Simulation-Based Optimization for Production Planning:
Integrating Meta-Heuristics, Simulation and Exact Techniques to Address the Uncertainty and Complexity of Manufacturing Systems

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Humanities

2016

Juan Esteban Diaz Leiva
Alliance Manchester Business School
Contents

List of Abbreviations 13

Abstract 16

Declaration 17

Copyright 18

Dedication 20

Acknowledgements 21

1 Introduction 22
   1.1 Research Context 22
   1.2 Research Aim 24
   1.3 Contributions of the Thesis 25
   1.4 Structure of the Thesis 26
   1.5 Publications Resulting from the Thesis 28

2 Description of the Manufacturing System Analysed 30
   2.1 Batch Manufacturing System 31
   2.2 Production Planning Challenges 41
   2.3 Production Planning Problem 42
## Problem Description

5.4 Problem Description .................................................. 87
  5.4.1 New Variants of the KP Problem: $d$-MBKAR and $d$-MBKARS 89

5.5 Simulation-Based Optimization Model ............................... 93
  5.5.1 Optimization Model ................................................. 96
  5.5.2 Initialization Operators ............................................ 97
  5.5.3 Repair Operator ..................................................... 101

5.6 Benchmark Analysis .................................................. 101
  5.6.1 Results ............................................................... 103

5.7 Conclusion ............................................................. 108

5.8 Limitations and Future Research .................................... 109

## A Matheuristic Optimizer to Address the $d$-MBKARS Problem

6 A Matheuristic Optimizer to Address the $d$-MBKARS Problem 115
  6.1 Abstract ............................................................... 115
  6.2 Introduction ......................................................... 116
  6.3 SBOMat Approach ...................................................... 118
  6.4 Benchmark Analysis .................................................. 120
  6.5 Results ................................................................. 120
  6.6 Conclusion ............................................................. 121

## Simulating Realistic Features of Manufacturing Systems

7 Simulating Realistic Features of Manufacturing Systems 125
  7.1 New Extensions ....................................................... 126
  7.2 Conclusion ............................................................. 135

## Evolutionary Robust Optimization (Manuscript 4)

8 Evolutionary Robust Optimization (Manuscript 4) 136
  8.1 Abstract ............................................................... 136
  8.2 Introduction ........................................................... 137
    8.2.1 Robust Optimization ............................................. 139
    8.2.2 Evolutionary Robust Optimization ............................ 140
8.2.3 Contributions .............................................. 142
8.3 Simulation-Based Optimization Model .......................... 144
  8.3.1 Optimization Model ....................................... 145
  8.3.2 Initialization Operator .................................... 150
8.4 Benchmark Analysis ............................................. 152
8.5 Performance Assessment ......................................... 155
8.6 Results ............................................................. 157
8.7 Conclusion .......................................................... 167

9 Conclusion ............................................................. 180

Word Count: 47558
List of Tables

2.1 Products manufactured in the different production lines. 32
2.2 Requirements of products manufactured in the system that are used as raw materials. 33
2.3 Product lot characteristics and demand level. 36
2.4 Requirements of labour and production lines per product lot. 38
2.5 Conservative probabilities of failure and average detention time per production line. 40
3.1 Number of product lots to be manufactured in a specific production line. 57
3.2 Demand, consolidated production plan per product and actual production achieved. 58
4.1 Parameters used for baseline, implicit averaging, explicit averaging and hybrid strategies. 68
4.2 $p_l$ per problem instance and PDFs specifications to model $\Lambda_l$. 69
4.3 Descriptive statistics of average profits and average computation times per strategy in problem instance 1. 71
4.4 Descriptive statistics of average profits and average computation times per strategy in problem instance 2. 71
4.5 Values for Mann-Whitney U statistic obtained in problem instance 1. 72
4.6 Values for Mann-Whitney U statistic obtained in problem instance 2. 72
8.6 Values for Mann-Whitney U statistic obtained from pairwise comparisons of hypervolume values of all single and multi-objective models in the objective space of average profit vs. minimum profit in problem instance 2.
List of Figures

3.1 Order processing subsystem for production line l. . . . . . . . . . . . . 50
3.2 Production subsystem for production line l. . . . . . . . . . . . . . . 52
3.3 Repair service centre of production line l. . . . . . . . . . . . . . . . 53
3.4 Best, mean and worst fitness value of the population at each iteration. 56

4.1 CDFs of average profit values obtained with production plans generated under the four different strategies in problem instance 1. . . . . . 73
4.2 CDFs of average profit values obtained with production plans generated under the four different strategies in problem instance 2. . . . . . 74

5.1 SBO model. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 94
5.2 CDFs of average profit values generated with solutions given by SBO1, SBO2, SBO3, CCP and ILP in problem instance 1 (low uncertainty). 106
5.3 CDFs of average profit values generated with solutions given by SBO1, SBO2, SBO3, CCP and ILP in problem instance 2 (medium uncertainty). 107
5.4 CDFs of average profit values generated with solutions given by SBO1, SBO2, SBO3, CCP and ILP in problem instance 3 (high uncertainty). 107
5.5 CDFs of average profit values generated with solutions given by SBO1, SBO2, SBO3 and SBO3x2 in problem instance 1 (low uncertainty). 112
5.6 CDFs of average profit values generated with solutions given by SBO1, SBO2, SBO3 and SBO3x2 in problem instance 2 (medium uncertainty). 112
5.7 CDFs of average profit values generated with solutions given by SBO1, SBO2, SBO3 and SBO3x2 in problem instance 3 (high uncertainty). 113

6.1 SBOMat approach. 119

6.2 CDFs of average profit values generated with solutions given by SBO1, SBO2 and SBOMat in problem instance 1 (low uncertainty). 122

6.3 CDFs of average profit values generated with solutions given by SBO1, SBO2 and SBOMat in problem instance 2 (medium uncertainty). 122

6.4 CDFs of average profit values generated with solutions given by SBO1, SBO2 and SBOMat in problem instance 3 (high uncertainty). 123

7.1 Order initialization sub-system of production line 2. 127

7.2 Final product sub-system of production line 2. 128

7.3 Manufacturing sub-system of production line 2. 129

7.4 Markov chain. 131

7.5 Repair sub-system of production line 2. 134

8.1 CDFs of hypervolume values obtained with multi-objective models (MSBO-1A, MSBO-2A, MSBO-1B, MSBO-2B) in the objective space of average profit vs. standard deviation of profit in problem instance 1. 158

8.2 CDFs of hypervolume values obtained with multi-objective models (MSBO-1A, MSBO-2A, MSBO-1B, MSBO-2B) in the objective space of average profit vs. minimum profit in problem instance 1. 159

8.3 CDFs of hypervolume values obtained with the best multi-objective model (MSBO-1B) and all single-objective models (SSBO-1A, SSBO-2A, SSBO-1B and SSBO-2B) in the objective space of average profit vs. standard deviation of profit. 160
8.4 CDFs of hypervolume values obtained with the best multi-objective model (MSBO-1B) and all single-objective models (SSBO-1A, SSBO-2A, SSBO-1B and SSBO-2B) in the objective space of average profit vs. minimum profit. .............................................. 161

8.5 Median (dashed line) and best (continuous line) attainment surfaces of the best multi-objective model and all single-objective models in the objective space of average profit vs. standard deviation of profit in problem instance 1. ............................................................. 168

8.5 Median (dashed line) and best (continuous line) attainment surfaces of the best multi-objective model and all single-objective models in the objective space of average profit vs. standard deviation of profit in problem instance 1. ............................................................. 169

8.6 Median (dashed line) and best (continuous line) attainment surfaces of the best multi-objective model and all single-objective models in the objective space of average profit vs. standard deviation of profit in problem instance 1. ............................................................. 170

8.6 Median (dashed line) and best (continuous line) attainment surfaces of the best multi-objective model and all single-objective models in the objective space of average profit vs. minimum profit in problem instance 1. ............................................................. 171

8.7 Median (dashed line) and best (continuous line) attainment surfaces of the best multi-objective model and all single-objective models in the objective space of average profit vs. standard deviation of profit in problem instance 1. ............................................................. 172
8.7 Median (dashed line) and best (continuous line) attainment surfaces of the best multi-objective model and all single-objective models in the objective space of average profit vs. standard deviation of profit in problem instance 2. ........................................................................ 173

8.8 Median (dashed line) and best (continuous line) attainment surfaces of the best multi-objective model and all single-objective models in the objective space of average profit vs. standard deviation of profit in problem instance 1. ........................................................................ 174

8.8 Median (dashed line) and best (continuous line) attainment surfaces of the best multi-objective model and all single-objective models in the objective space of average profit vs. minimum profit in problem instance 2. ........................................................................ 175

8.9 CDFs of hypervolume values obtained with multi-objective models (MSBO-1A, MSBO-2A, MSBO-1B, MSBO-2B) in the objective space of average profit vs. standard deviation of profit in problem instance 2. 177

8.10 CDFs of hypervolume values obtained with multi-objective models (MSBO-1A, MSBO-2A, MSBO-1B, MSBO-2B) in the objective space of average profit vs. minimum profit in problem instance 2. . . . . . . 177
List of Abbreviations

BKP: bounded knapsack problem
BS: baseline strategy
CCP: chance-constrained programming
CDF: cumulative distribution function
d: day
DES: discrete-event simulation
d-MBKAR: multidimensional multiple bounded knapsack problem with assignment restrictions
d-MBKARS: stochastic version of the multidimensional multiple bounded knapsack problem with assignment restrictions
d-KP: d-dimensional or multidimensional knapsack problem
EA: evolutionary algorithm
EAF: empirical attainment function
EF-OSI/DH-DR1S: evaluation function - simulation-based iterations/discrete heuristic-different - realizations for each solution
e.g.: for example
EMO: evolutionary multi-objective
ES: explicit averaging strategy
ESO: evolutionary single-objective
etc.: et cetera
**FIFO:** first-in first-out

**GA:** genetic algorithm

**h:** hour

**HS:** hybrid strategy

**i.e.:** it is, that is

**ILP:** integer linear programming

**IS:** implicit averaging strategy

**KP:** knapsack problem

**LP:** linear programming

**M-d-MBKARS:** stochastic version of the multi-objective multidimensional multiple bounded knapsack problem with assignment restrictions

**MEM:** multi-evaluation mode

**MEM-W:** multi-evaluation mode where the fitness of an individual is the worst fitness value

**MILP:** mixed integer linear programming

**MIP:** mixed integer programming

**MKAR:** multiple knapsack problem with assignment restrictions

**MKP:** multiple knapsack problem

**MPMP:** multi-period multi-product

**MSBO:** multi-objective simulation-based optimization

**NSGA:** non-dominated sorting genetic algorithm

**PDF:** probability density function

**PMF:** probability mass function

**PPC:** production planning committee

**RHS:** right-hand side

**s:** second

**SBO:** simulation-based optimization
**SEM**: single-evaluation mode

**SSBO**: single-objective simulation-based optimization

**TSP**: travelling salesman problem

**USD**: United States Dollar

**vs.**: *versus*
Abstract

Name of the University: The University of Manchester

Candidate's full name: Juan Esteban Diaz Leiva

Degree title: Doctor of Philosophy

Thesis title: Simulation-Based Optimization for Production Planning: Integrating Meta-Heuristics, Simulation and Exact Techniques to Address the Uncertainty and Complexity of Manufacturing Systems

Date: March 1, 2016

Abstract: This doctoral thesis investigates the application of simulation-based optimization (SBO) as an alternative to conventional optimization techniques when the inherent uncertainty and complex features of real manufacturing systems need to be considered. Inspired by a real-world production planning setting, we provide a general formulation of the situation as an extended knapsack problem. We proceed by proposing a solution approach based on single and multi-objective SBO models, which use simulation to capture the uncertainty and complexity of the manufacturing system and employ meta-heuristic optimizers to search for near-optimal solutions. Moreover, we consider the design of matheuristic approaches that combine the advantages of population-based meta-heuristics with mathematical programming techniques. More specifically, we consider the integration of mathematical programming techniques during the initialization stage of the single and multi-objective approaches as well as during the actual search process. Using data collected from a manufacturing company, we provide evidence for the advantages of our approaches over conventional methods (integer linear programming and chance-constrained programming) and highlight the synergies resulting from the combination of simulation, meta-heuristics and mathematical programming methods. In the context of the same real-world problem, we also analyse different single and multi-objective SBO models for robust optimization. We demonstrate that the choice of robustness measure and the sample size used during fitness evaluation are crucial considerations in designing an effective multi-objective model.
Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.
Copyright

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the “Copyright”) and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the thesis, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property
and/or Reproductions described in it may take place is available in the University IP Policy (see http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=487), in any relevant Thesis restriction declarations deposited in the University Library, The University Library’s regulations (please see http://www.manchester.ac.uk/library/aboutus/regulations) and in The University’s policy on presentation of Theses
Dedication

To my wife, parents and sister.
You are all I am, all I have and much more than I could ever possibly wish for.
Acknowledgements

First of all, I would like to thank God for giving me all what was required to achieve this goal.

I would also like to thank Professor Dong-Ling Xu and Dr. Julia Handl for their outstanding role as my supervisors. Thank you for all the support you gave me throughout the progress of this work, thanks for providing me with critical and constructing feedback and also for motivating me by recognizing my merits and listening to my ideas.

I am also extremely grateful to my examiners, Professor Theodor Stewart and Professor Juergen Branke, for their helpful comments and feedback regarding this work.

Thank you very much to my wonderful wife, parents and sister for all their love, support and understanding.

I would also like to acknowledge the assistance given by the IT Research Infrastructure Team and the use of the Computational Shared Facility at The University of Manchester.

Finally, I am very much obliged to the Secretariat for Higher Education, Science and Technology of Ecuador, who has supported me with a doctoral scholarship.
Chapter 1

Introduction

1.1 Research Context

Increasing market pressures have forced manufacturers to rethink the way production planning is being undertaken (Karmarkar and Rajaram, 2012). This in turn has enlarged the demand for decision tools at the strategic, tactical and operational level (Almeider et al., 2009). This study is focussed on production planning at the tactical level, where three serious limitations preclude manufacturers coping with contemporary challenges.

First, the multi-objective nature of production planning is rarely considered at the tactical level (Díaz-Madroñero et al., 2014). At this level, production planning is usually a profit driven activity, where the sole objective of maximizing profit governs production decisions. However, if the relevant objectives are not considered at this stage of the planning process, production decisions may undermine the company’s strategy instead of supporting it. For example, consider a manufacturer that has to guarantee to its clients, by means of contractual agreements, that products will be delivered in the agreed quantity and not later than the date specified. For this manufacturer the

1usually involves planning horizons between one month and two years (Díaz-Madroñero et al., 2014)
robustness of a production plan is an important additional criterion that needs to be considered because penalties due to breach of contracts might seriously undermine company’s profitability and reputation.

Second, production decisions are typically determined at an aggregated level (Wołosewicz et al., 2015). For instance, production quantities are usually specified for groups or families of products (Akhoondi and Lotfi, 2016), although the different products clustered into a group or family have different production requirements. This aggregation principle is applied in hierarchical planning to keep low levels of complexity at the tactical level. However, it delays until the end of the planning process (short-term planning) important considerations that should be taken into account from the very beginning such as product-specific machine and labour requirements.

Third, production plans, at the tactical level, are developed usually based on unrealistic assumptions. In general, system constraints are not considered in detail at the tactical level and are assumed to be deterministic, since in most of the cases the reliability of the manufacturing system is not taken into account (Stadtler et al., 2011, p. 125). A clear example of this is that at the tactical level, delays caused by machine failures are typically ignored (Khakdaman et al., 2014). It is not until those delays are evident, that shop floor control applies corrective actions (e.g. safety stocks or rolling schedules (Stadtler et al., 2011, p. 111)) to alleviate their negative effect on the system performance. In the best of the cases, uncertainty\(^2\) is considered at this planning level by incorporating into the problem different scenarios (see Khakdaman et al. (2014) for a recent example).

If tactical production plans are specified under the limitations mentioned above, it is almost imminent that corrective actions will be needed during their implementation at the operational level, since otherwise finding feasible solutions to subsequent manufacturing problems, such as lot-sizing, material requirements planning, master

---

\(^2\)several definitions of uncertainty are given in Stewart (2005)
production scheduling, short-range scheduling among others, might be impossible due to the hierarchical nature of the production planning process.

Deviations from the original plan and the implementation of corrective actions may have serious consequences not only on the company’s profitability, but also on its image and reputation. Therefore, there is a need for a methodology able to simultaneously consider the multi-objective nature of production planning and accurately capture the complexity and uncertainty intrinsic to manufacturing systems from the very beginning of the planning process.

1.2 Research Aim

Uncertainty and complexity are two inherent characteristics of real-world problems (Al-Aomar, 2006); however, their accurate incorporation into optimization problems is a challenge faced by conventional approaches such as stochastic programming, fuzzy programming, stochastic dynamic programming, among others (Figueira and Almada-Lobo, 2014).

Those conventional approaches usually rely on unrealistic assumptions to try to incorporate into the problem complex features for which no analytical expressions exist (Chu et al., 2015; Figueira and Almada-Lobo, 2014; Lee et al., 2008). The use of unrealistic assumptions restricts the real applicability of solutions obtained through those approaches because they may provide the right solutions, but for the wrong problems, and thus may lead to serious consequences, ranging from monetary losses to customer dissatisfaction. Despite this difficulty, many real-word problems are usually addressed through the approaches mentioned above, a clear example is production planning (especially at the tactical level).

Therefore, the primary purpose of this thesis is to investigate different approaches able to deal with uncertainty and complex features in optimization problems. We use
throughout this work a real-world production planning problem simply as a medium to test the approaches proposed. It is important to mention that our intention is not to fully resolve this specific problem, but to derive general insights from its analysis.

1.3 Contributions of the Thesis

This thesis contributes to the field of single and multi-objective optimization under uncertainty, especially in the context of simulation-based optimization (SBO). The specific contributions are:

- The development of a SBO approach able to accurately incorporate uncertainty and complex features of real systems into optimization problems and return near-optimal solutions that outperform exact optimization techniques (an initial attempt in Chapter 3, a single-objective approach in Chapter 5 and a multi-objective approach in Chapter 8).

- The generation of synergies through the combination of simulation and optimization approaches (Chapters 5 and 8).

- The formulation and analysis of new variants (one deterministic and two stochastic) of the classical knapsack problem that generalize a real-world production planning problem (Chapters 5 and 8).

- The development of a noise handling strategy for evolutionary single objective optimization (Chapter 4).

- The investigation and comparison of different noise handling strategies for evolutionary single and multi-objective optimization, in the context of real-world problems addressed via SBO (Chapters 4 and 8).
• The development of initialization operators that significantly improve the performance of evolutionary single and multi-objective algorithms (Chapters 5 and 8).

• The development of a repair operator able to fix the chromosome of some infeasible solutions created by evolutionary algorithms (EA) (Chapters 5).

• The investigation and comparison of different evolutionary single and multi-objective formulations to find robust solutions (Chapters 4 and 8).

• The investigation of the effect that noise has on the optimization performance of evolutionary single and multi-objective algorithms when employed to search for robust solutions (Chapter 8).

In a wider context, this thesis also makes the following contributions to the field of combinatorial optimization, simulation and matheuristics, which are approaches that combine meta-heuristics with mathematical programming techniques (Villegas et al., 2013; Boschetti et al., 2009):

• The development of a simulation model able to capture complex features of manufacturing systems by incorporating Markov chains into a discrete-event simulation (DES) model (Chapter 8).

• The generation of synergies through the combination of mathematical programming and meta-heuristic approaches (Chapters 5 and 6).

1.4 Structure of the Thesis

The remainder of this thesis is organized as follows:

Chapter 2 provides details about the main features of a real manufacturing company and the challenges it faces in its production planning at the tactical level.
Chapter 3 presents the article “Simulation-Based GA Optimization for Production Planning” (Diaz and Handl, 2014), where a SBO approach capable of accounting for the uncertainty intrinsic to the manufacturing systems is proposed. The uncertainty considered derives from the occurrence of failures in the system. It is an initial attempt to obtain an adequate formulation for the problem analysed and to present a SBO approach that employs DES and a genetic algorithm (GA) as an instrument able to support decision making in the area of production planning.

Chapter 4 presents the article “Implicit and Explicit Averaging Strategies for Simulation-Based Optimization of a Real-World Production Planning Problem” (Diaz and Handl, 2015), where we explore the impact of implicit and explicit averaging as noise handling strategies on the optimization performance of a more elaborated and refined version of the SBO model presented in Chapter 3. We also propose a hybrid approach that uses implicit averaging during the evolutionary process, but applies explicit averaging to refine fitness estimates during the final solution selection.

In Chapter 5, we present the article “Integrating Meta-heuristics, Simulation and Exact Techniques for Production Planning of Failure-Prone Manufacturing Systems” (Diaz et al., 2015a), where we are concerned with the optimization of production plans in batch manufacturing systems that include uncertainties. Specifically, we consider the occurrence of failures in production lines and the uncertainty around repair times, and how these impact on the optimality of production plans. We also provide a general formulation for the problem analysed, in the form of an extended knapsack problem, in order to highlight the wider applicability of the SBO approach proposed in that chapter. In this approach, the production system is modelled via DES and a GA is used as optimizer. Moreover, we introduce two specialized initialization operators in order to boost the performance of our GA, this could be seen as a matheuristic approach, and propose a repair operator which tries to fix unfeasible solutions generated during the optimization. Using data from a real-world production system, we benchmark
our model against integer linear programming (ILP), chance-constrained programming (CCP) and SBO without specialized initialization operators.

In Chapter 6, we further explore the potential of matheuristic approaches by investigating the performance of an optimizer that combines a GA with ILP in order to remove from the search space all unfeasible solutions.

In Chapter 7, we present a simulation approach that employs a combination of DES and Markov chains to accurately model complex features of real-world manufacturing systems such as different types of production line failure, near-perfect and imperfect repairs and deterioration caused by previous failures.

In Chapter 8, we present the article “Single and Multi-objective Formulations for Robust Optimization - Sensitivity to Sample Size and Choice of Robustness Measure” (Diaz et al., 2015b), where we explore different evolutionary single-objective and evolutionary multi-objective formulations for robust optimization. We also analyse how the choice of robustness measure and the level of noise in fitness evaluations affect the optimization performance of the different formulations.

Finally, conclusions derived from this thesis and future research directions are presented in Chapter 9.

1.5 Publications Resulting from the Thesis

Refereed journal papers


Refereed conference papers


Refereed conference abstract


Chapter 2

Description of the Manufacturing System Analysed

In this chapter we analyse the manufacturing system of a real company, in order to understand the production planning challenges it faces. The collaboration with this company started in November 2013, and during one and a half years of data collection and continuous communication with operators, members of the executive board and main share holders, deep insights about the main difficulties faced by this company could be gathered.

The people involved in the project did not evidence a sufficient level of commitment at the very beginning, although the executive board agreed to participate in this project. It was only after an intensive training program, that trust and commitment were obtained. The main objective of the training (provided to the people involved in the project) was to create awareness of the usefulness and practical applications of the project. Therefore, this training covered methodologies such as linear programming (LP), DES, evolutionary optimization, among others, but most importantly also dealt with how to deploy in their quotidian planning activities the models developed. A final meeting was held with the company’s chief executive officer and operations manager,
where the progress of the project was presented and models were handed over.

This collaboration contributed to this work in two ways. First, it provided very important insights about difficulties faced by a real manufacturing company, which motivated the analysis of issues not addressed in the existing literature. Second, it was the source of real information about a complex manufacturing system and its features, which was used to develop the models presented throughout this work.

2.1 Batch Manufacturing System

In this section we provide specific details about the manufacturing system analysed and the production planning problem it faces. The company specializes on the manufacturing of cleaning products. The type of processes undertaken in this manufacturing system are batch processes (Slack et al., 2013, p. 103). In this type of process, products are manufactured in batches, also known as lots, by following a sequence of operations with short production runs (Evans, 1990, p. 312). This means that several items of the same product are manufactured simultaneously. Unlike continuous processes, where a continuous output of finished products is obtained, in batch processes finished products are only obtained at the end of the manufacturing process (at discrete times).

This is a product oriented system, in the sense that special-purpose equipment is grouped together to form a dedicated production line, where a product or a set of products can be fully manufactured. Within a production line, such equipment is arranged according to the sequence of operations that needs to be performed. This manufacturing system consists of seven independent production lines and its product portfolio includes 31 products, some of which can be manufactured in several production lines. Table 2.1 presents the products that can be manufactured across the different production lines.
<table>
<thead>
<tr>
<th>Production line</th>
<th>Number of Products</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>18, 19, 24</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18, 20, 21, 22, 23</td>
</tr>
<tr>
<td>4</td>
<td>13</td>
<td>1, 2, 7, 8, 9, 12, 13, 14, 15, 16, 18, 26, 27</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>28, 29</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>31</td>
</tr>
</tbody>
</table>

Table 2.1: Products manufactured in the different production lines.
Furthermore, some products manufactured in this system are required as raw materials during the manufacturing of other products. Those products used as raw materials are here referred to as sub-products. Sub-products can be stored for future production periods in case they are not fully used during the current period. Table 2.2 presents the products that need sub-products and their corresponding requirements per lot.

The marginal (per lot) contribution to profit has been computed for every product by considering its standard cost per lot and its price per lot. Here, it is assumed that only standard costs are derived from the manufacturing of sub-products, since they don’t directly generate any revenue (they are not sold). In this sense, the monetary contribution of a given sub-product is only realized once the corresponding final product is sold. The standard costs of products that use sub-products exclude the corresponding standard costs of sub-products according to the proportions given in Table 2.2. An alternative approach is to reward the manufacturing of sub-products and discount that reward from the marginal profit of the corresponding final products. This approach however, may generate production plans where sub-products are manufactured without considering their future usage. For this reason and because assigning those rewards is not an easy task, as some sub-products are required by different final products, the former option is employed here. Table 2.3 presents the size of the technical lot, standard cost, price and profit per product lot and the demand forecast per product.

According to specifications given by the company, a set-up is required before the

<table>
<thead>
<tr>
<th>Product</th>
<th>Requirement per lot</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>16403.27 kg of product 25</td>
</tr>
<tr>
<td>28</td>
<td>2536.22 kg of product 26</td>
</tr>
<tr>
<td>29</td>
<td>2538.56 kg of product 26</td>
</tr>
<tr>
<td>30</td>
<td>2550 kg of product 28 and 2550 kg of product 29</td>
</tr>
<tr>
<td>31</td>
<td>6118 kg of product 26</td>
</tr>
</tbody>
</table>

Table 2.2: Requirements of products manufactured in the system that are used as raw materials.
manufacture of every product lot. A set-up is required even between consecutive lots of the same product. This is due to the nature of the operations undertaken in this manufacturing system. For instance, due to safety measures a deep cleaning of certain production line components has to be performed before processing a new lot, even if it is a lot of the same product, in order to remove chemical compounds (formed during manufacturing) that could react with raw materials (e.g. exothermic reactions).
<table>
<thead>
<tr>
<th>Product</th>
<th>Technical Lot</th>
<th>Standard Cost (USD/lot)</th>
<th>Price (USD/lot)</th>
<th>Profit (USD/lot)</th>
<th>Demand (lot/month)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1248 box /lot</td>
<td>18154</td>
<td>21917</td>
<td>3763</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>1100 box /lot</td>
<td>14344</td>
<td>18107</td>
<td>3763</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>1100 box /lot</td>
<td>14145</td>
<td>20104</td>
<td>5958</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>1100 box /lot</td>
<td>14258</td>
<td>20215</td>
<td>5957</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>936 box /lot</td>
<td>13466</td>
<td>18639</td>
<td>5173</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>1100 box /lot</td>
<td>14506</td>
<td>14960</td>
<td>454</td>
<td>13</td>
</tr>
<tr>
<td>7</td>
<td>1100 box /lot</td>
<td>13374</td>
<td>18954</td>
<td>5579</td>
<td>12</td>
</tr>
<tr>
<td>8</td>
<td>1100 box /lot</td>
<td>13942</td>
<td>18045</td>
<td>4102</td>
<td>11</td>
</tr>
<tr>
<td>9</td>
<td>1100 box /lot</td>
<td>14501</td>
<td>16812</td>
<td>2311</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>1100 box /lot</td>
<td>14272</td>
<td>19456</td>
<td>5183</td>
<td>8</td>
</tr>
<tr>
<td>11</td>
<td>1100 box /lot</td>
<td>14385</td>
<td>19276</td>
<td>4891</td>
<td>15</td>
</tr>
<tr>
<td>12</td>
<td>1120 box /lot</td>
<td>13982</td>
<td>20505</td>
<td>6523</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>1200 box /lot</td>
<td>14564</td>
<td>22044</td>
<td>7480</td>
<td>13</td>
</tr>
<tr>
<td>14</td>
<td>1700 box /lot</td>
<td>2665</td>
<td>14139</td>
<td>11474</td>
<td>12</td>
</tr>
<tr>
<td>15</td>
<td>1100 box /lot</td>
<td>13533</td>
<td>17875</td>
<td>4343</td>
<td>13</td>
</tr>
<tr>
<td>16</td>
<td>1100 box /lot</td>
<td>14099</td>
<td>17694</td>
<td>3594</td>
<td>9</td>
</tr>
<tr>
<td>17</td>
<td>936 box /lot</td>
<td>13607</td>
<td>14301</td>
<td>693</td>
<td>11</td>
</tr>
<tr>
<td>18</td>
<td>1260 box /lot</td>
<td>17110</td>
<td>18795</td>
<td>1685</td>
<td>20</td>
</tr>
<tr>
<td>19</td>
<td>760 box /lot</td>
<td>18260</td>
<td>21765</td>
<td>3505</td>
<td>30</td>
</tr>
<tr>
<td>20</td>
<td>17712 kg /lot</td>
<td>15538</td>
<td>22155</td>
<td>6617</td>
<td>15</td>
</tr>
<tr>
<td>21</td>
<td>17534.88 kg /lot</td>
<td>13977</td>
<td>22233</td>
<td>8257</td>
<td>15</td>
</tr>
<tr>
<td>22</td>
<td>17534.88 kg /lot</td>
<td>13443</td>
<td>22233</td>
<td>8790</td>
<td>12</td>
</tr>
<tr>
<td>23</td>
<td>12398.4 kg /lot</td>
<td>11185</td>
<td>15498</td>
<td>4313</td>
<td>9</td>
</tr>
<tr>
<td>24</td>
<td>1080 box /lot</td>
<td>17316</td>
<td>22201</td>
<td>4884</td>
<td>30</td>
</tr>
<tr>
<td>25</td>
<td>80000 kg /lot</td>
<td>35627</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>26</td>
<td>14400 kg /lot</td>
<td>14444</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

Continues on next page
<table>
<thead>
<tr>
<th>Product</th>
<th>Technical Lot</th>
<th>Standard Cost (USD/lot)</th>
<th>Price (USD/lot)</th>
<th>Profit (USD/lot)</th>
<th>Demand (lot/month)</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>14300 kg /lot</td>
<td>15431</td>
<td>15544</td>
<td>114</td>
<td>10</td>
</tr>
<tr>
<td>28</td>
<td>2700 kg /lot</td>
<td>432</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>29</td>
<td>2700 kg /lot</td>
<td>416</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>30</td>
<td>5000 kg /lot</td>
<td>556</td>
<td>6847</td>
<td>6290</td>
<td>10</td>
</tr>
<tr>
<td>31</td>
<td>5940 kg /lot</td>
<td>488</td>
<td>6077</td>
<td>5588</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 2.3: Product lot characteristics and demand level.
Moreover, the technical lot\(^1\) of every product is fixed and it has been specified by the company so that an entire lot can be manufactured within one shift. Here, a shift has a length of 8 hours, which corresponds to the daily amount of hours that an operator needs to work. Therefore, the theoretical manufacturing time for every product lot is 8 hours (if everything goes well). This theoretical manufacturing time already considers set-up time and time spent in transportation of necessary resources to and within the production line involved, but it does not consider unexpected events such as delays caused by failures of production lines.

In this manufacturing system, each operator needs to perform a wide variety of operations and needs to possess the skills required to operate the different components of the multiple production lines; for this reason any operator can be assigned to work in any of the 7 production lines. This system has available a total of 7680 man hours per working month (24 working days). Labour is a limiting factor of this manufacturing system, but due to the high level of skills required it is not possible to increase the capacity of the manufacturing system by hiring additional personnel, at least not in the short term. Specifications about the number of man hours and hours of production lines needed to manufacture one product lot are given in Table 2.4.

Maximum 3 shifts, of 8 hours each, can be undertaken per day at each production line, depending on the amount of labour available. Thus, without considering labour, the monthly design capacity (Heizer et al., 2004, p. 252) of every production line is 72 lots or 576 hours if expressed as a production rate or as the number of hours that the production line is available, respectively.

Based on the design capacities of production lines, on the number of man hours available and on the monthly level of demand, this system has insufficient capacity to fully cover demand requirements. Market conditions do not allow backordering, since any unmet demand is covered by competitors. Additionally this manufacturing system

\(^1\)the number of items produced per lot of product
<table>
<thead>
<tr>
<th>Product</th>
<th>Labour Requirement (h/lot)</th>
<th>Production Line Requirement (h/lot)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>72</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>64</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>64</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>64</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>56</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>64</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>64</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>64</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>64</td>
<td>8</td>
</tr>
<tr>
<td>10</td>
<td>64</td>
<td>8</td>
</tr>
<tr>
<td>11</td>
<td>64</td>
<td>8</td>
</tr>
<tr>
<td>12</td>
<td>64</td>
<td>8</td>
</tr>
<tr>
<td>13</td>
<td>56</td>
<td>8</td>
</tr>
<tr>
<td>14</td>
<td>40</td>
<td>8</td>
</tr>
<tr>
<td>15</td>
<td>64</td>
<td>8</td>
</tr>
<tr>
<td>16</td>
<td>64</td>
<td>8</td>
</tr>
<tr>
<td>17</td>
<td>56</td>
<td>8</td>
</tr>
<tr>
<td>18</td>
<td>40</td>
<td>8</td>
</tr>
<tr>
<td>19</td>
<td>48</td>
<td>8</td>
</tr>
<tr>
<td>20</td>
<td>64</td>
<td>8</td>
</tr>
<tr>
<td>21</td>
<td>64</td>
<td>8</td>
</tr>
<tr>
<td>22</td>
<td>64</td>
<td>8</td>
</tr>
<tr>
<td>23</td>
<td>56</td>
<td>8</td>
</tr>
<tr>
<td>24</td>
<td>64</td>
<td>8</td>
</tr>
<tr>
<td>25</td>
<td>40</td>
<td>8</td>
</tr>
<tr>
<td>26</td>
<td>40</td>
<td>8</td>
</tr>
<tr>
<td>27</td>
<td>40</td>
<td>8</td>
</tr>
<tr>
<td>28</td>
<td>32</td>
<td>8</td>
</tr>
<tr>
<td>29</td>
<td>32</td>
<td>8</td>
</tr>
<tr>
<td>30</td>
<td>32</td>
<td>8</td>
</tr>
<tr>
<td>31</td>
<td>16</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 2.4: Requirements of labour and production lines per product lot.
is subject to failures of production lines, which further complicates the situation.

Historical records from January 2010 to June 2014 were obtained about the number of failures occurred per production line during that period as well as the corresponding repair times measured in days caused by each production line failure. However, no records could be obtained for the number of lots manufactured during that period in the different production lines. For this reason we could not calculate the exact probability that a failure occurs in a production line during the manufacturing of a product lot. Nevertheless, a conservative lower bound for such probability was calculated for every production line, based on the historical information mentioned above and by considering that every production line was fully utilized (without considering labour requirements) and reliable (no failures) during a period of 54 months (number of months for which information is available). In other words, it was assumed that every production line produced monthly 72 lots during 54 months, which is an unrealistic (too optimistic) assumption, since labour constraints and the occurrence of failures in production lines are completely ignored. For this reason, it is reasonable to consider that actual values for those probabilities should be higher than the ones reported in Table 2.5, which presents those conservative probabilities as well as the average repair times per production line. As can be seen in Table 2.5, no failures were recorded for production lines 1 and 6 during that 54-month period.

All activities in a production line stop after the occurrence of a failure and are resumed once the repair service on that line has been completed. The delays caused by repair services compromise the realization of a given production plan, as lots at the end of the production sequence scheduled for that line might not be completed on time. In terms of labour, workers assigned to that production line are idle until it becomes operative again. When this happens, the workers on that shift resume operations until the end of their shift and leave the remaining activities to be completed by workers from the subsequent shift. Therefore, the occurrence of production line failures may
<table>
<thead>
<tr>
<th>Production line</th>
<th>Number of failures</th>
<th>Probability of failure (%)</th>
<th>Average detention (d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>138</td>
<td>3.55</td>
<td>2.03</td>
</tr>
<tr>
<td>3</td>
<td>182</td>
<td>4.68</td>
<td>2.24</td>
</tr>
<tr>
<td>4</td>
<td>183</td>
<td>4.71</td>
<td>3.21</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>0.08</td>
<td>4.33</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>0.13</td>
<td>3.80</td>
</tr>
</tbody>
</table>

Table 2.5: Conservative probabilities of failure and average detention time per production line.
force to produce a given product lot within different shifts, although the technical lot of every product has been specified with the intention of manufacturing a product lot within a single shift. In this sense, having the possibility to make decisions about the sizes of technical lots may bring more flexibility to improve the performance of this manufacturing system. Moreover, assigning new tasks to idle workers are also other important decisions, which can be made to improve the performance of this system. However, lot-sizing and re-allocation of labour are outside the scope of this study as those are decisions taken at the operational level rather than at the tactical level.

This company needs to purchase the materials needed for the next working month, before the beginning of the production period, and thus a production plan for the following month must be available in order to make adequate purchasing decisions. This is a difficult task because by the time a production plan needs to be developed specific due dates of orders are still unknown and only demand forecasts are available. This manufacturing company develops those tactical production plans on a monthly basis and the generation of those plans is responsibility of the production planning committee (PPC).

### 2.2 Production Planning Challenges

One difficulty faced by this manufacturing system is that production plans are developed with the sole intention of maximizing company’s profitability. Other important criteria such as robustness of production plans are not considered during the specification of production plans (the failure to consider the multi-objective nature of production planning mentioned in Section 1.1).

Also the level of detail contained in monthly production plans poses challenges to their implementation. Plans specified by the PPC only indicate the number of lots to produce for the different products, but they do not specify in what production lines
those product lots should be manufactured (the aggregation principle mentioned in Section 1.1). Not having the latter information causes serious difficulties during the implementation of those plans, mainly due to overestimation of system capacity.

One of the main difficulties faced by this manufacturing system is the consideration of system constraints during the development of monthly production plans. System constraints are considered by the PPC via simple rules of thumb (the use of unrealistic assumptions mentioned in Section 1.1). Moreover, this committee does not consider from the very beginning the capacity needed to manufacture sub-products. Those considerations are only made after a preliminary plan has been obtained, which is then revised to account for the lots of sub-products needed.

Finally, the inherent uncertainty of the manufacturing system is not taken into account by the PPC. For instance, the uncertainty around potential delays caused by the occurrence of production line failures are not considered during the specification of a production plan (the use of unrealistic assumptions mentioned in Section 1.1).

There is a clear correspondence between the difficulties mentioned in Section 1.1 and the challenges faced by this company, which suggests that those challenges may be the consequence of methodological limitations.

### 2.3 Production Planning Problem

In order to overcome the challenges mentioned in Section 2.2, the monthly production planning of this company is addressed in this work as an optimization problem that tries to determine the number of lots of the different products and sub-products that every production line needs to manufacture, during a finite planning horizon of one month, in order to simultaneously optimize at least two conflicting criteria, the profitability and robustness of production plans. The uncertainty derived from failures
in production lines, as well as complex features, such as the multi-product, multi-
production line, and multi-level nature of the manufacturing system, needs to be ac-
curately incorporated into the problem formulation in order to generate feasible and
applicable solutions. Considering all these requirements, this problem corresponds to
a multi-objective, big bucket, multi-product, multi-level (sub-products), capacitated
(constraints are considered) production planning problem of a failure-prone manufac-
turing system, conformed by multiple production lines with insufficient capacity to
fully cover demand requirements.
Chapter 3

Initial Simulation-Based Optimization Model (Manuscript 1)


3.1 Abstract

Effective production planning requires models that are capable of accounting for the complexity and uncertainty intrinsic to manufacturing systems. While the identification of a globally optimal plan is desirable, a more important requirement is the ability of a model to produce production plans that are sufficiently realistic to be implemented in practice and are robust to perturbations in the system. Here, we present a simulation-based optimization approach that employs DES and a GA as a methodology to support decision making in the area of production planning. The model aims to minimize the sum of expected backorders and inventory costs, while incorporating system constraints and the uncertainty that derives from variations of manufacturing
lead times, occurrence of production line failures and repair service times. Preliminary results for a real-world problem indicate that the model is capable of producing feasible production plans that correctly account for the uncertainty intrinsic to the underlying manufacturing system.

3.2 Introduction

Production planning, which specifies how resources should be allocated to production activities (Monostori et al., 2010), forms an integral part of medium-term planning within manufacturing processes. Given the increasing pressures faced by manufacturers, the development and deployment of effective models that support production planning is essential.

Ideally, an optimal production plan should be able to achieve customer satisfaction (Pochet, 2001) along with profit maximization, while considering the uncertainty in the system (Monostori et al., 2010; Li et al., 2004). Therefore, an appropriate methodology needs to perform optimization while accounting for the effects that uncertain parameters may have on the implementation of a production plan. This should then lead to an optimized solution that is robust towards various sources of uncertainty in the manufacturing system. The lack of an instrument that is fully able to meet this requirement is one of the main reasons why, currently, decisions in production planning are often made in a subjective manner (based on the experience and “sixth sense” of a few people) or guided by inappropriate methodologies.

Optimization and simulation models have been previously deployed to solve production planning problems, albeit independently. Optimization models are able to generate optimal or near-optimal solutions, but the real applicability of these solutions is often limited. This is because of the oversimplifying assumptions made by many exact optimization models and their inability to fully incorporate uncertainty (Gnoni et al.,
2003; Nikolopoulou and Ierapetritou, 2012). Furthermore, when trying to incorporate the high level of complexity and the stochastic (Lacksonen, 2001) and dynamic nature of manufacturing systems (Azzaro-Pantel et al., 1998) into optimization models, standard approaches become computationally intractable. On the other hand, simulation approaches are capable of capturing the uncertainty of the system (Monostori et al., 2010) and of accurately reproducing its behaviour (Hsieh, 2002). Therefore, simulation often provides a better representation of a real production system, since the variability introduced through exogenous and endogenous factors can be explicitly considered and the impact of these factors can be assessed (April et al., 2006).

However, in contrast to optimization approaches, the results obtained from simulation models are fundamentally descriptive: while a clear picture of the system is obtained, the results do not provide explicit guidance towards improved solutions.

In an attempt to combine the respective advantages of simulation and optimization techniques, SBO has been suggested as a means of handling problems where the high level of complexity precludes a complete analytic formulation and the ultimate goal is the identification of a robust, near-optimal solution (Gray et al., 2010). More specifically, the combined application of DES and GAs has been successfully applied to address several problems related to manufacturing systems. For instance, Azzaro-Pantel et al. (1998) were able to improve the efficiency of a multi-purpose, multi-objective plant with limited storage by accurately modelling the dynamic behaviour of the production system through DES and solving the scheduling problem using a GA. Al-Aomar (2006) combined DES and a GA to determine robust design parameters. The author integrated Taguchis’s robustness measures of signal-to-noise ratio and the quality loss function into a GA in order to enhance the selection scheme. Ding et al. (2005) employed DES to capture the uncertainty involved in the supplier selection process and used a GA to optimize the supplier portfolio. Cheng and Yan (2009) applied an integration of DES and a messy GA to determine the near optimal combination of resources.
in order to enhance the performance of construction operations. This approach enabled
the authors to cope with the complexity and large dimensionality of the problem and to
obtain adequate solutions. Wu et al. (2011) integrated DES with a GA to determine the
order point for different product types of a cross-docking center in order to minimize
total cost. Through this approach the solution space was efficiently reduced and more
simulation effort was allocated to promising areas via smart computing budget alloca-
tion. Korytkowski et al. (2013) proposed an evolutionary simulation-based heuristics,
where DES and a GA were deployed to find near optimal solutions for dispatching
rules allocation. The sequence of orders determined through this approach improved
the performance of a complex multi-stage, multi-product manufacturing system.

Here, we describe a SBO approach for production planning. The long-term aim of
our work is to derive an effective modelling approach that is capable of determining
feasible and robust monthly production plans. Here, we formulate production planning
as an optimization problem that requires the minimization of the expected sum of back-
orders and inventory costs, subject to a set of constraints of the manufacturing system
(e.g. resource constraints) and uncertainties deriving from variations of manufactur-
ing lead times, occurrence of production line failures and repair service times. Our
choice of methodology is motivated by the proven success of SBO in related problems
(see Korytkowski et al. (2013), Wu et al. (2011), Cheng and Yan (2009), Al-Aomar
(2006), Ding et al. (2005), Azzaro-Pantel et al. (1998) and above), and we develop a
model based on the combination of DES and a meta-heuristic optimizer (specifically,
a GA). Finally, we describe preliminary results on a real-world production planning
problem.
CHAPTER 3

3.3 Simulation-based Optimization Model

The production planning problem considered here is based on the real manufacturing system of a large company that specializes in the production of cleaning products, edible shortenings, fats and oils. This study focuses exclusively on its activities related to the manufacturing of cleaning products.

DES is a good option to model the dynamic behaviour of this production system (Azzaro-Pantel et al., 1998), as it allows for the incorporation of stochastic events and the variations of processes that occur in complex systems (Riley, 2013). Specifically, the use of DES enables us to capture the uncertainty intrinsic to production planning that cannot be represented by deterministic models (Monostori et al., 2010).

The application of SBO implies the absence of an analytical problem formulation, i.e. the functional relationships between dependent and independent variables are not known explicitly (Steponavičė et al., 2014). Consequently, a suitable optimization approach needs to be able to perform optimization based exclusively on function values obtained via simulation, a so called “black-box approach”. Considering the complexity and large dimensionality of the solution space, a suitable search strategy should be able to find near-optimal solutions in a large and complex solution space and be capable of escaping local optima. Finally, the optimization method needs to be robust with respect to noise, since the optimization procedure relies on stochastic responses generated by the simulation model (Gray et al., 2010). Meta-heuristics present suitable candidates for this setting, and in this study, a GA was selected as the optimizer. This choice was motivated by previous research indicating the robust performance of GAs under noisy conditions (Mitchell, 1998; Baum et al., 1995), specifically, in the context of DES optimization (Lacksonen, 2001).

The DES model of the production system was developed in SimEvents® (The MathWorks, Inc., 2013). This was integrated with Matlab® R2013a (The MathWorks,
Inc., 2013), and Matlab’s standard GA implementation was employed as the optimizer. Details of the simulation model and optimizer are described in the following sections.

3.3.1 Simulation Model

The DES model represents the production of 31 products $j$ within 7 production lines $l$. A production line corresponds to the set of resources (e.g. machines, people, etc.) needed to manufacture certain products. Given that some products can be manufactured in several production lines a total of 41 processes $y$ are considered in the DES model. A process $y$ includes all series of events involved in the initialization of orders of a product $j$, its manufacture in a specific production line $l$ and its storage in a specific container, denoted as $sink$ (with $sink = 1, 2, \ldots, 41$). Here orders are measured in number of lots. The simulation time $t$ of each simulation replication is 24 days, which corresponds to the number of working days in a month.

The model component designed for the generation of orders for a single production line $l$ is illustrated in Figure 3.1. A production plan is a vector of decision variables $x_y$, which are specified by a GA. The values of the different decision variables are the inputs for the simulation model (function-call generator blocks). Given that some products $j$ are required as raw materials during the manufacturing process of other products $j$, a higher priority is assigned to the initialization of orders for those sub-products in order to assure the static logic of the model.

Attributes are assigned to the different product lots (via attribute blocks). Specifications about the containers ($sink$), where final products will be stored, are assigned via an attribute called $OutputPort_y$. Furthermore, the time required to manufacture a specific product lot ($ManufacturingTime_y$) and the occurrence of a failure in a production line while processing a product lot ($ProductionLineFailure_y$) are additional attributes assigned to each lot of product. Two different event-based random number
generators are employed to set the last two attributes mentioned. Both event-based random number generators produce a signal sampled randomly from the probability distribution functions (PDFs) assigned to them. A synthetic data set was employed
to estimate PDFs for each stochastic variable included in the current study, as data collection for these aspects of the system is currently incomplete.

Once the attributes have been assigned to the production orders, those orders are transferred to a queue following a first-in first-out (FIFO) discipline. Subsequently, those queues of production orders that have to be processed by the same production line are merged (by a path combiner block) into a single FIFO queue.

The model components of a production subsystem and repair service centre are illustrated in Figure 3.2 and Figure 3.3, respectively. Each order is manufactured as soon as the corresponding production line (represented by an N-server block) becomes available. In case of failure, the activity of that production line is blocked by the control signal \( \text{Pause}_l \). This signal is generated from the corresponding repair service centre and it outputs the number of entities present in that repair centre. Therefore, a signal with value greater than zero indicates that the production line \( l \) is being repaired and stops its activity until that signal becomes zero (no entities present in the repair service centre).

In case that no failure occurs (\( \text{ProductionLineFailure}_y = 1 \)), the production batch is transferred to the corresponding container (sink) determined by \( \text{OutputPort}_y \). Whereas if a failure occurs (\( \text{ProductionLineFailure}_y = 2 \)), that product batch is transferred to a repair service centre prior to its storage. The delay caused by the production line failure is sampled from the corresponding PDF assigned to \( \text{RepairServiceTime}_l \). One important assumption made is that after a production batch has left the repair service centre no re-manufacture is required, since the manufacturing process has been already completed (passed through the N-server block). This is an effective way to model system failure without having conflicting events.

The stock of product \( j \) manufactured in production line \( l \), denoted by \( s_{l,j} \), is computed at the end of every replication and it is measured in number of lots. Based on \( s_{l,j} \),
Figure 3.2: Production subsystem for production line $l$. 
CHAPTER 3

the total stock of product \( j \) \( (s_j) \) is calculated at the end of every replication as follows:

\[
s_j = \sum_{l=1}^{7} s_{l,j}.
\]  

(3.1)

3.3.2 Optimization Model

The decision variables, denoted by \( x_y \), are the number of lots to be produced in process \( y \). A black box optimization approach is applied in which the decision variables specified by the GA provide the input to the DES model and the responses \( s_j \) from the DES model are employed to compute the value of the fitness function. A total of 41 decision variables \( x_y \), which are constrained to be positive integers, are considered in the model. Given the stochastic nature of the DES outputs, fitness is evaluated across \( \gamma \) independent simulation trials (with \( \gamma = 10 \)). Specifically, the fitness value \( f \) is estimated for each individual \( x \) as follows:

\[
f(x) = \frac{1}{\gamma} \sum_{r=1}^{\gamma} \sum_{j=1}^{31} [\text{InventoryCost}_{j,r} + \text{BackorderCost}_{j,r}],
\]  

(3.2)

where \( \text{InventoryCosts}_{j,r} \) and \( \text{BackorderCosts}_{j,r} \) are computed based on the \( r \)th simulated response \( s_{j,r} \) as follows:

\[
\text{InventoryCost}_{j,r} = \begin{cases} 
(s_{j,r} - D_j) \times \text{Cost}_j & \text{if } s_{j,r} > D_j \\
0 & \text{if } s_{j,r} \leq D_j
\end{cases}
\]
and

$$\text{BackorderCost}_{j,r} = \begin{cases} 
(D_j - s_{j,r}) \times Price_j & \text{if } s_{j,r} < D_j \\
0 & \text{if } s_{j,r} \geq D_j 
\end{cases}$$

where, \(D_j\) indicates the demand for product \(j\). Unsold amounts of product \(j\) are penalized proportionally to the corresponding standard cost per lot \((\text{Cost}_j)\), whereas backorders receive a fine equal to the product price, which is the income lost \((\text{Price}_j)\) for not selling that specific amount of product. Given a lack of information on real inventory costs and total cost derived from backorders per product (cost of customer dissatisfaction, cost of non-future purchases, cost of customers switching to other brands, etc.), standard costs and product prices are currently employed to penalize inventory and backorders, respectively. These two assumptions are not valid in reality for several reasons. First, excess of inventory can be sold in future periods and inventory costs are not equal to standard costs. Second, considering product price as the total loss caused by product backorders is inaccurate and unrealistic.

Additional constraints are imposed given that some products \(j\) are required as raw materials during the manufacturing process of other products \(j\). Therefore, the requirement of sub-products is represented through linear constraints as follows:

$$\sum_{y=1}^{41} a_{i,y} \times x_y \leq b_i \quad (i = 1, 2, \ldots, 4), \quad (3.3)$$

where \(b_i\) denotes the quantity available of sub-product \(i\) and \(a_{i,y}\) is the amount required of sub-product \(i\) to produce one lot in process \(y\).

Here, we use as optimizer the default Matlab implementation for solving integer and mixed integer problems using a GA. This is a real-coded GA that has a population size of 50 individuals and employs Laplace crossover (crossover probability: 0.8), power mutation (mutation probability: 0.005) and tournament selection (tournament size: 2) as operators. A detailed description of the GA and its truncation procedure
(which ensures compliance with integer constraints after crossover and mutation) can be found in Deep et al. (2009). The inbuilt constraint-handling approach is the parameter free penalty function approach proposed by Deb (2000). Elitism is implemented in this GA by having an elite set of 1 and according to Matlab customer service, the fitness of elite individuals is re-evaluated across generations in this default Matlab implementation.

3.4 Preliminary Results

The model enables an accurate incorporation of uncertainty derived from variations of manufacturing lead times, occurrence of production line failures and repair service times. The time required to run 15 iterations of the GA is 12.15 hours. For this reason, very limited results are reported in the present study, and we mostly focus on the validity of the model designed. More extensive benchmarking of the approach (including longer optimization runs and statistics across multiple trials) is currently in progress.

As shown in Figure 3.4, when run for 15 iterations, the GA successfully reduces the expected sum of backorders and inventory costs. The best production plan after 15 iterations is presented in Table 3.1. For reference, the amount of demand to be covered, the consolidated number of lots per product to be manufactured and the actual number of lots produced are shown in Table 3.2. The allocation for production line 204 provides a suitable illustration of the results obtained. For the majority of products (except for product B), production line 204 displays a more reliable performance than production line 203. For this reason, the suggested production plan (see Table 3.1) allocates a greater number of orders to production line 204. This solution is in accordance with our expectations and illustrates that the reliability of production line is correctly accounted for in the production plan generated.
3.5 Future Research

There are a number of ways in which this research will be extended in future work. Regarding the simulation component of the work, data collection (from the company) needs to be completed. The data obtained will be used to estimate PDFs of all stochastic variables, so that the use of synthetic data can be avoided.

Regarding the optimizer, future work will include an investigation of parameter settings, the sensitivity to noise and, potentially, the comparison to alternative metaheuristic optimization approaches. Moreover, the number \( n \) of simulation trials employed to evaluate fitness will be further analysed in order to balance quality of estimations and computational cost. Furthermore, a multi-objective formulation of the problem will be explored in order to account for the robustness of solutions in a more explicit manner. Specifically, the maximization of the signal-to-noise ratio may be used as an additional objective to directly account for the variability in the fitness values obtained (Al-Aomar, 2006).
<table>
<thead>
<tr>
<th>Product</th>
<th>Production line</th>
<th>Production plan$^a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>204</td>
<td>14</td>
</tr>
<tr>
<td>B</td>
<td>203</td>
<td>7</td>
</tr>
<tr>
<td>B</td>
<td>204</td>
<td>6</td>
</tr>
<tr>
<td>C</td>
<td>203</td>
<td>10</td>
</tr>
<tr>
<td>D</td>
<td>203</td>
<td>6</td>
</tr>
<tr>
<td>E</td>
<td>203</td>
<td>10</td>
</tr>
<tr>
<td>F</td>
<td>203</td>
<td>8</td>
</tr>
<tr>
<td>G</td>
<td>203</td>
<td>7</td>
</tr>
<tr>
<td>G</td>
<td>204</td>
<td>14</td>
</tr>
<tr>
<td>H</td>
<td>203</td>
<td>3</td>
</tr>
<tr>
<td>H</td>
<td>204</td>
<td>11</td>
</tr>
<tr>
<td>I</td>
<td>203</td>
<td>3</td>
</tr>
<tr>
<td>I</td>
<td>204</td>
<td>8</td>
</tr>
<tr>
<td>J</td>
<td>203</td>
<td>14</td>
</tr>
<tr>
<td>K</td>
<td>203</td>
<td>6</td>
</tr>
<tr>
<td>L</td>
<td>203</td>
<td>12</td>
</tr>
<tr>
<td>L</td>
<td>204</td>
<td>13</td>
</tr>
<tr>
<td>M</td>
<td>203</td>
<td>7</td>
</tr>
<tr>
<td>M</td>
<td>204</td>
<td>19</td>
</tr>
<tr>
<td>N</td>
<td>204</td>
<td>14</td>
</tr>
<tr>
<td>O</td>
<td>203</td>
<td>5</td>
</tr>
<tr>
<td>O</td>
<td>204</td>
<td>10</td>
</tr>
<tr>
<td>P</td>
<td>203</td>
<td>4</td>
</tr>
<tr>
<td>P</td>
<td>204</td>
<td>4</td>
</tr>
<tr>
<td>Q</td>
<td>203</td>
<td>12</td>
</tr>
<tr>
<td>R</td>
<td>202</td>
<td>5</td>
</tr>
<tr>
<td>R</td>
<td>203</td>
<td>3</td>
</tr>
<tr>
<td>R</td>
<td>204</td>
<td>7</td>
</tr>
<tr>
<td>S</td>
<td>202</td>
<td>7</td>
</tr>
<tr>
<td>T</td>
<td>203</td>
<td>9</td>
</tr>
<tr>
<td>U</td>
<td>203</td>
<td>12</td>
</tr>
<tr>
<td>V</td>
<td>203</td>
<td>13</td>
</tr>
<tr>
<td>W</td>
<td>203</td>
<td>2</td>
</tr>
<tr>
<td>X</td>
<td>202</td>
<td>9</td>
</tr>
<tr>
<td>Y$^b$</td>
<td>101</td>
<td>3</td>
</tr>
<tr>
<td>Z$^b$</td>
<td>204</td>
<td>10</td>
</tr>
<tr>
<td>AA</td>
<td>204</td>
<td>6</td>
</tr>
<tr>
<td>AB$^b$</td>
<td>205</td>
<td>8</td>
</tr>
<tr>
<td>AC$^b$</td>
<td>205</td>
<td>16</td>
</tr>
<tr>
<td>AD</td>
<td>208</td>
<td>7</td>
</tr>
<tr>
<td>AE</td>
<td>301</td>
<td>9</td>
</tr>
</tbody>
</table>

$^a$ best production plan generated by the model after 15 iterations with $n = 10$,  
$^b$ Sub-products

Table 3.1: Number of product lots to be manufactured in a specific production line.
<table>
<thead>
<tr>
<th>Product</th>
<th>Demand $^a$</th>
<th>Production $^a$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Planned</td>
<td>Actual $^b$</td>
</tr>
<tr>
<td>A</td>
<td>10</td>
<td>14</td>
</tr>
<tr>
<td>B</td>
<td>9</td>
<td>13</td>
</tr>
<tr>
<td>C</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>D</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>E</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>F</td>
<td>13</td>
<td>8</td>
</tr>
<tr>
<td>G</td>
<td>12</td>
<td>21</td>
</tr>
<tr>
<td>H</td>
<td>11</td>
<td>14</td>
</tr>
<tr>
<td>I</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>J</td>
<td>8</td>
<td>14</td>
</tr>
<tr>
<td>K</td>
<td>15</td>
<td>6</td>
</tr>
<tr>
<td>L</td>
<td>12</td>
<td>25</td>
</tr>
<tr>
<td>M</td>
<td>13</td>
<td>26</td>
</tr>
<tr>
<td>N</td>
<td>12</td>
<td>14</td>
</tr>
<tr>
<td>O</td>
<td>13</td>
<td>15</td>
</tr>
<tr>
<td>P</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>Q</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>R</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>S</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>T</td>
<td>15</td>
<td>9</td>
</tr>
<tr>
<td>U</td>
<td>15</td>
<td>12</td>
</tr>
<tr>
<td>V</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>W</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>X</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Y$^c$</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Z$^c$</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>AA</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>AB$^c$</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>AC$^c$</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>AD</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>AE</td>
<td>10</td>
<td>9</td>
</tr>
</tbody>
</table>

$^a$ measured in number of lots, $^b$ average value of 10 independent replications, $^c$ sub-products

Table 3.2: Demand, consolidated production plan per product and actual production achieved.
Chapter 4

Noise Handling Strategies

(Manuscript 2)


4.1 Abstract

In this chapter, we explore the impact of noise handling strategies on optimization performance in the context of the real-world production planning problem analysed in the previous chapter. Uncertainties intrinsic to the production system are captured using a modified version of the DES model presented in Chapter 3 (see Section 4.3.1 for further details about those modifications), and the production plan is optimized using an EA. Unlike Chapter 3, here we aim for profit maximization rather than for cost minimization as this is the objective that the manufacturing company intends to optimize. The stochastic nature of the fitness values (as returned by the DES simulation) may impact on optimization performance, and we explore explicit and implicit
averaging strategies to address this issue. Specifically, we evaluate the effectiveness of different strategies, when a limited budget of evaluations is available. Our results indicate a general advantage of implicit averaging in this setting, and a good degree of robustness with regard to population size. On the other hand, explicit averaging is found to be non-competitive, due to the cost of repeat-evaluations of the same solution. Finally, we explore a hybrid approach that uses explicit averaging to refine fitness estimates during final solution selection. Under increasing levels of fitness variability, this hybrid strategy starts to outperform pure implicit and explicit averaging strategies.

4.2 Introduction

Optimization problems that include uncertainty pose challenges that are difficult to address using standard optimization methodologies. While a portion of the optimization literature is concerned with the development of methodologies capable of identifying optimal solutions to problems with uncertainty, the application of these methods often requires stringent assumptions and/or simplifications that are necessary to satisfy relevant optimality conditions. Those methods are often insufficiently powerful to accurately incorporate the full complexity and uncertainty intrinsic to real-world problems into the problem formulation, even when their consideration is essential for the generation of reliable and feasible solutions. For this reason, solutions obtained from traditional approaches (such as fuzzy, stochastic and stochastic dynamic programming) to optimization problems under uncertainty may often be of limited value in producing realistic solutions for real-world problems.

SBO constitutes an interesting alternative in situations where the high level of complexity precludes a complete analytic formulation of a problem (Ehrenberg and Zimmermann, 2012) and where uncertainty needs to be considered (Figueira and Almada-Lobo, 2014). The SBO model used here consists of a detailed simulation model, which
is then coupled with an optimizer in a black-box fashion. In other words, the optimizer operates on a (sub-)set of model parameters and the optimization process is based exclusively on the (usually stochastic) simulation responses. EAs are well-suited to black-box optimization settings, as highlighted by their wide application to real-world optimization problems that cannot be handled by analytical approaches (Qian et al., 2014). The feasibility and reliability of solutions become the primary consideration in such settings (Figueira and Almada-Lobo, 2014), and the EAs’ flexibility in this respect typically offsets its disadvantages (specifically, the lack of guaranteed optimality of its identified solutions).

When EAs are employed as optimizers of SBO models, fitness values become subject to the variability arising from the stochastic responses within the simulation model. The resulting noisy nature of the fitness values poses a challenge to the evolutionary optimizer, for it may mislead selection procedures (Bhattacharya et al., 2014) and lead to the propagation of inferior individuals or to the elimination of superior ones, thereby undermining algorithm performance (Syberfeldt et al., 2010). Under these circumstances, noise handling strategies can play an important role in compensating for the impact of noise on the optimizer, and, specifically, in helping the optimizer to identify solutions that exhibit low fitness variability and give rise to high average fitness. Multiple studies have analysed situations in which noise causes perturbations during fitness evaluation, thus generating discrepancies between the observed and “true” fitness (Qian et al., 2014). We refer the reader to Jin and Branke (2005) for a comprehensive survey of noise handling strategies proposed in the existing literature.

Implicit and explicit averaging are the two strategies most commonly employed to reduce the influence of noise in evolutionary optimization under noise. Implicit averaging relies on the EA mechanism itself to compensate for the impact of noise. Specifically, it assumes that the use of sufficiently large populations will ensure that individuals from promising regions of the search space are sampled repeatedly (Jin and
Branke, 2005; Syberfeldt et al., 2010), and the impact of noise can be reduced in this manner. On the other hand, explicit averaging strategies ensure that individuals are evaluated using average fitness values obtained across a specific number \( (n) \) of fitness evaluations (replicates). Statistically, this approach ensures that the expected error of fitness estimates (i.e. the difference between the observed and the “true” fitness mean) reduces with a factor of \( \sqrt{n} \) (Jin and Branke, 2005).

Both implicit and explicit averaging strategies incur additional fitness evaluations due to \((i)\) the increase in population size and due to \((ii)\) the increase in the number of trials, respectively. Fitness evaluations present an important consideration in SBO, as each replication of a simulation is time-consuming and the number of these replications may be limited by available computational time. Here, we investigate the efficiency and effectiveness of different noise-handling strategies in a realistic SBO setting, in which the computational budget available for the optimization (and therefore, the overall number of simulation replicates) is limited. Specifically, we compare explicit averaging against implicit averaging strategies for two different population sizes. Finally, we investigate a hybrid scheme that aims to combine the strengths of both approaches.

The remainder of this chapter is organized as follows. Section 4.3 introduces the real-world optimization problem considered and the corresponding SBO model developed in this study. Explicit averaging, implicit averaging and a hybrid strategy combining both approaches are described in Section 4.4. Section 4.5 presents details about the comparative analysis, and empirical findings are presented in Section 4.6. Overall conclusions, limitations of this study and future research directions are discussed in Sections 4.7 and 4.8, respectively.
4.3 Simulation-Based Optimization Model

In this study, a SBO approach based on the integration of DES and a GA is applied to address the production planning problem of a real manufacturing company presented in Chapter 3 with the difference that, here, the objective is to achieve profit maximization. Additional modifications made to the original DES and optimization models presented in Chapter 3 are stated in Sections 4.3.1 and 4.3.2, respectively.

This problem corresponds to a big bucket, multi-product, multi-level (sub-products), capacitated (constraints are considered) production planning problem of a failure-prone manufacturing system, consisting of multiple production lines with insufficient capacity to fully cover demand requirements.

The DES model was developed in SimEvents® (The MathWorks, Inc., 2014) and Matlab’s GA (MI-LXPM) implementation (Deep et al., 2009) was used as the optimizer. This is the default Matlab® R2014a’s (The MathWorks, Inc., 2013) implementation for solving integer and mixed integer problems with GA. This GA employs Laplace crossover (crossover probability: 0.8), power mutation (mutation probability: 0.005) and binary tournament selection as operators. Information about population size and number of generations is presented in Section 4.4. The truncation procedure, which ensures compliance with integer constraints after crossover and mutation, is described in Deep et al. (2009). The inbuilt constraint-handling method is the parameter free penalty function approach proposed by Deb (2000). In this GA, elitism is implemented by having an elite set of 1. Please note that elite individuals are re-evaluated during the next generation, according to Matlab customer service.

All computations were executed in parallel on a 12 core Intel(R) Xeon(R) CPU L5640 @ 2.27GHz with 24 GB of RAM running Scientific Linux, release 6.2.
The DES model employed in this study is a modified version of the model presented in Chapter 3.

The DES model represents the production of 31 products $j$ within 7 production lines $l$. A production line corresponds to the set of resources (e.g., machines, people, etc.) needed to manufacture certain products. Given that some products can be manufactured in several production lines, a total of 41 processes $y$ are considered in the DES model. A process $y$ includes all series of events involved in the initialization of orders of a product $j$, its manufacturing in a specific production line $l$ and its storage in a specific container, denoted as $\text{sink}$ (with $\text{sink} = 1, 2, \ldots, 41$).

This model intends to capture the delays caused by production line failures and provides the stock of products manufactured during a production period of one month (24 days composed of 3 shifts of 8 hours each). The total stock of a specific product ($s_j$) corresponds to the sum of lots manufactured across the different production lines as shown by the following equation:

$$s_j = \sum_{l=1}^{7} s_{l,j},$$

where $s_{l,j}$ is the stock of product $j$ manufactured in production line $l$.

The first modification to the original simulation model is that here the manufacturing time of a product lot is treated as a constant when no failure occurs, rather than as a random variable. This is according to specifications provided by the company.

Moreover, we consider the occurrence of production line failures that are independent of the product type. The occurrence of a failure in a production line is modelled here by a random variable $H_l$, whose numerical values $h_l$ are sampled from a probability mass function (PMF) with sample space $\Omega_H = \{0, 1\}$, where $h_l = 1$ represents the occurrence of a failure and $h_l = 0$ indicates that no failure occurred. The probability
that \( h_l = 1 \) is \( p_{H_l}(1) = p_l \); consequently, \( p_{H_l}(0) = 1 - p_l \). Here, numerical values of \( H_l \) are sampled every time a product lot is manufactured; therefore, \( p_l \) is the probability that a production line fails during the manufacturing of a product lot. After a production line failure, a repair service needs to be carried out. The delay \( \lambda_l \) caused by a repair service is the numerical realization of a random variable \( \Lambda_l \), modelled by an exponential probability density function (PDF) with known mean \( \mu_l \).

Finally, an additional server was added to each process in the original simulation model, so that the first lot of every process is processed by this server during the entire duration of each simulation replication (24 days). This modification was made to allow decision variables to take values equal to zero, a possibility not accounted for in the original model (Chapter 3).

### 4.3.2 Optimization Model

The objective here is to generate production plans that try to maximize the expected sum of contributions to profit generated from processes undertaken by a failure-prone manufacturing system. The expected sum of contributions to profit is later referred to as “profit” for simplification purposes.

A total of 41 decision variables \((x_y)\) are considered, which correspond to the number of lots to be produced in each process \( y \), and are constrained to be non-negative integers. Those decision variables, specified by the GA, constitute the input to the DES model which outputs the realization of those decision variables, denoted here as \( s_j \). Those simulated responses \((s_j)\) are used to compute the fitness value of each production plan \( x \) across \( \gamma \) independent simulation replicates as follows:

\[
\text{maximize} \quad f(x) = \frac{1}{\gamma} \sum_{r=1}^{\gamma} \sum_{j=1}^{31} v_j \times s_{j,r},
\]

where \( v_j \) corresponds to the contribution margin per lot of product \( j \) and the value of \( \gamma \)
varies depending on the strategy applied (see Section 4.5 for details about $\gamma$).

Additional constraints are imposed in the form of Equation 4.3 to avoid production levels greater than the maximum demand, to represent the requirement of sub-products and labour needed to undertake each process.

$$\sum_{y=1}^{44} a_{i,y} \times x_y \leq b_i \quad (i = 1, 2, \ldots, 44),$$  \hspace{1cm} (4.3)

where $b_i$ denotes the magnitude of constraint $i$ and $a_{i,y}$ corresponds to the amount deployed from $b_i$ by manufacturing one lot in process $y$.

### 4.4 Noise Handling Strategies

In this study we focus exclusively on implicit and explicit averaging strategies, as these are straightforward to implement in any EA and present the approaches most commonly employed in practice. Other noise handling strategies, such as averaging by means of approximated models (Branke et al., 2001; Neri et al., 2008; Sano et al., 2000) and modifications of the selection scheme (Branke and Schmidt, 2003; Rudolph, 2001), have been proposed in the literature, but are not considered here.

An explicit averaging strategy uses a fixed number $\gamma$ of simulation replicates to obtain an average fitness value for each individual, as described in Equation 4.2. These average fitness estimates are then used to inform selection probabilities in the EA. Therefore, under the explicit averaging strategy (ES) here analysed, fitness of an individual $x$ is computed across 10 independent fitness evaluations ($\gamma = 10$) and the population size employed corresponds to 50 individuals. $\gamma$ is set at this relatively small level because a limited computational budget of 25000 fitness evaluations is available, and assigning higher values to $\gamma$ or using a larger population size would reduce the number of generations that can be executed under ES.
In contrast, an implicit averaging strategy uses a single simulation replicate \((\gamma = 1)\) to evaluate the quality of an individual. Increased robustness towards noise is then achieved by increasing population size relative to standard settings of this parameter. Consequently, under the implicit averaging strategy (IS), a single fitness evaluation \((\gamma = 1)\) is used to compute fitness and a population size of 100 individuals is applied.

Additionally, a baseline strategy (BS) is also analysed in order to evaluate the performance of implicit averaging when the population size is the same as in ES, and therefore, twice as many generations are evolved compared to IS.

Moreover, we further describe a hybrid strategy (HS) that attempts to combine aspects of implicit and explicit averaging. This strategy applies implicit averaging \((\gamma = 1\) and a population size of 100 individuals) throughout the evolution process, but switches behaviour towards the end of the optimization: instead of choosing the final solution based on a single fitness value, we propose to select from the final population the feasible individual with the best average fitness. Consequently, we implement a mechanism that computes the average fitness of every feasible individual of the final population across a number \(\gamma\) of fitness evaluations, generated from independent simulation replications. The value of \(\gamma\) depends on the number of feasible individuals \((\delta)\) in the final population and on the computational budget available for this last step \((E)\), which in this case corresponds to 1000 fitness evaluations. \(\gamma\) is computed as follows:

\[
\gamma = \left\lfloor \frac{E}{\delta} \right\rfloor. \tag{4.4}
\]

Therefore, having a population size of 100, if every individual in the final population is feasible, 10 fitness evaluations are used to compute the average fitness of each individual. However, if unfeasible individuals are present in the final population, those 1000 fitness evaluations are distributed amongst feasible individuals only.

Parameters specified for all four noise handling strategies introduced in this section
are presented in Table 4.1.

<table>
<thead>
<tr>
<th></th>
<th>BS</th>
<th>IS</th>
<th>ES</th>
<th>HS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\gamma$</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>Population Size</td>
<td>50</td>
<td>100</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>Generations</td>
<td>500</td>
<td>250</td>
<td>50</td>
<td>240</td>
</tr>
<tr>
<td>Fitness evaluations</td>
<td>25000</td>
<td>25000</td>
<td>25000</td>
<td>≤ 25000</td>
</tr>
</tbody>
</table>

Table 4.1: Parameters used for baseline, implicit averaging, explicit averaging and hybrid strategies.

### 4.5 Comparative Analysis

In order to test the effectiveness of the strategies under different levels of fitness variability, the following comparative analysis is undertaken for two different problem instances. Table 4.2 shows the different levels of uncertainty incorporated into each instance of the problem. According to the company analysed, capacity is a limiting factor only in production line 2, 3 and 4, where 26 out of the 31 products offered by this company are manufactured (see Table 2.1 for more details). The excess of capacity in production line 1, 5, 6 and 7 can be used to compensate the capacity loss caused by the occurrence of failures. This is something that cannot be done in production line 2, 3 and 4, and thus an accurate consideration of the uncertainty around failures and repairs in production line 2, 3 and 4 is much more relevant than in production line 1, 5, 6 and 7. For this reason and because failures in production line 1, 5, 6 and 7 rarely or never occurred (see Table 2.5 in Chapter 2), we assign a positive $p_l$ to production line 2, 3 and 4 and assume that production line 1, 5, 6 and 7 are fully reliable in this and in subsequent chapters of this thesis.

It is important to mention that the lack of failures in the historical records of a given production line does not imply that it is fully reliable. In situations when no or very limited information is available about the occurrence of failures of specific production
lines, statistical shrinkage estimators (Copas, 1983) are interesting approaches to estimate failure rates of individual lines, as they use the information available from other production lines to find those estimates. Some recent examples are presented in Xiao and Xie (2014) and Vaurio and Jänkälä (2006). The estimation of $p_l$ values via shrinkage estimators would capture a more realistic situation of this system; however, this is not implemented here as it is beyond the scope of this work. Having a more realistic representation of the system won’t have a major impact on the results reported in this chapter, but it might increase the performance gap between the exact optimization techniques and SBO approaches benchmarked in the next chapter (Chapter 5).

<table>
<thead>
<tr>
<th>Production line ($l$)</th>
<th>$p_l$</th>
<th>$p_l$</th>
<th>$\Lambda_l$</th>
<th>PDF</th>
<th>$\mu_l$ (d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>2</td>
<td>0.10</td>
<td>0.30</td>
<td>Exponential</td>
<td>0.0847</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.25</td>
<td>0.45</td>
<td>Exponential</td>
<td>0.0935</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.15</td>
<td>0.35</td>
<td>Exponential</td>
<td>0.1338</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.00</td>
<td>0.00</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>6</td>
<td>0.00</td>
<td>0.00</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>7</td>
<td>0.00</td>
<td>0.00</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

Table 4.2: $p_l$ per problem instance and PDFs specifications to model $\Lambda_l$.

The performance of the EA under the proposed HS is compared with the performance observed for IS, ES and BS. In order to provide a fair comparison of the four strategies analysed, the stopping criteria selected to terminate the optimization procedure is the number of fitness evaluations. As mentioned in Section 4.4, a total budget of 25000 fitness evaluations is allocated for every strategy as shown in Table 4.1.

The SBO model is run 60 different times for each strategy and we use in every run different seeds for the random number generator. In each run, the best solution (based on fitness values) is selected from the final population. Consequently, 240 production plans are generated per problem instance. The precise quality of each of these plans is
evaluated using extensive simulation: average profit, measured in United States Dollar (USD), is computed for every production plan across 1000 profit values obtained via stochastic simulation.

Subsequently, the four sets of average profit values are depicted as cumulative distribution functions (CDFs) and stochastic dominance criteria (Yitzhaki, 1982) is applied to determine whether or not the optimization performance, as measured in average profit values, differs between strategies.

Furthermore, Mann-Whitney U test (Mann and Whitney, 1947) is then conducted for paired comparisons to test whether the optimization performance achieved under the different strategies is statistically significant, expressed in the form of the following hypotheses:

\[ H_0 : \text{stochastic homogeneity of CDFs of average profit values obtained under both strategies} \]

\[ H_a : \text{average profit values obtained under one strategy are stochastically smaller than the ones obtained under the other strategy} \]

Mann-Whitney U test is employed instead of t-test, since distributions of the samples analysed do not fulfil the normality assumption.

### 4.6 Results

Descriptive statistics (means, minimum values, maximum values and standard deviations) of the average profit values (as computed across 1000 independent replications) obtained under the four strategies as well as the corresponding average computation times are presented in Tables 4.3 and 4.4 for problem instance 1 and 2, respectively.

Since our intention is to test the hypothesis presented in section 4.5 among the four strategies, homogeneity of variances of the ranked values across the different samples is a necessary condition for Mann-Whitney U test to be a reliable test (Ghasemi
et al., 2014). Therefore, non-parametric Levene tests (Nordstokke and Zumbo, 2010) were performed on every combination of samples in both problem instances. In both problem instances, results from these tests indicate that variances did not differ significantly \( (p > .05) \) between the samples of ranks analysed, confirming the suitability of Mann-Whitney U test to evaluate the hypothesis above mentioned (Section 4.5).

Figures 4.1 and 4.2 illustrate as CDFs the 60 average profit values obtained with production plans generated under each strategy in problem instance 1 and 2, respectively. Both figures clearly show that the CDFs of average profit obtained under BS, IS and HS dominate the CDF of profit obtained under ES (first-order stochastic dominance). Furthermore, results from Mann-Whitney U test statistically show that average profit values obtained under ES are stochastically smaller \( (p < .01) \) than the ones obtained under BS, IS and HS in both problem instances, as shown in Tables 4.5 and 4.6.

These results demonstrate that ES is an inadequate noise handling strategy in our

<table>
<thead>
<tr>
<th></th>
<th>BS</th>
<th>IS</th>
<th>ES</th>
<th>HS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean (USD)</td>
<td>703689</td>
<td>715376</td>
<td>555932</td>
<td>707503</td>
</tr>
<tr>
<td>Minimum (USD)</td>
<td>550417</td>
<td>484229</td>
<td>460416</td>
<td>550014</td>
</tr>
<tr>
<td>Maximum (USD)</td>
<td>793316</td>
<td>795716</td>
<td>689685</td>
<td>792696</td>
</tr>
<tr>
<td>Std dev (USD)</td>
<td>55539</td>
<td>60416</td>
<td>47322</td>
<td>60754</td>
</tr>
<tr>
<td>Average computation time (s)</td>
<td>1256</td>
<td>1144</td>
<td>1235</td>
<td>1369</td>
</tr>
</tbody>
</table>

Table 4.3: Descriptive statistics of average profits and average computation times per strategy in problem instance 1.

<table>
<thead>
<tr>
<th></th>
<th>BS</th>
<th>IS</th>
<th>ES</th>
<th>HS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean (USD)</td>
<td>707600</td>
<td>703420</td>
<td>543140</td>
<td>723460</td>
</tr>
<tr>
<td>Minimum (USD)</td>
<td>536840</td>
<td>538990</td>
<td>437930</td>
<td>561550</td>
</tr>
<tr>
<td>Maximum (USD)</td>
<td>785810</td>
<td>783040</td>
<td>656800</td>
<td>790460</td>
</tr>
<tr>
<td>Std dev (USD)</td>
<td>60137</td>
<td>61744</td>
<td>43412</td>
<td>51258</td>
</tr>
<tr>
<td>Average computation time (s)</td>
<td>1284</td>
<td>1108</td>
<td>1214</td>
<td>1265</td>
</tr>
</tbody>
</table>

Table 4.4: Descriptive statistics of average profits and average computation times per strategy in problem instance 2.
setting: this result is likely to be driven by the limited computational budget available, and a stronger performance of ES may potentially be achieved when considering performance upon convergence.

No dominance (neither first nor second degree stochastic dominance) can be determined among the CDFs of average profit obtained under BS, IS and HS in problem instance 1, where all three strategies appear equally competitive. These results are in accordance with results from Mann-Whitney U test, which indicate stochastic homogeneity ($p > .05$) among samples obtained under BS, IS and HS in problem instance 1. It is interesting that population size (i.e. different setups of implicit averaging) has no significant effect in this setting as evidenced under BS and IS.

In problem instance 2, results from Mann-Whitney U tests also indicate stochastic homogeneity ($p > .05$) among samples obtained under BS, IS and HS. However, the CDFs of average profit obtained under HS dominates the CDFs of average profit obtained under IS and under BS, respectively (first and second-order stochastic dominance). These results indicate that, even in a setting with a limited evaluation budget,

<table>
<thead>
<tr>
<th></th>
<th>BS</th>
<th>IS</th>
<th>HS</th>
<th>ES</th>
</tr>
</thead>
<tbody>
<tr>
<td>BS</td>
<td>1525</td>
<td>1705</td>
<td>107**</td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>1662</td>
<td>117**</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HS</td>
<td>113**</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ES</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

** $p < .01$

Table 4.5: Values for Mann-Whitney U statistic obtained in problem instance 1.

<table>
<thead>
<tr>
<th></th>
<th>BS</th>
<th>IS</th>
<th>HS</th>
<th>ES</th>
</tr>
</thead>
<tbody>
<tr>
<td>BS</td>
<td>1719</td>
<td>1549</td>
<td>103**</td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>1485</td>
<td>96**</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HS</td>
<td>36**</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ES</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

** $p < .01$

Table 4.6: Values for Mann-Whitney U statistic obtained in problem instance 2.
the accurate fitness estimates from explicit averaging can be beneficial to optimization. It is clear that the difference between IS and HS arises during the final stages of the optimization only, while IS continues optimization during additional generations, HS focuses resources on explicit averaging across its last population. When seen in combination with the poor performance of ES, our results suggest that the trade-off between improved exploration (from evaluating more individuals) and accurate fitness evaluations (through simulation replicates for the same individual) needs to be carefully balanced in this setting. This finding appears in line with previous research that has delivered contradictory results regarding the relative performance of explicit and implicit averaging.

Figure 4.1: CDFs of average profit values obtained with production plans generated under the four different strategies in problem instance 1.
Implicit averaging strategies reduce the impact of noise by having a sufficiently large population, which ensures that individuals from promising regions of the search space are sampled repeatedly (Jin and Branke, 2005; Syberfeldt et al., 2010). On the other hand, explicit averaging strategies use average fitness values, obtained across a specific number of fitness evaluations, to ensure that evaluation of individuals is based on fitness estimates that are more robust to noise (Jin and Branke, 2005).

One of the key findings of this study was that, in the context of our real-world problem, a noise-handling strategy based on explicit averaging did not provide a competitive performance. More generally, this points to the fact that the computational costs incurred by simulation replicates may be problematic in constrained time settings.

Furthermore, we found that implicit averaging performed robustly for both of the
population sizes used. The performance of our hybrid strategy does indicate that some effort towards explicit averaging may become important with increasing variability. Under low levels of fitness variability, the hybrid strategy, implicit averaging and our baseline showed a comparable performance. This situation changed with increasing levels of fitness variability, when HS started to enhance overall performance.

Compared to a pure, implicit averaging strategy, the hybrid strategy misses out on the last few generations of optimization. Our results show, however, that this disadvantage is more than counter-balanced by the benefits from an accurate final selection step that reduces the likelihood of choosing an inferior individual (in terms of average fitness) as the final solution.

4.8 Limitations and Future Research

The relevance of obtaining more reliable fitness estimates increases with the level of variability, since there is a higher risk of choosing an inferior solution. It is, therefore, intuitive that the final selection mechanism implemented in HS would be more beneficial in such circumstances. But at the same time, the number of fitness evaluations needed to obtain reliable estimates is expected to raise with higher fitness variability, leaving to the evolutionary process a smaller share of the computational budget. Therefore, further research may focus on investigating the right trade-off between exploration (IS) and accurate fitness evaluation (ES). In this sense, the application of different sampling techniques (e.g. Latin Hypercube) during the final selection mechanism might be worthy of future investigation, as it may allow a reduction in the number of fitness evaluations required in this last step.

Our results underline issues around the computational cost of explicit averaging, but also highlight that sporadic use of this strategy may, nevertheless, be beneficial. In this study, the use of explicit averaging in the hybrid strategy was limited to the
final selection step only. Future research may consider the possibility of using explicit averaging at earlier points during the optimization process.
Chapter 5

Integrating Meta-Heuristics and Exact Optimization Techniques (Manuscript 3)


5.1 Abstract

In this chapter, we are concerned with the optimization of production plans in batch manufacturing systems that include uncertainties, e.g. due to the possibility of production line failures and uncertainty around repair times. In order to account for more realistic situations, here we consider penalties imposed to deviations from a plan, which is something that was not analysed in previous chapters. We also provide a general formulation of this problem as an extended knapsack problem and describe a solution approach based on SBO. Specifically, DES is used to model the existing uncertainty
of the manufacturing process and to evaluate the average profit for any given production plan. A GA is used to optimize across the space of possible production plans, and we improve the algorithm’s optimization performance through the development of specialized initialization operators that make use of mathematical programming in combination with DES. We use our own GA implementation in order to have full control of all parameters, something that was not possible with the default Matlab implementation used in Chapter 3 and 4. The final models are benchmarked against ILP and CCP within the context of a real-world production planning problem. We find that our SBO approaches significantly outperform those exact optimization techniques under the different levels of uncertainty analysed.

5.2 Introduction

The production planning process is an essential part of manufacturing strategy, since it enables companies to achieve business objectives. In this study we address the production planning problem of a real manufacturing company with insufficient capacity to fully cover demand requirements. This company has to purchase the materials needed for the next working month, before the beginning of the production period, and thus a production plan for the following month must be available in order to make adequate purchasing decisions. This is a difficult task because by the time a production plan needs to be developed specific due dates of orders are still unknown and only demand forecasts are available. Since due date information is not yet available during the specification of a production plan, scheduling decisions are not considered here (this is not a scheduling problem).

Another feature of this system that further complicates the development of production plans is the failure-prone nature of production lines. The occurrence of production line failures reduces the total amount of products that can be manufactured
and ultimately sold. Therefore, this company aims to develop production plans that are not only profitable, but also plans that perform robustly under different realization of breakdown events. Once a production plan has been decided and a failure occurs, corrective actions cannot be implemented (at the operational level) without facing negative consequences. Here we try to incorporate those consequences into the problem by penalizing any deviation from a production plan. Therefore, not only profitability and system constraints have to be considered during the specification of a production plan, but also the uncertainty around the occurrence of failures in production lines.

Given the uncertainty inherent to the system, addressing this problem through mathematical programming approaches would require a number of assumptions that may be overly stringent, and therefore, affect the validity of the resulting solutions (Nikolopoulou and Ierapetritou, 2012; Gnoni et al., 2003). This is one of the reasons why obtaining production plans under realistic assumptions has been a difficult challenge for manufacturing companies. Overcoming this challenge is essential, especially in the context of failure-prone manufacturing systems, where the occurrence of failures in production lines can compromise the development of a production plan. Ignoring the presence of uncertainty and including oversimplifying assumptions during the problem formulation will inevitably lead to performance levels that deviate from the original plan (Goh and Tan, 2009, 190). The degree of those deviations will mainly depend on the levels of uncertainty present in the system or/and on the appropriateness of the assumptions made.

In such cases, SBO is an interesting alternative because complex system features can be explicitly and accurately incorporated into the problem via simulation (i.e. not requiring a closed-form formulation), while an optimizer tries to find near-optimal solutions. Therefore, in this study we present an effective SBO approach that combines DES together with a meta-heuristic optimizer. Through this approach it is possible to incorporate via DES the uncertainty derived from production line failures,
while optimization is performed by a GA. The optimization performance of this GA is
boosted by specialized initialization operators that combine DES, deterministic ILP
and CCP (Charnes and Cooper, 1959) in a variety of ways, as described in Sec-
tion 5.5.2. We demonstrate that our approach is able to generate better solutions to
the real-world problem analysed than the ones obtained via separate application of
meta-heuristics or mathematical optimization.

The remainder of this chapter is organized as follows: a review of relevant literature
and the contributions of this chapter are presented in Section 5.3. In Section 5.4, we
provide a general formulation of the problem analysed as an extended knapsack prob-
lem in order to demonstrate the wider applicability of our approach. In Section 5.5,
we describe a solution approach based on SBO. We also introduce a repair operator
which tries to fix unfeasible solutions generated during the optimization as well as two
specialized initialization operators which use a combination of mathematical program-
ming techniques (ILP and CCP) and DES to boost the optimization performance of the
SBO model. In Section 5.6, we address a real-world production planning problem of
a failure-prone manufacturing system to benchmark our models against ILP, CCP and
SBO without specialized initialization operators. Finally, conclusions derived from
this study, limitations and future research directions are given in Sections 5.7 and 5.8,
respectively.

5.3 Literature Review

The majority of studies in the existing literature on production planning of failure-
prone manufacturing systems are focused on finding optimal solutions through math-
ematical programming approaches. Those approaches; however, usually address ide-
alized cases where optimality conditions can be satisfied. For instance, Kouedeu et al.
(2014a) presented a hierarchical approach to determine production rates along with
corrective and preventive maintenance policies. They aimed to minimize the discounted overall cost of a system that manufactures a single product in a single machine, subject to random failures and that deteriorates with the number of failures. Kouedeu et al. (2014b) formulated stochastic dynamic programming equations to determine production rates for a single product manufactured by two machines, one with production-dependent failure rates and the other with constant failure rates, with the aim of minimizing inventory and shortage costs over an infinite time horizon. In a very similar study, Kouedeu et al. (2014c) applied the same approach presented in Kouedeu et al. (2014b) to minimize the discounted overall cost by specifying production rates for a manufacturing and for a re-manufacturing machine that produce a single product. Shi et al. (2014) proposed a discrete Markovian production model to determine, at every customer arrival, the production rate and selling price (low or high) of a single product manufactured by an unreliable machine, based on its inventory level.

Identifying a global optimal solution for real-world problems may often be unrealistic, due to the inherent complexity and uncertainty of real systems (Lacksonen, 2001), and thus practitioners are often satisfied with a good (but not necessarily optimal) solution that can be implemented in practice (Blum and Roli, 2008). This has motivated the use of meta-heuristics in this area. For instance, Dahane et al. (2012) addressed a multi-period multi-product (MPMP) production planning problem where in each period a single machine, with production-dependent failure rate, first manufactures a product covering strategic demand and then a second product covering secondary demand. The authors applied a GA to simultaneously determine the production rate of the first product as well as the duration of the production interval allocated in each period to the manufacturing of the second product, in order to maximize the total expected profit.

Another interesting alternative for this area is SBO. A smaller proportion of literature in this area has applied simulation in combination with optimization methods to
find (near-optimal) solutions to more complex cases. For instance, Kenne and Gharbi (2001) employed a combination of stochastic optimal control theory, DES, experimental design and response surface methodology to determine near-optimal production rates, for a multi-product manufacturing system with parallel machines subject to failures and repairs, that minimize the expected discounted cost of inventory/backlog over an infinite time horizon. Kenne and Gharbi (2004) determined near-optimal production and machine repair rates of a multi-product and multi-machine manufacturing system by applying the method proposed in Kenne and Gharbi (2001). Kenne and Gharbi (2004) and Kenne and Gharbi (2001) are two papers where DES was used to represent machine failures and repairs, and where the application of simulation in combination with optimization enabled the analysis of more complex situations. Sel and Bilgen (2014) iteratively applied mixed integer programming (MIP) based “Fix and Optimize” heuristic to determine production and distribution plans and then used DES to adjust capacities of production lines.

5.3.1 Simulation-based optimization

Over the past few decades, there has been a dramatic increase in the number of studies applying SBO to address different real-world problems (Korytkowski et al., 2013). Studies applying a combination of simulation and mathematical programming are very popular in the existing literature. For instance, Byrne and Bakir (1999) proposed a hybrid approach that combined LP and simulation to iteratively adjust the right-hand side (RHS) of capacity constrains in order to obtain feasible solutions for a MPMP production problem. Hung and Leachman (1996) applied a similar approach, but to modify the left-hand side of capacity constraints of a semiconductor manufacturing system, based on the results for production flow times obtained via DES. Kim and Kim (2001)
combined the two former approaches to determine both, the amount of total workload per machine as well as the actual machine capacity utilized by the production plan in each period. Byrne and Hossain (2005) incorporated the unit load concept into the model proposed by Kim and Kim (2001) to make it suitable for just in time production (Monden, 2011, p. 8). Lee and Kim (2002) and Safaei et al. (2010) used a combination of mixed ILP (MILP) and simulation to address a multi-site MPMP production and distribution planning problem. Almeder et al. (2009) tried to obtain robust production, stocking and transportation plans to support supply chain decisions at the operational level by applying a combination of DES and MILP. Ehrenberg and Zimmermann (2012) presented an approach where input parameters of a MILP model are specified iteratively via DES, to determine the scheduling of a make-to-order manufacturing system. Arakawa et al. (2003) successfully solved a job shop scheduling problem by implementing an optimization oriented method combined with simulation. Monostori et al. (2010) applied a branch and cut algorithm to find solutions to a medium-term production planning problem and to a short-term scheduling problem, and then used DES to assess the sensitivity of the deterministic production schedules and improve their robustness by supporting re-scheduling decisions.

Meta-heuristics and simulation techniques have also been combined, but to address more complex problems where assumptions needed by mathematical programming approaches cannot be satisfied. For instance, Kämpf and Köchel (2006) combined a GA with an event-oriented simulation model to determine sequencing and lot-sizing rules for a multi-item production system with limited storage capacity. Li et al. (2009) used a simulation model together with a cell evaluated GA to optimize resource allocation, inventory and production policies for a dedicated re-manufacturing system. Merkuryeva et al. (2010) combined stochastic simulation and multi-objective Pareto-based GA together with response surface method-based linear search to determine cycles and order-up-to levels of cyclic planning policies in multi-echelon supply chains.
Gansterer et al. (2014) investigated different SBO approaches to determine appropriate settings for planned leadtime, safety stock and lotsizing in a make-to-order environment, and concluded that a combination of DES with optimization procedures using variable neighbourhood search provided better results than other SBO approaches analysed. Taleizadeh et al. (2013) proposed a combination of fuzzy simulation and a GA to solve a multi-period inventory control problem with stochastic replenishment and stochastic period length, for multiple products with limited storage and fuzzy customer demand and showed that this method outperformed a combination of fuzzy simulation and simulated annealing. Almeder and Hartl (2013) used a DES model as objective function and proposed a variable neighbourhood search-based solution approach for an off-line stochastic flexible flow-shop problem with limited buffers. Hong et al. (2013) proposed a SBO method that combined continuous and discrete simulation with simulated annealing and meta-models to find optimal design configurations. In their approach the meta-models reduced the search space by providing good initial solutions and then the meta-heuristic optimizer tried to improve those solutions towards the optimum. Köse et al. (2015) presented a SBO approach where three meta-heuristic optimizers (binary GA, binary-simulated annealing and binary-tabu search) were integrated with a simulation model to solve a buffer allocation problem in a heat exchanger production plant.

More specifically, the integration of DES and GAs has been successfully deployed in several areas. For instance, Azzaro-Pantel et al. (1998) achieved efficiency improvements of a multi-purpose, multi-objective plant with limited storage. The authors applied DES and a GA to accurately model the dynamic behaviour of the production system and to solve the scheduling problem, respectively. It has also been applied to determine robust design parameters as presented by Al-Aomar (2006). In order to enhance the selection scheme, the author incorporated Taguchis’s robustness measures into the GA. The integration of DES and GAs has also been deployed to address other
problems such as the one presented by Ding et al. (2005), where the uncertainty involved in the supplier selection process was captured via DES and a GA was used to optimize the supplier portfolio. Cheng and Yan (2009) applied an integration of DES and a messy GA to determine the near optimal combination of resources in order to enhance the performance of construction operations. This approach enabled the authors to cope with the complexity and large dimensionality of the problem. Wu et al. (2011) integrated DES with a GA to determine the order point for different product types of a cross-docking center in order to minimize total cost. Through this approach the solution space was efficiently reduced and more simulation effort was allocated to promising regions via smart computing budget allocation. Korytkowski et al. (2013) proposed an evolutionary simulation-based heuristics, where DES and a GA were deployed to find near optimal solutions for dispatching rules allocation. The sequence of orders determined through this approach improved the performance of a complex multi-stage, multi-product manufacturing system.

The success of SBO approaches in addressing complex real-world problems motivates us to develop a SBO approach to tackle the production planning problem analysed in this study. However, rather than combining simulation with a single optimization technique, here we also take advantage of mathematical programming to enhance the performance of our meta-heuristic optimizer, namely a GA. Thus in this chapter, we highlight the synergies resulting from the integration of simulation techniques with what could be seen as a matheuristic (Villegas et al., 2013; Boschetti et al., 2009) optimizer. We test this approach in the context of a real-world production planning problem, for which we provide a general formulation. We account for the uncertainty in the problem via simulation and use the simulation model as our solution evaluation.

We use DES as our simulation technique due to its ability to incorporate stochastic events (Riley, 2013) and to represent functional relationships between variables that are not explicitly known or for which no analytical formulation exists (Steponavičė et al.,
2014). As mentioned above, meta-heuristics have been applied to stochastic problems where the solution evaluation is performed via simulation (across multiple replications) and are commonly used as optimizers in DES software (Figueira and Almada-Lobo, 2014). The ability of GAs to find near-optimal solutions in large, complex and discrete solution spaces as well as their robust performance under noisy conditions reported in previous studies (Mitchell, 1998; Baum et al., 1995), especially in optimization of DES (Lacksonen, 2001), motivate us to use a GA as our optimizer, although a different choice of meta-heuristic would also be suitable.

5.3.2 Contributions

The contributions of this chapter are the following: (i) we introduce a generalization of a real-world production planning problem as a form of knapsack problem, (ii) we propose two SBO approaches able to address that problem and (iii) demonstrate their effectiveness under different uncertainty levels through a benchmark analysis performed against ILP and CCP. However, we do not only provide evidence of the poor performance of both exact optimization techniques, (iv) but more importantly we demonstrate that ILP and CCP can be used to significantly improve the optimization performance of our SBO approach. More specifically, we illustrate that the implementation of specialized initialization operators that draw upon the solutions returned by ILP and CCP combined with DES in a variety of ways is able to significantly improve the optimization performance of a standard GA. In general, this chapter highlights the synergies resulting from the combination of simulation, mathematical programming and meta-heuristic methods.
5.4 Problem Description

The problem analysed in this study can be formulated as a knapsack problem (KP); therefore, we review here existing variants of the KP and in Section 5.4.1 we introduce two new variants.

Given a knapsack $l$ with capacity $c$ and an item set $N$, consisting of $n$ items, with item $j$ characterized by a value $v_j$ and a weight $w_j$, then the objective in the classical KP (Kellerer et al., 2004, p. 5) is to decide what items are going to be packed into the knapsack, such that the total value of items chosen is maximized and the total sum of their weights does not exceed the knapsack capacity. Therefore, decision variables $x_j$ indicate whether an item is chosen to be packed ($x_j = 1$) or not ($x_j = 0$). This problem can be formulated as an ILP problem as follows:

$$\text{maximize} \quad f(\mathbf{x}) = \sum_{j=1}^{n} v_j x_j$$

subject to

$$\sum_{j=1}^{n} w_j x_j \leq c,$$

$$x_j \in \{0, 1\} \quad (j = 1, 2, \ldots, n),$$

where $\mathbf{x}$ is a vector that contains all decision variables $x_j$.

In a production setting, $N$ corresponds to the portfolio of products and $\mathbf{x}$ is the production plan to be executed in production line $l$ with capacity $c$. Each product $j$ has a specific marginal profit $v_j$ and its manufacturing consumes a specific amount $w_j$ from capacity $c$.

However, if $N$ includes a finite number $b_j$ of identical copies of an item $j$, a more efficient way to model this problem is to represent the entire set of identical items as an integer decision variable, which indicates the number of items $j$ to be packed into
the knapsack. By doing this, the number of decision variables is equal to the number of different item types and not to the number of items, as is the case in the classical KP. This extension of the classical KP is known as the bounded KP (BKP) (Kellerer et al., 2004, p. 185), since upper bounds \( b_j \) are established for decision variables \( x_j \), which only take integer values. Thus Equation 5.3 in the classical KP needs to be replaced by the following:

\[
0 \leq x_j \leq b_j \quad (x_j \in \mathbb{Z}; j = 1, 2, \ldots, n),
\]

(5.4)

to obtain the ILP formulation for the BKP. In the context of a batch manufacturing system, the BKP considers the manufacturing of several lots of the same product and imposes upper bounds \( b_j \) to the decision variables \( x_j \) (e.g. maximum level of demand of a specific product \( j \)).

Unlike KP or BKP where only a single resource constraint is considered (e.g. maximum weight that the knapsack can carry), in more realistic situations a problem may be subject to not only one, but multiple resource constraints (e.g. maximum volume of the knapsack, shape of items). In manufacturing, for instance, it is necessary to explicitly consider the capacity of a production line, the amount of labour required to manufacture a product, the requirement of raw materials, among others. Another variant of the KP, the \( d \)-dimensional or multidimensional KP (\( d \)-KP) (Kellerer et al., 2004, p. 6), is able to capture such restrictions by including a set \( D = \{1, 2, \ldots, d\} \) of constraints into the problem formulation. In this case, Equation 5.2 needs to be replaced by:

\[
\sum_{j=1}^{n} w_{i,j} \times x_j \leq c_i \quad (i = 1, 2, \ldots, d),
\]

(5.5)

to obtain the mathematical formulation of the \( d \)-KP.

Although, the \( d \)-KP is able to capture more complex features than the classical KP,
it is still unable to model the situation when multiple knapsacks are available. When a
KP includes a set $M = \{1, 2, \ldots, m\}$ of knapsacks, where each knapsack $l$ has a given
capacity $c_l$, and every product $j$ has to be packed at most into one knapsack, it is
known as the multiple knapsack problem (MKP) (Kellerer et al., 2004, p. 7). When
only a subset $A_j \subseteq M$ of knapsacks can hold a specific item $j$, then this problem is
referred to as the MKP with assignment restrictions (MKAR) (Dawande et al., 2000).
In a manufacturing context, MKAR extends the MKP to the situation where multiple
production lines are available and where each production line is able to produce only
certain products. The mathematical formulation of the MKAR is as follows:

$$
\text{maximize } \quad f(x) = \sum_{j=1}^{n} \sum_{l \in A_j} v_j \times x_{l,j}
$$

subject to

$$
\sum_{j=1}^{n} w_j \times x_{l,j} \leq c_l \quad (l = 1, 2, \ldots, m),
$$

$$
\sum_{l \in A_j} x_{l,j} \leq 1 \quad (j = 1, 2, \ldots, n),
$$

$$
x_{l,j} \in \{0, 1\} \quad (l \in A_j; j = 1, 2, \ldots, n).
$$

5.4.1 New Variants of the KP Problem: $d$-MBKAR and $d$-MBKARS

In this section we introduce a deterministic and a stochastic variant of the KP to define
a more general class of problem that can be tackled with our approach. Both vari-
ants generalize features of a real-world production planning problem of a failure-prone
batch manufacturing system, whose details are presented in Chapter 2. In both variants
we relax Equation 5.8, so that several identical copies of item $j$ can be packed into different knapsacks. Moreover, instead of binary decision variables, integer variables are used to represent the number of units of item $j$ to be packed into a specific knapsack $l$. We also set upper bounds $b_j$ for the total number of copies of item $j$ packed across all knapsacks (Equation 5.12). Furthermore, we eliminate the general assumption (made in KPs to avoid trivial situations (Kellerer et al., 2004, p. 10)) that $v_j$, $w_{i,j}$ and $c_i$ can take only positive values, for it precludes the consideration of more complex features present in real-world problems such as the case of complementary items. In this case, complementary items must be packed across knapsacks in a specific proportion to derive value from them. For instance, in manufacturing systems, it is common that some products are employed (in a certain proportion) as raw materials during the manufacturing of another product. Representing these constraints in the form of Equation 5.11 requires that: (i) $v_j$ can take negative values to represent the marginal cost of products used as raw materials, (ii) $w_{i,j}$ can take negative values to represent the yield (per lot) of product $j$ in production line $l$ and positive values to represent the amount of raw materials needed to manufacture one lot of product $j$ and (iii) $c_i$ can be equal to zero.

The ILP formulation of such problem is the following:

$$\text{maximize } f(x) = \sum_{j=1}^{n} \sum_{l \in A_j} v_j \times x_{l,j}$$

subject to

$$\sum_{j=1}^{n} \sum_{l \in A_j} w_{i,j} \times x_{l,j} \leq c_i \quad (i = 1, 2, \ldots, d),$$

$$\sum_{l \in A_j} x_{l,j} \leq b_j \quad (x_{l,j} \in \mathbb{Z}_{\geq 0}; j = 1, 2, \ldots, n).$$

We refer to this deterministic variant of the KP as the multidimensional multiple
bounded knapsack problem with assignment restrictions (d-MBKAR).

However, reality is not deterministic, and thus a stochastic version of the d-MBKAR problem (d-MBKARS) is needed. In the d-MBKARS we consider that knapsacks are subject to random failures during the packing process. Here, uncertainty is incorporated into the problem by considering the reduction in capacity caused by knapsack failures. Therefore, given a packing plan \( x \), the actual number of units of item \( j \) packed into knapsack \( l \) at the end of the packing process, denoted here by \( s_{l,j} \), depends on the number of failures that have occurred during the packing process and on the capacity reduction caused by the corresponding repairs. Moreover, we assume that any deviation from the packing plan \( x \) is subject to a penalty \( (k_j) \) proportional to that deviation, i.e. \( k_j \times (x_{l,j} - s_{l,j}) \), and thus uncertainty needs to be carefully considered during the specification of a packing plan.

The realization of a packing plan, denoted as \( s \) (vector that contains all \( s_{l,j} \)), is obtained here via simulation due to the difficulty of finding a closed-form expression able to map a packing plan \( x \) onto its realization \( s \). In this sense, the simulation model can be seen here as the function \( g(x) \) that enables us to perform such mapping without the requirement of a closed-form expression. Other parameters such as price volatility, demand fluctuations and variability in production yields could also be incorporated into the problem by modelling them via simulation.

The occurrence of a knapsack failure during the packing process of an item is modelled by a random variable \( H_l \), whose numerical values \( h_l \) are sampled (every time an item needs to be packed into a knapsack) from a PMF with sample space \( \Omega_H = \{0, 1\} \). \( h_l = 1 \) represents the occurrence of a failure, whereas \( h_l = 0 \) indicates that no failure occurred. The probability that \( h_l = 1 \) is \( p_{H_l}(1) = p_l \); consequently, \( p_{H_l}(0) = 1 - p_l \). Here, \( p_l \) is the probability that a knapsack fails during the packing process of an item, and thus the number of knapsack failures depends on the number of items to be loaded. After every knapsack failure, a repair service needs to be undertaken. The
reduction in capacity $\lambda_l$ caused by a repair service is the numerical realization of a random variable $\Lambda_l$, modelled by an exponential PDF with known mean $\mu_l$.

The formulation of the $d$-MBKARS is as follows:

$$\text{maximize } f(\mathbf{x}) = \sum_{j=1}^{n} \sum_{l \in A_j} v_j \times s_{l,j} - k_j \times (x_{l,j} - s_{l,j})$$

subject to the set of constraints in the form of Equations 5.11 and 5.12.

Features of the $d$-MBKARS problem translated to a batch manufacturing system could be represented by unreliable production lines that require a repair service after one of its components stopped functioning properly. In this context, $p_l$ is the probability that a failure occurs in production line $l$ during the manufacturing of a product lot, and the delay $\lambda_l$ caused by a repair service is equivalent to the capacity loss in the $d$-MBKARS problem. The reduction in capacity caused by production line failures has the effect that a production plan cannot be fully realized, which means that some products will not be produced on time. Deviations from the original plan may have serious consequences not only on the company’s profitability, but also on its image and reputation. Here, we only consider consequences that can be quantified by a penalty proportional to that deviation.

Note that the RHS of Equation 5.11 is a constant, even with $c_i$ related to design (theoretical) knapsack capacities (Heizer et al., 2004, p. 252). Therefore, a packing plan $\mathbf{x}$ that is feasible according to the set of constraints in the form of Equations 5.11 and 5.12 might not always be realized in the $d$-MBKARS problem. In other words, $x_{l,j}$ is not always equal to $s_{l,j}$ in the $d$-MBKARS because the former is a number, whereas the latter is the numerical value of the random variable $S_{l,j}$.

Although, BKP (Pisinger, 2000), $d$-KP (Chen and Hao, 2014; Balev et al., 2008; Wilbaut et al., 2009), MKP (Garcia-Martinez et al., 2014; Yamada and Takeoka, 2009; Kataoka and Yamada, 2014), MKAR (Dawande et al., 2000) and different stochastic
versions of the KP (Chen and Ross, 2014; Perboli et al., 2014; Dean et al., 2008) have been analysed in the area of combinatorial optimization, to the best of our knowledge, our previous research (Diaz and Handl, 2014, 2015) presented in Chapter 3 and 4 are the only studies that have tried to tackle a problem similar to d-MBKARS. However, in this chapter we penalize deviations from a plan, something that was not considered in previous chapters.

5.5 Simulation-Based Optimization Model

In this section we introduce our SBO approach, in the context of a real-world production planning problem faced by a failure-prone manufacturing system with insufficient capacity to cover demand requirements. This real-world problem has all the features of the d-MBKARS problem and its detailed information is provided in Chapter 2.

This manufacturing company aims to develop production plans which specify the number of lots of the different products and sub-products that every production line needs to manufacture in order to maximize the expected sum of contributions to profit generated during a finite planning horizon of one working month. Apart from complex features such as the multi-product, multi-production line, and multi-level nature of the manufacturing system analysed, we also need to accurately consider the uncertainty derived from failures in production lines. Considering this uncertainty is extremely important because any deviation from a production plan is penalized. Since finding concrete values for those penalties is very complicated and subjective, we assume that any deviation from a plan results in penalties which are proportional to the profit loss caused by that deviation.

As illustrated in Figure 5.1, the SBO model developed in this study is an integration of DES and a GA, which is supported by specialized initialization operators that
combine DES with mathematical programming techniques (see Section 5.5.2). According to the taxonomy provided in Figueira and Almada-Lobo (2014), our model would be classified as an “evaluation function - simulation-based iterations/discrete heuristic-different - realizations for each solution” (EF-OSI/DH-DR1S) model, since here optimization is performed by a meta-heuristic optimizer based exclusively on fitness values computed with responses generated via simulation, across different realizations of integer solutions.

![Diagram](image)

Figure 5.1: SBO model.

The DES model employed in this study corresponds to the one used in Chapter 4, which is developed in SimEvents® (The MathWorks, Inc., 2014). It is responsible for capturing the delays ($\lambda_l$) caused by failures in production lines and provides to the GA information about the number of lots of a specific product $j$ manufactured by production line $l$, which is necessary to compute fitness values.

Since we need to search for a production plan that performs robustly under different realizations of orders due dates, in the DES model the production sequence of a production plan is randomly initialized for every production line, except for products that are used as raw materials during the manufacturing of other products. In order to assure the static logic of the model, products used as raw materials are manufactured before any other product to be manufactured in the same production line.
We developed in Matlab® R2014a’s (The MathWorks, Inc., 2014) a real-coded GA, that employs uniform crossover (crossover probability: 1), Gaussian mutation (mutation probability: 0.3, mutation fraction: 0.1, scale: 0.4 and shrink: 0.1), tournament selection (tournament size: 2), a population size of 40 individuals, 50 generations and the final solution selection employs a computational budget \( E \) of 3000 fitness evaluations (see Section 5.5.1 for more details about \( E \)). Here, we used the irace package (López-Ibáñez et al., 2011) to tune the parameters mentioned above, except for population size, which was determined after extensive experimentation. Table 5.3 in Appendix 5.A presents the possible parameter values given to irace to search for a tuned configuration for our GA. Those possible parameter values were chosen after preliminary experiments and the final configuration of parameters was identified by irace based on 300 experiments.

We implemented in this GA a truncation procedure that rounds each decision variable after crossover and mutation, in order to ensure compliance with integer constraints. We also implemented the constraint-handling method proposed by Deb (2000). Elitism is incorporated into this GA by combining the entire parent and offspring populations, and then extracting from this combined population the best individuals, which will constitute the new population for the next generation. During this extraction procedure and during tournament selection, feasible solutions are preferred over unfeasible ones and are ranked according to their fitness values, whereas unfeasible solutions are ranked first according to the number of constraint violations and then according to the magnitude of each violation (by how much a constraint was violated). For this reason fitness is not computed for unfeasible solutions. All computations are executed in parallel on a 16 core Intel(R) Xeon(R) CPU L5640 @ 2.27GHz with 24 GB of RAM running Scientific Linux, release 6.2.
5.5.1 Optimization Model

In this optimization problem, decision variables $x_{l,j}$ indicate the number of lots of product $j$ to be manufactured in production line $l$; therefore, a vector $x$ of decision variables constitutes a production plan. Those production plans are only feasible if the sets of constraints represented in the form of Equations 5.11 and 5.12 are satisfied. Production plans are here specified by the GA and then simulated by the DES model, which returns the actual number of lots of product $j$ manufactured in production line $l$ during the $r$-th simulation replication, denoted as $s_{l,j,r}$. Therefore, this optimization model is based on a black-box optimization approach that intends to maximize Equation 5.14 subject to the set of constraints in the form of Equations 5.11 and 5.12.

$$
\text{maximize } f(x) = \frac{1}{\gamma} \sum_{r=1}^{\gamma} \sum_{j=1}^{n} \sum_{l \in A_j} v_j \times s_{l,j,r} - k_j \times (x_{l,j,r} - s_{l,j,r}).
$$

(5.14)

Here, we apply a combination of two explicit averaging strategies to compute the fitness $f$ of an individual $x$, based on those simulated responses ($s_{l,j,r}$), as shown in Equation 5.14. More specifically, the evolutionary process is based on average fitness computed across 10 independent fitness evaluations ($\gamma = 10$), whereas the selection of the final solution is based on average fitness computed for every feasible individual of the final population across a number $\gamma$ of independent fitness evaluations, where $\gamma$ is computed as follows:

$$
\gamma = \left\lceil \frac{E}{\delta} \right\rceil.
$$

(5.15)

As shown in Equation 5.15, during the final solution selection, $\gamma$ depends on the number of feasible individuals ($\delta$) present in the final population and on the computational budget $E$ available for this last step, which is equal to 3000 fitness evaluations.
(\(E = 3000\)). We used the irace package to determine the computational budget allocated to the final solution selection.

Unlike in Chapter 4, where the optimization started with poor quality solutions and a very limited computational budget was available, here high quality solutions are created at the beginning of the optimization by specialized initialization operators (presented in Section 5.5.2). Therefore, we apply a modified version of the hybrid strategy proposed in Chapter 4 in order to select individuals based on more reliable fitness estimates during the optimization. After extensive experimentation, we concluded that using 10 fitness evaluations to compute average fitness during the evolutionary process returned similar solutions in terms of quality compared to using a sample size of 30, but at a much lower computational cost.

### 5.5.2 Initialization Operators

Exact and heuristic optimization techniques have been previously combined to take advantage of the synergies between those approaches. For instance, Gomes and Oliveira (2006) solved irregular strip packing problems by applying LP to generate neighbourhoods, while simulated annealing guided the search over the solution space. Another example is Reisi-Nafchi and Moslehi (2015), where the authors employed the rounded solution returned by LP as one of the individuals of the GA’s initial population when solving the two-agent order acceptance and scheduling problem. Here, we present two different initialization operators to create part of the GA’s initial population. Solutions created by the specialized initialization operators through DES or/and mathematical programming techniques (as described in Sections 5.5.2.1 and 5.5.2.2) are referred to as ILP-derived solutions. Both initializations intend to boost the optimization performance of the GA by including into the GA’s initial population feasible ILP-derived
solutions with high quality alleles, which are likely to be part of a good quality solution. After extensive experimentation we could determine that given a population size of 40, the incorporation of up to 4 ILP-derived solutions into the GA’s initial population promoted the creation of new high quality solutions during the optimization process. Our preliminary experiments revealed that the incorporation of more ILP-derived solutions into the GA’s initial population had a detrimental effect on the diversity of the initial population, which mainly led to premature convergence. Therefore, both initialization procedures incorporate into the GA’s initial population up to 4 ILP-derived solutions and the rest of individuals are randomly initialized.

5.5.2.1 Initialization 1

Initialization 1 incorporates as part of the GA’s initial population the ILP solution ($x^*$) for the $d$-MBKAR problem (optimize Equation 5.10 subject to constraints in the form of Equations 5.11 and 5.12). This deterministic ILP solution is a feasible and optimal solution for a fully reliable system. However, $s$ deviates from $x^*$ when uncertainty is present in the system ($s \neq x^*$). To capture those deviations we employ the DES model to run one independent simulation of $x^*$ and obtain its simulated response $s$, which is also included as part of the GA’s initial population.

Furthermore, the solution $x'^*$ obtained via CCP and one simulated response of $x'^*$ are also incorporated into the GA’s initial population. $x'^*$ is obtained by optimizing Equation 5.10 subject to constraints in the form of Equation 5.12 and the following:

$$
\sum_{j=1}^{n} \sum_{l \in A_j} w_{l,j} \times x_{l,j} \leq c_i \quad (i = 1, 2, \ldots, d - m),
$$

$$
P\left(\sum_{j=1}^{n} w_{l,j} \times x_{l,j} \leq C'_i\right) \geq \alpha_l \quad (l = 1, 2, \ldots, m),
$$

where $d - m$ represents the number of constraints not related to capacities of production.
lines. All resource constraints are in the form of Equation 5.16, except for those related to capacity of production lines, which are represented in the form of Equation 5.17. A constraint formulated in the form of Equation 5.17 is a probabilistic constraint, as \( C_i' \) is a random variable with known PDF, which models the number of hours that production line \( i \) is operative during a working month (24 days). This constraint restricts the probability of infeasibility to be no greater than a specified threshold \( 1 - \alpha_i \), where \( \alpha_i \) can take values between 0 and 1. In order to use an appropriate value for \( \alpha_i \), we computed CCP solutions with \( \alpha_i \in [0.95, 0.99] \) and then we calculated average profit values for each of those CCP solutions, across a sample of 5000 profit values (see Section 5.6 for more details about the sample size chosen) obtained by simulating each solution in the DES model. The CCP solution obtained with \( \alpha_i = 0.95 \) returned the highest average profit under the three problem instances analysed (see Section 5.6 for more details about problem instances analysed); therefore, in this study \( x^* \) was calculated with \( \alpha_i = 0.95 \).

If \( F_{C_i'} \) is CDF of \( C_i' \), then Equation 5.17 is equivalent to:

\[
F_{C_i'} \left( \sum_{j=1}^{n} w_{l,j} \times x_{i,j} \right) \leq 1 - \alpha_i \iff \sum_{j=1}^{n} w_{l,j} \times x_{i,j} \leq F_{C_i'}^{-1} \left( 1 - \alpha_i \right) \quad (l = 1, 2, \ldots, m).
\]  

(5.18)

This means that we first need to obtain the PDF of each \( C_i' \) before applying an available method to solve this problem (the reader is referred to Shapiro et al. (2014); Wallace and Ziemba (2005); Kall and Wallace (1995) for details about CCP). We implemented timers within the DES model in order to measure the exact number of hours that each production line was operative during a working month, denoted here as \( c_i' \), and used the DES model to generate 10000 numerical values of each \( C_i' \) by simulating a production plan that fully utilizes the design capacity of all production lines. Based on those simulated responses \( (c_i') \) we fitted a PDF to each sample. We used a sample size of
10000 numerical values of $C'_l$ because, under the highest uncertainty level analysed (see Section 5.6 for details about problem instance 3), the same solution $x'^*\text{r}$ was obtained when PDFs of $C'_l$ were estimated based on a sample size of 10000 and 50000 $c'_l$ values. Any production plan that fully utilizes the design capacity of all production lines can be simulated to obtain numerical values of $C'_l$, since $p_l$ is the probability that a failure occurs in production line $l$ during the manufacturing of a lot of any product. Here we used a production plan where all its decision variables were equal to 72 lots, which is the design capacity of the different production lines (see Chapter 2 for more details). It is important to note that using the PDFs of $C'_l$ to model the uncertainty present in the system is already a simplified version of the DES model because by doing that, the occurrence of failures which are production-level dependent and the repair times are not explicitly considered.

### 5.5.2.2 Initialization 2

In this initialization we use again the DES model to obtain numerical values of $C'_l$ by simulating a production plan that fully utilizes the design capacity of all production lines (see Initialization 1 for more details about that production plan), and then we use those values ($c'_l$) as the RHS of Equation 5.19:

$$\sum_{j=1}^{n} w_{l,j} \times x_{l,j} \leq c'_l \quad (l = 1, 2, \ldots, m).$$

(5.19)

Here, 4 numerical values of each $C'_l$ are generated and for each set of $c'_l$ values an ILP solution is found by optimizing Equation 5.10 subject to the sets of constraints in the form of Equations 5.12, 5.16 and 5.19. In order to maintain diversity, duplicates are eliminated among those 4 ILP solutions and then the remaining solutions are incorporated into the GA’s initial population.
5.5.3 Repair Operator

Due to the nature of our GA, unfeasible solutions are generated during the optimization procedure. For instance, the crossover and mutation operators may turn high quality solutions into unfeasible ones, and thus may lead to loss of valuable genetic information (since here feasible solutions are preferred over unfeasible solutions). In order to cope with this issue, we introduce a repair operator which tries to fix the chromosome of unfeasible solutions via simulation. This operator is applied before fitness evaluation and it replaces every unfeasible solution \( x \) present in the population (initial or offspring population) by one of its simulated responses \( s \), obtained from the DES model. This is a simple and effective procedure for repairing solutions where capacity constraints are violated, but it fails to repair solutions which violate other constraints, e.g. demand constraints.

5.6 Benchmark Analysis

Solutions obtained with the SBO model with initialization 1 (SBO1) and with initialization 2 (SBO2) are benchmarked against solutions generated via ILP and CCP, which are two mathematical programming techniques commonly applied in production planning. Stochastic programming with recourse is another alternative that could be used here; however, the problem of this approach is that a mathematical expression able to capture the uncertainty of the system is needed, which is difficult to obtain and will involve the consideration of multiple scenarios. Moreover, here we need a concrete production plan and stochastic programming with recourse returns a solution for every possible scenario. With this approach the model itself is the solution rather than a concrete production plan. Stochastic programming with recourse may be a better
alternative to make decisions at the operational level, namely to re-optimize a production plan after the occurrence of failures; however, this is outside of the scope of this study. Additionally, we benchmark SBO1 and SBO2 against the SBO model without any of the two initialization operators (SBO3). Here, our intention is not only to compare those different strategies, but more importantly, we are interested in understanding how they perform relative to each other and how performance differences evolve with increasing uncertainty levels, so we can get a clearer idea about what method is more appropriate under different circumstances.

Three problem instances are addressed here to evaluate the models mentioned under different levels of uncertainty. Table 5.1 presents the probabilities $p_l$ that a failure occurs during the manufacturing of a product lot in the different production lines per problem instance, as well as the parameter $\mu_l$ of the exponential PDFs used to model the random variables $\Lambda_l$, whose numerical values represent the delay caused per repair service of a production line. Please note that $p_l$ values in instance 1 as well as $\mu_l$ values are based on historical data collected over a period of 54 months. As mentioned in Chapter 2, $p_l$ values in instance 1 are conservative (too optimistic) lower bounds for such probabilities, since they were calculated based on the number of production line failures recorded and assuming that every production line remained operative and was fully utilized during a period of 54 months (number of months for which information is available), which is an unrealistic assumption; therefore, $p_l$ values in problem instance 2 and 3 are the double and triple, respectively, of the corresponding $p_l$ values in problem instance 1. Only production line 2, 3 and 4 have been assigned with a positive $p_l$ because failures in the other lines rarely or never occurred, according to the historical information provided by the company (see Table 2.5 in Chapter 2).

SBO1, SBO2 and SBO3 are executed 30 times using different seeds for the random number generator in every run. Average profit values, measured in United States Dollar (USD), are computed across a sample of 5000 profit values obtained via stochastic
### Table 5.1: $p_l$ per problem instance and PDFs specifications to model $\Lambda_l$.

<table>
<thead>
<tr>
<th>Production line ($l$)</th>
<th>$p_{l1}$</th>
<th>$p_{l2}$</th>
<th>$p_{l3}$</th>
<th>$p_{l4}$</th>
<th>$\Lambda_l$</th>
<th>PDF</th>
<th>$\mu_l$ (d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>2</td>
<td>0.0355</td>
<td>0.0710</td>
<td>0.1065</td>
<td>Exponential</td>
<td>2.03</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>3</td>
<td>0.0468</td>
<td>0.0936</td>
<td>0.1404</td>
<td>Exponential</td>
<td>2.24</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>4</td>
<td>0.0471</td>
<td>0.0942</td>
<td>0.1413</td>
<td>Exponential</td>
<td>3.21</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>5</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>6</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>7</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

Simulation for every final solution of each run performed with SBO1, SBO2 and SBO3. A sample size of 5000 profit values was chosen because it returned estimates that were reliable enough for the purpose of our analysis, given that the relative change of average profit computed across samples of 10000 and 5000 profit values in the problem instance with highest uncertainty level (problem instance 3) was lower than $1e^{-2}$. Additionally, 30 average profit values of the solutions obtained with ILP ($x^*$) and with CCP ($x'^*$) are also computed across a sample size of 5000 profit values obtained with those solutions via stochastic simulation. Those average profit values are used to evaluate the optimization performance of the different models.

#### 5.6.1 Results

Under all uncertainty levels analysed, SBO1 and SBO2 were able to generate production plans that outperformed in terms of average profitability the solutions given by ILP, CCP and SBO3. This is according to results from Mann-Whitney U tests (Mann and Whitney, 1947) presented in Tables 5.4a, 5.4b and 5.4c (Appendix 5.B), which statistically show that the average profit values obtained with $x^*$, $x'^*$ and with production plans given by SBO3 are stochastically smaller ($p < .01$) than the ones obtained with solutions given by SBO1 and SBO2, in all problem instances. The advantage of SBO1
and SBO2 over ILP, CCP and SBO3 under the uncertainty levels analysed is also illustrated in Figures 5.2, 5.3 and 5.4, where the CDFs of average profits generated with $x^\ast$, $x'^\ast$ and with production plans given by SBO3 are dominated (first-order stochastic dominance (Hadar and Russell, 1969)) by the CDFs of average profit values obtained with solutions given by SBO1 and SBO2.

Under low uncertainty levels (problem instance 1), the optimization performance of SBO1 and SBO2 did not evidence a significant ($p > .05$) difference. However, under medium and high uncertainty levels (problem instance 2 and 3) SBO1 was clearly outperformed by SBO2. This is confirmed by results from Mann-Whitney U tests presented in Tables 5.4a, 5.4b and 5.4c (Appendix 5.B), and are also illustrated in Figures 5.2, 5.3 and 5.4. These results demonstrate that SBO2 is a more effective strategy than SBO1 under increasing uncertainty levels. This is because initialization 2 draws upon simulation responses for the RHSs of capacity constraints ($c'_j$) to return solutions created according to the level of uncertainty captured by those RHSs, whereas in initialization 1 uncertainty is simply captured by incorporating $x'^\ast$ into the GA’s initial population as well as the simulated responses ($s$) of $x^\ast$ and $x'^\ast$.

Moreover, both specialized initialization operators significantly ($p < .01$) improved the optimization performance of the SBO model under all uncertainty levels analysed, as shown in Tables 5.4a, 5.4b and 5.4c (Appendix 5.B). This is also illustrated in Figures 5.2, 5.3 and 5.4, where the CDFs of average profits generated with production plans given by SBO3 are dominated (first-order stochastic dominance) by the CDFs of average profit values obtained with solutions given by SBO1 and SBO2. We executed 30 additional runs of SBO3 in every problem instance, but this time we allocated 100 rather than 50 generations to the optimization procedure of SBO3 (SBO3x2) in order to investigate whether the performance gap between SBO models with and without specialized initialization operators decreases or not when a bigger computational budget is available. Our results suggest that SBO1 and SBO2 outperform SBO3 even
when the latter optimizes across twice the number of generations employed by the former two models. These results are consistent across all problem instances analysed, as illustrated by Figures 5.5, 5.6 and 5.7 in Appendix 5.C.

In order to investigate if this boost in performance was due to the initialization operators and/or due to further improvements achieved by the GA, we checked if the final solutions returned by SBO1 and SBO2 were different to the corresponding ILP-derived solutions and also compared the average profit (computed across 5000 independent simulation replications) of every ILP-derived solution to the average profit of the final solution given by the GA, in every run of SBO1 and SBO2. Our GA was able to find better solutions (in terms of average profitability) than any of the ILP-derived solutions created with initialization 1, in every run of SBO1 and under all uncertainty levels analysed. The same results were obtained for SBO2 in problem instance 3, but under low (problem instance 1) and medium (problem instance 2) uncertainty levels, our GA returned the best ILP-derived solution created by initialization 2 as the final solution, in 22 and 7 occasions (out of 30), respectively. Sample means and sample standard deviations of the average profit values returned by the best ILP-derived solutions and by the final solutions obtained with SBO1 and SBO2 across 30 runs are presented in Table 5.5 (Appendix 5.D). These results demonstrate that both, the initialization operators and the GA were responsible for this boost in performance, except for SBO2 in problem instance 1, where most of the credit can be attributed to initialization 2.

Our experiments suggest that the application of a search procedure such as the one implemented in initialization 2 might be sufficient to address simpler instances of the problem analysed, i.e. instances with low uncertainty levels, but the application of SBO2 becomes more appropriate with increasing uncertainty levels. This is because with higher uncertainty, the (linear) objective function (Equation 5.10) used by initialization 2 to generate (via ILP) ILP-derived solutions becomes an increasingly worse
approximation of the real function that needs to be optimized, namely Equation 5.13, and thus the search mechanism of initialization 2 is only able to search in a subset of all possible production plans, whereas in SBO2 the GA has access to the entire space of all possible production plans. In other words, for the problem analysed the application of our GA becomes more relevant with increasing uncertainty levels.

![CDFs of average profit values generated with solutions given by SBO1, SBO2, SBO3, CCP and ILP in problem instance 1 (low uncertainty).](image)

Finally, the average computational times obtained with SBO1 and SBO2, presented in Table 5.2, show that our approach can be realistically applied in practice.

<table>
<thead>
<tr>
<th></th>
<th>SBO1</th>
<th>SBO2</th>
<th>SBO3</th>
<th>CCP</th>
<th>ILP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance 1</td>
<td>841</td>
<td>836</td>
<td>773</td>
<td>641a</td>
<td>&lt; 1</td>
</tr>
<tr>
<td>Instance 2</td>
<td>989</td>
<td>984</td>
<td>951</td>
<td>649a</td>
<td>&lt; 1</td>
</tr>
<tr>
<td>Instance 3</td>
<td>982</td>
<td>979</td>
<td>928</td>
<td>655a</td>
<td>&lt; 1</td>
</tr>
</tbody>
</table>

\(a\) including simulation time needed to estimate the PDFs of \(C_i\)

Table 5.2: Computational time in seconds for each model per problem instance.
Figure 5.3: CDFs of average profit values generated with solutions given by SBO1, SBO2, SBO3, CCP and ILP in problem instance 2 (medium uncertainty).

Figure 5.4: CDFs of average profit values generated with solutions given by SBO1, SBO2, SBO3, CCP and ILP in problem instance 3 (high uncertainty).
5.7 Conclusion

Our experiments illustrate how the performance of deterministic and stochastic global optimal solutions, obtained under overly stringent assumptions, deviates from the optimum in the presence of uncertainty. This finding is in accordance with results reported by Wang et al. (2015), where the optimal solution for a deterministic travelling salesman problem (TSP) evidenced a poor performance in a TSP under uncertainty.

Our results also show, as may be expected, that an accurate incorporation of the existing uncertainty into the problem formulation becomes more important with increasing uncertainty levels. In this sense, it is worth emphasizing that our SBO approach explicitly models (via simulation) the key features that bring uncertainty to the problem, namely the occurrence of failures that are production-level dependent and the delays caused by the corresponding repair services. CCP, on the other hand, loses information on those key features that we are interested in capturing, since it tries to model that uncertainty by using PDFs (which need to be estimated via simulation) of the RHSs of the capacity constraints ($C^i_l$). Furthermore, here CCP has no mechanism to consider penalties imposed to deviations from a plan during the specification of a production plan, which is something that can be explicitly considered in our SBO approach. In this sense, our approach is able to adjust a production plan according to the penalty level, whereas the production plan obtained with CCP is the same under different penalty levels.

We also demonstrate how the combined use of simulation (DES) and mathematical programming techniques (ILP and CCP) to specify part of the GA’s initial population significantly ($p < .01$) enhanced the optimization performance of the SBO model. These results support the idea that the incorporation of “high quality” alleles (from ILP-derived solutions created with initializations 1 and 2) into the GA’s initial population takes the GA search straight to a feasible region and guides it towards more
promising areas of the solution space, enhancing the performance of the GA and reducing the computational effort needed to find adequate solutions.

Finally, the fact that near-optimal solutions, determined with our approach, outperform mathematical programming solutions not only illustrates the importance that accurate incorporation of uncertainty has during the problem formulation, but also shows how the combination of simulation, mathematical programming and meta-heuristic methods can generate solutions that outperform the ones obtained via individual application of the approaches mentioned. Here we demonstrate that for the real-world problem analysed, ILP and CCP are limited in their ability to return good quality solutions compared to our approach; however, the advantage of our approach over mathematical programming techniques (ILP and CCP) will disappear when the impact of uncertainty becomes negligible.

5.8 Limitations and Future Research

This study can be extended by considering more realistic features of manufacturing systems such as deterioration of production lines due to previous failures, different types of failure as well as different repair types. This could be integrated into the simulation model by allowing for dynamic, rather than static probabilities of failures. The inclusion of Markov chains into the DES model is a possible approach to capture such extensions.

Furthermore, a multi-objective formulation of the problem that explicitly takes into account the robustness of solutions could be explored. For instance, an additional objective such as the maximization of the signal-to-noise ratio (Al-Aomar, 2006) or the minimization of the standard deviation of profit may be used to explicitly account for the variability in fitness values.
Appendix 5.A. Automatic Algorithm Configuration with irace.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crossover probability</td>
<td>0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1</td>
</tr>
<tr>
<td>Mutation probability</td>
<td>0.1, 0.2, 0.3, 0.4</td>
</tr>
<tr>
<td>Mutation fraction</td>
<td>0.025, 0.05, 0.1, 0.15, 0.2, 0.3</td>
</tr>
<tr>
<td>Scale</td>
<td>0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1</td>
</tr>
<tr>
<td>Shrink</td>
<td>0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1</td>
</tr>
<tr>
<td>Tournament size</td>
<td>2, 3</td>
</tr>
<tr>
<td>Number of generations</td>
<td>50, 60, 70, 80, 90, 100</td>
</tr>
<tr>
<td>$E$</td>
<td>1000, 2000, 3000, 4000, 5000</td>
</tr>
</tbody>
</table>

Table 5.3: Possible parameter values given to irace.
## Appendix 5.B. Mann-Whitney U tests

<table>
<thead>
<tr>
<th></th>
<th>SBO1</th>
<th>SBO2</th>
<th>SBO3</th>
<th>CCP</th>
<th>ILP</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBO1</td>
<td>—</td>
<td>414*</td>
<td>0**</td>
<td>0**</td>
<td>0**</td>
</tr>
<tr>
<td>SBO2</td>
<td>—</td>
<td>—</td>
<td>0**</td>
<td>0**</td>
<td>0**</td>
</tr>
<tr>
<td>SBO3</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>0**</td>
<td>355*</td>
</tr>
<tr>
<td>CCP</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>0**</td>
</tr>
<tr>
<td>ILP</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

** p < .01, * heteroscedasticity according to non-parametric Levene test (p > .05) (Nordstokke and Zumbo, 2010)

(a) Instance 1

<table>
<thead>
<tr>
<th></th>
<th>SBO1</th>
<th>SBO2</th>
<th>SBO3</th>
<th>CCP</th>
<th>ILP</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBO1</td>
<td>—</td>
<td>127**</td>
<td>0**</td>
<td>0**</td>
<td>0**</td>
</tr>
<tr>
<td>SBO2</td>
<td>—</td>
<td>—</td>
<td>0**</td>
<td>0**</td>
<td>0**</td>
</tr>
<tr>
<td>SBO3</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>0**</td>
<td>0**</td>
</tr>
<tr>
<td>CCP</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>0**</td>
</tr>
<tr>
<td>ILP</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

** p < .01

(b) Instance 2

<table>
<thead>
<tr>
<th></th>
<th>SBO1</th>
<th>SBO2</th>
<th>SBO3</th>
<th>CCP</th>
<th>ILP</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBO1</td>
<td>—</td>
<td>161**</td>
<td>0**</td>
<td>0**</td>
<td>0**</td>
</tr>
<tr>
<td>SBO2</td>
<td>—</td>
<td>—</td>
<td>0**</td>
<td>0**</td>
<td>0**</td>
</tr>
<tr>
<td>SBO3</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>0**</td>
<td>0**</td>
</tr>
<tr>
<td>CCP</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>0**</td>
</tr>
<tr>
<td>ILP</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

** p < .01; no heteroscedasticity according to non-parametric Levene test (p > .05)

(c) Instance 3

Table 5.4: Values for Mann-Whitney U statistic obtained for average profit values.
Appendix 5.C. SBO3x2 vs. SBO3, SBO1 and SBO2.

Figure 5.5: CDFs of average profit values generated with solutions given by SBO1, SBO2, SBO3 and SBO3x2 in problem instance 1 (low uncertainty).

Figure 5.6: CDFs of average profit values generated with solutions given by SBO1, SBO2, SBO3 and SBO3x2 in problem instance 2 (medium uncertainty).
Figure 5.7: CDFs of average profit values generated with solutions given by SBO1, SBO2, SBO3 and SBO3x2 in problem instance 3 (high uncertainty).
Appendix 5.D. Best ILP-derived solutions vs. final solutions returned by SBO1 and SBO2.

Table 5.5: Sample mean and sample standard deviation of the average profit values returned by the best ILP-derived solutions and by the final solutions obtained with SBO1 and SBO2 across 30 runs.
Chapter 6

A Matheuristic Optimizer to Address the $d$-MBKARS Problem

6.1 Abstract

In this chapter, we explore the potential of combining the global search mechanism of a GA with the ability of ILP to generate a feasible (and optimal) production plan for any given set of constraints. In particular, we use a GA to explore the space of capacity values of failure-prone production lines, rather than searching through the space of all possible production plans as done in Chapter 5. Here, the application of ILP can be seen as a translation step that maps a solution given by the GA (genotype) onto a production plan (phenotype). ILP is thus repeatedly called during the optimization procedure and not only at the beginning, as occurs with both specialized initialization operators from Chapter 5. We analyse the performance of this “matheuristic” optimizer in the context of the real-world production planning problem addressed in the previous chapter. Our results suggest that the approach performs well for problems with low and medium uncertainty levels as it excludes unfeasible solutions from the search space. However, we also find that the effectiveness of this approach decreases
with increasing level of uncertainty. This can be explained by the restriction of the accessible (phenotypic) search space to those production plans that are determined via ILP: with increasing uncertainty, the real (non-linear) objective function becomes less aligned with the (linear) objective function optimized in the ILP problem, limiting the quality of solutions that can be reached through this search process.

### 6.2 Introduction

Most of the available techniques to address combinatorial optimization problems (Pintea, 2014, p. 21) can be classified into two main categories: exact and heuristic approaches (Sørensen, 2015; Raidl and Puchinger, 2008). Exact approaches are able to find a guaranteed optimal solution, but their computational time increases dramatically with the scale of the problem, and thus its applicability to address real-world problems is usually limited. Therefore, in many real-world applications optimality is traded for acceptable computational times by employing heuristic approaches.

General purpose heuristics are also known as meta-heuristics (Burke et al., 2009). The term Meta-heuristics, first introduced by Glover (1986), is the composition of two Greek words: the prefix “meta” and “heuriskein”, which mean “beyond” and “to find”, respectively. Meta-heuristics are high level search strategies (as revealed by its meaning), able to return solutions, not necessarily optimal (Caserta and Voß, 2009), in acceptable computational times by implementing a search that tries to balance diversification (exploration) and intensification (exploitation) (Blum and Roli, 2008).

The combination of exact and meta-heuristic approaches seems ideal by drawing upon the strengths of both streams. However, it is only in recent years that the potential of hybrid meta-heuristics is being recognized in the optimization community (Raidl and Puchinger, 2008). In the broad sense, the term hybrid meta-heuristic refers to the
combination of a meta-heuristic with any other optimization technique, e.g. mathematical programming methods, heuristic methods or other meta-heuristics (Blum and Roli, 2008). More specifically, approaches that combine meta-heuristics with mathematical programming techniques are known as matheuristics (Villegas et al., 2013). This relatively new area of research has captured the attention of a growing research community (Boschetti et al., 2009), which is reflected in an increasing number of publications (Caserta and Voß, 2013; Blum et al., 2011; Blum and Roli, 2008; Raidl and Puchinger, 2008; Puchinger and Raidl, 2005; Raidl, 2006; Talbi, 2002).

Therefore, in line with this research trend we investigate in this chapter a new SBO approach that employs a matheuristic optimizer to address the \(d\)-MBKARS problem. The success of the matheuristic approaches (SBO1 and SBO2) introduced in Chapter 5, provided supporting evidence of the synergies resulting from the combination of mathematical programming techniques with a population-based meta-heuristic optimizer, namely a GA. In those approaches, ILP was only called at the beginning of the optimization to initialize part of the GA initial population. In this chapter, we investigate whether synergies between both techniques can be further exploited, by developing a matheuristic optimizer where mathematical programming techniques are repeatedly called during the optimization (not only at the beginning).

The motivation behind the approach presented here, latter referred to as SBOMat, is to create a search space for the meta-heuristic, in this case a GA, where all possible solutions are feasible and then apply ILP as a translation tool able to map every solution given by the GA onto a feasible production plan. The use of mathematical programming techniques may facilitate the generation of feasible solutions under constrained settings, a difficult task for meta-heuristics such as GAs, where standard genetic operators like crossover and mutation could potentially turn high quality solutions into unfeasible ones.
6.3 SBOMat Approach

As presented in Chapter 5, the original formulation of the $d$-MBKARS problem is as follows:

$$\text{maximize } f(x) = \sum_{j=1}^{n} \sum_{l \in A_j} v_j \times s_{l,j} - k_j \times (x_{l,j} - s_{l,j})$$ (6.1)

subject to

$$\sum_{j=1}^{n} \sum_{l \in A_j} w_{l,j} \times x_{l,j} \leq c_i \quad (i = 1, 2, \ldots, d),$$ (6.2)

$$\sum_{l \in A_j} x_{l,j} \leq b_j \quad (x_{l,j} \in \mathbb{Z}_{\geq 0}; j = 1, 2, \ldots, n).$$ (6.3)

In order to introduce the SBOMat approach, it is necessary to add to the original formulation a new set of constraints in the form of Equation 6.4 to separately capture the capacity requirements of failure-prone production lines, leaving the rest of resource constraints represented in the form of Equation 6.2.

$$\sum_{j=1}^{n} w_{l,j} \times x_{l,j} \leq c'_l \quad (l = 1, 2, \ldots, m).$$ (6.4)

The main difference between SBOMat and the SBO approaches presented in Chapter 5 is that the former uses a GA to search for capacity values of failure-prone production lines, namely the RHS $c'_l$ of Equation 6.4, whereas the latter uses a GA to directly search for a production plan $x$. As illustrated in Figure 6.1, the SBOMat approach first generates a population of solutions, where each solution is denoted here as $c'$ (vector of $c'_l$), and then employs ILP as a translation tool able to generate a production plan $x$ (phenotype), given the set of constraints $c'$ (genotype) determined by the GA. This second step is performed by optimizing, via ILP, Equation 6.5 subject to the sets of
The fitness of each solution $c'$ given by the GA, is computed by evaluating its corresponding ILP solution $(x)$ in the same way as described in Section 5.5.1.

Unlike the SBO approaches presented in Chapter 5, the SBOMat approach always generates (via ILP) production plans that satisfy all constraints, since every decision variable $c'_l$ given by the GA is constrained by a lower bound equal to zero and by an upper bound equal to the design capacity of the corresponding production line. Since the solution space of $c'_l$ is only constrained by lower and upper bounds, SBOMat has a wider exploration capability than the search mechanism implemented in the specialized initialization operator presented in Section 5.5.2.2, where configurations of $c'$ are limited to values returned by the DES model. Another potential advantage of SBOMat is that the optimization procedure is not biased by initial solutions created by specialized initialization operators, as occurs in SBO1 and SBO2 (see Section 5.5.2 for more details).
6.4 Benchmark Analysis

In order to analyse the performance of SBOMat, we apply it to the real-world problem analysed in Chapter 5 and compare its results against the ones returned by the SBO1 and SBO2 models, proposed in Chapter 5.

To enable a fair comparison, all different strategies employ the same GA (see Chapter 5 for details about the GA), the optimization is performed across 50 generations based on a population size of 40 individuals and the computational budget \((E)\) allocated to the final solution selection is 3000 fitness evaluations. However, neither the specialized initialization operators (see Section 5.5.2) nor the repair operator (see Section 5.5.3) are applied to SBOMat. The parameter configuration of this GA was determined separately for SBOMat based on preliminary experimentation and is as follows: uniform crossover (crossover probability: 0.8), Gaussian mutation (mutation probability: 0.2, mutation fraction: 0.33, scale: 0.5 and shrink: 0.1) and tournament selection (tournament size: 2).

In line with our previous experiments with SBO1 and SBO2, we execute 30 runs of SBOMat for the three problem instances analysed in Chapter 5. Here, we also use different seeds for the random number generator in every run. We compute average profit values for every final solution returned by SBOMat, across a sample of 5000 profit values obtained via stochastic simulation, and then use those estimates to evaluate the optimization performance of SBOMat against SBO1 and SBO2.

6.5 Results

Our results reveal that under low and medium uncertainty levels (problem instance 1 and 2), the SBOMat approach outperformed \((p < .01)\) SBO1 and SBO2 in terms of optimization performance, but evidenced a performance comparable \((p > .05)\) to SBO2.
under high uncertainty level (problem instance 3). This is confirmed by results from Mann-Whitney U tests presented in Tables 6.1a, 6.1b and 6.1c, and are also illustrated in Figures 6.2, 6.3 and 6.4.

Production plans in SBOMat are obtained via ILP, by optimizing Equation 6.5, subject to the sets of constraints in the form of Equations 6.2, 6.3 and 6.4. However, Equation 6.5 becomes an increasingly worse approximation for the real objective function (Equation 6.1) under increasing uncertainty levels. This may be the reason why the performance difference between SBOMat and SBO2 seems to decrease with increasing levels of uncertainty.

We adapted for SBOMat the specialized initialization operator presented in Section 5.5.2.2 (SBOMat2), in order to be sure that results obtained, especially in problem instance 3, are not a consequence from the lack of this initialization. Our results suggest that the implementation of this specialized initialization operator has no significant ($p > .05$) influence on the optimization performance of SBOMat in any of the three problem instances analysed, as shown in Tables 6.1a, 6.1b and 6.1c.

### 6.6 Conclusion

Our experiments suggest that SBOMat is a more effective ($p < .01$) strategy than SBO1 and SBO2 to address the $d$-MBKARS problem under low and medium uncertainty levels, but under high uncertainty level its performance is comparable ($p > .05$) to SBO2.

In SBOMat production plans are specified, based on the assumption that the linear objective function (Equation 6.5) optimized via ILP is a good approximation of the real (non-linear) objective function, namely Equation 6.1. As a consequence of this assumption the quality of solutions returned by SBOMat may be limited, since the (phenotypic) search space in this approach is restricted to production plans that
are only accessible via ILP. In other words, all production plans that can be found via ILP represent a subset of all possible production plans, as illustrated in Figure 6.1.
Figure 6.4: CDFs of average profit values generated with solutions given by SBO1, SBO2 and SBOMat in problem instance 3 (high uncertainty).

We expect that the optimization performance of SBOMat will decrease with increasing uncertainty levels, as the linear objective function becomes an increasingly worse approximation of the real objective function under higher levels of uncertainty. Therefore, SBO2 may be a better alternative than SBOMat to tackle more complex instances of this problem, but further investigation is still required to find conclusive evidence.
Table 6.1: Values for Mann-Whitney U statistic obtained for average profit values.

<table>
<thead>
<tr>
<th></th>
<th>SBO1</th>
<th>SBO2</th>
<th>SBOMat</th>
<th>SBOMat2</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBO1</td>
<td>—</td>
<td>414a</td>
<td>181**</td>
<td>174**</td>
</tr>
<tr>
<td>SBO2</td>
<td>—</td>
<td>—</td>
<td>267**</td>
<td>278***a</td>
</tr>
<tr>
<td>SBOMat</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>434</td>
</tr>
<tr>
<td>SBOMat2</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

** *p < .01; a heteroscedasticity according to non-parametric Levene test (p > .05) (Nordstokke and Zumbo, 2010)*

(a) Instance 1

<table>
<thead>
<tr>
<th></th>
<th>SBO1</th>
<th>SBO2</th>
<th>SBOMat</th>
<th>SBOMat2</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBO1</td>
<td>—</td>
<td>127**</td>
<td>81**</td>
<td>60**</td>
</tr>
<tr>
<td>SBO2</td>
<td>—</td>
<td>—</td>
<td>183**</td>
<td>212**</td>
</tr>
<tr>
<td>SBOMat</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>408</td>
</tr>
<tr>
<td>SBOMat2</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

** *p < .01*

(b) Instance 2

<table>
<thead>
<tr>
<th></th>
<th>SBO1</th>
<th>SBO2</th>
<th>SBOMat</th>
<th>SBOMat2</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBO1</td>
<td>—</td>
<td>161**</td>
<td>234**</td>
<td>228**</td>
</tr>
<tr>
<td>SBO2</td>
<td>—</td>
<td>—</td>
<td>423a</td>
<td>414a</td>
</tr>
<tr>
<td>SBOMat</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>447</td>
</tr>
<tr>
<td>SBOMat2</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

** *p < .01; a heteroscedasticity according to non-parametric Levene test (p > .05)*

(c) Instance 3
Chapter 7

Simulating Realistic Features of Manufacturing Systems

In this chapter, we present the details of the simulation model employed in the next chapter (Chapter 8), so that the reader can understand the modifications implemented. In order to capture more realistic features of manufacturing systems, this simulation model incorporates into the previous DES model (see Section 4.3.1) the extensions suggested in Section 5.8. Those extensions involve the consideration of different types of failure, different types of repair and the deterioration in production lines caused by the occurrence of previous failures.

As in the original simulation model (Chapter 5), here the production sequence of product lots is randomly initialized to model the uncertainty around realization of order due dates. However, in order to assure the static logic of the model, we assign a higher priority to the initialization of orders for sub-products. This setting (specified in the event-based entity generator block illustrated in Figure 7.1) makes that sub-products are manufactured before any other product to be manufactured in the same production line.

The number of product lots to be processed by every production line is given by the
optimizer. More specifically, the optimizer passes the value of each decision variable \( x_{l,j} \) to the corresponding function-call generator block (see Figure 7.1). In turn, the simulation model returns to the optimizer the number of lots of product \( j \) manufactured by production line \( l \), denoted as \( s_{l,j} \) (see Figure 7.2).

As mentioned in Chapter 4, function-call generator blocks do not accept as input values equal to zero. To overcome this difficulty we used as input the value of each decision variable \( x_{l,j} \) plus one additional product lot. This additional lot is then deviated to a single server (dummy server in Figure 7.1), which processes that lot during the entire simulation replication. This modification, illustrated in Figure 7.1, enables us to simulate production plans (\( \mathbf{x} \)) where production decisions (decision variable \( x_{l,j} \)) can take values equal to zero (\( x_{l,j} = 0 \)).

### 7.1 New Extensions

The simulation model presented in this section incorporates into the problem the following features of manufacturing systems. First, a production line may evidence different types of failure, which can be classified according to their level of seriousness. To capture this feature, we assume that two failure types can happen, a serious (\( T = 1 \)) and a non-serious (\( T = 2 \)) failure. Second, we also assume that after a non-serious failure an imperfect repair is undertaken, whereas after a serious failure a near-perfect repair is performed. We refer to the second repair type as near-perfect because a non-serious failure can still take place with certain probability after such repair has been undertaken. Third, we incorporate into the simulation model the deterioration in production lines caused by previous failures. More specifically, we consider that the occurrence of previous non-serious failures increases both, the probability that a failure occurs during the manufacturing of the next product lot and the probability that the next failure will be of a serious type.
Figure 7.1: Order initialization sub-system of production line 2.
CHAPTER 7

128

Figure 7.2: Final product sub-system of production line 2.

These features of the problem are modelled here by incorporating into the DES model a Markov chain for every production line present in the system. Figure 7.3 shows how Markov chains are implemented within the DES model. An illustration of such Markov chain and its transition probability matrix are presented in Figure 7.4 and in Table 7.1, respectively.

Figure 7.4 shows that this Markov chain is an ergodic chain (Alfa, 2016) because it consists entirely of one class of states, which is positive recurrent and aperiodic (Ching et al., 2006a, 15). Every node in Figure 7.4 represents a possible state of a specific production line. States are labelled as $O_s$ or $F_{T,s}$ to denote that the system was operative or that a failure occurred in production line $l$ during the manufacturing of the previous product lot, respectively. The sub-index $s$ indicates the number of non-serious failures occurred in a production line after a near-perfect repair has been undertaken, whereas the sub-index $T$ (in states denoted as $F_{T,s}$) indicates the occurrence of a serious ($T = 1$) or non-serious ($T = 2$) failure in production line $l$ during the manufacturing of the previous product lot.

Consequently, every state of the Markov chain indicates whether a production line
Figure 7.3: Manufacturing sub-system of production line 2.
Table 7.1: Transition probability matrix.
was operative or failed during the manufacturing of the previous product lot as well as the number of non-serious failures occurred in a production line after a near-perfect repair has been undertaken. For instance, if production line $l$ is in state $O_3$, it means that no failure occurred during the manufacturing of the previous product lot, but three non-serious failures occurred after the last near-perfect repair was undertaken on that line.

Moreover, this is a non-terminated system; therefore, the initial state of the production line is determined based on its steady state probabilities (Bertsekas and Tsitsiklis, 2008, p. 352) ($p_{F,T}^l$ and $p_{O}^l$). This feature of the system is captured in the simulation model by routing the first product lot through the first path (output port 1 of the output switch block in Figure 7.3) after passing the server block that models the production line (production line 2 in Figure 7.3). The initial state of the production line is then

---

$^1$There is a unique steady state vector, since our finite-state Markov chain is ergodic (Ching et al., 2006a, 122)
set as an attribute on that first product lot, based on the realization of a discrete random variable. Every outcome of this random variable represents a state in the Markov chain and it is realized according to the corresponding steady state probabilities. Once the first product lot has passed through, this first path is no longer accessible to any subsequent lot until the end of the simulation replication. This first path is blocked by a gate (see gate block in Figure 7.3) which forbids entity arrivals (product lots) when $Block\_Gate \leq 0$ (where the signal $Block\_Gate = 1 - \text{number of entities departed from the attribute block which sets the initial state of the production line}$).

Except for the initial state, every subsequent state of a production line is determined by the corresponding Markov chain, based exclusively on the transition probabilities of its previous state. Here, a Markov chain is triggered every time a product lot needs to be manufactured by the corresponding production line. The information about the state of the production line is then passed to the DES model, where it is set as an attribute on each product lot. Those attributes determine the route to be followed by a product lot. More specifically, the state ($F_{T,s}$ or $O_s$) of a production line is an attribute that specifies the output port through which a product lot is routed when it arrives to the output switch block shown in Figure 7.5.

Information about the previous state of the production line is given by the signal $Previous\_State$ (see Figure 7.3 and 7.5), which indicates the input port through which the previous product lot arrived at the path combiner block illustrated in Figure 7.5. This signal is then used as input for the Markov chain (see Figure 7.3) when the next product lot arrives. For simplicity, we assume that the transition probabilities of every state can be calculated as shown in Table 7.1. $p_l$ is the probability that a production line failure occurs (either serious or non-serious) during the manufacturing of the next product lot, given that the production line is in state $O_0$ (i.e. no non-serious failure has occurred after a near-perfect repair and the production line was operative during the manufacturing of the previous product lot). $\zeta_l$ determines the probability that a
serious failure occurs during the manufacturing of the next product lot, given that the production line is in state $O_0$ and given that a production line failure will occur during the manufacturing of the next product lot. The deterioration of a production line caused by a non-serious failure is modelled by raising $p_l$ and $\zeta_l$ to the power of $\varphi_l$.

If the current state of the production line is operative ($O_s$), no repair service is needed, the product lot is routed to its corresponding product sink (sink blocks shown in Figure 7.2) and the production line immediately becomes available to manufacture the next product lot in the queue, but if the state indicates a failure ($F_{T,s}$) a repair service has to be undertaken. In Figure 7.5, output ports 6 to 10 from the output switch block represent operative states and thus there is no repair server in those paths, whereas output port 1 takes that lot to a serious failure server and output ports 2 to 5 to non-serious failure servers, which model the delay caused by a repair service. The length of that delay, here denoted as $\lambda_{l,T}$, is the numerical realization of a random variable $\Lambda_{l,T}$, modelled by an exponential PDF with known mean $\mu_{l,T}$. The parameter $\mu_{l,T}$ is selected based on the current state of the production line, since it depends on the failure type $T = \{1, 2\}$. Once the repair has been completed, the product lot is routed to its corresponding product sink (sink blocks shown in Figure 7.2) and the production line becomes available to accept the next product lot in the queue, which triggers once again the Markov chain to determine the subsequent state of that production line. In this sense, when the production line is in an operative state ($O_s$), the actual manufacturing time of a product lot is equal to the theoretical manufacturing time. However, when the production line is in a failure state ($F_{T,s}$), the actual manufacturing time is equal to the theoretical manufacturing time plus the realization of the random variable $\Lambda_{l,T}$. 
Figure 7.5: Repair sub-system of production line 2.
7.2 Conclusion

The combination of DES and Markov chains provides a great flexibility to model realistic features of manufacturing systems. Markov chains enable us to keep track of the current state of a production line, and based on that information we can dynamically modify the different probabilities governing the system in order to model the occurrence and non-occurrence of failures in a more realistic manner. At the same time, this simulation approach enables us to consider different types of repair, e.g. near-perfect and imperfect repairs, and different failure types, i.e. the expected delay caused by repair services \( (\mu_{l,T}, T) \) depends on the failure type \( T = \{1, 2\} \), where \( \mu_{l,1} > \mu_{l,2} \). Finally, given that the probabilities of having serious and non-serious failures depend on the state of the production line (which accounts for near-perfect and imperfect repairs undertaken as well as the number of non-serious failures occurred previously), we can incorporate into the simulation model the deterioration caused by previous failures.
Chapter 8

Evolutionary Robust Optimization

(Manuscript 4)


8.1 Abstract

In this chapter, we aim to find robust solutions in optimization settings where there is uncertainty associated with the operating/environmental conditions, and the fitness of a solution is hence best described by a distribution of outcomes. In such settings, the nature of the fitness distribution (reflecting the performance of a particular solution across a set of operating scenarios) is of potential interest in deciding solution quality, and this is something that has not been analysed yet in previous chapters. Previous work has suggested the inclusion of robustness as an additional optimization objective. However, there has been limited investigation of different robustness criteria, and the impact this choice may have on the sample size needed to obtain reliable fitness
estimates. Here, we investigate different single and multi-objective formulations for robust optimization, in the context of a real-world problem addressed via SBO. For the (limited computational budget) setting considered here, our results highlight the value of an explicit robustness criterion in steering an optimizer towards solutions that are not only robust (as may be expected), but also associated with a profit that is, on average, higher than that identified by standard single-objective approaches. We also observe significant interactions between the choice of robustness measure and the sample size employed during fitness evaluation, an effect that is more pronounced for our multi-objective models.

8.2 Introduction

Real-world problems often require the consideration of multiple and often conflicting criteria (Deb, 2014; Wang et al., 2015; Mlakar et al., 2015). When faced with problems that incorporate uncertainty, one may arguably need to take into account the robustness of solutions (Branke and Lu, 2015) as an additional criterion of solution quality. Ideally, robust optimization would identify a solution that simultaneously offers the best and most robust performance; in reality, these two criteria are usually conflicting (Rooderkerk and Van Heerde, 2016), and a single optimal solution is unlikely to exist. Due to this expected trade-off between the quality of a solution and its robustness (Jin and Branke, 2005; Paenke et al., 2006), methodologies that are capable of identifying a desirable trade-off between these conflicting criteria are of significant interest in a robust optimization setting.

A complicating factor in robust optimization is the lack of a unique definition of the robustness of a solution. In some cases a solution is considered to be robust when, under certain levels of variation in the decision space\(^1\), it performs reasonably well in

\(^1\)Here we deal with variation in the decision space, which in turn generates variation in the objective
terms of quality (e.g. expected objective value) (Goh and Tan, 2009, p. 189), feasibility or optimality (Gabrel et al., 2014). Perhaps a more common idea is to consider as robust a solution that offers the best performance under the worst-case scenario (Gabrel et al., 2014) (e.g. maximizing the minimum value of a performance measure).

Independent of the exact definition of robustness, it has been argued that it may be advantageous to explicitly consider robustness as a separate objective (Jin and Sendhoff, 2003), instead of optimizing for a single measure that implicitly considers robustness (e.g. the average of a performance measure). This is because, in doing so (using e.g. a Pareto optimization approach), the trade-off between the main objective and its robustness measure may become more evident and facilitate decision making. On the other hand, more implicit considerations of robustness may have advantages in terms of the associated computational expense and their ease of implementation.

When EAs are applied as optimizers in single and multi-objective optimization problems, both approaches have a natural tendency to return robust solutions due to the implicit averaging mechanism intrinsic to an EA (Branke, 1998; Jin and Branke, 2005; Tsutsui and Ghosh, 1997). This tendency is emphasized further when explicit averaging (Jin and Branke, 2005) is implemented as a noise handling strategy. Given the above, our study sets out to investigate the extent to which evolutionary single-objective (ESO) optimization approaches (using an explicit averaging scheme) are already able to generate trade-off solutions between performance and robustness, and to what extent (and under what circumstances) these solutions are improved by evolutionary multi-objective (EMO) optimization approaches that consider robustness as a concrete optimization objective.
8.2.1 Robust Optimization

To some extent, robust solutions to a problem may be obtained by replacing the original objective(s) by its/their “robust” equivalent(s) (Jin and Branke, 2005), e.g. a robustness measure such as the average value instead of a single evaluation. Alternatively, measures of robustness may be integrated into the optimization as hard constraints (Goh and Tan, 2009, p. 216). A third possibility is to consider robustness as part of a composite function that is the weighted sum of different objectives (Deb, 2014, p. 408; Guo et al., 2008). This is, of course, a standard approach to transform a multi-objective optimization problem into a single-objective optimization problem, with a number of known limitations: the choice of appropriate values for the weight vector can be difficult and highly subjective, as it requires the translation of qualitative information into quantitative indicators (Deb, 2014, p. 409). Furthermore, the (single) solution obtained through this approach can be very sensitive to the relative weight assigned to each objective, and the method may thus be highly-user dependent. Although different solutions can be found by varying the weights, this approach is unable to find Pareto optimal solutions that lie on non-convex regions of the Pareto optimal front (Marler and Arora, 2010).

Pareto optimization overcomes these limitations by exploring different trade-off solutions between a set of objectives (Handl et al., 2007). This approach requires no prior knowledge of a preference vector and it returns a set of Pareto-optimal solutions to the decision maker. Thus, in Pareto optimization the aim is not only to find an optimal or near-optimal solution, but a set of diverse solutions of equal quality, called Pareto-optimal set. Solutions from this set dominate all solutions outside this set and none of them dominates other solution within this set. A solution dominates other solution if it is strictly better in at least one objective and not worse in any objective (Deb and Gupta, 2006; López-Ibáñez et al., 2010). The reader is referred to Deb (2014)
and Coello (2006) for more details about common concepts in Pareto optimization. In the context of robust optimization, we may use Pareto optimization to simultaneously optimize the original objective (performance measure) as well as a robustness measure. The Pareto front then serves to illustrate the trade-off between these objectives, and the decision maker may choose one solution from the set of Pareto-optimal solutions based on his experience and specific requirements.

8.2.2 Evolutionary Robust Optimization

Several previous works have searched for robust solutions via ESO optimization. Tsutsui and Ghosh (1997) proposed the single-evaluation mode (SEM) for GAs with a robust solution searching scheme, which consists of adding perturbations to the values of the decision variables, and then a single fitness evaluation is computed per individual. Under the SEM approach robust individuals are more likely to survive across generations than less robust ones. SEM can be seen as an equivalent to an implicit averaging strategy when uncertainty is present in the decision space. Also a multi-evaluation mode (MEM), which is equivalent to an explicit averaging strategy, was later proposed by Tsutsui (1999) to search for robust solutions with GAs. The author also proposed a MEM where the fitness of an individual corresponds to the worst fitness value (MEM-W) for situations where robustness is a crucial consideration. Branke (1998) analysed the potential of EAs to find robust solutions via different explicit averaging strategies. In Branke (1998), uncertainty is also modelled by adding random perturbations to the decision space before fitness evaluation. After analysing several strategies, the author concluded that both, re-evaluating fitter individuals and using previous fitness values to weight the fitness of those individuals are two promising ways to help EAs in the search for robust solutions. The MEM-W approach proposed by Tsutsui (1999) is later applied by Ong et al. (2006) within a max-min surrogate-assisted EA to generate robust
solutions for engineering design, since MEM-W was thought to be more conservative
in terms of solution robustness than MEM (explicit averaging).

Considerable research in robust optimization can also be found in the existing EMO
optimization literature. The explicit averaging strategy used in ESO optimization was
extended to EMO optimization by Deb and Gupta (2006). Two approaches to obtain
robust solutions are proposed in Deb and Gupta (2006). In the first approach, fitness
evaluations for the different objectives are computed by averaging a set of neighbour-
ing solutions in order to create perturbations in the decision space, which are later
reflected in the objective space. In the second approach the optimization is performed
based on fitness evaluations of the actual individuals (without perturbations) present in
the population and a constraint is added to the problem per every objective considered.
This constraint limits to a user-defined value the normalized, absolute or average nor-
malized difference between the perturbed fitness (which according to the authors can
be the average fitness or the worst fitness) and the original fitness (fitness of the actual
solution without perturbations). In this second approach the user can define the desired
level of robustness, since it is possible to limit the level of functional change caused
by local perturbations. In both approaches multiple fitness evaluations need to be com-
puted (using neighbouring solutions) to calculate the average fitness and the perturbed
fitness in the first and second case, respectively. In Jin and Sendhoff (2003), trade-off
solutions between robustness and performance were obtained by simultaneously op-
timizing both criteria via EMO optimization. According to Jin and Sendhoff (2003)
expectation measures such as the sample mean cannot sufficiently capture fluctuations
in performance, whereas variance-based measures fail to take into account the absolute
performance of a solution, thus both measures should be considered as two individual
objectives in the search for robust solutions. This idea of simultaneously optimizing
a performance and a robustness measure via EMO optimization has been applied in
several studies to search for robust solutions. For instance, Ray (2002) searched for

While many single and multi-objective studies have been published in the field of robust optimization, studies that directly compare ESO and EMO approaches to find robust solutions for the same problem are rare in the existing literature. Pereira et al. (2015) compared the ability of evolutionary single and multi-objective algorithms to find robust solutions (link weight configurations for traffic routing processes). However, in Pereira et al. (2015) a solution was considered to be robust when it performed well under two different network conditions; in this sense, two performance measures (rather than one performance measure and one robustness measure) were optimized. The weighted sum of two congestion functions (two performance measures) was applied as fitness function in the ESO algorithm, whereas in both EMO algorithms, the non-dominated sorting genetic algorithm II (NSGA II) (Deb et al., 2002) and strength Pareto evolutionary algorithm 2, both congestion functions were optimized separately. The performance advantage of NSGA II was evident under more complex network topologies and more demanding traffic requirements.

8.2.3 Contributions

In this chapter, we investigate the extent to which ESO and EMO optimization are able to generate comparable trade-off solutions between performance and robustness by analysing the sets of non-dominated solutions obtained with the single and multi-objective algorithms introduced in Section 8.3.1.1. Those non-dominated solutions are
identified based on estimates computed via stochastic simulation for the sample mean, sample standard deviation and sample minimum of every feasible solution present in the final population obtained with the different models analysed.

Furthermore, we investigate how the use of the sample minimum as robustness measure may affect the optimization performance of EAs compared to the use of less biased statistics such as the sample mean or the sample standard deviation. While, many measures of robustness have been proposed in the existing literature (Goh et al., 2010), the sample minimum (worst case scenario for maximization problems) is particularly popular (Gabrel et al., 2014), see Chen et al. (2016), Zhang et al. (2016), Goh and Tan (2007), Ong et al. (2006) and Tsutsui (1999) for some examples. It is also often stated that by using the sample minimum as robustness measure more robust solutions can be obtained (Ong et al., 2006; Tsutsui, 1999). However, very limited attention has been given to how the choice of a robustness measure may affect the optimization performance of single and multi-objective EAs, and how the sample minimum compares to other measures. As statistics tell us that the sample minimum is a biased estimator for the population (López-Ibáñez et al., 2010), it is possible that the use of the sample minimum as a robustness measure may mislead the selection operator and steer the EA towards undesirable regions of the objective space.

Finally, we investigate how the level of noise in fitness estimates (which we directly control through the sample size used during fitness evaluation) affects the optimization performance of the single and multi-objective optimizers analysed. Since we allocate the same computational budget (total number of simulation replications) to all optimization runs across models, more reliable fitness estimates imply shorter optimization runs (fewer generations). All three of these questions are considered for different levels of uncertainty, in order to analyse how their answers may change with increasing levels of uncertainty.

The remainder of this chapter is organized as follows: In Section 8.3, we introduce
the problem analysed and describe the ESO and EMO models. In Section 8.4, we introduce the details of the benchmark analysis, and the results obtained are presented in Section 8.6. Finally, conclusions derived from this study are given in Section 8.7.

8.3 Simulation-Based Optimization Model

In production planning, profitability is not the only criterion that needs to be considered during the specification of a production plan, but the robustness of that plan is also relevant (Paenke et al., 2006).

In this study we address a production planning problem of a real manufacturing system. Detailed features of this manufacturing system are presented in Chapter 2. The robustness of production plans is a criterion that needs to be considered here because the system analysed is subject to failures of its production lines, which may cause deviations between a plan and its actual outcome. Consequently, we aim to identify a set of production plans that makes evident the trade-off between profitability and robustness, instead of being interested in finding the most profitable production plan in terms of average profit as in Chapter 5.

Here, a production plan $\mathbf{x}$ is a vector of 41 decision variables, where each decision variable $x_{l,j}$ indicates the number of lots of product $j$ to be manufactured in production line $l$ during a finite planning horizon of one month. This batch manufacturing system has a set $M = \{1,2,\ldots,7\}$ of independent production lines and is able to manufacture 31 different products. A feature of this system is that only a subset $A_j \subseteq M$ of production lines can manufacture a specific product $j$. Additionally, this manufacturing system has insufficient capacity to fully cover demand requirements and is subject to serious and non-serious failures of its production lines. We also consider that near-perfect and imperfect repairs can be undertaken after the occurrence of a failure as well as the deterioration caused by previous failures.
SBO is an attractive approach to address this problem, for it can accurately capture the inherent uncertainty and complexity of the manufacturing system, while searching for near-optimal solutions for the problem analysed (Figueira and Almada-Lobo, 2014; Syberfeldt et al., 2010; Ehrenberg and Zimmermann, 2012). The SBO model consists of a simulation model and an EA coupled in a black-box fashion. Production plans specified by the optimizer are simulated by the simulation model, which returns the realization of that production plan. Those simulated responses are then used by the optimizer to compute the fitness values of a given production plan.

The simulation model used in this study is the one presented in Chapter 7. This simulation model developed in SimEvents® (The MathWorks, Inc., 2014) models the manufacturing system analysed. It employs a combination of DES and Markov chains to model realistic features of manufacturing systems such as different types of production line failure, near-perfect and imperfect repairs as well as the deterioration of production lines due to previous failures.

The reader is referred to Chapter 7 for a detailed description of the simulation model, whereas further details about the optimization model are provided in Section 8.3.1.

### 8.3.1 Optimization Model

In Chapter 5, we introduced the $d$-MBKARS problem as a generalization of the production planning problem addressed in that chapter. Here, we address an extended version of that problem that considers a robustness measure as a second objective. In this problem we do not penalize deviations from a plan and we refer to it as the multi-objective $d$-MBKARS (M-$d$-MBKARS) problem without penalty and its ILP formulation, in the context of the problem analysed, is as follows:
maximize \( f_1(x) = \frac{1}{\gamma} \sum_{r=1}^{\gamma} \sum_{j=1}^{31} \sum_{l \in A_j} v_j \times s_{l,j,r} \)  

(8.1)

and either

\[ \text{minimize } f_2(x) = \left[ \frac{1}{\gamma} \sum_{r=1}^{\gamma} \left( \sum_{j=1}^{31} \sum_{l \in A_j} v_j \times s_{l,j,r} - f_1(x) \right) \right]^2 \]

(8.2)

or

maximize \( f_3(x) = \min_{r=1,2,\ldots,\gamma} \sum_{j=1}^{31} \sum_{l \in A_j} v_j \times s_{l,j,r} \)  

(8.3)

subject to

\[ \sum_{j=1}^{31} \sum_{l \in A_j} w_{i,j} \times x_{l,j} \leq c_i \quad (i = 1, 2, \ldots, 13), \]

(8.4)

\[ \sum_{l \in A_j} x_{l,j} \leq b_j \quad (x_{l,j} \in \mathbb{Z}_{\geq 0}; \ j = 1, 2, \ldots, 31). \]

(8.5)

where \( s_{l,j,r} \) is obtained by simulating a production plan \( x \) (vector that contains all decision variables \( x_{l,j} \)) using the simulation model. In this sense, \( s_{l,j,r} \) corresponds to the number of lots of product \( j \) manufactured in production line \( l \) during the \( r \)-th simulation replication given a production plan \( x \). \( \gamma \) is the total number of simulation replications.
performed to compute fitness of a production plan, please see Section 8.3.1.1 for details about $\gamma$. The simulation model can be seen here as the function $g(x)$ able to map a production plan $x$ onto the actual production $s_r$ (vector that contains all simulated responses $s_{l,j,r}$) without the requirement of a closed form formulation. In other words $s_r$ is the realization of a production plan $x$ during the $r$-th simulation replication. $v_j$ is the marginal (per lot) contribution to profit of product $j$, $w_{i,j}$ is the amount of resource $i$ deployed by manufacturing one lot of product $j$ and $c_i$ is the amount of resource $i$ available at the beginning of the manufacturing process.

We represent in the form of Equation 8.4 the design capacities of production lines, the amount of labour needed to manufacture one lot of product $j$ in production line $l$ and the situation where a product manufactured in this system is subsequently used as raw material during the manufacturing of another product. We also impose additional constraints in the form of Equation 8.5 to avoid that the production level of product $j$ exceeds its maximum demand $b_j$. Please note that in this study, a production plan $x$ is feasible if it satisfies the set of constraints imposed as Equations 8.4 and 8.5.

### 8.3.1.1 Evolutionary Optimizers

For the multi-objective SBO (MSBO) approach, we require an optimizer able to find a set of solutions that are non-dominated and are widely spread over the approximated Pareto front. EAs are attractive optimizers for Pareto optimization not only due to their wide applicability, flexibility (Coello, 2006) and ease of use (Coello, 2015), but also because multiple non-dominated solutions can be found in a single iteration (Branke et al., 2016; Deb, 2014; Coello, 2006). The optimizer used in the MSBO approach is NSGA II, a well-known EMO algorithm commonly employed as benchmark in multi-objective optimization (Coello, 2006), that has been successfully used to solve SBO problems (see Brownlee and Wright (2015) and Sanchez et al. (2010) for recent examples).
In the MSBO approach, in addition to optimizing a performance measure (maximize average profit), we simultaneously optimize a robustness measure in order to explicitly capture as a second objective the robustness of a production plan. Two different measures of robustness are evaluated in the multi-objective approach, the standard deviation of profit and the minimum profit. The minimization of the standard deviation tries to reduce the variation in profitability, whereas the maximization of the minimum profit tries to improve the worst case scenario derived from a production plan. In this sense the first MSBO model (MSBO-1) tries to simultaneously maximize average profit (Equation 8.1) and minimize the standard deviation of profit (Equation 8.2), subject to the set of constraints in the form of Equations 8.4 and 8.5. The second MSBO model (MSBO-2) differs from MSBO-1 in that, instead of minimizing the standard deviation, MSBO-2 tries to maximize minimum profit (Equation 8.3). Both multi-objective models, MSBO-1 and MSBO-2, use NSGA II as optimizer.

Two different single-objective SBO (SSBO) models are also evaluated, the first SSBO model (SSBO-1) tries to maximize average profit (Equation 8.1) subject to the set of constraints in the form of Equations 8.4 and 8.5, whereas the second SSBO model (SSBO-2) tries to maximize minimum profit (Equations 8.3), subject to the same set of constraints. Both single-objective models employ as optimizer the MI-LXPM algorithm proposed by Deep et al. (2009), but here, there is no need to implement a final solution selection procedure (as done in Chapter 5, since we are interested in the entire final population.

In order to make a fair comparison between MSBO and SSBO approaches, we use the same parameters and operators for both optimizers, except for unavoidable discrepancies that arise due to the different mechanics of both GAs. Both optimizers use a population size of 100 individuals and employ binary tournament selection, Laplace crossover (crossover probability: 0.8) and power mutation (mutation probability: 0.005) as operators. See Deep et al. (2009) for more details about the last two
operators and the parameter values used. Also the constraint-handling method pro-
posed by Deb (2000) and the truncation procedure described in Deep et al. (2009),
which ensures compliance with integer constraints after crossover and mutation, are
implemented in both optimizers.

It is important to mention that the selection criterion in NSGA II is based on the
crowded-comparison operator (Deb et al., 2002), although it employs a binary tourna-
ment selection operator. Moreover, the way elitism is incorporated in both optimizers
diffs. In MI-LXPM, elitism is incorporated by having an elite set with a default size
of one (Deep and Thakur, 2007a,b). We do not adjust this parameter, as we are aiming
to maintain good levels of diversity throughout the optimization process. In NSGA II
elitism is incorporated by combining the entire parent and offspring populations before
extracting the new population.

Given the stochastic nature of simulation responses \((s_{l,j,r})\) and due to the nature of
the objectives optimized (a sample of independent fitness values is required to com-
pute the fitness of an individual), all fitness values are computed across a number \(\gamma\) of
independent fitness evaluations. Two explicit averaging strategies are implemented in
every single and multi-objective model. In strategy A, fitness is computed across 10
independent fitness evaluations \((\gamma = 10)\), whereas in strategy B, fitness is computed
across 30 independent fitness evaluations \((\gamma = 30)\). The letter A or B is added to the
notation of the different models according to the explicit averaging strategy that they
employ, e.g. MSBO-1A and MSBO-1B.

Explicit averaging is a very expensive approach in terms of computational effort.
This computational expense becomes even more critical when explicit averaging is ap-
plied in the context of SBO, where expensive simulation replications are required to
compute multiple fitness values of entire populations. Therefore, in order to boost the
optimization performance of NSGA II and MI-LXPM, and reduce their computational
effort, we devise the specialized initialization operator presented in the following sub-
section. We also use a parallel implementation in all models to speed up computations.

8.3.2 Initialization Operator

This initialization operator creates up to 22 individuals of the GA’s initial population
and the rest of individuals are randomly initialized. In order to support both objec-
tives, the initialization operator is implemented in two phases. The first phase creates
solutions that are biased towards maximization of average profit, whereas the second
phase tries to generate solutions that are biased towards minimization of the standard
deviation of profit.

In the first phase, 200 numerical values of $c'_l$ are generated. $c'_l$ is the number of
hours that production line $l$ was available during a working month (24 days) and is
computed as follows:

$$c'_l = \sum_{j=1}^{31} s_{l,j} \times t \quad (l = 1, 2, \ldots, 7). \quad (8.6)$$

where $s_{l,j}$ is obtained by simulating in the simulation model a production plan that
fully utilizes the theoretical capacity of all production lines (see Section 5.5.2.1 for
more details about this production plan) and $t$ corresponds to the manufacturing time
per lot (in hours) when no failure occurs, which according to the company is 8 hours
for all product lots. Subsequently, for each set of $c'_l$ an ILP solution is found by solving
the following ILP problem:

$$\text{maximize} \quad f(x) = \sum_{j=1}^{31} \sum_{l \in A_j} v_j \times x_{l,j} \quad (8.7)$$

subject to
Here, capacities of production lines are not represented in the form of Equation 8.8 as other resource constraints, but in the form of Equation 8.9, where \( c'_l \) is used as the RHS.

Duplicates among those 200 ILP solutions are eliminated in order to maintain diversity and only 10 ILP solutions are chosen, as described bellow, to be part of the set \( W \), which is incorporated into the GA’s initial population. If the number of ILP solutions left is less than or equal to 10, all of them are included into \( W \), whereas if the number of solutions left is greater than 10, average fitness is computed for every ILP solution across 30 independent fitness evaluations by solving Equation 8.1 with \( \gamma = 30 \). The 10 solutions with the highest average profit are then incorporated into \( W \).

The second phase of this initialization operator simulates (using the simulation model) every ILP solution contained in \( W \) and incorporates into the GA’s initial population one simulated response \( s \) per each ILP solution contained in \( W \). Those simulated responses \( (s) \) might evidence a poor performance in terms of average profit, since they may not utilize efficiently the resources of the manufacturing system, but their standard deviations of profit should be lower than or equal to the standard deviations of the corresponding ILP solutions from which the simulated responses were obtained. This is because \( s_{l,j} \leq x_{l,j} \), and thus \( s_{l,j} \) is more likely to be fully realized than \( x_{l,j} \).

Finally, we also incorporate into the GA’s initial population the solution \( x^* \) obtained by optimizing Equation 8.7 subject to the set of constraints in the form of Equations 8.4 and 8.5, as well as the solution \( x'^* \) obtained by optimizing Equation 8.7, subject to the set of constraints in the form of Equation 8.5, Equation 8.8 and the following:
\[ \sum_{j=1}^{31} w_{l,j} \times x_{l,j} \leq \mathbb{E}[C_l] \quad (l = 1, 2, \ldots, 7), \]  

(8.10)

where the expected effective capacity \( \mathbb{E}[C_l] \) of production line \( l \) is used as the RHS of the corresponding constraint. \( \mathbb{E}[C_l] \) is computed as follows:

\[
\mathbb{E}[C_l] = c_l - \frac{c_l}{l} \times \sum_{T=1}^{2} \sum_{s=0}^{3} p_{l,T}^{F_{T,s}} \times \mathbb{E}[\Lambda_{l,T}] \quad (l = 1, 2, \ldots, 7),
\]

(8.11)

where \( \mathbb{E}[\Lambda_{l,T}] \) is the expected value (given in hours) of the random variable \( \Lambda_{l,T} \), which models the delay caused by a repair service. \( p_{l,T}^{F_{T,s}} \) is the steady state probability (Ching et al., 2006b, p. 13–18) of state \( F_{T,s} \), please see Chapter 7 for details about state \( F_{T,s} \).

\( x^* \) and \( x'^* \) are incorporated into the GA’s initial population because they might contain useful alleles, especially under low levels of fitness variability.

### 8.4 Benchmark Analysis

The benchmark analysis undertaken here aims to investigate the following points:

First, we want to analyse the impact that the common practice in robust optimization of maximizing the worst case scenario (Gabrel et al., 2014) has on the optimization performance of the MSBO and SSBO approaches studied here. To do this we compare the solutions returned by multi-objective models that use the sample standard deviation as robustness measure (MSBO-1A, MSBO-1B) to the solutions obtained with multi-objective models that use the sample minimum instead (MSBO-2A, MSBO-2B). We also compare the solutions returned by single-objective models that maximize average profit (SSBO-1A, SSBO-1B) to the solutions obtained with single-objective models that maximize minimum profit (SSBO-2A, SSBO-2B).
Second, we intend to investigate how the allocation of limited computations between refinement of fitness values and longer optimization runs affects the optimization performance of the single and multi-objective models analysed, and whether the impact of this differs between the different formulations. To do this we compare (for every single and multi-objective model) two sets of solutions: the ones obtained when fitness values are computed across 10 (explicit averaging strategy A) and 30 (explicit averaging strategy B) fitness evaluations.

Third, we want to investigate to what extent ESO optimization is able to generate, from a diverse final population, trade-off solutions between performance and robustness by optimizing a single measure that implicitly considers robustness, and how this compares to EMO optimization, where robustness is considered explicitly as an additional objective. To do this, we compare the solutions returned by the different single and multi-objective models.

Finally, in order to understand how performance differences evolve with increasing uncertainty levels across the different models analysed, we address the points mentioned above in two different problem instances, presented in Table 8.1. Table 8.1 shows the values assigned to \( p_l, \zeta_l, \varphi_l, \) and \( \mu_{l,T} \), which are necessary to compute (as shown in Chapter 7) the transition probability matrix of each production line for both problem instances analysed in this study. \( p_l \) is the probability that a production line failure occurs (either serious or non-serious) during the manufacturing of the next product lot, given that the production line is in state \( O_0 \) (no non-serious failure has occurred after a near-perfect repair and the production line was operative during the manufacturing of the current product lot). \( \zeta_l \) determines the probability that a serious failure occurs during the manufacturing of the next product lot, given that the production line is in state \( O_0 \) and given that a production line failure occurs during the manufacturing of the next product lot. The deterioration of a production line caused by a non-serious failure is modelled by raising \( p_l \) and \( \zeta_l \) to the power of \( \varphi_l \). The random variable \( \Lambda_{l,T} \)
is modelled by an exponential PDF with known mean $\mu_{l,T}$. The value of $\mu_{l,T}$ depends on the failure type of the production line, where $T = 1$ denotes a serious failure and $T = 2$ a non-serious one. Please see Chapter 7 for a detailed description of the simulation model. Please note that artificial values have been assigned to the different $p_l$, $\zeta_l$, $\phi_l$ and $\mu_{l,T}$, due to the difficulty of obtaining real data for those parameters. In this manufacturing system, some products can be manufactured in several production lines (production lines 2, 3 and 4), and according to historical information provided by the company most of the failures occur on production lines 2, 3 and 4 (see Table 2.5), which makes them crucial for the robustness of a production plan. For this reason only those production lines have been assigned with a positive $p_l$.

<table>
<thead>
<tr>
<th>Production line ($l$)</th>
<th>$p_l$</th>
<th>$p_l$</th>
<th>$\mu_{l,1}$ (d)</th>
<th>$\mu_{l,2}$ (d)</th>
<th>$\zeta_l$</th>
<th>$\phi_l$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>2</td>
<td>0.05</td>
<td>0.10</td>
<td>1.00</td>
<td>0.33</td>
<td>0.02</td>
<td>0.80</td>
</tr>
<tr>
<td>3</td>
<td>0.15</td>
<td>0.20</td>
<td>1.50</td>
<td>0.60</td>
<td>0.05</td>
<td>0.50</td>
</tr>
<tr>
<td>4</td>
<td>0.10</td>
<td>0.15</td>
<td>1.00</td>
<td>0.33</td>
<td>0.04</td>
<td>0.70</td>
</tr>
<tr>
<td>5</td>
<td>0.00</td>
<td>0.00</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>6</td>
<td>0.00</td>
<td>0.00</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>7</td>
<td>0.00</td>
<td>0.00</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

Table 8.1: Values of $p_l$, $\mu_{l,T}$, $\zeta_l$ and $\phi_l$ per production line and problem instance.

In order to investigate the points mentioned above, 30 independent runs of every model are executed on both problem instances, using different seeds to initialize the random number generator in every run. To make a fair comparison, all models are evaluated upon a limited computational budget of 45000 independent simulation replications per run, excluding the fitness evaluations needed by the initialization operator. In this sense, a run of models MSBO-1A, MSBO-2A, SSBO-1A and SSBO-2A terminates after 45 generations, whereas a run of models MSBO-1B, MSBO-2B, SSBO-1B and SSBO-2B terminates after 15 generations. The main features of every model are
presented in Table 8.2, those parameters have been set based on preliminary experiments. All those models are implemented in Matlab® R2014a (The MathWorks, Inc., 2014) and all computations are executed in parallel on a 12 core Intel(R) Xeon(R) CPU L5640 @ 2.27GHz with 24 GB of RAM running Scientific Linux, release 6.2.

<table>
<thead>
<tr>
<th></th>
<th>Objective 1</th>
<th>Objective 2</th>
<th>$\gamma$</th>
<th>Generations</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSBO-1A</td>
<td>max mean</td>
<td>min std. dev.</td>
<td>10</td>
<td>45</td>
</tr>
<tr>
<td>MSBO-2A</td>
<td>max mean</td>
<td>max minimum</td>
<td>10</td>
<td>45</td>
</tr>
<tr>
<td>MSBO-1B</td>
<td>max mean</td>
<td>min std. dev.</td>
<td>30</td>
<td>15</td>
</tr>
<tr>
<td>MSBO-2B</td>
<td>max mean</td>
<td>max minimum</td>
<td>30</td>
<td>15</td>
</tr>
<tr>
<td>SSBO-1A</td>
<td>max mean</td>
<td>—</td>
<td>10</td>
<td>45</td>
</tr>
<tr>
<td>SSBO-2A</td>
<td>max minimum</td>
<td>—</td>
<td>10</td>
<td>45</td>
</tr>
<tr>
<td>SSBO-1B</td>
<td>max mean</td>
<td>—</td>
<td>30</td>
<td>15</td>
</tr>
<tr>
<td>SSBO-2B</td>
<td>max minimum</td>
<td>—</td>
<td>30</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 8.2: Objective/s, $\gamma$ and number of generations across the different models analysed.

8.5 Performance Assessment

To assess the “real” quality of the final population returned by the different optimizers (as compared to the “fitness estimates” used during the actual optimization process), we use a large number of simulation runs (100) to determine the profit distribution for each solution. These results are then used to identify the subsets of solutions that are non-dominated with respect to their average profit and standard deviation/minimum profit (measured in United States Dollar (USD)).

The performance assessment of different optimizers in terms of the quality of solutions is a complex issue when multiple objectives are involved (Minella et al., 2011). This assessment becomes even more complicated when EAs are used as optimizers, since a different set of solutions may be returned in every run of each EA (due to the randomized search mechanism of EAs) (Knowles et al., 2006).
Different approaches to assess performance of multi-objective optimizers have been proposed in the existing literature, the reader is referred to Knowles et al. (2006) for a comprehensive review of existing techniques. Here, in order to evaluate the points of interest mentioned at the beginning of this section (Section 8.4), we use the multiple sets of non-dominated solutions obtained with every model (MSBO-1A, MSBO-1B, MSBO-2A, MSBO-2B, SSBO-1A, SSBO-1B, SSBO-2A and SSBO-2B) to compute samples of values for the hypervolume indicator (Fleischer, 2003). We then conduct Mann-Whitney U tests (Mann and Whitney, 1947) on the samples of hypervolume values in order to derive statistical inferences about the relative performance of two models expressed in the form of the following hypotheses:

\[ H_0 : \text{stochastic homogeneity of hypervolume values} \]
\[ H_a : \text{stochastic heterogeneity of hypervolume values} \]

In this sense, Mann-Whitney U tests tell us if one model outperforms other model in terms of the quality of approximation sets generated, under the assumption that the hypervolume indicator reflects the preference of the decision maker (Knowles et al., 2006). A non-parametric test (Mann-Whitney U test) is employed because distributions of the samples analysed do not fulfil the normality assumption.

The use of the hypervolume indicator might favour multi-objective formulations over single-objective formulations given that diversity is explicitly considered in the former models, but not in the latter ones. In order to resolve this potential issue and make a fair comparison between single and multi-objective models, we apply the attainment function approach (da Fonseca and Fonseca, 2010; da Fonseca et al., 2001). This approach summarizes the outcome of multiple runs of the different models in the form of empirical attainment functions \(^2\) (EAF) (Binois et al., 2015), and thus enable

---

\(^2\)An attainment function describes the probability of an algorithm finding at least one solution whose objective vector dominates or is equal to a specific objective vector in a single run (da Fonseca et al., 2001; Fonseca and Fleming, 1996). The attainment function can be approximated from the outcomes of several independent runs of an algorithm, such approximation is known as the empirical attainment function (López-Ibáñez et al., 2010)
us to identify the region/s where performance differences arise between single and multi-objective models, in both objective spaces analysed (i average profit vs. standard deviation of profit and ii average profit vs. minimum profit). We employ the eaf package presented in López-Ibáñez et al. (2010) to create plots of the median (objective vectors attained by at least half of the runs) and best (best objective vector ever achieved) attainment surfaces as well as to illustrate the differences between EAFs obtained with different models.

8.6 Results

Yet, given that the sample minimum presents a biased estimator for the population (López-Ibáñez et al., 2010), it is possible that its as a robustness measure may mislead selection operators and steer an EA towards undesirable regions of the objective space. Our results provide evidence supporting the idea that the use of less biased statistics as robustness measures reduces the computational effort needed to obtain reliable fitness estimates, thus reducing the likelihood of choosing bad solutions over good ones during the optimization and making it possible to allocate more computational effort to the optimization procedure. This can be seen in results from Mann-Whitney U tests (presented in Tables 8.3 and 8.4 for problem instance 1 and Tables 8.5 and 8.6 for problem instance 2 located Appendix 8.A), which statistically show that in both objective spaces analysed (i average profit vs. standard deviation of profit and ii average profit vs. minimum profit) multi-objective models that used the sample standard deviation as robustness measure (MSBO-1A and MSBO-1B) achieved stochastically larger ($p < .01$) hypervolume values than their counterparts (MSBO-2A and MSBO-2B) that used the sample minimum as robustness measure. Figures 8.1 and 8.2 also support this finding in problem instance 1, by showing that the CDFs of hypervolume values obtained with MSBO-2A and MSBO-2B are dominated (first and second-order
For the multi-objective models, our results also reveal that, given the number of fitness evaluations considered here, the allocation of more computations to obtain more reliable fitness estimates (implicit averaging strategy B) is a better strategy than optimizing across a higher number of generations using less reliable fitness estimates (implicit averaging strategy A). This finding is supported by Figures 8.1 and 8.2 for problem instance 1, which show that in both objective spaces (i) average profit vs. standard deviation of profit and (ii) average profit vs minimum profit) CDFs of hypervolume values obtained with multi-objective models implementing explicit averaging strategy B (MSBO-1B and MSBO-2B) dominate (first-order stochastic dominance) the CDFs of hypervolume values obtained with their corresponding counterparts implementing strategy A (MSBO-1A and MSBO-2A). This finding is also corroborated by results.
Figure 8.2: CDFs of hypervolume values obtained with multi-objective models (MSBO-1A, MSBO-2A, MSBO-1B, MSBO-2B) in the objective space of average profit vs. minimum profit in problem instance 1.

from Mann-Whitney U tests, presented in Tables 8.3 and 8.4, which reveal that hypervolume values obtained with both multi-objective approaches implementing strategy B (MSBO-1B and MSBO-2B) are stochastically larger \((p < .01)\) than the ones obtained with their corresponding counterparts implementing strategy A (MSBO-1A and MSBO-2A), in both objective spaces. Supportive results for problem instance 2 can be found in Figures 8.9 and 8.10 and in Tables 8.5 and 8.6 presented in Appendix 8.A.

It is important to note that in both problem instances, having more reliable fitness estimates significantly improved \((p < .01)\) the optimization performance of all multi-objective models in both objective spaces analysed \(i\) average profit vs. standard deviation of profit and \(ii\) average profit vs. minimum profit. For single-objective models in problem instance 1, the implementation of strategy B significantly improved \((p < .01)\) the optimization performance of the single-objective model trying to maximize the sample minimum, but only in the solution space of average profit vs. standard
Figure 8.3: CDFs of hypervolume values obtained with the best multi-objective model (MSBO-1B) and all single-objective models (SSBO-1A, SSBO-2A, SSBO-1B and SSBO-2B) in the objective space of average profit vs. standard deviation of profit.
Figure 8.4: CDFs of hypervolume values obtained with the best multi-objective model (MSBO-1B) and all single-objective models (SSBO-1A, SSBO-2A, SSBO-1B and SSBO-2B) in the objective space of average profit vs. minimum profit.
deviation (see Table 8.3 and 8.5, where SSBO-2B achieved stochastically larger hypervolume values ($p < .01$) than SSBO-2A), whereas under higher uncertainty level (problem instance 2), strategy B improved the optimization performance of the model mentioned above not only in the solution space of average profit vs. standard deviation, but also in the solution space of average profit vs. minimum profit (see Figures 8.3b and 8.4b, where the CDF of hypervolume values obtained with SSBO-2B dominate the CDFs of hypervolume values obtained with SSBO-2A - first and second-order stochastic dominance, respectively). These results are interesting because they provide evidence that using less biased statistics as objective/s might reduce the computational effort (number of samples) needed to obtain reliable fitness estimates, and thus demonstrate that not only the level of uncertainty, but also the choice of objective/s have an impact on the noise sensitivity of the optimization approach. Our results also seem to suggest that the multi-objective approach is more sensitive to noise than the single-objective approach, but further investigation is still required to support this claim. This may occur because in the single-objective case the solution selection during the optimization procedure only considers one noisy fitness estimate, whereas in the multi-objective case the noise affecting the optimization comes from two different noisy fitness estimates, which makes the solution selection more difficult. Although we cannot claim that multi-objective approaches are in general more noise sensitive than the single-objective approaches, as this will highly depend on the choice of objective/s, we can say that our experiments demonstrate that for the multi-objective models considered here, allocating more computational effort towards the generation of more reliable fitness estimates seems to be more beneficial than optimizing across more generations.

Our results identified MSBO-1B as the best multi-objective model in both objective spaces analysed in problem instance 1 and 2. In the solution space of average profit vs. standard deviation of profit, this finding is confirmed by results from Mann-Whitney
U tests (Tables 8.3 and 8.5) and by analysing Figures 8.1 and 8.9, which indicate that MSBO-1B outperformed all its multi-objective contestants (MSBO-1A, MSBO-2A and MSBO-2B). Furthermore, hypervolume values from non-dominated solutions found with MSBO-1B in the solution space of average profit vs. minimum profit were stochastically larger (\( p < .01 \)) (see Tables 8.4 and 8.6) than hypervolume values obtained with any other multi-objective model (MSBO-1A, MSBO-2A and MSBO-2B), in both problem instances. For problem instance 1, these results are corroborated by Figure 8.2, where the CDF of hypervolume values obtained with MSBO-1B dominates the CDFs obtained with MSBO-1A (first-order stochastic dominance), MSBO-2A (first-order stochastic dominance) and MSBO-2B (second-order stochastic dominance).

It is not surprising that MSBO-1B outperforms both, MSBO-2A and MSBO-2B, in the solution space of average profit vs. standard deviation of profit, since MSBO-1B explicitly optimizes both criteria. However, it is surprising that MSBO-1B also outperforms MSBO-2A and MSBO-2B in the solution space of average profit vs. minimum profit, considering that maximization of minimum profit is an objective in MSBO-2A and MSBO-2B, but not in MSBO-1B. This counter-intuitive result further supports our finding that the use of the sample minimum as robustness measure may undermine the optimization performance of the EAs analysed, instead of returning more robust solutions, as reported in Ong et al. (2006) and Tsutsui (1999), compared to less biased statistics such as the sample mean or the sample standard deviation. Having a more biased statistic (sample minimum) as one of the two noisy fitness estimates may be the reason why MSBO-2B was outperformed in both objective spaces (based on CDFs and results from Mann-Whitney U tests) by MSBO-1B, which uses a less biased statistic (sample standard deviation) as fitness estimate. It is probably due to the less reliable fitness estimates that MSBO-1A performed so badly (based on CDFs and results from
Mann-Whitney U tests) in both objective spaces in terms of hypervolume values compared to MSBO-1B, which has more refined fitness estimates.

Our results also demonstrate that the simultaneous optimization of average profit as a performance measure and standard deviation of profit as a robustness measure (MSBO-1B) consistently generated better sets of trade-off solutions not only between average profit and standard deviation of profit, but also between average profit and minimum profit, than any other model here analysed. The supporting evidence for this finding is provided by results from Mann-Whitney U tests (Tables 8.3 and 8.4 for problem instance 1 and Tables 8.5 and 8.6 for problem instance 2), which determined that MSBO-1B achieved stochastically larger ($p < .01$) hypervolume values than any of the single-objective models in both objective spaces analysed. The superiority of MSBO-1B is also illustrated in Figures 8.3 and 8.4, where the CDFs of hypervolume values obtained with MSBO-1B dominate (first-order stochastic dominance) all the CDFs obtained with the single-objective models (SSBO-1A, SSBO-2A, SSBO-1B and SSBO-2B).

Finally, the median and best attainment surfaces, presented in Figures 8.5, 8.6, 8.7 and 8.8, illustrate that in problem instance 1 and 2, respectively, the regions attained in both objective spaces with the best multi-objective model (MSBO-1B) are more attractive than the ones attained with the different single-objective models (SSBO-1A, SSBO-2A, SSBO-1B and SSBO-2B). The advantage of MSBO-1B over single-objective models is also confirmed by looking at the shaded areas in those figures. Shaded areas point out the regions of the objective space where the EAF of one model is larger by at least 20% than the EAF of the other model, and thus dark regions indicate the location in the objective space where one model outperformed the other. Darker areas indicate larger differences between the estimated probability values of two models.

This finding is somewhat surprising when considering performance for one of the
Table 8.3: Values for Mann-Whitney U statistic obtained from pairwise comparisons of hypervolume values of all single and multi-objective models in the objective space of average profit vs. standard deviation of profit in problem instance 1.

<table>
<thead>
<tr>
<th></th>
<th>MSBO-1A</th>
<th>MSBO-1B</th>
<th>MSBO-2A</th>
<th>MSBO-2B</th>
<th>SSBO-1A</th>
<th>SSBO-1B</th>
<th>SSBO-2A</th>
<th>SSBO-2B</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSBO-1A</td>
<td>—</td>
<td>2**</td>
<td>14**</td>
<td>370</td>
<td>82**</td>
<td>94**</td>
<td>331</td>
<td>337</td>
</tr>
<tr>
<td>MSBO-1B</td>
<td>—</td>
<td>—</td>
<td>0**</td>
<td>6**</td>
<td>74**</td>
<td>135**</td>
<td>7**</td>
<td>12**</td>
</tr>
<tr>
<td>MSBO-2A</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>5**</td>
<td>0**</td>
<td>9**</td>
<td>115**</td>
<td>0**</td>
</tr>
<tr>
<td>MSBO-2B</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>111**</td>
<td>125**</td>
<td>280*</td>
<td>395</td>
</tr>
<tr>
<td>SSBO-1A</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>364a</td>
<td>83**</td>
<td>174**</td>
</tr>
<tr>
<td>SSBO-1B</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>89**</td>
<td>157**</td>
</tr>
<tr>
<td>SSBO-2A</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>233**</td>
</tr>
<tr>
<td>SSBO-2B</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

* $p < .05$; ** $p < .01$; $a$ heteroscedasticity according to non-parametric Levene test ($p < .05$) (Nordstokke and Zumbo, 2010)
** Table 8.4: Values for Mann-Whitney U statistic obtained from pairwise comparisons of hypervolume values of all single and multi-objective models in the objective space of average profit vs. minimum profit in problem instance 1.**

```
<table>
<thead>
<tr>
<th></th>
<th>MSBO-1A</th>
<th>MSBO-1B</th>
<th>MSBO-2A</th>
<th>MSBO-2B</th>
<th>SSBO-1A</th>
<th>SSBO-1B</th>
<th>SSBO-2A</th>
<th>SSBO-2B</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSBO-1A</td>
<td>—</td>
<td>99**</td>
<td>112**</td>
<td>294*</td>
<td>334</td>
<td>308*</td>
<td>299*</td>
<td>314*</td>
</tr>
<tr>
<td>MSBO-1B</td>
<td>—</td>
<td>—</td>
<td>1**</td>
<td>310**</td>
<td>71**</td>
<td>36**</td>
<td>62**</td>
<td>76**</td>
</tr>
<tr>
<td>MSBO-2A</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>30**</td>
<td>267***</td>
<td>235**</td>
<td>307**</td>
<td>215**</td>
</tr>
<tr>
<td>MSBO-2B</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>209**</td>
<td>178**</td>
<td>178**</td>
<td>180**</td>
</tr>
<tr>
<td>SSBO-1A</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>434</td>
<td>413</td>
<td>437</td>
</tr>
<tr>
<td>SSBO-1B</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>426</td>
<td>448</td>
</tr>
<tr>
<td>SSBO-2A</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>412</td>
</tr>
<tr>
<td>SSBO-2B</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>
```

**Bold indicates significant differences.**

**p < .01; ^ heteroscedasticity according to non-parametric Levene test (p < .05)**
extremes of the (approximated) Pareto front (e.g. high average profit, high minimum profit): as the single-objective optimizer is focusing on the optimization of a specific criterion, we may expect it to outperform the multi-objective optimizer in the corresponding region of the front. However, this does not occur. This appears to indicate that the simultaneous consideration of multiple objectives allows for a more comprehensive description (and recognition) of good solutions during the optimization: the solutions identified by the multi-objective approach “stand up” to the scrutiny of 100 simulations during final performance assessment, while the single-objective solutions “drop” in performance.

8.7 Conclusion

As may be expected, our results suggest that a dedicated multi-objective approach is able to consistently generate sets of solutions that present better trade-offs between performance and robustness than a single-objective approach. More surprisingly though, we observe that the single-objective models do not outperform the multi-objective ones in extreme regions of the Pareto front, indicating that the multi-objective approach is able to benefit from a more comprehensive description of solution quality.

We further establish that optimization performance for this problem is significantly affected by the choice of robustness measure. In particular, the use of the sample minimum as the robustness measure negatively impacts on the quality of the final solutions, for both single-objective and multi-objective optimizers. In conjunction with our experiments with sample size, we find that a larger number of evaluation replicates (i.e. more computational effort) is needed to obtain reliable estimates for this measure, compared to less biased statistics such as the sample mean or the sample standard deviation.

While our results for the multi-objective optimizers are generally positive, we do
Figure 8.5: Median (dashed line) and best (continuous line) attainment surfaces of the best multi-objective model and all single-objective models in the objective space of average profit vs. standard deviation of profit in problem instance 1.
Figure 8.5: Median (dashed line) and best (continuous line) attainment surfaces of the best multi-objective model and all single-objective models in the objective space of average profit vs. standard deviation of profit in problem instance 1.
Figure 8.6: Median (dashed line) and best (continuous line) attainment surfaces of the best multi-objective model and all single-objective models in the objective space of average profit vs. standard deviation of profit in problem instance 1.
Figure 8.6: Median (dashed line) and best (continuous line) attainment surfaces of the best multi-objective model and all single-objective models in the objective space of average profit vs. minimum profit in problem instance 1.
Figure 8.7: Median (dashed line) and best (continuous line) attainment surfaces of the best multi-objective model and all single-objective models in the objective space of average profit vs. standard deviation of profit in problem instance 1.
Figure 8.7: Median (dashed line) and best (continuous line) attainment surfaces of the best multi-objective model and all single-objective models in the objective space of average profit vs. standard deviation of profit in problem instance 2.
Figure 8.8: Median (dashed line) and best (continuous line) attainment surfaces of the best multi-objective model and all single-objective models in the objective space of average profit vs. standard deviation of profit in problem instance 1.
Figure 8.8: Median (dashed line) and best (continuous line) attainment surfaces of the best multi-objective model and all single-objective models in the objective space of average profit vs. minimum profit in problem instance 2.
identify a sensitivity of this approach to the number of evaluation replicates employed in estimating fitness values, which highlights the importance of carefully adjusting this parameter in multi-objective formulations for similar problems. In particular, our results show that an increase in sample size (from 10 to 30) leads to a distinct performance improvement for all multi-objective models, while this effect is much less pronounced for the comparable single-objective models. We speculate that this may be due to the simultaneous impact of noise in two objectives.
Appendix 8.A. Results from Problem Instance 2

Figure 8.9: CDFs of hypervolume values obtained with multi-objective models (MSBO-1A, MSBO-2A, MSBO-1B, MSBO-2B) in the objective space of average profit vs. standard deviation of profit in problem instance 2.

Figure 8.10: CDFs of hypervolume values obtained with multi-objective models (MSBO-1A, MSBO-2A, MSBO-1B, MSBO-2B) in the objective space of average profit vs. minimum profit in problem instance 2.
Table 8.5: Values for Mann-Whitney U statistic obtained from pairwise comparisons of hypervolume values of all single and multi-objective models in the objective space of average profit vs. standard deviation of profit in problem instance 2.

<table>
<thead>
<tr>
<th></th>
<th>MSBO-1A</th>
<th>MSBO-1B</th>
<th>MSBO-2A</th>
<th>MSBO-2B</th>
<th>SSBO-1A</th>
<th>SSBO-1B</th>
<th>SSBO-2A</th>
<th>SSBO-2B</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSBO-1A</td>
<td>—</td>
<td>0**</td>
<td>10**</td>
<td>362</td>
<td>101**</td>
<td>91**</td>
<td>240***</td>
<td>440^a</td>
</tr>
<tr>
<td>MSBO-1B</td>
<td>—</td>
<td>—</td>
<td>0**</td>
<td>1**</td>
<td>82**</td>
<td>53**</td>
<td>0**</td>
<td>5**</td>
</tr>
<tr>
<td>MSBO-2A</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>58**</td>
<td>11**</td>
<td>16**</td>
<td>257**</td>
<td>80**</td>
</tr>
<tr>
<td>MSBO-2B</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>138**</td>
<td>114**</td>
<td>290**</td>
<td>397</td>
</tr>
<tr>
<td>SSBO-1A</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>372</td>
<td>82**</td>
<td>220**</td>
</tr>
<tr>
<td>SSBO-1B</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>64**</td>
<td>164**</td>
</tr>
<tr>
<td>SSBO-2A</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>265**</td>
</tr>
<tr>
<td>SSBO-2B</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

* p < .05; ** p < .01; ^a heteroscedasticity according to non-parametric Levene test (p < .05)
Table 8.6: Values for Mann-Whitney U statistic obtained from pairwise comparisons of hypervolume values of all single and multi-objective models in the objective space of average profit vs. minimum profit in problem instance 2.

<table>
<thead>
<tr>
<th></th>
<th>MSBO-1A</th>
<th>MSBO-1B</th>
<th>MSBO-2A</th>
<th>MSBO-2B</th>
<th>SSBO-1A</th>
<th>SSBO-1B</th>
<th>SSBO-2A</th>
<th>SSBO-2B</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSBO-1A</td>
<td>—</td>
<td>264**</td>
<td>62**</td>
<td>313**</td>
<td>127**</td>
<td>238**</td>
<td>135**</td>
<td>173**</td>
</tr>
<tr>
<td>MSBO-1B</td>
<td>—</td>
<td>—</td>
<td>52**</td>
<td>194**</td>
<td>62**</td>
<td>105**</td>
<td>76**</td>
<td>79**</td>
</tr>
<tr>
<td>MSBO-2A</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>164**</td>
<td>419$^a$</td>
<td>241**</td>
<td>427$^a$</td>
<td>361$^a$</td>
</tr>
<tr>
<td>MSBO-2B</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>209**</td>
<td>360</td>
<td>215**</td>
<td>275**</td>
</tr>
<tr>
<td>SSBO-1A</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>295**</td>
<td>437</td>
<td>380</td>
</tr>
<tr>
<td>SSBO-1B</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>300**</td>
<td>363</td>
</tr>
<tr>
<td>SSBO-2A</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>376</td>
</tr>
<tr>
<td>SSBO-2B</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

** $p < .01$; $^a$ heteroscedasticity according to non-parametric Levene test ($p < .05$)

Table 8.6: Values for Mann-Whitney U statistic obtained from pairwise comparisons of hypervolume values of all single and multi-objective models in the objective space of average profit vs. minimum profit in problem instance 2.
Chapter 9

Conclusion

Throughout this doctoral thesis we have investigated SBO as an alternative to conventional optimization approaches to address real-world production planning problems, where uncertainty and complexity are inherent features. We demonstrate that, for the real-world production planning problem analysed in Chapter 5, both of our SBO approaches outperform two exact optimization techniques commonly applied in production planning, namely ILP and CCP. We do not only provide concrete evidence of the poor performance of those two exact techniques, but more importantly we demonstrate that ILP and CCP can be used to significantly improve the optimization performance of our SBO approach. More specifically, we illustrate that the implementation of specialized initialization operators that draw upon the solutions returned by ILP and CCP combined with DES in a variety of ways is able to significantly improve the optimization performance of a standard GA. This is an important finding that not only shows the advantages of SBO approaches, but also highlights the synergies resulting from the combination of population-based meta-heuristics with mathematical programming methods. The use of specialized initialization operators to partially initialize the GA population can be seen as a form of matheuristic (Villegas et al., 2013). Matheuristic approaches are a relatively new area of research that is capturing the attention of a
growing research community (Boschetti et al., 2009) due to its potential.

In Chapter 6, we further demonstrate the potential of matheuristics by implementing an approach where all unfeasible solutions are removed from the search space. In this approach a GA searches for capacity values of failure-prone production lines, rather than directly searching for possible production plans. ILP in this approach acts as a translation tool able to map every solution given by the GA onto a production plan. Our specialized initialization operators from Chapter 5 and the matheuristic optimizer proposed in Chapter 6 offer an opportunity to reduce the computational requirement of EAs working in black-box optimization settings, which according to Coello (2006) is an essential issue that deserves further investigation. We also introduce (Chapter 5) an effective repair operator, which employs simulation to repair solutions that violate capacity constraints, and propose a procedure based on explicit averaging (Jin and Branke, 2005) to support the final solution selection of ESO (Chapters 4 and 5) and EMO algorithms (Chapter 8).

In Chapter 8, we provide evidence for the advantages of a multi-objective formulation over a single-objective formulation within the context of robust optimization. Concretely, we illustrate that, for the real-world problem analysed, single-objective approaches are limited in their ability to steer the search towards optimal trade-off solutions between solution quality and robustness. As may be expected and suggested in previous studies (Jin and Sendhoff, 2003), an explicit consideration of robustness in the form of an additional objective proved to be a better alternative than optimizing for a single objective that implicitly considers robustness (e.g. sample mean, sample standard deviation or sample minimum). However, surprisingly our results also indicate that the use of the sample minimum as a robustness measure has a detrimental impact on the optimization performance of single and multi-objective formulations. This may be because it increases the computational effort required to obtain reliable estimates, compared to less biased statistics such as the sample mean or the sample
standard deviation. This is a controversial finding, since some previous studies (Ong et al., 2006; Tsutsui, 1999) have reported that more robust solutions can be obtained by using the sample minimum as a robustness measure, and because the sample minimum (worst case scenario for maximization problems) is a popular robustness measure used in the search for robust solutions (Gabrel et al., 2014), some examples of this can be found in Chen et al. (2016), Zhang et al. (2016), Goh and Tan (2007), Ong et al. (2006) and Tsutsui (1999). Our experiments also seem to indicate that, in the context of the problem analysed, multi-objective formulations may be more sensitive to noise than single-objective formulations. This may occur because in the multi-objective case the noise affecting the optimization comes from two different noisy fitness estimates, which makes the solution selection more difficult than in the single-objective case, where the solution selection only considers one noisy fitness estimate. For these reasons, the choice of robustness measure and the sample size used during fitness evaluation are essential considerations in designing a successful multi-objective approach for robust optimization. Chapter 8 is focused on the interactions between the choice of optimization paradigm (multi-objective versus single-objective), robustness criterion and sample size in the context of a production planning problem that is subject to machine failures. This problem is an example of a much larger set of problems in which the performance of a solution is subject to uncertainty in the operating or environmental conditions (Lim et al., 2007, 2006; Jin and Branke, 2005).

In this thesis, we have aimed to illustrate the generality of our findings by proposing general formulations of the problems analysed (Chapters 5, 6 and 8), by considering results across a variety of uncertainty levels (Chapters 4, 5, 6 and 8) and by exploring different levels of model complexity to model realistic features of manufacturing systems (Chapter 7). Nevertheless, a remaining limitation of this doctoral thesis is the fact that all of our experiments are performed in the context of a single real-world system. Future research may therefore focus on validating the findings of this work in the
context of different systems and different application areas.

We provide evidence that exact optimization techniques are limited in their ability to accurately capture the complexity and uncertainty intrinsic to the real-world system analysed (Chu et al., 2015; Güller et al., 2015). This is mainly because the level of mathematical sophistication needed to model those complex features is no longer within the framework of exact optimization techniques (Figueira and Almada-Lobo, 2014; Sel and Bilgen, 2014). Consequently, exact optimization techniques may need to rely on overly stringent assumptions to cope with that complexity and keep the problem computationally tractable (Gosavi, 2014, p. 31). However, the use of overly stringent assumptions compromises the real applicability of solutions obtained (Nikolopoulou and Ierapetritou, 2012; Gnoni et al., 2003), basically because those are the right solutions, but to the wrong problems. Therefore, we believe that the SBO approaches explored throughout this work may contribute to a wide range of real-world applications, as we present some general ideas related to the incorporation of uncertainty and complex features into the problem formulation as well as different solution procedures that could be tailored to areas where mathematical programming methods are being used to address problems where uncertainty is an inherent feature. Furthermore, the simultaneous consideration of tactical and operational decisions, e.g. having the ability to re-optimize the problem after deviations have occurred, could further extend the applicability of our approaches to other areas. Stochastic programming with recourse is an alternative worth exploring in this respect.

This thesis also provides general insights about the combined use of meta-heuristics and exact optimization techniques. This combination of techniques is of particular interest because of its potential to speed up convergence, which is utmost important when function evaluations are expensive in terms of computational effort. Our experiments suggest that when the problem formulation is beyond the framework of exact optimization techniques, mathematical programming could be used to create initial solutions by
solving a simplified formulation of the real problem, and then further improve those initial solutions by applying meta-heuristic approaches that operate upon a more accurate problem formulation. Given that evolutionary approaches are a common choice of meta-heuristic, the development of specialized crossover and mutation operators able to preserve the feasibility of solutions is an interesting avenue to improve the search mechanism of those approaches. The design of feasibility-preserving operators would be extremely useful to deal with constrained optimization problems, as turning good solutions into unfeasible ones is one of the drawbacks of applying crossover or mutation. Moreover, the elimination of unfeasible solutions from the search space by having a meta-heuristic finding solutions in the genotypic search space and then applying mathematical programming techniques to map a given genotype onto its phenotype is also a promising alternative to deal with constrained optimization settings, especially when the problem formulation addressed by the exact optimization method is a good approximation of the real problem. Finally, having an algorithm with the ability to adapt its parameter values during the optimization is extremely important for the case of meta-heuristics, where appropriate parameter tuning has a substantial impact on algorithmic performance. Therefore, on-line parameter tuning is another interesting topic that could be further explored towards the development of more effective meta-heuristics, as this self-adaptation ability may cause substantial improvements in the search mechanism and performance of the algorithm. For instance, it would be advantageous to have a GA able to alter its mutation probability as well as the magnitude of those mutations along the path towards the optimum, as at the beginning of the optimization big jumps might be beneficial (e.g. preventing getting trapped in local optima), whereas at latter stages such big jumps might preclude finding the (global) optimum.
Bibliography


M. Arakawa, M. Fuyuki, and I. Inoue. An optimization-oriented method for


H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947.


Vehicle Routing and Distribution Logistics.


