100.23	33002	4.80	33131	2.43	33002	4.80	4.06
	6	21474	4975	27577	28801	456.44	28801	456.44	29704	9.75	29439	2.29	29439	2.29	2.22
	7	27246	5091	26959	29814	3444.13	29814	3444.13	31319	8.55	31650	1.03	31319	8.55	5.05
	8	22704	4686	27948	28852	210.87	28852	210.87	29477	7.35	29534	1.28	29477	7.35	2.17
	9	24816	5040	21964	26294	1089.63	26294	1089.63	28347	3.39	28420	0.69	28347	3.39	7.81
	10	21790	5081	24412	26088	1181.84	26088	1181.84	27264	6.15	27352	1.78	27264	6.15	4.51
R=0,1	1	26490	6917	23821	27445	6294.14	27445	6294.14	30030	3.68	30190	1.34	30030	3.68	9.42
	2	25917	7180	24829	27842	8914.68	27842	8914.68	29868	3.76	30120	1.70	29868	3.76	7.28
	3	25054	7010	24672	26721	270.58	26721	270.58	28800	2.15	28848	1.00	28800	2.15	7.78
	4	23388	6658	29145	29943	3240.85	29943	3240.85	31081	9.03	30932	1.17	30932	1.17	3.30
	5	20600	6670	29181	29677	10.99	29677	10.99	30514	9.74	30539	1.76	30514	9.74	2.82
	6	24477	7210	23093	26103	1530.77	26103	1530.77	28661	4.63	28686	2.20	28661	4.63	9.80
	7	21298	6767	25712	26943	12.76	26943	12.76	28280	4.06	28300	1.09	28280	4.06	4.96
	8	26035	7443	28409	31639	1216.96	31639	1216.96	33457	5.59	33378	2.53	33378	2.53	5.50
	9	22550	7283	25774	27769	1339.63	27769	1339.63	29445	5.16	29708	1.03	29445	5.16	6.04
	10	23629	7170	26157	27996	708.17	27996	708.17	30036	4.71	29882	1.50	29882	1.50	6.74

Table A.20: Computational Results for N=35 – 2/3

N=35		LB 1	LB 2	LB 3	LB 4		Best Lower Bound		VNS 1		VNS 2		Min of VNS		Perc.Dev. of LB from VNS
Data Type	No				Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec,)	
R=0,5	1	31062	28256	45440	45373	0.58	45440	0	47245	10.75	47022	1.64	47022	1.64	3.48
	2	40969	28214	46486	46204	0.85	46486	0	50425	4.88	50153	1.48	50153	1.48	7.89
	3	38365	29076	36535	36419	57.89	38365	0	44685	5.27	44750	2.40	44685	5.27	16.47
	4	36709	28278	42427	42394	2.73	42427	0	46379	9.19	46818	1.75	46379	9.19	9.31
	5	39841	25938	43474	43508	2.1	43508	2.1	48496	7.19	48406	1.48	48406	1.48	11.26
	6	37917	28326	40880	26675	14.59	40880	0	46522	3.70	46901	1.42	46522	3.70	13.80
	7	33218	25344	40286	40005	0.51	40286	0	43107	2.59	43453	1.72	43107	2.59	7.00
	8	37550	27484	42817	42930	4.07	42930	4.07	47133	2.29	46765	1.50	46765	1.50	8.93
	9	35059	26582	39161	39149	5.3	39161	0	43063	5.85	42989	1.50	42989	1.50	9.78
	10	34141	29616	39814	39650	4.91	39814	0	44336	8.14	44694	0.70	44336	8.14	11.36
R=1	1	46518	54066	55887	52513	0.36	55887	0	63486	2.93	63726	1.03	63486	2.93	13.60
	2	49169	52852	48673	43409	1.1	52852	0	61913	3.43	61030	1.97	61030	1.97	15.47
	3	47388	50267	50690	47558	2.19	50690	0	61464	6.72	61096	1.01	61096	1.01	20.53
	4	39687	44238	49270	46195	2.24	49270	0	55241	4.85	55534	2.25	55241	4.85	12.12
	5	58485	57399	55365	50213	0.57	58485	0	67711	3.42	68274	1.73	67711	3.42	15.77
	6	48168	48574	53966	52010	0.37	53966	0	60431	1.98	59087	1.06	59087	1.06	9.49
	7	62652	58597	56868	53200	0.68	62652	0	69815	2.48	69099	1.37	69099	1.37	10.29
	8	50727	53517	58151	55943	0.28	58151	0	65987	11.28	66154	1.33	65987	11.28	13.48
	9	46135	51969	57796	54173	0.24	57796	0	64662	9.91	64753	1.37	64662	9.91	11.88
	10	51520	51366	53023	50934	0.48	53023	0	61887	2.86	63273	1.09	61887	2.86	16.72
R=5	1	24851	52345	47328			52345	0	53077	7.29	53077	0.73	53077	0.73	1.40
	2	34429	56116	49963			56116	0	56441	9.55	56439	1.00	56439	1.00	0.58
	3	29144	50729	43362			50729	0	51256	10.98	51256	0.89	51256	0.89	1.04
	4	31028	48570	42949			48570	0	48795	2.53	48787	0.75	48787	0.75	0.45
	5	28668	50940	47242			50940	0	52013	1.45	52013	1.34	52013	1.34	2.11
	6	27061	50708	46826			50708	0	51247	4.06	51247	0.97	51247	0.97	1.06
	7	31040	55095	49931			55095	0	55951	6.01	55960	0.64	55951	6.01	1.55
	8	33330	50498	44992			50498	0	50845	0.36	50845	0.36	50845	0.36	0.69
	9	28294	49781	44160			49781	0	50403	6.85	50403	0.95	50403	0.95	1.25
	10	26531	47583	42689			47583	0	47695	6.01	47695	0.62	47695	0.62	0.24

Table A.21: Computational Results for N=35 – 3/3

N=35		LB 1	LB 2	LB 3	LB 4		Best Lower Bound		VNS 1		VNS 2		Min of VNS		Perc.Dev. of LB from VNS
Data Type	No				Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec,)	
R=10	1	18397	50319	45009			50319	0	50737	2.82	50737	0.33	50737	0.33	0.83
	2	26550	46704	42625			46704	0	46758	0.73	46758	0.59	46758	0.59	0.12
	3	24139	48188	42083			48188	0	48501	2.28	48501	0.70	48501	0.70	0.65
	4	23394	45713	42217			45713	0	46193	1.70	46193	0.73	46193	0.73	1.05
	5	24769	46848	38376			46848	0	47091	0.41	47091	0.27	47091	0.27	0.52
	6	26536	48303	43302			48303	0	48338	1.31	48338	0.69	48338	0.69	0.07
	7	26439	49148	43854			49148	0	49296	2.26	49296	1.01	49296	1.01	0.30
	8	23676	50267	45683			50267	0	50476	2.39	50476	0.64	50476	0.64	0.42
	9	32203	63913	56332			63913	0	63913	1.39	63913	0.62	63913	0.62	0.00
	10	24158	54096	49523			54096	0	54325	2.06	54325	0.64	54325	0.64	0.42
R=50	1	23959	49251	43923			49251	0	49251	0.00	49251	0.34	49251	0.34	0.00
	2	19978	46229	40924			46229	0	46229	0.00	46229	0.41	46229	0.41	0.00
	3	24412	55487	47009			55487	0	55545	0.00	55545	0.39	55545	0.39	0.10
	4	20255	46767	41730			46767	0	46767	0.00	46767	0.34	46767	0.34	0.00
	5	21601	43864	39118			43864	0	43864	0.00	43864	0.33	43864	0.33	0.00
	6	21364	46574	40931			46574	0	46574	0.00	46574	0.33	46574	0.33	0.00
	7	23902	55201	50290			55201	0	55201	0.00	55201	0.27	55201	0.27	0.00
	8	23105	52253	48603			52253	0	52253	0.00	52253	0.25	52253	0.25	0.00
	9	19799	46656	40669			46656	0	46691	0.00	46691	0.39	46691	0.39	0.08
	10	20832	42952	37794			42952	0	42958	0.00	42958	0.67	42958	0.67	0.01
R=100	1	21558	43064	39173			43064	0	43064	0.00	43064	0.28	43064	0.28	0.00
	2	27662	54541	47132			54541	0	54541	0.00	54541	0.25	54541	0.25	0.00
	3	29763	59312	54445			59312	0	59312	0.00	59312	0.38	59312	0.38	0.00
	4	28167	57242	51922			57242	0	57242	0.00	57242	0.37	57242	0.37	0.00
	5	26344	54383	48129			54383	0	54383	0.00	54383	0.36	54383	0.36	0.00
	6	20893	47785	42200			47785	0	47785	0.00	47785	0.31	47785	0.31	0.00
	7	23883	57035	51518			57035	0	57035	0.00	57035	0.31	57035	0.31	0.00
	8	18687	44439	38222			44439	0	44439	0.00	44439	0.30	44439	0.30	0.00
	9	20449	47542	41237			47542	0	47542	0.00	47542	0.27	47542	0.27	0.00
	10	25151	51653	44314			51653	0	51653	0.00	51653	0.34	51653	0.34	0.00

Table A.22: Computational Results for N=50 – 1/3

N=50		LB 1	LB 2	LB 3	LB 5	Best Lower Bound	VNS 1		VNS 2		Min of VNS		Perc.Dev. of LB from VNS
Data Type	No						Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	
R=0,01	1	42892	4955	45277	51219	51219	53616	32.15	53159	11.42	53159	11.42	3.79
	2	44915	5100	49379	53739	53739	55902	33.49	55936	5.34	55902	33.49	4.03
	3	49385	5239	53112	56976	56976	58938	23.21	58964	6.61	58938	23.21	3.44
	4	45490	5024	32837	46195	46195	49714	40.92	49571	7.18	49571	7.18	7.31
	5	45031	5003	54370	57725	57725	59598	49.12	59195	14.87	59195	14.87	2.55
	6	36056	4654	38189	42250	42250	44052	18.13	44220	11.53	44052	18.13	4.27
	7	44028	5072	40922	48556	48556	51063	38.49	50903	5.73	50903	5.73	4.83
	8	44282	5043	46993	52872	52872	54528	25.82	55051	6.68	54528	25.82	3.13
	9	43269	4959	44883	52181	52181	54904	23.62	53875	4.17	53875	4.17	3.25
	10	49505	5304	42044	54792	54792	57409	20.22	57516	7.07	57409	20.22	4.78
R=0,05	1	43398	8154	45310	49391	49391	52782	29.63	53019	5.48	52782	29.63	6.87
	2	41303	8328	36647	44508	44508	48764	50.97	48866	6.99	48764	50.97	9.56
	3	45474	8496	49884	53096	53096	55828	20.30	56322	4.34	55828	20.30	5.15
	4	46942	8615	56323	59596	59596	62300	37.52	62025	6.46	62025	6.46	4.08
	5	51299	9308	57224	60264	60264	62828	42.89	62600	7.86	62600	7.86	3.88
	6	45425	8669	65396	66382	66382	67692	109.75	67910	8.08	67692	109.75	1.97
	7	46616	9121	46411	51157	51157	54706	42.43	54331	8.63	54331	8.63	6.20
	8	38382	8213	45591	49663	49663	52303	46.68	52421	6.79	52303	46.68	5.32
	9	49405	8950	49533	52970	52970	56254	33.67	56678	5.32	56254	33.67	6.20
	10	41607	8797	55779	58233	58233	60443	42.73	60290	6.18	60290	6.18	3.53
R=0,1	1	49557	13607	65241	67181	67181	70373	47.53	71654	5.85	70373	47.53	4.75
	2	67530	13537	57853	65961	67530	72573	25.97	72355	6.57	72355	6.57	7.14
	3	58411	13934	45485	55790	58411	63139	45.74	63635	6.90	63139	45.74	8.09
	4	51199	13354	48295	53991	53991	59492	16.57	59842	4.20	59492	16.57	10.19
	5	49002	13373	62867	63703	63703	66069	67.31	65951	4.35	65951	4.35	3.53
	6	41624	13041	45788	48401	48401	51969	18.02	52240	3.40	51969	18.02	7.37
	7	48951	13477	53711	56889	56889	61273	40.98	61770	6.57	61273	40.98	7.71
	8	54537	14094	51818	56812	56812	62629	10.20	62192	2.12	62192	2.12	9.47
	9	46459	13469	46656	51989	51989	57411	33.29	57081	8.25	57081	8.25	9.79
	10	53015	14747	51242	57374	57374	63584	38.19	63092	4.29	63092	4.29	9.97

Table A.23: Computational Results for N=50 – 2/3

N=50		LB 1	LB 2	LB 3	LB 5	Best Lower Bound	VNS 1		VNS 2		Min of VNS		Perc.Dev. of LB from VNS
Data Type	No						Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	
R=0,5	1	75348	53533	77732	77256	77732	89536	13.20	90582	3.37	89536	13.20	15.19
	2	68303	44666	80094	80135	80135	86028	15.13	86578	4.37	86028	15.13	7.35
	3	64509	52304	82685	82383	82685	88914	23.56	88813	5.62	88813	5.62	7.41
	4	66413	56574	78736	77571	78736	87781	30.08	88286	7.05	87781	30.08	11.49
	5	71241	57626	77447	75787	77447	89243	17.71	87977	5.46	87977	5.46	13.60
	6	66237	47398	85277	84406	85277	90667	50.23	91234	6.83	90667	50.23	6.32
	7	63612	49409	74104	73283	74104	80754	13.60	80795	5.44	80754	13.60	8.97
	8	72715	43368	76085	76211	76211	84965	12.00	84732	5.48	84732	5.48	11.18
	9	72166	45536	72208	71671	72208	85976	14.60	85896	6.77	85896	6.77	18.96
	10	65703	52216	73085	72474	73085	82587	21.92	82560	3.48	82560	3.48	12.96
R=1	1	111914	111817	109331	101704	111914	133934	18.25	133264	6.58	133264	6.58	19.08
	2	86782	95213	106727	103692	106727	116004	56.80	116776	7.69	116004	56.80	8.69
	3	96760	99603	99569	92224	99603	115066	21.45	113980	7.77	113980	7.77	14.43
	4	97375	95081	100244	93616	100244	119991	17.89	118579	6.62	118579	6.62	18.29
	5	90497	96099	109998	103844	109998	120445	55.46	120025	7.00	120025	7.00	9.12
	6	98408	108580	122704	116898	122704	136207	20.89	134072	10.90	134072	10.90	9.26
	7	100611	107088	120035	115174	120035	130726	40.19	132531	4.12	130726	40.19	8.91
	8	112096	101551	107574	99592	112096	129159	30.95	130058	7.15	129159	30.95	15.22
	9	97609	95242	99105	90318	99105	115736	35.94	116657	5.77	115736	35.94	16.78
	10	113444	109791	109741	97858	113444	131625	43.23	133289	7.21	131625	43.23	16.03
R=5	1	66845	101803	88536		101803	102364	28.58	102438	3.31	102364	28.58	0.55
	2	60163	108432	97600		108432	108980	42.98	108994	3.17	108980	42.98	0.51
	3	61747	106501	96948		106501	107697	4.48	107430	3.12	107430	3.12	0.87
	4	51977	104340	93268		104340	105534	3.84	105534	2.11	105534	2.11	1.14
	5	45770	83537	74093		83537	85039	4.03	85095	2.18	85039	4.03	1.80
	6	62112	98162	88802		98162	99064	6.72	99089	3.31	99064	6.72	0.92
	7	55177	94368	84396		94368	96886	39.45	97370	3.28	96886	39.45	2.67
	8	55330	100655	89932		100655	101751	3.70	101751	3.04	101751	3.04	1.09
	9	50522	90038	82113		90038	91358	22.62	91465	11.11	91358	22.62	1.47
	10	56733	94121	79598		94121	96102	6.88	96001	7.75	96001	7.75	2.00

Table A.24: Computational Results for N=50 – 3/3

N=50		LB 1	LB 2	LB 3	LB 5	Best Lower Bound	VNS 1		VNS 2		Min of VNS		Perc.Dev. of LB from VNS
Data Type	No						Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	Obj. Val.	Time (sec.)	
R=10	1	48573	103375	89559		103375	104170	22.65	104170	2.25	104170	2.25	0.77
	2	50581	99724	87276		99724	100282	16.32	100282	2.31	100282	2.31	0.56
	3	40835	97445	86828		97445	98046	20.78	98025	3.20	98025	3.20	0.60
	4	45357	88554	77691		88554	88877	15.34	88889	4.35	88877	15.34	0.36
	5	52460	108794	96578		108794	109275	14.63	109275	2.20	109275	2.20	0.44
	6	46280	96052	81893		96052	96346	2.90	96346	2.12	96346	2.12	0.31
	7	52397	97686	87387		97686	98026	2.09	98026	2.09	98026	2.09	0.35
	8	48439	90426	80199		90426	90904	19.80	90904	3.34	90904	3.34	0.53
	9	48917	108993	94212		108993	109149	16.40	109149	2.29	109149	2.29	0.14
	10	54055	98667	86073		98667	98960	16.41	98981	4.46	98960	16.41	0.30
R=50	1	41007	97041	87101		97041	97041	1.98	97041	1.03	97041	1.03	0.00
	2	46455	107824	95186		107824	107870	1.28	107870	2.00	107870	2.00	0.04
	3	43258	103443	90626		103443	103443	2.73	103443	1.87	103443	1.87	0.00
	4	53875	103146	93235		103146	103146	2.01	103146	0.89	103146	0.89	0.00
	5	44927	89373	80006		89373	89373	1.36	89373	1.16	89373	1.16	0.00
	6	42360	85208	75334		85208	85214	4.68	85214	2.12	85214	2.12	0.01
	7	48779	104602	94384		104602	104602	1.78	104602	1.05	104602	1.05	0.00
	8	47757	109480	94959		109480	109480	3.65	109480	2.14	109480	2.14	0.00
	9	43020	89194	79987		89194	89194	1.92	89194	1.08	89194	1.08	0.00
	10	38348	98658	82682		98658	98685	1.61	98685	1.11	98685	1.11	0.03
R=100	1	44591	95179	83854		95179	95179	1.00	95179	0.86	95179	0.86	0.00
	2	54198	103794	95511		103794	103794	1.11	103794	1.17	103794	1.17	0.00
	3	51832	104384	89842		104384	104384	1.06	104384	0.95	104384	0.95	0.00
	4	44479	97125	87263		97125	97125	0.89	97125	1.20	97125	1.20	0.00
	5	47591	97237	83242		97237	97237	1.15	97237	0.98	97237	0.98	0.00
	6	44065	100375	91948		100375	100375	0.95	100375	0.98	100375	0.98	0.00
	7	42788	87169	77506		87169	87169	1.14	87169	1.08	87169	1.08	0.00
	8	36999	87115	79127		87115	87115	1.01	87115	0.92	87115	0.92	0.00
	9	45173	97704	86324		97704	97704	1.01	97704	1.12	97704	1.12	0.00
	10	40717	103502	91716		103502	103502	1.15	103502	0.94	103502	0.94	0.00

Table A.25: Computational Results for N=100 – 1/2

N=100		LB 1	LB 2	LB 3	LB 5	Best Lower Bound	VNS 2		Perc.Dev. of LB from VNS
Data Type	No						Obj. Val.	Time (sec.)	
R=0,01	1	194239	15414	184818	216573	216573	225259	80.84	4.01
	2	161310	14869	159859	186852	186852	194040	87.83	3.85
	3	170354	15010	158265	190665	190665	200380	59.37	5.10
	4	187485	15254	191732	210119	210119	216992	65.72	3.27
	5	186120	15380	163840	202531	202531	210743	84.24	4.05
	6	200588	15614	175641	212441	212441	220776	61.90	3.92
	7	209202	15617	192482	217207	217207	225906	66.80	4.00
	8	170854	15009	182242	201560	201560	209789	106.22	4.08
	9	187193	15326	169416	202974	202974	213087	54.88	4.98
	10	169684	14924	188055	202028	202028	208946	92.26	3.42
R=0,05	1	162501	31428	205090	214109	214109	222001	55.82	3.69
	2	185774	29220	173559	197286	197286	210390	71.26	6.64
	3	167986	29880	182398	196906	196906	208453	81.03	5.86
	4	173019	29090	180173	193855	193855	204430	85.75	5.46
	5	233326	31832	190552	230223	233326	248943	87.92	6.69
	6	172879	27910	196021	205678	205678	215232	67.44	4.65
	7	178857	29873	195015	208369	208369	217348	101.04	4.31
	8	199565	30780	193642	211351	211351	223528	61.09	5.76
	9	205375	31303	195115	213666	213666	228241	98.23	6.82
	10	179649	29664	195819	207250	207250	218588	61.56	5.47
R=0,1	1	199965	48645	211274	223001	223001	237323	35.51	6.42
	2	171351	49980	188271	200970	200970	215407	66.96	7.18
	3	204882	47467	229011	238750	238750	250822	81.71	5.06
	4	188504	52565	194778	208868	208868	224559	94.65	7.51
	5	202577	52042	217047	231466	231466	246558	49.36	6.52
	6	200824	50970	194269	210610	210610	228318	39.70	8.41
	7	185162	48359	224737	233287	233287	247423	112.36	6.06
	8	183643	47170	181355	202996	202996	222503	59.34	9.61
	9	190304	47944	201660	210796	210796	227616	168.53	7.98
	10	181017	48870	189177	199405	199405	215677	120.94	8.16
R=0,5	1	321024	204359	275619	277253	321024	350834	58.39	9.29
	2	316099	197692	269280	267769	316099	343332	96.19	8.62
	3	289282	193877	304751	306273	306273	342456	80.92	11.81
	4	295310	198528	284573	282354	295310	334122	107.11	13.14
	5	288238	201512	284114	282544	288238	328347	46.60	13.92
	6	285105	194982	302419	300099	302419	334450	54.37	10.59
	7	261847	183358	304935	302312	304935	331739	48.80	8.79
	8	259677	199927	315811	313920	315811	335086	66.68	6.10
	9	305916	216576	299553	296743	305916	350297	173.98	14.51
	10	291143	209773	299881	301699	301699	343149	102.27	13.74
R=1	1	394240	419463	442119	413229	442119	493080	162.10	11.53
	2	402687	435306	484229	464692	484229	533579	47.71	10.19
	3	385124	400139	460183	444891	460183	501176	97.96	8.91
	4	394170	414853	423758	400115	423758	481083	109.27	13.53
	5	366001	392778	416346	389294	416346	469340	120.67	12.73
	6	421227	418627	422139	391472	422139	493011	97.14	16.79
	7	405683	414437	434719	395350	434719	495523	184.07	13.99
	8	383073	376694	386491	356333	386491	454974	206.63	17.72
	9	391582	431516	453714	426121	453714	499256	62.85	10.04
	10	400247	411838	440765	418975	440765	497217	89.47	12.81

Table A.26: Computational Results for N=100 – 2/2

N=100		LB 1	LB 2	LB 3	LB 5	Best Lower Bound	VNS 2		Perc.Dev. of LB from VNS
Data Type	No						Obj. Val.	Time (sec.)	
R=5	1	220341	421482	377876		421482	424094	47,46	0,62
	2	211246	394869	345033		394869	398432	72,59	0,90
	3	253640	422614	372552		422614	425987	82,67	0,80
	4	222938	394958	343334		394958	398126	75,29	0,80
	5	242623	430457	384307		430457	434647	31,09	0,97
	6	230479	413148	368248		413148	417241	41,24	0,99
	7	231812	428057	383164		428057	430843	34,01	0,65
	8	237112	425058	362366		425058	427745	70,22	0,63
	9	210756	387245	337609		387245	391954	29,30	1,22
	10	219405	370952	330628		370952	373813	35,04	0,77
R=10	1	193142	389914	339377		389914	391602	23,40	0,43
	2	224636	441279	391664		441279	442052	31,06	0,18
	3	243329	432158	380972		432158	432555	22,29	0,09
	4	202827	371870	325292		371870	372086	36,89	0,06
	5	207024	402909	350901		402909	403230	32,53	0,08
	6	179202	354179	313869		354179	355076	33,08	0,25
	7	198860	398507	358346		398507	400607	19,78	0,53
	8	200718	396279	342118		396279	397655	50,40	0,35
	9	188356	383137	333741		383137	384160	45,52	0,27
	10	216610	425093	371406		425093	425689	25,07	0,14
R=50	1	214116	442208	396871		442208	442478	20,88	0,06
	2	197106	404831	365778		404831	404931	10,81	0,02
	3	198859	375557	344114		375557	375557	10,92	0,00
	4	195617	416211	351251		416211	416212	21,30	0,00
	5	161850	390137	320818		390137	390137	21,92	0,00
	6	201956	390749	343236		390749	390749	18,44	0,00
	7	195098	415996	368874		415996	416165	21,59	0,04
	8	174137	397786	346820		397786	397871	33,96	0,02
	9	155736	374066	320683		374066	374164	19,40	0,03
	10	196718	403610	350099		403610	403610	22,64	0,00
R=100	1	178041	398192	343171		398192	398192	10,37	0,00
	2	143360	377724	328730		377724	377724	9,59	0,00
	3	176300	405122	349945		405122	405122	10,83	0,00
	4	177644	393461	334344		393461	393461	10,17	0,00
	5	221726	447518	393229		447518	447518	9,38	0,00
	6	168244	402691	349113		402691	402691	10,93	0,00
	7	168880	395686	344113		395686	395686	10,48	0,00
	8	179573	382233	333886		382233	382233	9,63	0,00
	9	186940	419608	357699		419608	419608	10,67	0,00
	10	163667	385012	339262		385012	385012	9,83	0,00