POLITECNICO DI MILANO
SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE
MSc Mechanical Engineering - Production Systems

MSc Thesis

A Data-driven Methodology for the Re-balancing of Paced Assembly Lines.

Supervisor:
ANDREA MATTA
Co-supervisor:
RAIK STOLLETZ
JOHANNES DIEFENBACH

Candidate: FLAVIO PRINCIOTTA
ID: 998829

Abstract

L'Assembly Line Balancing Problem (ALBP), consiste nel trovare un'assegnazione di ogni task del processo di assemblaggio ad una e una sola stazione di lavoro. L'ALBP è reso più complesso dalla stocasticità dei vari processi di assemblaggio nelle linee moderne. Questo tipo di problema viene affrontato sia nella fase di design della linea (Greenfield) che nelle fasi di modifica di una linea già esistente (Brownfield), nel qual caso si parla di re-balancing. La caratteristica del re-balancing è la possibilità di raccogliere dati direttamente dalla linea già esistente. Questa tesi va ad investigare il re-balancing di linee di assemblaggio di tipo paced, caratterizzate da tempi di processo variabili con l'obiettivo di minimizzare il numero di stazioni da aprire, dati sia il tempo di ciclo che un'affidabilità minima richiesta della linea. Lo studio punta allo sviluppo di una metodologia totalmente data-driven in grado di risolvere un re-balancing partendo dal Reliability-based Branch and Bound presentato in Diefenbach and Stolletz (2020), ma utilizzando un ridotto numero di osservazioni dei tempi di processo. L'utilizzo dell'algoritmo viene inserito all'interno di una procedura che trae ispirazione dalla metodologia di Bootstrap, andando a creare degli scenari da analizzare tramite un ricampionamento con sostituizione delle osservazioni iniziali, così da diminuire il numero di dati necessari all'ottenimento di un risultato affidabile. L'algoritmo viene inoltre affiancato a un processo di ottimizzazione che mira a massimizzare la capacità della linea di assemblaggio di svolgere tutte le attività richieste in una stazione entro il tempo limite. L'analisi svolta va a mostrare come la procedura sviluppata riesce a ridurre radicalmente la quantità di dati originariamente richiesta dal Branch and Bound sviluppato da Diefenbach and Stolletz (2020), comparandola non solo con quest'ultima ma anche con altri approcci che utilizzano lo stesso algoritmo originale. I risultati mostrano non solo una riduzione dei dati necessari, ma in molti casi anche una miglior performance in termini di affidabilità della linea. Tutti i test numerici sono stati fatti su linee con tempi di processo variabili, generate a partire da alcuni modelli deterministici presenti in letteratura.

Abstract

The Assembly Line Balancing Problem (ALBP) consists in the allocation of every task of the assembly process to exactly one workstations. The ALBP is made more complex by the stochasticity of the processes in the modern assembly lines. This type of problem is faced in both the design phase of the line (Greenfield) and the modification phases of already existing lines (Brownfield), in this case it is possible to talk about re-balancing. The main characteristic of the re-balancing is the possibility to collect data directly from the existing line. This thesis investigates the re-balancing of paced assembly lines, characterized by stochastic task times, with the objective to minimize the number of workstations to open, given not only the cycle time but also a minimum reliability required. The study is focused on the development of a methodology able to solve the re-balancing problem from the Reliability-based Branch and Bound presented by Diefenbach and Stolletz (2020), but using a reduced number of observations of the task times. The algorithm is inserted in a procedure inspired by the Bootstrap methodology, creating different scenarios using the re-sampling with replacement of the initial observations, in order to reduce the required number of samples. Furthermore, it is sequenced with an optimization process with the objective of maximizing the ability of the assembly line to complete all the activity required in a workstation within the defined cycle time. The analysis shows how the developed procedure is able to drastically reduce the quantity of data required by the Reliability-based Branch and Bound developed by Diefenbach and Stolletz (2020), comparing it not only to this algorithm but also to other two approaches developed starting from that. In addition, the results highlight in many cases also the capability to reach a balancing that performs better in terms of reliability of the line. All the numerical tests are done on lines with variable task times, created starting from some deterministic model in literature.

Contents

List of Figures

List of Tables

Nomenclature

$\bar{y}_i$ Mean between the $j-th$ observations for the $i-th$ configuration.

δ^* Indifference-zone

B Number of Bootstrap Replications

B_m Sample-variable

B_n^b Sample-variable to solve the task allocation problem for the $b-th$ bootstrap sample

c Cycle Time Decided

DS Original Data-set with observations of Task Times

DS^b $b-th$ Bootstrap Sample, generated resampling with replacement DS

DS^n Data-sets generated resampling with replacement DS in the ranking and selection procedure.

I Set of the alternatives not yet discarded. $I(k) = 0$ if $OTAS^k$ is discarded, $I(k) = 1$ otherwise.

M Vector of opened station. $M(m)$ is equal to 1 if station m is opened, 0 otherwise

M^b Resulting number of station to open, applying RB&B to DS^b

n_0 Number of initial Data-sets generated in the ranking and selection procedure.

OTA^b Optimized Task Allocation for the $b-th$ bootstrap sample

$OTAS$ Set of optiized task allocations(with no repetitions)

$OTAS_k$ $k-th$ element in $OTAS$

P^P Probability of achieving an optimal selection

R Reliability

$R_{k,b}$ Resulting Reliability of $OTAS_k$ tested on DS^b

S_{il}^2 Variance of the difference between the alternatives i and l

$SSOTA$ Subset of assignments that contains the optimal with probability $1-\alpha$

t_i time of the $i-th$ task

$X_{i,m}$ Assignment-variable

$X_{i,m}^b$ Station-variable to solve the task allocation problem for the $b-th$ bootstrap sample

Y_{ij} $j-th$ observation of the Reliability in the $i-th$ configuration.

Z_m Station-variable

Z_m^b Resulting Station-variable applying the RB&B to DS^b

Chapter 1

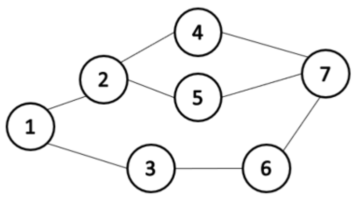
Introduction

The development of the assembly lines was a pivotal milestone for the evolution of the humankind. The concept became popular after the second industrial revolution, with the increase of the mass production. Moreover, some evidence of production line organizations is present even before the industrial revolution, however they never became popular in the industrial field. The invention of the first assembly line must be accredited to Ransom Eli Olds, founder of the Oldsmobile. Thus, he was the first inventor of this kind of process, with the purpose of producing the Oldsmobile Curved Dash. In order to understand the possible benefits of this production philosophy, Olds increased its production of 500%. Ford upgraded the same concepts to a larger scale for its Model T. The concept of the assembly line is based on the breakdown of a complex production process, into a series of simpler tasks. Consequently, this helps the increase of the effectiveness of the single operations, increasing the production volumes and reducing dramatically the related costs. The development of more efficient machines and transportation systems (e.g. steam powered conveyor belts) increases the potentiality of this kind of production, making it the basis of the mass production.

1.1 Definitions

The Assembly line is a production system composed by "a line of machines and workers in a factory that a product moves along while it is being built or produced. Each machine or worker performs a particular job that must be finished before the product moves to the next position in the line." (Cambridge Dictionary)

The single operations can be called "task" and, when dealing with the balancing, it is considered indivisible. The duration of a task i is called task time t_i. The task time can be deterministic (known and constant) or stochastic (affected by a certain variability). Also, the tasks are grouped in a series of workstations j and the sum of the task times associated to the workstation is the station time. It is possible to define the cycle time c as the time between the completion of two following finished products. The production rate is the inverse of the cycle time. The cycle time and the station time can differ, leaving an amount of idle time to the workstation. The assignment of a task to a workstation is, in most of the cases, constrained by a series of technological aspects creating some precedences between the tasks that must be respected. Various methods can be used to express and represent those constraints, such as the precedence diagrams and/or matrixes like the Hoffman matrix. (Hoffmann, 1963)



(a)

Operation	Predecessors
1	-
2	1
3	1
4	2
5	2
6	3
7	4,5,6

(b)

	1	2	3	4	5	6	7
1	0	1	1	0	0	0	0
2	0	0	0	1	1	0	0
3	0	0	0	0	0	1	0
4	0	0	0	0	0	0	1
5	0	0	0	0	0	0	1
6	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0

(c)

Figure 1.1: Precedence Graph (a); Precedence Table (b); Hoffman Matrix (c).

In the precedence diagram, each task is represented by a node and the precedences are expressed through the usage of some arcs. An arc pointing from j to k represents a precedence of the task j on the task k. For what concern the matrix, the presence of N_t tasks will create a $N_t \times N_t$ matrix with rows and columns labelled with consecutive task numbers. The position $[j, k]$ will be equal to 1 if task j immediately precedes task k, 0 otherwise.

1.2 Assembly Line Balancing Problem

A balance of the line is crucial to improve and to optimize the performances of the line itself. The Assembly Line Balancing Problem (ALBP) consists in finding a feasible line balance, which is an assignment of each task to exactly one workstation. This must be done under a series of constraints that include also the precedences.

It is possible to find in literature a large variety of approaches to the problem. These kinds of solutions can be divided into two macro categories: Simple Assembly Line Balancing Problems (SALBP) and General Assembly Line Balancing Problems (GALBP).

1.2.1 Simple Assembly Line Balancing Problem (SALBP)

The Simple Assembly Line Balancing Problems (SALBPs) are more studied with respect to their general counterpart. They have in common a series of hypothesis:

1. The assembly line refers to the mass-production of one homogeneous product (unique).
2. The production process is considered given, it means that all the tasks are perfomed in a predefined way and there are no production alternatives.
3. The line is paced defining a fixed cycle time.
4. The only assignment restrictions are provided by the precedence constraints.
5. The line is considered to have a serial layout.
6. Deterministic task times are considered.
7. No differences in terms of equipment and workers of the workstations are considered.
8. Any task can be performed at any workstation, without any zoning restrictions.
9. Any task must be assigned to maximum one workstation (they cannot be split among two or more stations).

Four different types of SALP can be identified in literature (Scholl and Becker, 2006a):

- SALBP - I: this type of problem was the most intensively studied. This has the aim of the minimization of the Number of Workstations (m) given a certain Cycle Time c.

- SALBP - II: it is strictly related to the previous type of problem. Its aims are the minimization of the Cycle Time (c) given a specific Number of Workstations m.
- SALBP - E: The aims of this approach is to maximize the Efficiency (E) of the line trying to minimize both the Cycle Time (c) and the Number of Workstations (m) maintaining the feasibility and considering their interrelationship. The line efficiency can be evaluated as the ratio between the sum of all the task times and the product between the cycle time and the number of workstations.
- SALBP-F : this is a feasibility problem with the aim of understanding if a feasible solution exists given a specific Cycle Time c and a specific Number of Workstations m.

1.2.2 General Assembly Line Balancing Problem (GALBP)

The manufacturing systems nowadays are much more complex than in the past. consequently, this led to a situation in which the assumptions used in the SALBP are too simplistic and they cannot be used to represent a system. The actual systems are characterized by a wide variety of constraints, layouts as well as objectives. Those are the reasons that brought the research toward an evolution, trying to keep into account possible aspects such set up times between the tasks, parallel workstations and zoning restrictions. This new kind of approach was called General Assembly Line Balancing Problem (GALBP). The hypothesis previously made, must be fine-tuned in order to meet the new types of problems (Baybars, 1986):

1. One or more products can be manufactured.
2. A Set of processing alternatives can be given. Those are given by the fact that in the design phase different processes are available to reach the same goal. This result in different task sequences or even different tasks numbers, by aggregating or desegregating some processes.
3. The line must be configured such that certain production quantities are satisfied for certain planning horizon.
4. The flow of the line is unidirectional (not necessarily serial).
5. The processing sequence of tasks is subjected to precedence restrictions.

It can be seen how there is no longer the necessity of deterministic task times, there can be other restriction besides the precedence, the task in some cases can be split among two or more stations and all the stations are not necessarily equally equipped. Morover, it was largely demonstrated that the presence of the human tasks creates a variability inside the line. This can be addressed to variable status of the human's skills level, morale, concentration and more. The GALBP will be the topic of this thesis and the literature about it will be further analysed in the following sections.

1.3 Scope of this study

Nowadays, almost all the methodologies to balance and to rebalance an assembly line are based on the assumption that the task times follow specific distributions. Most of the literature is based on normally distributed times.

In order to overcome this limit, a new approach was developed by Diefenbach and Stolletz (2020). The details of their methodology are going to be explained in Section 3.2. The new approach is totally independent from the distribution of the task times. Nevertheless, a new limit appears from this, the need of a huge quantity of observation of the task times.

The aim of this research is to find a procedure that, starting from the algorithm developed by Diefenbach and Stolletz (2020), is able to overcome both the limit of the knowledge of the distribution and of the huge amount of data required, being characterized by:

- No assumption on task times distributions.
- Requires a reasonable amount of data to work.

Since the work is focused on the usage of the data collected directly from the assembly lines, the collection of data requires the existence of the line. Therefore, it is possible to talk about re-balancing of the line. This concept does not differ from the "balancing" one except for the application. Usually, the re-balancing situation is addressed as "Brownfield" approach; on the other hand, the building of a new line from the ground up is addressed as "Greenfield" approach.

The work related to the creation of a new line does not end with the opening and this stresses out the centrality of the re-balancing. Due to the complexity of the modern lines, affected by a huge number of variables and characteristics, it is practically impossible to open an already perfectly optimized line.

Moreover, the stochasticity of the task can be related also to the presence of manual operations. This type of tasks are characterized by a learning curve given by the improvement of the skills of the worker, that represent another source of complexity.

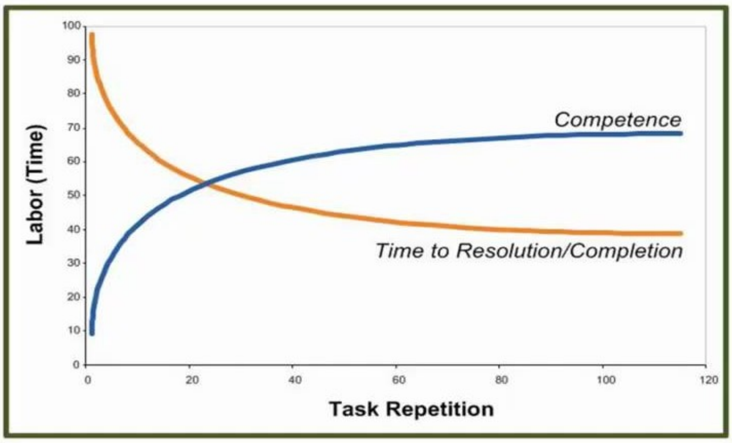


Figure 1.2: Representation of the Learning Curve.
Source: Danford (2011)

Therefore, after the installment, the assembly line requires a period of observation, to perform some improvements and modifications. This is the phase of the line launch on which this work is mainly focused. The aim is to use the data collected during the observation of the line in the first phase of the launch, after the warm up and after the period of training of the workers, in order to identify an assignment of the tasks that improve (if it exists) the configuration originally chosen.

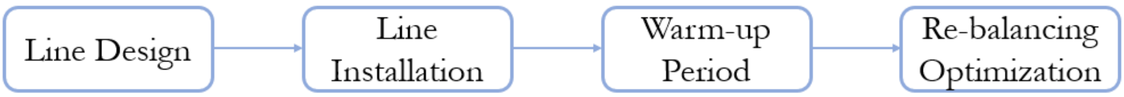


Figure 1.3: Phases of an Assembly Line opening

1.4 Literature Contribution

The research developed in this thesis finds a specific spot in the literature about the Assembly Line Balancing.

This work is focused on the Stochastic Assembly Line Balancing Problem. The assembly lines considered are paced lines characterized by stochastic task times. In the industrial field there are two main ways of managing the stochasticity on the assembly lines:

- Buffer Allocation: The usage of buffers between station allows to contain the uncertainty given by the stochasticity. In this case, the study of the line is not focused only on the allocation of the tasks to the workstations, but includes also the buffer allocation problem. If the line is not properly balanced and the buffers properly dimensioned, the line can suffer of blocking and starving phenomena that can affect radically its performances. Usually, with this approach, the stations are not forced to work within a cycle time, therefore this solution is mainly used on the unpaced lines.

- Policies for non completion: In this case, specific policies are used for workpieces exceeding the cycle time at a station. Two main types of policies are available. On one side, the line is stopped until the late job is completed, on the other side the job is completed outside the line. The decision is usually taken after a cost assessment. This approach is mainly used on the paced lines.

Since this work is focused on paced lines, the second approach, with the completion outside the line, is the one considered. Due to the fact that in the research there is no focus on the completion costs, there is no need to specify whether the pieces that exceed the cycle time, conclude all the path along the line, doing all the work that can be done without the completion of the previous task, or not.

Furthermore, no failures of the machines are considered. The failures can be somehow considered in the stochasticity itself of the task times, but in general this takes more into account micro stoppages of the machines and not a long failure that can last for longer period of time. In practical terms, it will be shown in future sections how the developed methodology is based on the observation of the task times, therefore can include long stoppages.

In terms of problem type, the one treated in this work is the stochastic ALBP of Type 1, aimed at the minimization of the number of workstations to open, given the cycle time.

Dealing with the stochastic ALBP, the reliability must be considered. It is the capability of the system to complete the tasks assigned to a station, without exceeding the cycle time. It is possible to talk about chance-constrained problem, when some constraints are applied on this parameter. In literature, two main approaches to consider the reliability are found.

The first one sets a constraint on the reliability of the single station. This approach is the most frequent. The problem here is that it cannot handle problems with a large number of workstations, because it does not take into account the reliability of the entire line, that decreases exponentially with the increase of the number of workstations.

The second one considers the reliability of the entire line. This approach to the problem is the least investigated in literature. In particular, only four articles are found managing the problem like this.

Liu et al. (2005) developed a bidirectional construction and a trade and transfer heuristic procedure to minimize the cycle time. Chiang et al (2016) aim at the minimization of the station of two-sided assembly lines, presenting a particle swarm optimization algorithm, while Tang et al. (2017), with the same aim, a hybrid teaching learning-based heuristic algorithm.

Finally, Diefenbach and Stolletz (2020) present a chance-constrained Reliability-based Branch and Bound algorithm with the aim of minimizing the number of workstations. The algorithm is the baseline of this research and will be further explained in the following sections.

Looking at the chance constrained model, it is possible to divide the literature into two classes:

- **Class I**: It is based on the assumption that the distributions followed by the task times are known. Furthermore, two sub-classes can be distinguished. The first one requires the assumption of specific distribution followed by the task times. Most of those cases in literature are based on normal distribution. In the second one, any distribution can be followed.

- **Class II**: it does not require any assumption on the distributions followed by the processing times. Those approaches are based on the usage of the observation of the task times, it means that they are fully data-driven procedures.

According to this differentiation, it can be interesting to reason about the motivations to use one of the two approaches. When dealing with the ALBP it is possible to highlight two industrial situations, depending on the "status" of the line. As previously said, it is possible to talk about greenfield when the line must be built from scratch, while the problem is addressed as brownfield, when the line already exists and must be modified and improved.

When dealing with the Class I methodologies, the mean and the variance or even the Probability and the Cumulative Density Functions are required. The knowledge of the distributions' characteristics can come from both a theoretical assumption about the tasks or from the analysis of datasets collected from an existing line. The methodologies can either use directly those informations, or can sample from the distribution to perform a Montecarlo simulation or similar. But with the knowledge of the distribution, there is no limit about the data that can be sampled. This kind of methods can be used for both Greenfield and Brownfield approach.

When using the Class II methodologies, the situation changes. Being fully data-driven, these methodologies requires the existence of the assembly line to collect the data. It makes them suitable only for the brownfield problem.

The focus of this work will be on this second class of problems and approaches. It will be possible to discover how the algorithm proposed by Diefenbach and Stolletz (2020) has excellent results when placed in the Class I, with no limitation on the data that can be generated from the distributions. The problem arises when it is included in the Class II approaches. This is due to the unreliability of the results of the algorithm, when it is fed with reduced data-sets. This condition is extremely probable in the reality of the industrial field. The collection of data on the assembly lines has a non-negligible cost, due to the resources, technology and/or workforce required. Therefore, it is common to be in the situation of a lack of observations of the task times, that does not allow a proper analysis to estimate the followed distribution.

The contribution of this research has two main objectives:

- Fill the gap in literature about fully Data-driven re-balancing methodologies for the assembly lines.

- Develop a procedure that is able to use the potentiality of the RB&B developed by Diefenbach and Stolletz (2020) with a reasonable amount of data to collect.

Chapter 2

Literature Review

The GALBP embraces a big window of possibilities, in terms of constraints and objectives. Boysen et al. (2007) introduced a classification to create a system approach to identify various types of GALBP. This classification is performed according to three main dimensions:

- Precedence Graph Characteristics.
- Station and line characteristics.
- Objectives.

2.1 Precedence Graph Characteristics

A precedence graph consists of nodes representing the tasks of the entire process, as previously described. In addition, six features can characterize it:

1. Product Specific Precedence Graph: the attribute explain whether a single product is produced or several one must be considered simultaneously. For the Assembly line balancing problem it is more important the degree of homogeneity between graphs than the number of different precedence graphs. The attribute can assume three type of values:
   (a) Single-model Line: the line can be related to a single product assembled, or by different products characterized by the same precedence graph. In this latter case, the various products do not need to be distinguished.

(b) Multi-Model Line: different products are produced on the line, but the production is organized in batches. Between the production of different batches, a set up occurs, requiring non negligible amount of time and resources.

(c) Mixed-Model Line: different models are present on the same system, but their production process is similar enough to consider negligible setup times. The different models are produced in an intermixed sequence.

2. Structure of the precedence graph: some researches are focused on precedence graphs with special structures in order to develop better algorithms. The attribute can assume two values:

   (a) Special: the research is restricted to special structures like linear, diverging or converging structures.

   (b) Common: the graph can have any acyclic structure.

3. Processing Times: task times can vary due to manual operations or machinery. Three different types of times can be highlighted.

   (a) Static and Deterministic: when the times have a very low variability. This is the case of highly automated and reliable processes.

   (b) Stochastic: when the variability of the time must be considered. This is the particularly case of manual operations, where the variability is introduced by the status of the worker in physical, physiological and psychological terms.

   (c) Dynamic Variation: this possibility arises due to some aspects like the learning effect of the operators.

4. Sequence Dependent task time increments: those occurs when the sequence of the operations influences their processing times. Three type of indicators can be defined:

   (a) No Consideration of sequence dependent task time increments.

   (b) Possible additional time when two operations are executed one after the other inside a station. This can be due to some specific setup operations.

   (c) Additional time when the status achieved completing a particular task has an effect on the processing time of other tasks executed later in the same or in another station.

5. Assignment restrictions: those can affect the grouping of tasks inside a station, forcing or forbidding certain combinations. This attribute can assume nine different values:

   (a) No assignment or restriction considered.

   (b) Link: a subset of tasks are linked, so that they must be assigned to the same station.

   (c) Incompatibility: a subset of tasks are incompatible and cannot be assigned to the same station.

   (d) Cumulative: The subset of tasks is subjected to constraints on the cumulated value of a particular attribute.

   (e) Fixed: some tasks can be assigned only to specific stations, due to some restriction on the movement of the resources.

   (f) Exclusive: some tasks cannot be assigned to some stations.

   (g) Type-based: some tasks must be assigned to stations of a certain type.

   (h) Minimum: the assignment of a task must consider minimum distances in terms of space times or sequence.

   (i) Maximum: the assignment of a task must consider maximum distances in terms of space times or sequence.

6. Processing alternatives: when there are different possible alternatives, the process can change. The problem that arises is the decision, among the others, of processing alternative. The attribute can assume 2 values:

   (a) No consideration of the processing alternative.

   (b) pa λ: differentiation of possible different alternatives, according to the value of lambda.

2.2 Station and line characteristics

The line characteristics can be classified according to six attributes:

1. Movement of the workpieces: it is possible to distinguish two different type of lines.

(a) Paced Lines: in a paced assembly production system typically a common cycle time is given, which restricts process times at all stations. "The pace is either kept up by a so called intermittent transport, where the work-piece comes to a full stop at every station, but it is automatically transferred as soon as a given time span is elapsed, or by continuously advancing material handling device, e.g. a conveyor belt, which forces operators to finish their operation before the work-piece has reached the end of the respective station". While in the case of the single-model production this time has to be fulfilled, in the case of mixed-model production it is usually fulfilled on average. In the case of the stochastic task times, the restriction is respected with a certain probability. (Figure 2.1)

		Cycle								
		1	2	3	4	5	6	7	8	9
Station	St1	P1	P2	P3	P4	P5	P6	-	-	-
	St2	-	P1	P2	P3	P4	P5	P6	-	-
	St3	-	-	P1	P2	P3	P4	P5	P6	-
	St4	-	-	-	P1	P2	P3	P4	P5	P6
		c	2c	3c	4c	5c	6c	7c	8c	9c

Figure 2.1: Paced Assembly Line example.

(b) Unpaced Lines: "In unpaced lines, workpieces are transferred whenever the required operations are completed, rather than being bound to a given time span. It can be further distinguished as to whether all stations pass on their workpieces simultaneously (synchronous) or whether each station decides on transference individually (asynchronous)" (Boysen et al., 2007). In case of synchronous configuration all the stations must wait for the bottleneck (slowest station) to finish its operations, before all the workpieces are moved to the next station. In case of asynchronous configuration, the presence of some buffers placed between the workstations, allows the movement of the workpieces as soon as the station in which they are, finish its operations. This allows to decouple the workstations giving more flexibility to the system.

2. Line Layout: it distinguishes the fact that the stations can be organized in a line configuration or in a U-shape line layout. The latter can be characterized also by n U-shaped segments.

3. Parallelization: there could be the possibility in which more than one parallel

line should be balanced and configured. The parallelization can include also the duplication of the resources in a single workstation.

4. Resource Assignment: usually, different type of resources can be necessary, in order to provide technological capabilities in terms of station equipment. It is possible to distinguish between:
   (a) No explicit consideration of the resources.
   (b) One equipment is chosen for each station, out of a set of alternatives. In this case the balancing problem includes the equipment selection problem. It is possible to talk in this case of the assembly line design problem.
   (c) The equipment is configured in a station, along with the task assignment.

5. Station Dependent Time increments: the time of the workstation can increase for different type of non-productive activities. Those unproductive activities can be the transportation of the workpieces or the movement of the worker to at the end of the cycle to return at the beginning of the station. It is therefore possible to distinguish the cases when this happens or not.

6. Additional aspects of the line configuration: some additional aspects could be necessary to be considered to balance the line. In those aspects there are:
   (a) Buffers: if they are required, they must be allocated and dimensioned.
   (b) Feeder: possible presence of feeder lines that requires another process of task assignment.
   (c) Material Boxes: they could be present and therefore dimensioned.
   (d) Changes: in terms of the position of the workpiece for specific tasks.

2.3 Objectives

Finally, the classification can be focused on different types of objectives. It is important to underline that more than one objective can be selected by the following group, in that case it is possible to talk about Multi-objective optimization. The distinction is made between:

1. Minimization of the Number of Workstations (m): it is performed for a given output target and a fixed cycle time.

2. Minimization of the cycle time (c): the aim is to minimize the cycle time (or maximize the production rate) given a fixed number of workstations.

3. Maximization of the efficiency (E): in this case it is possible to have constraints on both the production rate and the number of workstations.

4. Minimization of the costs: given an output target the aim is to minimize the costs, considering the cost of the tasks, of the equipment as well as wage costs.

5. Maximization of the profit: where the profit is defined as the difference between the revenues on the production and the costs.

6. Smoothing of the station times: in this case the aim is to obtain similar workload in every station. It is possible to distinguish different types of smoothing:

   (a) Horizontal Balance: Smoothing of the difference station time caused by the different products in the mixed model.

   (b) Vertical Balance: Smoothing of the station times, creating a balance over all the station of the line

7. Minimization of the Incompletion Probability: the incompletion probability is the probability that a task is not completed, leaving the piece unfinished with the need of further working after the assembly cycle. It is minimized at every workstation, trying to achieve a specific target for the entire line.

8. Minimization or maximization of an indicator created for specific application: the indicator can be related to one or more characteristics of the line.

9. Feasibility: the research is focused on a feasible solution.

Creating an analogy with the classification of the SALBP provided by (Scholl and Becker, 2006b), it is possible to highlight four macro categories focusing more on the constraints than the objectives of the problem:

- GALBP-I: the problem is constrained in terms of Cycle Time (c).

- GALBP-II: The problem is constrained in terms of Number of Workstations (m).

- GALBP-E: considers ranges of feasible values for both the cycle time (c) and the number of workstations (m).

- GALBP-F: both the cycle time (c) and the number of workstations (m) are constrained.

2.4 Solution Methodologies

Looking at the possible solution methodology for the problem, it is possible to highlight two main macro categories: Heuristic procedures and analytical/numerical ones that look for an optimum.

Heuristic Procedures The needs for the development of Heuristic Procedures, come from the nature of the problem. The ALBP is a NP-hard problem. It means that the search for an optimum can require a lot of computational effort. The heuristic procedures does not aim to reach an optimum, but just to achieve an adequate solution in a small amount of time. Erel and Sarin (1998) developed a survey of the heuristic methodologies used to solve both deterministic and stochastic ALBPs. In this context, the analysis will be focused on the stochastic ones. They individuate three main types of heuristic procedures for the resolution of the stochastic ALBP:

- Modified version of Single-Model Deterministic (SMD) problem Procedures. Those procedures add to the SMD procedures a constraint in terms of reliability. From the SMD procedures it is possible to distinguish four main categories:
  - Single pass decision rules: they develop the solution by opening the stations one at the time, assigning the task to the open station according to a defined priority rule. When the task considered or no task cannot be assigned, the station considered is closed and another one is opened. This procedure is repeated until all the tasks are assigned. The heuristic procedure from Kottas and Lau (1973) is probably the most famous example of this kind of procedures.
  - Multiple single-pass decision rules: they start with the development of more than one feasible solution through a single-pass approach, selecting then the one resulting the best.
  - Backtracking procedures: they are mainly developed into two stages. In the first one or more feasible solutions are developed with one of the

previous methodologies. Then, they try to improve them through some iterative procedures. The most common are the simulated annealing algorithms. They are a random search procedure often used when the search space is discrete. This procedure allows to avoid the technique to get stuck in local minimum. Another one are the Evolutionary algorithms. They use mechanism inspired by biological evolution such as mutation and reproduction. A set of candidates solutions are generated and used as individuals of the population that is then treated according to the type of algorithm. It is possible to distinguish between genetic algorithms, where the evolution is inspired by Charles Darwin's natural selection theory and is performed by exchanging "genes" between the various solutions. Then it is possible to have an Ant Colony Optimization algorithm, that simulate the creation of pheromones path as ants do to direct each other to resources while exploring. Another type is the Imperial Competitive Algorithm, that simulates the evolution of a series of empires that fights against each other ending with just one winner. And finally we have the Particle Swarm Optimization, where the research is done using a "swarm" of solutions moving, avoiding the usage of a gradient.

- Simulation Techniques: Simulation allows to study the system building a model of it. This allows to perform experiments on it and to study different solutions. The deep analysis of the systems through simulation is nowadays possible thanks to the increase of the computational power available. The balancing procedures can use simulation to approach the problem from different perspectives.

- Single model stochastic procedures: those heuristic procedures were developed specifically for the stochastic problem. For this reason, they consider costs related to the incompletion of the tasks. The procedures in this category has as objective function, the minimization of the total costs, composed by the total labour cost and the total expected incompletion costs.

Optimum seeking procedures According to Hua et al. (2012, p. 57) an optimum seeking method can be defined as: "a method to find technological production processes that are best in some sense, while using as few experiments as possible. It is a scientific method for arranging experiments." The complexity of the problem and the computational effort required in order to apply the optimum seeking strategies are strictly dependent on the number of tasks. For the resolution of the ALBP three main categories of optimum-seeking procedure can be highlighted:

- Chance-Constrained Linear Programming: it is one of the most common way to approach a stochastic constrained optimization problem. It ensures that a certain condition is met with a certain probability. It allows to address large-scale complex optimization, under various uncertainties, developing a linear program with constraints on the probability of realization of specific uncertain parameters.
- Dynamic Programming: it is based on simplifying a complex problem by breaking it into simpler sub-problems in a recursive way. Storing the results of the sub-problems allows to reduce drastically the computational time.
- Branch and Bound techniques: usually used to solve combinatorial problems. It entails a systematic enumeration of all candidates solutions. The algorithm keeps the best solution found so far, while the large subset of partial solutions are fathomed to identify those ones who cannot improve the current best, by using upper and lower estimated bounds of the quantity being optimized, and remove them from the following development of the solutions. The solutions are usually represented by a rooted tree.

2.5 Literature Classification

The literature regarding the GALBP can be classified into macro categories according to some of the previously introduced dimensions of the problem. Sivasankaran and Shahabudeen (2014), defines eight type of Assembly Line Balancing Problems. Half of them are related to the deterministic problem, while the others to the stochastic one.

In particular, three dimensions of the problem are used for the classification of the literature:

- Single or multi-model (considering also the mixed model).
- Deterministic or stochastic task times.

- Straight line problem against U-shaped line.

This project is focused on the stochastic problem. For information regarding the deterministic problem, the reader is referred to Erel and Sarin (1998); Scholl and Becker (2006b). In the following, different literature categories will be analysed, following an order given by the objective of the procedure.

Single-model Stochastic straight-type problem

It is possible to distinguish different branches of the Single-model Stochastic Straight-type solution methodologies, looking at the objective function and at the resolution methodologies.

One of the most common objectives in the literature regarding the single model stochastic straight type line is the minimization of the number of workstation m.

Different articles face the problem through Dynamic Programming procedures. Kao (1976), Sniedovich (1981) and Carraway (1989) develop solution based on dynamic programming, looking for an optimum solution to the problem. Kao (1976), develops a procedure based on the one formulated by Held and Karp (1962) for the deterministic line. Hence, the optimal return value is the vector of the task assignments and the recursive method for finding it is based on Preference Ordering Dynamic Programming (Mitten, 1974). Sniedovich (1981) and Carraway (1989) propose a model stemming from a modification of that of Kao (1976), trying to overpass its sources of sub-optimality. Both of them develop methodologies that are not able to always find an optimal solution, but the former, for simple lines (reduced number of tasks) and with normally distributed task times, it is able to find an optimal or sub-optimal solution in most of the cases. While, the latter highlights the superiority in finding efficient solutions of one of the two algorithms developed. However, both of them also explain how to address the possibility of negative task time (due to the normal distribution).

It is possible to find papers focused on the development of some mathematical models. For example, Sphicas and Silverman (1976) and Nkasu and Leung (1995) develop two different mathematical models. The latter is a Computer-integrated manufacturing assembly system design (CISMAD) and it integrates the Monte Carlo simulation procedure with the COMSOAL-based assembly system design algorithm structure. It specifies the cycle times and the task times and a confidence interval for them, in order to assign the tasks to the smallest number of workstation possible, by generating all the feasible solutions and searching through them the "optimal" one. Also Ağpak and Gökçen (2007) develop a chance-constrained integer programming model for both the straight line and the U-type lines. Diefenbach and Stolletz (2020) proposed a Reliability-based Branch and Bound that directly works on the observations of the task times. Being the baseline algorithm of this thesis, it will be further explained in the following sections.

Different types of algorithms are developed, as well as heuristic procedures. The paper by Özcan (2010) analyses the two-sided problem.This type of problem is characterized by task performed in parallel on the two side of a station. It develop a chance constrained piecewise-linear mixed integer program (CPMIP) to model and solve the problem and also a solution approach with simulated annealing algorithm was proposed. Hu et al. (2008) develop an enumerative algorithm, defining a time transfer function and computing the earliest and the latest start time of a task, considering precedence constraints. Chiang and Urban (2006) develop heuristic methodologies, starting from the generation of a configuration, and then looking for an improvement. Thus, all of the aforementioned methodologies aim at the minimization of the number of workstations m, consider the incompletion probability (IP), guaranteeing that for the single workstation or for the entire line, it does not exceed a given threshold previously fixed.

Another objective well treated in the literature is the smoothing of workload between the various workstations. Roy et al. (2011) Roy and Khan (2011) proposed a stochastic programming-based algorithm aimed at the minimization of the Balancing losses and the variance of the idle times. This variance is used as a measure for system loss for the system and the algorithm is based on the assumption that "The stability of the total system is on maximum level when this variance is on minimum level" (Roy et al., 2011, p. 331).

Consequently, different heuristic procedures are proposed by Moodie and Young (1965); Reeve and Thomas (1973); Suresh and Sahu (1994); Suresh et al. (1996) . These are all procedures starting from a first assignment and building of an initial configuration. This configuration is then improved following the objective of maximizing a smoothing index, in order to distribute the workload among workstations.

The third important objective is the minimization of the cost function. The cost function is usually composed by two parts: the incompletion costs and the operating costs. The incompletion costs depend on the incompletion probability. They are evaluated considering the possible remedial policy to complete the unfinished product. Usually two possibilities arises, the first is that before considering the finishing of the uncompleted tasks, the procedures consider finishing as many tasks as possible on the line, taking into account the precedence constraints, while the second is that the entire line is stopped until the uncompleted task is solved. In the first case, once the unfinished product exit from the line, different policies are available. The piece can be completed in site but in overtime or through dedicated stations or outside through an outsourcing. Those possibilities bring to different costs. The operating costs, on the contrary, depends only on the number of workstations and on the cycle time.

Different heuristics approaches are developed to face this problem with both remedial policies. Shin and Min (1991) develop a heuristic algorithm aiming at the most cost-efficient solution for an assembly line in Just In Time (JIT) environments. They explain that in the JIT environment, the effectiveness of the ALB depends on the variability in task processing times, therefore it is more important to focus on the prevention of line variance rather than the optimal balance of the assembly lines. They adopt an operating policy similar to the mutual-relief movement approach. The line is run so that, when a failure occurs at any workstation, the whole line is stopped until the problem is solved. Lyu (1997), proposes a solution of the problem based on the single-run optimization method. In particular, it starts from the work of Suri and Leung (1989) that proposed a PARMSR (Perturbation-Analysis-Robbins-Monro-Single-Run) algorithm. Both Shin and Min (1991); Kottas and Lau (1973, 1976, 1981) follow the first type of remedial policy, while Lyu (1997); Carter and Silverman (1984) set the stoppage of the line every time the cycle time is exceeded. Sarin et al. (1999); Tsujimura et al. (1995); Shttjb (1984) find an initial configuration through different assignment methodologies. This configuration is then improved trough heuristic or optimum seeking approaches. Sarin et al. (1999) propose a heuristic enumeration methodology for the problem, where the remedial policy tends to complete as many tasks as possible on-line and the remaining offline. Shttjb (1984) follows the same remedial policy. Tsujimura et al. (1995) present a heuristic genetic algorithm, that "permit to code the solution in a chromosome structure and the repetition of simple operators for these chromosomes in the evolutionally process" (Tsujimura et al., 1995, p.543), for solving fuzzy ALBP. The use of genetic algorithm to solve the problem is then further developed and studied in Gen et al. (1996). Gökçen and Baykoç (1999), define a new remedial policy for the ALB problem. In their model, when an incompletion occurs, the incompleted unit is moved to a mobile station where the operations are completed. In order to avoid the idle time of the sequent station, this is fed by the buffer storage placed between the two stations. When the units off-line, on the mobile station, is finished, it is added to the mentioned buffer storage.

The solution proposed by Sarin and Erel (1990) is a cost function implemented in a dynamic programming scheme that is then implemented heuristically. They also propose a solution for larger problems (in terms of number of tasks), which divides it into sub-problems, that are then solved through the methodology exposed, bringing to some approximate solutions that are further improved through a branch and bound procedures. Boysen and Fliedner (2008) discuss an algorithm to maximize revenues and minimize the costs, decomposing the problem into two stages. In the first stage a valid sequence of task is built, according to precedence restrictions, while in the second stage the tasks are assigned to the workstation following a shortest path procedure.

Another objective looking for a solution to the problem can be the minimization of the cycle time c. The work from Hazır and Dolgui (2013) discuss a model aiming at the creation of systems that hedge against disruptions. They present two mathematical models developing also solution algorithms. The first model considers just the interval uncertainty. Whereas the second one considers the uncertainty as function of the number of operations in a station and is more complex compared to the first one.

In terms of solution, the literature consists of nearly exclusively of heuristic procedures. Gu et al. (2006) develop an Electromagnetism like algorithm. It is a generation-based optimization heuristic for global optimum optimization problems. It starts sampling random points in a feasible region and use a mechanism similar to the attraction-repulsion one from the electromagnetism theory, to move a population of points toward optimality. Gen et al. (1996) propose a genetic algorithm to solve the problem, with the objective to minimize the total operation time in each workstation. Liu et al. (2005) present a bidirectional assignment heuristic algorithm composed by a first stage of assignment of the task to the workstations, from both forward and backward directions. In the second stage a trade and transfer procedure is used in order to balance the workload among workstations.

The other objective for the problem can be the maximization of the efficiency E.

This is the least treated problem, probably because of its non-linear objective function Zacharia and Nearchou (2013, p. 3034). Zacharia and Nearchou (2013) propose a new modified genetic algorithm for the solution of the problem.

Lastly, different studies focus on more than one objective simultaneously. Cakir et al. (2011) define a multi-objective optimization for the problem. Their model considers the possibility of parallel workstations and focuses at first the distribution of the workload among stations evenly, that is the equivalent of the minimization of a smoothness index. Being possible the paralleling of the stations, the second objective is the minimization of the costs related to labour and equipment requirements. In fact, the parallelization of a workstation, means the duplication of all the equipment and personnel.

Dong et al. (2018) define bi-criteria stochastic assembly line balancing that has the objective to simultaneously minimize the cycle time and the equipment costs, therefore the operating costs, through the use of a Particle Swarm Optimization (PSO) algorithm. Gamberini et al. (2006, 2007) present new heuristic methods, focusing on the rebalancing of existing lines. The objective of this methods is not only the minimization of the cost related to the assembly process, but also the costs related to the rebalancing itself, trying to minimize the task reassignment required during the reconfiguration of the line. Zacharia and Nearchou (2012) propose a method where the processing times are modelled as triangular functions and the cost function is formulated as the weighted sum of multiple objectives that are the minimization of both the cycle time and the smoothness index of the workload of the line. While, Hamta et al. (2013) formulate a hybrid meta-heuristic approach including Particle swarm optimization and Variable neighbourhood search. The objective function in this case has three components: minimization of the cycle time, of total equipment cost and of the smoothness index. Krishnan et al. (2016) propose a model that use as objective function, the minimization of the difference in the Risk Index (RI) between workstation, given precedence and cycle time constraints. The RI is evaluated as the product of three components for each task, the delay index, the contribution ratio and the criticality ratio, that represents respectively the number of data points exceeding the cycle time, the contribution of each task to its total cycle time and the magnitude of deviation away from the standard task time.

Objective	**Paper**	**Methodology**
	Kao (1976)	dynamic programming

Minimization m

	Sniedovich (1981)	dynamic programming
	Carraway (1989)	dynamic programming
	Sphicas and Silverman (1976)	mathematical model
	Nkasu and Leung (1995)	COMSOAL
	Ağpak and Gökçen (2007)	chance contrained integer programming
	Özcan (2010)	CPMIP
	Hu et al. (2008)	Enumerative algorithm
	Wu et al. (2008)	Branch and Bound
	Chiang and Urban (2006)	Heuristic
	Diefenbach and Stolletz (2020)	Chance-constrained Branch and Bound
Smoothing Procedure	Moodie and Young (1965)	Heuristic
	Reeve and Thomas (1973)	Heuristic
	Suresh and Sahu (1994)	Heuristic
	Suresh et al. (1996)	Heuristic
	Roy et al. (2011)	stochastic programming-based algorithm
Minimization of costs	Kottas and Lau (1973)	Heuristic algorithm
	Kottas and Lau (1976)	Heuristic algorithm
	Kottas and Lau (1981)	Heuristic algorithm
	Carter and Silverman (1984)	Heuristic algorithm
	Shin and Min (1991)	Heuristic algorithm
	Lyu (1997)	Single run optimization method
	Sarin et al. (1999)	Heuristic enumeration methodology
	Tsujimura et al. (1995)	Heuristic Genetic algorithm
	Shttjb (1984)	Heuristic algorithm
	Henig (1986)	Dynamic Programming
	Wilson (1986)	Dynamic Programming
	Sarin and Erel (1990)	Dynamic Programming
	Boysen and Fliedner (2008)	Two stage dynamic programming
Minimization of c	Gu et al. (2006)	Electromagnetism like algorithm
	Gen et al. (1996)	genetic algorithm
	Hazır and Dolgui (2013)	mathematical models
	Liu et al. (2005)	bidirectional assignment heuristic algorithm
Maximization Efficiency	Zacharia and Nearchou (2013)	genetic algorithm
Multi-objective	Cakir et al. (2011)	multi-objective optimization
	Dong et al. (2018)	Particle Swarm Optimization
	Gamberini et al. (2006)	heuristic methods
	Gamberini et al. (2007)	heuristic methods
	Zacharia and Nearchou (2012)	heuristic methods
	Hamta et al. (2013)	hybrid meta-heuristic

	Krishnan et al. (2016)	mathematical model

Table 2.1: Summary Literature Single-model Stochastich Straight-type Line

Multi-model Stochastic straight-type problem

Under this category, fall both the multi-model and the mixed-model assembly lines. The literature in this field is more limited with respect to the previous case, since the majority of the publications about mixed and multi-model Assembly line balancing address the deterministic case of the problem. All the methodologies

Only one article is found with the aim of minimizing the number of workstations. Van Hop (2006) proposes a methodology to solve the mixed-model assembly line balancing problem with a fuzzy heuristic improvement with this aim.

No specific model is found for what concern the usage of smoothness indexes, but they are utilized in multi-objectives models.

With the objective of minimizing the total costs, Vrat and Virani (1976) develops a heuristic algorithm for the mixed model assembly line. The procedures allows the paralleling of the workstations.

For the minimization of the cycle time a comment similar to the smoothness indexes can be done.

Finally, there are a series of methodologies characterized by multiple-objectives. McMullen et al. (1997) present a heuristic methodology that permits task paralleling to occur. In their paper, they evaluate different task selection rule through the use of four performance indicators: Work-In-Progress, Inventory level, Flow time, System Throughput, Unit labour cost, System Utilization and Percentage of Unit Completed within the specified cycle time in each workstation. Furthermore, they measure the ability of the layout to achieve the desired cycle time. They reduce the mixed-model into a single model through the usage of composite task times and then assign the tasks according to seven different rules, five of which are specifically developed. Morover, McMullen et al. (1997) develop a Simulated annealing process. Also in this case, the solution allows the paralleling of the stations to occurs. Six different objective function are investigated. The first three are the minimization of the design costs, the minimization of the smoothness index and the minimization of the probability of lateness. The other three objective functions are composite of the previous ones. The paper investigates the difficulty of the multiple-objectives model, due to possible contrasts that can arise between different ones. McMullen and Tarasewich (2003) propose an Ant Technique to solve the problem, making experiment with four different objective functions. The objectives tested were the maximization of the Utilization of the assembly line, of the completion probability and of a composite function between the previous two, or the minimization of the design costs associated to the layout. McMullen and Tarasewich (2006) investigate the problem building an objective function that is "a linear combination of multiple objectives associated with the line-balancing problem, such as required crew size, system utilization, the probability of jobs being assembled within a certain time frame, and system design cost".

Liu et al. (2019) propose an integrated optimization based on the evaluation of three types of time complexity on the mixed-model assembly line. The aim of their model is to maximize the productivity, to minimize the complexity and also to solve the buffer allocation problem minimizing the total buffer capacity of the system. Mendes et al. (2005) developed an hybrid simulation and analytical model to solve the mixed-model ALBP. With a simulated annealing meta-heuristic approach, they derived different line configurations with the aim of minimizing the number of workstations and to smooth the workload balance between the stations. The various solutions are generated according to the possible different market share of the products. Those solution are then investigated and fine-tuned through a discrete event simulation model. Samouei et al. (2017) explored a mixed-model assembly line with the objective to simultaneously maximize the system reliability and the weighted line efficiency (therefore minimizing the number of workstations), while minimizing the smoothness index. The procedure adopted is a Simulated annealing algorithm. Whereas, Xu and Xiao (2008) deal with a special case of mixed-model assembly line where the station lengths are longer than the distance conveyor moved within one cycle time. They proposed a fuzzy α model, solved through the use of a genetic algorithm, aimed at the minimization of the total overtime workload.

Objective	Paper	Methodology
Minimization of m	Van Hop (2006)	heuristic improvement
Smoothing Procedure	McMullen et al. (1997)	Simulated annealing process
Minimization of Costs	Vrat and Virani (1976) ? McMullen and Tarasewich (2003)	heuristic algorithm Simulated annealing process Ant Techinque
Multi-objective	McMullen et al. (1997) McMullen and Tarasewich (2003) McMullen and Tarasewich (2006) Liu et al. (2019) Mendes et al. (2005) Samouei et al. (2017)	Simulated annealing process Ant Techinque Ant Techinque integrated optimization hybrid simulation and analytical model Simulated annealing
Minimization of Lateness Probability	McMullen et al. (1997)	Simulated annealing process
Maximization of Utilization	McMullen and Tarasewich (2003)	Ant Techinque
Maximization of completion probability	McMullen and Tarasewich (2003)	Ant Techinque
Minimization of the total overtime workload	Xu and Xiao (2008)	Genetic algorithm

Table 2.2: Summary Literature Review Multi-model Stochastic Straight-type Line

Single-model Stochastic U-type Problem

The U-Type problem can leverage on a much smaller literature with respect to the straight type problem. that is the reason why in this context no division will be made between single and the multi-model.

Urban and Chiang (2006) propose an optimal solution methodology to address the problem. The problem is modelled as a linear integer program, with the objective to minimize the number of workstations. With the same aim in Chiang and Urban (2006) a hybrid heuristic is presented, generating an initial feasible solution and then improving it. Baykasoglu and Özbakir (2007) introduce a novel algorithm that includes the COMSOAL methodology and the Genetic Algorithm having the same objective. Zhuo et al. (2011) Zuho et al (2011) develop a genetic algorithm as well. Aydogan et al. (2019) explain a new novel Particle Swarm Optimization algorithm. Guerriero and Miltenburg (2003) present a recoursive algorithm for finding the optimal solution when task completion times have any distribution function. They provide also an equivalent shortest path method.

No particular model and solution methodology is found having the main objective the smoothing of the workload.

Erel et al. (2005) develop a beam search methodology to solve the problem with the aim of minimizing the expected costs, while Boysen and Fliedner (2008) provide an algorithm with the same aim, that decomposes the original problem into two stages solvable independently.

It is not possible to find literature focusing on the minimization of the cycle time.

For what concern the multi-objectives models, Delice et al. (2016) explain a methodology with the primary aim of minimizing the number of work position and as secondary one the minimization of the number of workstation. Agpak and Gökcen (2007) discuss a model with the same objective function developing a chance constrained integer programming model to take into account also the minimum reliability required. Bagher et al. (2011) introduce a new evolutionary algorithm, ICA, with the aim of minimizing the number of work stations, as well as idle time in each station and non-completion probabilities of each station. Finally, Dong et al. (2014) investigate a method to evaluate the expected Work overload time and trys to minimize it through the use of a simulated annealing algorithm.

Objective	**Paper**	**Methodology**
Minimization of m	Urban and Chiang (2006)	linear integer program
	Chiang and Urban (2006)	hybrid heuristic algorithm
	Baykasoglu and Özbakir (2007)	COMSOAL methodology + Genetic Algorithm
	Zhuo et al. (2011)	genetic algorithm
	Aydogan et al. (2019)	Particle Swarm Optimization
	Guerriero and Miltenburg (2003)	recoursive algorithm
Minimization of costs	Erel et al. (2005)	beam search methodology
	Boysen and Fliedner (2008)	two stage solving algorithm

Multi-objective	Delice et al. (2016)	mathematical model
	Agpak and Gökcen (2007)	chance constrained integer programming
	Bagher et al. (2011)	evolutionary algorithm, ICA
	Dong et al. (2014)	simulated annealing algorithm

Table 2.3: Summary Literature Review Single-model Stochastic U-type

Multi-Model Stochastic U-Type problem

The literature regarding this type of problem is really limited. Only two papers are found about this specific problem.

The first one, by Özcan (2010) formulate a genetic algorithm in order to minimize the cycle time, given a specific number of workstations.

The second one, by Agrawal and Tiwari (2008) present a collaborative ant colony optimization algorithm. The distinctive characteristic of this article is that it is based on the disassembly process instead of the assembly one.

Objective	**Paper**	**Methodology**
Minimization of m	Özcan (2010)	Genetic Algorithm
Minimization of c	Agrawal and Tiwari (2008)	Ant-colony optimization

Table 2.4: Summary Literature Review Multi-model Stochastic U-type

Chapter 3

Theoretical Background

3.1 Introduction to Bootstrap

The bootstrap methodology is a resampling with replacement method. Since it is based on resampling, the results of the application of the Bootstrap are random variable. In order to clearly understand the resampling and the bootstrap, the concept of *Empirical Distribution* must be introduced. Given the sample $\boldsymbol{x}$ from the distribution F

$$F \rightarrow (x_1, x_2, \ldots, x_n) = \boldsymbol{x} \tag{3.1}$$

The *Empirical Distribution Function* $\hat{F}$ is defined as the discrete distribution that puts probability $1/n$ on each value x_i

This is basically a representation in frequency of the available sample. It is important to notice that there is no loss of information moving between this type of representation and the full data set. The empirical distribution and the sample set are used in order to estimate a parameter, that is used as estimation of the real distribution. This process is called *Plug-in Principle*

$$\hat{\theta} = t(\hat{F}) \rightarrow \theta = t(F) \tag{3.2}$$

The Bootstrap methodology is introduced by Efron (1979) as a computer-based methodology to estimate the standard error of the estimate $\hat{\theta} = s(x)$ of a parameter of interest (of a distribution) from a sample, therefore to understand how good the plug-in principle is.

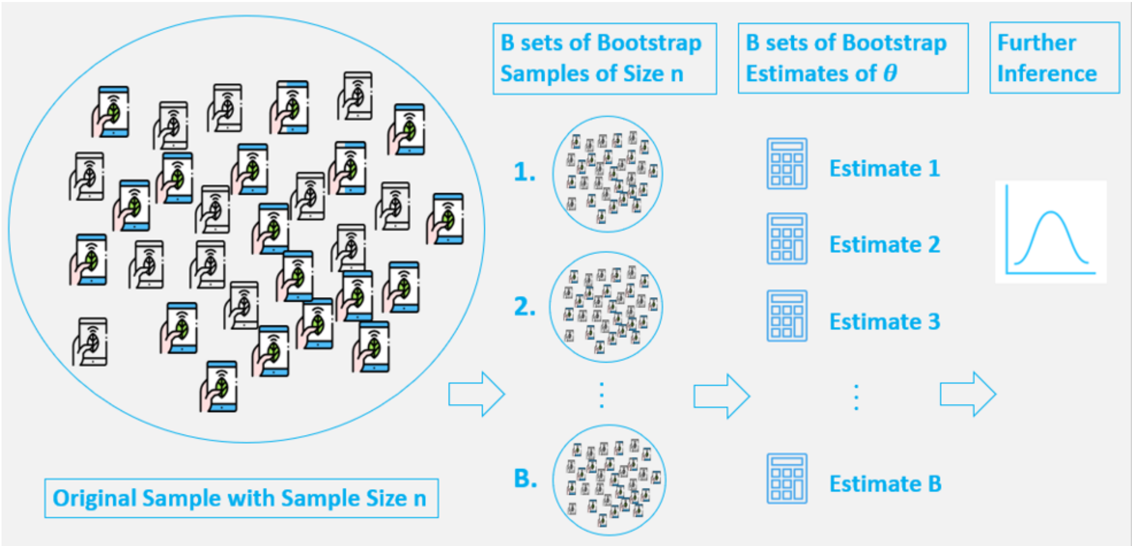


Figure 3.1: Application of the Bootstrap Methodology
Source: Lorna (2019)

3.1.1 Process

To perform the bootstrap on a set of data, the two main elements of the methodology must be introduced. Furthermore, it is important to specify that from now on, in this section, all the elements related to the bootstrap will be characterized by a star.

The first element to introduce is the *Bootstrap Sample*. Given a data set $\boldsymbol{x}$ of size n, a bootstrap sample is generated by resampling with substitution (so sampling from the empirical distribution $\hat{F}$) n elements from the available set.

$$\hat{F} \to (x_1^*, x_2^*, \dots, x_n^*) = \boldsymbol{x}^* \tag{3.3}$$

It is possible to define the bootstrap sample $\boldsymbol{x}^*$ as the resampled version of $\boldsymbol{x}$

To each bootstrap sample corresponds a *Bootstrap Replication* of $\hat{\theta}$

$$\hat{\theta}^* = s(\boldsymbol{x}^*) \tag{3.4}$$

Basically, starting from the original data, it is possible to create B bootstrap samples of the same size of the original data. From each of those bootstrap sample it is possible to evaluate the respective bootstrap estimate, obtaining a set of B estimates. To this set it is possible to apply further inference. (Figure 3.1)

The main advantages of the methodology are:

- Requires no theoretical calculation.
- Can be performed, no matter how complicated the estimator may be.
- It is easy to implement the bootstrap algorithm on a computer.

It is important to highlight also the assumptions made applying the methodology. The first assumption is related to the size of the number of samples. The method assumes that the original data sample is large enough so that the resampling distribution $\hat{F}$ approaches the true distribution F of the random variable of interest. To understand the meaning of large enough, Mooney et al. (1993) states that the quality of the approximation is in general satisfactory for sample of size $30 - 50$.
The second assumption is related to the number of the bootstrap estimates, saying that it is large enough, so that the bootstrapped estimate of the sampling distribution approximate the true distribution of the statistic. Again, the definition "large enough" is quite vague, but we have empirical values provided by Efron and Tibshirani (1994) stating that for standard error estimates values $B = 50-200$ are enough, while much larger samples are required for the definition of confidence intervals. In addition, they explain how having $B > 1000$ produces a slight improvement of the bootstrap estimate of the sampling distribution, that is not worth the additional computational effort required.

3.2 Baseline Reliability-based Branch and Bound

Diefenbach and Stolletz (2020) propose a Reliability-based Branch and Bound algorithm (RB&B). Every information in this section is taken from Diefenbach and Stolletz (2020). The aim of the algorithm is the minimization of the number of stations m and it follows the following Chance-constrained mathematical model:

$$\min \sum_{m=1}^{M} Z_m \tag{3.5}$$

$$s.t. \sum_{i=1}^{I} t_{n,i} \cdot X_{i,m} \leq c \cdot Z_m + (1 - B_n) \cdot BigM \qquad m = 1, \ldots, M; n = 1, \ldots, N \tag{3.6}$$

$$\frac{\sum_{n=1}^{N} B_n}{N} \geq R \tag{3.7}$$

$$\sum_{m=1}^{M} X_{i,m} = 1 \qquad i = 1, \ldots, I \tag{3.8}$$

$$\sum_{m=1}^{M} m \cdot X_{i,m} \leq \sum_{m=1}^{M} m \cdot X_{j,m} \qquad \forall (i,j) \in P \tag{3.9}$$

$$X_{i,m} \in \{0,1\} \qquad i = 1, \ldots, I; \quad m = 1, \ldots, M \tag{3.10}$$

$$Z_m \in \{0,1\} \qquad m = 1, \ldots, M \tag{3.11}$$

$$B_n \in \{0,1\} \qquad n = 1, \ldots, N \tag{3.12}$$

The chance-constrained model proposed by Diefenbach and Stolletz (2020) uses three types of binary variables:

- $X_{i,m}$: It is the assignment-variable. It is equal to 1 if task i is assigned to station m and 0 otherwise.
- Z_m: It is the station-variable. It is equal to 1 if station m is opened, 0 otherwise.
- B_n: It is the sample-variable. It is equal to 0 if the cycle time c is exceeded for sample n at least at one station.

The objective function (3.5) minimize the number of opened stations. The constraint (3.6) secure that the sum of the times of all the tasks assigned to a station m does not exceed the cycle time c, if the Sample-variable B_n is equal to 1. It means that if the cycle time is exceeded at least at one station, the sample-variable B_n has to be equal to 0. The constraint (3.7) assure that at least a fraction R of samples do not exceed the cycle time. The constraint (3.8) assure that each task is assigned to exactly one workstation. The constraint (3.9) assure that the precedence relations are respected. The constraints (3.10) - (3.12) defines the variables domains.

The procedure starts from the transformation of the Lower bounds used for the deterministic model into lower bounds for the stochastic one. The transformation proposed works for any bound already developed for the deterministic problem but also for any bound that could be developed in future.

The transformation follows the following theorem:

Theorem 1 *Let the samples be ordered such that the lower bounds on the number of stations for the deterministic problem for a single sample are increasing, i.e.* $LB(n) \leq LB(n+1)$ *holds. Let* $LB(\bar{n})$ *be the lower bound of sample* $\bar{n}$ *with* $\bar{n} := \lceil R \cdot N \rceil$*. Then,* $LB(\bar{n})$ *is a lower bound on the number of stations for the entire chance-constrained model with respect to samples* $n \in \{1, \ldots, N\}$ *and the line reliability* R*.*

The transformation requires the evaluation of the lower bound for each of the N samples. In particular, seven different lower bounds are used (the ones introduced by Becker and Scholl (2006)) and the maximum between them is used as the initial Global Lower Bound.

The RB&B open new nodes each time a new station is opened. Every node corresponds to a partial assignment of the tasks to a station. Not every tasks must be assigned in every node. The strategy used is bidirectional, so that the assignment is tested simultaneously forward and backward. The two starting independent nodes can lead to nodes that belong to both trees. The consideration of the partial assignment is necessary to respect the reliability constraints and allows to distinguish between R-maximal station load, if none of the unassigned task can be assigned without exceeding the reliability, and non R-maximal. The inclusion of both R-maximal and non R-maximal nodes, brings to a significant increase of the branching tree dimensions. In order to contain the "growing" of the tree, three different fathoming strategies, to identify the nodes that are dominated, are used:

- **Local Lower Bounds:** Local lower bounds are calculated on each node, with the same procedure used to evaluate the global lower bound, considering just the non assigned tasks in each node. The node k can be fathomed if the sum of the already open stations and the local lower bound is equal or bigger than the current upper bound.

$$\max\{LLB_k^1, \ldots, LLB_k^7\} + \text{used station of node } k \geq UB$$

- **Dominance Rule:** This rule performs a check on the non R-maximal nodes. This category of nodes can be fathomed if it cannot lead to an improvement in terms of reliability. This means that it can be fathomed if it exist a different node that has at least the same task assigned, that requires the same or less number of stations and has the same or higher reliability.

- **Logical Test:** It checks that the branch following a node can lead to a feasible solution. This is done, by evaluating the number of samples for which the cycle time is exceeded for any assignment in node k. "The node can be fathomed if the line reliability of node k minus the percentage of additionally guaranteed incomplete samples is less than R." (Diefenbach and Stolletz, 2020, p.14)

The application of the algorithm is limited by the availability of data regarding the task times. This limitation will be further investigated in the following sections.

3.3 Variability of the Algorithm Outcome

The RB&B developed in Diefenbach and Stolletz (2020) was tested on a series of instances. The problem instances were based on the benchmark set of Scholl (1993), considering instances up to 21 tasks. These ones were proposed as deterministic instances, that were made stochastic through the creation of some probability distribution functions, starting from the deterministic times of Scholl. The settings considered in this work are part of those included in the study of Diefenbach and Stolletz (2020) and any difference will be specified in the relative section. To generate the data of the stochastic lines, the task times are sampled from a Normal and a Gamma distribution. Since in literature coefficient of variations up to 0.5 are considered (Liu et al., 2005), three different values are used for the coefficient of variation $CV \in \{0.1, 0.3, 0.5\}$. The coefficient of variation is used to evaluate the parameters of the two distributions. For the Normal distribution, the location parameter is the mean of the task times, while the scale parameter is represented by $CV \cdot \bar{t_i}$, where $\bar{t_i}$ is the mean of the task times for the task i. For the Gamma distribution, the shape parameter is represented by $k = \frac{1}{CV^2}$ while the scale parameter as $\theta = \bar{t_i} \cdot CV^2$

For each model, different cycle times are tested. All the instances can be seen in Table 3.1.

Model	**CV**	**Cycle Time**
Mertens	0.1	$7, 8, 10, 15$
	0.3	10.15.18
	0.5	$15, 18$
Jaeschke	0.1	$7, 8, 10, 18$
	0.3	$10, 18$
	0.5	18

Jackson	0.1	$9, 10, 13, 14, 21$
	0.3	$10, 14, 21$
	0.5	$14, 21$

Table 3.1: Summary Instances Considered

The study of the influencing parameters is developed into three main steps:

- **Single Run:** One single run with different values of N, has been done for each setting.
- **Independent Replications:** Different independent runs of the algorithm are performed for each experimental settings. This is done for two different values of N.
  - $N = 10000$: 10 Independent replications performed.
  - $N = 100$: 1000 Independent replications performed.
- **Bootstrap Methodology:** The results of the algorithm are investigated through a Bootstrap Inspired approach.

3.3.1 Single Run

In order to evaluate the effects of N on the results of the algorithm, every instance is tested, performing a single run of the algorithm for seven different values of N. The values tested are $N = \{1, 10, 50, 100, 1000, 5000, 10000\}$. The first value is an extreme value for which the problem becomes deterministic, with task times randomly generated. The experiments are performed using randomly generated task times.

It is important to remember that the algorithm is a Chance-constrained Branch and Bound, therefore a value for the reliability must be specified. In this case $R = 95\%$ is used. Since, especially for the smaller values of N, there is the chance that the randomly generated task times cannot respect this reliability threshold, the algorithm is modified in order to avoid the stoppage in this situation. In these cases, the reliability threshold is lowered to the maximum threshold possible with the available data.

These runs show a variability between the results depending on the number of samples as can be seen in the example of Figure 3.2. The idea is that the variability of the results has a relation also with other characteristics of the instance evaluated. This kind of preliminary observations define the necessity of a deep investigation. To highlight proper relationship between parameters and assignments and number of stations to open, the running of independent replications is needed.

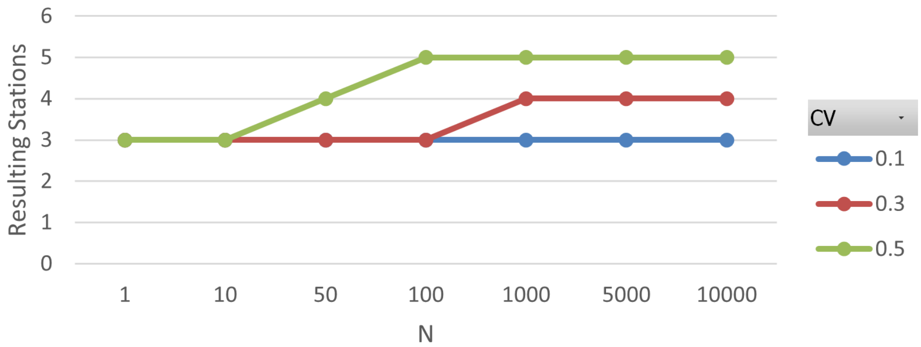


Figure 3.2: Example of Jackson model, Gamma distributed data with $c = 21$ for different CV values.

3.3.2 Independent Replications

The previous experiments give a first highlight on a variability of the results that must be deeply investigated. With this scope, different independent replications are performed. Only two line configurations are tested: Mertens (7 tasks) and Jackson (11 Tasks). Two values of N are used. 10 independent replication are performed for $N = 10'000$, while 1000 independent replications are performed for $N = 100$.

The results in terms of number of different minimum number of workstations to open and assignments found are reported in Table 3.2 for the configurations generated with the Gamma distribution. The same data from the configurations obtained with the Normal Distribution are reported in A.1.

Line configuration	CV	c	**Different Outcomes** $N = 10'000$	**Different Assignments** $N = 10'000$	**Different Outcomes** $N = 100$	**Different Assignments** $N = 100$
Mertens	0.1	7	1	1	2	7
		8	1	1	2	2
		10	1	1	2	13
		15	1	1	1	2
		18	1	1	1	2
	0.3	10	1	1	3	11
		15	1	1	2	17
		18	1	1	1	7
	0.5	15	1	1	4	79
		18	1	1	2	61
Jackson	0.1	9	1	1	2	16
		10	1	1	2	28
		13	1	1	1	10
		14	1	1	1	11
		21	1	1	1	4
	0.3	13	1	1	3	191
		14	1	1	3	73
		21	1	1	2	58
	0.5	14	2	5	2	243
		21	1	1	3	279

Table 3.2: Summary Independent Replication Experiments

The experiments' results for all the instances tested with $N = 10'000$ provide the same number of stations opened. The only exception to this result is:

$$\begin{aligned} Model &= Jackson \\ Distribution &= Gamma \\ CV &= 0.5 \\ c &= 14 \end{aligned}$$

It is possible to state that the results obtained for $N = 10'000$ can be considered as the correct results from now on. For the aforementioned exception the independent replications are performed again with $N = 20'000$. It results in a number of opened stations equal to 9 and just one task assignment.

Looking at the results it is possible to highlight the main trend of the influencing parameters. Two significant behaviour can be highlighted.

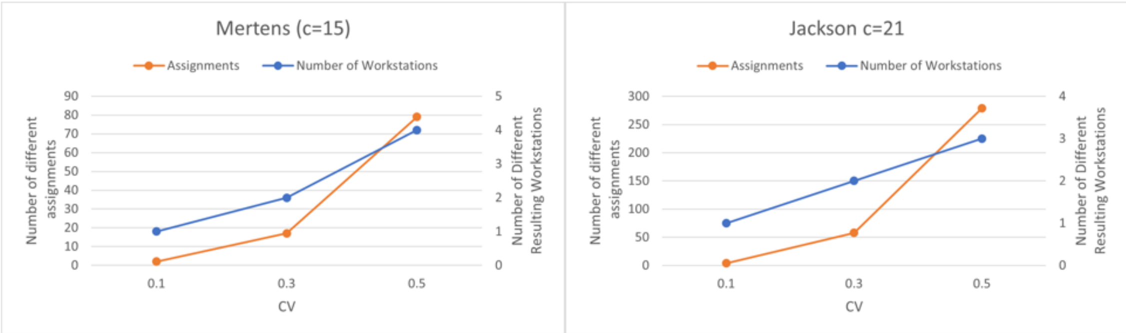


Figure 3.3: Effect of CV on the different Number of Assignments and resulting workstations among the replications performed with $N = 100$ for both Mertens and Jackson configurations created with a Gamma distribution.

First of all, the uncertainty given by a limited number of task times samples is strictly related to the coefficient of variation (CV) of the task times. For lower coefficient of variation, the achieved number of stations is always the same and it is the same given by the replications done with $N = 10'000$. Increasing the coefficient of variation, the number of solutions often increases for both the configurations analysed. This effect is amplified for the number of assignments found. (Figure 3.3)

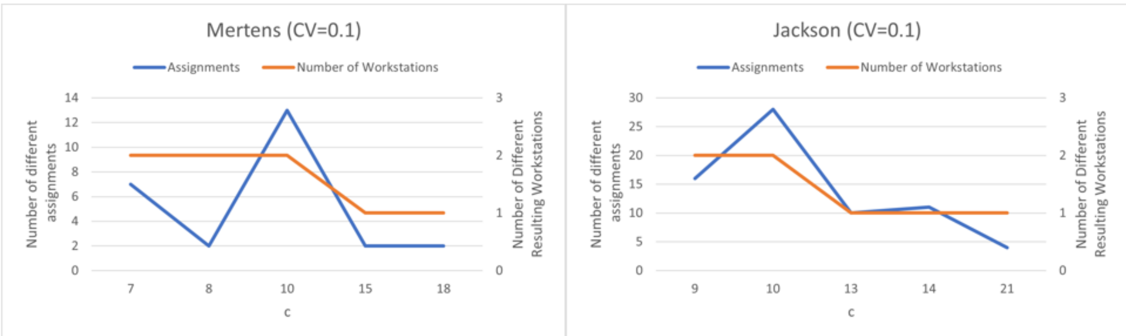


Figure 3.4: Effect of c on the different Number of Assignments and resulting workstations among the replications performed with $N = 100$ for both Mertens and Jackson configurations created with a Gamma distribution.

The second main behaviour highlighted is the relation with the cycle time. It is observed how for higher cycle time the solution is more robust, achieving lower

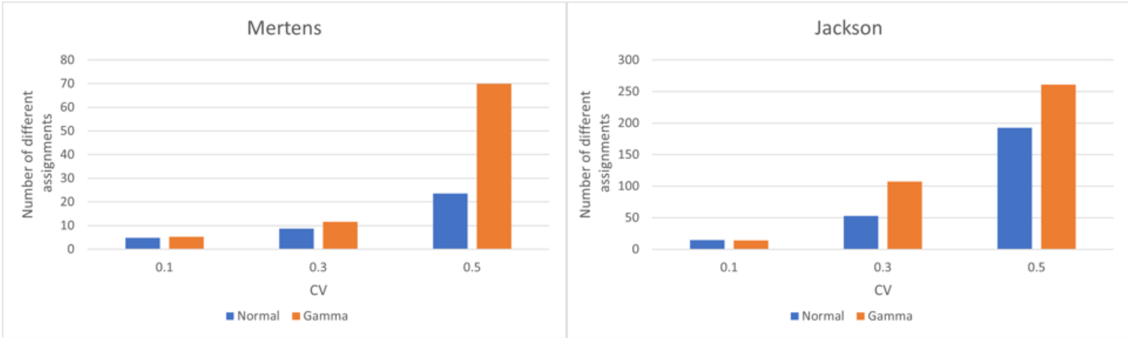


Figure 3.5: Number of different Assignments found for different values of CV and the two distributions

number of different outcomes. This was an expected result, because a larger time bucket, for each station, to fit the tasks, results in more flexibility in the assignment. Due to this flexibility, it can be observed how for the number of assignments it is not possible to observe the same trend. The number of resulting assignments between the replication is extremely variable and no particular trend can be identified.

Finally, it is important to notice also the effect of the distribution. With the Normal distribution all the times sampled below the zero were transformed into a 0. This seems to have an effect on the results, bringing in general to a lower number of stations. To explain this phenomenon, it is possible to think about the tail of a Gamma distribution, that is higher with respect to the right tail of a Normal one. This led to a more frequent sampling of higher task times, bringing to a lower reliability with the same number of stations. Therefore, it is possible to say that also the distribution have an impact on the results.

Furthermore, both the distribution and the coefficient of variation have a relation with the number of assignments achieved. In Figure 3.5 it is possible to observe that more assignments are observed genereting from a Gamma distribution and with higher values for the CV.

3.3.3 Parameters influencing the algorithm's results

Looking at the outcomes of all the tests performed so far, it is possible to see which are and with which trend some parameters influence the outcome of the algorithm:

1. **Number of Samples:** In general, it is possible to observe that the higher the number of sample available, the more reliable will be the result.

2. **Cycle Time:** For higher values of the cycle time, the results are more trustful. This can be explained due to the idle time. The larger the cycle time the higher the idle time available assigning one task to a station. It means that the algorithm has more "freedom" in the assignment. This allows to generate a robust solution. On the other side it is important to notice that not always the smaller the cycle time the less reliable the result. For smaller values of the cycle time the assignment is much more constrained, this can lead to a much more "guided" assignment. Therefore, it is possible to say that the relation with the cycle time is not linear. Even if, in general, the higher the cycle time the better, there are some specific values that lead to solutions that are extremely unreliable. This behaviour occurs in the cases that are somehow "borderline". Those are the values of the cycle time for which the result is in the proximity of the change, making difficult to distinguish the correct solution between two different values.

3. **Coefficient of Variation:** Being a proxy of the variability of the data, the bigger the coefficient of variation the more the results will be affected by a variability. Therefore, it is possible to say the the lower the coefficient of variation the better.

4. **Shape of the distribution:** The used distribution has an impact on the results. The output using the data generated from a Gamma distribution are always less reliable than the ones generated from a Normal distribution. In order to take into account the distribution and its shape the main parameters that can be considered are the Skewness or the Kurtosis. Further investigation will be performed in the following sections.

5. **Model:** The model has an impact on the results. The relationship is not completely clear and must be further investigated. In fact, the complexity of the model cannot be measured just in terms of number of tasks. Also the constraints must be considered. One parameter that can be used is the *Order Strength* (OS).

Chapter 4

Solution Methodology

4.1 Bootstrap application

The algorithm from Diefenbach and Stolletz (2020) is included in a bootstrap inspired procedure to deeply investigate the relations between the influencing parameters and to analyse the capability of reaching the correct solution. The procedure is tested among two line layouts between the ones analysed by Diefenbach and Stolletz (2020): Jackson line configuration with 11 tasks and Mertens configuration with 7 tasks. All the settings of the two layouts, presented in Table 3.1, are taken into account. The models, introduced by Scholl (1993), are characterized by deterministic task times. They are made stochastic through the same procedure seen in Section 3.2. The idea, inspired by the bootstrap methodology, is to re-sample with replacement the task times sets (of size N) available, generating B independent new task times sets of size N. The algorithm is then fed with those time sets and the results analysed.

$$\begin{bmatrix} t_{1,1} & \cdots & t_{I,1} \\ \vdots & \ddots & \vdots \\ t_{1,N} & \cdots & t_{I,N} \end{bmatrix}$$

Figure 4.1: Explanation of the applied resampling.

Considering $t_{n,i}$ as the $n-th$ time sample of the $i-th$ task, it is possible to represent all the task times observations as in Figure 4.1. The re-sampling is applied on the task times sequence, meaning that the single re-sampled observation is a row of the matrix $\{t_{i,n}, ..., t_{i,N}\}$.

Each bootstrap sample is used as input for the algorithm. It means that, in this application, the bootstrap estimates will be the results of the RB&B. In Figure 4.2 all the steps of the bootstrap methodology are depicted: the sampling with replacement from the original dataset (generation of Bootstrap Samples Set 1, Set 2, ...); Application of the RB&B to each Bootstrap Sample (Generation of the Bootstrap Estimate Res 1, Res 2, ...); Identification of the best solution.

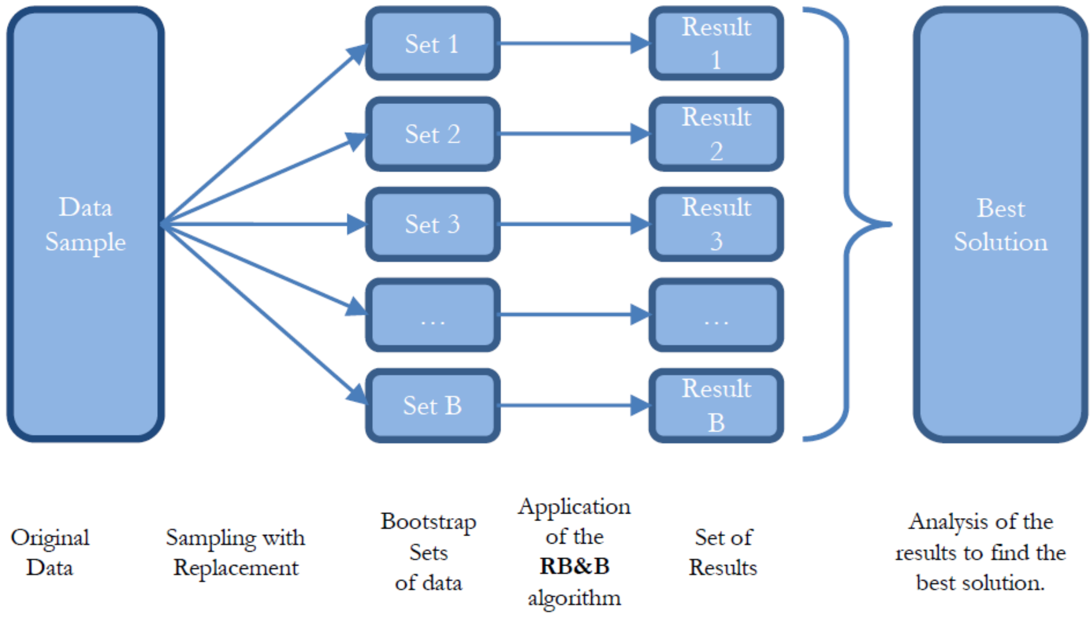


Figure 4.2: Application of the bootstrap methodology

4.1.1 Test of the Bootstrap Methodology

The first experiments are performed for three configurations of the line, respectively with 7 tasks (Mertens), 9 tasks (Jaeschke) and 11 tasks (Jackson). The original data-sets are generated using both a Normal and a Gamma distribution. The procedure was applied with $N = \{100, 500, 1000\}$ and setting $B = 1000$ [1]. The results of the application of the procedure to data-sets generated with a Gamma distribution are reported in Table 4.1. The ones with a Normal one are reported in the Appendix A.1.1. Both the tables report the number of times, the outcome from the

[1]The reason behind this choice is related to the fact that in literature it is the highest suggested values as explained in Section 3.1.1

RB&B applied to 10′000 observation data-set appears between the 1000 bootstrap replications.

The first analysis carried out, points out how the most frequent solution in terms of number of workstations is not always the same of the RB&B applied to 10′000 data. The conclusion is that, in most of the cases the bootstrap cannot solve the problem by simply re-sampling the data available generating possible scenarios.

Config.	**CV**	c	$N = 100$	$N = 500$	$N = 1000$
Mertens	0.1	7	1000	1000	1000
		8	750.4	996.7	999.8
		10	288.3	511.3	738.2
		15	1000	1000	1000
		18	1000	1000	1000
	0.3	10	607.5	961.8	985
		15	948.6	947	991
		18	998.9	1000	1000
	0.5	15	424.7	834.4	990.3
		18	260.2	406	618.5
Jaeschke	0.1	7	0	0	0
		8	760.7	992.9	1000
		10	280.8	664.9	847.3
		18	1000	1000	1000
	0.3	10	505	899.5	877.5
		18	762.1	921.4	904.2
	0.5	18	228.4	473.8	802.3
Jackson	0.1	9	788	997.8	1000
		10	322.1	633.8	791.6
		13	985.8	1000	1000
		14	1000	1000	1000
		21	1000	1000	1000
	0.3	13	96.3	373.4	696.6
		14	815	997.8	1000
		21	257.3	787.8	897.9
	0.5	14	826.1	647.7	495.7
		21	91	725.1	770.7

Table 4.1: Number of appearances between the 1000 bootstrap replications, of the outcome in terms of minimum number of workstations from the application of the RB&B to 10′000 observations.

4.1.2 Application of the Bootstrap inspired Methodology to real case Data

It is decided to see the effects of the Bootstrap inspired Procedure on a real problem data. The data used are collected on a Bosch Assembly Line considered in Pesavento (2018). The line is characterized by 15 different tasks. In Table 4.2 the characteristics of the data after a statistical analysis are summarized.

Task	# Observation	Mean	Variance	Chosen Distribution	Chi-square Test	P value
1	98	3.2922	0.7135	LogNormal	Reject	0.0057
2	98	3.7069	0.5788	Gamma	Cannot Reject	0.082747
3	98	3.6932	1.0781	Gamma	Cannot Reject	0.2176
4	98	3.3181	0.5933	Gamma	Cannot Reject	0.2695
5	110	3.1148	1.1	Gamma	Cannot Reject	0.0808
6	111	3.7079	0.969	Gamma	Cannot Reject	0.5313
7	118	4.0639	0.4467	Gamma	Cannot Reject	0.0608
8	119	3.2436	0.015	Gamma	Reject	$3.674 \cdot 10^{-6}$
9	119	1.2299	0.131	Lognormal	Cannot Reject	0.0858
10	119	3.1622	0.6891	Gamma	Cannot Reject	0.1903
11	150	20.3109	0.115	Gamma	Reject	$5.595 \cdot 10^{-6}$
12	50	20.0892	0.0081	Deterministic		
13	50	15.0558	0.0038	Deterministic		
14	150	21.781	0.1062	Gamma	Cannot Reject	0.56961
15	112	9.654	0.1178	Gamma	Reject	0.0001

Table 4.2: Summary of Bosch Data

In the summary, it is possible to observe the limited availability of data and the different availability of observations depending on the task considered. Since the algorithm requires the same number of observations for each task, a manipulation of the data is required to achieve this consistency. The first option consists of the cutting of the data-set according to the least observed tasks. This will bring to a

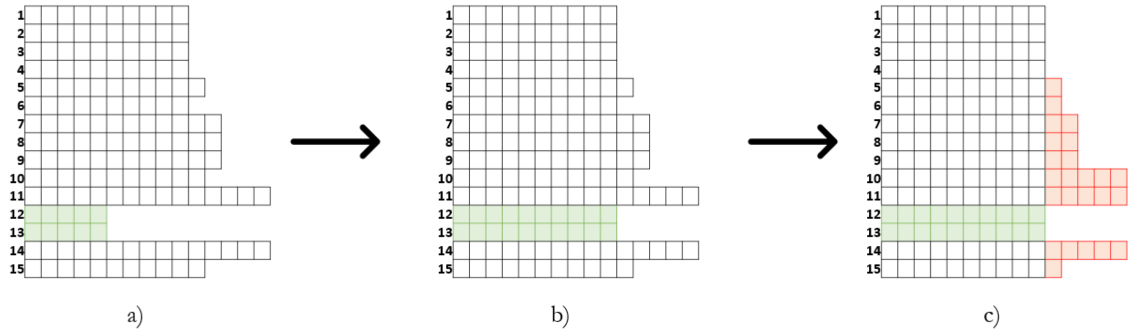


Figure 4.3: Manipulation Process of the Data from the Bosch Assembly Line.

data-set of just 50 observations for each task, having a huge loose in the information provided by the data. Since the two tasks, for which 50 observations are available, are deterministic, it is decided to proceed differently applying two steps that is possible to observe in Figure 4.3:

1. The first step consist into the duplication of the data regarding the two deterministic tasks. At the end of this step, the tasks with 50 observations ends with 100 observations.

2. The second step consists into the cutting of the data-set according to the least observed stochastic task. The data-set end to have 98 observations for each task.

The running of the procedure is performed for different values of B, in order to investigate the influence of this parameter. Three main results are collected and investigated from the procedure:

- Number of different Results (minimum number of workstations to open) Achieved.
- Number of different Assignment Achieved.
- Computational Time Required.

The results in terms of number of stations opened are summarized in Table 4.3

From the results, it is possible to evince some interesting behaviours. First of all, the results in terms of number of stations are not influenced by the value of B, but are strongly influenced by the values of c. This can be explained by the small amount of data available. The situation of uncertainty come from the value of c

c \ B	100	200	300	400	500	600	700	800	900	1000
22	7	7	7	7	7	7	7	7	7	7
23	7	7	7	7	7	7	7	7	7	7
25	6, 7	6, 7	6, 7	6, 7	6, 7	6, 7	6, 7	6, 7	6, 7	6, 7
27	6	6	6	6	6	6	6	6	6	6
30	6	6	6	6	6	6	6	6	6	6
35	5, 6	5, 6	5, 6	5, 6	5, 6	5, 6	5, 6	5, 6	5, 6	5, 6
40	4	4	4	4	4	4	4	4	4	4
45	3, 4	3, 4	3, 4	3, 4	3, 4	3, 4	3, 4	3, 4	3, 4	3, 4
50	3	3	3	3	3	3	3	3	3	3

Table 4.3: Results application of the Procedure to Bosch Data

in those cases when the setting is somehow “borderline” between one or the other number of stations. This variability comes for the stochasticity of the resampling with replacement methodology. In fact, the resampled data bring to a specific output, so, if the methodology resample most of the times sets below the average, the bootstrap estimate will result in the smaller number of workstations. On the contrary, if the resampling methodology will resample the majority of task times sets over the average values, the Bootstrap estimate will result in the higher number of workstations opened. If a decision must be made only with this type of methodology, two possible approaches are possible:

1. The simplest path is to make a conservative choice, choosing the highest number of workstations, that can cover all the possibilities.

2. The second possibility is to collect more data regarding the task times. Of course, this type of path requires the possibility to collect new data on the tasks location.

The second interesting aspect of the results is strictly related to the assignments of the tasks. The number of different assignments in relation with B and c can be found in Table 4.4. This is clearly influenced by the choice of B. The general trend is that the bigger the value of B the more different assignments are obtained. This kind of behaviour can be explained again by the stochasticity of the resampling process. In particular, the more the bootstrap samples generated (and so the bootstrap estimate) the more possible scenarios are investigated. This highlights the importance of using a high value of B, in order to be sure to investigate all the frequent scenarios.

c \ B	100	200	300	400	500	600	700	800	900	1000
22	2	2	2	2	3	2	3	2	2	2
23	6	7	6	7	8	8	8	8	8	8
25	11	12	13	17	15	16	15	15	16	16
27	3	3	4	5	5	5	5	5	5	5
30	4	4	4	4	4	4	5	5	4	4
35	3	4	5	5	5	5	5	5	5	5
40	2	2	2	2	2	2	2	2	2	2
45	4	4	4	4	4	4	4	4	4	4
50	2	2	2	2	2	2	2	2	2	2

Table 4.4: Assignments application of the Procedure to Bosch Data

It would be recommended to focus on the most frequent scenarios rather than investigating the whole range of possible outcomes. This can be addressed to the fact that, in order to balance the line, it is important to work in the most frequent settings and not to adjust it on the basis of an exceptional task times sequence.

This allows to cover the majority of the situations without lowering the performance. In general, there is no interest in investigating the assignments that appears one time between 1000 bootstrap estimates. Therefore there is no such necessity in using such high values of B.

After the generation of several Bootstrap estimates, it is possible to study the various assignments by looking at the reliability of the various settings. If a deeper investigation is required, it is possible to test the various assignments through simulation models, allowing to better assess the system in order to choose the best assignment. The main problem related to that is that the RB&B does not perform an optimization of the task assignment, meaning that it stops with the first assignment respecting one of the stopping conditions.

The last noticeable outcome of this application to real data is related to the computational times. The values are summarized in Table 4.5.

Plotting the resulting computational times (Figure 4.4), it is possible to see a linear growing trend following the growing of B, for all the cycle times tested. This trend was expected, since the only difference in the process between different B values, consists only in the number of times the resampling is performed. It is basically the same process repeated different times. The interesting aspect of the results comes from the values itself of the computational times: they are characterized for the

B / c	100	200	300	400	500	600	700	800	900	1000
22	26	46	60	92	107	129	165	194	212	233
23	23	36	55	72	93	128	143	168	178	207
25	231	505	791	1030	1296	1559	1787	2031	2384	2677
27	22	42	63	86	106	125	142	163	182	205
30	24	38	50	68	87	112	141	149	165	195
35	293	558	827	1080	1476	1642	1942	2215	2482	2753
40	21	44	71	84	104	117	150	168	156	176
45	25	48	73	104	130	155	190	224	260	278
50	19	34	51	67	85	101	121	140	153	172

Table 4.5: Computational Times application of the Procedure to Bosch Data

most of the cases by low values.

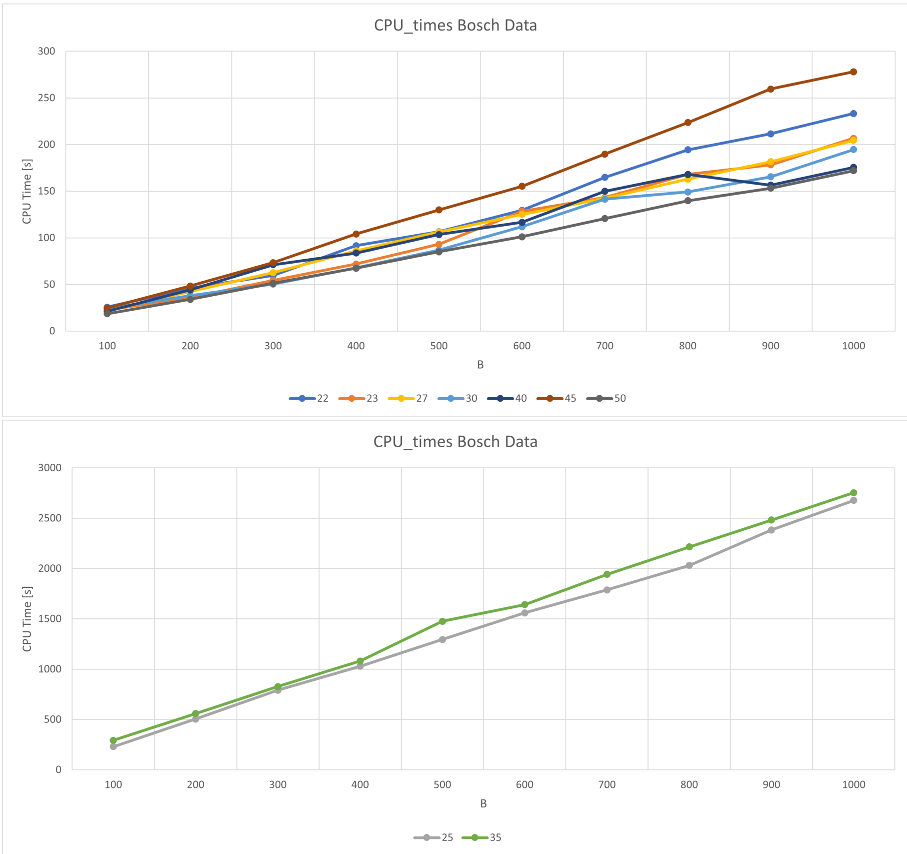


Figure 4.4: Computational Times with Bosch Data, for different values of B with different values of cycle time c

The two exceptions to this are the cases with the cycle time $c = 25$ and $c = 35$ for which more than one possible number of station results. Therefore, it is possible to connect this increase in the computational time, to the higher variability that must be managed by the algorithm and the process in general. The third setting, for which more than one value for the number of stations is achieved, is the one with $c = 45$. Even if the values reached are not as high as the ones reached in the previous cases.

4.1.3 Results analysis

To deeper investigate the reasons behind the limit of the bootstrap itself in bringing to the correct results as the most frequent, an analysis on the single samples is performed. The idea is to highlight at first the potential presence of recurrent samples that bring to an incorrect result. If this presence is discovered, an analysis of the specific samples is performed.

The experiments are done for the same two models on which the bootstrap inspired procedure was tested, but only for the cycle time values for which more than one result in terms of resulting number of stations was achieved in the independent replications experiments (Section 3.3.2).

The analysis is carried out by plotting the frequency of each sample, for each resulting number of stations, as an histogram as in Figure 4.5.

The resulting sample of interest are summarized in Table 4.6.

Model	**CV**	**Distribution**	**Cycle Time**	**# different Results**	**Noticeable sets (b)**
Mertens	0.1	Normal	7	1	
			8	2	(47, 78, 94)
			10	2	(69, 100)
		Gamma	7	1	
			8	2	(32, 39, 49)
			10	2	No particular set arises
	0.3	Normal	10	1	
			15	1	
		Gamma	10	2	No particular set arises
			15	2	(41, 76, 82)
	0.5	Normal	15	3	(36,78)
			18	1	
		Gamma	15	3	(35, 71)
			18	2	(3, 80)

Jackson	0.1	Normal	9	2	(49)
			10	2	(47, 90, 92)
		Gamma	9	2	(18, 59, 90)
			10	2	No particular set arises
	0.3	Normal	13	1	(29, 54, 73)
			14	2	
			21	1	
		Gamma	13	2	(73)
			14	3	(34, 75, 100)
			21	2	(65, 84)
	0.5	Normal	14	3	(28, 57, 62, 82, 100)
			21	1	
		Gamma	14	3	(85)
			21	2	(82)

Table 4.6: Experiments to find characteristics of ”bad” sets.

In the example provided in Figure 4.5, the correct outcome is a resulting number of stations equal to 7. Examining the resulting minimum number of stations to open for the two wrong results that occurs, it is possible to notice that for the case with 6 stations to open no particular task time sequence stand out. On the contrary five different tasks sequences stand out for the case in which 8 different station are opened by the algorithm.

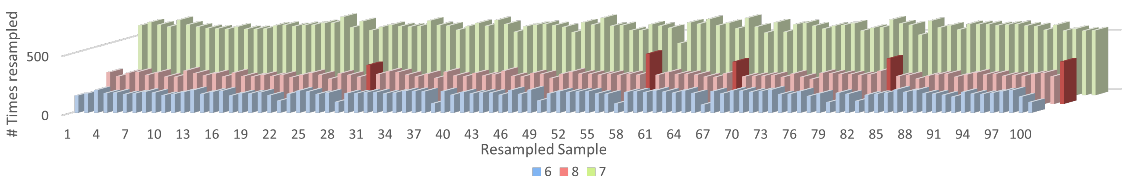


Figure 4.5: Example of the analysis with the histogram plots.

This kind of reasoning is performed for each experiment. All the sequences pointed out, are then analysed individually, in order to figure out if some specific characteristics are present. For each of them the following parameters are evaluated for each task:

- Mean.
- Variance.
- Minimum.
- Maximum.
- Skewness.
- Kurtosis.
- Coefficient of Variation.

Looking at the more frequent sets for the various replications, it is possible to notice that the minimum and the maximum coefficient of variation of the set between the various tasks, are always respectively below and over the one used to sample the times. Other similar behaviour can be observed also for the other parameters. This enforce the hypothesis that there is no particular characteristics of the sequence which brings to higher or lower number of workstations with respect to the optimal solution. It is possible to conclude that there is no specific characteristic of the task times sequences, that leads to an incorrect results, is just the combination of times for the various tasks.

4.2 Problem Definition: Optimization of the Task allocation

As previously said, the objective function of the algorithm proposed by Diefenbach and Stolletz (2020) is expressed as:

$$\min \sum_{m=1}^{M} Z_m$$

Therefore, two main output are obtained by the algorithm. The first one is a vector $X_{i,m}$ that represent an assignment of the task to the various stations. It is important to highlight that this assignment is not optimized, it is the first tested assignment that respect the constraints of the mathematical model proposed and achieve the conditions to stop the Reliability-based Branch and Bound presented in Diefenbach and Stolletz (2020). The second output will be a vector Z_m so that the sum of its elements represent the minimum number of workstations to be opened according to the problem.

When the algorithm is inserted in the Bootstrap Inspired Procedure, B different datasets DB^b are created resampling with replacement the original one and they can be called bootstrap samples. Those bootstrap samples represents different scenarios. The RB&B will be applied to each of them generating B different output, related to each bootstrap sample. In particular it is possible to identify a series of assignments that will not be considered since they are not optimized and a series of Z_m^b with $b \in \{1, \ldots, B\}$.

For each of those scenarios, it is possible to define the number of stations to open as:

$$M^b = \sum_{m=1}^{M} Z_m^b$$

Differently from the previous model, in which the number of workstations is not constrained, by fixing both a number of workstations to open M^b, specific for each scenario, and the cycle time c, common to all the B bootstrap samples analyzed, it is now possible to develop an ALBP-F problem with the aim of optimizing the task assignment in each of those scenarios. The aim of this model is to maximize the reliability of the system, therefore the percentage of completed jobs over the observations available. The variables of the problem can be defined as:

- B_n^b : it is the sample variable. It is equal to 0 if the cycle time c is exceeded for sample n at least at one station, for the $b-th$ bootstrap sample.
- $X_{i,m}^b$: it is the station-variable. It is equal to 1 if task i is assigned to station m and 0 otherwise, relative to the $b-th$ bootstrap sample.

The constraints of this problem must reflect the fact that each task have to be assigned to exactly one workstation. Furthermore, the precedence relationships must be respected.

The mathematical model of the problem can then be expressed as:

$$\max \sum_{n=1}^{N} B_n^b / N \tag{4.1}$$

$$s.t. \qquad \sum_{i=1}^{I} t_{n,i} \cdot X_{i,m}^b \leq c + (1 - B_n^b) \cdot BigM \qquad m = 1, \ldots, M; n = 1, \ldots, N \tag{4.2}$$

$$\sum_{m=1}^{M} X_{i,m}^b = 1 \qquad i = 1, \ldots, I \tag{4.3}$$

$$\sum_{m=1}^{M} m \cdot X_{i,m}^b \leq \sum_{m=1}^{M} m \cdot X_{j,m}^b \qquad \forall (i,j) \in P \tag{4.4}$$

$$X_{i,m}^b \in \{0,1\} \qquad i = 1, \ldots, I; m = 1, \ldots, M \tag{4.5}$$

$$B_n^b \in \{0,1\} \qquad n = 1, \ldots, N \tag{4.6}$$

The objective function (4.1) maximize the reliability for the sets of data available. The constraint (4.2) ensures that the sum of all task times assigned to station m stays within the cycle time c if the sample-variable B_n^b is equal to 1. In case of a sample n that exceeds the cycle time, B_n^b must be set equal to 0 in order to respect the condition. The constraint (4.3) assure that each task is assigned to exactly one workstation. The constraint (4.4) takes into account the precedence that must be respected. The constraints (4.5) and (4.6) define the domain of the variables.

In order to solve the problem, the mathematical model will be solved for $b \in \{1, \ldots, B\}$, therefore it will be solved B times, one for each bootstrap sample generated. This will generate at the maximum B different possible assignment. Each of them will be tested in order to find the most robust.

The entire Bootstrap inspired procedure with the optimal task assignment search requires an inizialization that includes the collection of the data. The steps of the procedure are:

1. Creation of the Data-set DS from the collected data and set of c, B and $b = 0$
2. Set R to the desired minimum value for the reliability and $b = b + 1$. If $b \leq B$ go to Step 3, Otherwise go to Step 9
3. Generation of DS^b by resampling with replacement DS.
4. Evaluation of the maximum reliability achievable R_{max} by DS^b. If $R_{max}\hat{a}R$ go to step 6 otherwise go to Step 5.
5. Set $R = R_{max}$
6. Application of the RB&B to DS^b, generating M^b
7. Application of the task assignment optimization, Generating OTA^b. If $OTA^b \in OTAS$ go to Step 2 otherwise go to Step 8.
8. Add OTA^b to $OTAS$. Go to step 2.
9. Stop.

The flow chart of the entire procedure is shown in Figure 4.6

4.2.1 Selection Procedure

Once the various optimal task assignments are collected in $OTAS$, a decision must be taken to identify the best option.

To define it, a multi-stage ranking and selection procedure is used, generating new Data-sets DS^n, resampling with replacement the original dataset DS.

Since each alternative assignment will be tested on the same data-set in order to evaluate the reliability, it is possible to state that all the alternatives are compared using common random numbers.

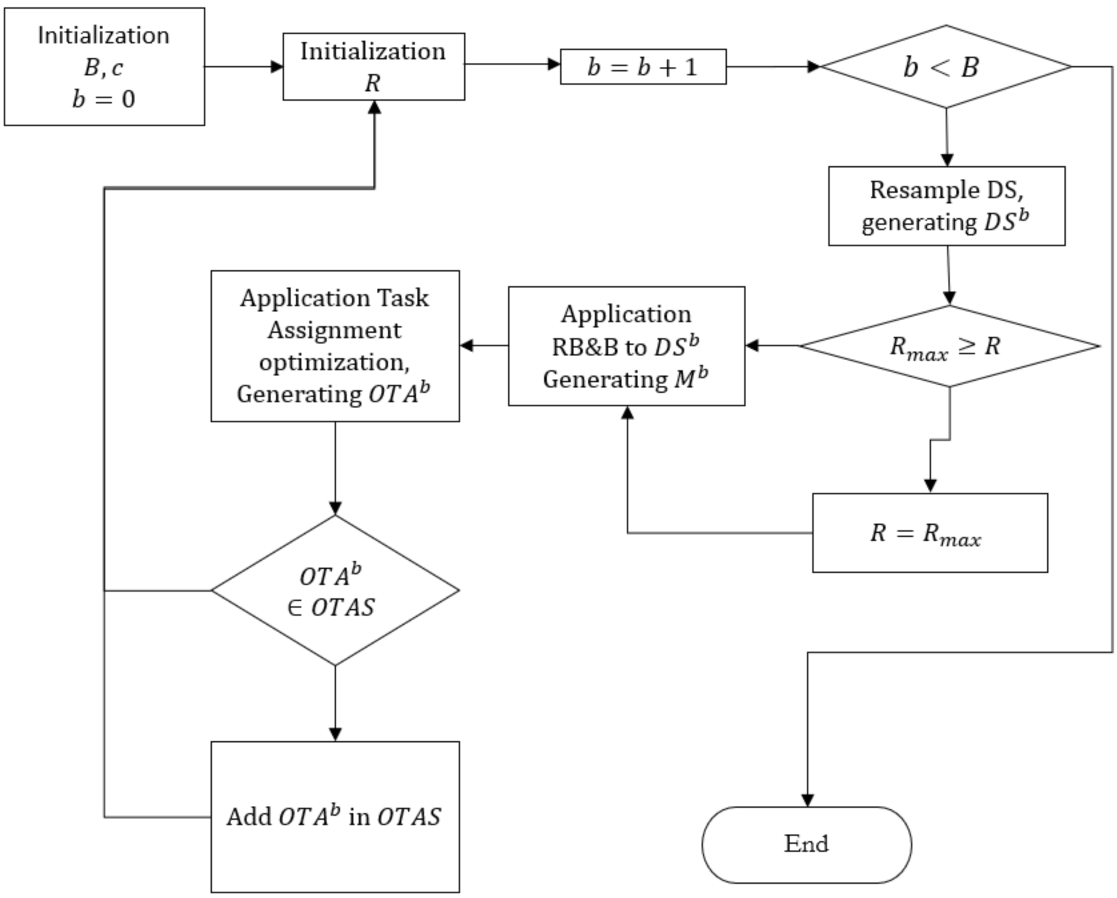


Figure 4.6: Flow Chart representing the Bootstrap Inspired Procedure with task assignment optimization.

The procedure used is a multi-stage ranking and selection procedure. It takes the observation from each population one-at-the-time and eliminates alternatives that seems dominated.

Three parameters must be specified a priori:

- P^*: It is the probability of achieving the optimal selection.
- δ^*: It is an indifference-zone. It represent the sensitivity with which the procedure is able to distinguish differences between the alternatives.
- n_0 : Number of initial dataset generated and used to test the different assignment.

The first step of the procedure include the evaluation of two constants.

$$\eta = 1/2[(\frac{2 \cdot (1 - P^*)}{(k-1)})^{\frac{-2}{n_0 - 1}} - 1]$$

$$h^2 = 2 \cdot \eta \cdot (n_0 - 1)$$

Generating n_0 Dataset DS^n it is possible to evaluate an initial sample of the observations called $Y_{i,j}$ with $1 \leq j \leq n_0$ and $1 \leq i \leq k$ representing the alternatives.

For each population the sample mean $\bar{Y}_i(n_0)$ of the observations is evaluated and based on this mean, and on the sample variance of the difference between the alternatives i and l that is evaluated as:

$$S^2_{i,l} = \frac{1}{n_0 - 1} \sum_{i=1}^{n_0} (Y_{i,j} - Y_{l,j} - [\bar{Y}_i(n_0) - \bar{Y}_l(n_0)])^2$$

For all $i \neq l$ set $N_{i,l} = \lfloor h^2 \frac{S^2_{i,l}}{\delta^{*2}} \rfloor$ and then set $N_i = \max_{i \neq l} N_{i,l}$

If $n_0 > \max_i N_i$ stop and select the population (or more than one) with the largest sample mean $\bar{Y}_i(n_0)$ as one having the largest mean. Otherwise set a counter $r = n_0$ and go to the screening phase.

If $|I| > 1$ generate a new scenario and take an additional observation $Y_{i,r+1}$ of the reliability for those scenarios $i \in I$

Finally set $r = r + 1$

The procedure stops when one of these condition is met:

- $|I| = 1$: The optimal configuration is found.
- If $r = \max_i N_i + 1$: stop and select one assignment from the ones having the index $i \in I$

The procedure can be described by the following steps:

1. Set $I = OTAS$ and evaluation of two constants η and h^2.
2. Generating n_0 Dataset DS^n.
3. Evaluation of an initial sample of the observation called $Y_{i,j}$ with $1 \leq j \leq n_0$ and $1 \leq i \leq k$ representing the alternatives.
4. Evaluation of the sample mean of the observation $\bar{Y}_i(n_0)$ for each of the population $i \in I$.
5. Set $r = n_0$.
6. Evaluation of the sample variance of the difference between the alternatives i and l as:
$$S^2_{i,l} = \frac{1}{n_0 - 1} \sum_{i=1}^{n_0} (Y_{i,j} - Y_{l,j} - [\bar{Y}_i(r) - \bar{Y}_l(r)])^2 \qquad \forall i \neq l, \forall i, l \in I$$
7. For all $i \neq l$ set $N_{i,l} = \lfloor h^2 \frac{S^2_{i,l}}{\delta^{*2}} \rfloor$ and then set $N_i = \max_{i \neq l} N_{i,l}$. If $n_0 > \max(N_i)$ go to Step 13, Otherwise go to Step 8.
8. $I^{old} = I$
9. Evaluation of
$$W_{il}(r) = \max\{0, \frac{\delta^*}{2r}(\frac{h^2 S^2_{il}}{(\delta^*)^2} - r)\}$$
10. Set $I = \{i : i \in I^{old} \text{ and } \bar{Y}_i \geq \bar{Y}_l(r) - W_i l(r), \forall l \in I^{old}, l \neq i\}$. If $|I| = 1$ go to Step 14, otherwise go to Step 11
11. Create one additional Dataset DS^n by resampling with replacement the original one DS.

12. Evaluation of the Reliability of the new dataset $Y_{i,r+1}$ for each configuration $i \in I$.

13. Set $r = r + 1$. If $r < \max_i N_i + 1$ Go to Step 5, otherwise go to Step 14 .

14. Select the best configuration between the ones having index $i \in I$.

The flowchart of the ranking and selection procedure is represented in Figure 4.7.

If, when the procedure stops, the set I contains more than one configuration, it is possible to choose between the different assignments remaining, looking at different performances. Since the main focus of this work is the rebalancing of an existing line, a parameter indicating the "distance" between the new configuration and the original one is the most appropriate evaluation to do.

Two different indexes are considered for this purpose, the first one is the Mean Similarity Factor (MSF) from Gamberini et al (2007), the other one is the Manhattan distance between the assignment as coordinates.

4.2.2 Developed procedure

The developed procedure can be finally summarized in 5 main steps:

1. Collection of the data from the already existing line.

2. Creation of the Bootstrap Samples, by resampling with replacement the original data-set.

3. Application of the RB&B from Diefenbach and Stolletz (2020) to each Bootstrap Sample

4. Application of the Optimization of the task assignment to each Bootstrap sample

5. Application of the Multi-stage Ranking and Selection Procedure.

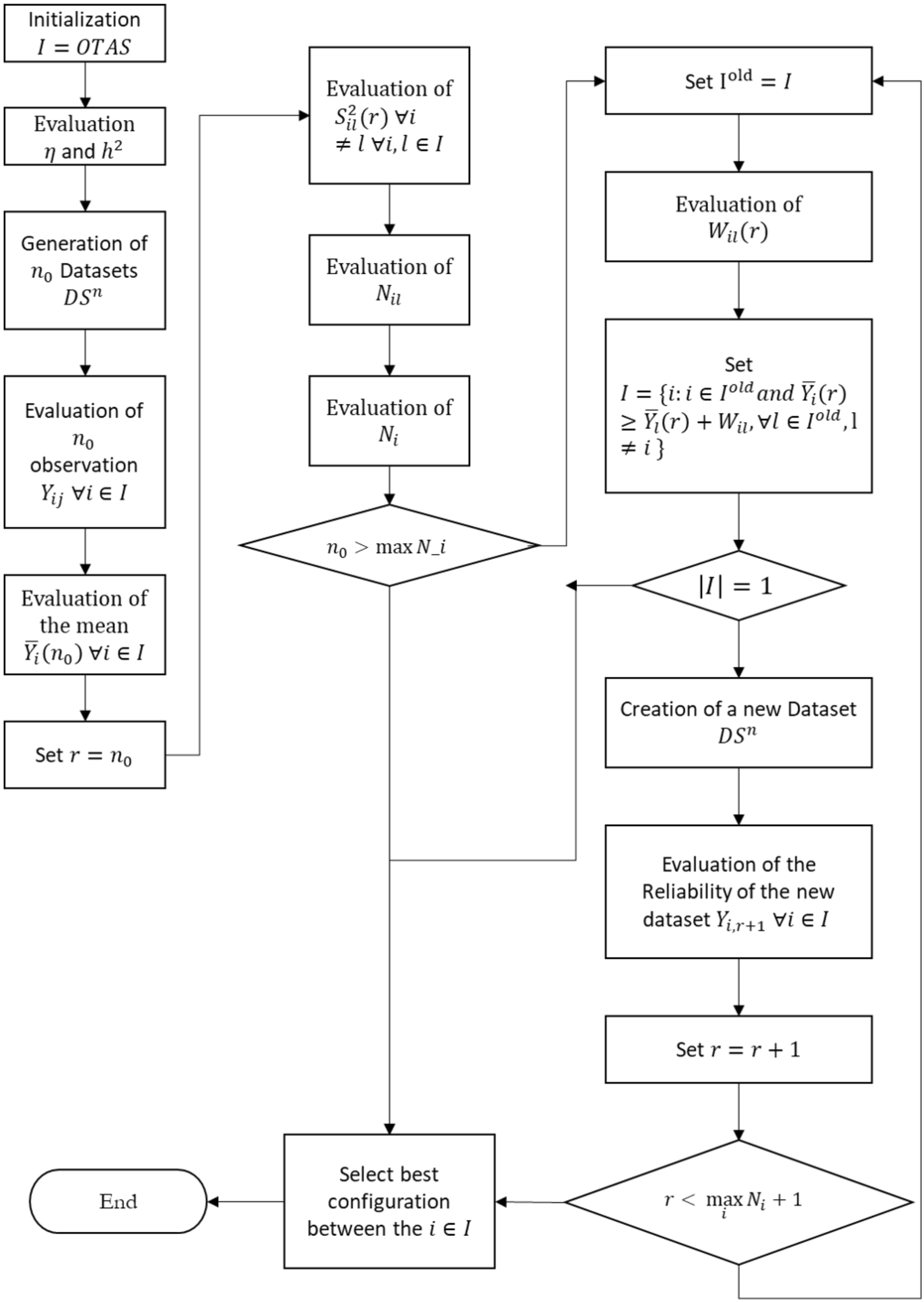


Figure 4.7: Flow Chart Ranking and Selection Procedure

Chapter 5

Experimental Campaign

The previously defined procedure must be tested to evaluate its capabilities. In order to do so, three deterministic model from Scholl (2006) will be used, that are made stochastic as explained in the previous sections.

5.1 Experimental Setting

The model used are: Mertens, characterized by seven tasks, Jaeschke, characterized by nine tasks and Jackson characterized by eleven tasks.

The deterministic task times are used as means of a distribution to generate the data. In particular three values of the coefficient of variation are used ($CV = \{0.1, 0.3, 0.5\}$) to sample the data from a Gamma distribution.

The preliminary step is the generation of the data. In particular $10'000$ observation for each of the 9 settings described are created.

Different approaches are tested, in order to evaluate the capabilities of the developed procedure with a fair comparison. In particular, the approaches tested are.

- Application of the RB&B to the entire dataset ($10'000$ observations).
- Application of the RB&B to a reduced dataset (100 observations).
- Application of the optimization procedure to the output of the RB&B applied on the entire dataset ($10'000$ observations).

- Application of the optimization procedure to the output of the RB&B applied on a reduced dataset (100 observations).
- Application of the Bootstrap Inspired Procedure on a reduced Dataset (100 observations), followed by a multistage Ranking and Selection process.
- Application of the developed methodology on a reduced Dataset (100 observations).

To test the performances of the various approaches, it has been decided to perform 5 replications of the experiment. Therefore 5 datasets for each of the 9 settings are created, for a total of 45 datasets.

The application of the different procedures is performed using one value of the cycle time for each dataset generated but 5 different values of the target for the reliability.

To perform the comparison, all the assignments resulting from each approach, are tested on the original dataset with $10'000$ samples of the task times.

5.1.1 Application of the RB&B

In order to evaluate the performances of the new procedure, the first step is the application of the Reliability Branch and Bound on the $10'000$ observation generated.

The total number of instances analysed is then 45, for each of the replications. The results of the first replication are reported in Table 5.1

Mod.	**Distr.**	**CV**	**c** [s]	R^* [%]	**Assignment**	**R** [%]
				75	1, 1, 4, 5, 2, 3, 6	92.2
				80	1, 1, 4, 5, 2, 3, 6	92.2
		0.1	7	85	1, 1, 4, 5, 2, 3, 6	92.2
				90	1, 1, 4, 5, 2, 3, 6	92.2
				95	1, 2, 3, 1, 4, 6, 5	94.76
				75	1, 1, 5, 2, 2, 3, 4	83.61
				80	1, 1, 5, 2, 2, 3, 4	83.61
Mertens	Gamma	0.3	10	85	1, 1, 2, 2, 3, 5, 4	92.5
				90	1, 1, 4, 4, 2, 3, 5	92.5
				95	1, 1, 4, 5, 2, 3, 6	95.53
				75	1, 1, 2, 3, 1, 2, 3	76.99
				80	1, 1, 4, 2, 1, 2, 3	81.8
		0.5	15	85	1, 1, 3, 3, 1, 2, 4	85.32

				90	2, 2, 3, 5, 3, 4, 5	92.17
				95	1, 1, 5, 1, 3, 4, 2	95.08
Jaeschke	Gamma	0.1	7	75	1, 3, 2, 4, 6, 5, 4, 7, 8	91.96
				80	1, 3, 2, 4, 6, 5, 4, 7, 8	91.96
				85	1, 3, 2, 4, 6, 5, 4, 7, 8	91.96
				90	1, 3, 2, 4, 6, 5, 4, 7, 8	91.96
				95	1, 2, 3, 4, 5, 7, 6, 6, 8	94.73
		0.3	10	75	1, 2, 2, 3, 4, 5, 6, 4, 6	78.34
				80	1, 2, 2, 3, 4, 5, 5, 4, 6	81.15
				85	1, 2, 3, 4, 5, 7, 6, 6, 8	94.73
				90	1, 2, 2, 3, 5, 4, 3, 6, 7	92.63
				95	1, 2, 3, 4, 5, 7, 7, 6, 8	95.63
		0.5	18	75	1, 1, 1, 2, 2, 2, 3, 3, 3	76.53
				80	1, 1, 1, 2, 3, 2, 2, 3, 4	90.3
				85	1, 1, 1, 2, 3, 2, 2, 3, 4	90.3
				90	1, 1, 1, 2, 3, 2, 2, 3, 4	90.3
				95	1, 2, 1, 2, 3, 3, 2, 4, 5	97.16
Jackson	Gamma	0.1	9	75	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	81.66
				80	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	81.66
				85	1, 2, 2, 4, 1, 3, 5, 3, 5, 6, 7	88.95
				90	1, 1, 3, 2, 4, 3, 6, 4, 7, 5, 8	93.15
				95	1, 4, 3, 2, 1, 5, 3, 6, 4, 7, 8	95.08
		0.3	13	75	1, 1, 3, 4, 1, 2, 4, 2, 5, 3, 5	76.79
				80	1, 3, 2, 4, 5, 3, 5, 3, 6, 5, 6	86.86
				85	1, 1, 4, 2, 2, 1, 4, 3, 5, 6, 6	87.67
				90	1, 2, 2, 4, 5, 3, 5, 3, 6, 5, 6	92.96
				95	1, 1, 3, 2, 1, 4, 3, 5, 4, 6, 7	95.31
		0.5	14	75	1, 2, 2, 4, 1, 2, 5, 3, 6, 5, 6	77.35
				80	1, 2, 3, 5, 1, 2, 6, 4, 6, 7, 7	81.73
				85	1, 4, 3, 2, 5, 5, 5, 6, 7, 8, 8	85.15
				90	1, 4, 3, 2, 1, 4, 4, 5, 6, 7, 8	90.33
				95	1, 3, 5, 2, 7, 3, 7, 4, 8, 6, 9	91.67

Table 5.1: Results of the Application of the RB&B to the full Datasets in the first replication.

Looking at the results, it is important to remark how the RB&B developed by Diefenbach and Stolletz (2020) stops when it finds the first assignment satisfying the stopping conditions, meaning that the assignment is not optimized. In some cases, it is possible to observe the same result for different values of the reliability target chosen. This happens when naturally that assignment is the first analysed during the running of the branch and bound.

5.1.2 Application of the RB&B to a reduced dataset

Since the procedure will be applied on 100 data, the RB&B has been applied also to a reduced dataset, composed by the first 100 observations of the entire task times set (the same dataset on which the procedure will be applied). In this way it has been possible to perform a fair comparison.

The resulting assignment is then tested on the entire set of data to evaluate the reliability. The results of the first replication are reported in Table 5.2. In the table, it is also reported the percentage variation of the performances with respect to the RB&B applied to the full dataset.

Mod.	**Distr.**	**CV**	**c** [s]	R^* [%]	**Assignment**	**R** [%]	**Variation**
Mertens	Gamma	0.1	7	75	1, 1, 4, 5, 2, 3, 6	92%	0%
				80	1, 1, 4, 5, 2, 3, 6	92%	0%
				85	1, 1, 4, 5, 2, 3, 6	92%	0%
				90	1, 1, 4, 5, 2, 3, 6	92%	0%
				95	1, 2, 3, 1, 4, 6, 5	95%	0%
		0.3	10	75	1, 1, 5, 2, 2, 3, 4	84%	0%
				80	1, 1, 5, 2, 2, 3, 4	84%	0%
				85	1, 1, 5, 2, 2, 3, 4	84%	-10%
				90	1, 1, 4, 4, 2, 3, 5	93%	0%
				95	1, 2, 3, 3, 4, 6, 5	94%	-2%
		0.5	15	75	1, 1, 2, 3, 1, 2, 3	77%	0%
				80	2, 3, 4, 2, 3, 4, 2	80%	-3%
				85	1, 1, 3, 3, 1, 2, 4	85%	0%
				90	1, 1, 1, 2, 2, 3, 4	90%	-2%
				95	1, 1, 1, 2, 2, 3, 4	90%	-5%
Jaeschke	Gamma	0.1	7	75	1, 3, 2, 4, 6, 5, 4, 7, 8	92%	0%
				80	1, 3, 2, 4, 6, 5, 4, 7, 8	92%	0%
				85	1, 3, 2, 4, 6, 5, 4, 7, 8	92%	0%
				90	1, 2, 3, 4, 5, 7, 7, 6, 8	92%	0%
				95	1, 2, 3, 4, 5, 7, 6, 6, 8	95%	0%
		0.3	10	75	1, 2, 2, 3, 4, 5, 6, 4, 6	78%	0%
				80	1, 1, 2, 3, 5, 4, 3, 6, 7	83%	3%
				85	1, 1, 2, 3, 5, 4, 3, 6, 7	83%	-7%
				90	1, 2, 2, 3, 5, 4, 3, 6, 7	93%	0%
				95	1, 2, 3, 4, 5, 7, 7, 6, 8	96%	0%

		0.5	18	75	1, 1, 1, 2, 2, 2, 3, 3, 3	77%	0%
				80	1, 1, 1, 2, 2, 2, 3, 3, 4	80%	-12%
				85	1, 1, 1, 2, 3, 2, 2, 3, 4	90%	0%
				90	1, 1, 1, 2, 3, 2, 2, 3, 4	90%	0%
				95	1, 2, 1, 2, 2, 3, 3, 3, 4	92%	-6%
Jackson	Gamma	0.1	9	75	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	82%	0%
				80	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	82%	0%
				85	1, 2, 2, 3, 2, 5, 4, 5, 4, 6, 7	86%	-4%
				90	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 7	89%	-4%
				95	1, 4, 3, 2, 1, 5, 3, 6, 4, 7, 8	95%	0%
		0.3	13	75	1, 3, 1, 2, 4, 3, 4, 3, 4, 5, 5	71%	-8%
				80	1, 4, 1, 2, 2, 4, 3, 5, 3, 6, 6	77%	-12%
				85	1, 1, 4, 2, 1, 1, 4, 3, 6, 5, 6	81%	-8%
				90	1, 5, 2, 3, 4, 5, 4, 5, 4, 6, 6	87%	-6%
				95	1, 2, 2, 3, 4, 5, 4, 5, 4, 6, 6	93%	-2%
		0.5	14	75	1, 4, 2, 3, 5, 4, 5, 4, 5, 6, 6	72%	-6%
				80	1, 1, 3, 4, 3, 1, 5, 2, 5, 6, 6	74%	-9%
				85	1, 2, 2, 3, 2, 5, 4, 5, 4, 6, 6	78%	-8%
				90	1, 2, 2, 3, 1, 4, 4, 5, 7, 6, 7	83%	-8%
				95	1, 2, 2, 3, 2, 4, 4, 5, 6, 7, 8	89%	-3%

Table 5.2: Results of the Application of the RB&B to the reduced Datasets

Looking at the results, it is possible to observe that the application to a reduced dataset always bring to a final assignment that end in a reliability equal or lower than the previous one. This happens with only one exception in the first replication and similar results are obtained also in the other replications.

5.1.3 Optimization of RB&B results

The effect of the optimization of the task assignment (explained in Section 4.2) has been tested as well. The optimization is applied to the results of the RB&B applied to both the entire dataset and the reduced one. The results of the first replication are summarized in the Table 5.3

					Optimization Complete Dataset		Optimization Reduced Dataset	
Mod.	**Distr.**	**CV**	**c** [s]	R^* [%]	**R** [%]	**Var.**	**R** [%]	**Var.**
				75	95%	3%	95%	3%
				80	95%	3%	95%	3%
		0.1	7	85	95%	3%	95%	3%
				90	95%	3%	95%	3%
				95	95%	0%	95%	0%
				75	93%	11%	93%	11%
				80	93%	11%	93%	11%
Mertens	Gamma	0.3	10	85	93%	0%	93%	0%
				90	93%	0%	93%	0%
				95	97%	1%	97%	1%
				75	80%	3%	80%	3%
				80	92%	13%	90%	10%
		0.5	15	85	92%	8%	90%	5%
				90	98%	6%	90%	-2%
				95	98%	3%	90%	-5%
				75	95%	3%	95%	3%
				80	95%	3%	95%	3%
		0.1	7	85	95%	3%	95%	3%
				90	95%	3%	95%	3%
				95	95%	0%	95%	0%
				75	81%	4%	81%	4%
				80	81%	0%	93%	15%
Jaeschke	Gamma	0.3	10	85	93%	4%	93%	4%
				90	93%	1%	93%	1%
				95	96%	1%	96%	1%
				75	79%	3%	77%	0%
				80	95%	5%	92%	2%
		0.5	18	85	95%	5%	92%	2%
				90	95%	5%	92%	2%
				95	98%	1%	92%	-5%
				75	89%	9%	89%	9%
				80	89%	9%	89%	9%
		0.1	9	85	89%	0%	89%	0%
				90	100%	7%	89%	-4%
				95	100%	5%	100%	5%
				75	77%	0%	77%	0%
				80	94%	9%	93%	7%
Jackson	Gamma	0.3	13	85	94%	8%	93%	6%
				90	94%	2%	93%	0%

				95	97%	2%	93%	-3%
				75	77%	0%	78%	1%
				80	85%	4%	78%	-5%
		0.5	14	85	91%	7%	78%	-8%
				90	91%	1%	83%	-8%
				95	92%	0%	89%	-3%

Table 5.3: Results of the optimization of the solution of the RB&B

As expected, the optimization of the task assignment of the solution of the RB&B applied to all data, brings to an improvement. In addition, also the gap in the performances between the application to the $10'000$ data and the one on 100 data is reduced, leading in most of the cases to an optimized solution with 100 observations, that is better than the non optimized one with $10'000$.

5.1.4 Application of the Bootstrap Procedure

Before the test of the entire procedure, a test of just the bootstrap inspired procedure, without the optimization, is performed, in order to understand if the optimization process will lead to certain improvement or not. In this case, 500 Bootstrap replications are performed using just the first 100 observations. The choice between the assignments highlighted is done through a ranking and selection process, like the one used in the procedure with the optimization. Different values of delta are used, chosen through preliminary studies. For all the instances a value $n_0 = 100$ is used.

In the cases in which the Ranking and Selection procedure highlights more than one solution, the decision will be made manually, looking not only at the resulting average reliability, but also at the "distance" between the assignment selected and the original assignment, picking an assignment with a trade-off between those two values. In those cases, the original assignment is considered the one resulting from the application of the RB&B applied to the entire $10'000$ observations long dataset.

The resulting assignment from the Ranking and Selection process, is then tested on the original full dataset.

Mod.	**Distr.**	**CV**	**c** [s]	R^* [%]	**Assignment**	**R** [%]	**Variation**
Mertens	Gamma	0.1	7	75%	1, 1, 4, 5, 2, 3, 6	92%	0%
				80%	1, 1, 4, 5, 2, 3, 6	92%	0%
				85%	1, 2, 3, 1, 4, 6, 5	95%	3%
				90%	1, 2, 3, 1, 4, 6, 5	95%	3%
				95%	1, 2, 3, 1, 4, 6, 5	95%	0%
		0.3	10	75%	2, 2, 4, 4, 3, 6, 5	93%	11%
				80%	1, 1, 4, 4, 2, 3, 5	93%	11%
				85%	1, 1, 4, 5, 2, 3, 6	96%	3%
				90%	1, 2, 3, 3, 4, 6, 5	94%	1%
				95%	1, 2, 3, 1, 4, 6, 5	97%	1%
		0.5	15	75%	1, 1, 3, 3, 1, 2, 4	85%	11%
				80%	1, 1, 1, 2, 2, 3, 4	90%	10%
				85%	4, 4, 6, 5, 6, 7, 5	92%	8%
				90%	1, 2, 5, 1, 3, 4, 1	95%	3%
				95%	3, 3, 5, 5, 4, 7, 6	98%	3%
Jaeschke	Gamma	0.1	7	75%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%	0%
				80%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%	0%
				85%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%	3%
				90%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%	3%
				95%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%	0%
		0.3	10	75%	1, 1, 2, 3, 5, 4, 3, 6, 7	83%	6%
				80%	1, 2, 2, 3, 5, 4, 3, 6, 7	93%	14%
				85%	1, 2, 2, 3, 5, 4, 6, 6, 7	93%	4%
				90%	1, 2, 2, 3, 5, 4, 5, 6, 7	93%	1%
				95%	1, 2, 3, 4, 5, 7, 6, 6, 8	96%	1%
		0.5	18	75%	1, 1, 1, 2, 3, 2, 2, 3, 4	90%	18%
				80%	1, 1, 1, 2, 3, 2, 2, 3, 4	90%	0%
				85%	1, 1, 1, 2, 2, 3, 2, 3, 4	92%	2%
				90%	1, 2, 1, 2, 3, 3, 2, 4, 4	95%	5%
				95%	1, 2, 1, 2, 4, 3, 2, 4, 5	98%	1%
Jackson	Gamma	0.1	9	75%	1, 2, 2, 3, 2, 5, 4, 5, 4, 6, 7	86%	5%
				80%	1, 1, 3, 2, 3, 3, 5, 4, 6, 5, 7	86%	5%
				85%	1, 1, 3, 2, 4, 3, 5, 4, 6, 5, 7	89%	0%
				90%	1, 4, 3, 2, 1, 5, 3, 6, 4, 7, 8	95%	2%
				95%	1, 3, 3, 2, 1, 4, 4, 5, 7, 6, 8	99%	5%
		0.3	13	75%	1, 5, 2, 3, 4, 5, 4, 5, 4, 6, 6	87%	13%
				80%	1, 5, 2, 3, 4, 5, 4, 5, 4, 6, 6	87%	0%
				85%	1, 1, 4, 2, 2, 1, 5, 3, 6, 5, 6	87%	-1%
				90%	1, 2, 2, 3, 4, 2, 4, 5, 4, 6, 6	93%	0%
				95%	1, 1, 3, 2, 1, 3, 5, 4, 6, 5, 6	92%	-3%
		0.5	14	75%	1, 1, 2, 4, 1, 2, 5, 3, 6, 5, 6	77%	-1%
				80%	1, 1, 2, 3, 1, 2, 4, 5, 7, 6, 7	79%	-3%

			85%	1, 1, 2, 4, 1, 2, 6, 3, 6, 5, 7	83%	-3%
			90%	1, 2, 2, 3, 2, 4, 4, 5, 6, 7, 8	89%	-1%
			95%	1, 2, 2, 3, 2, 4, 4, 5, 6, 7, 8	89%	-3%

Table 5.4: Results of the Application of the Bootstrap Procedure without the optimization.

Looking at the results, it is possible to observe how often the Bootstrap application brings to a solution that is better than the one achieved by applying the RB&B with 10′000 data. This can be explained through the capability of the Bootstrap procedure to explore the domain of the results, highlighting more than one solution and choosing the best one. It must be underlined how the running of the Bootstrap procedure with a reduced number of observations, leads several times to an assignment characterized by a significantly lower reliability, with respect to the one from the application of the RB&B to the entire dataset.

Therefore, it is possible to state that this kind of approach will not lead to an improvement of the solution, but it will just increase the variability of the results exploring the domain of the solutions.

5.1.5 Application of the Developed Procedure

The last step is the application of the procedure to a reduced quantity of data. In particular, only the first 100 observation of the generated data will be used to test the procedure.

The data used here correspond to 1% (100 obs.) of the data originally generated (10′000 obs.). This can be translated in the collection of the data of the line lasting 1% required to achieve reliable results using just the RB&B by Diefenbach and Stolletz (2020). The Procedure developed in this work is applied to all the instances explained in the previous section.

The selection of the optimized assignment from the subset $OTAS$ is performed through the Ranking and Selection procedure using different values of delta, chosen with preliminary studies. For all the instaces a value $n_0 = 100$ is used.

As in the previous Section, when the Ranking and Selection procedure highlights more than one solution, the choice is made manually.

To perform a comparison, the assignment chosen by the procedure must be tested on the same datasets used to run the RB&B. Therefore, a simulation to evaluate the reliability on the entire dataset is performed.

Firstly, the results will be observed for each model.

In general, it is expected to observe higher improvements on the instances with lower values of reliabilities used as targets. This happens because while the RB&B stops finding the first assignment that respects that stopping conditions, the procedure is aimed at optimizing the assignment to maximize the reliability.

The first model analysed is the Mertens with 7 tasks. The comparison of the results for the first replication is reported in Table 5.5.

Distr.	**CV**	**c [s]**	R^* [%]	δ	**# Chosen Ass.**	**Assignment**	**R**	**Improvement**
Gamma	0.1	7	75	0.01	26	1, 1, 5, 3, 2, 4, 6	0.922	0%
			80	0.01	25	1, 1, 4, 5, 2, 3, 6	0.922	0%
			85	0.01	24	1, 1, 2, 5, 3, 4, 6	0.922	0%
			90	0.01	27	1, 1, 3, 5, 2, 4, 6	0.922	0%
			95	0.01	28	1, 2, 3, 1, 4, 6, 5	0.9476	0%
	0.3	10	75	0.01	1	1, 1, 2, 2, 3, 4, 5	0.925	11%
			80	0.01	6	1, 1, 4, 4, 2, 3, 5	0.925	11%
			85	0.01	7	1, 1, 2, 2, 3, 5, 4	0.925	0%
			90	0.01	1	1, 2, 4, 1, 3, 6, 5	0.9658	4%
			95	0.01	30	1, 2, 3, 1, 4, 5, 6	0.9658	1%
	0.5	15	75	0.01	1	1, 1, 3, 2, 2, 4, 3	0.9221	20%
			80	0.01	1	1, 1, 3, 2, 3, 4, 2	0.9217	13%
			85	0.01	1	1, 1, 3, 2, 3, 4, 2	0.9217	8%
			90	0.01	1	1, 1, 4, 2, 2, 5, 3	0.9655	5%
			95	0.01	1	1, 1, 3, 3, 2, 4, 5	0.9777	3%
						Average		5.36%

Table 5.5: Results of the Application of the Procedure on Mertens model

In some of the cases the two methodologies bring to the same result, with no improvements in terms of reliability, but it is important to highlight how in none of the instances of this model the procedure brings to a result that is worse than the one from the RB&B. On average an improvement of 5% is observed. The second model analysed is the Jaeschke with 9 tasks. *(Table 5.6)*

Distr.	**CV**	**c** [s]	R^*[%]	δ	**# Chosen Ass.**	**Assignment**	**R**	**Improvement**
			75	0.01	17	1, 3, 2, 4, 6, 5, 6, 7, 8	0.9473	3%
			80	0.01	16	1, 3, 2, 4, 6, 5, 6, 7, 8	0.9473	3%
	0.1	7	85	0.01	16	1, 3, 2, 4, 6, 5, 6, 7, 8	0.9473	3%
			90	0.01	17	1, 3, 2, 4, 6, 5, 6, 7, 8	0.9473	3%
			95	0.01	18	1, 2, 3, 4, 5, 7, 6, 6, 8	0.9473	0%
			75	0.01	1	1, 2, 2, 3, 5, 4, 6, 6, 7	0.9332	19%
			80	0.01	1	1, 2, 2, 3, 5, 4, 5, 6, 7	0.9333	15%
Gamma	0.3	10	85	0.01	1	1, 2, 2, 3, 5, 4, 5, 6, 7	0.9333	-1%
			90	0.01	1	1, 2, 3, 4, 6, 5, 7, 7, 8	0.9643	4%
			95	0.01	1	1, 2, 3, 4, 5, 6, 7, 7, 8	0.9643	1%
			75	0.01	1	1, 2, 1, 2, 2, 3, 4, 3, 4	0.9218	20%
			80	0.01	1	1, 2, 1, 2, 3, 3, 2, 4, 4	0.9492	5%
	0.5	18	85	0.01	1	1, 2, 1, 2, 3, 3, 2, 4, 4	0.9492	5%
			90	0.01	1	1, 2, 1, 2, 3, 3, 2, 4, 4	0.9492	5%
			95	0.01	1	1, 2, 1, 2, 4, 3, 5, 4, 5	0.9815	1%
						Average		7.06%

Table 5.6: Results of the Application of the Procedure on Jaeschke model

As the previous model, in some cases no improvements are present. It is also possible to see that in one case out of the 30 tested for this model, the procedure bring to a result worse than the one from the RB&B. The diminishing of the reliability is around 1.4% so, certainly not a huge reduction, but it is not even negligible and it is worth of noticing.

The average improvement for this model is 7%

The last model analysed is the Jackson with 11 tasks. *(Table 5.7)*

Distr.	**CV**	**c** [s]	R^*[%]	δ	**# Chosen Ass.**	**Assignment**	**R**	**Improvement**
			75	0.01	1	1, 1, 2, 3, 4, 2, 5, 4, 6, 5, 7	0.8868	9%
			80	0.01	1	1, 1, 3, 2, 4, 3, 5, 4, 6, 5, 7	0.8868	9%
	0.1	9	85	0.01	1	1, 2, 2, 3, 4, 4, 4, 5, 7, 6, 8	0.9967	12%
			90	0.01	25	1, 2, 2, 3, 4, 4, 4, 5, 7, 6, 8	0.9967	7%

Gamma

		95	0.01	34	1, 4, 3, 2, 1, 5, 4, 6, 5, 7, 8	0.9946	5%
0.3	13	75	0.01	1	1, 1, 2, 3, 2, 2, 4, 5, 4, 6, 6	0.9439	23%
		80	0.01	1	1, 1, 2, 4, 2, 2, 5, 3, 6, 5, 6	0.9398	8%
		85	0.01	1	1, 1, 3, 2, 3, 3, 5, 4, 6, 5, 6	0.9398	7%
		90	0.01	1	1, 1, 2, 3, 2, 2, 5, 4, 6, 5, 6	0.9398	1%
		95	0.01	1	1, 2, 2, 3, 4, 4, 6, 5, 6, 7, 7	0.9564	0%
0.5	14	75	0.01	1	1, 1, 2, 3, 2, 2, 5, 4, 6, 5, 6	0.7798	1%
		80	0.01	1	1, 1, 2, 4, 5, 2, 6, 3, 6, 5, 7	0.8495	4%
		85	0.01	1	1, 1, 3, 2, 5, 3, 6, 4, 6, 5, 7	0.8495	0%
		90	0.01	1	1, 4, 3, 2, 3, 4, 4, 5, 7, 6, 8	0.9096	1%
		95	0.01	1	1, 4, 3, 2, 3, 4, 4, 6, 5, 7, 8	0.9096	-1%
					Average		7.51%

Table 5.7: Results of the Application of the Procedure on Jackson model

As the previous two models, for some instances there is no change in the performances. For the majority of the instances there is a significant improvement and for two instances the procedure results with a negligible lower reliability with respect to the simple application of the RB&B. The average improvement is 7%

Looking at the entire set of results the procedure show a beneficial behaviour with respect to the simple application of the Reliability branch and bound. In 37 instances over 45 it results in a significant improvement of the performance of the chosen assignment. Only in two cases it results in a reduction of the performance, 2 of which are negligible.

5.1.6 Results

To compare the capabilities of the procedure with respect to the other approaches, it is necessary to observe how many times the procedure achieves a better result with respect to the other solutions. This analysis is performed looking at all the instances of the five replications. In the Table 5.8 and in Figure 5.1, it is represented the number of times the procedure achieves a better, equal or worse result with respect to the approach of the corresponding column.

	vs.				
	RB&B on 10'000 obs	**Opt. RB&B on 10'000 obs.**	**RB&B on 100 obs.**	**Opt. RB&B on 100 obs.**	**Bootstrap procedure**
Higher	192	57	204	111	131
Equal	22	99	18	92	73
Lower	11	69	3	22	21

Table 5.8: Number of times the procedure achieve a higher, equal or lower results with respect to the other approaches

It is possible to highlight how the procedure achieve an equal or a better result in most of the cases, if compared to all the other methodologies except one. It outperforms all the approaches but the optimization performed on the results of the RB&B applied to the entire dataset. Most of the times the two approaches reach the same solutions and it is not possible to state which one is better, since the difference between the number of times that one, or the other, reaches a better solution is not significant.

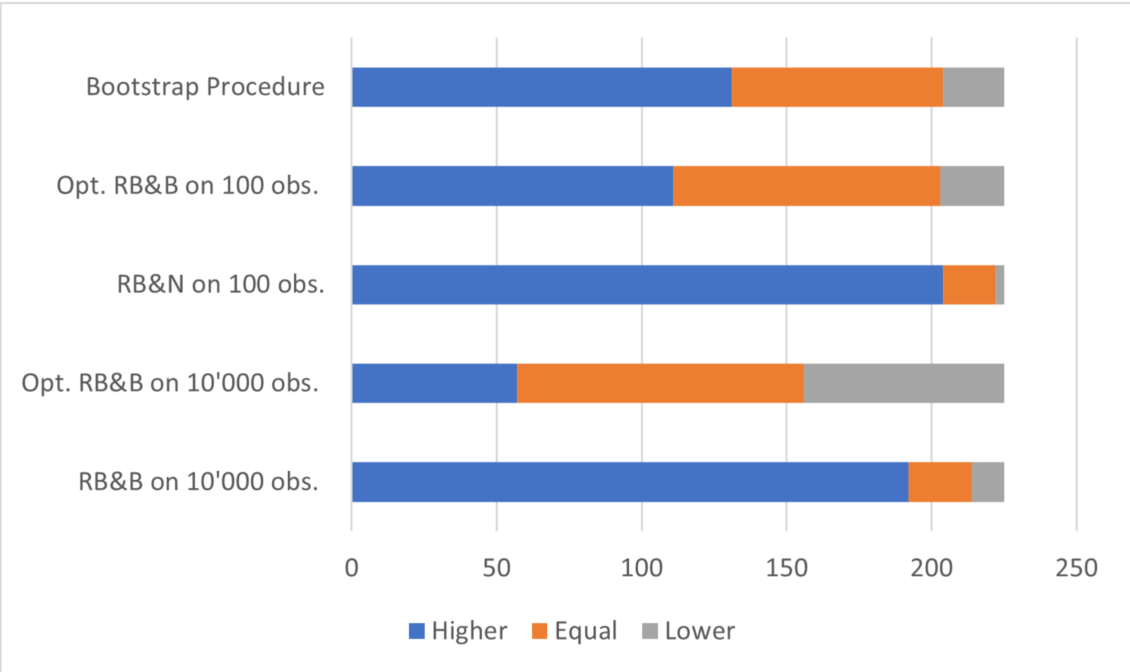


Figure 5.1: Comparison of the results of the developed methodology with the other approaches between the 225 tested instances.

The beneficial behaviour of the procedure comes from the quantity of data needed to feed it. In Table 5.9 it is possible to see the time required to collect the $10'000$ observations with the highest cycle time c value for each of these lines, and the one to collect the 100. The reduction is significant.

Model	c	**Time** 10000 **Observations [Hours]**	**Time** 100 **Observations [min]**
Mertens	15	41.67	25
Jaeschke	18	50	30
Jackson	14	38.89	23.33

Table 5.9: Observation time of the each model with the highest cycle time.

5.1.7 Sensitivity

Given those results it would be interesting to perform a sensitivity analysis on the procedure, increasing the number of observations used to feed it. This kind of study will show up to which point it is necessary to increase the collection of data using the procedure, to reach a result that is at least equal to the one provided by the optimization after the RB&B on $10'000$ data.

It is decided to apply the procedure to the first 500 observations of the $10'000$ generated. Due to constraints related to time and computational power availability, this kind of comparison is performed only to the five replication of the Mertens (7 tasks) and Jaeshke (9 tasks) configurations.

The results of the application of the procedure are reported in Table 5.10

Model	**CV**	**c**	**R***	**Ass. Procedure (500 obs)**	**R Procedure (500 obs)**
			75%	1, 2, 5, 1, 3, 4, 6	95%
			80%	1, 2, 5, 1, 3, 4, 6	95%
	0.1	7	85%	1, 2, 5, 1, 3, 4, 6	95%
			90%	1, 2, 5, 1, 3, 4, 6	95%
			95%	1, 2, 4, 1, 5, 6, 3	95%
			75%	1, 1, 4, 4, 2, 3, 5	93%
			80%	1, 1, 4, 4, 2, 3, 5	93%
Mertens7	0.3	10	85%	1, 1, 2, 2, 3, 5, 4	93%
			90%	1, 2, 4, 1, 3, 5, 6	97%

			95%	1, 2, 4, 1, 3, 5, 6	97%
			75%	1, 1, 3, 1, 2, 3, 2	80%
			80%	1, 1, 2, 3, 2, 4, 3	92%
	0.5	15	85%	1, 1, 2, 4, 2, 3, 4	92%
			90%	1, 1, 2, 4, 2, 3, 4	92%
			95%	1, 1, 2, 2, 3, 4, 5	98%
			75%	1, 3, 2, 4, 6, 5, 6, 7, 8	95%
			80%	1, 3, 2, 4, 6, 5, 6, 7, 8	95%
	0.1	7	85%	1, 3, 2, 4, 6, 5, 6, 7, 8	95%
			90%	1, 3, 2, 4, 6, 5, 6, 7, 8	95%
			95%	1, 3, 2, 4, 5, 7, 6, 6, 8	95%
			75%	1, 2, 2, 3, 5, 4, 4, 5, 6	81%
			80%	1, 2, 2, 3, 4, 6, 5, 5, 7	93%
Jaeschke9	0.3	10	85%	1, 2, 2, 3, 4, 5, 6, 6, 7	93%
			90%	1, 2, 2, 3, 5, 4, 6, 6, 7	93%
			95%	1, 2, 3, 4, 5, 7, 6, 6, 8	96%
			75%	1, 2, 1, 2, 3, 3, 2, 4, 4	95%
			80%	1, 2, 1, 2, 3, 3, 2, 4, 4	95%
	0.5	18	85%	1, 2, 1, 2, 3, 3, 2, 4, 4	95%
			90%	1, 2, 1, 2, 3, 3, 2, 4, 4	95%
			95%	1, 1, 2, 2, 3, 4, 4, 3, 5	98%

Table 5.10: Results application of the procedure with 500 observations.

The results are then compared with the results of the optimization applied on the results of the RB&B implemented on $10'000$ observations. Between the 150 instances investigated, it is possible to notice that:

- 20 times out of 150 the procedure outperforms the optimization of the results of the RB&B.
- 118 times out of 150 the two approaches lead to the same results.
- 12 times out of 150 the procedure achieves a worse result in terms of reliability.

On that results it is possible to state that the two approaches lead in general to similar results. The advantage of using the procedure can be found in the input cost required to achieve a reliable solution. The 500 observations used to run the entire procedure represent the 5% of the $10'000$ required to run the RB&B. Thinking about the $18s$ of the Jaeschke configuration, only 2.5 hours of data collection are required, while 50 hours for the running of the RB&B. The time required for the three systems to collect the data is represented in Figure 5.2.

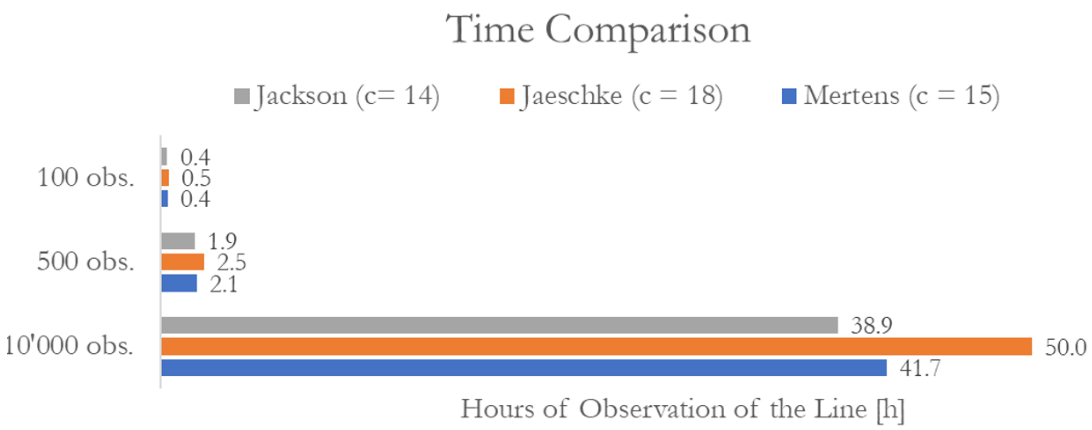


Figure 5.2: Comparison of time required to collect data.

5.1.8 Complex System

The models tested in the previous sections represents simple lines. The most complex configuration is Jackson characterized by 11 tasks and 13 precedence relations to take into account. In order to understand the applicability of the system for future development, it has been decided to test the procedure on the Mitchell configuration, with 21 tasks and 27 precedence relations to consider.

Rel. Target	**RB&B on $10'000$ obs.**	**Procedure**
75	76%	99% (29%)
80	81%	99% (22%)
85	94%	99% (5%)
90	94%	99% (5%)
95	96%	99% (2%)

Table 5.11: Results of the RB&B and the procedure on 21 tasks configuration

The task times observations are created using a $CV = 0.1$ and the RB&B is applied with 5 different reliability targets $R^* = \{75\%, 80\%, 85\%, 90\%, 95\%\}$. The cycle time used is $c = 21$ These configurations are tested with 5 independent replications. The comparison is made between the RB&B applied to $10'000$ observations and the developed procedure applied to 500 observations.

The average between the 5 replications is reported in Table 5.11

It is possible to observe how the outcome of the procedure outperform in all the instances tested the one from the RB&B. The main drawback is related to the computational times. The repeated application of the optimization to the Bootstrap samples requires a huge effort that must be kept in consideration.

This represents the main limit of the procedure. The necessity of hundreds of applications of both the RB&B and the optimization problem is expensive in terms of computational power required. This leaves space for further development of the procedure.

Chapter 6

Conclusions

This work is focused in finding a new evolution to the stochastic assembly line balancing problem. A gap in literature is discovered, due to the extremely limited presence of data-driven methodologies with this purpose. One of the few approach of this kind is the one proposed by Diefenbach and Stolletz (2020). Nonetheless, it is affected by limitations coming from it sample-path nature.

A data-driven method finds straightforward the optimal solution for the entered dataset, bringing to an outcome that is reliable only if the dataset truly represent the line, therefore, only if it is "large enough". This straightforwardness become the major limit of this approach when the collection of the data results difficult and/or onerous.

The aim of this work was to create a methodology able to investigate the line starting from a limited dataset, bringing to a solution that can be an optimum for the line and not only for the dataset available. In order to accomplish this target, the thesis starts from the algorithm developed in Diefenbach and Stolletz (2020).

The first step was to identify how the small amount of data available will create variability into the results of the Reliability-Based Branch & bound developed by Diefenbach and Stolletz (2020). This kind of study allows to define on one side the parameters that influence the results and on the other side a threshold in the data required to achieve a reliable solution. This threshold was identified to be $10'000$ observation for each task time.

When the number of observations for each task results to be lower than this value, the need of exploration of the field of the solution found by the RB&B arises. To satisfy this need, the algorithm was inserted in a Bootstrap inspired procedure, where different scenarios are created resampling with replacement the original dataset. The RB&B was applied to each of those scenarios obtaining a series of possible results, each of which has the same importance.

In order to achieve a single task assignment as an outcome, an optimization of the assignment for each of the results and a ranking and selection procedure was needed. Those two steps were sequenced to the Bootstrap inspired procedure, creating a complete methodology able to provide a finite solution to the problem starting from a reduced dataset.

This methodology is then tested on a series of instances generated according to the most common way in literature, making stochastic some deterministic benchmark models from Scholl (2006).

The procedure was compared to more than one approach:

- The RB&B from Diefenbach and Stolletz (2020), applied on $10'000$ observations.
- An Optimization of the tasks assignment after the application of the RB&B applied on $10'000$ observations.
- The RB&B applied on 100 observations.
- An Optimization of the tasks assignment after the application of the RB&B applied on 100 observations.
- The RB&B inserted in a Bootstrap inspired procedure, sequenced with a ranking and selection procedure (with no optimization). This approach was applied on 100 Observations.

The methodology developed was tested using only 100 observations. It shows good performances, not only using 1% of the data with respect the ones required by the RB&B, but also outperforming almost all the other approaches, bringing to results that are characterized by an equal or even higher reliability achieved. The only approach that resulted with slightly better performances is the application of the optimization after the RB&B on the 10'000 dataset. Hence, it was decided to perform a sensitivity analysis increasing the data used in the developed methodology. It was tested with 500 samples for each task time. This analysis showed a reduction of the gap between the two procedures that achieved the same result in most of the cases. In this case the benefits of using the developed methodology are the usage of 5% of the input data required by the simple usage of the RB&B followed by the optimization problem.

Finally it was decided to test the procedure on a more complex configuration with 21 tasks. This test is performed for just a restricted number of instances and the comparison was performed only between the procedure developed and the RB&B applied to the entire dataset ($10'000$ observations). The developed procedure outperform the RB&B for all the instances tested. This result shows how the effectiveness of the developed methodology is not limited by the characteristics of the analysed configuration.

This kind of methodology, being based on the real observation of the task times, requires the existence of the studied line to be applied, but can find a wide application in real life for different reasons. First of all, it is common to deal with stochastic tasks in today's industrial environment. This happens with both manual and automated tasks. Then, the fact that an existing line is required, makes this procedure suitable to perform a re-balancing of the line, that is a common process after the opening of new one. Being the performance of a line subject to a wide variety of influencing parameter, after the theoretical study that lead to the opening of the line, some adjustments are always required after the opening, moving tasks between stations to optimize the overall behaviour.

At the same time, some limitations can be identified in the procedure. First of all its applicability it is intended to be reduced with the advent of industry 4.0 that allows a smoother and easier collection of data on the lines. Nonetheless, as previously said, the data collection will always have a cost and there will always be cases in which the observation of the line must be limited as much as possible.

Furthermore, the biggest limit of the procedure can be highlighted from the performed tests. The computational effort required by the procedure cannot be neglected. The generation of all the scenarios and the application of the branch and bound and of the optimization problem to all the Bootstrap samples generated require a substantial computational power. The requested power increases with the complexity of the configurations, limiting the instances on which the procedure can be tested in this work.

Based on these limits, it is possible to find space for future development and improvement. One of the possibilities is the structuring of the entire procedure in order to reduce the overall computational times. In this context the optimization is performed through the "Optimization Toolbox" in MatLab. Possible ways to improve the efficiency of the procedure can be studied, looking at the relation between the complexity of the line and the number of bootstrap estimate required to achieve reliable results. The development of a new optimization procedure can help in the reduction of the time required from this part of the developed methodology.

Finally, it would be interesting to run an experimental campaign both on data from real lines with the availability of collection of huge quantities of data and from a more complex systems. This would allows to test the performance on the real field and to perform a proper stress test of the procedure, opening to a new research to deal with higher complexity.

In general, the literature about data-driven stochastic assembly line balancing remains poor. Therefore, any kind of study that enlarge the knowledge about this type of problem and this type of approach to the solution is useful to create a substantial literature about one of the problems that is quite close to the reality and can find huge practical applications. The limited amount of assumptions required on this types of problems makes it worth it for future investigations.

Appendix A

Appendix

A.1 RB&B Replications with data from a Normal Distribution

Line configuration	CV	c	**Different Outcomes** $N = 10'000$	**Different Assignments** $N = 10'000$	**Different Outcomes** $N = 100$	**Different Assignments** $N = 100$
Mertens	0.1	7	1	1	2	5
		8	1	1	2	2
		10	1	1	2	13
		15	1	1	1	2
		18	1	1	1	2
	0.3	10	1	1	3	9
		15	1	1	2	11
		18	1	1	1	6
	0.5	15	1	1	2	31
		18	1	1	2	16
Jackson	0.1	9	1	1	2	27
		10	1	1	2	22
		13	1	1	2	10
		14	1	1	1	10
		21	1	1	1	4
	0.3	13	1	1	2	50
		14	1	1	2	67
		21	1	1	2	42
	0.5	14	1	1	3	305
		21	1	1	2	80

Table A.1: Summary Independent Replication Experiments on data from a Normal Distribution.

A.1.1 Application of the Bootstrap Methodology with $B = 1000$ Bootstrap replications to data-set with $N = 100, 500, 1000$

Config.	**CV**	c	$N = 100$	$N = 500$	$N = 1000$
Mertens7	0.1	7	1000	1000	1000
		8	938.5	999.6	1000
		10	224.5	634	674.5
		15	1000	1000	1000
		18	1000	1000	1000
	0.3	10	711.4	861.6	887.9
		15	996.5	999.9	1000
		18	992.6	1000	1000
	0.5	15	951.8	983.9	985.6
		18	999.7	999.7	1000
Jaeschke9	0.1	7	1000	1000	1000
		8	862.2	997.5	1000
		10	355.8	557.5	687.3
		18	1000	1000	1000
	0.3	10	887.7	817.4	916.3
		18	944.2	1000	1000
	0.5	18	998.5	1000	1000
Jackson11	0.1	9	650.9	995.1	1000
		10	411.7	514.4	660.4
		13	988.2	1000	1000
		14	999.5	1000	1000
		21	1000	1000	1000
	0.3	13	992.4	1000	1000
		14	346.5	938.1	996.8
		21	945.5	922.3	851.6
	0.5	14	342.3	875.5	969.6
		21	999.2	1000	1000

Table A.2: Number of appearances between the 1000 bootstrap replication, of the outcome in terms of minimum number of workstations from the application of the RB&B to $10'000$ observations.

A.1.2 Application of the RB&B to the full dataset

Application of the RB&B to the full dataset (10'000 observations) of all the 5 replications.

Rep.	**Mod.**	**CV**	**c** [s]	R^* [%]	**Assignment**	**R** [%]
Rep 1	Mertens	0.1	7	75%	1, 1, 4, 5, 2, 3, 6	92%
				80%	1, 1, 4, 5, 2, 3, 6	92%
				85%	1, 1, 4, 5, 2, 3, 6	92%
				90%	1, 1, 4, 5, 2, 3, 6	92%
				95%	1, 2, 3, 1, 4, 6, 5	95%
		0.3	10	75%	1, 1, 5, 2, 2, 3, 4	84%
				80%	1, 1, 5, 2, 2, 3, 4	84%
				85%	1, 1, 2, 2, 3, 5, 4	93%
				90%	1, 1, 4, 4, 2, 3, 5	93%
				95%	1, 1, 4, 5, 2, 3, 6	96%
		0.5	15	75%	1, 1, 2, 3, 1, 2, 3	77%
				80%	1, 1, 4, 2, 1, 2, 3	82%
				85%	1, 1, 3, 3, 1, 2, 4	85%
				90%	2, 2, 3, 5, 3, 4, 5	92%
				95%	1, 1, 5, 1, 3, 4, 2	95%
	Jaeschke	0.1	7	75%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				80%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				85%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				90%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				95%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
		0.3	10	75%	1, 2, 2, 3, 4, 5, 6, 4, 6	78%
				80%	1, 2, 2, 3, 4, 5, 5, 4, 6	81%
				85%	1, 2, 2, 3, 4, 6, 7, 5, 7	89%
				90%	1, 2, 2, 3, 5, 4, 3, 6, 7	93%
				95%	1, 2, 3, 4, 5, 7, 7, 6, 8	96%
		0.5	18	75%	1, 1, 1, 2, 2, 2, 3, 3, 3	77%
				80%	1, 1, 1, 2, 3, 2, 2, 3, 4	90%
				85%	1, 1, 1, 2, 3, 2, 2, 3, 4	90%
				90%	1, 1, 1, 2, 3, 2, 2, 3, 4	90%
				95%	1, 2, 1, 2, 3, 3, 2, 4, 5	97%
	Jackson	0.1	9	75%	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	82%
				80%	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	82%
				85%	1, 2, 2, 4, 1, 3, 5, 3, 5, 6, 7	89%
				90%	1, 1, 3, 2, 4, 3, 6, 4, 7, 5, 8	93%
				95%	1, 4, 3, 2, 1, 5, 3, 6, 4, 7, 8	95%
		0.3	13	75%	1, 1, 3, 4, 1, 2, 4, 2, 5, 3, 5	77%
				80%	1, 3, 2, 4, 5, 3, 5, 3, 6, 5, 6	87%

				85%	1, 1, 4, 2, 2, 1, 4, 3, 5, 6, 6	88%
				90%	1, 2, 2, 4, 5, 3, 5, 3, 6, 5, 6	93%
				95%	1, 1, 3, 2, 1, 4, 3, 5, 4, 6, 7	95%
				75%	1, 2, 2, 4, 1, 2, 5, 3, 6, 5, 6	77%
				80%	1, 2, 3, 5, 1, 2, 6, 4, 6, 7, 7	82%
		0.5	14	85%	1, 4, 3, 2, 5, 5, 5, 6, 7, 8, 8	85%
				90%	1, 4, 3, 2, 1, 4, 4, 5, 6, 7, 8	90%
				95%	1, 3, 5, 2, 7, 3, 7, 4, 8, 6, 9	92%
				75%	1, 1, 4, 5, 2, 3, 6	92%
				80%	1, 1, 4, 5, 2, 3, 6	92%
		0.1	7	85%	1, 1, 4, 5, 2, 3, 6	92%
				90%	1, 1, 4, 5, 2, 3, 6	92%
				95%	1, 2, 3, 1, 5, 6, 4	95%
				75%	1, 1, 5, 2, 2, 3, 4	83%
				80%	1, 1, 5, 2, 2, 3, 4	83%
	Mertens	0.3	10	85%	1, 1, 2, 2, 3, 5, 4	92%
				90%	1, 1, 4, 4, 2, 3, 5	92%
				95%	1, 1, 4, 5, 2, 3, 6	95%
				75%	1, 1, 2, 3, 1, 2, 3	77%
				80%	1, 1, 4, 2, 1, 2, 3	82%
		0.5	15	85%	1, 1, 3, 3, 1, 2, 4	85%
				90%	2, 2, 5, 4, 3, 4, 5	90%
				95%	1, 1, 2, 3, 4, 5, 3	96%
				75%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				80%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
		0.1	7	85%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				90%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				95%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				75%	1, 2, 2, 3, 4, 5, 6, 4, 6	79%
				80%	1, 2, 2, 3, 4, 5, 5, 4, 6	82%
Rep 2	Jaeschke	0.3	10	85%	1, 2, 2, 3, 4, 6, 7, 5, 7	89%
				90%	1, 2, 2, 3, 5, 4, 3, 6, 7	92%
				95%	1, 2, 3, 4, 5, 7, 7, 6, 8	95%
				75%	1, 1, 1, 2, 2, 2, 3, 3, 3	76%
				80%	1, 1, 1, 2, 3, 2, 2, 3, 4	90%
		0.5	18	85%	1, 1, 1, 2, 3, 2, 2, 3, 4	90%
				90%	1, 1, 2, 2, 3, 4, 4, 3, 4	91%
				95%	1, 2, 1, 2, 3, 3, 2, 4, 5	97%
				75%	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	82%
				80%	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	82%
		0.1	9	85%	1, 2, 2, 3, 2, 5, 4, 5, 4, 6, 7	85%
				90%	1, 1, 3, 2, 4, 3, 6, 4, 7, 5, 8	93%
				95%	1, 3, 3, 2, 1, 5, 4, 6, 4, 7, 8	95%

Jackson

		0.3	13	75%	1, 1, 3, 4, 1, 2, 4, 2, 5, 3, 5	77%
				80%	1, 3, 2, 4, 5, 3, 5, 3, 6, 5, 6	88%
				85%	1, 1, 5, 2, 2, 1, 5, 3, 6, 4, 6	87%
				90%	1, 2, 2, 4, 5, 3, 5, 3, 6, 5, 6	93%
				95%	1, 1, 3, 2, 1, 4, 3, 5, 4, 6, 7	95%
		0.5	14	75%	1, 2, 2, 4, 1, 2, 5, 3, 5, 6, 6	78%
				80%	1, 2, 3, 5, 1, 2, 6, 4, 6, 7, 7	82%
				85%	1, 6, 3, 2, 3, 6, 4, 7, 5, 8, 8	85%
				90%	1, 4, 3, 2, 1, 4, 4, 5, 6, 7, 8	90%
				95%	1, 5, 2, 3, 2, 5, 4, 6, 7, 8, 9	91%
Rep 3	Mertens	0.1	7	75%	1, 1, 4, 5, 2, 3, 6	92%
				80%	1, 1, 4, 5, 2, 3, 6	92%
				85%	1, 1, 4, 5, 2, 3, 6	92%
				90%	1, 1, 4, 5, 2, 3, 6	92%
				95%	1, 2, 3, 1, 4, 6, 5	95%
		0.3	10	75%	1, 1, 5, 2, 2, 3, 4	84%
				80%	1, 1, 5, 2, 2, 3, 4	84%
				85%	1, 1, 2, 2, 3, 5, 4	93%
				90%	1, 1, 4, 4, 2, 3, 5	93%
				95%	1, 1, 4, 5, 2, 3, 6	96%
		0.5	15	75%	1, 1, 2, 3, 1, 2, 3	77%
				80%	1, 1, 4, 2, 1, 2, 3	82%
				85%	1, 1, 3, 3, 1, 2, 4	85%
				90%	2, 2, 3, 5, 3, 4, 5	92%
				95%	1, 1, 2, 4, 3, 5, 4	96%
	Jaeschke	0.1	7	75%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				80%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				85%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				90%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				95%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
		0.3	10	75%	1, 2, 2, 3, 4, 5, 6, 4, 6	79%
				80%	1, 2, 2, 3, 4, 5, 5, 4, 6	81%
				85%	1, 2, 2, 3, 4, 6, 7, 5, 7	89%
				90%	1, 2, 2, 3, 5, 4, 3, 6, 7	92%
				95%	1, 2, 3, 4, 5, 7, 7, 6, 8	95%
		0.5	18	75%	1, 1, 1, 2, 2, 2, 3, 3, 3	76%
				80%	1, 1, 1, 2, 3, 2, 2, 3, 4	90%
				85%	1, 1, 1, 2, 3, 2, 2, 3, 4	90%
				90%	1, 1, 2, 2, 3, 4, 4, 3, 4	91%
				95%	1, 2, 1, 2, 3, 3, 2, 4, 5	97%
	Jackson	0.1	9	75%	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	82%
				80%	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	82%
				85%	1, 2, 2, 4, 1, 3, 5, 3, 5, 6, 7	89%

				90%	1, 1, 3, 2, 4, 3, 6, 4, 7, 5, 8	93%
				95%	1, 3, 3, 2, 1, 4, 4, 5, 6, 7, 8	99%
				75%	1, 1, 3, 4, 1, 2, 4, 2, 5, 3, 5	76%
				80%	1, 5, 2, 3, 4, 5, 4, 5, 4, 6, 6	87%
		0.3	13	85%	1, 1, 5, 2, 2, 1, 5, 3, 6, 4, 6	88%
				90%	1, 2, 2, 3, 4, 5, 4, 5, 4, 6, 6	93%
				95%	1, 1, 3, 2, 1, 4, 3, 5, 4, 6, 7	95%
				75%	1, 2, 2, 4, 5, 3, 5, 3, 6, 5, 6	77%
				80%	1, 2, 3, 5, 1, 2, 6, 4, 6, 7, 7	81%
		0.5	14	85%	1, 4, 3, 2, 4, 5, 6, 5, 6, 7, 8	86%
				90%	1, 4, 2, 3, 2, 4, 4, 5, 6, 7, 8	90%
				95%	1, 3, 2, 6, 7, 3, 7, 4, 8, 5, 9	91%
				75%	1, 1, 4, 5, 2, 3, 6	92%
				80%	1, 1, 4, 5, 2, 3, 6	92%
		0.1	7	85%	1, 1, 4, 5, 2, 3, 6	92%
				90%	1, 1, 4, 5, 2, 3, 6	92%
				95%	1, 2, 3, 1, 4, 6, 5	94%
				75%	1, 1, 5, 2, 2, 3, 4	84%
				80%	1, 1, 5, 2, 2, 3, 4	84%
	Mertens	0.3	10	85%	1, 1, 2, 2, 3, 5, 4	92%
				90%	1, 1, 4, 4, 2, 3, 5	92%
				95%	1, 1, 4, 5, 2, 3, 6	96%
				75%	1, 1, 2, 3, 1, 2, 3	77%
				80%	1, 1, 4, 2, 1, 2, 3	81%
		0.5	15	85%	1, 1, 4, 3, 2, 4, 3	89%
				90%	2, 2, 5, 3, 3, 4, 5	92%
				95%	1, 1, 2, 4, 3, 5, 4	96%
				75%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				80%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
		0.1	7	85%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				90%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				95%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				75%	1, 2, 2, 3, 4, 5, 6, 4, 6	79%
				80%	1, 2, 2, 3, 4, 5, 5, 4, 6	81%
Rep 4	Jaeschke	0.3	10	85%	1, 2, 2, 3, 4, 6, 7, 5, 7	90%
				90%	1, 2, 2, 3, 5, 4, 3, 6, 7	92%
				95%	1, 2, 3, 4, 5, 7, 7, 6, 8	95%
				75%	1, 1, 1, 2, 2, 2, 3, 3, 3	75%
				80%	1, 1, 1, 2, 3, 2, 2, 3, 4	89%
		0.5	18	85%	1, 1, 1, 2, 3, 2, 2, 3, 4	89%
				90%	1, 1, 1, 2, 3, 2, 3, 3, 4	91%
				95%	1, 2, 1, 2, 3, 3, 2, 4, 5	97%
				75%	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	82%

0.1 9

Jackson

				80%	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	82%
				85%	1, 2, 2, 3, 2, 5, 4, 5, 4, 6, 7	86%
				90%	1, 1, 3, 2, 3, 3, 6, 4, 7, 5, 8	90%
				95%	1, 3, 3, 2, 1, 4, 4, 5, 7, 6, 8	99%
				75%	1, 1, 3, 4, 1, 2, 4, 2, 5, 3, 5	77%
				80%	1, 5, 2, 3, 4, 5, 4, 5, 4, 6, 6	87%
		0.3	13	85%	1, 1, 5, 2, 2, 1, 5, 3, 6, 4, 6	87%
				90%	1, 2, 2, 3, 4, 5, 4, 5, 4, 6, 6	93%
				95%	1, 1, 3, 2, 1, 4, 3, 5, 4, 6, 7	96%
				75%	1, 2, 2, 4, 5, 3, 5, 3, 6, 5, 6	78%
				80%	1, 2, 3, 5, 1, 2, 6, 4, 6, 7, 7	82%
		0.5	14	85%	1, 4, 3, 2, 3, 4, 4, 5, 6, 7, 7	85%
				90%	1, 4, 3, 2, 1, 4, 4, 5, 6, 7, 8	91%
				95%	1, 4, 2, 3, 2, 4, 4, 5, 6, 7, 8	91%
				75%	1, 1, 4, 5, 2, 3, 6	92%
				80%	1, 1, 4, 5, 2, 3, 6	92%
		0.1	7	85%	1, 1, 4, 5, 2, 3, 6	92%
				90%	1, 1, 4, 5, 2, 3, 6	92%
				95%	1, 2, 3, 1, 4, 6, 5	95%
				75%	1, 1, 5, 2, 2, 3, 4	84%
				80%	1, 1, 5, 2, 2, 3, 4	84%
	Mertens	0.3	10	85%	1, 1, 2, 2, 3, 5, 4	93%
				90%	1, 1, 4, 4, 2, 3, 5	93%
				95%	1, 1, 4, 5, 2, 3, 6	96%
				75%	1, 1, 2, 3, 1, 2, 3	77%
				80%	1, 1, 4, 2, 1, 2, 3	82%
		0.5	15	85%	1, 1, 3, 3, 1, 2, 4	85%
				90%	2, 2, 3, 5, 3, 4, 5	92%
				95%	1, 1, 2, 4, 3, 5, 4	96%
				75%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				80%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
		0.1	7	85%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				90%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				95%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				75%	1, 2, 2, 3, 4, 5, 6, 4, 6	79%
				80%	1, 2, 2, 3, 4, 5, 5, 4, 6	81%
Rep 5	Jaeschke	0.3	10	85%	1, 2, 2, 3, 4, 6, 7, 5, 7	89%
				90%	1, 2, 2, 3, 5, 4, 3, 6, 7	92%
				95%	1, 2, 3, 4, 5, 7, 7, 6, 8	95%
				75%	1, 1, 1, 2, 2, 2, 3, 3, 3	76%
				80%	1, 1, 1, 2, 3, 2, 2, 3, 4	90%
		0.5	18	85%	1, 1, 1, 2, 3, 2, 2, 3, 4	90%
				90%	1, 1, 2, 2, 3, 4, 4, 3, 4	91%

			95%	1, 2, 1, 2, 3, 3, 2, 4, 5	97%
			75%	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	82%
			80%	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	82%
	0.1	9	85%	1, 2, 2, 4, 1, 3, 5, 3, 5, 6, 7	89%
			90%	1, 1, 3, 2, 4, 3, 6, 4, 7, 5, 8	93%
			95%	1, 3, 3, 2, 1, 4, 4, 5, 6, 7, 8	99%
			75%	1, 1, 3, 4, 1, 2, 4, 2, 5, 3, 5	76%
			80%	1, 5, 2, 3, 4, 5, 4, 5, 4, 6, 6	87%
Jackson	0.3	13	85%	1, 1, 5, 2, 2, 1, 5, 3, 6, 4, 6	88%
			90%	1, 2, 2, 3, 4, 5, 4, 5, 4, 6, 6	93%
			95%	1, 1, 3, 2, 1, 4, 3, 5, 4, 6, 7	95%
			75%	1, 2, 2, 4, 5, 3, 5, 3, 6, 5, 6	77%
			80%	1, 2, 3, 5, 1, 2, 6, 4, 6, 7, 7	81%
	0.5	14	85%	1, 4, 3, 2, 4, 5, 6, 5, 6, 7, 8	86%
			90%	1, 4, 2, 3, 2, 4, 4, 5, 6, 7, 8	90%
			95%	1, 3, 2, 6, 7, 3, 7, 4, 8, 5, 9	91%

Table A.3: Results of the Application of the RB&B to the full Datasets in all the replications.

A.1.3 Application of the RB&B to the reduced Dataset.

Application of the RB&B to the reduced dataset (100 observations) of all the 5 replications.

Rep.	**Mod.**	**CV**	**c** [s]	R^* [%]	**Assignment**	**R** [%]
				75%	1, 1, 4, 5, 2, 3, 6	92%
				80%	1, 1, 4, 5, 2, 3, 6	92%
		0.1	7	85%	1, 1, 4, 5, 2, 3, 6	92%
				90%	1, 1, 4, 5, 2, 3, 6	92%
				95%	1, 2, 3, 1, 4, 6, 5	95%
				75%	1, 1, 5, 2, 2, 3, 4	84%
				80%	1, 1, 5, 2, 2, 3, 4	84%
	Mertens7	0.3	10	85%	1, 1, 5, 2, 2, 3, 4	84%
				90%	1, 1, 4, 4, 2, 3, 5	93%
				95%	1, 2, 3, 3, 4, 6, 5	94%
				75%	1, 1, 2, 3, 1, 2, 3	77%
				80%	2, 3, 4, 2, 3, 4, 2	80%
		0.5	15	85%	1, 1, 3, 3, 1, 2, 4	85%
				90%	1, 1, 1, 2, 2, 3, 4	90%
				95%	1, 1, 1, 2, 2, 3, 4	90%
				75%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				80%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
		0.1	7	85%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				90%	1, 2, 3, 4, 5, 7, 7, 6, 8	92%
				95%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				75%	1, 2, 2, 3, 4, 5, 6, 4, 6	78%
				80%	1, 1, 2, 3, 5, 4, 3, 6, 7	83%
Rep 1	Jaeschke9	0.3	10	85%	1, 1, 2, 3, 5, 4, 3, 6, 7	83%
				90%	1, 2, 2, 3, 5, 4, 3, 6, 7	93%
				95%	1, 2, 3, 4, 5, 7, 7, 6, 8	96%
				75%	1, 1, 1, 2, 2, 2, 3, 3, 3	77%
				80%	1, 1, 1, 2, 2, 2, 3, 3, 4	80%
		0.5	18	85%	1, 1, 1, 2, 3, 2, 2, 3, 4	90%
				90%	1, 1, 1, 2, 3, 2, 2, 3, 4	90%
				95%	1, 2, 1, 2, 2, 3, 3, 3, 4	92%
				75%	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	82%
				80%	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	82%
		0.1	9	85%	1, 2, 2, 3, 2, 5, 4, 5, 4, 6, 7	86%
				90%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 7	89%
				95%	1, 4, 3, 2, 1, 5, 3, 6, 4, 7, 8	95%
				75%	1, 3, 1, 2, 4, 3, 4, 3, 4, 5, 5	71%
				80%	1, 4, 1, 2, 2, 4, 3, 5, 3, 6, 6	77%
	Jackson11	0.3	13			

				85%	1, 1, 4, 2, 1, 1, 4, 3, 6, 5, 6	81%
				90%	1, 5, 2, 3, 4, 5, 4, 5, 4, 6, 6	87%
				95%	1, 2, 2, 3, 4, 5, 4, 5, 4, 6, 6	93%
				75%	1, 4, 2, 3, 5, 4, 5, 4, 5, 6, 6	72%
				80%	1, 1, 3, 4, 3, 1, 5, 2, 5, 6, 6	74%
		0.5	14	85%	1, 2, 2, 3, 2, 5, 4, 5, 4, 6, 6	78%
				90%	1, 2, 2, 3, 1, 4, 4, 5, 7, 6, 7	83%
				95%	1, 2, 2, 3, 2, 4, 4, 5, 6, 7, 8	89%
				75%	1, 1, 4, 5, 2, 3, 6	92%
				80%	1, 1, 4, 5, 2, 3, 6	92%
		0.1	7	85%	1, 1, 4, 5, 2, 3, 6	92%
				90%	1, 2, 3, 1, 5, 6, 4	95%
				95%	1, 2, 3, 1, 5, 6, 4	95%
				75%	1, 1, 2, 4, 2, 3, 5	69%
				80%	1, 1, 2, 4, 3, 5, 4	83%
	Mertens7	0.3	10	85%	1, 1, 2, 2, 3, 5, 4	92%
				90%	1, 1, 4, 4, 2, 3, 5	92%
				95%	1, 1, 4, 5, 2, 3, 6	95%
				75%	1, 1, 3, 2, 1, 2, 3	79%
				80%	1, 1, 4, 2, 1, 2, 3	82%
		0.5	15	85%	1, 1, 3, 3, 1, 2, 4	85%
				90%	2, 2, 4, 3, 4, 5, 3	92%
				95%	1, 1, 2, 4, 3, 5, 4	96%
				75%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				80%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
		0.1	7	85%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				90%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				95%	1, 2, 3, 4, 5, 7, 7, 6, 8	92%
				75%	1, 1, 2, 3, 5, 4, 4, 5, 6	74%
				80%	1, 1, 2, 3, 5, 4, 3, 6, 7	83%
Rep 2	Jaeschke9	0.3	10	85%	1, 1, 2, 3, 5, 4, 3, 6, 7	83%
				90%	1, 2, 2, 3, 5, 4, 6, 6, 7	93%
				95%	1, 3, 2, 4, 6, 5, 4, 7, 8	95%
				75%	1, 1, 1, 2, 2, 2, 3, 3, 3	76%
				80%	1, 1, 1, 2, 3, 2, 2, 3, 3	76%
		0.5	18	85%	1, 1, 1, 2, 2, 3, 3, 2, 3	79%
				90%	1, 1, 1, 2, 3, 2, 2, 3, 4	90%
				95%	1, 1, 1, 2, 2, 3, 2, 3, 4	91%
				75%	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	82%
				80%	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	82%
		0.1	9	85%	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	82%
				90%	1, 1, 3, 2, 4, 3, 5, 4, 5, 6, 7	89%
				95%	1, 1, 3, 2, 4, 3, 6, 4, 7, 5, 8	93%

Jackson11

				75%	1, 2, 1, 2, 2, 4, 3, 4, 3, 5, 5	71%
				80%	1, 2, 1, 2, 3, 3, 4, 3, 4, 5, 5	73%
		0.3	13	85%	1, 1, 5, 2, 2, 1, 5, 3, 6, 4, 6	87%
				90%	1, 1, 5, 2, 1, 2, 5, 3, 6, 4, 6	89%
				95%	1, 2, 2, 3, 4, 5, 4, 5, 4, 6, 6	93%
				75%	1, 2, 2, 4, 1, 2, 5, 3, 6, 5, 6	78%
				80%	1, 2, 2, 5, 1, 3, 6, 4, 7, 6, 7	81%
		0.5	14	85%	1, 2, 2, 4, 1, 2, 5, 3, 5, 6, 7	83%
				90%	1, 2, 5, 3, 1, 2, 5, 4, 6, 7, 8	88%
				95%	1, 2, 2, 3, 2, 4, 4, 5, 6, 7, 8	89%
				75%	1, 1, 4, 5, 2, 3, 6	92%
				80%	1, 1, 4, 5, 2, 3, 6	92%
		0.1	7	85%	1, 1, 4, 5, 2, 3, 6	92%
				90%	1, 1, 4, 5, 2, 3, 6	92%
				95%	1, 2, 3, 1, 4, 6, 5	95%
				75%	1, 1, 5, 2, 2, 3, 4	84%
				80%	1, 1, 5, 2, 2, 3, 4	84%
	Mertens7	0.3	10	85%	1, 1, 5, 2, 2, 3, 4	84%
				90%	3, 3, 6, 6, 4, 5, 7	93%
				95%	1, 1, 4, 5, 2, 3, 6	96%
				75%	1, 1, 2, 3, 1, 2, 3	77%
				80%	1, 1, 2, 3, 1, 2, 3	77%
		0.5	15	85%	1, 1, 3, 3, 1, 2, 4	85%
				90%	1, 1, 1, 2, 2, 3, 4	89%
				95%	1, 1, 5, 1, 2, 3, 4	95%
				75%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				80%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
		0.1	7	85%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				90%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				95%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				75%	1, 2, 2, 3, 4, 5, 6, 4, 6	79%
				80%	1, 2, 2, 3, 4, 5, 6, 4, 6	79%
Rep 3	Jaeschke9	0.3	10	85%	1, 2, 2, 3, 4, 6, 7, 5, 7	89%
				90%	1, 2, 2, 3, 5, 4, 3, 6, 7	92%
				95%	1, 2, 3, 4, 5, 7, 8, 6, 8	92%
				75%	1, 1, 1, 2, 2, 2, 3, 3, 3	76%
				80%	1, 1, 1, 2, 3, 2, 2, 3, 3	76%
		0.5	18	85%	1, 1, 1, 2, 2, 3, 3, 2, 3	78%
				90%	1, 1, 1, 2, 3, 2, 2, 3, 4	90%
				95%	1, 1, 1, 2, 2, 3, 2, 3, 4	91%
				75%	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	82%
				80%	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	82%
		0.1	9	85%	1, 2, 2, 4, 1, 3, 5, 3, 6, 5, 7	89%

Jackson11

				90%	1, 1, 3, 2, 4, 3, 6, 4, 7, 5, 8	93%
				95%	1, 4, 3, 2, 1, 5, 3, 6, 4, 7, 8	95%
				75%	1, 3, 2, 4, 5, 3, 5, 3, 6, 5, 6	86%
				80%	1, 2, 2, 4, 5, 3, 5, 3, 6, 5, 6	92%
		0.3	13	85%	1, 2, 2, 4, 5, 3, 5, 3, 6, 5, 6	92%
				90%	1, 2, 2, 4, 4, 3, 5, 3, 6, 5, 6	93%
				95%	1, 1, 3, 2, 1, 4, 3, 5, 4, 6, 7	95%
				75%	1, 1, 4, 2, 5, 1, 5, 3, 5, 6, 6	72%
				80%	1, 1, 3, 2, 2, 1, 3, 4, 6, 5, 6	72%
		0.5	14	85%	1, 2, 2, 4, 1, 3, 5, 3, 7, 6, 7	80%
				90%	1, 2, 6, 4, 2, 2, 6, 3, 7, 5, 8	89%
				95%	1, 4, 2, 3, 2, 4, 4, 5, 6, 7, 8	90%
				75%	1, 1, 4, 5, 2, 3, 6	92%
				80%	1, 1, 4, 5, 2, 3, 6	92%
		0.1	7	85%	1, 1, 4, 5, 2, 3, 6	92%
				90%	1, 1, 4, 5, 2, 3, 6	92%
				95%	1, 1, 4, 5, 2, 3, 6	92%
				75%	1, 1, 5, 2, 2, 3, 4	84%
				80%	1, 1, 5, 2, 2, 3, 4	84%
	Mertens7	0.3	10	85%	1, 1, 2, 4, 3, 5, 4	84%
				90%	1, 1, 4, 4, 2, 3, 5	92%
				95%	1, 2, 3, 1, 4, 6, 5	96%
				75%	1, 1, 2, 3, 1, 2, 3	77%
				80%	1, 1, 2, 3, 1, 2, 3	77%
		0.5	15	85%	1, 1, 4, 2, 1, 2, 3	81%
				90%	1, 1, 1, 2, 2, 3, 4	89%
				95%	2, 2, 4, 2, 4, 5, 3	90%
				75%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				80%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
		0.1	7	85%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				90%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				95%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				75%	1, 1, 2, 3, 5, 4, 3, 5, 6	74%
				80%	1, 2, 2, 3, 4, 5, 6, 4, 6	79%
Rep 4	Jaeschke9	0.3	10	85%	1, 2, 2, 3, 5, 4, 3, 5, 6	81%
				90%	1, 2, 2, 3, 5, 4, 3, 6, 7	92%
				95%	1, 2, 2, 3, 5, 4, 3, 6, 7	92%
				75%	1, 1, 1, 2, 2, 3, 2, 2, 3	75%
				80%	1, 1, 1, 2, 3, 2, 2, 3, 4	89%
		0.5	18	85%	1, 1, 1, 2, 3, 2, 2, 3, 4	89%
				90%	1, 1, 1, 2, 3, 2, 2, 3, 4	89%
				95%	1, 1, 2, 2, 3, 3, 4, 4, 4	93%
				75%	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	82%

0.1 9

Jackson11

				80%	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	82%
				85%	1, 1, 3, 2, 2, 3, 5, 4, 6, 5, 7	83%
				90%	1, 1, 3, 2, 4, 3, 5, 4, 6, 5, 7	90%
				95%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 7	89%
				75%	1, 3, 1, 2, 4, 3, 4, 3, 4, 5, 5	71%
				80%	1, 1, 3, 4, 3, 2, 4, 2, 5, 3, 5	70%
		0.3	13	85%	1, 1, 4, 2, 2, 1, 5, 3, 5, 4, 6	82%
				90%	1, 2, 2, 4, 5, 3, 5, 3, 6, 5, 6	93%
				95%	1, 2, 2, 4, 2, 3, 5, 3, 6, 5, 6	94%
				75%	1, 1, 4, 2, 5, 1, 5, 3, 5, 6, 6	73%
				80%	1, 2, 2, 4, 5, 3, 5, 3, 6, 5, 6	78%
		0.5	14	85%	1, 2, 2, 4, 1, 2, 5, 3, 5, 6, 6	78%
				90%	1, 3, 5, 2, 1, 3, 5, 4, 6, 7, 7	82%
				95%	1, 4, 3, 2, 1, 4, 4, 5, 6, 7, 8	91%
				75%	1, 1, 4, 5, 2, 3, 6	92%
				80%	1, 1, 4, 5, 2, 3, 6	92%
		0.1	7	85%	1, 1, 4, 5, 2, 3, 6	92%
				90%	1, 1, 4, 5, 2, 3, 6	92%
				95%	1, 2, 3, 1, 4, 6, 5	95%
				75%	1, 1, 5, 2, 2, 3, 4	84%
				80%	1, 1, 5, 2, 2, 3, 4	84%
	Mertens7	0.3	10	85%	1, 1, 5, 2, 2, 3, 4	84%
				90%	3, 3, 6, 6, 4, 5, 7	93%
				95%	1, 1, 4, 5, 2, 3, 6	96%
				75%	1, 1, 2, 3, 1, 2, 3	77%
				80%	1, 1, 2, 3, 1, 2, 3	77%
		0.5	15	85%	1, 1, 3, 3, 1, 2, 4	85%
				90%	1, 1, 1, 2, 2, 3, 4	89%
				95%	1, 1, 5, 1, 2, 3, 4	95%
				75%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				80%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
		0.1	7	85%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				90%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				95%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				75%	1, 2, 2, 3, 4, 5, 6, 4, 6	79%
				80%	1, 2, 2, 3, 4, 5, 6, 4, 6	79%
Rep 5	Jaeschke9	0.3	10	85%	1, 2, 2, 3, 4, 6, 7, 5, 7	89%
				90%	1, 2, 2, 3, 5, 4, 3, 6, 7	92%
				95%	1, 2, 3, 4, 5, 7, 8, 6, 8	92%
				75%	1, 1, 1, 2, 2, 2, 3, 3, 3	76%
				80%	1, 1, 1, 2, 3, 2, 2, 3, 4	90%
		0.5	18	85%	1, 1, 1, 2, 3, 2, 2, 4, 4	88%
				90%	1, 2, 1, 2, 2, 3, 3, 3, 4	91%

			95%	1, 2, 1, 2, 3, 3, 2, 4, 4	95%
			75%	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	82%
			80%	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	82%
	0.1	9	85%	1, 2, 2, 4, 1, 3, 5, 3, 6, 5, 7	89%
			90%	1, 1, 3, 2, 4, 3, 6, 4, 7, 5, 8	93%
			95%	1, 4, 3, 2, 1, 5, 3, 6, 4, 7, 8	95%
			75%	1, 3, 2, 4, 5, 3, 5, 3, 6, 5, 6	86%
			80%	1, 2, 2, 4, 5, 3, 5, 3, 6, 5, 6	92%
Jackson11	0.3	13	85%	1, 2, 2, 4, 5, 3, 5, 3, 6, 5, 6	92%
			90%	1, 2, 2, 4, 4, 3, 5, 3, 6, 5, 6	93%
			95%	1, 1, 3, 2, 1, 4, 3, 5, 4, 6, 7	95%
			75%	1, 1, 4, 2, 5, 1, 5, 3, 5, 6, 6	72%
			80%	1, 1, 3, 2, 2, 1, 3, 4, 6, 5, 6	72%
	0.5	14	85%	1, 2, 2, 4, 1, 3, 5, 3, 7, 6, 7	80%
			90%	1, 2, 6, 4, 2, 2, 6, 3, 7, 5, 8	89%
			95%	1, 4, 2, 3, 2, 4, 4, 5, 6, 7, 8	90%

Table A.4: Results of the Application of the RB&B to the Reduced Datasets in all the replications.

A.1.4 Optimization of the RB&B results applied on the full dataset

Rep.	**Mod.**	**CV**	**c** [s]	R^* [%]	**Assignment**	**R** [%]
				75%	1, 2, 6, 1, 4, 5, 3	95%
				80%	1, 2, 6, 1, 4, 5, 3	95%
		0.1	7	85%	1, 2, 6, 1, 4, 5, 3	95%
				90%	1, 2, 6, 1, 4, 5, 3	95%
				95%	1, 2, 6, 1, 4, 5, 3	95%
				75%	1, 1, 2, 2, 4, 5, 3	93%
				80%	1, 1, 2, 2, 4, 5, 3	93%
	Mertens7	0.3	10	85%	1, 1, 2, 2, 4, 5, 3	93%
				90%	1, 1, 2, 2, 4, 5, 3	93%
				95%	1, 2, 3, 1, 5, 6, 4	97%
				75%	1, 2, 3, 1, 2, 3, 1	80%
				80%	1, 1, 3, 2, 2, 4, 3	92%
		0.5	15	85%	1, 1, 3, 2, 2, 4, 3	92%
				90%	1, 1, 3, 3, 2, 4, 5	98%
				95%	1, 1, 3, 3, 2, 4, 5	98%
				75%	1, 3, 2, 4, 5, 6, 7, 7, 8	95%
				80%	1, 3, 2, 4, 5, 6, 7, 7, 8	95%
		0.1	7	85%	1, 3, 2, 4, 5, 6, 7, 7, 8	95%
				90%	1, 3, 2, 4, 5, 6, 7, 7, 8	95%
				95%	1, 3, 2, 4, 5, 6, 7, 7, 8	95%
				75%	1, 2, 2, 3, 4, 5, 3, 4, 6	81%
				80%	1, 2, 2, 3, 4, 5, 3, 4, 6	81%
Rep 1	Jaeschke9	0.3	10	85%	1, 2, 2, 3, 4, 6, 5, 5, 7	93%
				90%	1, 2, 2, 3, 4, 6, 5, 5, 7	93%
				95%	1, 2, 3, 4, 5, 6, 7, 7, 8	96%
				75%	1, 1, 1, 2, 2, 3, 3, 2, 3	79%
				80%	1, 2, 1, 2, 3, 3, 2, 4, 4	95%
		0.5	18	85%	1, 2, 1, 2, 3, 3, 2, 4, 4	95%
				90%	1, 2, 1, 2, 3, 3, 2, 4, 4	95%
				95%	1, 1, 2, 2, 4, 3, 3, 4, 5	98%
				75%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 7	89%
				80%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 7	89%
		0.1	9	85%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 7	89%
				90%	1, 3, 3, 2, 4, 4, 4, 5, 6, 7, 8	100%
				95%	1, 3, 3, 2, 4, 4, 4, 5, 6, 7, 8	100%
				75%	1, 1, 3, 4, 2, 2, 4, 2, 5, 3, 5	77%
				80%	1, 1, 2, 4, 2, 2, 5, 3, 5, 6, 6	94%
	Jackson11	0.3	13	85%	1, 1, 2, 4, 2, 2, 5, 3, 5, 6, 6	94%
				90%	1, 1, 2, 4, 2, 2, 5, 3, 5, 6, 6	94%

				95%	1, 1, 2, 5, 4, 2, 6, 3, 6, 4, 7	97%
				75%	1, 4, 3, 2, 2, 5, 3, 5, 4, 6, 6	77%
				80%	1, 2, 2, 3, 5, 4, 6, 4, 6, 5, 7	85%
		0.5	14	85%	1, 4, 2, 3, 2, 4, 4, 5, 6, 7, 8	91%
				90%	1, 4, 2, 3, 2, 4, 4, 5, 6, 7, 8	91%
				95%	1, 2, 6, 5, 7, 2, 7, 3, 8, 4, 9	92%
				75%	1, 2, 3, 1, 4, 5, 6	95%
				80%	1, 2, 3, 1, 4, 5, 6	95%
		0.1	7	85%	1, 2, 3, 1, 4, 5, 6	95%
				90%	1, 2, 3, 1, 4, 5, 6	95%
				95%	1, 2, 3, 1, 4, 5, 6	95%
				75%	1, 1, 4, 4, 2, 3, 5	92%
				80%	1, 1, 4, 4, 2, 3, 5	92%
	Mertens7	0.3	10	85%	1, 1, 4, 4, 2, 3, 5	92%
				90%	1, 1, 4, 4, 2, 3, 5	92%
				95%	1, 2, 5, 1, 3, 6, 4	96%
				75%	1, 1, 3, 1, 2, 3, 2	80%
				80%	1, 1, 2, 4, 2, 3, 4	92%
		0.5	15	85%	1, 1, 2, 4, 2, 3, 4	92%
				90%	1, 1, 2, 2, 4, 5, 3	98%
				95%	1, 1, 2, 2, 4, 5, 3	98%
				75%	1, 2, 3, 4, 6, 5, 7, 7, 8	95%
				80%	1, 2, 3, 4, 6, 5, 7, 7, 8	95%
		0.1	7	85%	1, 2, 3, 4, 6, 5, 7, 7, 8	95%
				90%	1, 2, 3, 4, 6, 5, 7, 7, 8	95%
				95%	1, 2, 3, 4, 6, 5, 7, 7, 8	95%
				75%	1, 2, 2, 3, 5, 4, 4, 5, 6	82%
				80%	1, 2, 2, 3, 5, 4, 4, 5, 6	82%
Rep 2	Jaeschke9	0.3	10	85%	1, 2, 2, 3, 4, 6, 5, 5, 7	93%
				90%	1, 2, 2, 3, 4, 6, 5, 5, 7	93%
				95%	1, 3, 2, 4, 5, 7, 6, 6, 8	96%
				75%	1, 1, 1, 2, 2, 3, 3, 2, 3	79%
				80%	1, 2, 1, 2, 3, 3, 2, 4, 4	95%
		0.5	18	85%	1, 2, 1, 2, 3, 3, 2, 4, 4	95%
				90%	1, 2, 1, 2, 3, 3, 2, 4, 4	95%
				95%	1, 2, 1, 2, 4, 3, 3, 4, 5	98%
				75%	1, 1, 2, 4, 3, 2, 5, 3, 6, 5, 7	88%
				80%	1, 1, 2, 4, 3, 2, 5, 3, 6, 5, 7	88%
		0.1	9	85%	1, 1, 2, 4, 3, 2, 5, 3, 6, 5, 7	88%
				90%	1, 4, 2, 3, 2, 4, 4, 5, 6, 7, 8	100%
				95%	1, 4, 2, 3, 2, 4, 4, 5, 6, 7, 8	100%
				75%	1, 1, 3, 4, 2, 2, 4, 2, 5, 3, 5	77%
				80%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 6	95%
	Jackson11	0.3	13			

				85%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 6	95%
				90%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 6	95%
				95%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 7	97%
				75%	1, 1, 2, 4, 2, 2, 5, 3, 6, 5, 6	78%
				80%	1, 1, 2, 5, 4, 2, 6, 3, 6, 4, 7	85%
		0.5	14	85%	1, 4, 2, 3, 2, 4, 4, 5, 7, 6, 8	91%
				90%	1, 4, 2, 3, 2, 4, 4, 5, 7, 6, 8	91%
				95%	1, 4, 2, 3, 4, 6, 4, 7, 5, 8, 9	91%
				75%	1, 2, 6, 1, 4, 5, 3	95%
				80%	1, 2, 6, 1, 4, 5, 3	95%
		0.1	7	85%	1, 2, 6, 1, 4, 5, 3	95%
				90%	1, 2, 6, 1, 4, 5, 3	95%
				95%	1, 2, 6, 1, 4, 5, 3	95%
				75%	1, 1, 2, 2, 4, 5, 3	93%
				80%	1, 1, 2, 2, 4, 5, 3	93%
	Mertens7	0.3	10	85%	1, 1, 2, 2, 4, 5, 3	93%
				90%	1, 1, 2, 2, 4, 5, 3	93%
				95%	1, 3, 6, 1, 4, 5, 2	97%
				75%	1, 2, 3, 1, 2, 3, 1	80%
				80%	1, 1, 2, 4, 2, 3, 4	92%
		0.5	15	85%	1, 1, 2, 4, 2, 3, 4	92%
				90%	1, 1, 2, 2, 4, 5, 3	98%
				95%	1, 1, 2, 2, 4, 5, 3	98%
				75%	1, 2, 3, 4, 5, 7, 5, 6, 8	95%
				80%	1, 2, 3, 4, 5, 7, 5, 6, 8	95%
		0.1	7	85%	1, 2, 3, 4, 5, 7, 5, 6, 8	95%
				90%	1, 2, 3, 4, 5, 7, 5, 6, 8	95%
				95%	1, 2, 3, 4, 5, 7, 5, 6, 8	95%
				75%	1, 2, 2, 3, 5, 4, 4, 5, 6	81%
				80%	1, 2, 2, 3, 5, 4, 4, 5, 6	81%
Rep 3	Jaeschke9	0.3	10	85%	1, 2, 2, 3, 5, 4, 5, 6, 7	93%
				90%	1, 2, 2, 3, 5, 4, 5, 6, 7	93%
				95%	1, 2, 3, 4, 5, 7, 5, 6, 8	96%
				75%	1, 1, 1, 2, 2, 3, 3, 2, 3	78%
				80%	1, 2, 1, 2, 3, 3, 2, 4, 4	95%
		0.5	18	85%	1, 2, 1, 2, 3, 3, 2, 4, 4	95%
				90%	1, 2, 1, 2, 3, 3, 2, 4, 4	95%
				95%	1, 2, 1, 2, 3, 4, 4, 3, 5	98%
				75%	1, 3, 3, 2, 1, 4, 5, 4, 6, 5, 7	89%
				80%	1, 3, 3, 2, 1, 4, 5, 4, 6, 5, 7	89%
		0.1	9	85%	1, 3, 3, 2, 1, 4, 5, 4, 6, 5, 7	89%
				90%	1, 2, 2, 3, 4, 4, 4, 5, 6, 7, 8	100%
				95%	1, 2, 2, 3, 4, 4, 4, 5, 6, 7, 8	100%

Jackson11

				75%	1, 1, 3, 4, 2, 2, 4, 2, 5, 3, 5	76%
				80%	1, 2, 2, 4, 1, 3, 5, 3, 6, 5, 6	94%
		0.3	13	85%	1, 2, 2, 4, 1, 3, 5, 3, 6, 5, 6	94%
				90%	1, 2, 2, 4, 1, 3, 5, 3, 6, 5, 6	94%
				95%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 7	97%
				75%	1, 1, 2, 3, 2, 2, 5, 4, 6, 5, 6	78%
				80%	1, 4, 3, 2, 3, 4, 4, 5, 7, 6, 7	84%
		0.5	14	85%	1, 4, 2, 3, 2, 4, 4, 6, 5, 7, 8	90%
				90%	1, 4, 2, 3, 2, 4, 4, 6, 5, 7, 8	90%
				95%	1, 2, 5, 3, 2, 2, 7, 4, 8, 6, 9	91%
				75%	1, 3, 6, 1, 4, 5, 2	94%
				80%	1, 3, 6, 1, 4, 5, 2	94%
		0.1	7	85%	1, 3, 6, 1, 4, 5, 2	94%
				90%	1, 3, 6, 1, 4, 5, 2	94%
				95%	1, 3, 6, 1, 4, 5, 2	94%
				75%	1, 1, 3, 3, 2, 4, 5	92%
				80%	1, 1, 3, 3, 2, 4, 5	92%
	Mertens7	0.3	10	85%	1, 1, 3, 3, 2, 4, 5	92%
				90%	1, 1, 3, 3, 2, 4, 5	92%
				95%	1, 2, 4, 1, 3, 5, 6	96%
				75%	1, 2, 3, 1, 2, 3, 1	80%
				80%	1, 1, 4, 2, 2, 3, 4	92%
		0.5	15	85%	1, 1, 4, 2, 2, 3, 4	92%
				90%	1, 1, 2, 2, 4, 5, 3	97%
				95%	1, 1, 2, 2, 4, 5, 3	97%
				75%	1, 3, 2, 4, 5, 7, 5, 6, 8	95%
				80%	1, 3, 2, 4, 5, 7, 5, 6, 8	95%
		0.1	7	85%	1, 3, 2, 4, 5, 7, 5, 6, 8	95%
				90%	1, 3, 2, 4, 5, 7, 5, 6, 8	95%
				95%	1, 3, 2, 4, 5, 7, 5, 6, 8	95%
				75%	1, 2, 2, 3, 4, 5, 3, 4, 6	81%
				80%	1, 2, 2, 3, 4, 5, 3, 4, 6	81%
Rep 4	Jaeschke9	0.3	10	85%	1, 2, 2, 3, 5, 4, 6, 6, 7	93%
				90%	1, 2, 2, 3, 5, 4, 6, 6, 7	93%
				95%	1, 3, 2, 4, 6, 5, 6, 7, 8	96%
				75%	1, 1, 1, 2, 2, 3, 3, 2, 3	77%
				80%	1, 2, 1, 2, 3, 3, 2, 4, 4	94%
		0.5	18	85%	1, 2, 1, 2, 3, 3, 2, 4, 4	94%
				90%	1, 2, 1, 2, 3, 3, 2, 4, 4	94%
				95%	1, 2, 1, 2, 3, 4, 4, 3, 5	98%
				75%	1, 2, 2, 5, 1, 3, 6, 3, 6, 4, 7	89%
				80%	1, 2, 2, 5, 1, 3, 6, 3, 6, 4, 7	89%
		0.1	9	85%	1, 2, 2, 5, 1, 3, 6, 3, 6, 4, 7	89%

Jackson11

				90%	1, 3, 3, 2, 1, 4, 4, 5, 6, 7, 8	99%
				95%	1, 3, 3, 2, 1, 4, 4, 5, 6, 7, 8	99%
				75%	1, 1, 3, 4, 2, 2, 4, 2, 5, 3, 5	77%
				80%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 6	94%
		0.3	13	85%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 6	94%
				90%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 6	94%
				95%	1, 4, 3, 2, 1, 4, 3, 5, 4, 6, 7	96%
				75%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 6	79%
				80%	1, 4, 3, 2, 1, 4, 4, 6, 5, 7, 7	85%
		0.5	14	85%	1, 4, 3, 2, 1, 4, 4, 6, 5, 7, 7	85%
				90%	1, 4, 2, 3, 4, 4, 4, 6, 5, 7, 8	91%
				95%	1, 4, 2, 3, 4, 4, 4, 6, 5, 7, 8	91%
				75%	1, 2, 6, 1, 4, 5, 3	95%
				80%	1, 2, 6, 1, 4, 5, 3	95%
		0.1	7	85%	1, 2, 6, 1, 4, 5, 3	95%
				90%	1, 2, 6, 1, 4, 5, 3	95%
				95%	1, 2, 6, 1, 4, 5, 3	95%
				75%	1, 1, 2, 2, 4, 5, 3	93%
				80%	1, 1, 2, 2, 4, 5, 3	93%
	Mertens7	0.3	10	85%	1, 1, 2, 2, 4, 5, 3	93%
				90%	1, 1, 2, 2, 4, 5, 3	93%
				95%	1, 3, 6, 1, 4, 5, 2	97%
				75%	1, 2, 3, 1, 2, 3, 1	80%
				80%	1, 1, 2, 4, 2, 3, 4	92%
		0.5	15	85%	1, 1, 2, 4, 2, 3, 4	92%
				90%	1, 1, 2, 2, 4, 5, 3	98%
				95%	1, 1, 2, 2, 4, 5, 3	98%
				75%	1, 2, 3, 4, 5, 7, 5, 6, 8	95%
				80%	1, 2, 3, 4, 5, 7, 5, 6, 8	95%
		0.1	7	85%	1, 2, 3, 4, 5, 7, 5, 6, 8	95%
				90%	1, 2, 3, 4, 5, 7, 5, 6, 8	95%
				95%	1, 2, 3, 4, 5, 7, 5, 6, 8	95%
				75%	1, 2, 2, 3, 5, 4, 4, 5, 6	81%
				80%	1, 2, 2, 3, 5, 4, 4, 5, 6	81%
Rep 5	Jaeschke9	0.3	10	85%	1, 2, 2, 3, 5, 4, 5, 6, 7	93%
				90%	1, 2, 2, 3, 5, 4, 5, 6, 7	93%
				95%	1, 2, 3, 4, 5, 7, 5, 6, 8	96%
				75%	1, 1, 1, 2, 2, 3, 3, 2, 3	78%
				80%	1, 2, 1, 2, 3, 3, 2, 4, 4	95%
		0.5	18	85%	1, 2, 1, 2, 3, 3, 2, 4, 4	95%
				90%	1, 2, 1, 2, 3, 3, 2, 4, 4	95%
				95%	1, 2, 1, 2, 3, 4, 4, 3, 5	98%
				75%	1, 3, 3, 2, 1, 4, 5, 4, 6, 5, 7	89%

0.1 9

Jackson11

			80%	1, 3, 3, 2, 1, 4, 5, 4, 6, 5, 7	89%
			85%	1, 3, 3, 2, 1, 4, 5, 4, 6, 5, 7	89%
			90%	1, 2, 2, 3, 4, 4, 4, 5, 6, 7, 8	100%
			95%	1, 2, 2, 3, 4, 4, 4, 5, 6, 7, 8	100%
			75%	1, 1, 3, 4, 2, 2, 4, 2, 5, 3, 5	76%
			80%	1, 2, 2, 4, 1, 3, 5, 3, 6, 5, 6	94%
	0.3	13	85%	1, 2, 2, 4, 1, 3, 5, 3, 6, 5, 6	94%
			90%	1, 2, 2, 4, 1, 3, 5, 3, 6, 5, 6	94%
			95%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 7	97%
			75%	1, 1, 2, 3, 2, 2, 5, 4, 6, 5, 6	78%
			80%	1, 4, 3, 2, 3, 4, 4, 5, 7, 6, 7	84%
	0.5	14	85%	1, 4, 2, 3, 2, 4, 4, 6, 5, 7, 8	90%
			90%	1, 4, 2, 3, 2, 4, 4, 6, 5, 7, 8	90%
			95%	1, 2, 5, 3, 2, 2, 7, 4, 8, 6, 9	91%

Table A.5: Results of the Optimization performed after the application of the RB&B to the full Datasets in all the replications.

A.1.5 Optimization of the RB&B results applied on the reduced dataset

Rep.	**Mod.**	**CV**	**c** [s]	R^* [%]	**Assignment**	**R** [%]
				75%	1, 2, 6, 1, 4, 5, 3	95%
				80%	1, 2, 6, 1, 4, 5, 3	95%
		0.1	7	85%	1, 2, 6, 1, 4, 5, 3	95%
				90%	1, 2, 6, 1, 4, 5, 3	95%
				95%	1, 2, 6, 1, 4, 5, 3	95%
				75%	1, 1, 4, 4, 2, 3, 5	93%
				80%	1, 1, 4, 4, 2, 3, 5	93%
	Mertens	0.3	10	85%	1, 1, 4, 4, 2, 3, 5	93%
				90%	1, 1, 4, 4, 2, 3, 5	93%
				95%	1, 2, 3, 1, 5, 6, 4	97%
				75%	1, 2, 3, 1, 2, 3, 1	80%
				80%	1, 1, 1, 2, 2, 4, 3	90%
		0.5	15	85%	1, 1, 1, 2, 2, 4, 3	90%
				90%	1, 1, 1, 2, 2, 4, 3	90%
				95%	1, 1, 1, 2, 2, 4, 3	90%
				75%	1, 3, 2, 4, 5, 6, 7, 7, 8	95%
				80%	1, 3, 2, 4, 5, 6, 7, 7, 8	95%
		0.1	7	85%	1, 3, 2, 4, 5, 6, 7, 7, 8	95%
				90%	1, 3, 2, 4, 5, 6, 7, 7, 8	95%
				95%	1, 3, 2, 4, 5, 6, 7, 7, 8	95%
				75%	1, 2, 2, 3, 4, 5, 5, 4, 6	81%
				80%	1, 2, 2, 3, 4, 6, 4, 5, 7	93%
Rep 1	Jaeschke	0.3	10	85%	1, 2, 2, 3, 4, 6, 4, 5, 7	93%
				90%	1, 2, 2, 3, 4, 6, 4, 5, 7	93%
				95%	1, 2, 3, 4, 5, 6, 5, 7, 8	96%
				75%	1, 1, 1, 2, 2, 3, 2, 2, 3	77%
				80%	1, 2, 1, 2, 2, 3, 4, 3, 4	92%
		0.5	18	85%	1, 2, 1, 2, 2, 3, 4, 3, 4	92%
				90%	1, 2, 1, 2, 2, 3, 4, 3, 4	92%
				95%	1, 2, 1, 2, 2, 3, 4, 3, 4	92%
				75%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 7	89%
				80%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 7	89%
		0.1	9	85%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 7	89%
				90%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 7	89%
				95%	1, 4, 3, 2, 3, 4, 4, 6, 5, 7, 8	100%
				75%	1, 1, 3, 4, 1, 2, 4, 2, 5, 3, 5	77%
				80%	1, 1, 2, 3, 1, 2, 4, 5, 4, 6, 6	93%
	Jackson	0.3	13	85%	1, 1, 2, 3, 1, 2, 4, 5, 4, 6, 6	93%
				90%	1, 1, 2, 3, 1, 2, 4, 5, 4, 6, 6	93%

				95%	1, 1, 2, 3, 1, 2, 4, 5, 4, 6, 6	93%
				75%	1, 3, 3, 2, 3, 5, 4, 5, 4, 6, 6	78%
				80%	1, 3, 3, 2, 3, 5, 4, 5, 4, 6, 6	78%
		0.5	14	85%	1, 3, 3, 2, 3, 5, 4, 5, 4, 6, 6	78%
				90%	1, 3, 3, 2, 3, 4, 4, 5, 7, 6, 7	83%
				95%	1, 3, 3, 2, 3, 4, 4, 5, 6, 7, 8	89%
				75%	1, 2, 4, 1, 3, 5, 6	95%
				80%	1, 2, 4, 1, 3, 5, 6	95%
		0.1	7	85%	1, 2, 4, 1, 3, 5, 6	95%
				90%	1, 2, 4, 1, 3, 5, 6	95%
				95%	1, 2, 4, 1, 3, 5, 6	95%
				75%	1, 1, 2, 2, 3, 5, 4	92%
				80%	1, 1, 2, 2, 3, 5, 4	92%
	Mertens	0.3	10	85%	1, 1, 2, 2, 3, 5, 4	92%
				90%	1, 1, 2, 2, 3, 5, 4	92%
				95%	1, 1, 3, 5, 2, 4, 6	95%
				75%	1, 1, 2, 1, 2, 3, 3	78%
				80%	1, 1, 2, 4, 2, 3, 4	92%
		0.5	15	85%	1, 1, 2, 4, 2, 3, 4	92%
				90%	1, 1, 3, 3, 2, 4, 5	98%
				95%	1, 1, 3, 3, 2, 4, 5	98%
				75%	1, 2, 3, 4, 5, 6, 7, 7, 8	95%
				80%	1, 2, 3, 4, 5, 6, 7, 7, 8	95%
		0.1	7	85%	1, 2, 3, 4, 5, 6, 7, 7, 8	95%
				90%	1, 2, 3, 4, 5, 6, 7, 7, 8	95%
				95%	1, 2, 3, 4, 5, 6, 7, 7, 8	95%
				75%	1, 2, 2, 3, 4, 5, 5, 4, 6	82%
				80%	1, 2, 2, 3, 4, 5, 4, 6, 7	93%
Rep 2	Jaeschke	0.3	10	85%	1, 2, 2, 3, 4, 5, 4, 6, 7	93%
				90%	1, 2, 2, 3, 4, 5, 4, 6, 7	93%
				95%	1, 2, 3, 4, 6, 5, 7, 7, 8	96%
				75%	1, 1, 1, 2, 2, 3, 3, 2, 3	79%
				80%	1, 1, 1, 2, 2, 3, 3, 2, 3	79%
		0.5	18	85%	1, 1, 1, 2, 2, 3, 3, 2, 3	79%
				90%	1, 2, 1, 2, 3, 4, 3, 3, 4	93%
				95%	1, 2, 1, 2, 3, 4, 3, 3, 4	93%
				75%	1, 1, 3, 2, 5, 3, 6, 4, 6, 5, 7	90%
				80%	1, 1, 3, 2, 5, 3, 6, 4, 6, 5, 7	90%
		0.1	9	85%	1, 1, 3, 2, 5, 3, 6, 4, 6, 5, 7	90%
				90%	1, 1, 3, 2, 5, 3, 6, 4, 6, 5, 7	90%
				95%	1, 4, 3, 2, 3, 4, 4, 5, 6, 7, 8	100%
				75%	1, 2, 1, 2, 3, 3, 4, 3, 4, 5, 5	73%
				80%	1, 2, 1, 2, 3, 3, 4, 3, 4, 5, 5	73%
	Jackson	0.3	13			

				85%	1, 1, 2, 4, 3, 2, 5, 3, 5, 6, 6	94%
				90%	1, 1, 2, 4, 3, 2, 5, 3, 5, 6, 6	94%
				95%	1, 1, 2, 4, 3, 2, 5, 3, 5, 6, 6	94%
				75%	1, 2, 2, 4, 2, 2, 5, 3, 6, 5, 6	76%
				80%	1, 1, 3, 2, 3, 4, 3, 5, 4, 6, 7	83%
		0.5	14	85%	1, 1, 3, 2, 3, 4, 3, 5, 4, 6, 7	83%
				90%	1, 3, 3, 2, 4, 4, 4, 5, 7, 6, 8	90%
				95%	1, 3, 3, 2, 4, 4, 4, 5, 7, 6, 8	90%
				75%	1, 2, 3, 1, 4, 5, 6	95%
				80%	1, 2, 3, 1, 4, 5, 6	95%
		0.1	7	85%	1, 2, 3, 1, 4, 5, 6	95%
				90%	1, 2, 3, 1, 4, 5, 6	95%
				95%	1, 2, 3, 1, 4, 5, 6	95%
				75%	1, 1, 4, 4, 2, 3, 5	93%
				80%	1, 1, 4, 4, 2, 3, 5	93%
	Mertens	0.3	10	85%	1, 1, 4, 4, 2, 3, 5	93%
				90%	1, 2, 4, 1, 3, 7, 5	97%
				95%	1, 3, 5, 1, 4, 6, 2	97%
				75%	1, 2, 3, 1, 2, 3, 1	80%
				80%	1, 2, 3, 1, 2, 3, 1	80%
		0.5	15	85%	1, 1, 3, 2, 2, 4, 3	92%
				90%	1, 1, 3, 2, 2, 4, 3	92%
				95%	1, 1, 2, 2, 4, 5, 3	98%
				75%	1, 3, 2, 4, 5, 6, 7, 7, 8	95%
				80%	1, 3, 2, 4, 5, 6, 7, 7, 8	95%
		0.1	7	85%	1, 3, 2, 4, 5, 6, 7, 7, 8	95%
				90%	1, 3, 2, 4, 5, 6, 7, 7, 8	95%
				95%	1, 3, 2, 4, 5, 6, 7, 7, 8	95%
				75%	1, 2, 2, 3, 4, 5, 6, 4, 6	79%
				80%	1, 2, 2, 3, 4, 5, 6, 4, 6	79%
Rep 3	Jaeschke	0.3	10	85%	1, 2, 2, 3, 5, 4, 7, 6, 7	89%
				90%	1, 2, 2, 3, 5, 4, 7, 6, 7	89%
				95%	1, 2, 3, 4, 5, 6, 8, 7, 8	92%
				75%	1, 1, 1, 2, 2, 2, 3, 3, 3	76%
				80%	1, 1, 1, 2, 2, 2, 3, 3, 3	76%
		0.5	18	85%	1, 1, 1, 2, 2, 2, 3, 3, 3	76%
				90%	1, 2, 1, 2, 3, 3, 3, 4, 4	94%
				95%	1, 2, 1, 2, 3, 3, 3, 4, 4	94%
				75%	1, 1, 2, 4, 3, 2, 5, 3, 6, 5, 7	89%
				80%	1, 1, 2, 4, 3, 2, 5, 3, 6, 5, 7	89%
		0.1	9	85%	1, 1, 2, 4, 3, 2, 5, 3, 6, 5, 7	89%
				90%	1, 2, 2, 3, 4, 4, 4, 5, 6, 7, 8	100%
				95%	1, 2, 2, 3, 4, 4, 4, 5, 6, 7, 8	100%

Jackson

				75%	1, 1, 5, 4, 4, 2, 5, 2, 6, 3, 6	93%
				80%	1, 1, 5, 4, 4, 2, 5, 2, 6, 3, 6	93%
		0.3	13	85%	1, 1, 5, 4, 4, 2, 5, 2, 6, 3, 6	93%
				90%	1, 1, 5, 4, 4, 2, 5, 2, 6, 3, 6	93%
				95%	1, 1, 5, 2, 2, 3, 6, 3, 6, 4, 7	95%
				75%	1, 1, 3, 2, 3, 3, 5, 4, 6, 5, 6	78%
				80%	1, 1, 3, 2, 3, 3, 5, 4, 6, 5, 6	78%
		0.5	14	85%	1, 3, 3, 2, 4, 4, 4, 5, 7, 6, 7	84%
				90%	1, 2, 2, 3, 4, 5, 4, 6, 5, 7, 8	89%
				95%	1, 2, 2, 3, 4, 5, 4, 6, 5, 7, 8	89%
				75%	1, 2, 5, 1, 3, 4, 6	94%
				80%	1, 2, 5, 1, 3, 4, 6	94%
		0.1	7	85%	1, 2, 5, 1, 3, 4, 6	94%
				90%	1, 2, 5, 1, 3, 4, 6	94%
				95%	1, 2, 5, 1, 3, 4, 6	94%
				75%	1, 1, 3, 3, 2, 4, 5	92%
				80%	1, 1, 3, 3, 2, 4, 5	92%
	Mertens	0.3	10	85%	1, 1, 3, 3, 2, 4, 5	92%
				90%	1, 1, 3, 3, 2, 4, 5	92%
				95%	1, 2, 5, 1, 4, 6, 3	96%
				75%	1, 2, 3, 1, 2, 3, 1	80%
				80%	1, 2, 3, 1, 2, 3, 1	80%
		0.5	15	85%	1, 1, 2, 1, 2, 3, 4	90%
				90%	1, 1, 2, 1, 2, 3, 4	90%
				95%	1, 1, 2, 1, 3, 5, 4	95%
				75%	1, 3, 2, 4, 5, 6, 5, 7, 8	95%
				80%	1, 3, 2, 4, 5, 6, 5, 7, 8	95%
		0.1	7	85%	1, 3, 2, 4, 5, 6, 5, 7, 8	95%
				90%	1, 3, 2, 4, 5, 6, 5, 7, 8	95%
				95%	1, 3, 2, 4, 5, 6, 5, 7, 8	95%
				75%	1, 2, 2, 3, 4, 5, 5, 4, 6	81%
				80%	1, 2, 2, 3, 4, 5, 5, 4, 6	81%
Rep 4	Jaeschke	0.3	10	85%	1, 2, 2, 3, 4, 5, 5, 4, 6	81%
				90%	1, 2, 2, 3, 5, 4, 6, 6, 7	93%
				95%	1, 2, 2, 3, 5, 4, 6, 6, 7	93%
				75%	1, 1, 1, 2, 2, 3, 2, 2, 3	75%
				80%	1, 2, 1, 2, 3, 3, 4, 4, 4	93%
		0.5	18	85%	1, 2, 1, 2, 3, 3, 4, 4, 4	93%
				90%	1, 2, 1, 2, 3, 3, 4, 4, 4	93%
				95%	1, 2, 1, 2, 3, 3, 4, 4, 4	93%
				75%	1, 3, 3, 2, 1, 4, 5, 4, 6, 5, 7	89%
				80%	1, 3, 3, 2, 1, 4, 5, 4, 6, 5, 7	89%
		0.1	9	85%	1, 3, 3, 2, 1, 4, 5, 4, 6, 5, 7	89%

Jackson

				90%	1, 3, 3, 2, 1, 4, 5, 4, 6, 5, 7	89%
				95%	1, 3, 3, 2, 1, 4, 5, 4, 6, 5, 7	89%
				75%	1, 1, 3, 4, 2, 2, 4, 2, 5, 3, 5	77%
				80%	1, 1, 3, 4, 2, 2, 4, 2, 5, 3, 5	77%
		0.3	13	85%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 6	94%
				90%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 6	94%
				95%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 6	94%
				75%	1, 2, 2, 3, 1, 5, 4, 5, 4, 6, 6	79%
				80%	1, 2, 2, 3, 1, 5, 4, 5, 4, 6, 6	79%
		0.5	14	85%	1, 2, 2, 3, 1, 5, 4, 5, 4, 6, 6	79%
				90%	1, 4, 2, 3, 2, 4, 4, 5, 6, 7, 7	85%
				95%	1, 4, 3, 2, 4, 4, 4, 6, 5, 7, 8	91%
				75%	1, 2, 3, 1, 4, 5, 6	95%
				80%	1, 2, 3, 1, 4, 5, 6	95%
		0.1	7	85%	1, 2, 3, 1, 4, 5, 6	95%
				90%	1, 2, 3, 1, 4, 5, 6	95%
				95%	1, 2, 3, 1, 4, 5, 6	95%
				75%	1, 1, 4, 4, 2, 3, 5	93%
				80%	1, 1, 4, 4, 2, 3, 5	93%
	Mertens	0.3	10	85%	1, 1, 4, 4, 2, 3, 5	93%
				90%	1, 2, 4, 1, 3, 7, 5	97%
				95%	1, 3, 5, 1, 4, 6, 2	97%
				75%	1, 2, 3, 1, 2, 3, 1	80%
				80%	1, 2, 3, 1, 2, 3, 1	80%
		0.5	15	85%	1, 1, 3, 2, 2, 4, 3	92%
				90%	1, 1, 3, 2, 2, 4, 3	92%
				95%	1, 1, 2, 2, 4, 5, 3	98%
				75%	1, 3, 2, 4, 5, 6, 7, 7, 8	95%
				80%	1, 3, 2, 4, 5, 6, 7, 7, 8	95%
		0.1	7	85%	1, 3, 2, 4, 5, 6, 7, 7, 8	95%
				90%	1, 3, 2, 4, 5, 6, 7, 7, 8	95%
				95%	1, 3, 2, 4, 5, 6, 7, 7, 8	95%
				75%	1, 2, 2, 3, 4, 5, 6, 4, 6	79%
				80%	1, 2, 2, 3, 4, 5, 6, 4, 6	79%
Rep 5	Jaeschke	0.3	10	85%	1, 2, 2, 3, 5, 4, 7, 6, 7	89%
				90%	1, 2, 2, 3, 5, 4, 7, 6, 7	89%
				95%	1, 2, 3, 4, 5, 6, 8, 7, 8	92%
				75%	1, 1, 1, 2, 2, 2, 3, 3, 3	76%
				80%	1, 2, 1, 2, 3, 3, 3, 4, 4	94%
		0.5	18	85%	1, 2, 1, 2, 3, 3, 3, 4, 4	94%
				90%	1, 2, 1, 2, 3, 3, 3, 4, 4	94%
				95%	1, 2, 1, 2, 3, 3, 3, 4, 4	94%
				75%	1, 1, 2, 4, 3, 2, 5, 3, 6, 5, 7	89%

0.1 9

Jackson

			80%	1, 1, 2, 4, 3, 2, 5, 3, 6, 5, 7	89%
			85%	1, 1, 2, 4, 3, 2, 5, 3, 6, 5, 7	89%
			90%	1, 2, 2, 3, 4, 4, 4, 5, 6, 7, 8	100%
			95%	1, 2, 2, 3, 4, 4, 4, 5, 6, 7, 8	100%
			75%	1, 1, 5, 4, 4, 2, 5, 2, 6, 3, 6	93%
			80%	1, 1, 5, 4, 4, 2, 5, 2, 6, 3, 6	93%
	0.3	13	85%	1, 1, 5, 4, 4, 2, 5, 2, 6, 3, 6	93%
			90%	1, 1, 5, 4, 4, 2, 5, 2, 6, 3, 6	93%
			95%	1, 1, 5, 2, 2, 3, 6, 3, 6, 4, 7	95%
			75%	1, 1, 3, 2, 3, 3, 5, 4, 6, 5, 6	78%
			80%	1, 1, 3, 2, 3, 3, 5, 4, 6, 5, 6	78%
	0.5	14	85%	1, 3, 3, 2, 4, 4, 4, 5, 7, 6, 7	84%
			90%	1, 2, 2, 3, 4, 5, 4, 6, 5, 7, 8	89%
			95%	1, 2, 2, 3, 4, 5, 4, 6, 5, 7, 8	89%

Table A.6: Results of the Optimization performed after the application of the RB&B to the reduced Datasets in all the replications.

A.1.6 Application of the Bootstrap Procedure

Rep.	Mod.	CV	c [s]	R^* [%]	Assignment	R [%]
				75%	1, 1, 4, 5, 2, 3, 6	92%
				80%	1, 1, 4, 5, 2, 3, 6	92%
		0.1	7	85%	1, 2, 3, 1, 4, 6, 5	95%
				90%	1, 2, 3, 1, 4, 6, 5	95%
				95%	1, 2, 3, 1, 4, 6, 5	95%
				75%	2, 2, 4, 4, 3, 6, 5	93%
				80%	1, 1, 4, 4, 2, 3, 5	93%
	Mertens	0.3	10	85%	1, 1, 4, 5, 2, 3, 6	96%
				90%	1, 2, 3, 3, 4, 6, 5	94%
				95%	1, 2, 3, 1, 4, 6, 5	97%
				75%	1, 1, 3, 3, 1, 2, 4	85%
				80%	1, 1, 1, 2, 2, 3, 4	90%
		0.5	15	85%	4, 4, 6, 5, 6, 7, 5	92%
				90%	1, 2, 5, 1, 3, 4, 1	95%
				95%	3, 3, 5, 5, 4, 7, 6	98%
				75%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				80%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
		0.1	7	85%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				90%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				95%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				75%	1, 1, 2, 3, 5, 4, 3, 6, 7	83%
				80%	1, 2, 2, 3, 5, 4, 3, 6, 7	93%
Rep 1	Jaeschke	0.3	10	85%	1, 2, 2, 3, 5, 4, 6, 6, 7	93%
				90%	1, 2, 2, 3, 5, 4, 5, 6, 7	93%
				95%	1, 2, 3, 4, 5, 7, 6, 6, 8	96%
				75%	1, 1, 1, 2, 3, 2, 2, 3, 4	90%
				80%	1, 1, 1, 2, 3, 2, 2, 3, 4	90%
		0.5	18	85%	1, 1, 1, 2, 2, 3, 2, 3, 4	92%
				90%	1, 2, 1, 2, 3, 3, 2, 4, 4	95%
				95%	1, 2, 1, 2, 4, 3, 2, 4, 5	98%
				75%	1, 2, 2, 3, 2, 5, 4, 5, 4, 6, 7	86%
				80%	1, 1, 3, 2, 3, 3, 5, 4, 6, 5, 7	86%
		0.1	9	85%	1, 1, 3, 2, 4, 3, 5, 4, 6, 5, 7	89%
				90%	1, 4, 3, 2, 1, 5, 3, 6, 4, 7, 8	95%
				95%	1, 3, 3, 2, 1, 4, 4, 5, 7, 6, 8	99%
				75%	1, 5, 2, 3, 4, 5, 4, 5, 4, 6, 6	87%
				80%	1, 5, 2, 3, 4, 5, 4, 5, 4, 6, 6	87%
	Jackson	0.3	13	85%	1, 1, 4, 2, 2, 1, 5, 3, 6, 5, 6	87%
				90%	1, 2, 2, 3, 4, 2, 4, 5, 4, 6, 6	93%
				95%	1, 1, 3, 2, 1, 3, 5, 4, 6, 5, 6	92%

				75%	1, 1, 2, 4, 1, 2, 5, 3, 6, 5, 6	77%
				80%	1, 1, 2, 3, 1, 2, 4, 5, 7, 6, 7	79%
		0.5	14	85%	1, 1, 2, 4, 1, 2, 6, 3, 6, 5, 7	83%
				90%	1, 2, 2, 3, 2, 4, 4, 5, 6, 7, 8	89%
				95%	1, 2, 2, 3, 2, 4, 4, 5, 6, 7, 8	89%
				75%	1, 1, 4, 5, 2, 3, 6	92%
				80%	1, 2, 3, 1, 5, 6, 4	95%
		0.1	7	85%	1, 2, 3, 1, 5, 6, 4	95%
				90%	1, 2, 3, 1, 5, 6, 4	95%
				95%	1, 2, 3, 1, 5, 6, 4	95%
				75%	1, 1, 4, 4, 2, 3, 5	92%
				80%	1, 1, 4, 4, 2, 3, 5	92%
	Mertens	0.3	10	85%	1, 2, 3, 1, 4, 6, 5	96%
				90%	1, 2, 3, 1, 4, 6, 5	96%
				95%	1, 2, 3, 1, 4, 6, 5	96%
				75%	1, 1, 3, 3, 1, 2, 4	85%
				80%	4, 4, 6, 5, 6, 7, 5	92%
		0.5	15	85%	4, 4, 6, 5, 6, 7, 5	92%
				90%	3, 3, 6, 4, 5, 7, 4	96%
				95%	1, 1, 4, 4, 2, 3, 5	98%
				75%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				80%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
		0.1	7	85%	1, 2, 3, 4, 5, 7, 7, 6, 8	92%
				90%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				95%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				75%	1, 2, 2, 3, 4, 6, 7, 5, 7	89%
				80%	1, 2, 2, 3, 4, 6, 7, 5, 7	89%
Rep 2	Jaeschke	0.3	10	85%	1, 2, 3, 4, 5, 7, 8, 6, 8	92%
				90%	1, 2, 3, 4, 6, 5, 7, 7, 8	96%
				95%	1, 2, 3, 4, 6, 5, 7, 7, 8	96%
				75%	1, 1, 1, 2, 2, 3, 3, 2, 3	79%
				80%	1, 1, 1, 2, 3, 2, 2, 3, 4	90%
		0.5	18	85%	1, 1, 1, 2, 3, 2, 2, 3, 4	90%
				90%	1, 1, 1, 2, 2, 3, 3, 3, 4	91%
				95%	1, 1, 2, 3, 4, 3, 3, 4, 5	95%
				75%	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	82%
				80%	1, 1, 3, 2, 4, 3, 5, 4, 6, 5, 7	88%
		0.1	9	85%	1, 1, 3, 2, 4, 3, 5, 4, 5, 6, 7	89%
				90%	1, 1, 3, 2, 5, 3, 6, 4, 7, 5, 8	94%
				95%	4, 7, 6, 5, 7, 8, 7, 9, 8, 10, 11	100%
				75%	1, 5, 2, 3, 4, 5, 4, 5, 4, 6, 6	88%
				80%	1, 1, 5, 2, 1, 2, 5, 3, 6, 4, 6	89%
	Jackson	0.3	13	85%	1, 3, 2, 3, 4, 5, 4, 5, 4, 6, 6	90%

				90%	1, 2, 2, 3, 4, 5, 4, 5, 4, 6, 6	93%
				95%	1, 1, 4, 3, 5, 2, 6, 2, 6, 5, 7	96%
				75%	1, 2, 5, 3, 1, 2, 5, 4, 7, 6, 7	82%
				80%	1, 3, 5, 2, 1, 3, 5, 4, 6, 7, 8	88%
		0.5	14	85%	1, 2, 5, 4, 2, 2, 5, 3, 6, 7, 8	89%
				90%	1, 2, 2, 3, 2, 4, 4, 5, 6, 7, 8	89%
				95%	1, 2, 2, 3, 2, 4, 4, 5, 6, 7, 8	89%
				75%	1, 1, 4, 5, 2, 3, 6	92%
				80%	1, 1, 4, 5, 2, 3, 6	92%
		0.1	7	85%	1, 2, 3, 1, 4, 6, 5	95%
				90%	1, 2, 3, 1, 4, 6, 5	95%
				95%	1, 2, 3, 1, 4, 6, 5	95%
				75%	1, 1, 5, 2, 2, 3, 4	84%
				80%	1, 1, 5, 2, 2, 3, 4	84%
	Mertens	0.3	10	85%	2, 2, 5, 5, 3, 4, 6	93%
				90%	1, 1, 5, 3, 2, 6, 4	96%
				95%	1, 3, 6, 1, 4, 5, 2	97%
				75%	1, 1, 3, 3, 1, 2, 4	85%
				80%	1, 1, 1, 2, 2, 3, 4	89%
		0.5	15	85%	1, 1, 1, 4, 2, 3, 4	90%
				90%	1, 1, 5, 1, 2, 3, 4	95%
				95%	3, 3, 5, 5, 4, 7, 6	98%
				75%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				80%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
		0.1	7	85%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				90%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				95%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				75%	1, 3, 2, 3, 4, 6, 7, 5, 7	80%
				80%	1, 2, 2, 3, 5, 4, 3, 6, 7	92%
Rep 3	Jaeschke	0.3	10	85%	1, 2, 2, 3, 5, 4, 3, 6, 7	92%
				90%	1, 2, 3, 4, 5, 7, 8, 6, 8	92%
				95%	1, 2, 3, 4, 5, 7, 8, 6, 8	92%
				75%	1, 1, 1, 2, 3, 2, 2, 3, 4	90%
				80%	1, 1, 1, 2, 2, 3, 3, 3, 4	91%
		0.5	18	85%	1, 1, 1, 2, 2, 3, 3, 3, 4	91%
				90%	1, 2, 1, 2, 3, 3, 2, 4, 4	95%
				95%	1, 2, 1, 2, 3, 3, 4, 4, 5	97%
				75%	1, 1, 3, 2, 2, 3, 5, 4, 6, 5, 7	82%
				80%	1, 1, 3, 2, 4, 3, 5, 4, 6, 5, 7	89%
		0.1	9	85%	1, 1, 2, 4, 3, 2, 5, 3, 6, 5, 7	89%
				90%	1, 4, 3, 2, 1, 5, 3, 6, 4, 7, 8	95%
				95%	1, 3, 3, 2, 1, 4, 4, 5, 6, 7, 8	99%
				75%	1, 2, 2, 4, 5, 3, 5, 3, 6, 5, 6	92%
	Jackson	0.3	13			

				80%	1, 2, 2, 4, 4, 3, 5, 3, 6, 5, 6	93%
				85%	1, 1, 3, 2, 5, 4, 5, 4, 5, 6, 6	92%
				90%	1, 1, 3, 2, 5, 4, 5, 4, 5, 6, 7	95%
				95%	1, 2, 2, 4, 1, 3, 5, 3, 5, 6, 7	97%
				75%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 6	78%
				80%	1, 1, 2, 4, 2, 1, 5, 3, 6, 7, 8	82%
		0.5	14	85%	1, 2, 2, 4, 5, 2, 5, 3, 6, 7, 8	87%
				90%	1, 4, 2, 3, 2, 4, 4, 5, 6, 7, 8	90%
				95%	1, 4, 2, 3, 2, 4, 4, 5, 6, 7, 8	90%
				75%	1, 1, 4, 5, 2, 3, 6	92%
				80%	1, 1, 4, 5, 2, 3, 6	92%
		0.1	7	85%	1, 1, 4, 5, 2, 3, 6	92%
				90%	1, 1, 4, 5, 2, 3, 6	92%
				95%	1, 2, 3, 1, 4, 6, 5	94%
				75%	2, 2, 5, 5, 3, 4, 6	92%
				80%	1, 1, 4, 4, 2, 3, 5	92%
	Mertens	0.3	10	85%	1, 1, 4, 5, 2, 3, 6	96%
				90%	1, 2, 6, 1, 4, 5, 3	96%
				95%	1, 2, 3, 1, 4, 6, 5	96%
				75%	2, 3, 4, 2, 3, 4, 2	80%
				80%	1, 1, 4, 3, 2, 4, 3	89%
		0.5	15	85%	1, 1, 1, 2, 2, 3, 4	89%
				90%	1, 1, 2, 4, 3, 5, 4	96%
				95%	1, 2, 3, 4, 4, 6, 5	96%
				75%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				80%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
		0.1	7	85%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				90%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				95%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				75%	1, 2, 2, 3, 4, 5, 3, 4, 6	81%
				80%	1, 2, 2, 3, 4, 6, 7, 5, 7	90%
Rep 4	Jaeschke	0.3	10	85%	1, 2, 2, 3, 5, 4, 3, 6, 7	92%
				90%	1, 2, 2, 3, 4, 5, 6, 6, 7	93%
				95%	1, 2, 3, 4, 5, 6, 7, 7, 8	96%
				75%	1, 1, 2, 2, 3, 4, 4, 3, 4	91%
				80%	1, 1, 2, 2, 3, 4, 4, 3, 4	91%
		0.5	18	85%	1, 1, 2, 2, 3, 4, 3, 3, 4	93%
				90%	1, 1, 2, 2, 3, 3, 4, 4, 4	93%
				95%	1, 2, 2, 3, 3, 4, 5, 4, 5	97%
				75%	1, 1, 4, 3, 4, 2, 5, 2, 5, 6, 7	84%
				80%	1, 1, 3, 2, 4, 3, 5, 4, 6, 5, 7	90%
		0.1	9	85%	1, 1, 3, 2, 4, 3, 5, 4, 6, 5, 7	90%
				90%	1, 2, 2, 4, 1, 3, 5, 3, 5, 6, 7	89%

Jackson

				95%	1, 3, 3, 2, 1, 5, 4, 6, 4, 7, 8	95%
				75%	1, 3, 2, 4, 5, 3, 5, 3, 6, 5, 6	87%
				80%	1, 2, 2, 4, 5, 3, 5, 3, 6, 5, 6	93%
		0.3	13	85%	1, 2, 2, 4, 5, 3, 5, 3, 6, 5, 6	93%
				90%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 6	94%
				95%	1, 3, 3, 2, 1, 3, 5, 4, 6, 5, 7	96%
				75%	1, 2, 2, 4, 1, 3, 5, 3, 5, 6, 6	79%
				80%	1, 2, 2, 4, 1, 2, 5, 3, 6, 5, 7	84%
		0.5	14	85%	1, 4, 3, 2, 1, 4, 4, 5, 6, 7, 8	91%
				90%	4, 7, 6, 5, 7, 7, 7, 8, 9, 10, 11	91%
				95%	4, 7, 6, 5, 7, 7, 7, 8, 9, 10, 11	91%
				75%	1, 1, 4, 5, 2, 3, 6	92%
				80%	1, 1, 4, 5, 2, 3, 6	92%
		0.1	7	85%	1, 2, 3, 1, 4, 6, 5	95%
				90%	1, 2, 3, 1, 4, 6, 5	95%
				95%	1, 2, 3, 1, 4, 6, 5	95%
				75%	1, 1, 5, 2, 2, 3, 4	84%
				80%	1, 1, 5, 2, 2, 3, 4	84%
	Mertens	0.3	10	85%	1, 1, 5, 3, 2, 6, 4	96%
				90%	1, 2, 3, 1, 4, 6, 5	97%
				95%	1, 3, 6, 1, 4, 5, 2	97%
				75%	2, 3, 4, 2, 3, 4, 2	80%
				80%	1, 1, 1, 2, 2, 3, 4	89%
		0.5	15	85%	1, 1, 1, 4, 2, 3, 4	90%
				90%	4, 4, 5, 7, 5, 6, 7	92%
				95%	1, 1, 2, 2, 4, 5, 3	98%
				75%	1, 3, 2, 4, 6, 5, 4, 7, 8	92%
				80%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
		0.1	7	85%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				90%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				95%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				75%	1, 3, 2, 3, 4, 6, 7, 5, 7	80%
				80%	1, 2, 2, 3, 5, 4, 3, 6, 7	92%
Rep 5	Jaeschke	0.3	10	85%	1, 2, 3, 4, 5, 7, 8, 6, 8	92%
				90%	1, 2, 3, 4, 5, 7, 8, 6, 8	92%
				95%	1, 2, 3, 4, 5, 7, 8, 6, 8	92%
				75%	1, 1, 1, 2, 2, 3, 2, 3, 4	91%
				80%	1, 1, 1, 2, 2, 3, 3, 3, 4	91%
		0.5	18	85%	1, 1, 1, 2, 2, 3, 3, 3, 4	91%
				90%	1, 2, 1, 2, 3, 3, 2, 4, 4	95%
				95%	1, 2, 1, 2, 3, 3, 4, 4, 5	97%
				75%	1, 1, 3, 2, 2, 4, 3, 5, 4, 6, 7	82%
				80%	1, 1, 3, 2, 4, 3, 5, 4, 6, 5, 7	89%
		0.1	9			

Jackson

			85%	1, 1, 3, 2, 4, 3, 5, 4, 6, 5, 7	89%
			90%	1, 4, 3, 2, 1, 5, 3, 6, 4, 7, 8	95%
			95%	1, 3, 3, 2, 1, 4, 4, 5, 6, 7, 8	99%
			75%	1, 2, 2, 4, 5, 3, 5, 3, 6, 5, 6	92%
			80%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 6	94%
	0.3	13	85%	1, 1, 5, 4, 3, 2, 5, 2, 6, 3, 6	93%
			90%	1, 1, 5, 4, 3, 2, 5, 2, 6, 3, 7	96%
			95%	1, 2, 3, 4, 1, 2, 5, 6, 5, 7, 8	98%
			75%	1, 1, 2, 4, 2, 2, 5, 3, 5, 6, 6	78%
			80%	1, 2, 2, 3, 1, 4, 6, 4, 6, 5, 7	84%
	0.5	14	85%	1, 4, 3, 2, 4, 5, 4, 6, 5, 7, 8	90%
			90%	1, 4, 3, 2, 3, 4, 4, 5, 6, 7, 8	90%
			95%	1, 4, 3, 2, 3, 4, 4, 5, 6, 7, 8	90%

Table A.7: Results of the experiments of the Bootstrap Procedure.

A.1.7 Application of the developed Procedure

Rep.	Mod.	CV	c [s]	R^* [%]	Assignment	R [%]
				75%	1, 1, 5, 3, 2, 4, 6	92%
				80%	1, 1, 4, 5, 2, 3, 6	92%
		0.1	7	85%	1, 1, 2, 5, 3, 4, 6	92%
				90%	1, 1, 3, 5, 2, 4, 6	92%
				95%	1, 2, 3, 1, 4, 6, 5	95%
				75%	1, 1, 2, 2, 3, 4, 5	93%
				80%	1, 1, 4, 4, 2, 3, 5	93%
	Mertens	0.3	10	85%	1, 1, 2, 2, 3, 5, 4	93%
				90%	1, 2, 4, 1, 3, 6, 5	97%
				95%	1, 2, 3, 1, 4, 5, 6	97%
				75%	1, 1, 3, 2, 2, 4, 3	92%
				80%	1, 1, 3, 2, 3, 4, 2	92%
		0.5	15	85%	1, 1, 3, 2, 3, 4, 2	92%
				90%	1, 1, 4, 2, 2, 5, 3	97%
				95%	1, 1, 3, 3, 2, 4, 5	98%
				75%	1, 3, 2, 4, 6, 5, 6, 7, 8	95%
				80%	1, 3, 2, 4, 6, 5, 6, 7, 8	95%
		0.1	7	85%	1, 3, 2, 4, 6, 5, 6, 7, 8	95%
				90%	1, 3, 2, 4, 6, 5, 6, 7, 8	95%
				95%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				75%	1, 2, 2, 3, 5, 4, 6, 6, 7	93%
				80%	1, 2, 2, 3, 5, 4, 5, 6, 7	93%
Rep 1	Jaeschke	0.3	10	85%	1, 2, 2, 3, 5, 4, 5, 6, 7	93%
				90%	1, 2, 3, 4, 6, 5, 7, 7, 8	96%
				95%	1, 2, 3, 4, 5, 6, 7, 7, 8	96%
				75%	1, 2, 1, 2, 2, 3, 4, 3, 4	92%
				80%	1, 2, 1, 2, 3, 3, 2, 4, 4	95%
		0.5	18	85%	1, 2, 1, 2, 3, 3, 2, 4, 4	95%
				90%	1, 2, 1, 2, 3, 3, 2, 4, 4	95%
				95%	1, 2, 1, 2, 4, 3, 5, 4, 5	98%
				75%	1, 1, 2, 3, 4, 2, 5, 4, 6, 5, 7	89%
				80%	1, 1, 3, 2, 4, 3, 5, 4, 6, 5, 7	89%
		0.1	9	85%	1, 2, 2, 3, 4, 4, 4, 5, 7, 6, 8	100%
				90%	1, 2, 2, 3, 4, 4, 4, 5, 7, 6, 8	100%
				95%	1, 4, 3, 2, 1, 5, 4, 6, 5, 7, 8	99%
				75%	1, 1, 2, 3, 2, 2, 4, 5, 4, 6, 6	94%
				80%	1, 1, 2, 4, 2, 2, 5, 3, 6, 5, 6	94%
	Jackson	0.3	13	85%	1, 1, 3, 2, 3, 3, 5, 4, 6, 5, 6	94%
				90%	1, 1, 2, 3, 2, 2, 5, 4, 6, 5, 6	94%
				95%	1, 2, 2, 3, 4, 4, 6, 5, 6, 7, 7	96%

				75%	1, 1, 2, 3, 2, 2, 5, 4, 6, 5, 6	78%
				80%	1, 1, 2, 4, 5, 2, 6, 3, 6, 5, 7	85%
		0.5	14	85%	1, 1, 3, 2, 5, 3, 6, 4, 6, 5, 7	85%
				90%	1, 4, 3, 2, 3, 4, 4, 5, 7, 6, 8	91%
				95%	1, 4, 3, 2, 3, 4, 4, 6, 5, 7, 8	91%
				75%	1, 2, 4, 1, 3, 5, 6	95%
				80%	1, 2, 4, 1, 3, 5, 6	95%
		0.1	7	85%	1, 2, 4, 1, 3, 5, 6	95%
				90%	1, 2, 4, 1, 3, 5, 6	95%
				95%	1, 2, 3, 1, 5, 6, 4	95%
				75%	1, 1, 2, 2, 3, 5, 4	92%
				80%	1, 1, 4, 4, 2, 3, 5	92%
	Mertens	0.3	10	85%	1, 3, 4, 1, 5, 6, 2	96%
				90%	1, 2, 3, 1, 4, 6, 5	96%
				95%	1, 2, 4, 1, 3, 5, 6	96%
				75%	1, 1, 2, 3, 2, 4, 3	92%
				80%	1, 1, 2, 4, 2, 3, 4	92%
		0.5	15	85%	1, 2, 2, 1, 3, 4, 1	91%
				90%	1, 1, 4, 4, 2, 3, 5	98%
				95%	1, 1, 3, 3, 2, 5, 4	98%
				75%	1, 3, 2, 4, 6, 5, 6, 7, 8	95%
				80%	1, 3, 2, 4, 6, 5, 6, 7, 8	95%
		0.1	7	85%	1, 3, 2, 4, 6, 5, 6, 7, 8	95%
				90%	1, 3, 2, 4, 6, 5, 6, 7, 8	95%
				95%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				75%	1, 2, 2, 3, 5, 4, 5, 6, 7	93%
				80%	1, 2, 2, 3, 4, 6, 4, 5, 7	93%
Rep 2	Jaeschke	0.3	10	85%	1, 2, 3, 4, 5, 7, 6, 6, 8	96%
				90%	1, 2, 3, 4, 5, 7, 6, 6, 8	96%
				95%	1, 3, 2, 4, 5, 7, 6, 6, 8	96%
				75%	1, 1, 1, 2, 2, 3, 3, 2, 3	79%
				80%	1, 2, 1, 2, 3, 4, 3, 3, 4	93%
		0.5	18	85%	1, 2, 1, 2, 3, 3, 4, 4, 4	93%
				90%	1, 2, 1, 2, 3, 3, 3, 4, 4	94%
				95%	1, 3, 2, 3, 4, 4, 5, 5, 5	94%
				75%	1, 1, 2, 5, 4, 2, 6, 3, 6, 4, 7	90%
				80%	1, 1, 2, 5, 4, 2, 6, 3, 6, 4, 7	90%
		0.1	9	85%	1, 4, 3, 2, 1, 5, 4, 6, 5, 7, 8	99%
				90%	1, 4, 3, 2, 3, 4, 4, 5, 7, 6, 8	100%
				95%	1, 4, 3, 2, 1, 4, 4, 5, 7, 6, 8	99%
				75%	1, 1, 5, 3, 4, 2, 5, 2, 6, 4, 6	94%
				80%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 6	95%
	Jackson	0.3	13	85%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 6	95%

				90%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 6	95%
				95%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 7	97%
				75%	1, 4, 3, 2, 1, 4, 3, 5, 4, 6, 7	84%
				80%	1, 4, 2, 3, 4, 5, 4, 6, 5, 7, 8	91%
		0.5	14	85%	1, 3, 3, 2, 4, 4, 4, 6, 5, 7, 8	90%
				90%	1, 2, 2, 3, 4, 4, 4, 5, 6, 7, 8	90%
				95%	1, 3, 3, 2, 4, 4, 4, 5, 7, 6, 8	90%
				75%	1, 2, 4, 1, 3, 5, 6	95%
				80%	1, 2, 4, 1, 3, 5, 6	95%
		0.1	7	85%	1, 2, 4, 1, 3, 5, 6	95%
				90%	1, 2, 4, 1, 3, 5, 6	95%
				95%	1, 2, 3, 1, 4, 6, 5	95%
				75%	1, 1, 2, 2, 3, 4, 5	93%
				80%	1, 1, 3, 3, 2, 4, 5	93%
	Mertens	0.3	10	85%	1, 1, 2, 2, 3, 5, 4	93%
				90%	1, 2, 5, 1, 4, 6, 3	97%
				95%	1, 3, 5, 1, 4, 6, 2	97%
				75%	1, 2, 2, 1, 3, 4, 1	91%
				80%	1, 1, 4, 2, 2, 3, 4	92%
		0.5	15	85%	1, 1, 3, 2, 2, 4, 3	92%
				90%	1, 1, 2, 2, 4, 5, 3	98%
				95%	1, 1, 2, 2, 3, 4, 5	98%
				75%	1, 3, 2, 4, 6, 5, 6, 7, 8	95%
				80%	1, 3, 2, 4, 6, 5, 6, 7, 8	95%
		0.1	7	85%	1, 3, 2, 4, 6, 5, 6, 7, 8	95%
				90%	1, 3, 2, 4, 6, 5, 6, 7, 8	95%
				95%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				75%	1, 2, 2, 3, 4, 5, 4, 6, 7	93%
				80%	1, 2, 2, 3, 4, 5, 4, 6, 7	93%
Rep 3	Jaeschke	0.3	10	85%	1, 2, 2, 3, 4, 6, 5, 5, 7	93%
				90%	1, 3, 2, 4, 5, 7, 6, 6, 8	96%
				95%	1, 2, 3, 4, 5, 7, 6, 6, 8	96%
				75%	1, 2, 1, 2, 3, 3, 3, 4, 4	94%
				80%	1, 1, 2, 2, 3, 3, 3, 4, 4	94%
		0.5	18	85%	1, 2, 1, 2, 3, 3, 2, 4, 4	95%
				90%	1, 1, 2, 2, 3, 3, 3, 4, 4	94%
				95%	1, 2, 2, 3, 4, 4, 3, 5, 5	96%
				75%	1, 1, 3, 2, 4, 3, 5, 4, 6, 5, 7	89%
				80%	1, 1, 3, 2, 4, 3, 5, 4, 6, 5, 7	89%
		0.1	9	85%	1, 1, 3, 2, 4, 3, 5, 4, 6, 5, 7	89%
				90%	1, 3, 3, 2, 4, 4, 4, 5, 7, 6, 8	100%
				95%	1, 3, 3, 2, 1, 4, 4, 5, 6, 7, 8	99%
				75%	1, 3, 3, 2, 1, 5, 4, 5, 4, 6, 6	93%
	Jackson	0.3	13			

				80%	1, 3, 3, 2, 1, 4, 5, 4, 5, 6, 6	93%
				85%	1, 4, 3, 2, 2, 4, 3, 5, 4, 6, 7	96%
				90%	1, 1, 3, 5, 4, 2, 6, 2, 6, 4, 7	96%
				95%	1, 1, 4, 5, 3, 2, 6, 2, 6, 3, 7	96%
				75%	1, 1, 3, 2, 3, 3, 5, 4, 5, 6, 6	78%
				80%	1, 1, 3, 2, 3, 3, 6, 4, 6, 5, 7	84%
		0.5	14	85%	1, 4, 2, 3, 2, 4, 4, 5, 7, 6, 8	90%
				90%	1, 4, 2, 3, 2, 4, 4, 5, 6, 7, 8	90%
				95%	1, 4, 2, 3, 2, 4, 4, 5, 7, 6, 8	90%
				75%	1, 2, 4, 1, 3, 5, 6	94%
				80%	1, 2, 4, 1, 3, 5, 6	94%
		0.1	7	85%	1, 2, 4, 1, 3, 5, 6	94%
				90%	1, 2, 4, 1, 3, 5, 6	94%
				95%	1, 2, 3, 1, 4, 6, 5	94%
				75%	1, 1, 2, 2, 3, 5, 4	92%
				80%	1, 1, 4, 4, 2, 3, 5	92%
	Mertens	0.3	10	85%	1, 2, 3, 1, 5, 6, 4	96%
				90%	1, 3, 6, 1, 4, 5, 2	96%
				95%	1, 2, 5, 1, 3, 4, 6	96%
				75%	1, 2, 3, 1, 2, 3, 1	80%
				80%	1, 1, 2, 1, 2, 3, 4	90%
		0.5	15	85%	1, 1, 2, 1, 2, 4, 3	90%
				90%	1, 1, 3, 2, 2, 4, 5	96%
				95%	1, 2, 6, 1, 3, 5, 4	98%
				75%	1, 3, 2, 4, 6, 5, 6, 7, 8	95%
				80%	1, 3, 2, 4, 6, 5, 6, 7, 8	95%
		0.1	7	85%	1, 3, 2, 4, 6, 5, 6, 7, 8	95%
				90%	1, 3, 2, 4, 6, 5, 6, 7, 8	95%
				95%	1, 3, 2, 4, 6, 5, 6, 7, 8	95%
				75%	1, 2, 2, 3, 4, 5, 5, 4, 6	81%
				80%	1, 2, 2, 3, 5, 4, 5, 6, 7	93%
Rep 4	Jaeschke	0.3	10	85%	1, 2, 2, 3, 4, 6, 5, 5, 7	93%
				90%	1, 3, 2, 4, 5, 7, 5, 6, 8	96%
				95%	1, 3, 2, 4, 5, 7, 6, 6, 8	96%
				75%	1, 2, 1, 2, 3, 3, 2, 4, 4	94%
				80%	1, 2, 1, 2, 3, 3, 2, 4, 4	94%
		0.5	18	85%	1, 2, 1, 2, 3, 3, 2, 4, 4	94%
				90%	1, 2, 1, 2, 3, 3, 2, 4, 4	94%
				95%	1, 1, 2, 2, 3, 4, 3, 3, 5	98%
				75%	1, 3, 3, 2, 1, 4, 5, 4, 5, 6, 7	89%
				80%	1, 2, 2, 4, 1, 3, 6, 3, 6, 5, 7	89%
		0.1	9	85%	1, 2, 2, 5, 1, 3, 6, 3, 6, 4, 7	89%
				90%	1, 3, 3, 2, 1, 4, 6, 4, 6, 5, 7	89%

Jackson

				95%	1, 4, 2, 3, 1, 4, 4, 5, 7, 6, 8	100%
				75%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 6	94%
				80%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 6	94%
		0.3	13	85%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 6	94%
				90%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 6	94%
				95%	1, 2, 2, 3, 4, 2, 5, 4, 6, 5, 7	96%
				75%	1, 4, 3, 2, 1, 5, 3, 5, 4, 6, 6	79%
				80%	1, 2, 2, 4, 1, 2, 5, 3, 6, 5, 7	84%
		0.5	14	85%	1, 4, 3, 2, 4, 4, 4, 5, 7, 6, 8	91%
				90%	1, 4, 2, 3, 2, 4, 4, 6, 5, 7, 8	91%
				95%	1, 4, 2, 3, 2, 4, 4, 5, 6, 7, 8	91%
				75%	1, 2, 4, 1, 3, 5, 6	95%
				80%	1, 2, 4, 1, 3, 5, 6	95%
		0.1	7	85%	1, 2, 4, 1, 3, 5, 6	95%
				90%	1, 2, 4, 1, 3, 5, 6	95%
				95%	1, 2, 3, 1, 4, 6, 5	95%
				75%	1, 1, 2, 2, 3, 4, 5	93%
				80%	1, 1, 3, 3, 2, 4, 5	93%
	Mertens	0.3	10	85%	1, 2, 6, 1, 3, 4, 5	97%
				90%	1, 2, 3, 1, 4, 5, 6	97%
				95%	1, 2, 4, 1, 3, 5, 6	97%
				75%	1, 2, 3, 1, 2, 3, 1	80%
				80%	1, 1, 3, 2, 2, 4, 3	92%
		0.5	15	85%	1, 1, 4, 2, 2, 3, 4	92%
				90%	1, 1, 2, 3, 2, 4, 3	92%
				95%	1, 1, 2, 2, 4, 5, 3	98%
				75%	1, 3, 2, 4, 6, 5, 6, 7, 8	95%
				80%	1, 3, 2, 4, 6, 5, 6, 7, 8	95%
		0.1	7	85%	1, 3, 2, 4, 6, 5, 6, 7, 8	95%
				90%	1, 3, 2, 4, 6, 5, 6, 7, 8	95%
				95%	1, 2, 3, 4, 5, 7, 6, 6, 8	95%
				75%	1, 2, 2, 3, 5, 4, 5, 6, 7	93%
				80%	1, 2, 2, 3, 4, 6, 4, 5, 7	93%
Rep 5	Jaeschke	0.3	10	85%	1, 3, 2, 4, 5, 6, 7, 7, 8	96%
				90%	1, 3, 2, 4, 6, 5, 6, 7, 8	96%
				95%	1, 2, 3, 4, 5, 7, 6, 6, 8	96%
				75%	1, 1, 2, 2, 3, 3, 3, 4, 4	94%
				80%	1, 2, 1, 2, 3, 3, 3, 4, 4	94%
		0.5	18	85%	1, 1, 2, 2, 3, 3, 3, 4, 4	94%
				90%	1, 2, 1, 2, 3, 3, 3, 4, 4	94%
				95%	1, 2, 2, 3, 4, 4, 3, 5, 5	96%
				75%	1, 1, 2, 4, 3, 2, 5, 3, 6, 5, 7	89%
				80%	1, 1, 2, 4, 3, 2, 5, 3, 6, 5, 7	89%
		0.1	9			

Jackson

			85%	1, 1, 3, 2, 4, 3, 5, 4, 6, 5, 7	89%
			90%	1, 3, 3, 2, 4, 4, 4, 5, 7, 6, 8	100%
			95%	1, 3, 3, 2, 1, 4, 4, 5, 6, 7, 8	99%
			75%	1, 1, 2, 3, 4, 2, 5, 4, 5, 6, 6	94%
			80%	1, 1, 5, 2, 4, 3, 5, 3, 6, 4, 6	93%
	0.3	13	85%	1, 2, 2, 3, 1, 4, 5, 4, 5, 6, 6	93%
			90%	1, 2, 2, 3, 1, 4, 6, 4, 6, 5, 7	97%
			95%	1, 3, 6, 2, 1, 3, 7, 4, 7, 5, 8	98%
			75%	1, 4, 3, 2, 3, 4, 3, 5, 4, 6, 7	82%
			80%	1, 2, 2, 3, 4, 4, 4, 5, 6, 7, 8	90%
	0.5	14	85%	1, 4, 2, 3, 2, 4, 4, 5, 6, 7, 8	90%
			90%	1, 4, 2, 3, 2, 4, 4, 6, 5, 7, 8	90%
			95%	1, 4, 3, 2, 3, 4, 4, 6, 5, 7, 8	90%

Table A.8: Results of the experiments of the developed methodology

Bibliography

Ağpak, K. and H. Gökçen (2007). A chance-constrained approach to stochastic line balancing problem. *European Journal of Operational Research 180*(3), 1098–1115.

Agpak, K. and H. Gökcen (2007). A chance-constrained approach to stochastic line balancing problem. *European Journal of Operational Research 180*(3), 1098 – 1115.

Agrawal, S. and M. K. Tiwari (2008). A collaborative ant colony algorithm to stochastic mixed-model u-shaped disassembly line balancing and sequencing problem. *International Journal of Production Research 46*(6), 1405–1429.

Aydogan, E. K., Y. Delice, U. Özcan, C. Gencer, and O. Bali (2019). Balancing stochastic u-lines using particle swarm optimization. *Journal of Intelligent Manufacturing 30*(1), 97–111.

Bagher, M., M. Zandieh, and H. Farsijani (2011). Balancing of stochastic u-type assembly lines: an imperialist competitive algorithm. *The International Journal of Advanced Manufacturing Technology 54*(1-4), 271–285.

Baybars, I. (1986). A survey of exact algorithms for the simple assembly line balancing problem. *Management science 32*(8), 909–932.

Baykasoglu, A. and L. Özbakir (2007). Stochastic u-line balancing using genetic algorithms. *The International Journal of Advanced Manufacturing Technology 32*(1-2), 139–147.

Becker, C. and A. Scholl (2006). A survey on problems and methods in generalized assembly line balancing. *European journal of operational research 168*(3), 694–715.

Boysen, N. and M. Fliedner (2008). A versatile algorithm for assembly line balancing. *European Journal of Operational Research 184*(1), 39–56.

Boysen, N., M. Fliedner, and A. Scholl (2007). A classification of assembly line balancing problems. *European journal of operational research 183*(2), 674–693.

Cakir, B., F. Altiparmak, and B. Dengiz (2011). Multi-objective optimization of a stochastic assembly line balancing: A hybrid simulated annealing algorithm. *Computers & Industrial Engineering 60*(3), 376–384.

Carraway, R. L. (1989). A dynamic programming approach to stochastic assembly line balancing. *Management Science 35*(4), 459–471.

Carter, J. C. and F. N. Silverman (1984). A cost-effective approach to stochastic line balancing with off-line repairs. *Journal of Operations Management 4*(2), 145–157.

Chiang, W.-C. and T. L. Urban (2006). The stochastic u-line balancing problem: A heuristic procedure. *European Journal of Operational Research 175*(3), 1767–1781.

Danford, T. (2011). Learning curve. *https://www.youtube.com/watch?v=ivJFhSL3H9g*.

Delice, Y., E. K. Aydoğan, and U. Özcan (2016). Stochastic two-sided u-type assembly line balancing: a genetic algorithm approach. *International Journal of Production Research 54*(11), 3429–3451.

Diefenbach, J. and R. Stolletz (2020). Stochastic assembly line balancing: General bounds and reliability-based branch-and-bound. *INFORMS Journal on Computing /*(/), /.

Dong, J., L. Zhang, and T. Xiao (2018). A hybrid pso/sa algorithm for bi-criteria stochastic line balancing with flexible task times and zoning constraints. *Journal of Intelligent Manufacturing 29*(4), 737–751.

Dong, J., L. Zhang, T. Xiao, and H. Mao (2014). Balancing and sequencing of stochastic mixed-model assembly u-lines to minimise the expectation of work overload time. *International Journal of Production Research 52*(24), 7529–7548.

Efron, B. (1979, 01). Bootstrap methods: Another look at the jackknife. *Ann. Statist. 7*(1), 1–26.

Efron, B. and R. Tibshirani (1994). *An Introduction to the Bootstrap*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis.

Erel, E., I. Sabuncuoglu, and H. Sekerci (2005). Stochastic assembly line balancing using beam search. *International Journal of Production Research 43*(7), 1411–1426.

Erel, E. and S. C. Sarin (1998). A survey of the assembly line balancing procedures. *Production Planning & Control 9*(5), 414–434.

Gamberini, R., A. Grassi, and A. Regattieri (2007). A multiple single-pass heuristic algorithm for the stochastic assembly line re-balancing problem. In *19th international conference on production research*, Volume 32. -.

Gamberini, R., A. Grassi, and B. Rimini (2006). A new multi-objective heuristic algorithm for solving the stochastic assembly line re-balancing problem. *International Journal of Production Economics 102*(2), 226–243.

Gen, M., Y. Tsujimura, and Y. Li (1996). Fuzzy assembly line balancing using genetic algorithms. *Computers & Industrial Engineering 31*(3-4), 631–634.

Gökçen, H. and Ö. F. Baykoç (1999). A new line remedial policy for the paced lines with stochastic task times. *International Journal of Production Economics 58*(2), 191–197.

Gu, L., S. Hennequin, A. Sava, and X. Xie (2006). Em algorithm for solving the stochastic assembly line balancing problem. *IFAC Proceedings Volumes 39*(3), 419–424.

Guerriero, F. and J. Miltenburg (2003). The stochastic uâline balancing problem. *Naval Research Logistics (NRL) 50*(1), 31–57.

Hamta, N., S. F. Ghomi, F. Jolai, and M. A. Shirazi (2013). A hybrid pso algorithm for a multi-objective assembly line balancing problem with flexible operation times, sequence-dependent setup times and learning effect. *International Journal of Production Economics 141*(1), 99–111.

Hazır, Ö. and A. Dolgui (2013). Assembly line balancing under uncertainty: Robust optimization models and exact solution method. *Computers & Industrial Engineering 65*(2), 261–267.

Held, M. and R. M. Karp (1962). A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied mathematics 10*(1), 196–210.

Henig, M. I. (1986). The shortest path problem with two objective functions. *European Journal of Operational Research 25*(2), 281–291.

Hoffmann, T. R. (1963). Assembly line balancing with a precedence matrix. *Management Science 9*(4), 551–562.

Hu, X., E. Wu, and Y. Jin (2008). A station-oriented enumerative algorithm for two-sided assembly line balancing. *European Journal of Operational Research 186*(1), 435–440.

Hua, L.-K. et al. (2012). *Popularizing mathematical methods in the People's Republic of China: Some personal experiences*, Volume 2. Springer Science & Business Media.

Kao, E. P. (1976). A preference order dynamic program for stochastic assembly line balancing. *Management Science 22*(10), 1097–1104.

Kottas, J. F. and H.-S. Lau (1973). A cost-oriented approach to stochastic line balancing1. *AIIE Transactions 5*(2), 164–171.

Kottas, J. F. and H.-S. Lau (1976). A total operating cost model for paced lines with stochastic task times. *AIIE Transactions 8*(2), 234–240.

Kottas, J. F. and H.-S. Lau (1981). A stochastic line balancing procedure. *International Journal of Production Research 19*(2), 177–193.

Krishnan, K. K., A. T. Almaktoom, and P. Udayakumar (2016). Optimisation of stochastic assembly line for balancing under high variability. *International Journal of Industrial and Systems Engineering 22*(4), 440–465.

Liu, S., H. Ong, and H. Huang (2005). A bidirectional heuristic for stochastic assembly line balancing type ii problem. *The International Journal of Advanced Manufacturing Technology 25*(1-2), 71–77.

Liu, X., M. Lei, Q. Zeng, and A. Li (2019). Integrated optimization of mixed-model assembly line balancing and buffer allocation based on operation time complexity. *Procedia CIRP 81*, 1040–1045.

Lorna, J. (2019). An introduction to the bootstrap method. *https://towardsdatascience.com/an-introduction-to-the-bootstrap-method-58bcb51b4d60*.

Lyu, J. (1997). A single-run optimization algorithm for stochastic assembly line balancing problems. *Journal of Manufacturing Systems 16*(3), 204–210.

McMullen, P. R., G. V. Frazier, et al. (1997). A heuristic for solving mixed-model line balancing problems with stochastic task durations and parallel stations. *International Journal of Production Economics 51*(3), 177–190.

McMullen, P. R. and P. Tarasewich (2003). Using ant techniques to solve the assembly line balancing problem. *Iie Transactions 35*(7), 605–617.

McMullen, P. R. and P. Tarasewich (2006). Multi-objective assembly line balancing via a modified ant colony optimization technique. *International Journal of Production Research 44*(1), 27–42.

Mendes, A. R., A. L. Ramos, A. S. Simaria, and P. M. Vilarinho (2005). Combining heuristic procedures and simulation models for balancing a pc camera assembly line. *Computers & Industrial Engineering 49*(3), 413–431.

Mitten, L. (1974). Preference order dynamic programming. *Management Science 21*(1), 43–46.

Moodie, C. and H. Young (1965). A heuristic method of assembly line balancing. *Journal of Industrial Engineering, XVI* (1).

Mooney, C., C. Mooney, C. Mooney, R. Duval, R. Duvall, M. Lewis-Beck, and i. Sage Publications (1993). *Bootstrapping: A Nonparametric Approach to Statistical Inference*. Bootstrapping: A Nonparametric Approach to Statstical Inference. SAGE Publications.

Nkasu, M. M. and K. H. Leung (1995). Computer-integrated manufacturing assembly system design. *Integrated Manufacturing Systems*.

Özcan, U. (2010). Balancing stochastic two-sided assembly lines: A chance-constrained, piecewise-linear, mixed integer program and a simulated annealing algorithm. *European Journal of Operational Research 205*(1), 81–97.

Pesavento, M. (2018). A bidirectional heuristic for stochastic assembly line balancing.

Reeve, N. R. and W. H. Thomas (1973). Balancing stochastic assembly lines. *AIIE Transactions 5*(3), 223–229.

Roy, D. et al. (2011). Optimum assembly line balancing: A stochastic programming approach. *International Journal of Industrial Engineering Computations 2*(2), 329–336.

Samouei, P., V. Khodakaramia, and P. Fattahib (2017). Mixed-model assembly line balancing with reliability. *INTERNATIONAL JOURNAL OF ENGINEERING 30*(3), 412–424.

Sarin, S. C. and E. Erel (1990). Development of cost model for the single-model stochastic assembly line balancing problem. *THE INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH 28*(7), 1305–1316.

Sarin, S. C., E. Erel, and E. M. Dar-El (1999). A methodology for solving single-model, stochastic assembly line balancing problem. *Omega 27*(5), 525–535.

Scholl, A. (1993). Data of assembly line balancing problems. darmstadt technical university, department of business administration, economics and law. *Institute for Business Studies (BWL)*.

Scholl, A. and C. Becker (2006a). State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research 168*(3), 666–693.

Scholl, A. and C. Becker (2006b). State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research 168*(3), 666–693.

Shin, D. and H. Min (1991). Uniform assembly line balancing with stochastic task times in just-in-time manufacturing. *International Journal of Operations & Production Management*.

Shttjb, A. (1984). The effect of incompletion cost on line balancing with multiple manning of work stations. *THE INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH 22*(2), 235–245.

Sivasankaran, P. and P. Shahabudeen (2014). Literature review of assembly line balancing problems. *The International Journal of Advanced Manufacturing Technology 73*(9-12), 1665–1694.

Sniedovich, M. (1981). Analysis of a preference order assembly line problem. *Management Science 27*(9), 1067–1080.

Sphicas, G. P. and F. N. Silverman (1976). Deterministic equivalents for stochastic assembly line balancing. *AIIE Transactions 8*(2), 280–282.

Suresh, G. and S. Sahu (1994). Stochastic assembly line balancing using simulated annealing. *The International Journal of Production Research 32*(8), 1801–1810.

Suresh, G., V. Vinod, and S. Sahu (1996). A genetic algorithm for assembly line balancing. *Production Planning & Control 7*(1), 38–46.

Suri, R. and Y. T. Leung (1989). Single run optimization of discrete event simulations - an empirical study using the m/m/l queue. *IIE transactions 21*(1), 35–49.

Tsujimura, Y., M. Gen, and E. Kubota (1995). Solving fuzzy assembly-line balancing problem with genetic algorithms. *Computers & Industrial Engineering 29*(1-4), 543–547.

Urban, T. L. and W.-C. Chiang (2006). An optimal piecewise-linear program for the u-line balancing problem with stochastic task times. *European Journal of Operational Research 168*(3), 771–782.

Van Hop, N. (2006). A heuristic solution for fuzzy mixed-model line balancing problem. *European Journal of Operational Research 168*(3), 798–810.

Vrat, P. and A. Virani (1976). A cost model for optimal mix of balanced stochastic assembly line and the modular assembly system for a customer oriented production system. *The International Journal of Production Research 14*(4), 445–463.

Wilson, J. (1986). Formulation of a problem involving assembly lines with multiple manning of work stations. *International Journal of Production Research 24*(1), 59–63.

Wu, E.-F., Y. Jin, J.-S. Bao, and X.-F. Hu (2008). A branch-and-bound algorithm for two-sided assembly line balancing. *The International Journal of Advanced Manufacturing Technology 39*(9-10), 1009–1015.

Xu, W. and T. Xiao (2008). Mixed model assembly line balancing problem with fuzzy operation times and drifting operations. In *2008 Winter Simulation Conference*, pp. 1752–1760. IEEE.

Zacharia, P. T. and A. C. Nearchou (2012). Multi-objective fuzzy assembly line balancing using genetic algorithms. *Journal of Intelligent Manufacturing 23*(3), 615–627.

Zacharia, P. T. and A. C. Nearchou (2013). A meta-heuristic algorithm for the fuzzy assembly line balancing type-e problem. *Computers & Operations Research 40*(12), 3033–3044.

Zhuo, C., Y. Xu, and Y. Tan (2011). A research of the stochastic u-type assembly line balancing problem. *IEEE 18th International Conference on Industrial Engineering and Engineering Management. IEEE*, 875–878.