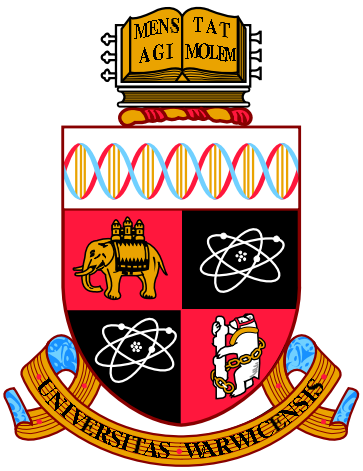


Meta-Heuristic & Hyper-Heuristic Scheduling Tools for Biopharmaceutical Production

by

Folarin Bolude Oyebolu

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Warwick Business School

July 2019

THE UNIVERSITY OF WARWICK

Contents

List of Tables

List of Figures

Acknowledgments

Writing these acknowledgements, I am reminded of the proverb: “It takes a village to raise a child”. While the thought of an entire village writing one thesis amused me, I realised how many people have given me support and shown me kindness along my PhD journey. I am eternally grateful to all of them — my ‘village people’.

I am indebted to my supervisor, Prof Jürgen Branke. His consistent guidance, patience, encouragement and enthusiasm have proven invaluable to my progress. I would also like to thank Prof Suzanne Farid for her advice and insight, and for initially suggesting this project to me. I am also thankful for the discussions with Dr Richard Allmendinger — his suggestions are helpful in improving my work.

I wish to acknowledge financial support from the EPSRC Centre for Doctoral Training in Emergent Macromolecular Therapies, hosted by University College London. I appreciate the help and support of the staff there and at WBS.

I would like to thank my friends and colleagues at WBS, UCL, and everywhere else for their camaraderie, kindness, and support. I especially thank Ivano Colao and Roberto Chiocchio for their helpful comments on this thesis.

Finally, my sincere gratitude goes to my family for their seemingly infinite love and support — particularly my parents, Bolade and Olusola, and my sister Motunrola. I appreciate all they have done for me, including proof-reading my work (to verify that the money spent on my education in my youth was not wasted). This thesis was typeset with LaTeX 2ε [1] by the author.

[1] LaTeX 2ε is an extension of LaTeX. LaTeX is a collection of macros for TeX. TeX is a trademark of the American Mathematical Society. The style package *warwickthesis* was used.

Declarations

I, *Folarin Bolude Oyebolu*, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

The manufacturing of biopharmaceuticals requires substantial investments and necessitates long-term planning. Complicating the task of determining optimal production plans are large portfolios of products and facilities which limit the tractability of exact solution methods, and uncertainties & stochastic events which often render plans obsolete when reality deviates from the expectation. This thesis therefore describes decisional tools that are able to cope with these complexities.

First, a capacity planning problem for a network of facilities and multiple products was tackled. Inspired by meta-heuristic approaches to job shop scheduling, a tailored construction heuristic that builds a production plan based on a sequence — optimised by a genetic algorithm — of product demands was proposed. Comparisons to a mathematical programming model demonstrated its competitiveness on certain scenarios and its applicability to a multi-objective problem.

Next, a custom object-oriented model was introduced for a manufacturing scheduling system that utilised a failure-prone perfusion-based bioprocess. With this, process design decisions such as cell culture run time and process configuration, and single-product facility scheduling strategies were evaluated whilst incorporating simulations of stochastic failure events and uncertain demand.

This model was then incorporated into a larger hyper-heuristic to determine optimal scheduling policies for a multi-product problem. Various policy representations are tested and a few policies are adapted from the literature to fit this specific problem. In addition, a novel policy utilising a look-ahead heuristic is proposed. The benefit of parameter tuning using evolutionary algorithms is demonstrated and shows that tuned policies perform much better than a policy that estimates parameters based on service level considerations. In addition, the disadvantages of relying on a fixed or rigid production sequence policy in the face of uncertainty is highlighted.

Abbreviations

ACO	Ant Colony Optimisation
ANN	Artificial Neural Network
ATF	alternating tangential flow
BLA	biologic licence application
CIP	clean-in-place
CMA-ES	Covariance Matrix Adaptation Evolution Strategy
CMO	contract manufacturing organisation
CLSP	capacitated lot-sizing problem
CoG	Cost of Goods
CSL	customer service level
DLSP	discrete lot sizing and scheduling problem
DSP	downstream processing
EA	evolutionary algorithm
ELSP	economic lot scheduling problem
EOQ	economic order quantity
EPQ	economic production quantity

FCI fixed capital investment

GA genetic algorithm

GLSP general lot sizing and scheduling problem

GP genetic programming

IND investigational new drug

JSP job-shop scheduling problem

mAb monoclonal antibody

MILP mixed-integer linear programming

MPF multi-product facility

NDA new drug application

NPV net present value

NSGA-II non-dominated sorting genetic algorithm II

PLSP proportional lot sizing and scheduling problem

PPX Precedence Preserving Crossover

PSO particle swarm optimisation

RMU relative monetary units

RTN Resource Task Network

SCLSP stochastic capacitated lot sizing problem

SELSP stochastic economic lot scheduling problem

SIP sterilization-in-place

SLSP stochastic lot scheduling problem

SMDP Semi-Markov Decision Process

SUB single use bioreactor

SUS stochastic universal sampling

TS tabu search

USP upstream processing

Chapter 1

Introduction

1.1 Motivation

The production of biopharmaceuticals is an expensive and time-consuming endeavour requiring large investments and long-term production planning. The average cost to bring a new biopharmaceutical to market is estimated at $1.2-1.8 billion given the high attrition rates (DiMasi and Grabowski, 2007; Paul et al., 2010), and building large multiproduct manufacturing facilities can take 4-5 years to complete and costs $40-650 million (Farid, 2007) in addition to the decade-long timeline required for drug development.

Due to the high cost and long timeframes, biopharmaceutical companies have to plan production well in advance, often utilising demand forecasts to guide their decision-making. However, biopharmaceutical production has a number of characteristics that make it challenging to optimise including: interdependent decisions; multiple conflicting objectives; heterogeneity in product and manufacturing facility portfolios; options for third-party partnerships or outsourcing; batch and semi-continuous manufacturing processes; uncertainties in the manufacturing process or external factors; and product lability (Lim et al., 2006; Lakhdar et al., 2007; George and Farid, 2008).

Literature in the pharmaceutical or biopharmaceutical industry on planning and production scheduling is small but growing (Vieira et al., 2015). However, the models used either are restricted to deterministic problems, only model one mode of processing, or do not focus on scheduling. In addition, there is a dependence, in much of the existing literature, on exact solution methods which whilst useful and powerful, become intractable with increasing problem size. They also may be

restricted in their ability to model real-life systems due to simplifying assumptions made in order to formulate the problem(s).

As a result, there exists a gap to use heuristic, meta-heuristic, and/or hyper-heuristic techniques to: (i) provide alternative solution approaches to previous problems by providing improvements in speed or objective value performance, or a more accurate model of reality; (ii) pose questions and investigate aspects of biopharmaceutical production planning that have so far not been within the scope of the current literature; and (iii) deal with the stochastic and uncertain nature of planning and manufacturing systems.

1.2 Research Questions

The challenges and opportunities described above may be mitigated by process design & scale-up, planning, strategy, and decision-making which is efficient and effective (in terms of cost, time and effort). Within this context, the questions posed by this thesis and which it will attempt to answer include:

- What is a suitable meta-heuristic alternative to exact solution methods to capacity planning and scheduling for a network of biopharmaceutical manufacturing facilities?
  - How does the performance of this meta-heuristic compare empirically with existing approaches?
  - What is the scope for providing a decision-maker with 'equally good' solutions with different trade-offs regarding multiple conflicting objectives?
- What is the effect on scheduling of uncertainty due to fluctuating demand and randomly failing processes?
  - What process design decisions (i.e., process run time and process configuration) deliver optimum performance in the face of random process failure?
  - Given process configuration and uncertain demand, what is a suitable strategy to simultaneously determine optimum run times and schedule manufacturing batches for a single product?

- Additionally, given instead, a system of multiple products with uncertain demand and process failure:
  - What are good representations and parameters for scheduling policies to dynamically allocate facility capacity optimally between products?
  - What is the benefit of implementing flexible process run times?

To answer these questions, decision tools based on heuristic, meta-heuristic, and hyper-heuristic techniques are developed and problem instances are formulated to test and validate them. These tools can support biopharmaceutical companies in planning and scheduling decisions.

1.3 Structure of Thesis

This thesis is composed of six more chapters plus some additional appendices and is structured as described below. The earlier chapters aim to provide necessary background to the research questions whilst the latter chapters develop the tools required and tackle these questions head-on.

Chapter 2 on page 5 provides an overview of the biopharmaceutical industry, detailing the effort required in drug development and describing a typical manufacturing process. The aim of this chapter is to demonstrate the opportunities but also the challenges facing the companies operating in the industry.

Chapter 3 on page 14 delves into a detailed survey on literature related to production planning research. This focuses on lot sizing and scheduling models with purely deterministic or part stochastic components. There are also brief introductions to the algorithms utilised in this thesis. Additionally, it highlights the gaps in the planning literature for the biopharmaceutical industry.

In Chapter 4 on page 28, a meta-heuristic is proposed to solve the complex combinatorial optimisation problem of deterministic scheduling and planning of multi-site biopharmaceutical manufacturing. This novel proposal is subsequently compared with an exact solution method from the existing literature in an attempt to highlight its usefulness in certain scenarios.

A novel hyper-heuristic framework is proposed and developed in Chapter 5 on page 62 which incorporates a custom object-oriented model simulating scheduling decisions and operational events in semi-continuous perfusion processes. This

framework is then evaluated in terms of determining optimal process design and scheduling decisions in the face of stochastic and uncertain events.

Further to that, Chapter 6 on page 98 uses this framework and tests a range of scheduling policies on a synthetic case study of a single multi-product facility. In addition to demonstrating the benefit and robustness of policies compared to more basic rules, the usefulness of flexible process run times is explored.

Chapter 7 on page 139 wraps up this thesis by revisiting and summarising the contributions of the thesis while giving an outlook on possible future work.

Chapter 2

Background on the Biopharmaceutical Industry

The biopharmaceutical industry is concerned with the development, manufacturing, and marketing & commercialisation of biopharmaceutical drugs[1]. These biopharmaceuticals are complex biological molecules which are marketed and commercialised for therapeutic or diagnostic uses. Worldwide sales of biopharmaceuticals totalled $140 billion in 2013 (La Merie Business Intelligence, 2013), growing to $228 billion in 2016 (Troein, 2017), and represent the fastest growing sector of the overall pharmaceutical market. Biopharmaceuticals are usually distinguished from other pharmaceuticals or small molecule drugs which, unlike biopharmaceuticals, can be synthesised chemically and can be described atom-by-atom (Rader, 2005). Biopharmaceuticals require biotechnology methods for manufacture, often in a host cell organism which often has been genetically engineered to produce the biopharmaceutical.

A non-exhaustive list of the types of biopharmaceuticals includes vaccines, gene therapy products, recombinant proteins, cultured cells and tissues, human blood products, enzymes, and monoclonal antibodies (mAbs). Table 2.1 on the following page lists pharmaceuticals (including small molecule drugs) with the highest worldwide revenue in 2017, highlighting the clout of biopharmaceuticals in the broader market, especially mAbs. A mAb refers to therapeutic antibodies which are made from a cell line of identical clones sourced from a single unique parent. Antibodies — which are large protein molecules — have very strong affinity and

[1] These may also be referred to as *biologics*.

Table 2.1: The top fifteen selling pharmaceuticals of 2017 worldwide, their manufacturers, therapeutic indication, and classification. Sourced from www.pharmacompass.com.

Product Name	Active Ingredient	Type	Manufacturer(s)	Main Therapeutic Indication	Revenue (Millions USD)
Humira®	Adalimumab	mAb	AbbVie Inc., Eisai	Immunology	18,946
Enbrel®	Etanercept	fusion protein	Amgen, Pfizer Inc., Takeda	Immunology	8,262
Eylea	Aflibercept	fusion protein	Regeneron, Bayer	Ophthalmology	8,260
Revlimid	Lenalidomide	small molecule	Celgene	Oncology	8,187
MabThera®/ Rituxan®	Rituximab	mAb	Roche	Oncology	7,831
Remicade®	Infliximab	mAb	Johnson & Johnson, Merck, Mitsubishi Tanabe	Autoimmune Disorders	7,784
Herceptin®	Trastuzumab	mAb	Roche	Oncology	7,435
Eliquis®	Apixaban	small molecule	Bristol-Myers Squibb, Pfizer Inc.	Cardiovascular Diseases	7,395
Avastin®	Bevacizumab	mAb	Roche	Oncology	7,089
Xarelto™	Rivaroxaban	small molecule	Bayer, Johnson & Johnson	Cardiovascular Diseases	6,590
Opdivo	Nivolumab	mAb	Bristol Myers Squibb, Ono Pharmaceutical	Oncology	5,815
Lantus	Insulin Glargine	protein	Sanofi	Diabetes	5,731
Prevnar 13/ Prevenar 13	Pneumococcal 7-Valent Conjugate	vaccine	Pfizer Inc.	Anti-bacterial	5,601
Lyrica	Pregabalin	small molecule	Pfizer Inc., Eisai	Neurological Disorders	5,318
Neulasta®	Pegfilgrastim	PEG-protein	Amgen, Kyowa Hakko Kirin	Blood Disorders	4,553

specificity to targets (antigens) unique to each antibody clone. These antigens can be pathogenic bacteria, viruses or even cancer cells.

2.1 Drug and Process Development

Biopharmaceutical drug development refers to the process of taking a molecule from first discovery of the potential therapeutic action to a commercial product on the market available for use by patients and the healthcare providers administering them to the patients (see Figure 2.1 on the next page). This drug development is extremely lengthy (Werner, 2004), expensive (DiMasi and Grabowski, 2007), complex, and risky (DiMasi et al., 2010). The estimated total capitalised cost for each new biopharmaceutical molecule to achieve market approval can exceed $1.2 billion (DiMasi and Grabowski, 2007).

Part of this R&D effort takes place on two parallel but interacting and interdependent tracks. On one track is clinical development which involves a series of clinical trials and essentially aims to ascertain if the therapeutic is safe and if it works. The other track is process development in which a manufacturing process for producing the therapeutic for testing in clinical trials *and* for commercial production is designed and undergoes scale-up. Decisions made in clinical development affect process development and vice versa. This is because the molecule that goes through clinical testing comes from the manufacturing process that is concurrently undergoing development. Due to the complexity of biologics and biological systems, the molecule's identity is often linked to (and in part determined by) its manufacturing process. As a result, process design decisions have to be consistent through development to ensure that it is the same molecule going through all stages of clinical development. Similarly, decisions made in clinical development on dosage and method of delivery mean that the manufacturing process will need to be productive enough and produce the molecule in the correct formulation to match the clinical efforts. In addition, in the early stages of development, a company will often need to commit capital towards a facility to house the manufacturing process without the guarantee that the drug candidate will successfully clear clinical trials.

In the stages of developing a biopharmaceutical, the initial drug discovery comes before clinical or process development. This drug discovery is composed of four stages: target identification, target validation, lead identification, and lead optimisation. This process involves identifying potential targets for therapeutic ac-

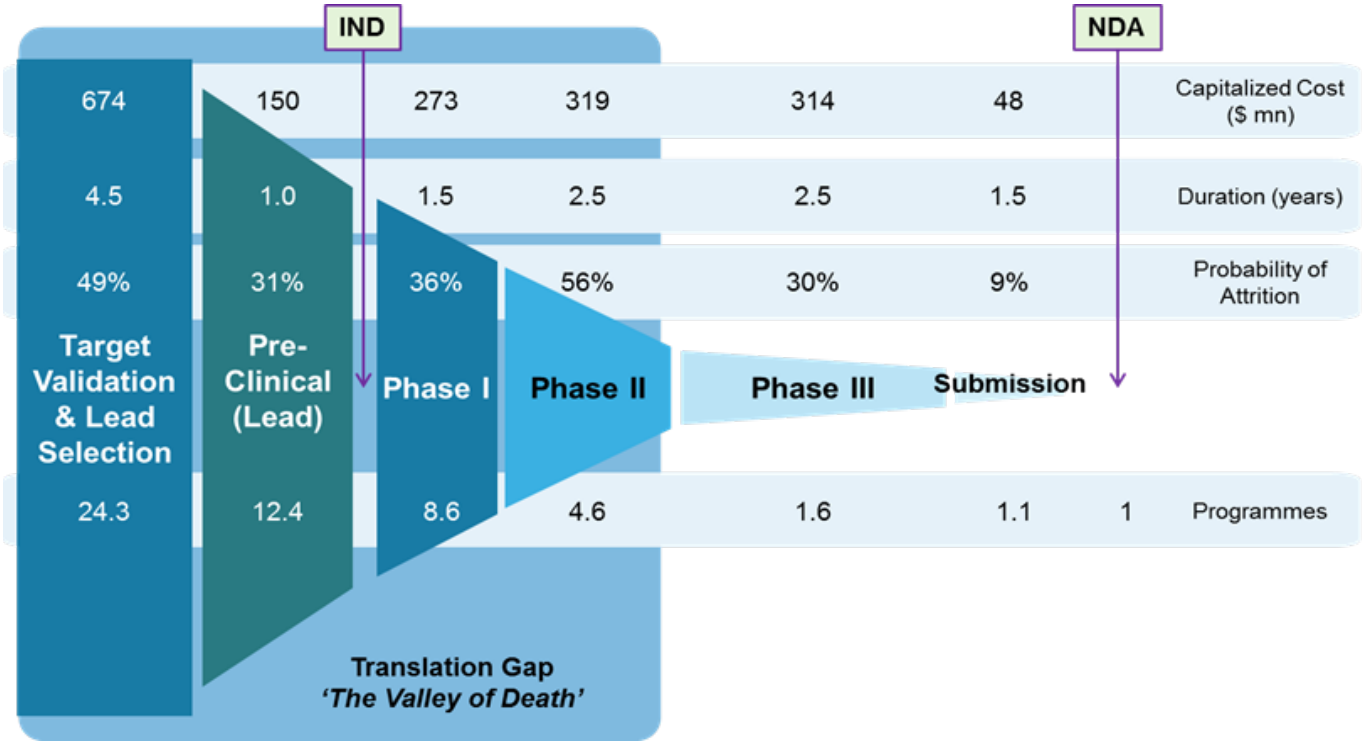


Figure 2.1: Drug development pathway showing the duration of each stage, capitalised costs, and probability of failure or attrition rate. 'The Valley of Death' references the fact that the majority of development projects fail in those stages (sourced from www.lonza.com).

tion that are involved in the model of the disease in question and selecting the most promising of those targets. Then these targets are used to screen a large number of molecules to test and identify the molecules (leads) that have specificity for the target and cause the desired changes to the target. The leads are subsequently optimised to improve activity. Selection of the optimal lead or drug candidate moves the process into clinical development. The potential difficulty or feasibility of manufacturing is often a factor in drug candidate selection.

Clinical development is made up of several different stages. Before any in-human trials, pre-clinical testing is carried out to assess if the drug candidate is safe to administer to humans. To commence human clinical trials in the US, an investigational new drug (IND) application has to be made to and successfully granted by the regulatory body — the FDA. Usually, between this point and drug candidate selection, a patent would have been applied for and granted. This patent protection gives market exclusivity to the patent holder for the molecule for 20 years from the patent application date.

There are three major stages of human clinical trials for drug candidates to go through. Phase I focuses on the safety of the drug and will involve around ten healthy subjects. Phase II is the major stumbling block for most drugs as it has the highest attrition rate (i.e., probability of transition to the next phase is lowest in Phase II). This is where the efficacy of the drug is determined and tested as well as further safety studies. Phase III requires up to thousands of volunteer subjects to determine the range and severity of side effects on a larger sample size, final dose sizes, as well as further safety and efficacy testing. Results and findings from these studies are filed and submitted to request regulatory approval in the form of a new drug application (NDA), which when granted allows the drug to be marketed and sold.

Process development aims to take the techniques from synthesizing the molecule at a lab-scale to an economical manufacturing process capable of producing the therapeutic at commercial scale. The process evolves in stages at intermediate scales (e.g., pilot-plant scale), progressively increasing and optimising equipment sizes and productivities. As previously mentioned, clinical manufacturing takes place concurrently to provide the drug that is used during clinical trials. Ultimately, process development ensures that the manufacturing process is validated, safe, consistent, and robust whilst also trying to be as economical and cost-effective as possible. For biologics, in addition to an NDA, a biologic licence application (BLA)

is required for approval and this BLA is needed for each different manufacturing facility it is produced in.

This drug development process may take up to 12 years. So that means that a patent granted at the start of the development process therefore only leaves eight years of market exclusivity (in this case). Work done on optimising the structure and development pathway of biopharmaceutical drug portfolios includes the study by George and Farid (2008).

2.2 Biopharmaceutical Manufacturing

Biopharmaceutical manufacturing is a substantial endeavour as it involves growing single-cell organisms on a large scale. As experience and expertise has built up in this area, a lot of biopharmaceuticals especially mAbs have started to adopt more standardised processes (Fahrner et al., 2001; Shukla et al., 2007).

The bioprocess for manufacturing mAbs is normally a batch process and comprises of many steps (unit operations) but can be broadly divided into two main parts: upstream processing (USP) and downstream processing (DSP). USP is the part of the manufacturing process where the cells are grown and the biopharmaceutical is actually made (by the cells). DSP separates the biopharmaceutical molecule from its host cell and removes any impurities from the product stream(s). Bulk manufacturing ends after DSP, and after that the purified bulk product undergoes final fill, formulation, and packaging (usually in a secondary facility or location) to put the molecule in the delivery format that is administered to patients.

USP starts from the expansion and thaw of frozen seed cells in cell banks and goes on to make the actual drug molecule using cells grown in one or more bioreactors. The selection of expression system depends on a number of factors (Verma et al., 1998), and the available choices range from bacterial (e.g., *E. coli*), yeast, insect, to mammalian cells (e.g., Chinese hamster ovary (CHO)). Generally, mAbs are produced in mammalian cells (Farid, 2006) because other cells, like *E. coli*, cannot secrete antibodies — this makes the purification process more cumbersome as the cells have to be broken up to release the antibodies. Also, they may not be able to carry out post-translational modifications such as *glycosylation* — this glycosylation is often necessary for antibody specificity and effector functions, and also improves half-life.

There are two main types of processing modes in USP: fed-batch and perfusion

(Chu and Robinson, 2001; Birch, 2003). In fed-batch processes, the bioreactor is seeded and the cells are grown for a certain period of time (with a few intermittent feed additions) at which point the entire contents are harvested and passed on to DSP for purification. Perfusion processes, on the other hand, are semi-continuous. The bioreactor is seeded as normal and cells are grown for a specified amount of time (with regular feed additions) until they reach a steady-state cell density — this period of time is called *ramp-up*. From this point onwards, regular harvests are made from the bioreactor (which is replenished by equally regular feed additions); these harvests can be purified separately, or pooled and/or frozen for later downstream processing. As a result, the perfusion processes last significantly longer than the fed-batch processes — a CHO fed-batch cell culture is typically 14 days long but a perfusion cell culture can be as long as 60 days. The fed-batch process is more prevalent in industry, though the industry is showing new interest in perfusion processes. There are advantages and disadvantages to both processing modes (Chu and Robinson, 2001; Birch, 2003; Bibila and Robinson, 1995): fed-batch is easier to control but is less productive, and perfusion offers a smaller facility footprint but is more susceptible to process contamination and failure.

The unit operations involved in DSP for a platform mAb process generally include: clarification steps to separate the whole cells from the product stream; multiple chromatography steps; and orthogonal viral clearance steps (FDA, 1998; Sofer and Lister, 2003) (depending on the cells used). Although most mAb downstream processing trains tend to consist of the same unit processes, each mAb introduces unique differences which means process engineers must tune unit operations and the DSP as a whole to each mAb molecule (Marichal-Gallardo and Alvarez, 2012). The chromatography unit operations, particularly the affinity chromatography step which uses Protein A resin for mAbs, have recently been identified as the bottleneck (in time and costs) in the overall process (Kelley, 2007) as cell culture titres have increased. Work done by Allmendinger et al. (2012, 2013) has sought to deal with this by optimising cost-effective chromatography column-sizing strategies for mAb manufacture while also incorporating user preferences.

In addition to decisions on equipment sizes and selection of unit operations, it is possible to choose process configurations where there are multiple staggered bioreactors to one DSP train. Rouf et al. (2000) investigated the economy of scaling-up a bioreactor by using multiple units versus one larger bioreactor. This found that the flowsheet with multiple reactors had a higher return on investment resulting from

the smaller size of DSP units since DSP accounts for such a large proportion of processing costs. Multiple reactors however will incur higher equipment purchase costs even as their DSP is smaller and more utilised.

Manufacturing can take place in a dedicated facility or suite (i.e., a single-product facility) or in a facility where equipment and resources are shared between different manufacturing processes (i.e., a multi-product facility (MPF)). Operating an MPF brings up a few more concerns in addition to those experienced in a single-product facility. First of all, there needs to be an adequate cleaning and robust validation procedure to ensure there is no product carryover from one process to another. Secondly, as well as the potential to cause cross-contamination and product carryover, the changeovers between different products can be labour and time intensive (with the extra cleaning and validation). So it may be operationally beneficial to manufacture in campaigns[2], and minimise product changeovers.

2.3 Emerging Trends and Challenges

The unique qualities of biopharmaceuticals and its market bring many varying challenges to biopharmaceutical companies (Närhi and Nordström, 2005). As briefly implied above, these companies face great uncertainty and risk. They have to manage uncertain development times and costs, uncertain phase transition probabilities, and uncertain market forecasts — ideally in a cost-effective and efficient manner.

Biosimilars are a growing trend in the biopharmaceutical industry and pose an increasing competitive threat to manufacturers of originator biopharmaceuticals. Biosimilars refer to close copies of branded biologics marketed after the expiration of the patent and regulatory protection period of the established branded biologic (Weise et al., 2011). The sales erosion in the first year after generic entry for small molecule drugs can cause the originator's unit share to fall to 11% (Grabowski et al., 2013). It is unlikely that such drastic sales erosion will occur in the biopharmaceutical market. However, new biosimilars are set to reap the benefits and success of the originator when put in context with the highly profitable, growing market and potential selling prices 20-30% lower than the originator (Walsh, 2010).

Biopharmaceutical companies are also faced with manufacturing bottlenecks;

[2]Here a campaign means a long sequence of batches of one product which is uninterrupted by batches of other products. It has a second similar meaning: after changeover from product b, starting the first batch of another product, a, can be referred to as starting a campaign of product a (regardless of how many more batches of a are planned). Both meanings are used interchangeably within context throughout this thesis.

strict risk-averse regulation (Eichler et al., 2013); and long, difficult and complex manufacturing processes (Ransohoff, 2004; McGurk, 2004). There have been recent efforts to improve the productivity and utilisation of bottlenecked chromatography processes by introducing semi-continuous chromatography (Mahajan et al., 2012; Warikoo et al., 2012). Pollock et al. (2013a) evaluate the potential of semi-continuous chromatography systems for clinical and commercial manufacture.

The pursuit of efficiency and cost-effectiveness is coupled with the increasing need for flexibility (Kelley, 2009) as companies compete for limited resources whilst bringing their products to market. Legacy facilities were designed and built to manufacture large amounts of just one blockbuster drug each (i.e., single-product facilities), and it has been forecast to no longer be the norm (Carson, 2005). Companies designing new facilities will need multi-product facilities that are more flexible — i.e., facilities that are capable of manufacturing several mAbs with diverse characteristics and process variations. This is to account for several different leads going through the development process and the inherent uncertainty in which specific mAb candidates will be approved. A main problem facing MPFs is the risk of cross-contamination, but this can be mitigated by the correct facility design (GEN, 2006). MPFs require more rigorous cleaning procedures and extra validation costs between each product campaign but these extra procedures and costs have been found to still be economically feasible (Sofer and Nyström, 1991). The use of disposable systems further extends the flexibility of an MPF as it reduces the costs and time related with cleaning validation and product changeover respectively. In addition, they reduce some running costs by minimising water and reagents used by clean-in-place (CIP) and sterilization-in-place (SIP) operations. However they incur higher consumables and waste disposal costs. Disposable components that can be used include but are not limited to single-use bioreactors (up to 2000L scale), filters, filter housings, pipes, pre-packed columns or membrane adsorbers; disposable technology finds many applications in both MPFs and single-product facilities (Allison and Richards, 2013; Shukla and Gottschalk, 2013).

Finally, decisions faced by biopharmaceutical companies are interdependent, i.e., the decision about which drugs to develop depends on the available capacities and resource needs of the other drugs, and vice versa. At the same time, process development determines the technical characteristics of the manufacturing process producing drugs either for clinical development or the commercial market.

Chapter 3

Literature Review

3.1 Introduction

Production planning aims to make best use of production resources in order to satisfy production goals or demand over a planning horizon. It is omnipresent in any manufacturing environment including bioprocessing and the manufacture of biologics. Determining a good or an optimal production plan is not a new problem and various approaches exist that tackle it wholly or focus on specific aspects in order to achieve the stated objectives.

This chapter aims to lay out the current state-of-the art in production planning research especially in the realm of lot sizing and scheduling and in a pharmaceutical or bioprocessing context. This interfaces between deterministic and stochastic lot-sizing, scheduling, capacity planning, bioprocess optimisation, and product cycling problems and solution techniques such as meta- and hyper-heuristics, simulation-optimisation, and exact methods.

Literature on lot sizing and scheduling, will be discussed first in Section 3.2 on the following page. This will include descriptions of its deterministic and stochastic types as well as the distinctions between their many variants. Secondly, Section 3.3 on page 19 will describe and compare several solution methods and algorithms which have been used in prior work. Following that, relevant research tackled in a bioprocessing or biopharmaceutical context will be outlined in Section 3.4 on page 26, and the conclusion of this chapter will summarise the present research gaps this thesis attempts to fill.

3.2 Lot Sizing and Scheduling

Lot sizing models determine the optimal timing and level of production. The simplest of these was formulated by Wagner and Whitin (1958) whereby production levels for a product on a single machine (or facility) over a finite planning horizon are determined. Lot sizing models mostly focus on the trade-off between set-up cost and inventory cost.

Scheduling was defined by Graves (1981) as *"...the allocation of available production resources over time to best satisfy some set of criteria"*. Framinan et al. (2014) provided an extensive overview on scheduling systems models, and algorithms for exact and approximate solutions. Within the context of lot sizing, scheduling has to do with the precise sequencing and timing of jobs within time periods. Usually the lot sizing is solved first before the scheduling problem which is based on the results from the lot sizing model.

Different extensions have been developed and investigated for lot sizing in general (Jans and Degraeve, 2008), and also with scheduling considered as part of the problem (Drexl and Kimms, 1997). One of the main distinctions between the variants (when using mathematical programming) is whether they are classed as a *'small-bucket' formulation* or a *'big-bucket' formulation*. This refers to the size of the **time periods** and how sequencing and scheduling is handled within them. The former describes models or formulations which can only have at most one product per machine produced in each time period which effectively determines a sequence. On the other hand, 'big-bucket formulations' may have more than one product produced on a single machine per time period. As a result, these models usually do not deal with sequencing or explicit scheduling.
Other characteristics that may differentiate model variants are as follows:

- The **planning horizon** which determines the length of the production schedule. This can either be *finite* or *infinite*.
- A *single-level* system is one where end products or items are produced directly from raw materials. Conversely, a *multi-level* system involves parent-component relationships among the products. That is, one product serves as an input or raw material for the processing or production of another, so demand for one operation (level) is dependent on a prior one.
- The **number of products** has an important effect on complexity, with *single-product* problems significantly simpler than *multi-product* cases.

- **Capacity or resource constraints** can be put on the production operations leading to *capacitated* problems. If these constraints are not present, the problem is said to be *uncapacitated*.

- **Demand** is classed as *static* if it remains constant (or stationary) over the entire planning horizon and *dynamic* if it changes with time. In addition, it may be uncertain or *stochastic* if not known ahead of time — usually based on probabilities and probability functions. Otherwise if it is known exactly, it is *deterministic*. Related to the number of levels, *independent* demand has products where their demands do not depend on the decisions of other products but *dependent* demand has products with demands depending on demands of other products.

- Implementation of **setup** can take many forms. Usually, changeover (switching) from one product to another incurs a *setup cost* and/or *setup time*. If the setup applied at the start of production of an item differs based on the preceding item, this is classed as *sequence-dependent setup*. In addition (and specific to mathematical programming), a problem has *setup-carryover* if production of an item over two periods only required one setup before production in the first period.

- Finally, the flow of **inventory** may have several constraints. There may be *maximum inventory levels* (maximum storage capacity) imposed per product, per facility (or machine), globally, or combinations thereof. In addition, inventory may be *perishable*. This can either be by defining a maximum holding time (or shelf-life) or a spoilage rate (i.e., a fraction of inventory spoils each period and needs to be discarded). Also, inventory shortage may be allowed in two ways[1]. *Backlogging* means that the demand of the current period can be delivered in future periods and *lost sales* means that not meeting demand at all is allowed.

The rest of the discussion in this chapter will distinguish primarily between deterministic and stochastic lot sizing and scheduling problems. That is, of the characteristics just mentioned, the type of demand is what shall be focused on.

[1] It is possible for both forms of inventory shortage to be used in the same system simultaneously.

3.2.1 Deterministic Models

As has previously been mentioned, there are several extensions for the lot sizing and scheduling problem which generally evolved from the economic order quantity (EOQ) problem — a single-product single-item problem with deterministic stationary demand, infinite planning period, and no capacity constraints (Erlenkotter, 1990).

Some of the deterministic variants include: the economic lot scheduling problem (ELSP) which has an infinite planning horizon, where products have stationary demand, and time is continuous (instead of discrete) (Gallego and Shaw, 1997); the capacitated lot-sizing problem (CLSP), which considers capacity constraints for the machines and as a 'large-bucket' model, allows several items to be produced per period (Eppen and Martin, 1987); the discrete lot sizing and scheduling problem (DLSP), where periods are very short, only one product can be made per period, and if so, the production uses the full capacity (Lasdon and Terjung, 1971); the proportional lot sizing and scheduling problem (PLSP), in which not more than two products can be made in a time period (Drexl and Haase, 1995; Drexl and Kimms, 1997); and the general lot sizing and scheduling problem (GLSP) which attempts to take take a 'large-bucket' model and simultaneously do scheduling by assigning each lot in a period a unique number to determine a sequence (Fleischmann and Meyr, 1997).

In addition, Potts and Wassenhove (1992) highlighted the close relationship between scheduling, and lot sizing and scheduling. This idea of a relationship between lot sizing problems and scheduling models in general was motivation for two efforts to solve the DLSP as a batching and scheduling problem. Here, each demand is interpreted as a job and characterised by its size and deadline and must be processed in one piece without splitting: one by Jordan (1996), and another by Jordan and Drexl (1998).

Of these deterministic models, CLSPs have received a large share of research attention and been the subject of several extensive reviews and surveys (Drexl and Kimms, 1997; Karimi et al., 2003; Jans and Degraeve, 2008).

3.2.2 Stochastic Models

Historically, most research has been on problems that assume deterministic demand and no randomness or uncertainty in general. However, real-life systems often are not as simple and suffer from uncertainty either in demand, production rates or

setup times.

This different class of the problem is termed the stochastic lot scheduling problem (SLSP). In their review of the SLSP, Sox et al. (1999) made a distinction between the stochastic economic lot scheduling problem (SELSP) and the stochastic capacitated lot sizing problem (SCLSP) to be consistent with their deterministic counterparts. The former assumes continuous time, an infinite horizon, and stationary demand while the latter assumes a finite planning horizon, discrete time periods, and may have non-stationary demand. However, Winands et al. (2011) blurred this by defining the SELSP as allowing finite planning horizons but restricting it to stationary demand.

In addition to these surveys, Aloulou et al. (2014) compiled an extensive bibliography of publications on the non-deterministic lot-sizing problem and classified them according to the number of products, time-periods, machines, the uncertain parameters, and the modelling approaches. Li and Ierapetritou (2008) reviewed the main methodologies that have been developed to address the problem of uncertainty in production scheduling as well as to identify the main challenges in this area. Ouelhadj and Petrovic (2009) surveyed dynamic scheduling in manufacturing systems, covering the limitations of static schedules and approaches in dynamic scheduling.

An extension looks at production processes which are prone to random machine/equipment failure. In the case of equipment failure, *corrective maintenance* is done to restore the machine to its 'normal' state and any imperfect product items are either reworked or discarded. Also, *preventive maintenance* may be carried out in order to mitigate the occurence of failure events. For example, Liao and Sheu (2011) presented an economic production quantity (EPQ) model for randomly failing production process with minimal repair and imperfect maintenance.

In general, a production or control policy is required for the SELSP which defines decisions to make for the possible states of the system. These decisions are: whether to continue production of the current product; whether to switch to another product; or whether to idle the machine. The implication is that finite production capacity has to be dynamically allocated between products in order to be responsive to stochastic demands. This adds to the complexity of the problem and means that determining an optimal control policy is non-trivial (Sox et al., 1999). The critical aspects of these policies are the lot-sizing decisions and the sequencing decisions. The lot-sizing decision may either depend on the state of the entire system (i.e., a

global lot-sizing decision) or just on the stock level of the product currently set up (i.e., a local lot-sizing decision). In addition, the production sequence can either be dynamic, fixed with variable cycle length, or fixed with the cycle length fixed as well (Winands et al., 2011).

Though there are formulations of these stochastic models that can be solved analytically — e.g., Tempelmeier (2013) and Tempelmeier and Hilger (2015) — often, to evaluate solutions to the SLSP, stochastic *simulation* is employed over a set of specific or random scenarios and instances.

3.3 Solution Approaches

This section will outline the application(s) of popular approaches to both deterministic and stochastic versions of the lot sizing and scheduling problem while briefly describing the algorithms underlying them where necessary.

3.3.1 Exact Methods

Very often, CLSPs are modelled as mixed-integer linear programming (MILP) problems and solved with software such as IBM's CPLEX (Ramya et al., 2016; Dangelmaier and Kaganova, 2013; Walser et al., 1998). However, the CLSP is NP-hard (Bitran and Yanasse, 1982), and so there is a limit to the size and complexity of CLSPs that can be tackled with exact mathematical programming methods. Similarly, the SELSP can be formulated as a Semi-Markov Decision Process (SMDP) but this approach does not scale well (Graves, 1980). It is also possible to formulate an SCLSP with service-level constraints as a linearised model and solve with a standard MIP solver, but this too means prohibitive computation times with increasing products and time periods (Tempelmeier, 2013).

3.3.2 Heuristics, Meta-Heuristics, and Hyper-Heuristics

For larger and more complex scenarios, various approaches based on meta-heuristics or heuristic solution approaches have been proposed for both deterministic and stochastic lot sizing and scheduling problems. These algorithms are invaluable when problem sizes become intractable to solve with exact methods and computation costs are prohibitively large.

A *heuristic* is an algorithm which does not guarantee that the optimal solution to the problem will be found. They trade optimality for speed in order to gener-

ate ‘good enough’ solutions in a reasonable time frame. What counts as a ‘good enough’ solution will depend on the use case or specific problem. For example, for a problem where it is difficult to manually construct a (feasible) solution, a ‘good enough’ solution is one that is feasible and quickly generated. Similarly, a heuristic can be said to give a ‘good enough’ solution if subject to a computation or time budget, the solution is close in quality to known optima or benchmarks. Or in the case where this information does not exist, the solution improves significantly on random ones. Heuristics may be implemented by themselves or in conjunction with other heuristics or optimisation algorithms.

On the other hand, a *meta-heuristic* is a class of heuristic that does not make assumptions about the structure or characteristics of the underlying problem that is to be solved. Meta-heuristics work to efficiently direct the search of a subordinate heuristic on a larger decision (or search) space. To quote a more pithy definition:

“Metaheuristics are typically high-level strategies which guide an underlying, more problem specific heuristics, to increase their performance. The main goal is to avoid the disadvantages of iterative improvement and, in particular, multiple descent by allowing the local search to escape from local optima. This is achieved by either allowing worsening moves or generating new starting solutions for the local search in a more “intelligent” way than just providing random initial solutions. Many of the methods can be interpreted as introducing a bias such that high quality solutions are produced quickly. This bias can be of various forms and can be cast as descent bias (based on the objective function), memory bias (based on previously made decisions) or experience bias (based on prior performance). Many of the metaheuristic approaches rely on probabilistic decisions made during the search. But, the main difference to pure random search is that in these algorithms randomness is not used blindly but in an intelligent, biased form.” (Stützle, 1998).

Blum and Roli (2003) and Luke (2013) have other definitions of meta-heuristics and extensive descriptions of many popular algorithms as well as intuitive explanations as to their particular use cases.

In addition to heuristics and meta-heuristics, there have been some applications of hyper-heuristics to lot sizing and scheduling problems. *Hyper-heuristics* are heuristic search methods that attempt to automate the selection or design of subordinate heuristics to solve hard computational problems (Burke et al., 2013). The

distinction between a hyper-heuristic and a meta-heuristic is that the latter searches a solution space (i.e., the search space is comprised of solutions to the problem), however, the former searches within a space of heuristics.

3.3.2.1 Heuristic solution methods

Variations of construction heuristics have been used for various types of lot sizing problems — a *construction* heuristic is one that starts with an 'empty' solution and gradually builds or assembles a complete solution as determined by its algorithm procedure.

For example, Ho et al. (2006) developed two construction heuristics for the uncapacitated dynamic lot-sizing problem that are extensions of earlier heuristics by Silver and Meal (1973), and show that they outperform six other construction heuristics including the original Silver and Meal heuristic. James and Almada-Lobo (2011) proposed, along with other heuristics, a MILP-based 'relax-and-fix' construction heuristic for the parallel-machine capacitated lotsizing and scheduling problem with sequence-dependent setups (CLSD-PM). This construction heuristic solves a sequence of decomposed 'sub-MILPs' in order to construct an initial solution for the various search algorithms it is coupled with. Finally, Almada-Lobo et al. (2007) proposed a five step heuristic for finding good feasible solutions. Each step of the heuristic is either a forward or backward pass (or a combination of both) through the schedule.

For the SCLSP, Leachman and Gascon (1988) developed a dynamic cycle lengths heuristic in a discrete-time model under the assumption of non-stationary demand and deterministic production and setup times. The first step in their heuristic is the calculation of target cycle lengths in each review period via a deterministic approach by using moving averages of the demand forecasts. Graves (1980) proposed a composite-product heuristic with a composite-product defined as an aggregation of the products in the scenario. Graves then tested this heuristic against a naive procedure and other heuristics based on (Q,R)[2] and (s,S)[3] policies using simulation and showed that the novel composite-product heuristic outperforms the others. Tempelmeier (2013) focused on discrete time SCLSP models with random demands, fixing production periods and fixing lot sizes under service level

[2]This policy requires that Q items are ordered whenever the inventory position falls to the reorder point, R (Gallego, 1992).

[3]This policy dictates that no new orders are made until inventory falls to or below s, at which point an order is made to restore inventory to the level S (Caplin, 1985).

constraints. In addition, Tempelmeier and Hilger (2015) proposed linear programming models with non-linear constraints approximated by piecewise linear functions and compared a variant of the Fix-and-Optimise heuristic with the column generation heuristic proposed by Tempelmeier (2011) on a large number of test problem instances. Wagner and Smits (2004) described a model for the SELSP with the objective of minimising long-run average holding and setup costs whilst fulfilling a given service level. They used a cyclic scheduling approach with cycle times of each product a multiple of a fundamental cycle (or base period). A local search algorithm was implemented along with a myopic construction heuristic as the solution method and compared with deterministic benchmarks and on a large set of stochastic instances.

3.3.2.2 Evolutionary algorithms and genetic algorithms

Most of the deterministic lot sizing and scheduling meta-heuristic approaches use evolutionary algorithms (EAs), particularly genetic algorithms (GAs). EAs are a class of population-based meta-heuristics inspired by biological evolution. They maintain a population of candidate solutions which are improved by applying one or more of recombination, reproduction, and mutation operators on them and then selecting the fittest individuals (Vikhar, 2016; Michalewicz et al., 1997). In addition, EAs have already demonstrated some promise in dealing with problems that integrate uncertainty (Jin and Branke, 2005). The most popular of evolutionary algorithms are GAs. These GAs generally 'evolve' a set candidate solutions, each represented by a chromosome, over a specified number of generations (or iterations) to produce high-quality solutions to a search problem (Holland, 1975; Goldberg and Holland, 1988; Goldberg, 1989). Chromosomes are versatile and flexible, and can take a wide range of representations e.g., binary or bit strings, permutations, real-valued numbers, rule-sets, or combinations thereof. The set of candidate solutions is improved on generationally by mimicking the concept of natural selection and survival-of-the-fittest. Fitter individuals are generally selected to be 'parents' and be combined in some fashion to create 'children' that will make up the subsequent generation. As these progeny generally share traits with their parents, mutation operators are applied that make small random changes to them in order to explore more of the search space.

The GA-based approaches to lot sizing can be broadly divided into approaches using a *direct representation* or an *indirect representation*, where the former ap-

pears much more often. In a direct representation, the sequence and lot sizes are directly encoded in the chromosome. The main challenge with such an approach is that mutations and crossovers can generate infeasible solutions, which is usually dealt with by discarding those solutions or by special repair operators (Özdamar and Birbil, 1998). Methods with an indirect representation use a mapping function or heuristic to derive a production plan from a solution's chromosome. An indirect GA representation has been proposed by Kimms (1999). In Kimms' paper, a two-dimensional matrix is used as chromosome, with each entry representing a rule for selecting the set up state for a machine at the end of a period (e.g., the item with maximum holding costs, minimum set up cost, maximum depth, maximum number of predecessors). To compute the fitness value of a chromosome, a construction scheme is called, which constructs the solution backwards, starting from the end of the planning horizon.

A lot more work has been published on GAs for the job-shop scheduling problem (JSP), and they typically use a permutation-based representation, and then apply a construction heuristic to actually construct the schedule based on the permutation (Cheng et al., 1999; Branke and Mattfeld, 2005; Bierwirth and Mattfeld, 1999). A typical construction heuristic is the Giffler-Thompson algorithm (Giffler and Thompson, 1960), which generates active schedules by iteratively selecting the job with the highest priority (lowest permutation index) from the set of eligible jobs, and then scheduling it at the earliest possible time. This approach is used in work such as Branke and Mattfeld (2005) where the objective is to minimise tardiness. However, this approach does not directly transfer to biopharmaceutical capacity planning or lot sizing, because (i) scheduling as early as possible would lead to excessive storage costs and (ii) the existence of a heterogeneous set of alternative facilities.

3.3.2.3 Other meta-heuristic approaches

Apart from GAs, other meta-heuristic approaches to the deterministic problem include tabu search (TS) or particle swarm optimisation (PSO), see, e.g., Piperagkas et al. (2012), and Guner Goren et al. (2008). Ant Colony Optimisation (ACO) has also been used for uncapacitated and capacitated multi-level problems (Pitakaso et al., 2007; Almeder, 2010). In both cases, ACO was used to determine production decisions from top items to raw materials and a MILP solver is used to calculate the corresponding production and inventory levels. Almada-Lobo and James

(2010) extended previous work of Almada-Lobo et al. (2007) by using their five-step heuristic as an initial starting solution for a TS and variable neighbourhood search meta-heuristic to solve the CLSP with sequence-dependent times and costs.

In the case of a bi-objective CLSP problem, Mehdizadeh et al. (2016) developed two novel multi-objective meta-heuristic algorithms and compared them with the non-dominated sorting genetic algorithm II (NSGA-II).

In general, Jans and Degraeve (2007) reviewed and compared meta-heuristic solution approaches for the CLSP, and there are good introductions to these other meta-heuristic approaches in Luke (2013), Eberhart et al. (2001), Dorigo and Stützle (2010), Glover and Laguna (1999), and by Deb et al. (2000). Usually, TS and ACO are more suited to combinatorial optimisation problems whilst PSO can be used on problems that have a real-valued decision space.

3.3.2.4 Hyper-Heuristics

Branke et al. (2016) surveyed the use of hyper-heuristics in generating or designing reusable construction heuristics for production scheduling. In the indirect GA representation proposed by Kimms (1999) (which was mentioned previously), since the entries in the chromosome represented a rule for selecting set up states, the approach can be seen as a selection hyper-heuristic as the search space is on potential rules and not direct solutions to the problem.

Hyper-heuristics may incorporate machine-learning techniques such as Artificial Neural Networks (ANNs) or genetic programming (GP). GP is an EA technique where computer programs, usually based on a tree-representation, are searched for and evolved (Koza, 1992). Burke et al. (2007) have demonstrated automated heuristic generation with genetic programming. For a complex dynamic scheduling problem, Pickardt et al. (2013) proposed a two-stage hyper-heuristic for the generation of work centre-specific dispatching rules. This hyper-heuristic comprised a genetic program that evolves a composite rule and an EA that searches for good allocation of rules between the work centres. An ANN is a computing system that comprises a set of connected nodes (artificial neurons) which can transmit signals to each other computed as a function of each neuron's inputs (Haykin, 1994). The edges connecting neurons have associated weights which determine the connection strength — i.e., it weights the effect of the respective incoming signal in the computation of a neuron's output signal. Branke et al. (2015) investigated three different rule representations for optimising rules to compute priority indices for

new/arriving jobs in a jobshop environment. In addition to a linear representation, a feed-forward ANN, and GP with tree-representation was employed.

3.3.3 Simulation Optimisation

For the SELSP *simulation optimisation* is often used as an alternative approach. This refers to, as the name suggests, attempts to couple optimisation techniques with simulation analysis. Its objective is to find decision variables for optimal system performance with performance being evaluated via simulation. Simulation optimisation is a powerful technique useful for problems with complex or unknown structure where uncertainty is present (Amaran et al., 2016) and its applications include supply-chain management, inventory replenishment, process design, and bioprocess control (Chu et al., 2015; Jalali and Van Nieuwenhuyse, 2015; Caballero, 2015; Renotte and Wouwer, 2003) with heuristics and meta-heuristics often used as the optimisation algorithm.

Recent work in terms of the multi-item SELSP includes the study by Löhndorf and Minner (2013) who formulated the problem as a SMDP, and compared different solution approaches including approximate value iteration and global search on simple production policies that had either fixed or dynamic cycles. They found that global control policy search outperforms average value iteration on large problems. For their global search algorithm, Löhndorf and Minner utilised a type of EA called Covariance Matrix Adaptation Evolution Strategy (CMA-ES) (Hansen and Ostermeier, 2001). CMA-ES generates individuals by sampling from a multivariate normal distribution. Each generation, the underlying distribution is mutated to explore the search space. Dependencies between the different coordinates in the chromosome vector are described by a covariance matrix which is updated each generation in order to guide the search to more promising regions. Löhndorf et al. (2014) then extended that work to consider sequence-dependent setup times. Both of these papers use meta-heuristics to conduct the global search for control policies, and both of these papers' approaches can also be classed as a hyper-heuristic. Briskorn et al. (2016) presented a fixed cyclic production scheme for multiple products with control strategies to stabilise the cycle length and consider sequence-dependent setup times, backlogging with service level constraints, and limited storage capacity. They used a nested solution approach comprising three levels utilising iterative and neighbourhood search procedures.

Nourelfath (2011) determined robust production plans for the SCLSP to ensure

that specified service level is met with high probability. The model accounted for random machine breakdowns and random repair times independent of product type and lot size. It did not consider random demand nor preventive maintenance planning. On the other hand, Purohit and Kumar Lad (2016) presented a mathematical model to provide an integrated plan incorporating job sequencing, lot sizing, and a schedule for preventative maintenance which is solved with the use of a simulation-based GA approach and outperforms previous conventional approaches.

3.4 The Bioprocessing Context

Literature on capacity planning or lot sizing and scheduling in the pharmaceutical or biopharmaceutical industry represent complicated extensions to the CLSP, with multiple products and facilities, product-specific manufacturing rates and costs, multi-stage processing, and perishable products. This also applies to the SLSP which must also consider semi-continuous processes that are prone to different types of failure events, as well as variable reactor titres. For the deterministic problem, they have applied primarily mathematical programming models based on discrete time-periods which are solved using MILP solver software.

For example, Lakhdar et al. (2005) developed a deterministic mixed-integer linear program for the planning and scheduling of a multi-product biopharmaceutical manufacturing facility and later extended it for use with a multi-facility model where multiple criteria were considered using goal programming (Lakhdar et al., 2007). Siganporia et al. (2014) also developed a MILP model, in this case to optimise an eight-year planning horizon for a mixture of fed-batch and continuous bioprocesses while considering capacity decisions in a few scenarios with different demands and bioreactor titres. Siganporia et al. utilised rolling-time horizons to improve computational performance. Each of these models is based on discrete time periods and allows only one product to be manufactured in each time-period. In the case of Lakhdar et al. (2007), where discrete 90 day periods are used, this means that at most four different campaigns (lots) can be scheduled per year and facility. As a result, this effectively artificially restricts the search space.

Recently, the work done by Lakhdar et al. (2005), has been extended to alternative approaches by other authors. First, Vieira et al. (2016) solved a set of example problems based on a Resource Task Network (RTN) continuous-time single-grid formulation focusing on addressing specific operational characteristics of biopro-

cesses. Jankauskas et al. (2017) then used a continuous-time model optimised by a GA which is underpinned by a dynamic chromosome structure that is allowed to vary in length.

For problems with stochastic elements considered, Gatica et al. (2003) and Levis and Papageorgiou (2004) presented a mathematical programming approach for the capacity planning problem, but with a focus on long-term planning and capacity investment decisions under clinical trials uncertainty rather than scheduling. Lakhdar et al. (2006) extended their deterministic medium-term planning formulation to include uncertain production rates and dealt with this using chance-constrained programming. Marques et al. (2017) presented a simulation optimization approach combining a MILP model and Monte Carlo simulation procedure to integrate process design and planning decisions under clinical trial and demand uncertainty for the pharmaceutical industry. Finally, Pollock et al. (2013b) developed a discrete-event simulation model focused on investigating the economic benefits of continuous perfusion culture and single-use technology for a mAb. As part of this evaluation, stochastic process failure events and their consequences are considered using simulation. This is then extended by Pollock et al. (2017) to include an assessment of various integrated continuous process flowsheets.

Chapter 4

Lot-Sizing & Scheduling for Biopharmaceuticals

4.1 Introduction

To spread risk, companies usually have a portfolio of various products, and manufacturing takes place across a network of different facilities, including in-house facilities and outsourced manufacturing at contract manufacturing organisations (CMOs). The facilities' capabilities usually vary with respect to the set of products they can produce and technical, operational and economic characteristics will often differ between facilities as well as for different products that can be manufactured on the facility. Furthermore, products have a finite shelf-life and cannot be stored for very long.

Given the large investments, high costs, and long time-frames they face, biopharmaceutical companies have to plan ahead over a long time horizon, based on a demand forecast for each time period. It is important that production schedules are optimised to make best use of the available production capacity, and even small improvements can have a substantial impact on a company's profit. Taken together, these characteristics make biopharmaceutical capacity planning and scheduling challenging to optimise. The result — a variant of the capacitated lot-sizing problem — constitutes a complex combinatorial optimisation problem.

Because of the simplifications required to model the problem in a mixed-integer linear programming (MILP) approach — such as large discrete time periods that permit the manufacture of only one product in each period on a facility — the solution potentially suffers from an artificial restriction of the search space. This chap-

ter describes the development of a more flexible meta-heuristic approach for the biopharmaceutical lot sizing and scheduling problem, and contrasts it with the proposed mixed-integer programming approach as described by Lakhdar et al. (2007). The work in this chapter is based on a 2017 publication in *Journal of Hueristics* (see Appendix D).

To this end, a genetic algorithm (GA) with an embedded problem-specific construction heuristic is designed which is inspired by previous GA approaches to job shop scheduling. The GA uses an indirect permutation encoding, i.e., the specifically developed construction heuristic schedules demands sequentially in the order prescribed by the chromosome. As will be demonstrated, the use of a GA allows for a more flexible and realistic model of the real-life problem and avoids some of the simplifications necessitated by available mathematical programming models.

It is interesting to note that the construction heuristics previously reviewed (see Chapter 3) operate sequentially in either a forwards or backwards pass through the schedule, or a combination thereof. Instead, the construction heuristic proposed here inserts jobs in an order of importance determined by the GA and not necessarily in any chronological order.

The chapter is structured as follows. First, the problem is formally described in Section 4.2. Section 4.3 describes in more detail the case study used to evaluate the approach. The GA and the associated construction heuristic are explained in Section 4.4. The results of the empirical evaluation, including a comparison with an MILP approach, are reported in Section 4.5. This chapter closes with a section summarising the major conclusions of the sections preceding it.

4.2 Mathematical Formulation

This section summarises the mathematical formulation used here and introduced by Lakhdar et al. (2007) to solve a deterministic long-term multi-product scheduling/capacity planning problem on multiple facilities.

4.2.1 Notation

The indices i, p, and t denote individual facilities, products, and time periods respectively. The subsets characterising the facilities being considered are: PI_i, the set of products produced by facility i; IP_p, the set of facilities that can produce product p; and TI_i, the set of time periods in which facility i is available for use.

Binary Variables

Y_{ipt}	1 if product p is produced over period t at facility i; 0 otherwise
Z_{ipt}	1 if a new campaign of product p at facility i is started in period t; 0 otherwise

Integer Variables

B_{ipt}	amount of product p produced over period t at facility i, batches

Continuous Variables

I_{pt}	amount of product p stored over period t, kilograms
K_{ipt}	amount of product p produced over period t at facility i, kilograms
$Prof$	expected operating profit, RMU[1]
S_{pt}	amount of product p which is sold over period t, kilograms
T_{ipt}	production time for product p at time period t at facility i
Tf_{it}^{tot}	total production time over period t at facility i
W_{pt}	amount of product p wasted over period t, kilograms
Δ_{pt}	amount of product p which is late over period t, kilograms

[1] relative monetary units (RMU)

Parameters

C_p	storage capacity of product p, kilograms
D_{pt}	demand of product p at time period t, kilograms
r_{ip}	production rate of product p at facility i, batches per unit time
H_t	available production time horizon over time period t
T_{ip}^{max}	maximum production time for product p
T_{ip}^{min}	minimum production time for product p
yd_{ip}	yield conversion factor, kilograms per batch
α_{ip}	lead time for production of first batch of product p at facility i
ζ_p	life time of product p, number of time periods t
υ_p	unit sales price for each kilogram of product p, RMU per kilogram
η_{ip}	unit cost for each batch produced of product p in facility i, RMU per batch
ψ_p	unit cost for each new campaign of product p, RMU
δ_p	unit cost charged as penalty for each late kilogram of product p, RMU per kilogram
ρ_p	unit cost for each stored kilogram of product p, RMU per kilogram
π	rate of backlog decay

4.2.2 Constraints

4.2.2.1 Production constraints

Constraint (4.1) represents batch processing. The number of batches produced in facility i of product p at time period t, B_{ipt}, is determined by a continuous production rate, r_{ip}, production lead time, α_{ip}, and production time T_{ipt}. The lead time allows for the duration of the first batch of a campaign plus the setup and cleaning time before the first batch commences. Incorporation of lead time is enforced by a binary variable Z_{ipt}.

$$B_{ipt} = Z_{ipt} + r_{ip}(T_{ipt} - \alpha_{ip}Z_{ipt}) \quad \forall\, i,\ \ p \in PI_i,\ \ t \in TI_i. \tag{4.1}$$

Constraint (4.2) converts the number of batches into kilograms produced using a yield conversion factor yd_{ip} which differs for each combination of facility and

product. Lead time is only avoided in a facility if the same product is manufactured in the preceding period; this is covered in (4.3), with Y_{ipt} being a variable that specifies whether product p is produced by facility i in time period t. Constraint (4.4) ensures that at most one product p is manufactured in any given facility i per time period t.

$$K_{ipt} = B_{ipt}\, yd_{ip}, \quad \forall\, i,\ \ p \in PI_i,\ \ t \in TI_i. \tag{4.2}$$

$$Z_{ipt} \geq Y_{ipt} - Y_{ip,t-1}, \quad \forall\, i,\ \ p \in PI_i,\ \ t \in TI_i. \tag{4.3}$$

$$\sum_{p \in PI_i} Y_{ipt} \leq 1, \quad \forall\, i,\ \ t \in TI_i. \tag{4.4}$$

4.2.2.2 Timing constraints

Constraints (4.5) and (4.6) represent the appropriate minimum and maximum production time constraints. These are only active if Y_{ipt} is equal to 1, otherwise the production times are forced to 0.

$$T_{ip}^{min} Y_{ipt} \leq T_{ipt}, \quad \forall\, i,\ \ p \in PI_i,\ \ t \in TI_i. \tag{4.5}$$

$$T_{ipt} \leq \min\{T_{ip}^{max}, H_t\}\, Y_{ipt}, \quad \forall\, i,\ \ p \in PI_i,\ \ t \in TI_i. \tag{4.6}$$

4.2.2.3 Storage constraints

The following constraints enforce an inventory balance for production and force total production to meet product demand. In (4.7), the amount of product p stored at the end of the time period, I_{pt}, is equal to the amount stored in the previous period, plus the total amount produced across all facilities i, less the amount sold, S_{pt}, and the amount of product wasted, W_{pt}, in the current time period t. Product stored cannot be negative and should not exceed maximum product storage capacity in (4.8); and total inventory at any point cannot exceed the global storage capacity in (4.9).

$$I_{pt} = I_{p,t-1} + \sum_i K_{ipt} - S_{pt} - W_{pt}, \quad \forall\, p \in PI_i,\ \ t \in TI_i. \tag{4.7}$$

$$0 \leq I_{pt} \leq C_p, \quad \forall\, p,t. \tag{4.8}$$

$$0 \leq \sum_{p} I_{pt} \leq C_P^{tot}, \quad \forall\, t. \tag{4.9}$$

The duration a product can be stored in inventory is limited by its shelf-life in (4.10). Stored material will not be left to expire whilst in inventory and this is done by ensuring that the final product is sold in less than ζ_p time periods from when it is first stored.

$$I_{pt} \leq \sum_{\theta=t+1}^{t+\zeta_p} S_{p\theta}, \quad \forall\, p,t. \tag{4.10}$$

4.2.2.4 Backlog constraints

A penalty is incurred for every time period t that a given amount of product p is late. For a given product p at time t, the amount of product that is late, Δ_{pt}, is equal to the amount of undelivered product from the previous time period, $\Delta_{p,t-1}$, multiplied by a factor, π_p (which allows for the backlog to decay), plus demand at time t, D_{pt}, less the sales at time t, S_{pt}.

$$\Delta_{pt} = \pi_p \Delta_{p,t-1} + D_{pt} - S_{pt}, \quad \forall\, p,t. \tag{4.11}$$

4.2.3 Objective Functions

4.2.3.1 Single objective

The objective function is to maximise profit, which is the difference between total revenue (sales in kilogram times price υ_p), and total operating costs which include the changeover cost at ψ_p per setup, storage cost at ρ_p per kilogram of product, late delivery penalties of δ_p per kilogram of product, and batch manufacturing cost at υ_{ip} for every product-facility combination. All costs and prices are in relative monetary units (RMU).

$$\max \quad \text{Profit} = \sum_{p} \sum_{t \in TI_i} (\upsilon_p S_{pt} - \rho_p I_{pt} - \delta_p \Delta_{pt} - \sum_{i \in IP_p} (\eta_{ip} B_{ipt} + \psi_{ip} Z_{ipt})). \tag{4.12}$$

The equations (4.1) – (4.12) comprise the complete formulation of the MILP problem to be optimised and subsequently compared with the GA approach.

4.2.3.2 Other objective(s)

In addition to single-objective of maximising profit, there is a second objective of maximising customer service level (CSL) which is used in multi-objective optimisation(s).

$$\max \quad \text{CSL} = (\sum_{p}\sum_{t} S_{pt})/(\sum_{p}\sum_{t} D_{pt}). \tag{4.13}$$

4.2.4 Formulation Assumptions

This derived formulation assumes (and consequently makes the restriction that) not more than one product can be manufactured in any given period. As a result, the MILP requires this to be solved but the GA does not which is an advantage of the GA approach that will be elaborated on later in Section 4.5.1 of this chapter.

In addition, in any given period, lead times and associated setup costs can be avoided on a facility if the same product is manufactured in the preceding period. What this means is, that for the MILP, there could potentially be as much as period's length of time between two batches which are in two different but adjacent periods. In order to fairly compare with the MILP, the GA is implemented such that lead times and associated setup costs are accrued if the time between two batches of the same product is greater than the time period used by the MILP. This maximum idle time allowed between two batches before setup time and cost is accrued is referred to as the *setup 'expiration' period*.

4.3 Industrial Case Study

An industrial case study presented by Lakhdar et al. (2007) was used to evaluate the proposed method. This is anonymized real world data comprising anticipated market demand and manufacturing facility characteristics. This benchmark problem features multiple products to be produced on multiple facilities with different efficiencies and costs, setup times, batch production, perishable inventory, and the ability to backlog demand.

The demand forecast comprises a time horizon of 15 years and 15 products ($p1$ – $p15$). The forecast indicates yearly market demands, assumed to be fulfilled at the end of each year (Table 4.1)[2]. The demand can be scheduled across 10

[2]Note that the product 1 demand for year 10 in Table 4.1 in Lakhdar et al. (2007) was 63, which is not consistent with the general trend of the other years, so it was changed to 163.

Table 4.1: Product Demand Forecast for Industrial Case Study (Products p1–p15) [kg]

	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y11	Y12	Y13	Y14	Y15
p1	21	32	18	28	61	104	153	156	164	163	161	162	162	163	165
p2	6	5	4	4	4	3	3	3	3	3	3	3	2	2	2
p3	12	43	38	5	22	52	97	132	133	135	137	118	109	100	90
p4	583	628	655	687	758	921	989	941	993	649	621	573	521	468	421
p5	12	12	11	10	9	7	6	5	4	3	2	2	2	2	3
p6	211	200	245	246	257	266	284	274	226	180	166	151	137	123	110
p7	4	5	5	7	6	5	8	9	8	9	7	7	6	5	5
p8	5	5	5	7	6	5	8	9	8	9	7	7	6	5	5
p9	15	15	15	13	12	9	8	6	5	4	3	3	2	2	2
p10	72	99	104	102	111	120	130	139	188	120	106	93	81	69	58
p11	552	615	699	737	743	733	684	572	518	471	424	381	342	307	274
p12	5	5	5	7	6	5	8	9	8	9	7	7	6	5	5
p13	211	252	290	298	286	216	169	153	150	145	110	100	93	84	102
p14	2	2	4	3	3	3	16	11	13	16	16	16	16	17	17
p15	4	4	5	6	16	11	24	32	37	40	41	42	42	43	44

facilities (i1 – i10), but not all facilities can produce all 15 products. All facilities are assumed to be available for the entire time horizon apart from facility 6 (i6) which is unavailable until Y2, and facility 9 (i9) which is unavailable until Y11. Of the ten manufacturing facilities, i1, i4, i6, and i9 are in-house facilities while the rest are owned by CMOs.

Production rates (Table 4.2), manufacturing yields (Table 4.3) and manufacturing costs (Table 4.4) are specified for all facility-product combinations (*RMU* in these tables denotes relative monetary unit). The manufacturing yield determines how many kilograms of a specific product are produced in a batch for a specific facility. The manufacturing cost of a product is thus also dependent on the yield. Setup cost and time are incurred when a facility is switching between products. For consecutive batches of the same product, no setup time/cost is involved.

The setup 'expiration' time is 90 days (equal to the discrete time period used by the MILP) which defines the maximum amount of time that a facility can be idle without accruing setup times and costs on the subsequent batch produced. This accounts for the extra equipment preparation activities (cleaning, sterilisation, etc.) required after prolonged idle time. There is also a restriction on the time a product may be stored before it has to be thrown away which is the maximum shelf-life. In the case that the demand cannot be fulfilled in time, it is backlogged, but there is a backlog penalty for every unit that is not delivered on time. Also, backlogged demand decays exponentially at a rate of 50% every three months. For example, if a demand of 100 kg cannot be delivered on time, 6 months later, only 25 kg could actually be sold, and 75 kg of the demand would have been lost, reducing the revenue correspondingly.

The case study assumes a fixed sales price, changeover cost, storage cost, and setup time for all products (Table 4.5)[3]. The setup time includes the time of production of the first batch. In addition, it is assumed that a month is 30 days and, subsequently, a year is equal to 360 days.

The objective is to maximize the overall profit, calculated as total revenue minus the cost for production, storage, setups and backlog penalties. Given a set of heterogeneous facilities with different manufacturing yields, manufacturing cost, and batch production rates for different products, this takes into account maximizing the amount of products sold, and minimizing the manufacturing cost, the

[3]Note that the description by Lakhdar et al. (2007) had some inconsistencies in the units specified, so in Table 4.5 the units for setup cost, sales price, storage cost and backlog penalty were updated to be consistent with the other data.

Table 4.2: Production Rates of Facilities (i1–i10) for Case Study [batch/day]

	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p13	p14	p15
i1	0.35	0.39	0	0.45	0	0.29	0	0.35	0.25	0.39	0.41	0.39	0	0.12	0.35
i2	0.6	0	0	0.61	0	0.6	0	0.6	0	0.43	0.56	0	0.6	0.6	0.6
i3	0	0	0	0	0	0	0	0	0	0	0	0	0.23	0	0
i4	0	0	0	0.12	0	0	0	0	0	0	0	0	0	0	0
i5	0	0	0	0.45	0	0	0	0.45	0	0.45	0.45	0	0	0.45	0.45
i6	0	0	0	0.45	0	0	0	0.45	0	0.45	0.45	0	0	0.45	0.45
i7	0	0	0	0	0	0	0.45	0	0	0.45	0	0	0	0	0
i8	0	0	0.58	0	0.45	0	0	0	0	0	0	0	0	0	0
i9	0.45	0	0	0.45	0	0.45	0	0	0	0.45	0.45	0	0	0.45	0.49
i10	0.45	0.45	0	0.45	0	0.45	0	0.45	0.45	0.45	0.49	0.45	0.45	0.45	0.45

Table 4.3: Manufacturing Yields of Facilities for Case Study [kg/batch]

	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p13	p14	p15
i1	10	1	0	8	0	6	0	10	2	9	7	1	0	12	12
i2	9	0	0	8	0	6	0	9	0	8	10	0	10	12	11
i3	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0
i4	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0
i5	0	0	0	10	0	0	0	10	0	8	8	0	0	11	11
i6	0	0	0	12	0	0	0	10	0	8	17	0	0	17	14
i7	0	0	0	0	0	0	10	0	0	10	0	0	0	0	0
i8	0	0	36	0	10	0	0	0	0	0	0	0	0	0	0
i9	10	0	0	12	0	5	0	0	0	8	16	0	0	12	13
i10	9	1	0	12	0	5	0	10	2	8	14	1	10	12	12

Table 4.4: Manufacturing Costs of Facilities for Case Study [RMU/batch]

	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p13	p14	p15
i1	1	1	0	10	0	3	0	1	1	1	3	1	0	1	1
i2	10	0	0	5	0	2	0	5	0	10	2	0	2	5	2
i3	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
i4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
i5	0	0	0	20	0	0	0	20	0	20	20	0	0	5	20
i6	0	0	0	10	0	0	0	10	0	10	10	0	0	1	10
i7	0	0	0	0	0	0	10	0	0	10	0	0	0	0	0
i8	0	0	1	0	5	0	0	0	0	0	0	0	0	0	0
i9	10	0	0	10	0	10	0	0	0	10	8	0	0	1	10
i10	15	15	0	15	0	15	0	15	15	15	15	15	15	15	15

Table 4.5: Case Study Parameters

Parameter	Value	[Unit]
Setup time	14	days
Setup cost	2	RMU/changeover
Setup 'expiration' time	90	days
Sales price	2.5	RMU/kg
Storage cost	0.01	RMU/(kg $\times$ *period*)
Storage period	90	days
Shelf life	2	years
Production time per year	360	days
Backlog decay	0.5	per 3 months
Backlog penalty	0.1	RMU/kg

storage cost, the setup cost, and any backlog penalty.

Lakhdar et al. (2007) used mixed-integer linear programming (MILP) to solve this problem (described in Section 4.2) and the GA proposed in this chapter is subject to the same constraints, except that production on each facility is not restricted to one product per time period. This seemed to be an artificial restriction imposed only to reduce the modelling complexity of the MILP.

4.4 The Proposed Genetic Algorithm with Construction Heuristic

For job shop scheduling, many successful GAs use indirect encodings (e.g., Branke and Mattfeld (2005) and Cheng et al. (1999)), with the GA only searching the space of permutations of jobs. For evaluation, a schedule is constructed from the permutation by a construction heuristic, often Giffler-Thompson, which iteratively selects the job with the highest priority (lowest permutation index) from the set of eligible jobs, and then schedules it at the earliest possible time. This avoids infeasible solutions and introduces a desirable heuristic bias, in the sense that it excludes obviously bad solutions (such as schedules with big gaps) from the search space. Inspired by this work, an indirect, permutation-based encoding combined with a construction heuristic is proposed. The construction heuristic, however, had to be carefully designed for the problem at hand.

The following two subsections first explains the proposed construction heuristic, then provides details on the GA used.

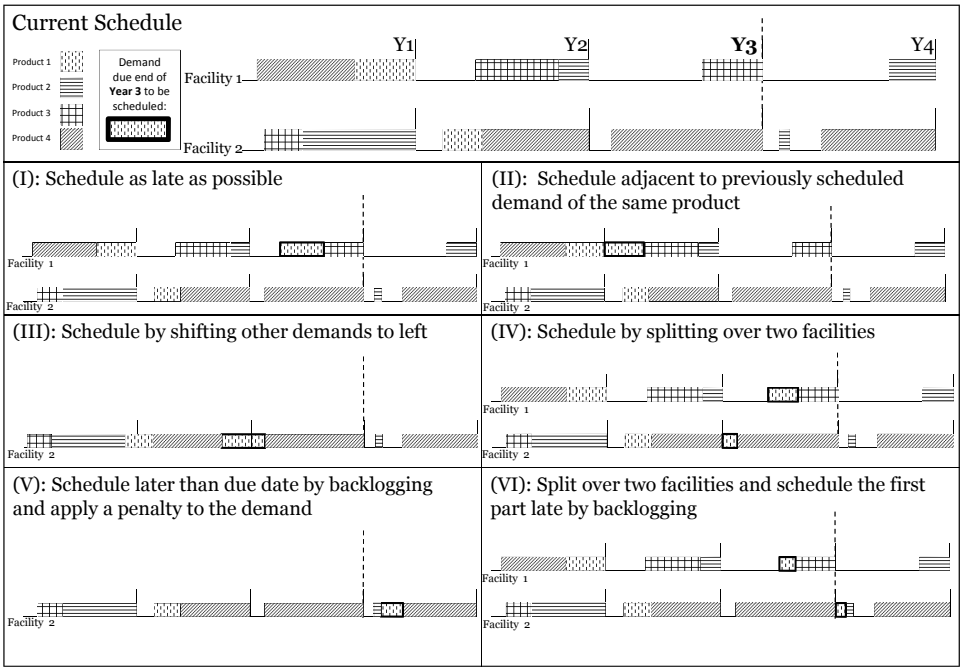


Figure 4.1: Visualisation of construction heuristic, based on a simple example with two facilities and four demands. Items *(I)* — *(VI)* show the alternatives the heuristic considers when identifying the most profitable place to insert a new demand into the schedule. The rectangle representing a demand includes the setup time (so length varies depending on where the new demand is inserted). Note that just using *(I)* or *(II)*, feasible options are found on Facility 1 but not on Facility 2. Therefore the heuristic will terminate its search on Facility 1 but continue on Facility 2 using *(III)*, *(IV)*, *(V)*, and *(VI)*.

4.4.1 Construction Heuristic

The construction heuristic works on the basis of forecasted demands, in this case demands for each year and product (see Table 4.1). Its task is to schedule the production to satisfy the demand (referred to as demand from now for the sake of simplicity) sequentially, in the order prescribed by the GA. When deciding at what time and what facility to insert a new demand into the schedule, the heuristic explores a number of different alternatives, and then greedily picks the alternative that creates the smallest additional cost. So, the heuristic will consider each facility in which the product may be produced. It then tries to schedule the entire demand in an uninterrupted way as late as possible to minimise storage cost, and as late as possible but adjacent to already scheduled demand of the same product to avoid setup cost and time.

Only if these alternatives are not feasible for a facility, e.g., because a facility does not have a sufficiently large gap in its schedule, further options are explored that either move some of the already scheduled demands to make sufficient space for the new demand, split the demand into two parts and schedule the second part in another facility, or backlog the demand.

Figure 4.1 provides a simple example based on just two facilities and four products, while Algorithm 1 lays out brief pseudo-code of the construction heuristic. The six alternatives considered shall now be explained in detail.

(I) **Schedule as late as possible.** The first alternative considered is to schedule the entire demand as late as possible but before the due date, as one uninterrupted block, which minimizes storage cost at this facility. In the example, this is possible for Facility 1, see Figure 4.1 *(I)*, but not for Facility 2, since there is not sufficient uninterrupted capacity available to schedule the entire demand.

(II) **Schedule next to previous demand.** To avoid setup times and setup costs, it may be beneficial to schedule a demand adjacent to the same product already scheduled. The heuristic picks the latest time slot before the due date that allows it to link to a previously-scheduled demand of the same product, and has sufficient available capacity to schedule the entire demand - see Figure 4.1 item *(II)*. Again, this is only possible on Facility 1, as Facility 2 does not have sufficient uninterrupted capacity. Note that due to the avoided setup time, the overall time required to produce the demand is smaller.

Algorithm 1 Pseudocode of the construction heuristic.

procedure CONSTRUCTION HEURISTIC(job J)
  Determine possible time window for J ensuring batches finish before due date but do not expire before due date
  for each facility i **do**
    G_{i1} = latest gap that can fit J ▷ *(I)*
    G_{i2} = latest gap that fits J and links to job of same product ▷ *(II)*
    if $G_{i1} + G_{i2} == \{\}$ **then**
      Find latest gap that can fit at least one batch, G_{i3}
      Split J into two parts r_1 and r_2 such that r_1 is largest size that can fit in G_{i3}
      $G_{i3} = G_{i3}+$ SECOND FACILITY SEARCH(i, G_{i3}, r_2) ▷ *(IV)*
      Find latest gap
      Attempt to enlarge gap by left-shifting already scheduled jobs without violating shelf-life dependencies, G_{i4} ▷ *(III)*
      $Gi5$ = the first gap past or straddling due date, that is big enough for penalized job, J' ▷ *(V)*
      Find earliest gap past or straddling due date that fits at least one batch, G_{i6}
      Split J' into two parts r'_1 and r'_2 such that r'_1 is largest size that can fit in G_{i6}
      $G_{i6} = G_{i6}+$ SECOND FACILITY SEARCH(i, G_{i6}, r'_2) ▷ *(VI)*
    end if
  end for
  Evaluate overall cost for each facility and gap, and pick the one with minimal cost, min Cost$(G_{ij})\ \forall\ i, j$ ▷ Construct and add to schedule
end procedure

procedure SECOND FACILITY SEARCH(facility i, gap G, remainder of job R)
  for each facility $k \neq i$ **do** ▷ The remaining facilities
    F_{k1} = latest gap that can fit R
    F_{k2} = latest gap that fits R and links to job of same product
    if $F_{k1} + F_{k2} == \{\}$ **then**
      F_{k3} = the first gap past or straddling due date, that is big enough for penalized remainder of job, R'
    end if
  end for
  Evaluate overall costs, Cost$(F_{kj})\ \forall\ k, j$, and return cheapest option
end procedure

If, on a particular facility, none of the above two alternative insertion attempts resulted in a feasible solution, the following options are explored. Specifically, the steps attempted are: to move already scheduled demands, to split a demand, and to backlog a demand.

(III) **Move previously scheduled demands.** Since there was not a sufficiently long gap in the current schedule to allocate the entire production for the new demand, one possibility to create a feasible schedule may be to shift previously-scheduled demands to an earlier time to make space for the new demand. Thereby, the heuristic identifies the latest gap in the considered facility before the due date. All conflicting scheduled demands before this gap are shifted backward in time (towards the start of the planning horizon), without changing the order, and just enough to make space for the new demand. This can be seen in Figure 4.1 item *(III)* for Facility 2, where four previously scheduled demands had to be left-shifted to make space for the new demand.

(IV) **Split demand**. Another option to fit the demand may be to split the new demand. In this alternative, the heuristic will again consider the latest gap before the due date, and use all available consecutive capacity. Then, it will attempt to schedule the rest of the demand at each of the other facilities, but only considering options *(I)*, *(II)*, and *(V)* (which is described below). An example is provided in Figure 4.1 item *(IV)*, where only a small fraction of the demand can be scheduled at Facility 2, and the remainder is then moved to Facility 1. Note that splitting the demand may cause an additional setup time and setup cost. A demand can only be split into two i.e., a demand cannot be split more than once.

(V) **Backlog.** If the facilities are really busy, it may be best (or the only feasible option) to backlog the demand. This means that the time slot allocated to produce the material to meet the demand falls partly or wholly later than the due date for the demand. As described in the case study, this will result in a monetary penalty and part of the demand being lost, as is reflected in Figure 4.1 *(V)* by the smaller rectangle for the scheduled demand. In order to reduce the magnitude of the penalty, the heuristic will schedule the demand as early as possible in a gap that either straddles, or is later than, the due date. An example is provided in Figure 4.1 item *(V)*.

(VI) **Backlog and split.** As a kind of last resort, with this alternative, the heuristic will combine steps *(IV)* and *(V)*. As in *(IV)*, the demand is split, but rather than using the latest gap before the due date, the first part of the demand is scheduled in the earliest gap after the due date. The remaining portion of the demand is attempted again to be scheduled in all other facilities, but only using options *(I)*, *(II)* or *(V)*. This is illustrated in Figure 4.1 item *(VI)*.

The above alternatives will be evaluated for all the facilities that are capable of producing the product. Then, the demand is inserted into the schedule according to the most profitable alternative examined, and the algorithm moves on to schedule the next demand.

Overall, if there are n facilities, in the worst case the heuristic considers $6n^2 - 2n$ alternatives: (n) alternatives for each of the options *(I)*, *(II)*, *(III)* and *(V)*, and then $3n(n-1)$ alternatives each for option *(IV)* and option *(VI)*, due to different possibilities in scheduling the remaining part of a demand in case of a split. This means that in the worst case the complexity of the construction heuristic is $O(mn^2)$, where m is the number of demands and n the number of facilities. In practice, however, as will be shown later, in the majority of cases, only options *(I)* and *(II)* are explored per facility.

Note that batch production means that unless the demand is exactly equal to an integer multiple of the batch size (which itself is different for different facilities) it is not possible to produce exactly the required demand. In such cases, the number of produced batches is always rounded up to the minimal integer number of batches necessary to fulfill the demand. The amount overproduced in such a case is put in storage, possibly to be used to (partly) fulfill future demand. Before going through the steps above to insert a demand into the schedule, the construction heuristic will always check whether the product is in the storage, and try to partially fulfill the demand from storage. The cost associated with this is storage cost only, as manufacturing costs are invoked at the time of production, i.e., when a previous scheduled demand produced that overcapacity. Products left in storage that the heuristic can not use in later steps are considered lost and have no value.

It is interesting to note that the construction heuristics previously reviewed (see Chapter 3 on page 14) operate sequentially in either a forwards or backwards pass through the schedule, or a combination thereof. Instead, the construction heuristic proposed here inserts jobs in an order of importance determined by the GA and not necessarily in any chronological order.

4.4.1.1 Reneging

With *reneging*, the heuristic has the choice of either producing as much of the demand as it can or deciding not to produce the whole demand or just a part of it. This is simultaneously implemented in two ways.

1. Demand reneging: In all options *(I)-(VI)* the heuristic can decide based on a cost evaluation, whether it will schedule the entire demand or just fulfil what it can from storage and renege on the rest of the obligated demand.

2. Split reneging: In options *(IV)* or *(VI)*, after the job/demand has been split into two, the heuristic can decide based on a cost evaluation whether to schedule the second part of the split or renege on it.

The cost evaluation is done by comparing the cost of scheduling the most profitable alternative as normal, C_A, and the cost of reneging (i.e., not producing), C_R. The cost of reneging includes the backlog costs and also the lost revenue(s). This comparison is of the form:

$$C_R < R_c C_A. \tag{4.14}$$

Where R_c is an arbitrary coefficient and $0.0 < R_c \leq 1.0$. The comparison is such that if Equation (4.14) evaluates as true, the heuristic will renege on the demand (or second part of the split) so manufacturing capacity is not allocated for its production. However, as with demands that are scheduled, part of the demand would still be fulfilled by any existing material in storage.

4.4.2 Genetic Algorithm

The quality of the solution produced by the above construction heuristic is to some extent dependent on the order in which the demands are inserted into the schedule since available production capacity is more restricted the later a demand is considered. By giving priority to certain demands, mainly three situations can be created.

1. Demands of the same product that should be ideally scheduled consecutively to avoid setup costs, can be assigned similar priorities, making it very likely that the construction heuristic will link them together.

2. Demands that are best scheduled just before the due date to save storage cost can be given a high priority. This will lead to the construction heuristic

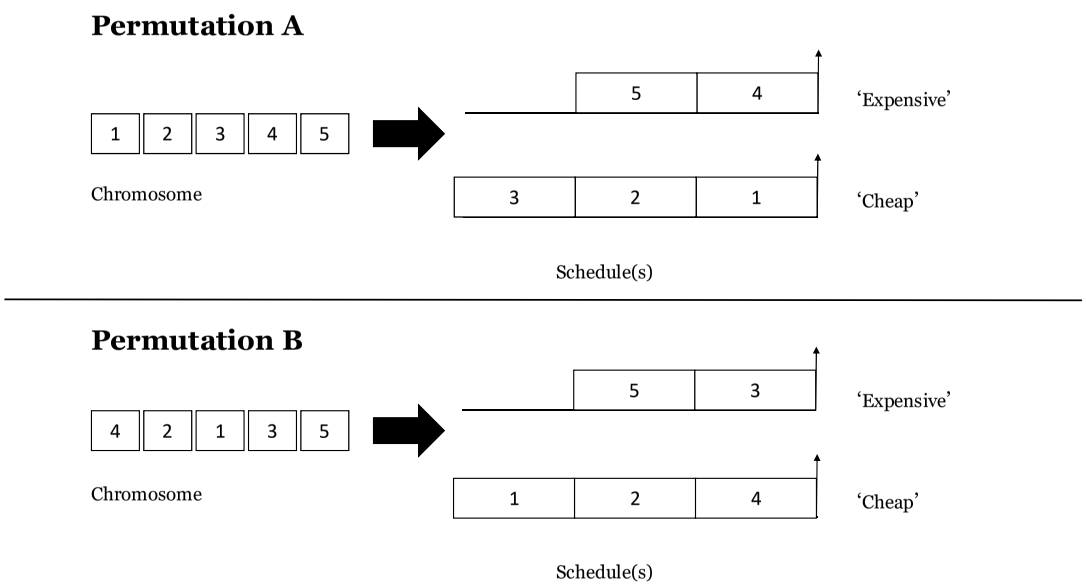


Figure 4.2: Illustration of the chromosome encoding, based on a simple example with two facilities and five demands. Individuals A & B are both different permutations of demands 1-5. Here the construction heuristic is processing demands in the chromosome from left to right and filling up the 'Cheap' facility first before placing demands on the 'Expensive' facility. In this case this results in distinctly different schedules.

scheduling these demands early on, at a time where still a lot of capacity is available, and the cheapest option just before the due date would be selected.

3. Demands that benefit most from a highly utilised facility (e.g., because all other facilities are much more expensive), can also be given high priority, which will lead to early scheduling when this highly demanded facility is still available.

Optimising this order is left to the GA, which was implemented in Java™ using the ECJ library (Version 23) (Luke, 1998). It used a permutation representation of all the demands to be scheduled, i.e., 225 demands in the industrial case study used here (the number of elements in Table 4.1). Specifically, each demand was given a unique ID number (from 1 to 225), and the chromosome is a permutation of these numbers. The ordering of the numbers on the chromosome determines the order by which the construction heuristic processes the respective demands (from first to last position) and thus influences the resulting schedule (see Figure 4.2 on the previous page). In the cases that reneging was turned on and the coefficients were being optimised, the chromosome would resemble that which was just described but two extra genes would be appended to the chromosome. The 226th gene would correspond to the reneging coefficient for 'demand reneging' whilst the 227th gene would hold the value for that of 'split reneging'. The range of values allowed for these two extra genes was $0.0 < x \leq 1.0$ to correspond with the description of reneging coefficients in Section 4.4.1.1.

Originally, individuals were initialised randomly, but then it was determined that better solutions are produced if demands from a single year are grouped together on the chromosome. Unless stated otherwise, the results in this chapter are thus based on runs where 50% of the population is initialised randomly, whereas the other 50% only randomise the sequence of demands from the same year, but maintain the sequence of years (i.e., all demands of a particular year appear in the permutation before the demands of later years).

The genetic operators shared by both single- and multi-objective optimisations are as follows. For crossover, the operator used was the Precedence Preserving Crossover (PPX) proposed by Bierwirth et al. (1996) which ensures that if a demand i is before a demand j in both individuals, this will also be true in the offspring. To ensure the absolute precedence relations of two parent permutations, an empty child is initialised and a vector equal to the lengths of the parent is randomly filled from the set $\{1,2\}$ to determine the order in which genes are picked from the

first and second parent respectively. When a gene is picked from either parent, it is appended to the child and deleted from both parents. This ends when both parents are empty and the child contains all the genes. In the case that genes 226 and 227 were included in the chromosome, one of each was picked from either parent (with equal probability) to be in the resultant child of the PPX procedure. Selection of potential parents for crossover was done using stochastic universal sampling (SUS). SUS is a fitness proportionate selection method that has minimum spread and zero bias (Baker, 1987). Compared with standard fitness proportionate roulette selection where there is a single pointer indicating the 'winner', SUS has N equally spaced pointers where N is the number of samples to keep. This gives members of the population with weaker fitness a chance to be chosen by sampling the population at evenly spaced intervals. The mutation operator used was shift mutation, which iterates through every element of the permutation and, with probability $p_m = 0.02$, removes a demand and re-inserts it at a new random position. The reneging coefficient genes were mutated also with the same probability but with Gaussian mutation and a standard deviation $\sigma = 0.1$.

The operators and parameters specific to either one of single- or multi-objective optimisation runs are described separately below. In either case, for fitness evaluation, the GA called the construction heuristic described in Section 4.4.1 which builds a schedule by inserting demands iteratively in the order prescribed by the solution's chromosome.

4.4.2.1 Single objective

In the single objective case the actual fitness is the overall profit of the resulting schedule, i.e., revenue minus storage, production, setup cost and backlog penalty. This objective function is Equation (4.12) defined in Section 4.2. Experiments with different population sizes and mutation rates showed that results were rather insensitive to the parameter settings (see Table 4.6).

For the rest of the chapter, the GA parameters used were: a population size of 30, and generational reproduction with elite of 6. The algorithm was run for 1500 generations, and all results are based on averages over 50 runs (unless otherwise stated). These, and the other parameters used were chosen based on preliminary experiments and rules-of-thumb for evolutionary computation practice.

Table 4.6: Profit performance for base case (in RMU) ± std. err. for three different population sizes and mutation rates.

Population Size	Mutation Rate		
	0.01	0.02	0.03
20	66612 ± 0.9	66601 ± 1.0	66594 ± 0.9
30	66613 ± 0.8	66604 ± 0.9	66593 ± 0.9
60	66612 ± 0.8	66603 ± 0.8	66592 ± 0.8

4.4.2.2 Multiple objectives

In addition to the single-objective of maximising profit, investigations were conducted into multi-objective optimisation. The other objective considered was CSL.

This multi-objective optimisation maximised both profit and CSL, and therefore was a bi-objective optimisation. The profit objective is as described previously and the CSL is likewise described in Section 4.2 as Equation (4.13) on page 34. The optimisation algorithm used was the non-dominated sorting genetic algorithm II (NSGA-II) which was also implemented in Java™ using the ECJ library (Luke, 1998). The evolution parameters were such that a population size of 150 was run for 300 generations with a 0.9 crossover probability for 50 separate runs. These parameters were chosen after preliminary experiments to give a suitably sized Pareto set and enable clear visualisations of the attainment surfaces.

4.5 Empirical Evaluation

4.5.1 Comparison with Mathematical Programming

The algorithm was run on the case study described in Section 4.3, and results for this are reported in Table 4.7 on the next page as "Standard case/GA Model". As it turns out, the case study has ample production capacity, which is due to modelling the option of outsourcing production at higher cost as additional facilities. To see how the algorithm would perform in a more loaded scenario, variations of the case study were also tested where the demand is increased in each year by a factor of 2 or 3, and the results of these experiments are reported in Table 4.7 as well.

To judge the performance of the proposed algorithm, it was compared with the MILP implementation as described by Lakhdar et al. (2007) and replicated in Section 4.2. However, there is one important difference that deserves discussion.

Table 4.7: Profit, CSL, and other characteristics for GA Model and MILP Model. For the GA, mean ± std. err. are listed, whereas MILP is a deterministic method and was only run once. Best mean is highlighted in **bold** - where the difference is not significant both are highlighted. GA timing values are the average time elapsed for each of the 50 runs (i.e., the total runtime of 50 runs divided by 50).

	"Standard case"		2× Demand		3× Demand	
	GA Model	MILP Model	GA Model	MILP Model	GA Model	MILP Model
Revenue	**74533**	74490.9	**148952** ± 18.3	148389.8	**222530** ± 89.8	221603
Manufacturing Costs	**7274** ± 0.69	7452	**20367** ± 25.12	20541	42427 ± 83.7	**40346**
Storage Costs	**337**± 0.9	447.4	**862** ± 4.8	952.8	1698 ± 11.7	**1559.9**
Setup Costs	318 ± 0.9	**272**	330 ± 1.20	**274**	342 ± 1.56	**276**
Backlog penalties	**0** ± 0.0	3.3	**9** ± 1.5	53.4	**88** ± 7.5	156.7
Profit	**66604** ± 0.9	66316	**127385** ± 11.0	126568.7	177975 ± 52.6	**179265**
CSL	**100%**	99.9%	**99.9%** ± 0.01%	99.5%	**99.5%** ± 0.04%	99.1%
Time (s)	105.1	600.5	184.3	600.4	269.5	600.4
Optimality Gap	-	0.25%	-	0.64%	-	0.92%

The MILP model has variables that specify how much is produced for each facility, product and time period. It thus requires the problem to be broken down into discrete time periods, and it allows for at most one product to be produced in a particular facility and time period. The choice of the length of a time period is somewhat arbitrary, but has huge implications. If the time period is chosen very large, then most demands would require only a fraction of a time period to be produced. The facility would then be idle in the remaining part of the time period, leading to poor solutions. On the other hand, if the length of a time period is chosen to be rather short, because the number of batches to be produced is integer, often a fraction of the time period remains unused (e.g., if a time period is 5 days, and producing a batch takes 3 days, only one batch can be produced in each time period and 2 days in each time period remain unused — up to the point where a time period is too short for even one batch and there is no feasible solution). Furthermore, reducing the length of the time period increases the number of variables and constraints quite significantly, with corresponding drastic implications on running time. Some experimenting concluded that the 90 day period used by Lakhdar et al. (2007) indeed performs well, and all results are based on this time granularity.

In contrast to the MILP model, the GA can work with arbitrary time periods, and even continuous time, without any implications on running time. In this implementation, the smallest time unit was chosen to be a single day. This allows a model closer to reality than the mathematical programming implementation. As a result, the GA sometimes is able to produce solutions with a higher profit than the MILP approach, even if MILP is run to optimality. Note that it is not claimed that the MILP implementation by Lakhdar et al. (2007) is the best possible, or that it is not possible to design an MILP formulation that circumvents or at least reduces the impact of the time period length. However, the MILP model is the only one found in the literature for this problem, and one of the advantages of GAs is their greater flexibility in modelling the real world, and that solving a problem heuristically that is close to reality can sometimes work better than a model further from reality but solved to optimality.

All these being considered, although the GA and the MILP are attempting to solve the same problem, the two approaches are actually solving very similar (but not exactly the same) models. Hence, in the comparison between both approaches they will be distinguished by calling the former ‘GA model’ and the latter ‘MILP model’ from here on.

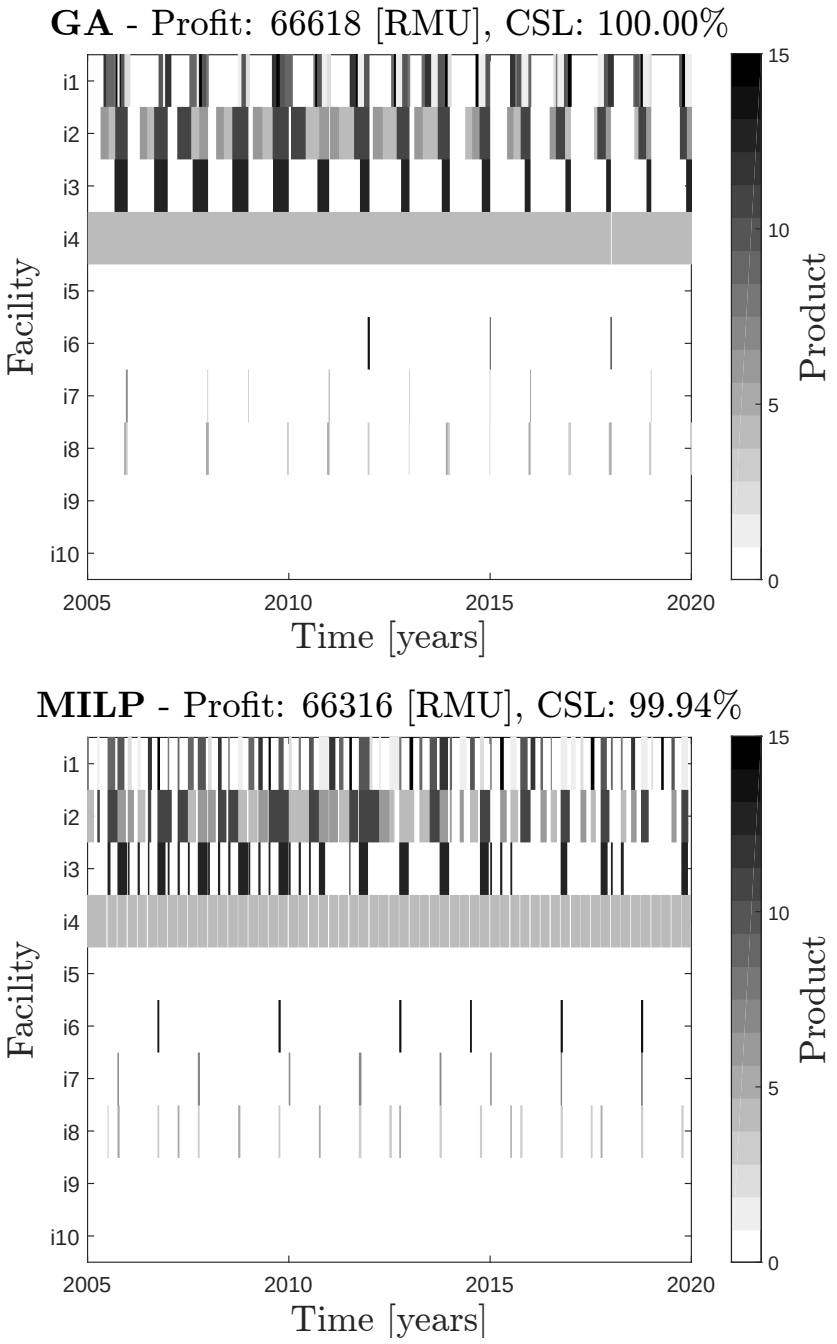


Figure 4.3: Exemplary Gantt chart of a schedule generated by the GA model (top) and MILP model (bottom), for the standard case. The profit and customer service level (CSL) for each schedule is also indicated.

Table 4.8: Runtime of MILP until it reached an optimality gap of 0.25%, and runtime of GA to reach the same solution quality as was reached by MILP, for different problem sizes, depending on problem size.

	15 years	23 years	30 years
Target (RMU)	66284	90236	111229
MILP Time (s)	200.86	824.134	1332.59
GA Time (s)	0.07	0.131	0.195

Results from the MILP model and the GA model are compared in Table 4.7. As can be seen, in the standard case as taken from Lakhdar et al. (2007), the GA model solution has lower manufacturing costs (i.e., utilises the low-cost facilities better), and lower storage costs. It also manages to satisfy all the demand (CSL of 100%), whereas the MILP model chooses to backlog some of the demand. This is because the GA tries to satisfy all the demand as first priority and only backlogs if there is no other feasible option. The MILP model, however, has an explicit trade-off between backlog and other costs, and backlogs if the resulting solution has a higher profit. On the other hand, the setup costs of the GA model solution are higher. Overall, the profit generated by the GA model solution is consistently higher, and by more than the 0.25% optimality gap, i.e., the difference between the best solution found and the upper bound determined by the MILP solver. This is possible because the MILP model, due to its imposed time granularity, has an artificially restricted search space. It can switch less often between products, resulting in lower setup cost and higher storage cost. Also, it sometimes wastes part of a time period, which may mean the need to use occasionally more expensive facilities, resulting in higher manufacturing costs. These differences can be seen also by comparing the Gantt charts of the optimal solutions found in the MILP model and the GA model which are depicted in Figure 4.3. The Gantt chart of the MILP solution generally shows shorter campaigns (sequences of batches of the same product), and, especially visible on facility i4, small gaps between production in different time periods, simply because the time period (of 90 days) is not equivalent to a duration spanned by a multiple of batches for this product in this facility. The schedule optimised by the GA model has longer un-interrupted idle time, which may be advantageous if a new product is introduced to the facility or if a third party is seeking to rent and use production capacity.

For the scenario with twice the demand, the conclusions are similar to the base

case. However, for three times the demand, it seems backlogging becomes crucial, and the MILP model approach seems better in doing that. While backlogging reduces the products sold due to lost demand and thus reduces revenue, the savings that can be achieved in terms of manufacturing cost and setup cost seem to outweigh this loss, and the overall profit of the MILP model approach is higher in this scenario. Whether a slightly higher profit justifies a lower CSL is a different issue. The GA's construction heuristic, always tries to meet all the demand, even if this may lead to a possibly lower profit. Finally, it can be observed that the GA model's solution still has lower storage cost and higher setup cost, probably due to not being constrained by the coarse time periods.

Runtimes strongly depend on the implementation skills of the developer, the hardware used, and software tools used, and thus have to be handled with caution. Nonetheless, Table 4.7 also reports on the runtime of the two algorithms. For the MILP, the stopping criterion was 600s, so the runtime remained the same, but the optimality gap increased as the problem became more difficult by increasing the demand and thus utilisation level. The GA was run for a fixed number of generations. The computational time still increased with increasing demand level. The reason is that an increasing demand raises the utilisation level and the construction heuristic is then less likely to be able to schedule a demand in steps *(I)* or *(II)*, and thus more often has to look at the other alternatives for scheduling it. This will be explored further in the next subsection.

Also, the scaling behaviour of both optimisation methods with increasing problem sizes. To do this, the GA and MILP were run for problems with longer time horizons of 23 years and 30 years (in addition to the 15 year-long base case). For the longer time horizons, the demand forecasts for the years after year 15 were set equal to the forecast for each product in year 15. To compare the two methods, the MILP was first run for all three problem sizes with a stopping criterion of a 0.25% optimality gap, at which point, the solution quality (profit) was recorded along with the time taken to achieve the solution. This profit value was then used as a target for the GA. The average time over the 50 runs that it took for the GA to match or beat those targets was recorded. The results of this experiment are shown in Table 4.8. As can be seen, the time required for the MILP model and GA model increases roughly linearly with problem size, however the factor by which runtime increases when moving from 15 to 30 years is 6.6 for the MILP model, but only 2.8 for the GA model.

Table 4.9: Breakdown of how often each part of the heuristic is used in optimised solutions, mean ± std. error. Also detailed is the percentage of separate jobs that are delivered late.

	1 × Demand	2 × Demand	3 × Demand
(I)	71.9% ± 0.18%	70.7% ± 0.27%	64.1% ± 0.25%
(II)	20.9% ± 0.19%	16.8% ± 0.22%	12.8% ± 0.20%
(III)	0.3% ± 0.03%	2.8% ± 0.13%	4.9% ± 0.14%
(IV)	6.9% ± 0.03%	8.3% ± 0.17%	10.4% ± 0.12%
(V)	0.0%	1.3% ± 0.18%	7.7% ± 0.22%
(VI)	0.0%	0.0%	0.04% ± 0.02%
Total Backlogged Jobs	0.0%	1.8% ± 0.25%	9.2% ± 0.25%

Overall, from these results, the conclusion is that the suggested GA approach is competitive with the MILP approach, but does not suffer from the introduction of artificial time periods and thus is sometimes able to find better solutions than MILP. The trend seems to be that the computation times for both optimisation methods are going up linearly. However the increase of the GA approach is of a smaller factor than that of the MILP optimisations. This suggests that the relative performance of the GA is less susceptible to the detrimental impact of increasing the scale of the problem.

4.5.2 Algorithm Components

In order to better understand the importance and robustness of the various components of the algorithm, some additional experiments were done.

Table 4.9 examines how often the various alternatives to insert a demand are actually selected by the construction heuristic, averaged over the best solution found in each of the 50 runs. As can be seen in the table, in the standard case, the majority of demands (92.8%) are inserted by either scheduling it as late as possible *(I)*, or adjacent to a previous demand of the same type *(II)*. This is reassuring, since if such an insertion is possible, the other options are not tested, which significantly speeds up the algorithm. Moving to the scenarios with higher demand, the percentage drops from 92.8% to 78.9%. This still constitutes the majority of cases, but clearly the other insertion alternatives of the heuristic become more important.

Figure 4.4 looks at the relevance of the alternatives *(III)-(VI)* in terms of their impact on profit. It shows the ratio of the obtained profit depending on whether the construction heuristic during the GA search was limited to looking at alterna-

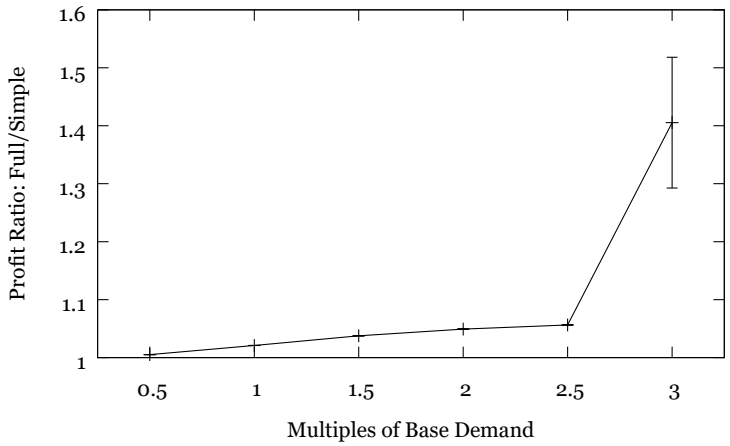


Figure 4.4: The ratio in profit of the full model compared to the simple model, optimized by the GA, shown for multiples of the base market demand as presented in the case study. The increasing ratio indicates the increasing benefit the full model will have in more complicated scheduling problems than the basic case study. Error bars represent the standard error.

tives *(I)* + *(II)* (denoted as "Simple"), or all alternatives ("Full"). A profit ratio of 1 means that the two models obtain the same profit, while a greater profit ratio indicates that the full model is able to achieve higher profits than the simple model. It confirms that the more complicated cases with splitting, backlogging and moving previously scheduled demand are responsible for an increasing share of the profit as the overall demand is increased. Especially once the demand is increased to three times the original values, there seems to be a step change and the more complicated alternatives seem to become indispensable.

Figure 4.5 shows the convergence of the GA over generations. This is compared with random search using fully random permutations, and random search using limited random permutations generated in the same way as the GA's initial population (i.e., when half of the permutations are only random amongst demands of the same year, but the order on years is kept). Both random search algorithms evaluate N points at random, where N is the product of the number of generations and population size used by the GA. As can be seen, the results optimised by the GA are considerably better than the results obtained by random search. The limited randomisation helps in particular for the less loaded problems ($1 \times$ Demand), but is no longer better than fully randomised permutations for the case of $3 \times$ Demand. This also makes sense, as with higher utilisation of the facilities, there is

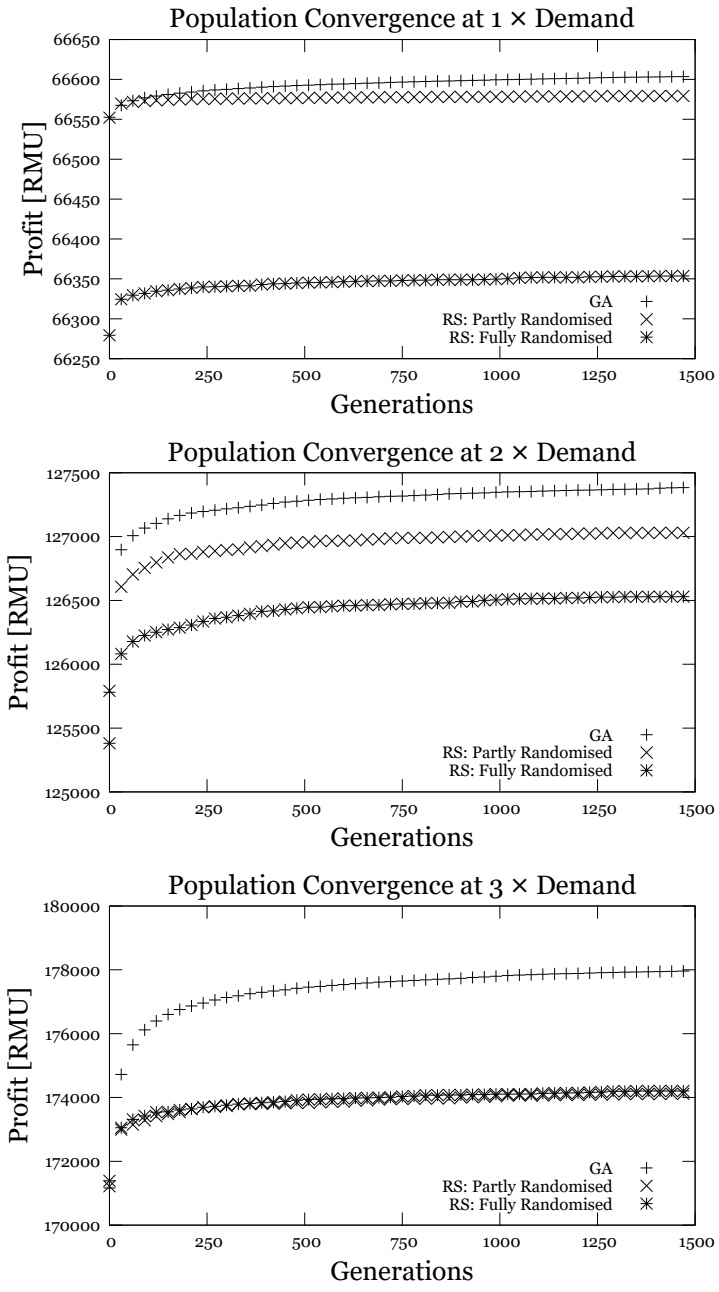


Figure 4.5: Convergence of profit over generations, for GA and random search. Random search is tested with fully randomised permutations and where 50% of the permutations are randomised only amongst demands of the same year, but keep the order on years. The points reported for random search are the best so far every 30^{th} evaluation. RS = random search.

Table 4.10: Comparison of profit, CSL, and other characteristics for the GA at 3× demand with reneging turned off, and the reneging coefficients fixed or optimised. Figures are mean ± std. error over 50 runs, and units are in RMU unless otherwise indicated. Best mean is highlighted in **bold** - where the difference is not significant, both are highlighted.

		Reneging	
	No Reneging	Fixed	Optimised
Revenue	**222704 ± 36.8**	222492 ± 37.3	222475 ± 38.4
Manufacturing	42331 ± 56.13	**41132 ± 55.36**	**41244 ± 58.45**
Storage	**1710 ± 11.8**	1732 ± 9.1	**1702 ± 9.9**
Setup	355 ± 1.65	**326 ± 1.55**	330 ± 1.98
Backlog penalties	**58 ± 3.0**	90 ± 3.5	84 ± 3.2
Profit	178251 ± 54.2	**179213 ± 34.9**	179116 ± 48.2
CSL (%)	**99.601 ± 0.0164**	99.506 ± 0.0167	99.498 ± 0.0172

increasing need to schedule demands outside the year the demand is delivered, and the artificial limitation of randomisation to within a year is no longer helpful.

To examine the effect of reneging on performance, the heuristic was run three more times at the 3× Demand case. In the first, the option to renege was turned off. In the second case, the reneging coefficients were both fixed to be 1.0. However, in the third, the reneging coefficients were allowed to be independently optimised by the GA. In all of these cases, any reneged or otherwise unscheduled demands could be partly fulfilled from stored material in inventory if any was available.

The results from this are summarized in Table 4.10. As expected, with reneging, the heuristic is able to achieve higher profits than without but at the expense of a lower CSL. This shows that the Reneging component can be an improvement to the performance of the optimisation on the whole because it enables the GA to explore more of the search space. Comparing the results of when the reneging coefficients are fixed or are allowed to be optimised by the GA, it is seen that the former approach performs better in terms of profit achieved but the difference in CSL is not significant. This seems to suggest that, for single-objective of maximising profit, the best strategy in implementing a reneging rule is to choose a common-sense value, like 1, for the coefficients and there is no apparent benefit in trying to search for potentially more optimal values.

It is worth noting that these results are significantly better than the corresponding results for the GA at 3× Demand in Table 4.7 with the ability to partly fulfil

demands that are not scheduled (or that are reneged) which demonstrates that this addition to the heuristic is an improvement to its performance[4]. Also, with reneging turned on, the GA is actually now able to produce solutions that achieve a better profit than the optimal MILP solution though the mean profit is still slightly lower.

One of the drawbacks of heuristic methods is that they do not always guarantee that the optimum solution can be reached. This is true of the GA and heuristic that has been presented and evaluated here. In fact, in this case, the optimum cannot be guaranteed to be in the search space as the heuristic cannot cover the entire search space. For one, it excludes schedules that are not as close to their deadline as possible or linked to another demand. Secondly (and more importantly), the heuristic can only split a demand once. This limits the ability to spread heavy demand loads over all available facilities. As previously shown in Table 4.9 on page 54, the splitting mechanics — i.e., *(IV)* and *(VI)* — become more important with increased load. Therefore, there are solutions the heuristic can never reach even with brute-force approaches.

4.5.3 Multi-Objective Optimisation

In order to demonstrate the versatility of the algorithm in dealing with more than one objective, it was run twice as a multi-objective problem using NSGA-II at 3 $\times$ Demand case. In both cases reneging was turned on but for one, the reneging coefficients were fixed to be equal to 1 and in the other the coefficients were left to be optimised by the algorithm. The parameters used by NSGA-II are as previously described in Section 4.4.2.2 on page 48.

4.5.3.1 Bi-objective: profit and customer service level

As was seen in Section 4.5.1, at increasing demand loads a trade-off begins to appear between maximising profit and CSL. Therefore this multi-objective optimisation would enable a decision-maker to select a solution that meets their priorities.

First, the multi-objective results from fixing reneging coefficients and from optimised reneging coefficients were compared. In both cases, the non-dominated final solutions (an approximation/nondominated set) from each of the 50 NSGA-II runs were used to generate 50 attainment surfaces using the method and source

[4]The only difference between the '3$\times$Demand - GA' results column in Table 4.7 and the 'No Reneging' results column in Table 4.10 is that the latter has this ability to partly fulfil demands that are not scheduled (or are reneged) from material held in inventory.

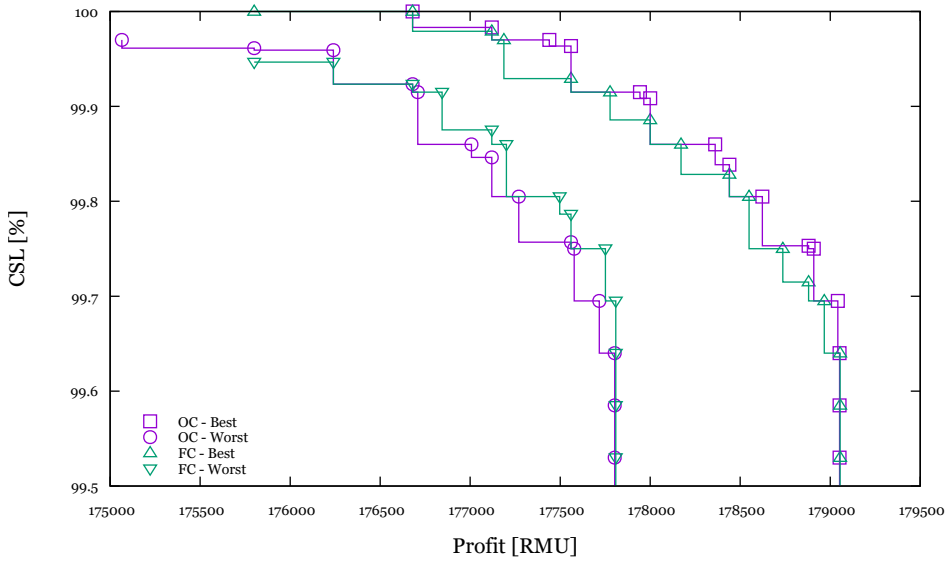


Figure 4.6: Attainment surfaces from the bi-objective optimisation at 3 × Demand to simultaneously maximise profit and CSL over 50 runs. OC = optimised reneging coefficients; FC = fixed reneging coefficients.

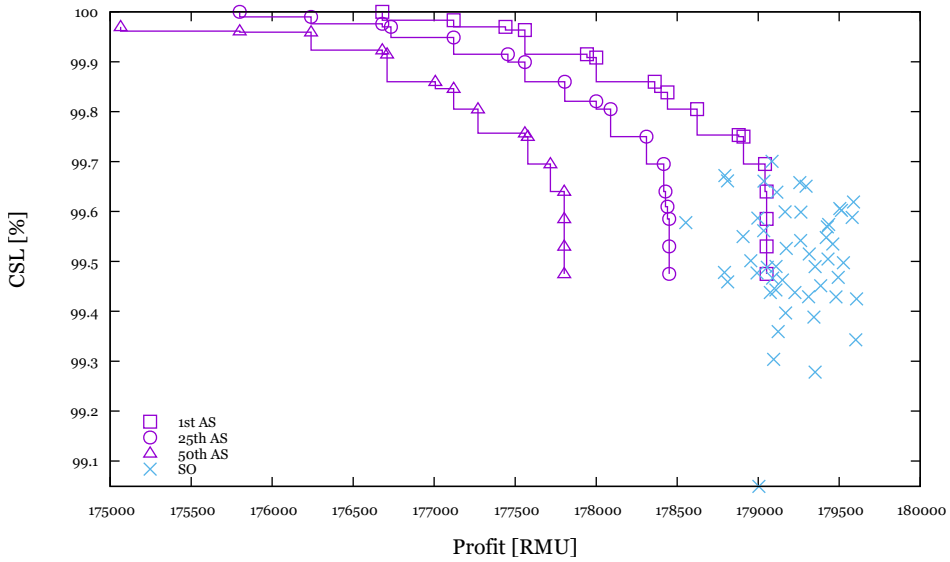


Figure 4.7: The final solutions of the single-objective optimisation to maximise profit compared with the 1st, 25th, and 50th attainment surfaces from the bi-objective optimisation to maximise profit and CSL at 3 × Demand. AS = attainment surface; SO = single-objective solutions.

code by Knowles (2005). Figure 4.6 compares these two cases by plotting the best and worst attainment surfaces for each. The figure shows how allowing the NSGA-II to optimise the reneging coefficients may lead to better multi-objective results. In the best case (i.e., the first attainment surfaces) having optimised reneging coefficients leads to a superior performance — this set largely dominates the corresponding set for the fixed coefficients. The likely reason for this is because fixed reneging coefficients don't allow a wider sampling of potential solutions, i.e., it has a narrower search space. In the worst case (i.e., the 50^{th} attainment surfaces), the attainment surface corresponding to fixed reneging coefficients appears to be dominating the optimised reneging coefficients, meaning that optimised reneging coefficients have a larger variance across its 50 approximation sets. This observation is not surprising because, as mentioned earlier, having optimised reneging coefficients allows for a wider search space. In either of the best cases, the attainment surfaces include solutions that achieve a 100% CSL.

Then, in Figure 4.7, the single-objective results (from Section 4.5.2) were compared with the multi-objective results at the 3 $\times$ Demand case. This shows the first, median, and last (worst) attainment surfaces for the multi-objective optimisation where the reneging coefficients are allowed to be optimised plotted with the final solutions from the single-objective case where the reneging coefficients were fixed (to equal 1). As can be seen, the single-objective solutions clearly dominate the multi-objective solutions with regard to profit which is unsurprising as its single objective *is* to maximise profit. However, as previously mentioned, the NSGA-II was able to capture a solution which manages to achieve 100% CSL which may be of benefit to a decision-maker who prioritises meeting all customer demand over a higher profit; the maximum CSL the single-objective GA can manage in its final solutions is 99.7%. In general, the NSGA-II captures a greater spread of different solutions whilst the GA produces a smaller cluster of similar solutions.

4.6 Summary

In this chapter, the lot sizing and scheduling problem was considered for a complex biopharmaceutical production scenario featuring multiple products, multiple facilities, and batch processing. For this challenging optimisation problem, a GA was proposed based on an indirect permutation encoding that is decoded into a full schedule by a novel construction heuristic tailored to the problem at hand.

As noted, e.g., by a recent survey (Jans and Degraeve, 2008), most meta-heuristics developed for lot sizing are validated only on artificial test data, failing to demonstrate that they can tackle the complexities of real-world problems. However, the work in this chapter validates the proposed meta-heuristic approach on industrial data.

A comparison with an MILP approach from the literature showed that the GA is at least competitive, and often produces even better results than the MILP approach. The reason is that the MILP model artificially imposes a time granularity by dividing the time into discrete periods that is not needed in the GA approach. This shows that although GAs are heuristic methods, they can sometimes outperform exact methods not only in terms of running time, but also because they are able to work with a model closer to reality. In addition, the performance of both approaches were compared at increasing problem scale (by investigating longer planning horizons) with the results suggesting that the relative performance of the GA is less sensitive to the effects of increasing the problem size.

Finally, the heuristic was extended such that it was able to explore more of the search space. As a result, it was demonstrated that this approach is easily adaptable to multiple-objectives by optimising a set of non-dominated solutions that maximised profit and CSL which provides a decision-maker with a set of optimised alternatives.

Chapter 5

Scheduling Strategies for Continuous Bioprocesses

5.1 Introduction

The run time[1] of a continuous operation is determined by several factors including cell line stability, culture productivity, product quality, process economics, and operational reliability and consistency (Ozturk, 2015). Clearly, this can have an effect on any scheduling or capacity planning that takes place on the facility that it is processed on. This is because there is a trade-off between short and longer run times. As the process goes on for longer, the probability of failure increases, which has associated costs, clean-up, and long lead-times to restart the process. On the other hand, if the process is too short, because of the seed train, ramp-up and changeover costs, these setup costs become dominating. That is sales of product are not sufficient to cover these costs as not enough product is made before the process is ended and restarted.

This is illustrated in Figure 5.1 on the following page: given the turnaround on a bioreactor, and the seed train duration, one can choose a target batch run time from which then follows the time (relative to the progress of the USP operation) at which to start a new seed train so that it ends just before the bioreactor is able to start its next batch (if the current batch runs for the target batch time). Here the seed train takes 14 days, the target USP duration is 30 days, and the turnaround for the bioreactor is 4 days. Therefore the stagger left for the seed train is such that the

[1] Alternatively: process/batch length or process/batch duration.

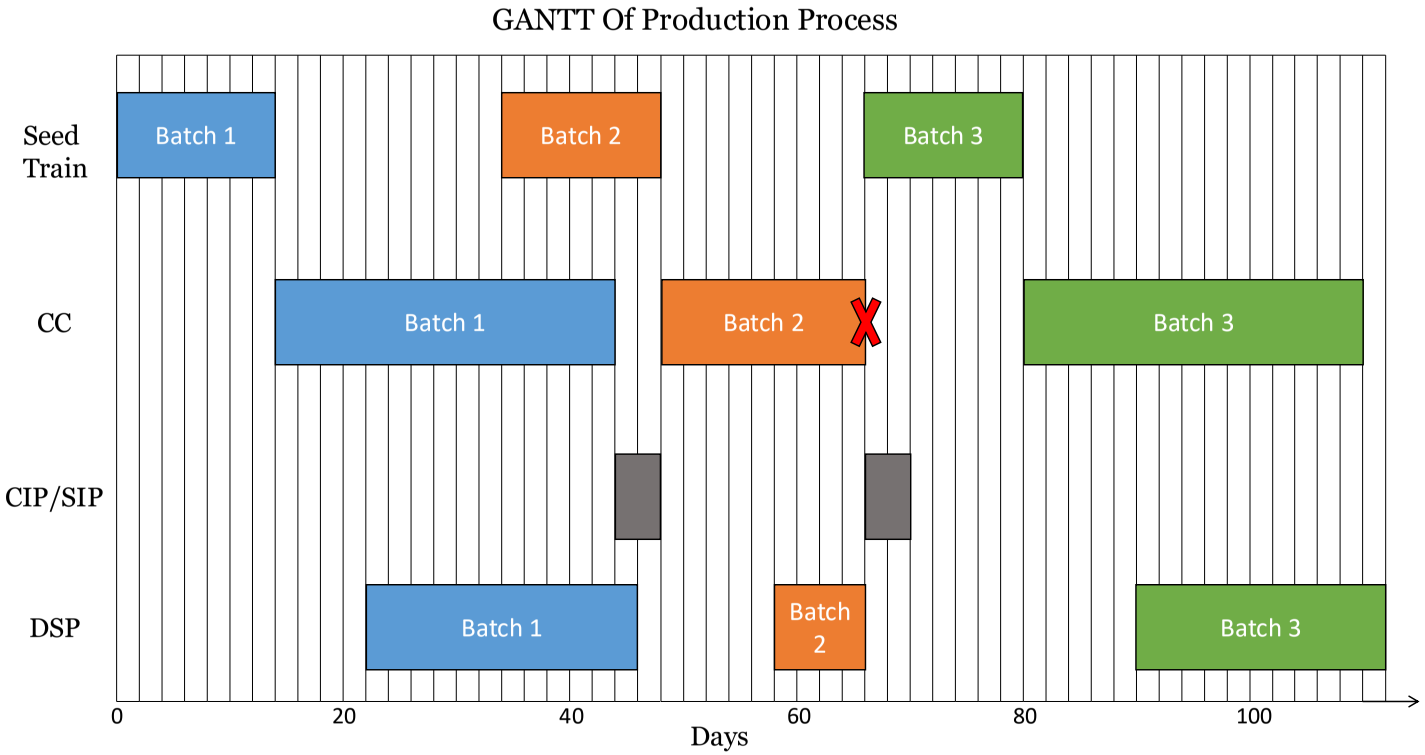


Figure 5.1: Mini GANTT of continuous process over three batches (and 112 days). The first and third batches run to completion without failure(s). However Batch 2 fails on Day 18 of its USP operation which triggers the start of Batch 3's seed train. CC = Cell Culture; CIP/SIP = Cleaning/Sterilisation-in-place; DSP = Downstream processing

seed train for the next batch (Batch 2) starts on the 21st day of the USP operation. In the case of failure (e.g., Batch 2, Day 66), the seed train is started immediately for the next batch. However this means that there is a greater period of downtime for the bioreactor (a maximum of 14 days) as compared to the cases where the process runs to completion without a failure. Given these considerations it is important to determine the optimal cell culture duration, in a capacity planning context, for a particular process given its process economics data, technical characteristics and commercial targets.

Further to previous work on integrated continuous bioprocessing, the aims of this chapter are to investigate the consequences to be considered when scheduling and/or capacity planning for these types of processes. The optimal scheduling and capacity planning strategy for these continuous processes that are prone to failure are to be investigated as previous models did not aim to optimise schedules or capacity plans.

Process configuration — i.e., the number of parallel bioreactors and the ratio between the number of reactors and the number of DSP trains — has an impact on the decision making. Given that all the parallel bioreactors share and are fed by the same seed train, the trade-off between a large single reactor and multiple smaller reactors with the same failure profiles is such that with more reactors the probability of achieving a low output is minimised. This is at the cost of also reducing the likelihood of achieving maximum output from the process. In addition, there is obviously increased probability of a failure event (because there are more reactors) although each failure event has a reduced magnitude.

Thirdly, there needs to be consideration of the decision-making required after a reactor fails in the context of multiple bioreactors or a more complex scenario of scheduling & capacity planning for a multi-product facility. In the case of the former, one decision is to let any reactors that haven't failed run to its planned end (e.g. if there are four reactors started at the same time and planned to last 60 days but one fails before that time, the other reactors are run until the 60 days are up or they all fail). Another decision is to stop the process early and restart it as soon as possible. The trade-off there is that with a *'normal' restart*, the remainder of the batch proceeds at an indeterminate lower productivity but with the *'early' restart* there is an inital period of zero productivity as the reactors restart (due to the ramp-up) but after that the productivity is back to the maximum for an indeterminate time period (there may be another failure).

This chapter attempts to factor scheduling and capacity planning concerns into this decision-making while dealing with failure-prone perfusion processes and the consequences of failure. First a model for perfusion processes that allows manufacturing schedules to be simulated is developed. It comprises a new stochastic simulation framework for evaluating operational decisions for a facility utilising perfusion bioprocesses, as well as optimisation algorithms which may be utilised to tune any scheduling strategy or policy as part of a larger *hyper-heuristic* framework. Second, based on a mAb production process with peak production capability of ca. 460 kg/annum, this chapter demonstrates that selecting the run time of the cell culture operation based on the expected process economics of a singular batch is inferior to decision-making that considers the expected annual demand or utilisation for a given facility and process. Third, strategies are developed and applied — inspired by the EPQ and inventory replenishment model(s) — to simultaneously optimise the selection of process run time and scheduling of batches. These are compared with a standard scheduling approach at different facility utilisations in a dynamic simulation environment. Finally, investigations on the impact of process configuration are undertaken with respect to determining an optimal run time for the cell culture process and any operational decisions that need to be taken after a cell culture failure event.

In Section 5.2 the proposed modelling framework is detailed. This includes the design of its components and their interactions as a hyper-heuristic. This is followed by an evaluation of the bioprocess and simulation models as part of an effort to ascertain the impact of failure rates and process configuration on optimum process run times. Section 5.4 on page 85 evaluates the hyper-heuristic on a simple scenario featuring a single-product facility by tuning two scheduling policies to dictate operational decisions in the face of uncertain demand. Finally a section summarising the work preceding it concludes this chapter.

5.2 Model Framework

The modelling framework proposed in this chapter is designed as a custom framework comprising: a model for the manufacturing bioprocesses; an object-oriented discrete-event model used to simulate the scheduling environment on the manufacturing facility in which the bioprocesses are operated; policies that dictate scheduling decisions; and optimisation algorithms to tune the parameters of the scheduling

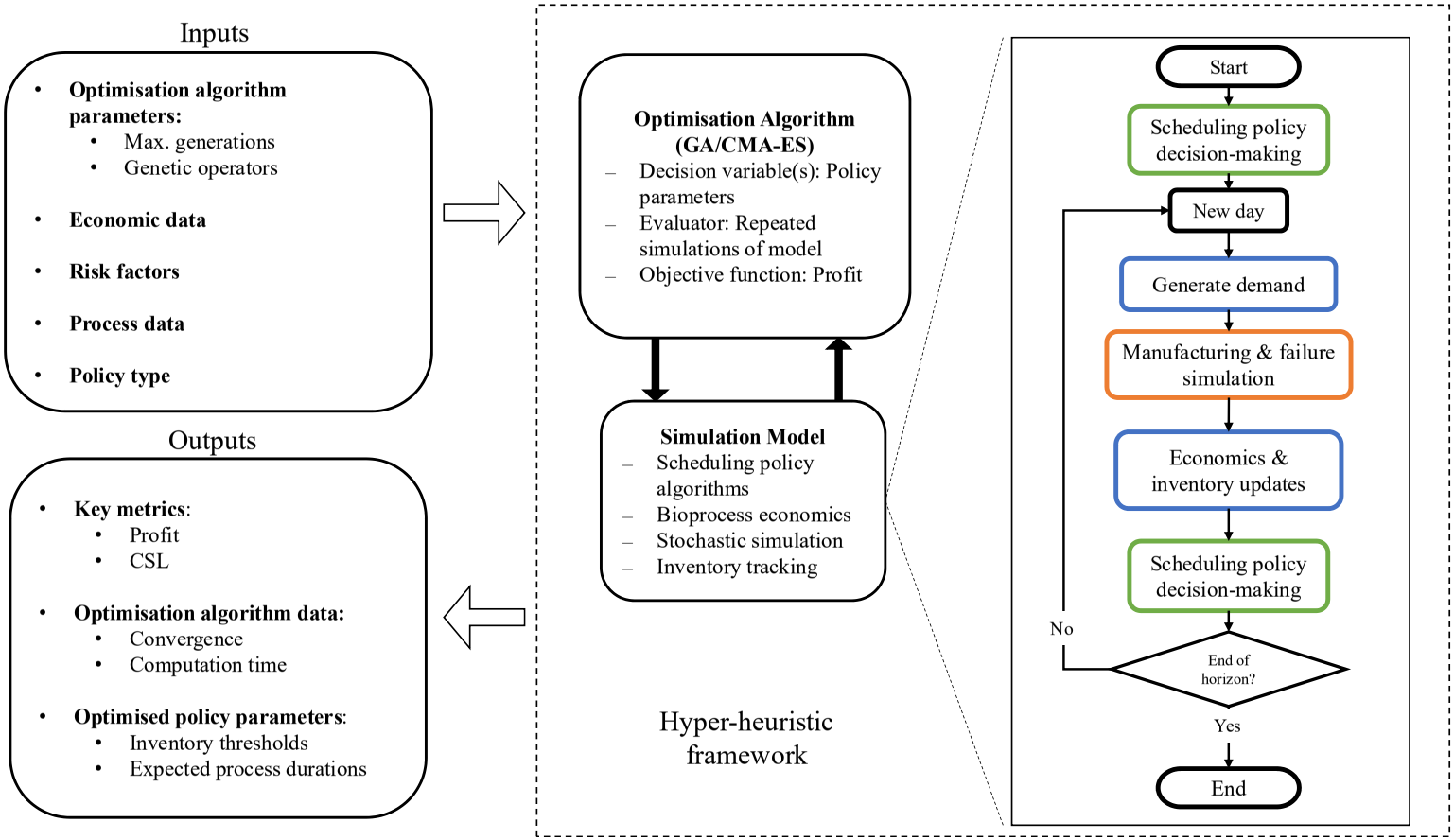


Figure 5.2: Overview of the hyper-heuristic, showing the interactions of its components and including a flow sheet of the simulation framework.

policies. Figure 5.2 on the preceding page gives an overview of this framework. Detailed descriptions of the components of the hyper-heuristic framework follow.

5.2.1 Bioprocess Model

The basis of the bioprocess used is a platform mAb manufacturing process utilising a perfusion bioreactor with an alternating tangential flow (ATF) filtration system for cell retention. Perfusion bioreactor systems equipped with an ATF filter have been shown to perform well in economic analyses compared to other cell-retention filter systems and do not suffer consequences as severe in the event of filter failure (Pollock et al., 2013b). A flow sheet of this process is shown in Figure 5.3. In general, the economic, operational, and technical data for the continuous process(es) used for the bioprocess model is adapted from Pollock (2013) and summarised in Appendix B.

For simplicity, the unit operations of the process can be grouped together into three main steps:

- A seed train which encompasses all cell thawing and expansion operations;
- USP which is just the cell culture; and
- DSP which accounts for all unit operations from the capture chromatography step (Protein A) to the final finish & polishing steps (UF/DF).

Processes may have more than one DSP train per bioreactor and so are not limited to a 1:1 USP:DSP train configuration. In the case that there are multiple bioreactors, all bioreactors are fed from the same seed train simultaneously. This means that all the bioreactors have to be started at the same time.

The seed train takes 14 days and then the production bioreactor can be innoculated. The ramp-up time to reach the desired cell density for harvests is ten days. In this period, no harvests are collected as the process has not yet reached steady-state. From day 11 onwards, daily harvests are collected from the bioreactor and then taken through DSP. Each DSP 'batch' then takes two days to be fully processed and product coming out of it then can be put in inventory, sold or otherwise delivered. Therefore, a process that has a cell culture run time of 60 days and ramp-up time of 10 days will produce 50 separate DSP batches. The turnaround on the bioreactor determining the earliest time it can be reused is four days and accounts for clean-in-place (CIP) and sterilization-in-place (SIP) operations and

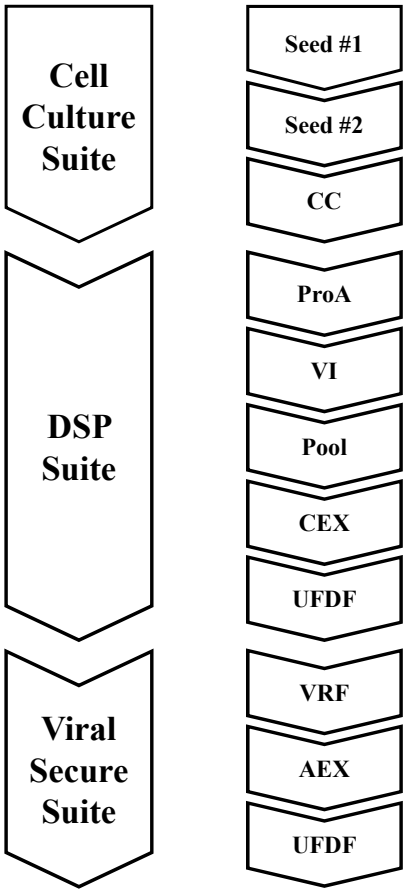


Figure 5.3: Process sequence and suite configuration for the model perfusion-based bioprocess. CC = cell culture, ProA = Protein A chromatography, VI = virus inactivation, Pool = daily perfusate volume pooling, CEX = cation exchange chromatography, UFDF = ultrafiltration/diafiltration, AEX = anion exchange chromatography, VRF = virus retention filtration. Adapted from Pollock et al. (2013b).

Table 5.1: Process scheduling parameters.

Parameters	Value	units
Seed train	14	*days*
Bioreactor turnaround	4	*days*
Ramp-up Time	10	*days*
DSP duration	2	*days*

other activities in preparation for a new cell-culture operation. As it is possible for parts of the seed train process to use a different suite, the earliest portion of the seed train of a product can take place concurrently with the latter days of the USP of a previous batch (of the same or a different product). A *threshold* can then be defined, in the case the facility is not idle, as the minimum time elapsed for a USP operation before a decision on starting a new batch (i.e., a new seed train) can be made. The seed-restart threshold is for starting a new batch of the same product and the changeover threshold is for starting a new batch of a different product. Process timing information is contained in Table 5.1.

The economics associated with this process are such that a *seed train cost* is attributed to every seed operation that is started, and a *cell culture setup cost* for the setup and prep activities that go into starting up each bioreactor in a batch. The *daily cell culture perfusion costs* are accrued for every day a bioreactor is in operation, and for every DSP batch commenced, *DSP batch cost* is accrued. A cost is associated with replacing a fouled *ATF filter*; this is also captured in the batch setup costs as a new ATF filter is needed for each one. Finally, if the process is idle for more than the *setup expiry period*, there is a cost of re-establishing sterile and clean holds for all equipment before another batch/campaign is started. This is referred to as *changeover costs* and also captures the costs of setting up the facility when changeover to manufacturing another product occurs.

To capture the stochastic failure events and the consequences, previous data adapted from Pollock et al. (2013b) and Pollock (2013) was used and this is presented in Table 5.2 on the next page. Those studies used a fixed perfusion duration of 60 days so the probability of ATF culture contamination and ATF filter failure events were within the 60 days. As this study is looking at various process durations, this requires some adjusting. Ideally, the rate of failure should be low in the early stages of the process and be relatively high towards the end. The process for Pollock (2013) choosing a 6% failure rate for the cell culture contamination

Table 5.2: Process failure events, consequences, and the associated risk (adapted from Pollock et al. (2013b).

Process Event	p(Failure)	Consequence
ATF culture contamination	6%	Batch loss & discard two pooled perfusate volumes
ATF filter failure	2%	Replace filter & discard next 24 hours of perfusate

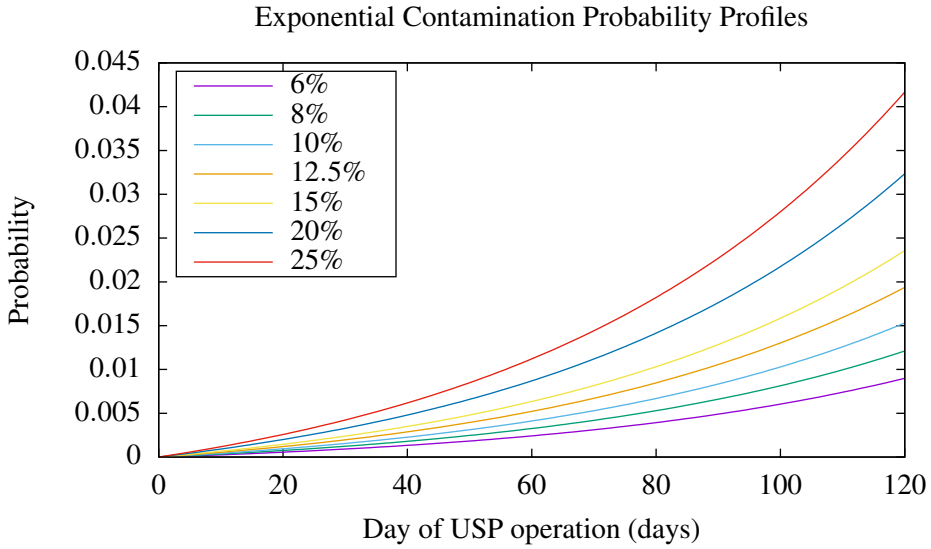


Figure 5.4: The probability distribution functions illustrating the probability of a failure event occurring on a particular day of the USP operation for different rates, $6-25\%$. These failure rates are defined as the probability of failure event occurring within the first 60 days.

is that each addition to (or sample from) the bioreactor has a 1 in 1000 chance of introducing contamination to the system — the 60 day batch had approximately sixty such additions leading to the 6% chance of failure. This has the result that the chance of failure on any specific day is independent of how far along in the process it currently is.

It is reasoned that not only additions can cause contamination, but also equipment that wears or stresses over the course of the process (such as tubing, gaskets, valves, O-rings, filters, seals or connectors). For the equipment failure due to filter fouling, however, Pollock chooses a probability of 2% and then weights failure to occur at latter stages of the cell culture.

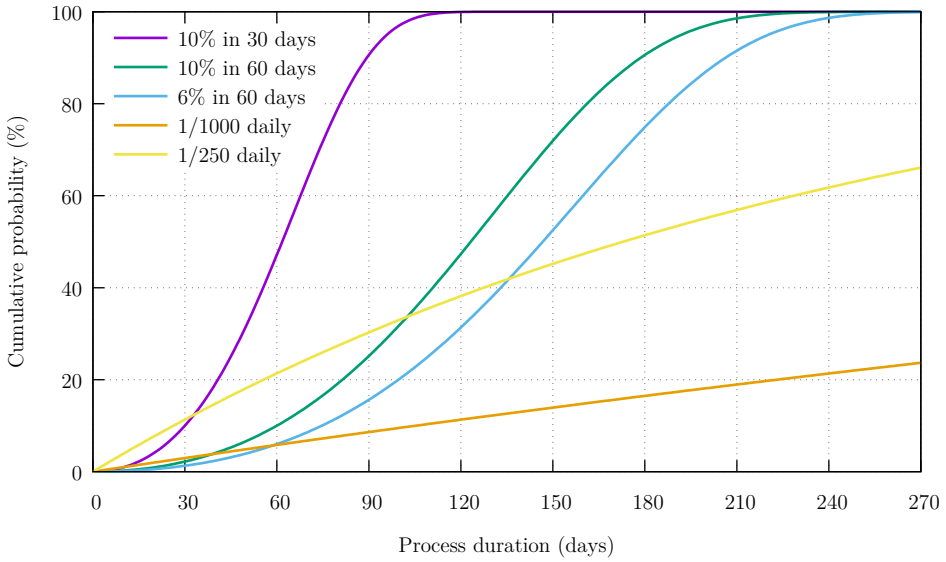


Figure 5.5: The cumulative probabilities showing how likely the USP operation of a certain duration will fail based on different failure rates.

To this end an exponential function of the form shown below is used to describe the probability of a failure event occurring $P(x)$, on a specific day x:

$$P(x) = \frac{\exp(x/a) - 1}{b} \quad . \tag{5.1}$$

Here, a is benchmarked to 60 and also represents the amount of time it takes (in days) for the probability of failure to increase by a factor of e; and b is a scaling constant

The significance of the difference in this assumption is illustrated in Figure 5.5 which shows the cumulative probability of a cell culture contamination event occurring within the duration of a process. For profiles where the daily absolute risk of failure is constant, the cumulative probability of failure in the early parts of the process can be significantly larger than profiles modelled from an exponential function.

Mainly seven failure rates were investigated for culture contamination: 6%, 8%, 10%, 12.5%, 15%, 20%, and 20%. These are defined as the probability of failure event occurring within the first 60 days unless otherwise specified. Some additional failure rates were also used in experiments: 10% failure within 45 and 30 days. In addition, a 2% rate is used for the ATF filter failure. The probability

distribution functions are shown in Figure 5.4 on page 70.

5.2.2 Discrete-Event Simulation Framework

Based on the bioprocess model discussed in Section 5.2.1, a custom discrete-event simulation model was developed in Java™ (Oracle Corp., Redwood Shores, CA, USA). Other software for simulating bioprocesses exist. For example, BioSolve Process (Biopharm Services, Chesham, UK) is an Excel-based software package that enables detailed cost analysis and scheduling for a single batch. INOSIM (INOSIM Software GmbH, Dortmund, Germany) also allows for process design and optimisation at the process level. Both of these software packages are not suitable for high-level planning and scheduling. Commercial discrete-event simulation software such as ExtendSim (ImagineThat! Inc, San Jose, USA) could have been a candidate as it allows for detailed simulation models to be designed and stochastic events implemented, even if it is not generally used for capacity planning.

However, the choice was made for a custom discrete-event simulation model so that it could be better integrated with the overall hyper-heuristic framework (and the optimisation algorithms). These would all be developed in the same programming language (i.e., Java). In addition, using these external programs for evaluating solutions would add a considerable computation expense as they are quite complex. It seemed reasonable to design a custom model that was detailed enough for the purposes of this research without being a large drain on computation resources.

The model simulated the processing of batches on a facility as a multi-stage process comprising of a seed train, USP, and DSP. In addition, based on the state of the facility (i.e., what it is currently manufacturing), inventory levels, and stochastic events, it evaluates the economics of operational decisions and reports key metrics, inventory profiles, and the facility schedule of the given time horizon. Finally, it allows the use of dynamic scheduling rules or policies to make operational decisions based on the state of the simulation so these policies can react to any changes.
On each day:

- any new activities or operations are started if required whilst any existing ones have their durations advanced by a day;
- any manufacturing takes place and the random variables (such as demand, process failure events, or yield) are realised by sampling the associated probability distributions once each;

- the process economics related to the model are evaluated;
- scheduling decisions are made if necessary; and
- activities or operations are brought to an end at their target run time (or terminated early due to equipment failure).

The timing of scheduling decisions and the set of decisions available to make are determined by the specific type of scheduling policy or strategy (see Section 5.2.3). Decisions made on one day are implemented and take effect the following day.

In the event a decision is made to start a batch and/or campaign, a seed train operation is created and started up. When that is completed, it triggers the start of a USP operation which proceeds until it reaches its natural/intended end — specified by the process run time — or is terminated because of a process failure event. During the course of USP but past its ramp-up, the daily harvests trigger separate DSP operations which deposit product in the inventory when completed. Over the course of the simulation, these operations are recorded in a local history. This is so that at the end of the simulation, a facility schedule can be generated describing the history of operational decisions taken, the workload of the facility, and identifying any batches terminated early due to contamination. Similarly, the inventory levels for each product are recorded at each time point so a decision-maker can evaluate the impact of scheduling decisions on inventory levels.

Depending on the assumptions made for the purposes of the simulation, demand constraints may be set to be periodic (yearly or monthly), or set to be continuous (i.e., daily). According to the set demand frequency, any available product in inventory is used to satisfy the demand for that period. If the inventory is not sufficient, unfulfilled demand is added to backlog on which a decay function may be applied. Also, any product that has exceeded its shelf-life is deleted from inventory and discarded.

5.2.3 Scheduling Strategies

The scheduling strategies employed in a dynamic simulation environment are based on control policies. They initiate new production orders (batches in this case) based on current inventory levels as well as the state of the facility (i.e., the product currently being manufactured), m, in a make-to-stock fashion. In addition, these policies may have parameters that determine the run times of the processes.

As previously mentioned, when decision-making takes place, the set of available decisions is determined by the type of scheduling policy or strategy implemented. Generally, simpler policies make decisions at the end of a batch — either because it has suffered process failure or because it has reached its predetermined run time — or if the facility is idle. Specifically, because the seed train for a subsequent batch can be started before the completion of the current batch, the decision epoch begins at the seed-restart threshold or the changeover threshold. These are determined respectively by: the bioreactor turnaround time, and the changeover time between products. As a result of the lag between the point when a decision is required and the end of the current batch, policies will need to (implicitly or explicitly) take into account the expected extra product that would be produced by the end of the current batch (if the facility is not idle). In cases of those simpler policies, the set of decisions is usually whether to start a new batch of the current product, or to changeover to another product (in a multi-product scenario), or to keep the facility idle. A more flexible policy would allow decision-making at any point in the horizon. Specifically, everyday it would be able to cut short a batch currently in production to either switch to a new product (in a multi-product scenario), or to make the facility idle, or simply to start a new batch of the product currently being produced.

5.2.4 Optimisation Algorithms

In the case that the scheduling strategies or policies required tuning, evolutionary algorithms (EAs) were used to optimise their parameters. These EAs include a genetic algorithm (GA) and a Covariance Matrix Adaptation Evolution Strategy (CMA-ES) implemented using the ECJ Library (Version 24) in Java™ (Luke, 1998). The representation used for the chromosomes of the tuned policies was tailored to each specific type of policy to enable efficient search procedure. As a result, the EA parameters will be discussed in tandem with the specific policies utilised.

5.3 Evaluating the Bioprocess and Simulation Model

In this section the results from experiments on determining optimal processing run times are discussed. First the economics of a singular batch are considered, followed by separate analyses of the effect of demand targets, capacity constraints

and process configuration on the optimal cell culture run time.

5.3.1 Optimal Processing Run Time

5.3.1.1 Optimal run time of a typical batch without demand constraints

Given the costs for an ATF filter, cell culture setup costs, daily cell culture perfusion batch costs, DSP batch costs, costs for the seed train, and process yields (all in relative monetary units (RMU)), the simulation could be used to assess the expected performance — the Cost of Goods (CoG) — of each batch length for each failure rate . Each failure rate, $P(x)$, is the probability the batch will fail within the first 60 days. The failure rate used for the ATF units was 2%. In addition, a *batch termination penalty* is incorporated which was applied if the batch suffered a cell culture contamination and was subsequently aborted. This is because when analysing the expected performance of a single batch in isolation, the lost production time in a manufacturing campaign discussed earlier and illustrated in Figure 5.1 on page 63 cannot be directly modelled (because there are no subsequent batches to delay). So this batch termination penalty is used as a surrogate for this scheduling inefficiency and is heuristically set to be roughly equal to the observed worst case CoG/g (ca. 20 RMU/kg) multiplied by the time that would be lost in a scheduling scenario in the worst case (i.e., 10 days). The values for the process economics parameters are listed in Table 5.3 on the next page.

Optimal process run time at different failure rates The simulation environment was set up to start a seed train (at the beginning of the simulation horizon) which eventually triggered a USP operation. This operation was run until it ended either by reaching its target duration or termination due to process failure. The simulation environment would then be terminated after the completion of the last DSP operation. The process economic data and process failure statistics would be captured as well as how much product was made. This procedure was replicated 10,000 times to evaluate the metrics for each process run time.

From this it is possible to ascertain the most cost-effective process duration in isolation by evaluating the CoG/g. In this scenario, CoG/g is calculated by dividing the amount of product manufactured by the costs associated with operating the process. This is illustrated in Figure 5.6 which shows the trend of CoG/g decreasing to an optimal point, increasing after it and then plateauing with increasing process duration. At very short process durations, the ratio of costs to material produced

Table 5.3: Process economics parameters. Only the first seven parameters are used for evaluating the Cost of Goods (CoG) in Figure 5.6 and Table 5.4. The remaining parameters are used in the experiments that take into account demand.

Parameters	Value	Units
Replacement ATF filter cost	16	RMU
Cell culture setup cost	29	RMU
Daily cell culture perfusion cost	3	RMU/day
DSP batch cost	12	RMU
Seed train cost	5	RMU
Batch termination penalty cost	200	RMU
Process yield	1.55	kg/DSP batch
Sales price	100	RMU/kg
Inventory cost	0.01	RMU/kg/day
Waste cost	5	RMU/kg
Changeover cost	35	RMU
Backlog decay	0.25	
Backlog decay period	1	year
Backlog penalty	0.1	RMU/kg/day
Setup expiry period	30	days
Shelf-life	720	days

is heavily dominated by the fixed costs involved in operating a batch, hence the high CoG/g. Batch failures are unlikely at these process durations. As the process duration is increased, so does the amount produced which offsets those fixed costs and leads to lower CoG/g. However, an increasing likelihood of batch failure and associated penalty cost tempers this decrease in CoG/g up to the optimum duration where the trend reverses and an increasing CoG/g is observed. The CoG/g value stabilises and remains more or less the same at very long process durations where the probability of batch failure is practically 100%. In addition, one can see that the optimum decreases with increasing failure rates and also that the CoG/g in general increases with more failure-prone processes.

Sensitivity analysis The process run time with the lowest CoG/g for each failure rate[2] is shown as well as results of a $\pm 50\%$ sensitivity analysis in Table 5.4 on the following page.

Looking at the base case, intuitive results are observed where the optimum

[2]The values for the optimum process run time are the average of the process lengths that have CoG/g values within the standard error of the lowest CoG/g value.

Table 5.4: Sensitivity analysis showing the optimal process duration at base parameter values and how this is affected by ±50% variation in the parameter values. The top row for each parameter shows change in the optimal process duration in the −50% case while the lower row is for the +50% case. A green highlight represents an increase in the optimal process duration whilst a red highlight represents a reduction; darker colours indicate a greater difference in the optimum.

	Cell culture contamination/failure rates (%)						
	6	8	10	12.5	15	20	25
Base case	97.9	89.1	85.5	80.4	76	68	61
Replacement ATF Filter	0.5	0.9	0.5	0.5	0	-0.5	0
	-0.5	0.4	0	-0.5	0	-1	0
Cell culture setup cost	-8.4	-4.1	-6.6	-5.8	-5.5	-7.5	-4
	6.1	6.4	4	4.1	5.5	6.9	5.7
Daily cell culture perfusion cost	-8.4	-4.6	-5.5	-6.4	-5.5	-7.5	-4
	5.6	5.9	3.5	4.6	5.5	6.9	5.7
DSP batch cost	1.2	0.8	0.2	0.4	1.4	0.1	1.3
	-0.9	1.9	-2	-1.4	-1	0.5	0.5
Seed train cost	-1.8	-0.8	0	-1	0	-1	0
	0.5	1.4	0.5	0.5	0.5	2	0.5
Bioreactor Failure Penalty Cost	19.6	18.9	16	13.6	14.5	17	17.5
	-12.9	-6.5	-9.4	-8.5	-10.4	-10.2	-6

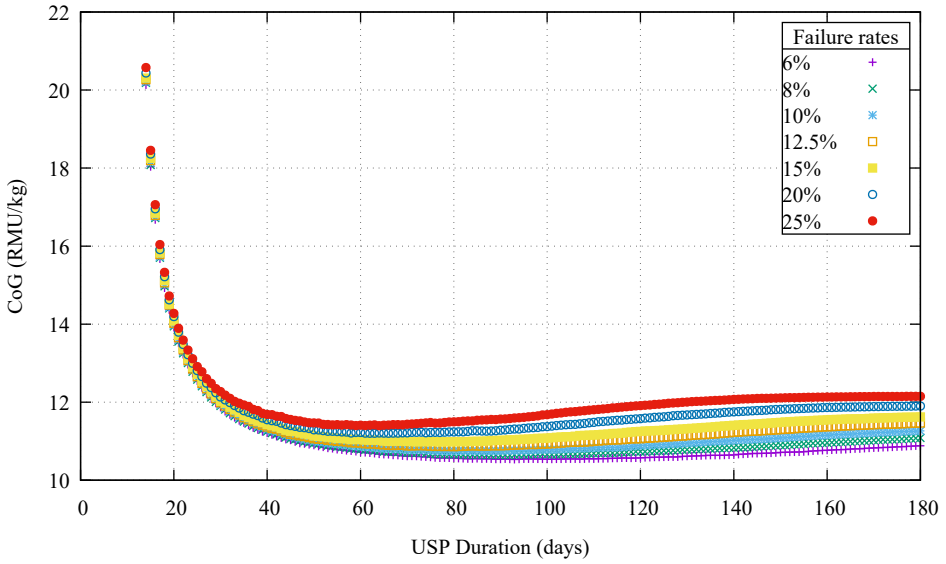


Figure 5.6: Expected CoG/kg for differing process run times and contamination rates

cell culture duration gets smaller with increasing probability of failure — from an average of 97.9 days at a 6% failure rate to 85.5 days at 10% failure rate and the extreme of 61 days when the probability of failure is one-in-four within the first 60 days of the process.

Table 5.4 also shows that the optimal batch duration was not affected by the cost of a replacement ATF filter and was also barely sensitive to variations in the cost of the seed train. The cost of the replacement ATF filter has no impact because it is only applied in the event of a filter failure (which does not terminate the process) and is not very likely[3]. Similarly, the cost of the DSP batches does not affect the optimum batch durations when varied. The daily cell culture perfusion and cell culture setup costs have a moderate positive effect on the optimum when compared to the bioreactor failure penalty cost which has the greatest but opposite effect on process duration. That is to say, a decrease in the cell culture setup cost or daily perfusion cost reduces the optimum process duration while an increase would make it longer while the inverse relationship applies to the failure penalty cost.

[3]Although the cost of an ATF filter is used in deriving the value of the cell culture setup cost, in the sensitivity analysis the knock-on effect the ATF filter would have on the cell culture setup cost — and consequently the optimal process duration — is not considered. Instead, for ease and clarity in determining dependencies, each parameter is treated independently of the others in this analysis. Otherwise, increasing the cost of an AFT filter would increase the optimum process duration.

One can explain these trends by considering these parameters as either depending on how much product is manufactured or independent of the output. For example, the cell culture setup is applied regardless of how much is produced, so the higher its value, the more product has to be manufactured (and consequently the longer the process duration has to be) to offset this cost. Conversely the total DSP cost is more or less a function of how much is manufactured. One of the consequences of process contamination is that product is discarded. This discarded product may have been in the middle of DSP processing and thus accrue DSP costs regardless. A larger DSP batch cost exacerbates this inefficiency and so a shorter batch would be better as the chance of process contamination is lower. Clearly, this particular effect was not large. These effects apply to the cell culture perfusion cost because it is applied during the ramp-up period as well as during the productive period of the process; in this scenario the former effect is more significant. The effect of the bioreactor penalty cost is more obvious — if there is a larger direct penalty for process failure, longer processes that are more likely to fail become more costly.

Although the differences in the optimum process duration are reduced when comparing the 6% and 25% failure rates, there does not seem to be an overall trend when considering all the other failure rates examined. In fact, these differences are not strictly monotonic (as a function of changing failure rates) even as the reported optimum durations are. This observed behaviour is due to the simulation noise and the shape of the CoG/g curves which means that the average of the process lengths within the standard error of the minimum CoG/g value can be skewed. With increasingly failure-prone processes, there may be a limit to how much the parameters can affect the optimum process duration. If this is true, it would be at a failure rate that is higher than those considered here and so would be unrealistic in a real-world scenario.

5.3.1.2 Optimal duration with demand targets or time constraints

In comparison, when examining a scenario that takes into account demand requirements over the course of years, or one where there are capacity constraints on the manufacturing facility (or both) the optimal process duration changes. The difference between this and previous analysis is that here, sequences of batches (rather than isolated batches) are examined.

Another way of describing this difference is that in the previous analysis, for

each process run time that is tested, the facility is assumed to have *infinite time* available to schedule a fixed number of batches, after which the average economic statistics of the average batch is analysed. In the following analysis, however, for each process run time tested, this facility has a finite *fixed time* available so as many batches as possible need to be manufactured in that time frame to either maximise production output or meet a demand target. Hence, the consequence of process failure is captured as a loss in available production time instead of relying on an explicit failure penalty cost[4].

Time constraints This is first illustrated with time constraints by Figure 5.7 on the next page which shows the expected throughput for each process run time if the facility was run at maximum utility — i.e., the objective was to maximise the amount of product manufactured within a time horizon of five years. The trend observed for all failure rates is such that with increasing process duration, the productivity increases until it peaks, goes down a little and then plateaus shortly afterwards. This is because at one extreme, if the process is very long, it will always fail well before it reaches its 'threshold' where the seed train for the next batch is started. So there is always wasted time equal to the difference between the seed train duration and turnaround time. As the process duration gets smaller (towards the optimum) the failure will tend to happen after its threshold but before the scheduled run time. For example, if a 130-day process will fail on its 127^{th} day (wasting the last three days), it is more productive to have scheduled a 129-day process as there is only two days wasted, which is less productive than a 128-day process, and a 127-day process would be better still. A 126-day process would be best as the 127-day process would require previous days' harvests to be discarded making it as productive as a 125-day process. Processes shorter than 126 days would be increasingly less productive.

From this figure, the maximum throughput achievable by the facility is determined to be 456 kg when the process duration is 126 days at a failure rate of 10% within 60 days. This means that the maximum fraction of productive days is 81.9% in that time horizon. Comparing with the longest batch possible of 180 days and a perfect process (i.e., there is no failure) the facility has a peak productivity of about 89% of its capacity. The figure also shows the profiles for 10% failure within 30 and 45 days where the maximum throughputs are 378 kg if the process duration

[4]Therefore, bioreactor failure penalty cost is not used in cost calculations from this point onward.

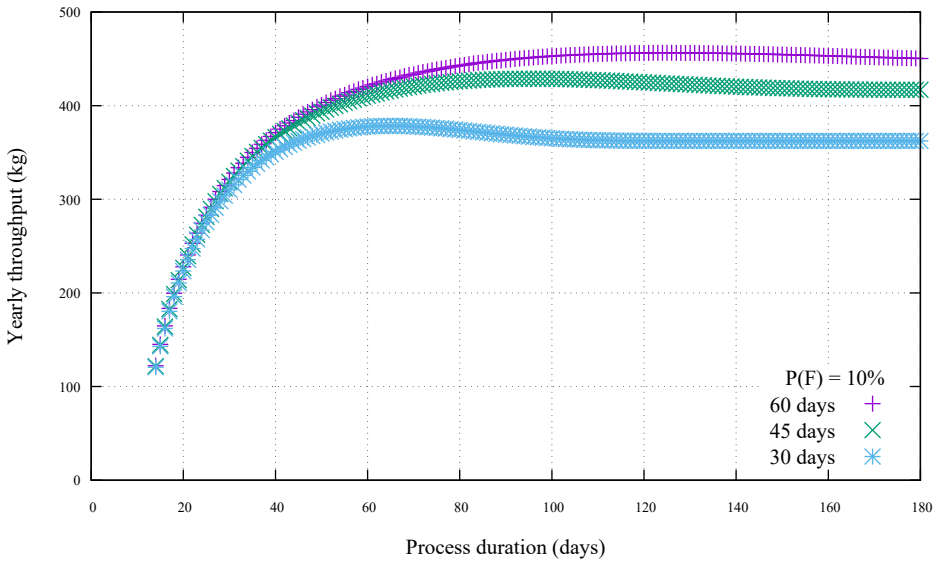


Figure 5.7: Productivity of processes with different cell culture failure rates: 10% within 30, 45, and 60 days.

is 64 days, and 429 kg at 96 days respectively. Compared to the previous analysis, this suggests that at a 10% failure rate within 60 days, the most cost-effective processes (those minimising costs) are shorter than the most productive processes (those maximising production) — i.e., ca. 85 days vs. 126 days).

Demand targets To demonstrate this also with demand targets, a simulation of a facility with one product over an extended time horizon (fifty years in this case) and yearly deterministic demand (i.e., demand is delivered from inventory once each year at the end of the year) was carried out.

To determine when the campaigns to meet the demand each year were started, a simple heuristic was implemented. This heuristic starts campaigns at the latest possible date (for a perfect process) so that enough product is made to meet the demand at the end of the year whilst taking into account any product already in inventory and the expected production of any batches that are currently in progress. A consequence of the demand being delivered at the end of each year and the shelf-life of inventory being measured in days is that during a year, a lot of product may expire before the end of the year if it was not delivered at the end of the previous year. This means that there would be material in inventory accruing storage costs that would never be used. To deal with this, at the end of each year, material

Table 5.5: The optimal process durations at various yearly demand targets and 10% contamination rate for different objectives: maximising profit and minimising CoG/g. The values for the optimum process length are the average of the process lengths that have profit or CoG/g values within the standard error of the highest profit or lowest CoG/g value respectively.

	Best process run time (days)	
Demand (kg)	Profit	CoG/g
30	49	49
75	42	43
155	67	63.5
230	94	64
310	84	81.5
455	119.5	120

that will expire before the next delivery date is thrown away to avoid unnecessary storage costs.

In a real-life bioprocess scenario, facilities and the processes they house are designed to maximise utilisation. This means it is unrealistic and inefficient to produce a small fraction of the facility's maximum output. However, if it is assumed that the different demand targets represent different products that can be manufactured in a multi-product facility, a ball-park estimate of an efficient choice when deciding the processing length for such products can be determined.

Different processing lengths for the given parameters in Table 5.3 were simulated and performance captured at various yearly deterministic demand targets. The demand targets investigated were 30, 75, 155, 230, 310, 465 kg per annum and the cell culture contamination rate was 10% (per first 60 days) while the ATF fouling rate was 2%. The processing lengths with the best profit and CoG/g performances are listed in Table 5.5. Profit is calculated as the difference between revenues (product of sales price and amount sold) and the total costs which is sum of inventory costs, waste costs, and manufacturing costs (replacement ATF filters, cell culture setup, daily cell culture perfusion, DSP batches, seed train costs). CoG/g, meanwhile is calculated as the total costs divided by the amount of product sold.

From these results, it is observed that when considering demand targets, the optimal process duration is not necessarily the same as when demand is not taken into account. Although the trend shows that the best process duration when judged based on the overall profit and the CoG/g is going up with increasing demand, at

various points there appears to be some interaction of different factors causing the optimal process duration based on either performance measure to fluctuate. This is partly due to a lot-sizing trade-off between long and short process durations. With a long(er) batch, setup costs are minimised and enough is made for each year without needing to store much for the next year or that is wasted. However it is more susceptible to failure leading to uncompleted demand. On the other hand, multiple (i.e., more than one) shorter batches are less likely to fail and miss demand but incur greater setup costs. This explains the drop in optimal process duration from 30 kg to 75 kg and again from 230 kg to 310 kg with respect to profit. In addition, at 230 kg, the optimal process durations with respect to profit and CoG/g are significantly different.

The best process duration for a minimal CoG/g may be significantly different from that which gives maximum profit. This is because the CoG/g measures the ratio of total costs to the amount of product made (or in this case, sold) while profit is simply the difference between revenues and total costs. A low CoG/g may indicate an 'efficient' process but does not reveal the whole story as it is also possible that not enough product was sold. For this reason, it is better to use profit as a metric rather than relying solely on the CoG/g measure. Finally, given these assumptions and process, really short batches (less than three to four weeks) are always sub-optimal.

5.3.2 Process Configurations and Multiple Bioreactors

In investigating the impact of multiple reactors in determining an optimal runtime for the cell culture process, process configurations with one to four parallel reactors were investigated. The processes that utilised multiple parallel reactors were designed such that they were comparable to the scenario with one reactor. This means that absent failure, the productivity of each configuration would be identical. So taking the single reactor case as a basis, if that reactor can produce a daily yield of Y, a process configuration comprising n reactors would have a daily yield per reactor of Y/n. The parallel reactors all share the same seed train, i.e., they are innoculated from the same seed source so they always start at the same time. In addition, the harvest from all reactors are pooled together and fed into one single downstream train so DSP is identical in all process configurations. Table 5.6 con-

Table 5.6: Process economics parameters for multiple parallel bioreactor scenarios in RMU. The replacement ATF filter cost is per bioreactor in the configuration while the other costs represent the entire process configuration.

	Number of parallel reactors		
	2	3	4
Seed costs	5	5	5
Cell culture setup costs	52	68	54
Daily cell culture perfusion costs	3	3	3
Replacement ATF filter	16	16	7
DSP batch costs	12	12	12

tains the process economics parameters for the three new process configurations[5].

Firstly, the performance and productivity of the multi-bioreactor configurations was assessed and compared with the base case of a singular bioreactor. To do this, a five year horizon was simulated for each configuration without any demand or shelf-life constraints and recorded the total inventory at the end of the period. The purpose of this was to measure how much can be manufactured in each design scenario. This simulation was carried out over 10,000 replications and the mean amount of product manufactured was captured for each process duration from 14 to 180 days inclusive at a 10% cell culture contamination rate and a 2% ATF fouling rate.

In event of a cell culture contamination event, two different strategies were employed. In the normal case, the next seed train would only be started at the predetermined threshold point unless all the bioreactors had become contaminated and failed — a **normal seed restart**. The second strategy/response was such that if a bioreactor became contaminated and failed, a new seed would be started immediately and the current cell culture operation would be terminated early so as to turn around the bioreactors and prepare them to receive the innoculum from this new seed train. This is referred to as an **immediate** or **early seed restart**. The results from these are shown in Figure 5.8 on page 86.

This shows that a process configuration that utilises only one bioreactor is able to manufacture more than the other process configurations that comprise multiple

[5]The cell culture setup costs increase in a non-monotonic manner because the single use bioreactors (SUBs) come in discrete set sizes of 2000, 1000, and 500 litres which cost about 9.8, 8.3, and 5.5 RMU respectively. The one-reactor process needs just one 2000L SUB; the two-reactor and three-reactor processes require identical 1000L SUBs; the process with four parallel reactors needs its SUBs to be only 500L each.

bioreactors. In fact, a trend can be observed such that with increasing number of reactors, the maximum output of the process reduces across all process durations. In addition, when comparing the normal seed restart decision to the early or immediate seed restart decision, it is clear to see that the latter strategy is superior to the other. This indicates that the trade-off does not favour a prolonged period of indeterminate low(er) production as compared to an interval with no production followed by an indeterminate period of maximum production.

Some of the insights from Figure 5.8 are made clearer in Figure 5.9. For instance, one can see that the gap between the multiple reactors and a single reactor widens with increasing process duration. With the normal seed restart this trend holds true for the entire scope of process durations examined. However, when the seed is restarted immediately upon a failure, this is only true until a process duration of about 130 days at which point the trend reverses and the difference in productivity begins to narrow. This is because at higher process durations, the single reactor process starts to decline in productivity whilst the productivity of configurations with multiple reactors merely plateaus.

In addition, the marginal difference in productivity reduces with increasing reactors is observed. That is to say that, at any given process duration, the difference in productivity between one reactor and two reactor configurations is larger than that between two reactors and three reactors which is larger than between three reactors and four. This is evident with normal seed restart decisions and more muted with immediate restart decisions. This observation may be useful in the case of facility retrofitting if there is a decision to make about adding more bioreactors to the USP suite(s) in order to increase the scale or capacity of the facility.

5.4 Single-Product Scheduling Strategies

Here, control strategies for a single-product facility are introduced and evaluated. These strategies are based on simple stock replenishment policies that determine when to order a new batch (by starting a new seed train) and how much to order (by setting the process run time).

5.4.1 The (s, B) Inventory Policy

One of the approaches to dealing with stochastic uncertain demand is by trying to sustain strategic inventory levels at a certain amount. In other words, some safety

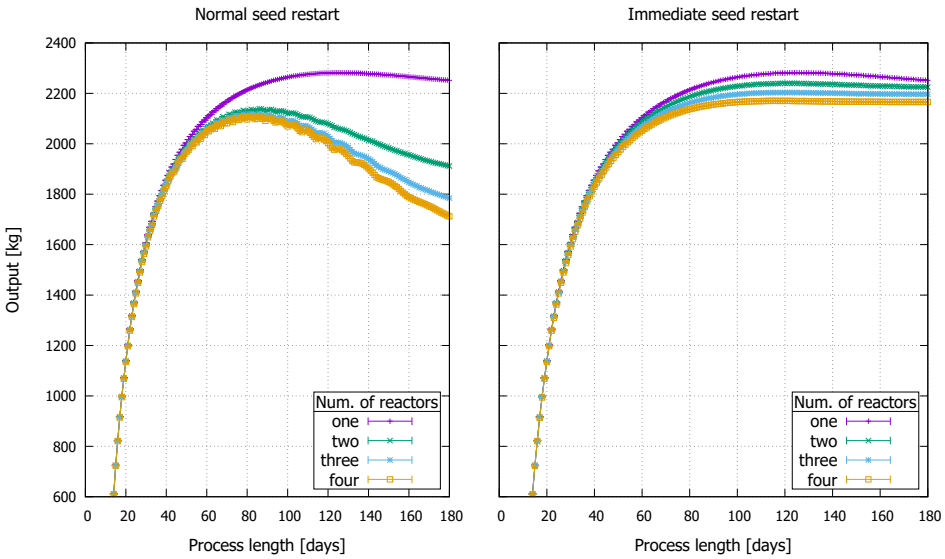


Figure 5.8: Productivity of USP configurations with different failure responses.

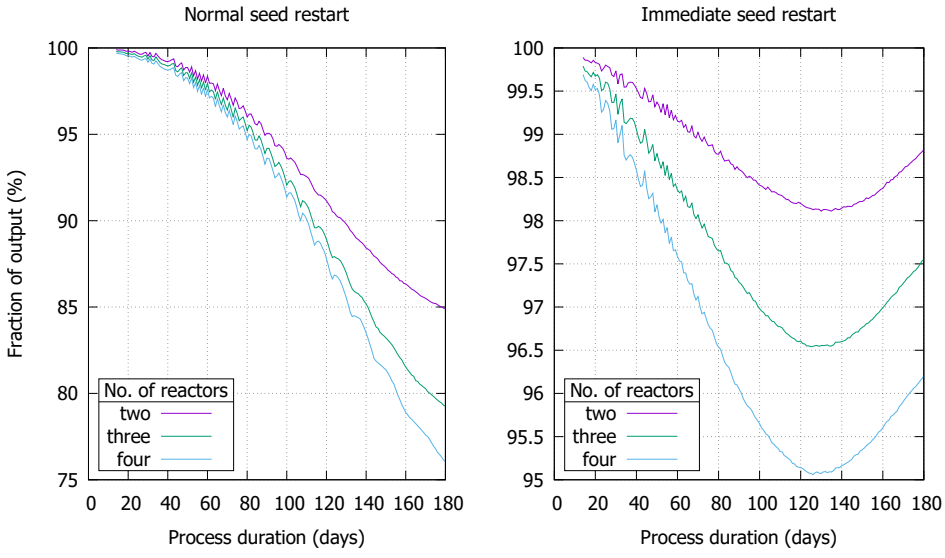


Figure 5.9: Difference in productivity of USP configurations with different failure responses.

stock is maintained as a buffer to mitigate fluctuations in actualised demand, and in this case, failure-prone processes.

In this case two decision variables are defined:

1. The re-order level, s
2. The size of batches in the campaign, B

The logic for determining when to start a new seed train is detailed in Algorithm 2. This essentially says that each day the inventory is compared with the re-order level and will attempt to start a new seed train to start the following day if the amount in inventory is less than that. Obviously, a new seed train is not started if one already exists in operation, or if there is an existing USP process that has not been in operation for as long as the threshold time.

Algorithm 2 Pseudocode of the (s, B) strategy.

procedure NEW SEED TRAIN(current time t)
    ST_t = the existing seed train at time t
    USP_t = the existing USP operation at time t
    T^{ST} = the seed train run time
    T^{USP} = the USP run time
    T_t^{USP} = time elapsed in USP operation at time t
    T^{BT} = time required to turnaround bioreactor
    T^{THD} = threshold
    $T^{\text{THD}} = T^{\text{USP}} + T^{BT} - T^{ST}$
    if $ST_t == \{\}$ **then**
        I_t = the inventory level at time t
        if $\text{USP}_t == \{\}$ **OR** $\left(\text{USP}_t \neq \{\}\right.$ **AND** $\left. T_t^{\text{USP}} \geq T^{\text{THD}}\right)$ **then**
            if $I_t \leq s$ **then**
                Start seed train, ST_{t+1}
            end if
        end if
    end if
end procedure

With this policy, s has to be sufficiently high so that stock-outs do not occur but must not be too high such that large inventory holding costs are not accrued. This must be balanced with an optimal B which if too small would mean more batches would be needed and lead to an increase in setup costs.

5.4.2 The (s_1,s_2,B) Inventory Policy

The second inventory policy was designed to be a potential improvement on the (s, B) policy by including a third parameter. All three decision variables are identified below:

1. The re-order point, s_1
2. The size of batches in the campaign, B
3. The campaign re-order point, s_2

The significance of the campaign re-order point, s_2, is such that it (instead of s_1) is compared with inventory in the case where a potential new batch would be part of the current campaign (i.e., the time between the USP of the latest batch and the USP of the new batch does not exceed the setup expiry period) to determine if a new seed train should be started. The reasoning for this policy especially in contrast to the (s, B) policy is that the new parameter s_2 is an attempt to allow the optimisation to choose values for the policy parameters so that batches can be linked together or manufactured in close proximity to each other in order to minimise changeover costs that apply if there is a sufficiently long idle time between two consecutive batches.

These three decision variables were optimised using a GA for the objective of maximising expected profit. The logic for determining when to start a new seed train is detailed in Algorithm 3.

5.4.3 Evaluation of the Scheduling Strategies

Next, the performance of the scheduling strategies previously devised were analysed. These strategies would identify the best way to schedule batches in the face of imperfect processes and stochastic demand by determining an optimum process duration and the right time to start a new seed train (and ultimately a new cell culture operation). They would be applied in a simulation environment similar to previous experiments but with a few differences.

Firstly, and most importantly, the demand is due daily and (as previously mentioned) is stochastic. The random variable of daily demand is defined as a Normal distribution, $N(\mu, (\sigma)^2)$, where μ is the expected demand and σ is its standard deviation; sampled demand is truncated and not allowed to be negative. In practice,

Algorithm 3 Pseudocode of the (s_1,s_2,B) strategy.

procedure NEW SEED TRAIN(current time t)
  ST_t = the existing seed train at time t
  USP_t = the existing USP operation at time t
  T^{ST} = the seed train run time
  T^{USP} = the USP run time
  T_t^{USP} = time elapsed in USP operation at time t
  T^{BT} = time required to turnaround bioreactor
  T^{SEP} = the setup expiry period
  T_t^{SEP} = the time elapsed in setup expiry period
  T^{THD} = threshold
  $T^{\text{THD}} = T^{\text{USP}} + T^{BT} - T^{ST}$
  if $ST_t == \{\}$ **then**
    I_t = the inventory level at time t
    if $T_t^{\text{SEP}} + T^{ST} > T^{\text{SEP}}$ **AND** $I_t \leq s_1$ **then**
      if $\text{USP}_t == \{\}$ **OR** ($\text{USP}_t \neq \{\}$ **AND** $T_t^{\text{USP}} \geq T^{\text{THD}}$) **then**
        Start seed train, ST_{t+1}
      end if
    else if $T_t^{\text{SEP}} + T^{ST} \leq T^{\text{SEP}}$ **AND** $I_t \leq s_2$ **then**
      if $\text{USP}_t == \{\}$ **OR** ($\text{USP}_t \neq \{\}$ **AND** $T_t^{\text{USP}} \geq T^{\text{THD}}$) **then**
        Start seed train, ST_{t+1}
      end if
    end if
  end if
end procedure

this means that:

$$\mu = \frac{\text{yearly demand}}{360} \tag{5.2}$$

and,

$$\sigma = 0.025 \times \frac{\text{yearly demand}}{\sqrt{360}} \quad . \tag{5.3}$$

In addition, the inventory cost is 0.6 RMU/kg/day and since demand is delivered on a daily basis, material is only discarded when it has reached the end of its shelf life. The backlog decay is also applied daily at a pro rata rate of the yearly value. This is detailed in Equation (5.4) where Δ_t is the amount of product that is undelivered at time t, θ is the daily backlog decay rate, D_t is the observed demand at time t, and S_t is the product sales at time t.

$$\Delta_t = \theta\Delta_{t-1} + D_t - S_t, \ \ \forall\, t. \tag{5.4}$$

Meaning that each day, any outstanding demand is multiplied by the equivalent of $\sqrt[360]{0.25}$ and the product of that is carried over to the next day. Otherwise, the parameters used are as detailed in Table 5.3 & 5.6, the cell culture contamination rate is 10%, and the ATF fouling rate is at 2% (both rates are per the first 60 days).

5.4.3.1 Genetic algorithm parameters

In order to evaluate the performance of the two strategies, they were tested at various demand rates. This was also to determine if there were any discernible trends dependent on the facility utilisation (i.e, the demand).

A GA was designed to tune the parameters in each strategy and its parameters are as follows: The number of generations was set to be 100, the population size was 30 with elites of six. The chromosome was made up of three genes for the (s_1, s_2, B) strategy: the first two representing s_1 and s_1 respectively are real-valued whilst the third representing B is integer. In the case of the (s, B) strategy, there are two genes — the first for s and the latter for B. The selection process for individuals to be crossed over was a tournament with a size of two. The probability of crossover being applied was 0.9 and the crossover operator was uniform crossover. Probability of a gene being mutated was 0.3 and 0.5 for the (s_1, s_2, B) strategy and (s, B) strategy, respectively. Gaussian mutation was used (with a standard deviation of 10) for the real-valued genes whilst the integer gene used (± 1) random walk as its mutation operator with the probability 0.9 that the walk continues. Random walk mutation performs a random walk starting at the current value of the gene. At each step in the walk it sets a variable k to either 1 or -1 (with uniform probability). It then attempts to add k to the current value. Then with a probability value of 0.9 it iterates another step, or else it immediately quits the walk. At the end of the walk, the gene is set to the current value, as was modified during the walk. The minimum and maximum values for the genes in the chromosome are detailed in Table 5.7.

For both strategies, the GA was run for 50 independent runs and the results which are described and analysed below are based on averages of these runs unless otherwise specified.

5.4.3.2 Comparison of scheduling strategies

As well as being compared to each other, the strategies were also compared with two 'standard' strategies, SSH1 & SSH2. These are based on the (s, B) strategy but

Table 5.7: The minimum and maximum values for genes in the chromosome.

Parameter	Type	Min. value	Max. value
s	continuous	0	100
s_1	continuous	0	100
s_2	continuous	0	100
B	integer	14	120

Table 5.8: Comparing the profit (in RMU) for the different strategies at various annual demand targets. Demand is a random variable which follows a Normal distribution as previously described where the values in this table correspond to μ. Values reported are mean $\pm$ std. error, and in the case of the GA optimised strategies are over 50 runs. **Bold** indicates best performance.

Demand (kg)	(s_1, s_2, B)	(s, B)	SSH1	SSH2
220	**95661 $\pm$ 2.0**	95650 $\pm$ 2.0	93893 $\pm$ 8.0	94185 $\pm$ 7.8
250	**109102 $\pm$ 1.9**	109095 $\pm$ 1.8	107364 $\pm$ 9.0	107471 $\pm$ 8.9
280	122607 $\pm$ 1.9	**122612 $\pm$ 2.5**	120814 $\pm$ 9.9	120837 $\pm$ 10.2
310	**136179 $\pm$ 2.7**	**136182 $\pm$ 2.0**	134247 $\pm$ 11.1	134300 $\pm$ 11.1
340	**149738 $\pm$ 2.2**	149732 $\pm$ 3.0	147486 $\pm$ 12.2	147784 $\pm$ 12.3
370	**163191 $\pm$ 2.6**	**163187 $\pm$ 3.4**	161083 $\pm$ 13.6	161275 $\pm$ 13.4
400	**176427 $\pm$ 4.1**	**176430 $\pm$ 2.2**	174615 $\pm$ 15.4	174743 $\pm$ 14.9
440	**192979 $\pm$ 3.1**	**192973 $\pm$ 3.5**	185191 $\pm$ 27.0	192379 $\pm$ 18.5

in both of these cases, s is set so that there is not more than a 5% probability of a stock-out during the lead time. Then for SSH1, B is chosen to be 60 days and for SSH2, the optimal B is chosen via an enumerative search. To evaluate SSH1 & SSH2 at the different demands, the strategies were simply simulated 20,000 times with the same random number seeds used to evaluate the performance of the final solutions evolved by the GA. The results of these experiments and comparisons are summarised in Table 5.8.

From these results, one can see that the (s, B) and (s_1, s_2, B) strategies are more or less similar in terms of overall performance. This would mean that the extra parameter in the (s_1, s_2, B) strategy does not offer any advantages in achieving a superior performance. Also, the 'standard' strategies are seen to perform worse than the other strategies. SSH2 performs better than SSH1 across all demand scales which adds more merit to the idea of selecting process durations based on the demand faced, especially at high demands. In Table 5.9 the values of parameters for the best (s, B) policy and 'standard' strategies are directly compared over the

Table 5.9: Summary of GA evolved parameters for the (s,B) policy compared with that of the 'standard' strategies. Units for parameter s and annual demand are in kg and in days for parameter B. Values for (s,B) are the out of all GA runs. Note that SSH1 and SSH2 share the same values for s.

	(s,B)		SSH1		SSH2
Demand	s	B	s	B	B
220	11.6	52	28.2	60	46
250	14.6	53	32.1	60	53
280	16.7	56	36.0	60	59
310	19.2	62	39.8	60	61
340	20.7	70	43.7	60	71
370	23.7	84	47.5	60	85
400	26.4	103	51.4	60	105
440	35.4	106	56.5	60	110

different demand scales. The parameter s is always larger in the SSH1 & SSH2 case and with increasing demand load, the difference becomes larger. However, there is a step jump at 440 kg where that difference narrows slightly. The run times for SSH2 are generally similar to that of the (s,B) policy. On the other hand, at 310 kg and 280 kg, the value of B for the (s,B) policy almost equals that of the SSH1 strategy and exceeds it by a wide margin at larger demand loads. These higher demands require greater productivity which are achieved by longer run times.

The implications of this are that a facility operating at 75% or higher of its maximum capacity[6] requires increasingly longer run times for its cell culture and using a 'standard' run time would mean the facility is being run inefficiently. Below that but above 50% of max. capacity, the variance in optimum run times over different demands is much smaller and a more moderate run time should be implemented. However, it remains important that this is paired with a good reorder level, otherwise performance (profit) will suffer.

In order to determine the importance of each parameter to the performance of the strategy, a 'rough' visualisation of the (s,B) search space was generated. This is presented in Figure 5.10 for an expected annual demand of 440 kg. It shows that for shorter process run times, B, the reorder level, s does not affect the performance. This is because shorter processes are not as productive and so inventory is perpetually below the reorder level meaning that new batches are always being

[6]Maximum capacity here is defined as the maximum throughput (or productivity) of 456 kg per annum as shown in Figure 5.7 on page 81.

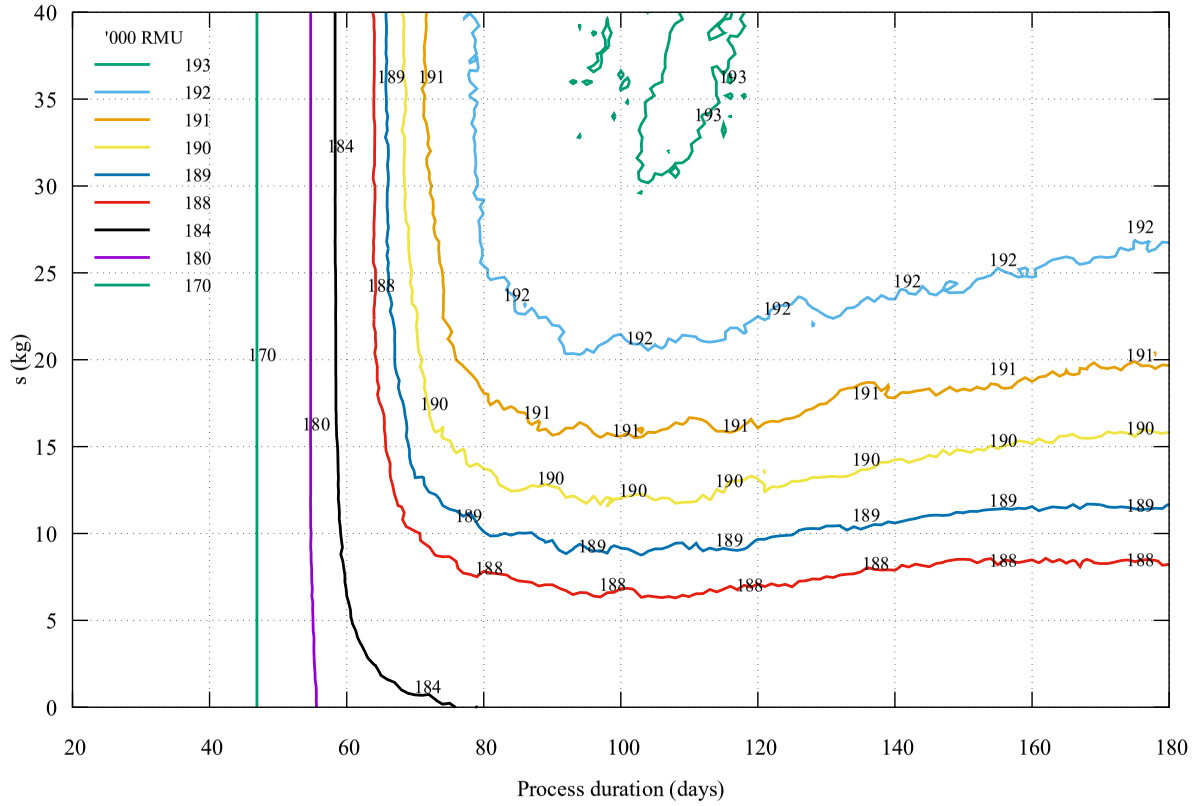


Figure 5.10: Contour plot of the profit from the (s, B) strategy for a combination of reorder levels, s, and process run times, B at an expected demand of 440 kg annually. Profit and its isolines are reported in '000s RMU.

Table 5.10: Summary of GA evolved (s, B) policy parameters for the various USP:DSP process configurations. Units for parameter s in kg and in days for parameter B. The best policy out of all GA runs is reported for each configuration.

Number of reactors	s	B
1	35.3	113
2	34.9	107
3	40.7	109
4	42.3	116

ordered. Past process durations of 55 days or so, the reorder level becomes increasingly more significant to the point that process duration has minimal effect on profit. Here, the reorder level has to be large enough to keep inventory levels sufficiently high.

5.4.3.3 Comparison of parallel-bioreactor process configurations

The (s, B) strategy was used to compare the different process configurations at a mean yearly demand of 440 kg — processes with multiple parallel bioreactors are compared with a configuration with just one larger reactor. The purpose of this is twofold. The first is to determine the optimal economic performance of each process configuration. The second is to observe if there is a difference in the optimised strategy parameters. On one hand, because the configurations with fewer reactors have a higher productivity at comparatively shorter process run times, configurations with more reactors will need longer run times to compensate especially at a high demand load. However, with more consequential failures in configurations with fewer reactors, the safety stock needed to mitigate stock-outs would need to be larger.

Fifty runs of the GA were therefore run for each of the process configurations and the results over these optimisations are reported below. In the cases with multiple reactors, the 'early/immediate seed restart' response (as described previously in Section 5.3.2 on page 83) is used if the cell culture fails before its scheduled end. The GA parameters are identical to previously described except the minimum value for B was increased to 60 days and the number of generations was set to 50. This was done to focus the search on the space illustrated in Figure 5.10 on the preceding page.

Table 5.11 on the next page summarises the performance of each configura-

Table 5.11: Profit and other characteristics for the (s,B) strategy at the various USP:DSP process configurations. Figures reported are in RMU (apart from CSL) stated and are an average over 50 runs. White background indicates best performance and darker shades denote worse performance for each metric; figures highlighted in colour are the differences between the best and performance in that scenario.

	1:1	2:1	3:1	4:1
Profit	192977	-1713	-3188	-4564
Revenue	217793	-475	-1417	-2975
Total costs	24816	1238	1771	1589
Seed	85	11	16	23
USP	5310	602	1032	830
Replacement ATF filters	19	15	31	8
Cell culture setup	491	505	880	671
DSP	17086	622	607	457
Changeover	19	6	2	36
Storage	611	439	254	870
Backlog penalties	690	119	345	705
Wastage	109	69	145	205
CSL (%)	99	-0.22	-0.65	-1.35

tion. This shows the single-reactor configuration performing better than the other process configurations and a trend of decreasing profit with more parallel reactors. This difference in profits is mainly driven by high(er) USP costs, especially in the 3:1 configurations. This is most likely due to the fact that SUBs get comparatively cheaper the larger they are (i.e., a 2000L SUB costs less than two 1000L SUB). At the 4:1 process configuration, the difference is due to significantly lower revenue. It is possible that this is because with four reactors, there are often failures, causing lower production/productivity which leads to more lost sales (reflected by much higher backlog penalties).

Meanwhile Table 5.10 reports the best optimised (s,B) parameters for each of the process configurations. It is worth noticing that generally, the reorder level — and consequently the amount of safety stock required — increases with more reactors in the process configuration. This is contrary to what was initially predicted; it appears instead that more safety stock is needed because these processes are less productive. The process duration also somewhat follows this trend; although the two- and three-reactor configurations have slightly lower duration than that of the 1:1 process, the run times increase when going from 2:1 to 4:1 config-

urations. It may be that at 2:1 and 3:1 configurations, the need to mitigate process failure dominates the lower productivity but at 4:1 configurations a longer run time is required to compensate for lower productivity exacerbated by a large demand load. Multi-reactor process configurations have more process failures and are less productive which necessitates longer run times to compensate and meet demand. However, with longer durations the risk of failures increases even more and may require higher safety stock levels to mitigate stock-outs (or higher reorder levels to increase the frequency of batches).

5.5 Summary

This chapter set out to investigate the scheduling considerations of operating perfusion cell culture processes prone to failure especially through the prism of process run times. This involved exploring the economics of the perfusion cell culture process, identifying appropriate scenarios for using short or long batches, as well as determining the impacts of process configuration and scheduling strategies and policies.

To achieve this, a custom object-oriented simulation framework modelling continuous bioprocesses in a scheduling environment was first proposed and developed. In addition to a bioprocess and scheduling model, its components include a custom discrete-event simulation engine, and modules for implementing scheduling policies and optimisation algorithms to tune them.

Utilising this framework, the work in this chapter found that given the assumptions made, optimum run times are very sensitive to demand and process failure rates. Despite this, really short batches (i.e., less than three weeks) are sub-optimal even in high failure or low demand scenarios. Utilising parallel reactors, which may offer flexibility, is less productive than one equivalent larger reactor. This difference, however, heavily depends on the response to a reactor failing. The superior option is to implement an *immediate seed restart* response — this limits the dip in productivity in a 4:1 configuration, for example, to just above 95% of the 1:1 configuration as compared to *normal seed restart* which can go as low as around 76%.

There is no major difference between the (s,B) and (s_1,s_2,B) strategies for scheduling continuous bioprocessing and anticipating stochastic events. So the former strategy would be preferred as it is simpler and requires less effort to tune.

Comparing the (s, B) policy with 'standard parameters' revealed that operating the facility at more than 75% of maximum capacity would require run times significantly longer than the 'standard' 60 days otherwise performance would suffer; it is also necessary to pair the run time with a good value for the reorder level.

While this chapter focused entirely on a single product, the following chapter will, using the framework presented here, tackle a scenario with a multi-product scenario. There, scheduling decisions or policy parameters for each product cannot be 'greedy' but have to be complementary to the others in the portfolio.

Chapter 6

Dynamic Scheduling for Continuous Bioprocesses in a Multi-Product Facility

6.1 Introduction

Prior scheduling or planning frameworks for bioprocesses are static, deterministic, and almost exclusively consider fed-batch processes. Those that model any continuous processes either do not account for uncertainty — either in demand, process/equipment failure, and/or process titres and yields — or do not intend on optimising facility schedules. As previous models did not aim to optimise schedules or capacity plans (Pollock et al., 2013b; Farid et al., 2014), there is a research gap in investigating optimal scheduling and capacity planning strategies for these continuous processes. Similarly, deterministic planning models (Siganporia et al., 2014) did not focus on uncertain events — so require complete re-optimisation after stochastic events.

Take, for example, a scheduling & capacity planning scenario for a multi-product facility where two products are scheduled and are to be manufactured. In the event a batch of the first scheduled product fails, restarting another batch would lead to the schedule running late and demand (of that product and/or the subsequent product) being delivered late. This raises the question of what the optimal rescheduling strategy is. One can simply carry on the campaign until the demand target is met, thereby moving back everything else scheduled later. Conversely,

one may decide to reschedule the manufacturing campaign(s). Deciding when to reschedule and how to implement a strategy is another question. An alternative approach to constant re-optimisation and rescheduling is to have a dynamic policy that does not prescribe an entire schedule ahead of time. Instead it makes just-in-time scheduling decisions, reacting to and anticipating changes such as failed batches, fluctuating titres & yields, and uncertain demand.

For most perfusion processes, the duration of the cell culture operation (the process run time) is fixed for each batch. However it may be optimal to allow the duration to be flexible. That is to say, if a 60 day batch produces 20 kg, and the yearly demand is 24 kg, instead of running two 60 day batches it may be better to run either a single longer batch or run two or more smaller batches. So the relevant question is to determine how beneficial is it for the batches to be of flexible duration and how best that is implemented.

In addition, it is worth noting that the reviewed literature on the SLSP with maintenance schedules (see Chapter 3) is not exactly transferable to the problem considered in this chapter. This is because 'maintenance scheduling' is inextricably linked to lot-sizing decisions in this case while the previous literature considers them as separate decisions. Equipment failure in this chapter is an increasing function of process run-time and therefore lot-size, so a decision on lot-size may be seen as implicitly scheduling preventive maintenance. However it seems prudent to consider process run-time as only a lot-sizing decision and treat its preventive maintenance and process restorative properties as a side-effect.

This chapter has multiple aims. The first is to evaluate the previously described model (Chapter 5 on page 62) on a multi-product facility. The second is to adapt and develop dynamic scheduling policies that make operational decisions in a multi-product facility. These policies anticipate and react to changes in the simulation environment (such as uncertain demand and process failure events). Finally, the relative benefits of a scheduling policy capable of implementing batches with flexible process run times is investigated.

To achieve this, a hyper-heuristic is utilised to tune the parameters of policies tailored to the problem of scheduling multiple perfusion products on a facility. The policy search comprises a simulation-optimisation approach which uses an EA as an optimisation algorithm and a custom stochastic bioprocess scheduling model to evaluate performance of candidate policies. The use of scheduling policies allows natural reactions to demand changes and process failure even if no firm schedule is

generated in advance.

As a result, existing policies from the SELSP or SCLSP literature are tailored to the peculiarities of biopharmaceutical manufacturing, in particular the semi-continuous operation of perfusion processes. In addition, it proposes a novel policy with a custom look-ahead heuristic which enables better performance on the test problem. Third, process run times are optimised for each product in the portfolio. Finally, a neural network representation is proposed and implemented for a scheduling policy that adapts process run times based on the current state of the environment.

The remainder of this chapter consists of the problem statement and description in Section 6.2 followed by detailed descriptions of the scheduling policies and representations in Section 6.3 on page 102, and the EAs used as optimisation algorithms for parameter tuning in Section 6.4. The case study on which the hyper-heuristic is evaluated is laid out in Section 6.5, which is followed by a report and discussion of the empirical evaluation in Section 6.6 on page 117.

6.2 Problem Domain

6.2.1 Problem Description

The problem that this chapter considers is a variant of the SELSP applied to biopharmaceutical manufacturing. It involves a facility and a set of drug products PF, each associated with a bioprocess which, when operated, manufactures the corresponding product. The state of the facility, m, refers to the product, p, whose bioprocess is currently in operation on the facility or is 0 (zero) if idle. The bioprocesses are comprised of the same multiple stages (unit operations) and are operated in a semi-continuous manner — the bioprocess has to be operated in multiples of a pre-defined batch length (i.e., process duration or run-time) but processed material is made available over the course of the batch and not just at the end. In general, no more than one bioprocess may be in operation in the facility at any time to avoid cross-contamination issues and ease the validation burden, though there is some exception to this which allows the earlier stages of a subsequent process to be in operation simultaneously with the latter stages of the previous process.

In addition to stationary stochastic demand, the manufacturing process is prone to equipment failure of differing types, risks, and consequences without the ability to explicitly carry out preventive maintenance (PM) *per se*. Equipment failure is a

function of process duration so ending a batch *may*, in a sense, be construed as PM but process duration is primarily a lot-sizing decision and treated as such. Finally this implementation uses a finite time horizon and discrete-time periods of one day each which is effectively continuous time given the time-scales of the bioprocesses and the stochastic events.

The objective is to maximize the overall profit, calculated as total revenue minus the costs for production, storage, process changeover, wastage, and backlog penalties given a facility with different manufacturing yields and manufacturing costs for the different products.

6.2.2 Notation

The indices p and t denote individual products and discrete time points respectively. The subset characterising the facility being considered is PF, the set of products produced by the facility.

Parameters

α_p	lead time for production of first DSP batch of product p, days
δ_p	unit cost charged as penalty for each kilogram of unfulfilled demand of product p, RMU per kilogram per day
ζ_p	shelf-life of product p, days
μ_p^D	mean daily demand of product p, kilograms
ρ_p	unit cost for each stored kilogram of product p, RMU per kilogram
σ_p^D	standard deviation of daily demand of product p, kilograms
θ	daily backlog decay rate
υ_p	unit sales price for each kilogram of product p, RMU per kilogram
ψ_p	changeover cost for starting a campaign of product p, RMU
A_p	sum of all costs incurred up till the end of ramp-up for first batch in campaign of product p, RMU
br_p	bioreactor yield per daily perfusion harvest, kilograms
yd_p	overall process yield, %

State variables

Δ_{pt}	amount of product p which is late at time t, kilograms
D_{pt}	observed demand of product p at time t, kilograms
I_{pt}	the amount of product p stored at time t, kilograms
m_t	the product being manufactured on the facility at time t, 0 if idle
s_t	the state of the system at time t
S_{pt}	the amount of product p sold at time t, kilograms
$Z_t^{(i)}$	the set of products p, such that $I_{pt} \leq Y_p^{(i)}$

Decision variables (policy parameters)

B	run time of perfusion cell culture batch, days
$Q^{(i)}$	product cycle sequence
w	neural network connection weights
$Y^{(1)}$	reorder point, kg
$Y^{(2)}$	order-up-to level, kg
$Y^{(3)}$	can-order point, kg
$Y^{(4)}$	can-order-up-to level, kg
$Y^{(5)}$	reorder point, kg (BSP3)

6.3 Scheduling Policies

The scheduling policies laid out here are tailored for the problem described in Section 6.2. These policies can be grouped into three main representations: a fixed cyclic policy, a base-stock policy, or an Artificial Neural Network (ANN).

6.3.1 Fixed Cycle Policies

These policies function on the basis that the facility will manufacture batches according to the sequence associated with the specific policy. The main parameter for this type of policy representation is a product sequence and process run times. Depending on the complexity, it may optionally include parameters for inventory levels to determine how many batches to order. In each policy, one sequence is defined while the process run times and inventory parameters are defined for each product in the scenario the policy is designed for. Three of these fixed cycle poli-

cies are laid out here.

6.3.1.1 Basic fixed cycle policy (FCP1)

This policy ensures that the manufacturing facility follows a fixed sequence of batches. The decision variables for this policy are:

1. The sequence of product batches, $Q^{(1)}$
2. The process run time of batches in the campaign, B

The sequence is made up of a number elements which can be one of any p in PF which indicates that a single batch of product p is to be manufactured with a run time of B_p. In addition, any element in the sequence may be zero (0) which indicates that the facility should be made idle. This policy is implemented by cycling through each element in the sequence, producing a batch of the product the item corresponds to, and then moving to the next element in the sequence — when it gets to the end of the sequence, it starts from the beginning again. Consecutive identical but non-zero elements in the sequence mean that consecutive batches of that product are produced. However, consecutive zeroes in the sequence are pruned down to one instance[1] which means that the length of sequences $|Q|$ can be variable. The time the facility spends being idle is determined by the inventory levels of the products. Specifically, idle time is ended when any product's run-out time falls below 90 days. At that point, the facility will then move on to the next product in the sequence. Also, if a batch ends prematurely due to process failure, the facility will move to the next element in the sequence.

6.3.1.2 Fixed cycle with order-up-to level (FCP2)

In addition to a sequence and process run time, this policy also has a third set of decision variables. The decision variables are defined below:

1. The sequence of product campaigns, $Q^{(2)}$
2. The order-up-to level, $Y^{(2)}$
3. The process run time of batches in the campaign, B

[1] A sequence with a leading and trailing zero will have the leading element preserved and the last element deleted.

Instead of a sequence of product batches in FCP1, this policy has a sequence of product campaigns. The difference being that each element in the sequence of campaigns commits the facility to produce at least one batch (with run time of B_p) of the associated product. The facility will move on to the next product in the sequence after the batch that brings the inventory of the currently manufactured product past its order-up-to level. The facility does not go idle until all products in the sequence are above their respective order-up-to levels. However, if product $Q^{(2)}_{(i+2) \bmod |Q|}$ is below its order-up-to level but the product in the sequence preceding it $Q^{(2)}_{(i+1) \bmod |Q|}$, is not, the facility will still need to produce at least a batch of $Q^{(2)}_{(i+1) \bmod |Q|}$ before proceeding to $Q^{(2)}_{(i+2) \bmod |Q|}$. Again, when the facility reaches the end of the sequence, it returns to the start and begins the sequence again in a cyclical fashion. Each element in the sequence can be any p in PF but consecutive identical elements in the sequence are pruned down to one instance (in the same manner as consecutive zeroes in FCP1) so the sequences can be of variable length.

6.3.1.3 Fixed cycle with skip-ahead (FCP3)

This policy, adapted from Löhndorf and Minner (2013), is an improvement to FCP2. The decision variables are:

1. The sequence of product campaigns, $Q^{(2)}$
2. The order-up-to level, $Y^{(2)}$
3. The process run time of batches in the campaign, B

FCP3 is identical to FCP2 apart from the fact that it can skip ahead in the sequence. What this means is that if a facility is to move on to $Q^{(2)}_{(i+2) \bmod |Q|}$ but the product is already above its order-up-to level, it does not need to manufacture that product so it skips ahead to the next element in the sequence that has its inventory below its order-up-to level.

6.3.2 Base-Stock Policies

Three base-stock policies (BSPs) are presented here, two of which are adapted from Löhndorf and Minner (2013), and the third is a novel development. The adaptations from the versions presented in that previous work are necessary because the problem in this chapter introduces processes that can fail and which consist of a

fixed sequence of multiple batch (and semi-continuous) unit processes. In addition, there is a need to identify an optimum run time for one of the constituent unit processes — the perfusion cell culture in this case. Therefore this problem setting differs from that in Löhndorf and Minner (2013) because here, there are batches that require their run times optimised. Also, since the processes here are composed of multiple stages, the decision to make the next batch has to be made before the current batch is over. This means that the amount of product made and delivered in that time lag needs to be estimated and taken into account.

In this case, because the seed train for a subsequent batch can be started before the completion of the current batch, the decision epoch begins at the seed-restart threshold or the changeover threshold. These thresholds are determined by the bioreactor turnaround time, and the changeover time between products respectively. Decisions are also made once the cell culture is contaminated and fails or if the facility is idle. As a result of the lag between the point when a decision is required and the end of the current batch, the policies presented here will take into account the expected extra material that would be produced by the end of the current batch (if the facility is not idle). So, γ is set to be the estimated net production (i.e., material produced minus expected demand) in the interval between the decision-making time point and the end of the batch. Decision epochs end when a decision to start a batch (of any product) has been made or selected.

6.3.2.1 Simple base-stock policy (BSP1)

This is the first of the policies adapted from Löhndorf and Minner (2013). In this case, there are three decision variables defined for each product:

1. The reorder point, $Y^{(1)}$
2. The order-up-to level, $Y^{(2)}$
3. The process run time of batches in the campaign, B

The logic for determining when to start a new seed train is detailed in Algorithm 4. This states that if the facility is idle and there is a product p with inventory below its $Y^{(1)}$, a new batch — with a cell culture run time of B_p days — for this product is initiated. If there is more than one product to which this applies, the product selected to be manufactured next is the one with the smallest *run-out time* — i.e., time till product runs out of stock, I_p/μ_p, where I_p is inventory of product

Algorithm 4 Pseudocode of BSP1.

procedure BSP1(current time t)
    PF = the set of products manufactured
    m_t = the product manufactured at time t, 0 if idle
    I_{pt} = the inventory level of product p at time t
    γ_{pt} = the estimated net output of p between t and end of its batch
    μ_p = the expected demand of product p
    $Z_t = \{p : I_{pt} \leq Y_p^{(1)}\}\ \forall p \in \text{PF}$
    if $m_t > 0$ **AND** $I_{mt} + \gamma_{mt} < Y_m^{(2)}$ **then**
        Start new seed train of product m_t
    else if $Z_t^{(1)} \neq \{\}$ **then**
        Start new seed train of product $\arg\min_{i \in Z_t}\{I_{it}/\mu_i\}$
    else if $\forall p$: $I_{pt} > Y_p^{(1)}$ **then**
        Keep facility idle
    end if
end procedure

p and μ_p is its expected demand. Once the facility is manufacturing a product, it does not go idle or switch products (changeover) until it has reached $Y^{(2)}$. The drawback of this is that if the inventory of another product is falling critically low, the facility cannot be switched to the critical product to prevent a stock-out.

6.3.2.2 Can-order base-stock policy (BSP2)

This policy was designed to be an improvement on BSP1 by including two more parameters. This allows the control policy to interrupt a campaign (a set of consecutive batches of the same product) if the inventory of another product falls critically low. The decision variables for each product are identified below:

1. The reorder point, $Y^{(1)}$
2. The order-up-to level, $Y^{(2)}$
3. The can-order point, $Y^{(3)}$
4. The can-order-up-to level, $Y^{(4)}$
5. The process run time of batches in the campaign, B

The logic for determining when to start a new seed train is detailed in Algorithm 5. If the facility is idle and there is a product p with inventory below $Y^{(3)}$, a

Algorithm 5 Pseudocode of BSP2.

procedure BSP2(current time t)
  PF = the set of products manufactured
  m_t = the product manufactured at time t, 0 if idle
  I_{pt} = the inventory level of product p at time t
  γ_{pt} = the estimated net output of p between t and end of its batch
  μ_p = the expected demand of product p
  $Z_t^{(1)} = \{p : I_{pt} \leq Y_p^{(1)}\}\ \forall p \in \text{PF}$
  $Z_t^{(3)} = \{p : I_{pt} \leq Y_p^{(3)}\}\ \forall p \in \text{PF}$
  if $m_t > 0$ **AND** $I_{mt} + \gamma_{mt} < Y_m^{(4)}$ **then**
    Start new seed train of product m_t
  else if $Z_t^{(1)} \neq \{\}$ **then**
    Start new seed train of product $\arg\min_{i \in Z_t^{(1)}} \{I_{it}/\mu_i\}$
  else if $m_t > 0$ **AND** $I_{mt} + \gamma_{mt} < Y_m^{(2)}$ **then**
    Start new seed train of product m_t
  else if $Z_t^{(3)} \neq \{\}$ **then**
    Start new seed train of product $\arg\min_{i \in Z_t^{(3)}} \{I_{it}/\mu_i\}$
  else if $\forall p$: $I_{pt} > Y_p^{(3)}$ **then**
    Keep facility idle
  end if
end procedure

new batch for this product is started. The campaign cannot be interrupted until the inventory of that product exceeds $Y^{(4)}$. When this occurs, changeover to another product is allowed if its inventory level is less than $Y^{(1)}$ — again any ties are settled by picking the product with the smallest run-out time. However, if there are no products with inventory below their reorder points, the campaign may continue until it exceeds $Y^{(2)}$.

So the facility does not go idle until all products are above their can-order points. In this manner, this policy works similarly to BSP1 but with the can-order point and the can-order-up-to level it enables interruptions. In general, $Y^{(1)} \leq Y^{(3)} \leq Y^{(4)} \leq Y^{(2)}$ and BSP1 can be considered a special case of BSP2 where $Y^{(1)} = Y^{(3)}$ and $Y^{(4)} = Y^{(2)}$. The disadvantage of this policy is that with more parameters (compared to BSP1 for example), it requires a greater computation effort in searching for good parameters. In addition, due to the ability to interrupt campaigns, this policy may introduce more product changeovers which require substantial operational and validation effort in practice.

6.3.2.3 Forecasting base-stock policy (BSP3)

This policy is a novel contribution proposed here and utilises a 'look-ahead' heuristic. The following decision variables are identified:

1. The reorder point, $Y^{(5)}$
2. The process run time of batches in the campaign, B

Algorithm 6 Pseudocode of BSP3.

procedure BSP3(current time t)
  PF = the set of products manufactured
  m_t = the product manufactured at time t, 0 if idle
  I_{pt} = the inventory level of product p at time t
  γ_{pt} = the estimated net output of p between t and end of its batch
  μ_p = the expected demand of product p
  $Z_t^{(5)} = \{p : I_{pt} \leq Y_p^{(5)}\}\ \forall p \in \text{PF}$
  $CE(\pi)$ = the estimated cost of manufacturing a permutation, π
  if $Z_t^{(5)} \neq \{\}$ **then**
    $\mathfrak{S}(Z_t^{(5)})$ = all permutations of products in $Z_t^{(5)}$
    Select cheapest permutation, $\pi^* = \arg\min_{\pi \in \mathfrak{S}(Z_t^{(5)})} \{CE(\pi)\}$
    Start new seed train of product π_1^*
  else if $\forall p\colon I_{pt} + \gamma_{pt} > Y_p^{(5)}$ **then**
    Keep facility idle
  end if
end procedure

When the inventory of a product p falls below the reorder point during a decision epoch (i.e., during idle time, or after a batch ends or is contaminated), a new batch of that product is started with cell culture run time of B_p. However, if there is more than one product with inventory below $Y^{(5)}$, the heuristic first generates all possible permutations of manufacturing a single batch of each of the products with inventory less than $Y^{(5)}$. Next the heuristic estimates the costs of each permutation. This estimated cost is the projected sum of inventory costs, lost sales penalties, and any changeover costs assuming that the processes will be 'perfect' and the demand realised is equal to the mean or mode of its probability distribution — other manufacturing costs will be the same regardless of manufacturing permutation. Cost estimation is done by generating a function describing the piecewise linear estimation of each product's inventory before, during, and after production of its corresponding batch within the permutation time frame. The inventory cost

for each product is the product of the inventory rate and the absolute value of the sum of positive integrals of the function; and the lost sales penalty is the product of the backlog penalty cost and the absolute value of the sum of negative integrals of the function. The permutation of batches that has the lowest estimated cost is selected and a batch of the first product in the permutation is started.

It should be noted that this policy does not automatically schedule the entirety of the sequence of cheapest permutation but at the next decision time, the heuristic generates a new set of permutations, evaluates them, and then makes a decision based on the new evaluations. The logic of this policy is laid out in Algorithm 6. The drawbacks of this policy include the need to enumerate all permutations of elements in $Z^{(5)}$ which, in the worst case, gives the policy a complexity of $O(n!)$ where n is the number of products. In addition, the cost estimation is not exact and does not anticipate or account for the possibility of process failure.

6.3.3 Artificial Neural Network Policy

This policy type is made up of an Artificial Neural Network (ANN) representation of a priority rule. For each decision available to the neural network to make, a priority value is computed using state variables. The decision that has the highest computed priority is then chosen and implemented. The decision-making that is available to this policy is flexible because decisions are made at very regular decision epochs. These decision epochs are generally each day — the exception being during the early stages of a new batch.
Every day, this calculates priority values for these set of decisions:

1. Keep/make the facility idle
2. Continue the current batch (if there is one) for another day
3. Start a new batch of product p_i $\forall i$

The network is structured as a fully connected feed-forward neural network with a single hidden layer. This ANN representation has been shown to give good results when computing priority indices in a stochastic dynamic environment (Branke et al., 2015). The network's input layer comprises information on the state of the system and a bias node that is always set with the value of 1. Apart from the bias node, there are $2n+2$ input nodes, where n is the number of products. Of those nodes, $n+1$ reflect the state of the facility m: one for each p, plus one

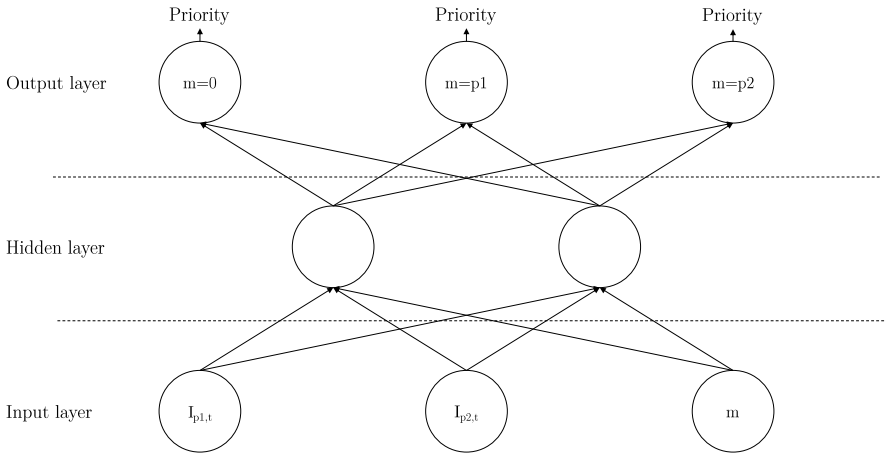


Figure 6.1: Illustration of ANN representation of priority rule for a scenario with two products and three available decisions. Both input and output layers have three nodes and the hidden layer has two nodes (the bias node and its connections are not shown in this figure).

for when the facility is idle. The node corresponding to the observed state of the facility is set to an input value of $+1$ while the others are set to -1. Another n nodes correspond to the inventory or run-out times of the different products. The last input node represents how long the current batch has been in operation. Every node in the hidden layer is connected with every node in the input layer. The output layer comprises $n+2$ nodes and every node in that layer is connected with all the nodes in the hidden layer. Each of the output nodes corresponds to one of the decisions mentioned previously.

Each neuron computes the function described below in Equation (6.1) as its output value.

$$\text{out}(z,w) = \tanh\left(\sum_{i=1}^{N} w_i z_i\right). \tag{6.1}$$

Here, N is the number of incoming edges, the vector z contains the activation levels (i.e, output values) with an incoming connection to the neuron, and w is the weight vector specifying a weight for each incoming connection. This means that in the case of the neurons in the input layer, w and z will each have one element and the element in w will be equal to 1.

As a result, the number of weights in this network can be calculated as in

Equation (6.2). Where N_{output} is the number of neurons in the output layer, N_{input} is the number of neurons in the input layer (excluding the bias node), and N_{hidden} is the number of nodes in the hidden layer.

$$N_{\text{output}} + N_{\text{hidden}}(N_{\text{input}} + N_{\text{output}} + 1). \tag{6.2}$$

Therefore the priority value for any given decision corresponding to its output node u is calculated as:

$$\text{priority}(u,s) = \tanh\left(w_u^{BO} + \sum_{h=1}^{N_{\text{hidden}}} w_{u,h}^{O} \tanh\left(w_h^{BH} + \sum_{i=1}^{N_{\text{input}}} \tanh\left(w_{u,h,i}^{H} s_i\right)\right)\right). \tag{6.3}$$

In Equation (6.3): s is the vector containing the attributes that describe the state of the system; w^O is the weight matrix for the connection weights from the hidden layer to the ouput layer; w^H is the weight array for the connection weights from the input layer to the hidden layer; w^{BO} is the vector of connection weights from the bias node to the output layer; and w^{BH} is the vector of connection weights from the bias node to the hidden layer. Therefore, it is these weights (i.e., w^O, w^H, w^{BO}, w^{BH}) that determine the policy (and its quality) and what the optimisation algorithm will search for during its procedure.

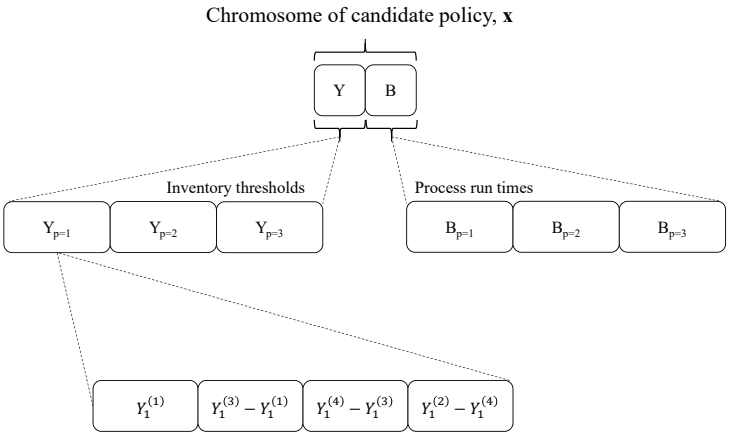
This neural network approach implicitly optimises process run times but also means that they can be flexible — i.e., each batch of the same product does not have a fixed run time. This means a batch can be cut short in order to change over to a more critical product in reaction to a change in the state of the environment. Figure 6.1 shows an example of a neural network structure.

6.4 Optimisation Algorithms

Evolutionary algorithms were used to tune the parameters of the scheduling policies. Specifically, the performance of a GA and a CMA-ES was compared for the optimisation of the FCP and BSP parameters; for the ANN, only a CMA-ES is used. These were implemented in Java™ using the ECJ Library (Version 24) (Luke, 1998).

The genomes used to represent BSPs were designed to deal with the constraints on policy parameters. For example, a BSP1 policy needs to have its parameters such that $Y^{(1)} \leq Y^{(2)}$ and a BSP2 policy needs its parameters so $Y^{(1)} \leq Y^{(3)} \leq Y^{(4)} \leq Y^{(2)}$. This is illustrated in Figure 6.2, where instead of the genome represent-

(a)



(b)

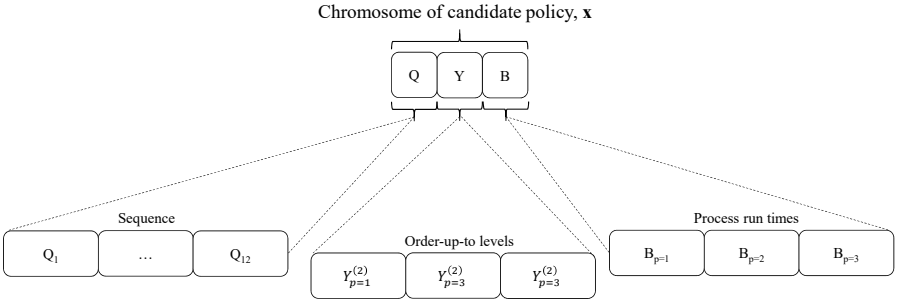


Figure 6.2: (a) Structure of a BSP2 policy chromosome for a facility manufacturing three drug products where the process run times are simultaneously optimised. (b) Structure of an FCP2 or FCP3 policy chromosome with max. sequence length of 12, and three products where run times are optimised.

ing the inventory thresholds directly, the difference between adjacent parameters is encoded. The ranges for the gene encodings are detailed in Table 6.1.

All results reported are an average of 50 runs unless otherwise stated. The parameters of the two EAs deployed are detailed below.

6.4.1 Genetic Algorithm

A GA was designed to optimise the parameters in each BSP and FCP strategy. The GA parameters chosen are as follows: the number of generations was set to be

200, the population size was 30 with elitism[2]. The genomes were determined by the sub-type of policy and the number of products.

So, for the BSPs, if the policy did not simultaneously optimise process run time, the length of the chromosome would be the number of parameters in the policy multiplied by the number of products. In the case where the policy was to also optimise the process duration, then additional genes for each product would be added to the chromosome. The first segment of the chromosome representing the policy parameters for each product were real-valued whilst the last few genes representing the process duration were integer (see Figure 6.2).

If process run times were not optimised, the length of the chromosome representing an FCP1 policy was the maximum sequence length. For FCP2 and FCP3, genes for each product for the order-up-to levels are added. And in the case that run times are also optimised, additional genes for each product would be added to the chromosome. The first segment of the chromosome coding for the product sequence and the third segment representing the process run times are integer. The middle segment coding for the policy parameters for each product, meanwhile, were real-valued (see Figure 6.2).

The selection process for individuals to be crossed over was a tournament[3] (with replacement) with a size of two. The probability of crossover being applied was 0.9, and the crossover operator was uniform crossover for the BSPs or two-point crossover in the case of the FCPs. Probability of a gene being mutated was the inverse of the chromosome length (i.e., 1/chromosome length). Gaussian mutation was used (with a standard deviation of 6) for the real-valued genes. The integer genes representing process run times used random walk (± 1) as its mutation operator with the probability 0.9 that the walk continues. For the segment encoding the sequence, shift mutation was used which entails removing the element in the sequence and re-inserting it in a new random position. The minimum and maximum values for the genes in the chromosome are detailed in Table 6.1.

6.4.2 Covariance Matrix Adaptation Evolution Strategy

As previously described in Chapter 3, the CMA-ES algorithm is a state-of-the-art EA for difficult black-box optimisation problems in continuous domain. Here, a

[2]The elitism implemented was such that the fittest six individuals in the population each generation (the elites) are carried over to the next generation automatically.

[3]This was used instead of stochastic universal sampling (SUS) because SUS does not work with the negative fitness values that some solutions produced.

Table 6.1: The minimum and maximum values for the genes in the chromosome.

Gene	Type	Min value	Max value
$Y^{(i)}$, $Y^{(i)} - Y^{(j)}$	continuous	0	60
$Q_i^{(1)}$	integer	0	n
$Q_i^{(2)}$	integer	1	n
B	integer	14	120

population of λ new candidate solutions is sampled at every generation from a multi-variate normal distribution. The mean of the distribution is updated by taking a weighted average of selected points from the current sample. The covariance matrix is also incrementally updated using a self-adaptation strategy to improve the likelihood of successive search steps. As part of the covariance matrix adaptation, the algorithm implements step-size control. This is to ensure longer steps are taken instead of more but smaller steps in the same direction, or that shorter steps are taken instead of longer steps that cancel out each other. The aim of this is to prevent premature convergence but allow fast convergence to an optimum by making the expected consecutive movements/steps of the distribution approximately orthogonal.

The CMA-ES does not have many user-specified parameters, as a lot of them are calculated based on the chromosome specified. As a result the default settings were used (Hansen and Ostermeier, 2001). The population size λ is by default $4 + \lfloor 3 \ln |\mathbf{x}| \rfloor$ where $|\mathbf{x}|$ is the size of the genome $\mathbf{x}$; a weighted average of $\lfloor \lambda/2 \rfloor$ individuals is used to update the distribution mean; and the initial covariance matrix is set to the identity matrix.

The chromosome for each FCP or BSP policy has the same structure as previously discussed but with the values of the encoded genes normalised to fall between $[-1, +1]$. These values are then transformed to the actual policy parameters at the point of fitness evaluation. To enable comparison with the GA, the CMA-ES was set the same budget of $6,000$ fitness evaluations and the starting point for the search was set randomly in the decision space.

To tune the ANN policy, the chromosome used consisted of a vector of all the weights for the neural network. These weights are encoded directly onto the chromosome with values in the interval $[-1, +1]$. The search for the neural network weights was allocated a budget of $12,000$ fitness evaluations.

Table 6.2: Process economics parameters for products in relative monetary units (RMU) unless otherwise stated.

	$p1$	$p2$	$p3$
Seed cost	4.6	5.2	5.1
Daily cell culture cost	3.4	3.2	3.6
Cell culture batch setup cost	26	26.9	33.7
ATF replacement cost	17.8	14.6	15.7
DSP batch cost	10.7	11	14.2
Sales price (RMU/kg)	150	95	100
Backlog penalty cost (RMU/kg/day)	0.25	0.1	0.1
Annual demand (kg)	60	120	115
Reactor yield (kg)	2.03	2.25	1.38

6.5 Case Study Description

To evaluate the scheduling policies, a biopharmaceutical industrial case study was designed. The data comprises anticipated market demand and manufacturing facility characteristics. This problem features multiple products to be produced in a single facility with different efficiencies, yields and costs, perishable inventory, and backlogging and lost sales allowed. The processes to manufacture these products are all based on a platform mAb process as described in Chapter 5.2.1.

The demand forecast is made up of three different products $(p1 - p3)$ to be manufactured over a period of seven years. The demand forecast shows the expected annual demand which is stationary (i.e., does not change from year to year) but is stochastic and is sampled either daily, monthly, quarterly, or annually (Table 6.2). The different constituents of the manufacturing costs are also listed in that table as well as the reactor yields, the sales price per kilogram of each product, and the periodic penalty cost for each unit of unfulfilled demand. The reactor yield indicates how much product is in each daily harvest from the bioreactor. The product of this and the overall process yield is how much material is deposited in inventory after each DSP batch. For all products, the overall process yield is 69%.

Changeover cost is accrued when switching between two different products or after the setup expiry period lapses between batches of the same product. Changeover and turnaround times are defined as the minimum time between two USP operations for different and same product respectively. Therefore the seed-restart threshold for a product's process is the sum of its cell culture run-time and turnaround time minus its seed run time. Likewise, the changeover threshold is cell culture

Table 6.3: Case study parameters

Parameter	Value	Unit
Shelf life	720	Days
Backlog decay	0.5	Per 180 days
Inventory rate	0.01	RMU/kg/day
Bioreactor turnaround	4	Days
Changeover time	10	Days
Wastage rate	5	RMU/kg
Setup expiry	30	Days
Number of reactors	1	
Changeover cost	35	RMU
Overall process yield	69	%

run-time plus changeover time minus seed train run time of the subsequent product. There is also a daily backlog decay on unfulfilled demand so that it becomes less important over time. This is detailed in Equation (6.4) where for product p, Δ_{pt} is the amount of product p that is late at time t, θ is the daily backlog decay rate, D_{pt} is the observed demand of product p at time t, and S_{pt} is the amount of product p that is sold at time t.

$$\Delta_{pt} = \theta\Delta_{p,t-1} + D_{pt} - S_{pt}, \quad \forall\, p,t. \tag{6.4}$$

This means that if 1kg of product is undelivered at time t, the amount due at time $t+1$ is $(1 \times \sqrt[180]{0.5})$ kg plus whatever new demand arrives at $t+1$ less any sales at $t+1$. Each product has a maximum period of time (its shelf-life) that it can be stored for before it perishes. Any product that has to be thrown away because it has exceeded its shelf-life or as a consequence of process failure will also accrue a wastage penalty per kilogram. These parameters are specified in Table 6.3. It is assumed that processes utilise a single reactor so that the process configuration is a 1:1 USP:DSP ratio.

The case study is designed so that each of the three products has a particular characteristic. The first product, $p1$, is high-value and low-demand; $p2$ is high-demand and high-yield; and $p3$ is high-demand and low-yield. So the trade-off between $p1$ and $p2$ is that the former commands a higher sales price per kilogram manufactured but also a larger penalty per kg of unfulfilled demand. On the other hand, comparing $p3$ to $p2$, we see that $p3$ has a similar demand profile as $p2$ but its yield is a third lower.

This case study will be the basis of simulation optimisation experiments performed and reported in the following section.

6.6 Results and Discussion

To evaluate the performance of the policies, optimisation runs were carried out in the manner described in Section 6.4 on page 111. This section presents and discusses the scenarios investigated and their results.

6.6.1 Description of the Standard Policy

In addition to the proposed policies, the performance of a standard policy is also reported. This policy serves as a baseline to compare the optimised policies. It takes the form of the simple base-stock policy where the parameters are heuristically chosen by the Doll & Whybark heuristic (Doll and Whybark, 1973) adapted by Gascon et al. (1994) and used as an initial guess in the direct policy search by Löhndorf and Minner (2013). This policy is denoted as **BSP0**. This heuristic attempts to construct a schedule by producing products in repetitive cycles. So each product is manufactured once every certain period of time (that may be unique to each product) where each of these periods is a multiple of a *fundamental cycle period* or a *common cycle time*. The values of the parameters of the standard policy are based on common cycle time and the method to calculate these values is as follows: Let $\hat{T}$ be the common cycle time and k be the safety factor.

$$\hat{T} = \max\left\{\sqrt{\frac{2\sum_{\forall p\in PF} A_p}{\sum_{\forall p\in PF}\rho_p\mu_p^D(1-\mu_p^D/(br_p yd_p))}}, \frac{\sum_{\forall p\in PF}\alpha_p}{1-\sum_{\forall p\in PF}\mu_p^D/(br_p yd_p)}\right\}, \tag{6.5}$$

$$k = \Phi^{-1}\left(\frac{\delta_p(1+\theta^{\hat{T}})/2}{\delta_p(1+\theta^{\hat{T}})/2+\rho_p}\right). \tag{6.6}$$

Where μ_p^D is the expected daily demand; σ_p^D is the standard deviation of the daily demand; α_p is the process lead time which in this case is the sum of the seed train time, the ramp-up time, and the DSP time; br_p is the amount harvested from the daily perfusate of the bioreactor; yd_p is the process yield; δ_p is the lost demand penalty cost; θ is the daily backlog decay rate; ρ_p is the inventory holding cost; and A_p represents all setup costs associated with the first batch of a campaign prior

to its first perfusion harvest, i.e., the sum of the changeover cost, seed cost, cell culture setup cost, and the perfusion costs in the ramp-up period.
Then, the reorder level and order-up-to point are, for each product p, set to:

$$Y_p^{(1)} = \max\left\{\mu_p^D \alpha_p + k_p \sigma_p^D \sqrt{\hat{T}},\ 0\right\}, \tag{6.7}$$

$$Y_p^{(2)} = Y_p^{(1)} + \max\left\{\mu_p^D (1 - \mu_p^D / (br_p yd_p))\hat{T},\ 1\right\}. \tag{6.8}$$

In this standard policy, the process run time for each product is set to 60 days.

6.6.2 Evaluation of Optimised Policies

The policies were optimised twice: once with the process durations fixed to 60 days, and a second time where the optimisation algorithm is free to optimise the process duration. These are differentiated by the suffixes appended to the policy name. Where the process duration is fixed, the suffix is *'A'* and if the process duration is optimised, the suffix is *'B'*.

For the base-stock policies (BSP1-3), the results reported are averaged over 50 independent runs of the GA — the results were similar to that of the CMA-ES so those were reported. For FCP1, the reported results are from GA runs as these results outperformed the CMA-ES. The opposite is true of FCP2 and FCP3; CMA-ES results are reported for those policies. The ANN results (using the CMA-ES) are also an average of 50 independent runs like the other tuned policies.

The optimisations were run on the case study data previously described for a time horizon of seven years (with a year comprising 360 days). Initially, the first two years of the horizon served as a warm-up period but this was changed because it caused unrealistic artefacts in the inventory profiles of simulations of ANN policy solutions. During the EA optimisation, the fitness evaluation uses the average performance of 500 simulations as this was found to give a good estimate without making computation time(s) too long. The stochastic demand for each product follows a Normal distribution, $\mathrm{N}(\mu, (\sigma)^2)$, where μ is the yearly forecast demand (Table 6.2) and σ is 0.025μ. The demand frequency is set to be daily which means that during each simulation, demand is sampled, is due, and can be delivered everyday; sampled demand is truncated and not allowed to be negative. This means the distribution describing the daily demand is $\mathrm{N}(\mu^D, (\sigma^D)^2)$ where

$\mu^D = \frac{\mu}{360}$ and $\sigma^D = \frac{\sigma}{\sqrt{360}}$. Failure rates were set such that the probability of cell culture contamination was 10% in 60 days and the probability of ATF filter failure was 2% in 60 days too. Finally, each simulation was started with each product having an initial inventory equal to a quarter of the expected yearly demand.

6.6.2.1 Policies with standard run-times

Here, the *'A'* sub-types of the base-stock and fixed cycle policies are compared with the standard policy.

Base-stock policies Table 6.4 compares the performance characteristics of the standard policy (BSP0) to those tuned base-stock policies (BSP1-3) where the process run times are fixed to 60 days. The performance characteristics for BSP0 were evaluated from $20,000$ simulations of the policy and the same was done (on the same random number seeds) for each optimised solution from all runs of BSP1A, BSP2A, and BSP3A. The table shows that the tuned base-stock policies outperform the standard policy in terms of the expected profit generated. This is driven primarily by differences in the revenue (and consequently, backlog penalties and CSL) and storage costs. By scrutinizing the seed costs and the cell culture setup costs, it appears that both the standard and the optimised base-stock policies start a similar number of batches so the advantage of the optimised base-stock policies is due to the sequence or timing of the batches ordered. It is worth pointing out that BSP0 does outperform the optimised policies in one metric, changeover costs. This means that although the overall timing and sequencing of batches in BSP0 is sub-optimal in terms of the overall objective, it is able to schedule batches of the same product together in longer campaigns thereby reducing changeover costs and potentially making the operation of the facility more straightforward with fewer manufacturing switches between products.

Differences in performance between the optimised policies are much smaller when compared to the standard policy; between the optimised policies, BSP1A is the worst, with BSP2A performing a bit better, and BSP3A best. That ranking is the same when looking at revenue but is reversed with regards to the total costs. The only other major differences between them is that BSP2A appears to have on average at least two fewer product changeovers than the other policies. This is however offset by it having larger storage costs than BSP1A and BSP3A. Any differences between two policies in total seed, USP or DSP costs are due to a policy

having marginally fewer or more batches started than the other, since they all have the same process run times.

Fixed cycle policies As with the base-stock policies, the performance of the standard policy is compared with the fixed cycle policies with process run times fixed to 60 days. This is shown in Table 6.5. Comparing BSP0 with the tuned fixed cycle policies, similar trends to the base-stock policies are observed. The tuned policies are again substantially superior to the standard policy in terms of maximising profit and revenue, and minimising total costs. The standard policy appears to start slightly fewer batches than the other policies and although the FCPs have lower changeover costs than the BSPs, they are still a lot higher than the standard policy. It is also worth noting that the best FCP performs worse than the worst BSP (FCP3A vs. BSP1A).

Comparing the individual FCPs, the differences between them are again less substantial than when compared to the standard policy. However these differences are larger than observed between the tuned BSPs. FCP3A performs best and of the other two, FCP1A is worse. This ranking is mostly driven by the number of batches that FCP1A and FCP2A have to start and run as well as the higher inventory levels they maintain. Also, though FCP1A has higher seed and cell culture setup costs than FCP3A, its changeover costs are lower which indicates that FCP1A schedules in multi-batch campaigns more often than FCP3A. Similarly, since FCP2A has to make at least a batch of each product in the sequence between its current position and the product that has just fallen below its order-up-to level, it will start and run a large number of batches compared with FCP3A. Changeover costs for FCP2A are also lower than for FCP3A (and also lower than for FCP1A too). This may seem counter-intuitive because FCP2A may have to start campaigns and accrue changeover costs that are not necessary in order to get to a critical product in the sequence. But this means that if the critical product is far away enough on the sequence, by the time the facility gets to that product, multiple batches would be required to bring it above its order-up-to level.

6.6.2.2 Policies with optimised run-times

In Table 6.6 and Table 6.7 the performance characteristics of the best performing FCP and BSP policies that had a fixed process run time — FCP3A and BSP3A in this case — are compared with the fixed cycle policies and the base-stock poli-

Table 6.4: Profit, costs, customer service level (CSL), and other performance characteristics for the three BSPs with process duration fixed to 60 days as well as the standard heuristic (BSP0) solution. Mean ± std. err. are listed of 50 runs each and values reported are in RMU apart from CSL values. Statistically best values are highlighted in **bold**.

	BSP0	BSP1A	BSP2A	BSP3A
Profit	179015 ± 20.9	189589 ± 1.6	189651 ± 4.5	**189711 ± 1.7**
Revenue	214690 ± 17.9	222908 ± 6.8	222996 ± 6.0	**223075 ± 7.5**
Total costs	35676 ± 5.7	**33319 ± 7.1**	33345 ± 6.1	33363 ± 8.0
Seed	**179.0 ± 0.04**	181.3 ± 0.03	181.7 ± 0.02	181.9 ± 0.03
USP	**8059 ± 0.9**	8155 ± 1.1	8173 ± 1.1	8181 ± 1.5
Replacement ATF filters	10.3 ± 0.09	10.5 ± 0.01	10.6 ± 0.01	10.6 ± 0.01
Cell culture setup	**1064.6 ± 0.22**	1076.9 ± 0.15	1079.6 ± 0.15	1080.7 ± 0.21
DSP	**21022 ± 2.5**	21265 ± 2.8	21313 ± 3.1	21342 ± 4.2
Changeover	**478.0 ± 0.24**	1132.5 ± 0.44	1053.1 ± 4.00	1150.1 ± 1.11
Storage	3065 ± 1.0	2253 ± 5.0	2315 ± 4.7	**2210 ± 4.7**
Backlog penalties	2827.3 ± 5.24	285.3 ± 1.70	263.8 ± 1.55	**251.8 ± 1.74**
Wastage	**44.9 ± 0.16**	45.8 ± 0.02	45.9 ± 0.02	45.9 ± 0.02
CSL	95.88% ± 0.007	99.50% ± 0.003	99.54% ± 0.004	**99.58% ± 0.004**

Table 6.5: Profit, costs, customer service level (CSL), and other performance characteristics for the three FCPs with process duration fixed to 60 days as well as the standard heuristic (BSP0) solution. Mean ± std. err. are listed of 50 runs each and values reported are in RMU apart from CSL values. Statistically best values are highlighted in **bold**.

	BSP0	FCP1A	FCP2A	FCP3A
Profit	179015 ± 20.9	186544 ± 14.9	188044 ± 41.6	**189445 ± 3.5**
Revenue	214690 ± 17.9	221708 ± 21.5	222529 ± 24.3	**222816 ± 6.1**
Total costs	35676 ± 5.7	35164 ± 33.4	34485 ± 46.5	**33371 ± 6.7**
Seed	**179.0 ± 0.04**	185.3 ± 0.12	184.8 ± 0.15	181.4 ± 0.03
USP	**8059 ± 0.9**	8346 ± 5.4	8309 ± 8.3	8161 ± 1.5
Replacement ATF filters	10.3 ± 0.09	10.9 ± 0.02	10.7 ± 0.02	10.6 ± 0.01
Cell culture setup	**1064.6 ± 0.22**	1100.0 ± 0.68	1096.5 ± 0.96	1077.7 ± 0.20
DSP	**21022 ± 2.5**	21736 ± 13.7	21644 ± 19.9	21281 ± 4.6
Changeover	**478.0 ± 0.24**	1091.9 ± 12.17	1064.4 ± 7.94	1100.0 ± 2.26
Storage	3065 ± 1.0	3114 ± 25.6	2815 ± 29.6	**2291 ± 7.0**
Backlog penalties	2827.3 ± 5.24	642.8 ± 7.82	419.4 ± 10.03	**310.8 ± 1.78**
Wastage	**44.9 ± 0.16**	47.8 ± 0.09	47.2 ± 0.13	45.9 ± 0.02
CSL	95.88% ± 0.007	98.96% ± 0.009	99.36% ± 0.009	**99.46% ± 0.004**

Table 6.6: Profit, costs, customer service level (CSL), and other performance characteristics for the three BSPs with process durations optimised by the GA compared to the best performing BSP and FCP with fixed process duration. Policy names with a suffix of 'A' are for optimisations with process duration fixed to 60 days while those with suffix 'B' have process duration optimised by the EA. Mean $\pm$ std. err. are listed of 50 runs each and values reported are in RMU apart from CSL values. Statistically best values are highlighted in **bold**.

	FCP3A	BSP3A	BSP1B	BSP2B	BSP3B
Profit	189445 ± 3.5	189711 ± 1.7	189972 ± 6.3	190016 ± 13.9	**190125 ± 1.9**
Revenue	222816 ± 6.1	223075 ± 7.5	223101 ± 9.2	223123 ± 9.4	**223240 ± 7.3**
Total costs	33371 ± 6.7	33363 ± 8.0	33129 ± 6.8	**33107 ± 10.2**	33116 ± 7.2
Seed	181.4 ± 0.03	181.9 ± 0.03	182.2 ± 0.72	**178.8 ± 0.86**	185.8 ± 0.36
USP	8161 ± 1.5	8181 ± 1.5	8148 ± 8.3	**8110 ± 9.5**	8178 ± 5.0
Replacement ATF filters	**10.6 ± 0.01**	**10.6 ± 0.01**	11.1 ± 0.11	11.5 ± 0.09	11.3 ± 0.07
Cell culture setup	1077.7 ± 0.20	1080.7 ± 0.21	1065.7 ± 4.34	**1043.1 ± 4.87**	1077.0 ± 2.68
DSP	21281 ± 4.6	21342 ± 4.2	**21248 ± 3.6**	21282 ± 5.7	21256 ± 4.0
Changeover	**1100.0 ± 2.26**	1150.1 ± 1.11	1211.1 ± 4.13	1150.5 ± 6.02	1249.5 ± 1.63
Storage	2291 ± 7.0	2210 ± 4.7	2050 ± 8.2	2102 ± 12.6	**2002 ± 6.1**
Backlog penalties	310.8 ± 1.78	251.8 ± 1.74	243.5 ± 2.26	235.3 ± 3.04	**197.8 ± 2.06**
Wastage	45.9 ± 0.02	**45.9 ± 0.02**	46.9 ± 0.39	48.4 ± 0.36	46.4 ± 0.19
CSL	99.46% ± 0.004	99.58% ± 0.004	99.60% ± 0.004	99.61% ± 0.004	**99.66% ± 0.004**

Table 6.7: Profit, costs, customer service level (CSL), and other performance characteristics for the three FCPs with process durations optimised by their EA compared to the best performing BSP and FCP with fixed process duration. Policy names with a suffix of 'A' are for optimisations with process duration fixed to 60 days while those with suffix 'B' have process duration optimised by the EA. Mean $\pm$ std. err. are listed of 50 runs each and values reported are in RMU apart from CSL values. Statistically best values are highlighted in **bold**.

	FCP3A	BSP3A	FCP1B	FCP2B	FCP3B
Profit	189445 ± 3.5	189711 ± 1.7	186585 ± 127.1	189234 ± 39.5	**189910 ± 3.3**
Revenue	222816 ± 6.1	**223075 ± 7.5**	221389 ± 73.8	222733 ± 25.8	223047 ± 6.1
Total costs	33371 ± 6.7	33363 ± 8.0	34804 ± 99.5	33499 ± 17.2	**33137 ± 7.2**
Seed	181.4 ± 0.03	181.9 ± 0.03	**172.2 ± 3.72**	188.7 ± 0.84	179.0 ± 0.72
USP	8161 ± 1.5	8181 ± 1.5	8128 ± 36.3	8231 ± 10.2	**8102 ± 8.4**
Replacement ATF filters	**10.6 ± 0.01**	**10.6 ± 0.01**	13.5 ± 0.36	11.1 ± 0.12	11.5 ± 0.07
Cell culture setup	1077.7 ± 0.20	1080.7 ± 0.21	**1004.9 ± 21.24**	1096.9 ± 5.16	1041.7 ± 4.33
DSP	21281 ± 4.6	21342 ± 4.2	21602 ± 44.8	21293 ± 6.7	**21265 ± 4.3**
Changeover	**1100.0 ± 2.26**	1150.1 ± 1.11	1119.2 ± 21.22	1230.2 ± 6.92	1180.9 ± 4.55
Storage	2291 ± 7.0	2210 ± 4.7	2998 ± 96.7	2152 ± 13.8	**2107 ± 10.6**
Backlog penalties	310.8 ± 1.78	251.8 ± 1.74	721.5 ± 21.33	357.8 ± 8.64	**254.8 ± 1.78**
Wastage	**45.9 ± 0.02**	**45.9 ± 0.02**	62.6 ± 3.41	**46.2 ± 0.45**	47.7 ± 0.25
CSL	99.46% ± 0.004	**99.58% ± 0.004**	98.83% ± 0.035	99.44% ± 0.012	99.57% ± 0.003

cies that also optimised the process run time. First, one can observe that the FCP policies — apart from FCP3B — are always worse than the base stock policies regardless of process run time optimisation. This suggests that utilising a rigid fixed schedule in an uncertain environment is sub-optimal compared to policies reactive to changes. In general, the 'B' policy variants are better than their corresponding 'A' counterparts across the board. The only exception to this is FCP1 which does not seem to show improvement when optimising the run-times as well. This is likely due to the FCP1B expending its computation budget before fully converging. On the other hand, the greatest improvement observed from optimising the run-times is seen going from FCP2A to FCP2B.

The ranking of the 'B' policies carries over from the 'A' policies. For the BSPs, BSP3B is best followed by BSP2B then BSP1B. The pattern is similar for the FCP policies: FCP1B is worst, FCP2B performs better than that, and FCP3B is best. Because the run-times are different between the policies and among the individual solutions within each policies, it is harder to draw insights or conclusions based on the performance characteristics than with policies with standard run-times.

6.6.3 Policy with Flexible Process Duration

To ascertain the benefit of a policy that allows for flexible process run-times, the ANN policy was compared with the previously evaluated polices.

The implementation of this policy, as previously described, means that on every day, a batch can be cut short even if it was just started the previous day. To exclude very bad solutions, a constraint on a minimum process run-time was enforced. This meant that upon making a decision to start a batch, no new decisions could be made until the cell culture reached a specified duration milestone. If this lower bound is small, it means that there is a larger search space which increases the scope for flexibility but also makes the search for a good solution harder. A large lower bound improves computation time but means some batches may be unnecessarily long. As a result, some duration lower bounds were tested: lower bounds of 14, 20, 30, 40, and 45 days for all of the products, and 'optimised' lower bounds where the EA searched for individual lower bounds for each product in addition to the neural network weights. Table 6.8 reports the results of that experiment and shows that setting the lower bound to 30 days for all products is best, so that strategy was used for the rest of the following analysis.

Although the ANN policy is substantially better than the standard policy, it

Table 6.8: Profit performance (in RMU) for tuned ANN policies utilising different minimum process run-times.

	Process run-time lower bound					
	14 days	20 days	30 days	40 days	45 days	'Optimised'
Profit	188983 ± 365.6	189315 ± 39.6	189421 ± 42.3	188842 ± 100.9	188709 ± 48.1	189147 ± 31.5

Table 6.9: Profit, costs, customer service level (CSL), and other performance characteristics for the ANN, the best of each of FCP-B and BSP-B, and the overall worst 'B' policy variant. Mean ± std. err. are listed for 50 runs each and values reported are in RMU apart from CSL values. Statistically best values are highlighted in **bold**.

	FCP1B	FCP3B	BSP3B	ANN
Profit	186585 ± 127.1	189910 ± 3.3	**190125 ± 1.9**	189421 ± 42.3
Revenue	221389 ± 73.8	223047 ± 6.1	**223240 ± 7.3**	222833 ± 28.2
Total costs	34804 ± 99.5	**33137 ± 7.2**	**33116 ± 7.2**	33412 ± 20.5
Seed	**172.2 ± 3.72**	179.0 ± 0.72	185.8 ± 0.36	200.6 ± 0.38
USP	**8128 ± 36.3**	**8102 ± 8.4**	8178 ± 5.0	8367 ± 4.0
Replacement ATF filters	13.5 ± 0.36	11.5 ± 0.07	**11.3 ± 0.07**	**11.3 ± 0.14**
Cell culture setup	**1004.9 ± 21.24**	1041.7 ± 4.33	1077.0 ± 2.68	1160.7 ± 2.62
DSP	21602 ± 44.8	21265 ± 4.3	21256 ± 4.0	**21223 ± 7.8**
Changeover	**1119.2 ± 21.22**	1180.9 ± 4.55	1249.5 ± 1.63	1385.2 ± 4.96
Storage	2998 ± 96.7	2107 ± 10.6	2002 ± 6.1	**1843 ± 13.4**
Backlog penalties	721.5 ± 21.33	254.8 ± 1.78	**197.8 ± 2.06**	346.2 ± 9.28
Wastage	62.6 ± 3.41	47.7 ± 0.25	**46.4 ± 0.19**	**46.8 ± 0.49**
CSL	98.83% ± 0.035	99.57% ± 0.003	**99.66% ± 0.004**	99.52% ± 0.012

fails to compete with the other optimised BSPs — even the BSP policies where the process run-time is not optimised. Table 6.9 compares the performance of the ANN policy and other policies with optimised process run-times. It is slightly worse than FCP3A but it outperforms the FCP1 and FCP2 policies. By inspecting the expected profit values, the simulated performance appears to vary much more than the other policies (apart from the two FCP1 policies). This policy starts on average more batches than the others compared in Table 6.9 and also accrues more changeover costs; if the ratio between DSP costs and seed costs (or cell culture setup) for each policy is compared it suggests that in addition to having more batches, the ANN has shorter batches.

6.6.4 Tuned Policy Parameters

Table 6.10 lists the parameters in the optimised base-stock policies as well as the calculated parameters for the standard base stock policy and Table 6.11 reports the parameters for the FCP policies. This data represents the overall best solution for

each policy from all the EA runs.

The striking difference between the standard policy and the optimised BSPs is that the standard policy has much higher order-up-to levels. Coupled with slightly lower reorder points, this means that batches are more likely to be ordered in campaigns of the same product instead of the facility switching more frequently between products. Conversely, the optimised policies have their inventory parameters in much narrower ranges which means that campaigns are more likely to have just one batch — the amount produced per batch is much larger than the difference between the inventory parameters — and subsequently more product changeover(s). In fact, with BSP1, the best policies have $Y^{(1)}$ and $Y^{(2)}$ parameter values almost equal which suggests that the second parameter is not particularly useful. This would mean that the policy could be replaced by one with just $Y^{(1)}$ (much like how BSP3 only has $Y^{(5)}$) which would require less effort tuning it — i.e., if it were to be run again but with one parameter it should give faster convergence but similar results. This assumption has been confirmed experimentally.

The reason that the optimised $Y^{(1)}$ values are higher than that of the standard policy is most likely due to the fact that the optimised policies implicitly account for the fact that the product changeovers can only happen after the end of a batch of a predetermined run time. This means it can maintain higher levels of safety stock and reduce the likelihood of stock-outs. And with the narrower ranges of the inventory parameters, more frequent product changeovers help avoid product inventories from falling critically low. The standard base-stock policy is based on heuristics that do not model batch or semi-continuous production; this highlights how previous approaches to lot-sizing approaches don't easily apply to biopharmaceutical manufacturing contexts.

The best FCP1 solutions also highlight that campaigns of multiple batches are seemingly sub-optimal at this problem load. Though these policies can schedule consecutive batches of the same product, the best solutions do not implement that strategy. The best FCP1 sequences are longer than the other fixed cycle policies partly because the set of possible elements in the sequence is larger and the sequences attempt to encode idle times and length of manufacturing campaigns. The fixed cycle policies with standard process run times tend to have $p1$ less frequently in their sequences compared to the other products. This is due to its low demand — all of a year's demand can be produced in a 60 day batch. Apart from FCP1B, the best FCPB policies have sequences that are shorter than their 'A' counterparts —

Table 6.10: Policy parameters for $p1$-$p3$ in each of the base-stock policies and the ANN policy. The best solutions (i.e., the best solution out of all EA runs for each tuned policy) are reported. For the ANN, the mean $\pm$ std. err. of batch run-times over several simulations is reported; the run-times for batches that were terminated due to process failures or those that had not been completed by the end of the simulation horizon were not included in these calculations.

	$p1$					
	$Y^{(1)}$	$Y^{(2)}$	$Y^{(3)}$	$Y^{(4)}$	$Y^{(5)}$	B
BSP0	6.2	52.5	–	–	–	60
BSP1A	16.4	16.7	–	–	–	60
BSP1B	18.7	19.9	–	–	–	47
BSP2A	10.5	27.5	15.2	16.2	–	60
BSP2B	12.8	20.3	15.0	17.1	–	43
BSP3A	–	–	–	–	16.3	60
BSP3B	–	–	–	–	16.3	43
ANN	–	–	–	–	–	33.7 ± 0.00
	$p2$					
	$Y^{(1)}$	$Y^{(2)}$	$Y^{(3)}$	$Y^{(4)}$	$Y^{(5)}$	B
BSP0	11.1	93.6	–	–	–	60
BSP1A	28.8	28.9	–	–	–	60
BSP1B	25.1	25.5	–	–	–	58
BSP2A	23.5	38.4	25.3	26.1	–	60
BSP2B	22.4	31.5	23.4	25.3	–	59
BSP3A	–	–	–	–	23.8	60
BSP3B	–	–	–	–	29.2	51
ANN	–	–	–	–	–	49.4 ± 0.01
	$p3$					
	$Y^{(1)}$	$Y^{(2)}$	$Y^{(3)}$	$Y^{(4)}$	$Y^{(5)}$	B
BSP0	10.7	77.5	–	–	–	60
BSP1A	23.8	23.9	–	–	–	60
BSP1B	21.0	21.6	–	–	–	65
BSP2A	19.1	39.9	21.3	21.5	–	60
BSP2B	6.6	36.4	18.8	19.9	–	74
BSP3A	–	–	–	–	28.8	60
BSP3B	–	–	–	–	19.2	79
ANN	–	–	–	–	–	71.9 ± 0.02

Table 6.11: The optimal sequence and process run times for the fixed cycle policies overall from all EA runs.

	Sequence	$Y_{p1}^{(2)}$	$Y_{p2}^{(2)}$	$Y_{p3}^{(2)}$	B_{p1}	B_{p2}	B_{p2}
FCP1A	1-3-2-3-0-3-0-2	–	–	–	60	60	60
FCP1B	2-0-1-3-0-2-3-0-1-0-2-3	–	–	–	42	48	73
FCP2A	1-3-2-3-2-3	45.3	25.0	12.5	60	60	60
FCP2B	2-1-3	15.3	40.6	22.5	37	55	87
FCP3A	1-2-3-1-2-3-2-3-2	17.7	24.3	26.1	60	60	60
FCP3B	2-3-2-1-2-3	21.4	22.3	22.9	47	58	74

an extreme being FCP2B which is just a simple permutation of the three products.

Observed solutions for process run times are based on striking a balance between the process yields and demand forecasts of each product as well as the increasing risk of process failure with longer process duration. It is intuitive, based on the specifics of the case study, that $p1$ would have a shorter process run time than the other products because it has much lower demand. By the same token, it makes sense that $p3$ has a longer process run time than $p2$ because it has similar demand but the yield of its manufacturing process is a third lower.

There are no major differences in the optimised process run times between the different base-stock policies apart from $p2$ in BSP3B and $p3$ in BSP1B where the run times are significantly lower than that of the other tuned BSPs. The same cannot be said for the FCPs where the optimised run times vary substantially between the best policy solutions. The ANN policy, as predicted by inspecting the cost breakdowns, has significantly lower expected run times compared to all the other policies. This indicates that this policy suffers from reduced efficiency — more time is spend setting up batches or changing over between different products which reduces the facility's productivity.

All the optimised durations deviate from the 'standard' process run time of 60 days especially with $p1$ and $p3$. It is interesting to note that the optimised policies tend to have process run times that cannot produce all of the expected demand for a year in a single batch even if it is possible — process durations of 53 and 88 days would suffice for $p1$ and $p2$ respectively. As previously mentioned, the factors determining these decisions are the need to mitigate process failure and the ability to changeover to other critical products. It is not clear which role each of those factors play in each case but it is fair to say that mitigating process failure is more influential for $p2$ than it is for $p1$.

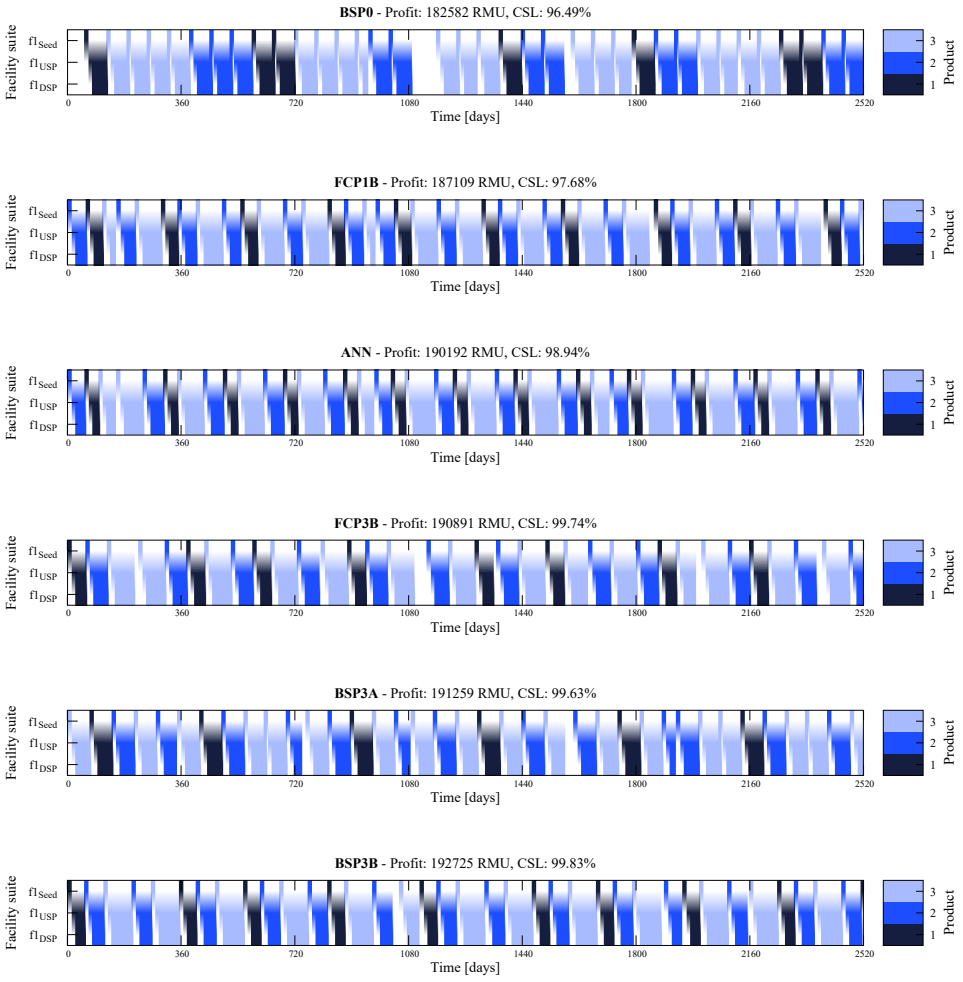


Figure 6.3: Exemplar schedules from a single simulation run with the same random number generator seed for the standard, FCP1B, ANN, FCP3B, BSP3A, and BSP3B scheduling policies.

6.6.5 Production Schedule(s) for the Facility

Although the Gantt charts in Figure 6.3 are just for one scenario and simulation run, many of the points previously discussed are illustrated here. The Gantt charts shown are for BSP0 and the best performing solutions for FCP1B, FCP3B, BSP3A, BSP3B, and the ANN.

BSP0 schedules batches of the same product together so that campaigns have multiple batches and minimises changeover costs as a result. As a result of these sustained campaigns, the inventory will be built up and incur higher storage costs.

Moving from BSP0 to BSP3A and then to BSP3B, the number of multi-batch campaigns decreases with BSP3B not scheduling any two batches of the same product together in this particular scenario — this also applies to FCP1B and the ANN. For FCP1B the sequence does not include any consecutive batches of the same product. With increasing number of product changeover, there is more production time lost due to the fact that changeover between batches of different products requires more setup time than the turnaround between two batches of the same product. This is why the BSP0 chart appears to be less utilised. Scheduling of idle time on the facility are decisions directly available to the ANN and FCP1B policies, but it is these policies that actually have the least amount of idle time. In the case of the ANN, there is none scheduled at all.

What these charts also illustrate is a disadvantage of FCP1B: although it is able to dictate a schedule ahead-of-time, it is not able to find a 'good' decision to make when the environment changes. This is true especially in the event of process failure where it will move on to the next batch in the sequence regardless of any critically low inventory levels. On the other hand, while the BSPs cannot fix a schedule in advance or determine an absolute decision to make when process failure occurs, they can implicitly react to changes by deciding at each point whether to continue with that product, go idle or switch to another product. Similarly, although FCP3B is based on a 'fixed' sequence which should help mitigate planning nervousness, it is often the case that production skips ahead by one or more elements in the sequence. In this respect, this policy is more alike to the BSP policies than a rigid schedule or manufacturing sequence.

6.6.6 EA Performance and Statistical Analysis

Statistical testing on the performance of the policies was carried out, the results of which are presented in Table 6.12 on the next page. It contains a matrix of p-values from Mann-Whitney U tests comparing the profits of the final solutions from each policy. It indicates that in terms of performance, the assertion that ANN $<$ BSP1A $<$ BSP2A $<$ BSP3A $<$ BSP1B $<$ BSP2B $<$ BSP3B is statistically significant (at the level of 0.05) when looking at pairwise comparisons between each policy and the rest. Similarly, FCP1 $<$ FCP2A $<$ FCP2B $<$ ANN $<$ FCP3A $<$ FCP3B. It also shows that the difference between both FCP1 policies is not significant.

Figure 6.4 on page 131 shows the convergence of the optimisations of the ANN with different minimum process run times and Figure 6.5 on page 132 shows that

Table 6.12: Matrix of the observed significance level of Mann-Whitney tests comparing the means profit of the final solutions from each policy. The p-values are for the test that H_0 : policy$_1 \not> $ policy$_2$ and H_1 : policy$_1 >$ policy$_2$.

policy$_1$	policy$_2$												
	FCP1A	FCP1B	FCP2A	FCP2B	FCP3A	FCP3B	ANN	BSP1A	BSP1B	BSP2A	BSP2B	BSP3A	BSP3B
FCP1A	–	0.824	1	1	1	1	1	1	1	1	1	1	1
FCP1B	0.176	–	1	1	1	1	1	1	1	1	1	1	1
FCP2A	<0.001	<0.001	–	1	1	1	1	1	1	1	1	1	1
FCP2B	<0.001	<0.001	<0.001	–	1	1	1	1	1	1	1	1	1
FCP3A	<0.001	<0.001	<0.001	<0.001	–	1	<0.001	1	1	1	1	1	1
FCP3B	<0.001	<0.001	<0.001	<0.001	<0.001	–	<0.001	<0.001	1	<0.001	1	<0.001	1
ANN	<0.001	<0.001	<0.001	<0.001	1	1	–	1	1	1	1	1	1
BSP1A	<0.001	<0.001	<0.001	<0.001	<0.001	1	<0.001	–	1	1	1	1	1
BSP1B	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	–	<0.001	1	<0.001	1
BSP2A	<0.001	<0.001	<0.001	<0.001	<0.001	1	<0.001	<0.001	1	–	1	1	1
BSP2B	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	–	<0.001	1
BSP3A	<0.001	<0.001	<0.001	<0.001	<0.001	1	<0.001	<0.001	1	<0.001	1	–	1
BSP3B	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	–

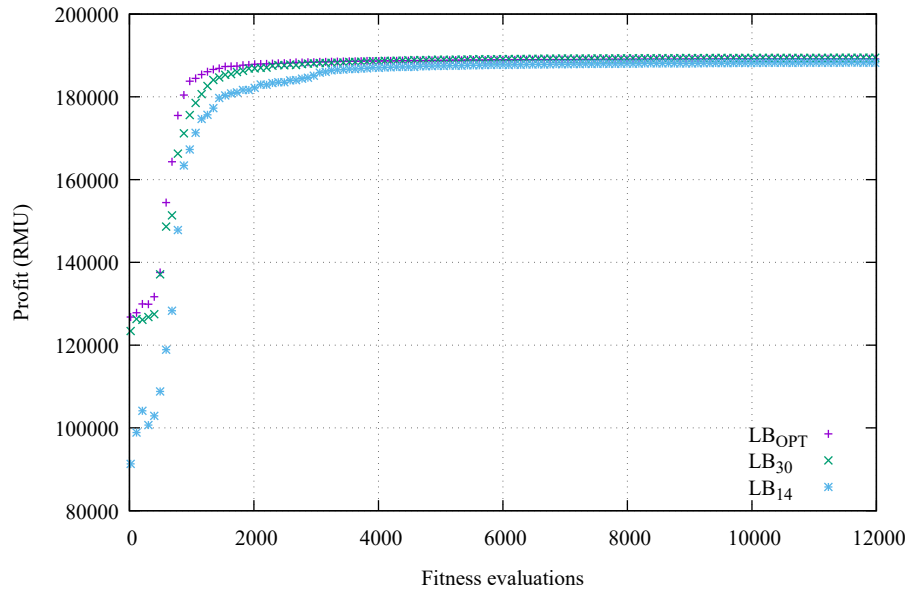


Figure 6.4: Convergences of the ANN optimisations runs with different lower bounds on process run time. LB_{14} = minimum run time of 14 days for all products; LB_{30} = minimum run time of 30 days for all products; LB_{OPT} = minimum run time optimised for each product.

for the BSPs. Across all the BSP policies both optimisation algorithms tested (i.e., the GA and the CMA-ES) deliver essentially the same quality of final solutions. However, the more state-of-the-art CMA-ES converges faster than the GA. This gets more pronounced the more decision variables the problem has — e.g., BSP2 which has four inventory parameters per product — and with the policies that have to optimise process durations in addition to the inventory parameters. In addition, these EAs were compared with random search and it is shown that the EAs perform significantly better on most of the policies. The exceptions to this are the BSP3 policies where there is no statistical difference between the performance of random search and the EAs. This suggests that BSP3, by virtue of its in-built forecasting heuristic, is able to make good scheduling decisions during simulation. It also helps that the policy has very few decision variables to optimise (six at most and as few as three when process run times are fixed) so the search space is smaller.

To test the benefit of the forecasting heuristic, the BSP3 policies were compared with BSP1 policies where $Y^{(1)} = Y^{(2)}$. This constraint is to ensure that both policies work in the same way except that the BSP3 policies break ties with the forecasting heuristic and the BSP1 policies break ties by comparing run-out times.

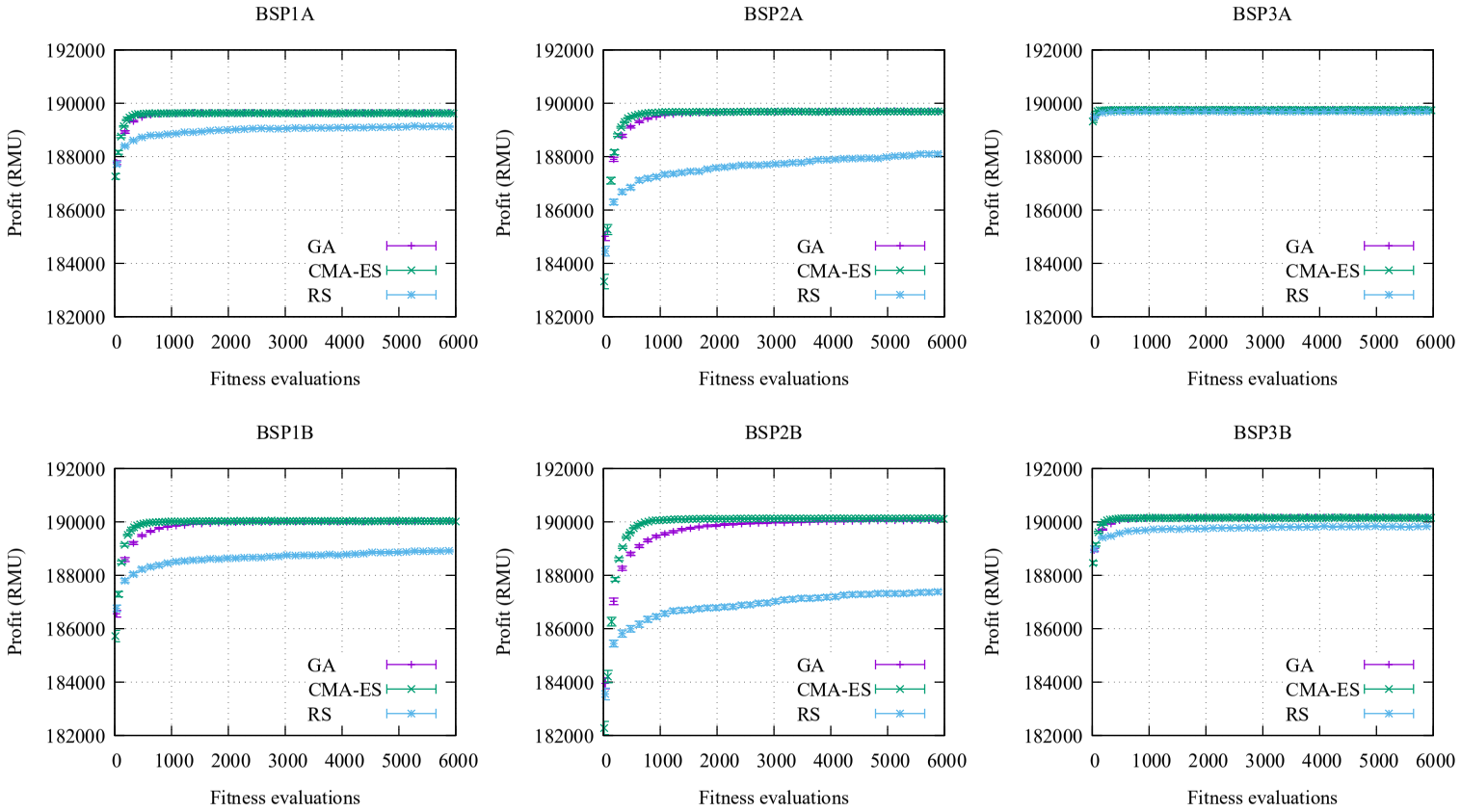


Figure 6.5: Convergences of the all the BSP optimisations runs with the genetic algorithm (GA), CMA-ES, and with random search (RS). Errorbars indicate standard error.

So for both the BSP3 and the constrained BSP1, 2,000 policies were randomly generated and evaluated. The mean performance ($\pm$ standard error) of the BSP3 policies was 167933 ± 850 RMU and the performance of the constrained BSP1 policies was 166709 ± 855 RMU. A two-tailed Mann-Whitney U test was performed on the policies and indicated that the increase in performance of the BSP3 policies was statistically significant (observed significance level of <0.001).

The ANN convergence plot demonstrates how much more of a computation effort is required to find a good policy — at least when compared with the BSPs. Lower bounds on the process run time significantly affect the speed of convergence; while a lower bound of 14 days on average converges to values close to the others, it takes substantially more fitness evaluations to get there. While more parameters to tune usually means slower convergence, the optimised lower bounds converge with the least fitness evaluations with lower bound of 30 days slightly worse. This is due to the ability to exclude bad policies which schedule inefficiently short batches.

In Table 6.13 on the following page, the computation times for the different EA optimisations are reported. These values are from one run each of the hyper-heuristic for each policy type. This experiment was carried out on an Intel® Core™ i5-7500 Quad-core 3.40GHz processor, with 8GB RAM running Microsoft Windows 10 64-bit. In cases where both the GA and CMA-ES are used, generally, the latter is slower. This is due to the updates of the covariance matrix carried out within the CMA-ES algorithm. The considerably longer CPU times needed by the ANN can be explained by two reasons. First, the ANN tuning was given a budget of fitness evaluations that is twice that of the BSPs and FCPs. Secondly, the implementation of this policy dictates that decisions are made almost every day in the simulation which requires querying the neural network any time a decision is required. This is a significantly larger computation load. For example, if all process run times are 60 days, a base-stock policy would make decisions on around 40 days of the seven-year simulation. In contrast, an ANN policy with run time lower bound of 14 days would need to make over $1,000$ days of decisions in order to have its process durations also be 60 days in a simulation horizon of the same length. This explains why increasing the run time lower bounds (or allowing them to be optimised) gives a considerable increase in speed. In fact, the CPU time of the ANN with optimised lower bounds, if interpolated to 6,000 fitness evaluations, is comparable to the FCPs (and to a lesser extent the BSPs) which use the CMA-ES.

Table 6.13: CPU times (in seconds) for a single run of each EA for the scheduling policies. $ANN_{LB\text{-}14}$ and $ANN_{LB\text{-}30}$ are ANN policies with minimum process run time fixed to 14 days and 30 days respectively. $ANN_{LB\text{-}OPT}$ is ANN policy with optimised process run time lower bounds for each product.

	GA	CMA-ES
BSP1A	817	906
BSP1B	812	830
BSP2A	839	793
BSP2B	839	952
BSP3A	836	784
BSP3B	830	937
$ANN_{LB\text{-}14}$	–	6474
$ANN_{LB\text{-}30}$	–	4590
$ANN_{LB\text{-}OPT}$	–	1923
FCP1A	822	769
FCP1B	818	924
FCP2A	808	931
FCP2B	804	929
FCP3A	801	943
FCP3B	797	940

Overall, these analyses have shown that it is beneficial to tune parameters for the scheduling policies instead of relying on estimated parameters or a fixed schedule or sequence of batches. Furthermore, optimising process run times offers additional advantages as it allows the policy to schedule batches so that product changeover can occur when inventory falls to critical levels and also select run times with efficient productivities and an acceptable risk of process failure. Thirdly, there does not appear to be any benefit of utilising flexible process run times compared to the BSPs or the more complex of the fixed cycle policies — at least in this scenario and specific neural network implementation. In addition, intelligent choice of process run time lower bounds may help improve speed of convergence and minimise computation time. Finally, the choice of optimisation algorithm, for the BSPs at least, does not play a significant effect in the quality of final solution. However, as the number of decision variables increases, a more efficient algorithm such as CMA-ES, can contribute to a faster convergence (with regards to fitness evaluations).

6.6.7 Evaluating the Sensitivity of Optimised Policy Solutions

The case study that the policies have been optimised for and evaluated on has a demand forecast which is stochastic but is a known and defined distribution. In reality, it is not easy to accurately predict or forecast demand in advance — which is why the demand is defined as a probability distribution with a mean based on targets and market research (and variance based on estimated margin of error in predictions). However, as actual demand is being realised it can be difficult to determine whether the observed demand actually is sampled from the same distribution as the one predicted. So the observed demand could potentially be from a distribution with a different mean and/or variance. This section aims to determine the behaviour of the previously obtained solutions on slightly different problem instances to the one they were trained on.

With those considerations, some of the policies were tested at different demand scales. Specifically, the best solution from each of FCP3B, BSP1B, BSP2B, BSP3B, and the ANN as well as the previously estimated BSP0 solution were evaluated over $20,000$ simulations on demand load cases in the range of $\pm 15\%$ of the standard case. As before, the demand due in each year is stochastic and described by a Normal distribution, $N(\mu, (0.025\mu)^2)$. Apart from the change in demand load, all the other model and problem parameters were unchanged from those in the case study description or those used in Section 6.6.2.

Results of the sensitivity tests are shown in Figure 6.6 on the next page where the performances are compared relative to BSP0. From the figure one can observe that the best solution of BSP3B is better than the rest at the standard demand case with BSP2B, BSP1B, FCP3B, and the ANN following in that order. However, the BSP3B solution does not maintain its ranking when looking at demand cases below and above the standard case, dropping into second place in cases 85% to 95% of the standard demand while BSP2B replaces it at the top. At the 95% demand case, though the mean of BSP2B is greater than that of BSP3B, the difference is not statistically significant to say that either policy is better than the other. The performance of the ANN falls away rather quickly either side of the standard demand case to the point that at 85% it is actually worse than the standard policy. BSP2B remains the best performing policy solution above the standard demand case — closely followed by FCP3B — apart from in the most loaded problem instance where it falls into third place. In that case, BSP3B and BSP1B perform similarly and the difference between them is not statistically significant. At this demand

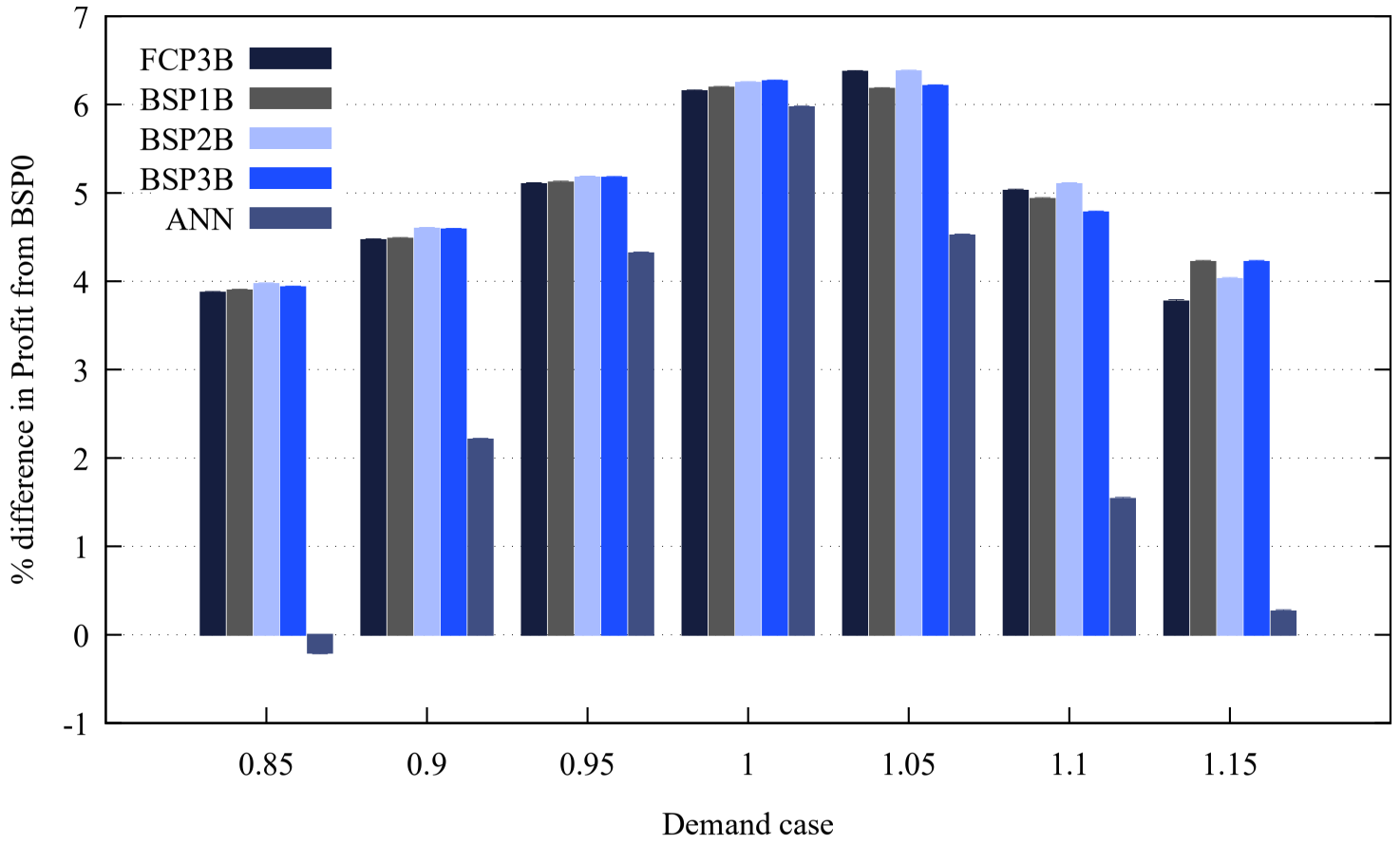


Figure 6.6: Sensitivity of the optimised policies to demand overestimates (i.e., 0.85 to 0.95 demand cases) and demand underestimates (i.e., 1.05 to 1.15 demand cases) relative to the standard heuristic, BSP0.

case, FCP3B is only better than the ANN.

Relative to the standard policy, which has its parameters heuristically chosen, the performances of the tuned policies generally decline as one moves away from the standard demand case. For the tuned BSPs and FCP3B this effect is more pronounced at the demand loads less than the standard case. For the BSPs, a partial explanation for this trend is that at less loaded problems, sub-optimal decisions incur less harsh penalties. For example, the main driver for the difference between BSP0 and the tuned policies at the standard demand case is backlog penalties. So with lower observed demand, clearly this becomes less important and the difference decreases. A more general explanation applicable to all policies is that tuning it exploits the specific structure and characteristics of the problem instance it is trained on which means it loses its applicability to moderately different or more general problems — this is referred to as *overfitting*.

Overall what can be observed from this testing is that BSP2B is the best performer across all slightly different demand scale instances apart from the most loaded problem. This is despite BSP3B being best at the problem it was trained on. The implication is that one will need to check the sensitivity of best solutions to demand case changes instead of automatically taking the best policy solution trained on the current problem.

6.7 Summary

This chapter considered the stochastic economic lot scheduling problem (SELSP) in the context of a biopharmaceutical manufacturing scenario consisting of multiple products on a single facility utilising semi-continuous perfusion processes that are prone to various types of process failure which have significant operational consequences. To deal with the challenges that this problem poses, a simulation-optimisation approach was developed and implemented.

First, to complement the previously developed custom discrete-event simulation framework, a few dynamic scheduling policies were adapted from the literature to fit the problem being investigated. In addition to this, a novel policy with a forecasting heuristic was proposed. These policies were then tuned on a synthetic case study using evolutionary algorithms (EAs) in what amounts to a hyper-heuristic search.

Evaluation of these policies and further comparison with a heuristically determined standard policy as well as policies based on a fixed sequence demonstrated the benefit of tuning parameters and utilising policies that use the current state of the scheduling environment to make decisions. Further tuning of process run times led to improved performance as this enables better lot-sizing decisions which may allow hedging against process failure by utilising a shorter run time.

In addition to these policies, a flexible policy based on an Artificial Neural Network (ANN) representation was proposed and compared with tuned policies. This comparison showed that, in the scenario examined in this chapter, the implementation of a policy capable of flexible process run times did not improve on the performance of the best tuned fixed cycle policies and base stock policies.

Finally, it is recommended that a potential decision-maker evaluate the policies at different demand loads in order to determine their behaviour and sensitivity. This is because relative performance may differ when presented with slightly different problems than it was trained on.

Chapter 7

Conclusion and Future Work

7.1 Summary and Contribution of Thesis

Over the course of this thesis, deterministic and stochastic variants of lot sizing and scheduling problems have been considered within the context of the biopharmaceutical industry. The main contributions lie with the decision tools that were introduced here.

First, a meta-heuristic approach for a capacity planning problem for fed-batch processes was proposed. Compared to mixed-integer linear programming (MILP), it avoided simplifying assumptions on the discrete time periods and as a result is able to model reality more accurately. In addition, it demonstrated better scaling properties (computation time grows slower than the MILP with increasing problem size) and could easily be adapted to multi-objective problems with the implementation of a multi-objective evolutionary algorithm such as the non-dominated sorting genetic algorithm II (NSGA-II).

After that there was a change in focus to processes incorporating perfusion unit operations. Though there is existing literature on capacity planning models that include perfusion processes and there are tools evaluating the economics of utilising perfusion processes while taking into account their increased failure rates, there was no evidence of tools or models that focused on scheduling whilst also considering these processes' greater propensity for failure. In other words, no pre-existing literature attempted to determine optimal schedules for these perfusion processes whilst anticipating or expecting they would fail and asking the question of what the best decision(s) to make after process failure events.

The result of this was the development of a custom model to simulate semi-

continuous bioprocesses in a scheduling system which was sensitive to the consequences of failure events attributed to the perfusion cell culture and its cell retention system. This was used to evaluate process design decisions such as the cell culture run time and process configuration(s).

Eventually, this became part of a larger novel hyper-heuristic framework which was utilised to determine optimal process run times and scheduling decisions for single-product and multi-product facility scenarios. For the former, simple inventory replenishment-type strategies were examined and for the latter, rule representations included base-stock policies, fixed-cycle policies, and an Artificial Neural Network (ANN).

7.2 Future Work

There are several avenues for extending the work presented in this thesis. This shall be discussed in relation to the chapters in this thesis that cover the work to be extended. However, in general, the work in this thesis would benefit from further validation on more case studies and different problem instances.

7.2.1 Chapter 4

First, in reality, demand is estimated and uncertain, so the approach can be adapted to deal with stochastic problems. Since subsequent chapters have dealt with stochasticity in a dynamic fashion, the more suitable approach with this meta-heuristic is to conduct scenario analysis on the products or facilities in the scenario which are deemed to be critical or seemingly more susceptible to uncertainty.

This would also be a good juncture to investigate different instances of the problem solved in this work. This may involve smaller or larger problems or problems with different cost structures. For example, sequence-dependent setup costs and times could be incorporated into a different problem, as well as explicit costs associated with facility overheads, transportation, and using CMOs. In addition facility-build or retrofit decisions may be added to the set of decision variables. With regards to multiple objectives, it is possible to extend the optimisation to consider alternative or additional objectives (e.g., utilsation of in-house or CMO facilities). Suffice to say, there are numerous permutations of different problems to which the tool may be applied to within biopharmaceutical manufacturing.

Third, it may be useful to extend the splitting mechanism to increase the search

space so that it is more likely that the optimal schedule is indeed contained in the search space of the search algorithm. Finally, one might explore also other optimisation methodologies to solve this problem such as constraint programming (Laborie, 2009) or hybrid approaches (Blum and Raidl, 2016). In addition, it may be worth comparing this method to recent work by Jankauskas et al. (2019) which uses a GA approach on a similar biopharmaceutical problem by utilising a dynamic chromosome structure to directly encode the solution.

7.2.2 Chapter 5

The work in this chapter focuses on processes that utilise single-use technology in the upstream part of the process. As a result, there is scope to extend the model so that it considers the use of conventional stainless-steel equipment for its upstream processing (USP). This obviously would require a slightly different cost model since the current cost structure of the model is solely based on single use bioreactors (SUBs). This would mean including the costs of fixed capital investment (FCI), labour costs, and all direct and indirect costs in general.

Second, in order to assess the optimum process run times, the bioreactor volumes, chromatography column sizes and resin volumes, and the process design in general are fixed. This means that shorter process run times have an upper bound on productivity and longer process run times have a lower bound on productivity. In reality, the USP and DSP design can be optimised around a selected run time (by changing bioreactor volumes, column sizes, pooling strategy, etc.) to meet a productivity target. Process design is not trivial and it would be exhaustive to optimise process design for every possible run time and demand combination. However, it may be worthwhile exploring a few process run times with the process redesigned to examine what aspects of the process need to be re-optimised and also what the knock-on effect on scheduling performance is. Alternatively, commercial software that is designed for process design and simulation could be coupled with expensive-optimisation algorithms as another approach.

Finally, combining the previous points, existing literature on assessing the benefits (in fed-batch processes) of multiple reactors compared to a larger single reactor has rested on the trade-off between equipment costs of the reactors and consumable costs related to chromatography resins and viral filters in DSP. The investigation into process configuration in this chapter does not capture this as the costs of individual SUBs dominate and the DSP remains the same. However, since SUBs

are currently limited to 2000L, there is scope to compare a large stainless-steel perfusion process with multiple parallel 2000L SUBs (even if both DSPs are identical) as this not only captures the trade-off between frequent but smaller failures vs. fewer but more significant failures, but also the trade-off between conventional equipment and disposable equipment. Also, multiple reactor configurations can be compared where reactors share one common seed train vs. where each has its own.

7.2.3 Chapter 6

There are a few possible ways to extend the work presented in this chapter. The obvious one is to apply this method to different case studies. For example, a clinical manufacturing scenario will likely have less established processes and therefore probably higher error, failure rates, or a larger variance in probability distributions describing the stochastic processes in general. In addition and complementary to this, instances may have process yields and cell culture titres also uncertain and stochastic. In addition, problems that also mix different bioprocessing modes (i.e., fed-batch and perfusion cell cultures) together would represent a good extension to this work as it would introduce sequence-dependent changeover times.

Second, one can extend the problem tackled so that it is able to deal with multiple facilities. The scope of that problem is much larger and requires heuristics and/or policies that not only need to coordinate facilities in terms of what they manufacture at any time instant, but also potentially anticipate the future workload of critical facilities within the network. This will require careful consideration of strategies to deal with expensive evaluations and noise as the problem size increases such as meta- and surrogate-modelling (Jin and Branke, 2005; Branke, 2018). In fact, such methods could be used to manage the computation time of the optimisation procedure as it is now and are worth exploring even without the added computational burden and load of additional facilities.

Finally, there is potential to consider product attrition or project selection in an R&D perspective. The decision variables may be adapted to select an initial portfolio out of an initial set of alternatives with different attrition rates and market potential and the objective function would be to maximise expected net present value (NPV) or risk-adjusted NPV.

Bibliography

N. Allison and J. Richards. Current status and future trends for disposable technology in the biopharmaceutical industry. *Journal of Chemical Technology & Biotechnology*, pages 1283–1287, 2013.

R. Allmendinger, A. S. Simaria, and S. S. Farid. Efficient Discovery of Chromatography Equipment Sizing Strategies for Antibody Purification Processes Using Evolutionary Computing. In C. A. C. Coello, V. Cutello, K. Deb, S. Forrest, G. Nicosia, and M. Pavone, editors, *Parallel Problem Solving from Nature - PPSN XII SE - 47*, volume 7492 of *Lecture Notes in Computer Science*, pages 468–477. Springer Berlin Heidelberg, 2012.

R. Allmendinger, A. S. Simaria, R. Turner, and S. S. Farid. Closed-loop optimization of chromatography column sizing strategies in biopharmaceutical manufacture. *Journal of Chemical Technology & Biotechnology*, pages 1481–1490, 2013.

B. Almada-Lobo and R. J. James. Neighbourhood search meta-heuristics for capacitated lot-sizing with sequence-dependent setups. *International Journal of Production Research*, 48(3):861–878, 2010.

B. Almada-Lobo, D. Klabjan, M. Antónia carravilla, and J. F. Oliveira. Single machine multi-product capacitated lot sizing with sequence-dependent setups. *International Journal of Production Research*, 45(20):4873–4894, 2007.

C. Almeder. A hybrid optimization approach for multi-level capacitated lot-sizing problems. *European Journal of Operational Research*, 200(2):599–606, 2010.

M. A. Aloulou, A. Dolgui, and M. Y. Kovalyov. A bibliography of non-deterministic lot-sizing models. *International Journal of Production Research*, 52(8):2293–2310, 2014.

S. Amaran, N. V. Sahinidis, B. Sharda, and S. J. Bury. Simulation optimization: a review of algorithms and applications. *Annals of Operations Research*, 240(1): 351–380, 2016.

J. E. Baker. Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the second international conference on genetic algorithms*, volume 206, pages 14–21, 1987.

T. A. Bibila and D. K. Robinson. In pursuit of the optimal fed-batch process for monoclonal antibody production. *Biotechnology Progress*, 11(1):1–13, 1995.

C. Bierwirth and D. C. Mattfeld. Production scheduling and rescheduling with genetic algorithms. *Evolutionary Computation*, 7(1):1–18, 1999.

C. Bierwirth, D. C. Mattfeld, and H. Kopfer. On permutation representations for scheduling problems. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature — PPSN IV*, pages 310–318, 1996.

J. R. Birch. Cell Products—Antibodies. In *Encyclopedia of Cell Technology*. John Wiley & Sons, Inc., 2003.

G. R. Bitran and H. H. Yanasse. Computational Complexity of the Capacitated Lot Size Problem. *Management Science*, 28(10):1174–1186, 1982.

C. Blum and G. R. Raidl. *Hybrid Metaheuristics*. Artificial Intelligence: Foundations, Theory, and Algorithms. Springer International Publishing, 2016. ISBN 978-3-319-30882-1.

C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, pages 268–308, 2003.

J. Branke. Simulation Optimisation: Tutorial. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '18, pages 745–772, 2018.

J. Branke and D. C. Mattfeld. Anticipation and flexibility in dynamic scheduling. *International Journal of Production Research*, 43(15):3103–3129, 2005.

J. Branke, T. Hildebrandt, and B. Scholz-Reiter. Hyper-heuristic Evolution of Dispatching Rules: A Comparison of Rule Representations. *Evolutionary Computation*, 23(2):249–277, 2015.

J. Branke, S. Nguyen, C. W. Pickardt, and M. Zhang. Automated Design of Production Scheduling Heuristics: A Review. *IEEE Transactions on Evolutionary Computation*, 20(1):110–124, 2016.

D. Briskorn, P. Zeise, and J. Packowski. Quasi-fixed cyclic production schemes for multiple products with stochastic demand. *European Journal of Operational Research*, 252(1):156–169, 2016.

E. K. Burke, M. R. Hyde, G. Kendall, and J. Woodward. Automatic heuristic generation with genetic programming. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation - GECCO '07*, page 1559, 2007.

E. K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, and R. Qu. Hyper-Heuristics: A Survey of the State of the Art. *Journal of the Operational Research Society*, 64(12):1695–1724, 2013.

J. A. Caballero. Logic hybrid simulation-optimization algorithm for distillation design. *Computers & Chemical Engineering*, 72:284–299, 2015.

A. S. Caplin. The variability of aggregate demand with (S, s) inventory policies. *Econometrica*, 53(6):1395–1409, 1985.

K. L. Carson. Flexibility - the guiding principle for antibody manufacturing. *Nature Biotechnology*, 23(9):1054–8, 2005.

R. Cheng, M. Gen, and Y. Tsujimura. A tutorial survey of job-shop scheduling problems using genetic algorithms, part II: hybrid genetic search strategies. *Computers & Industrial Engineering*, 36(2):343–364, 1999.

L. Chu and D. K. Robinson. Industrial choices for protein production by large-scale cell culture. *Current Opinion in Biotechnology*, 12(2):180–187, 2001.

Y. Chu, F. You, J. M. Wassick, and A. Agarwal. Simulation-based optimization framework for multi-echelon inventory systems under uncertainty. *Computers and Chemical Engineering*, 73:1–16, 2015.

W. Dangelmaier and E. Kaganova. Robust Solution Approach to CLSP Problem with an Uncertain Demand. In *Robust Manufacturing Control*, pages 455–467. Springer, 2013.

K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature PPSN VI*, pages 849–858, 2000.

J. A. DiMasi and H. G. Grabowski. The cost of biopharmaceutical R&D: is biotech different? *Managerial and Decision Economics*, 28(4-5):469–479, 2007.

J. A. DiMasi, L. Feldman, A. Seckler, and A. Wilson. Trends in risks associated with new drug development: success rates for investigational drugs. *Clinical Pharmacology and Therapeutics*, 87(3):272–7, 2010.

C. L. Doll and D. C. Whybark. An Iterative Procedure for the Single-Machine Multi-Product Lot Scheduling Problem. *Management Science*, 20(1):50–55, 1973.

M. Dorigo and T. Stützle. Ant Colony Optimization: Overview and Recent Advances. In M. Gendreau and J.-Y. Potvin, editors, *Handbook of Metaheuristics*, pages 227–263. Springer US, 2010.

A. Drexl and K. Haase. Proportional lotsizing and scheduling. *International Journal of Production Economics*, 40(1):73–87, 1995.

A. Drexl and A. Kimms. Lot sizing and scheduling — Survey and extensions. *European Journal of Operational Research*, 99(2):221–235, 1997.

R. C. Eberhart, Y. Shi, and J. Kennedy. *Swarm Intelligence (Morgan Kaufmann Series in Evolutionary Computation)*. Morgan Kaufmann Publishers, 2001.

H. G. Eichler, B. Bloechl-Daum, D. Brasseur, A. Breckenridge, H. Leufkens, J. Raine, T. Salmonson, C. K. Schneider, and G. Rasi. The risks of risk aversion in drug regulation. *Nature Reviews Drug Discovery*, 12:907–16, 2013.

G. D. Eppen and R. K. Martin. Solving Multi-Item Capacitated Lot-Sizing Problems Using Variable Redefinition. *Operations Research*, 35(6):832–848, 1987.

D. Erlenkotter. Ford whitman Harris and the Economic Orden Quantity Model. *Operations Research Society of America*, 38(6):937–946, 1990.

R. L. Fahrner, H. L. Knudsen, C. D. Basey, W. Galan, D. Feuerhelm, M. Vanderlaan, and G. S. Blank. Industrial Purification of Pharmaceutical Antibodies: Development, Operation, and Validation of Chromatography Processes. *Biotechnology and Genetic Engineering Reviews*, 18(1):301–327, 2001.

S. S. Farid. Established bioprocesses for producing antibodies as a basis for future planning. *Advances in Biochemical Engineering/Biotechnology*, 101:1–42, 2006.

S. S. Farid. Process economics of industrial monoclonal antibody manufacture. *Journal of Chromatography B*, 848(1):8–18, 2007.

S. S. Farid, J. Pollock, and S. V. Ho. Evaluating the Economic and Operational Feasibility of Continuous Processes for Monoclonal Antibodies. In G. Subramanian, editor, *Continuous Processing in Pharmaceutical Manufacturing*, pages 433–456. Wiley-VCH Verlag GmbH & Co. KGaA, 2014.

FDA. FDA Q5A Guidance Document: Viral Safety Evaluation of Biotechnology Products Derived from Cell Lines of Human or Animal Origin, 1998. Available at http://www.fda.gov/cder/guidance/Q5A-fnl.PDF.

M. Felo, B. Christensen, and J. Higgins. Process cost and facility considerations in the selection of primary cell culture clarification technology. *Biotechnology Progress*, 29(5):1239–1245, 2013.

B. Fleischmann and H. Meyr. The general lotsizing and scheduling problem. *OR Spektrum*, 19(1):11–21, 1997.

J. M. Framinan, R. Leisten, and R. Ruiz García. *Manufacturing Scheduling Systems*. Springer London, 2014. ISBN 978-1-4471-6271-1.

G. Gallego. A minmax distribution free procedure for the (Q, R) inventory model. *Operations Research Letters*, 11(1):55–60, 1992.

G. Gallego and D. X. Shaw. Complexity of the ELSP with general cyclic schedules. *IIE Transactions (Institute of Industrial Engineers)*, 29(2):109–113, 1997.

A. Gascon, R. C. Leachman, and P. Lefrançois. Multi-item, single-machine scheduling problem with stochastic demands: a comparison of heuristics. *International Journal of Production Research*, 32(3):583–596, 1994.

G. Gatica, L. Papageorgiou, and N. Shah. Capacity Planning Under Uncertainty for the Pharmaceutical Industry. *Chemical Engineering Research and Design*, 81(6):665–678, 2003.

GEN. Multiproduct Manufacturing Facilities. *Genetic Engineering & Biotechnology News*, 26(6), 2006.

E. D. George and S. S. Farid. Strategic Biopharmaceutical Portfolio Development: An Analysis of Constraint–Induced Implications. *Biotechnology Progress*, 24 (3):698–713, 2008.

B. Giffler and G. L. Thompson. Algorithms for Solving Production-Scheduling Problems. *Operations Research*, 8(4):487–503, 1960.

F. Glover and M. Laguna. Tabu Search. In D.-Z. Du and P. M. Pardalos, editors, *Handbook of Combinatorial Optimization: Volume1–3*, pages 2093–2229. Springer US, 1999.

D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989. ISBN 0201157675.

D. E. Goldberg and J. H. Holland. Genetic Algorithms and Machine Learning. *Machine Learning*, 3(2):95–99, 1988.

H. Grabowski, G. Long, and R. Mortimer. Recent trends in brand-name and generic drug competition. *Journal of Medical Economics*, pages 1–8, 2013.

S. C. Graves. The Multi-Product Production Cycling Problem. *AIIE Transactions*, 12(3):233–240, 1980.

S. C. Graves. A Review of Production Scheduling. *Operations Research*, 29(4): 646–675, 1981.

H. Guner Goren, S. Tunali, and R. Jans. A review of applications of genetic algorithms in lot sizing. *Journal of Intelligent Manufacturing*, 21(4):575–590, 2008.

N. Hansen and A. Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, 1st edition, 1994. ISBN 0023527617.

J. C. Ho, Y.-L. Chang, and A. O. Solis. Two modifications of the least cost per period heuristic for dynamic lot-sizing. *Journal of the Operational Research Society*, 57(8):1005–1013, 2006.

J. H. Holland. *Adaption in Natural and Artificial Systems*. The University of Michigan Press, 1975.

H. Jalali and I. Van Nieuwenhuyse. Simulation optimization in inventory replenishment: A classification. *IIE Transactions*, 47(11):1217–1235, 2015.

R. J. W. James and B. Almada-Lobo. Single and parallel machine capacitated lot-sizing and scheduling: New iterative MIP-based neighborhood search heuristics. *Computers & Operations Research*, 38(12):1816–1825, 2011.

K. Jankauskas, L. G. Papageorgiou, and S. S. Farid. Continuous-Time Heuristic Model for Medium-Term Capacity Planning of a Multi-Suite, Multi-Product Biopharmaceutical Facility. In A. Espuña, M. Graells, and L. Puigjaner, editors, *27th European Symposium on Computer Aided Process Engineering*, volume 40, pages 1303–1308, Elsevier Masson SAS, 2017.

K. Jankauskas, L. G. Papageorgiou, and S. S. Farid. Fast genetic algorithm approaches to solving discrete-time mixed integer linear programming problems of capacity planning and scheduling of biopharmaceutical manufacture. *Computers & Chemical Engineering*, 121:212–223, 2019.

R. Jans and Z. Degraeve. Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches. *European Journal of Operational Research*, 177(3):1855–1875, 2007.

R. Jans and Z. Degraeve. Modeling industrial lot sizing problems: a review. *International Journal of Production Research*, 46(6):1619–1643, 2008.

Y. Jin and J. Branke. Evolutionary Optimization in Uncertain Environments—A Survey. *IEEE Transactions on Evolutionary Computation*, 9(3):303–317, 2005.

C. Jordan. *Batching and scheduling: models and methods for several problem classes*, volume 437. Springer Science & Business Media, 1996.

C. Jordan and A. Drexl. Discrete Lotsizing and Scheduling by Batch Sequencing. *Management Science*, 44(5):698–713, 1998.

B. Karimi, S. M. T. Fatemi Ghomi, and J. M. Wilson. The capacitated lot sizing problem: A review of models and algorithms. *Omega*, 31(5):365–378, 2003.

B. Kelley. Very large scale monoclonal antibody purification: The case for conventional unit operations. *Biotechnology Progress*, 23(5):995–1008, 2007.

B. Kelley. Industrialization of mAb production technology: the bioprocessing industry at a crossroads. *mAbs*, 1(5):443–452, 2009.

A. Kimms. A genetic algorithm for multi-level, multi-machine lot sizing and scheduling. *Computers & Operations Research*, 26(8):829–848, 1999.

J. Knowles. A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers. *Proceedings - 5th International Conference on Intelligent Systems Design and Applications 2005, ISDA '05*, 2005:552–557, 2005.

J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992. ISBN 0-262-11170-5.

La Merie Business Intelligence. Blockbuster Biologics 2012. Technical report, La Merie Business Intelligence, 2013. Available at www.pipelinereview.com.

P. Laborie. IBM ILOG CP Optimizer for detailed scheduling illustrated on three problems. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5547:148–162, 2009.

K. Lakhdar, Y. Zhou, J. Savery, N. J. Titchener-Hooker, and L. G. Papageorgiou. Medium term planning of biopharmaceutical manufacture using mathematical programming. *Biotechnology Progress*, 21(5):1478–1489, 2005.

K. Lakhdar, S. S. Farid, J. Savery, N. J. Titchener-Hooker, and L. G. Papageorgiou. Medium term planning of biopharmaceutical manufacture under uncertainty. *Computer Aided Chemical Engineering*, 21:2069–2074, 2006.

K. Lakhdar, J. Savery, L. G. Papageorgiou, and S. S. Farid. Multiobjective long-term planning of biopharmaceutical manufacturing facilities. *Biotechnology Progress*, 23(6):1383–1393, 2007.

L. S. Lasdon and R. C. Terjung. An Efficient Algorithm for Multi-Item Scheduling. *Operations Research*, 19(4):946–969, 1971.

R. C. Leachman and A. Gascon. A Heuristic Scheduling Policy for Multi-Item, Single-Machine Production Systems with Time-Varying, Stochastic Demands. *Management Science*, 34(3):377–390, 1988.

A. A. Levis and L. G. Papageorgiou. A hierarchical solution approach for multi-site capacity planning under uncertainty in the pharmaceutical industry. *Computers & Chemical Engineering*, 28(5):707–725, 2004.

Z. Li and M. Ierapetritou. Process scheduling under uncertainty: Review and challenges. *Computers & Chemical Engineering*, 32(4-5):715–727, 2008.

G. L. Liao and S. H. Sheu. Economic production quantity model for randomly failing production process with minimal repair and imperfect maintenance. *International Journal of Production Economics*, 130(1):118–124, 2011.

A. C. Lim, J. Washbrook, N. J. Titchener-Hooker, and S. S. Farid. A computer-aided approach to compare the production economics of fed-batch and perfusion culture under uncertainty. *Biotechnology and Bioengineering*, 93(4):687–697, 2006.

N. Löhndorf and S. Minner. Simulation optimization for the stochastic economic lot scheduling problem. *IIE Transactions*, 45(7):796–810, 2013.

N. Löhndorf, M. Riel, and S. Minner. Simulation optimization for the stochastic economic lot scheduling problem with sequence-dependent setup times. *International Journal of Production Economics*, 157(1):170–176, 2014.

S. Luke. ECJ Evolutionary Computation Library, 1998. Available for free at http://cs.gmu.edu/~eclab/projects/ecj/.

S. Luke. *Essentials of Metaheuristics*. Lulu, second edition, 2013. ISBN 9781300549628.

E. Mahajan, A. George, and B. Wolk. Improving affinity chromatography resin efficiency using semi-continuous chromatography. *Journal of Chromatography A*, 1227:154–162, 2012.

P. A. Marichal-Gallardo and M. M. Alvarez. State-of-the-art in downstream processing of monoclonal antibodies: process trends in design and validation. *Biotechnology Progress*, 28(4):899–916, 2012.

C. M. Marques, S. Moniz, J. P. de Sousa, and A. P. Barbosa-Póvoa. A simulation-optimization approach to integrate process design and planning decisions under technical and market uncertainties: A case from the chemical-pharmaceutical industry. *Computers & Chemical Engineering*, 106:796–813, 2017.

T. L. McGurk. Ramping Up and Ensuring Supply Capability for Biopharmaceuticals. *Biopharm International*, 17(1), 2004.

E. Mehdizadeh, V. Hajipour, and M. R. Mohammadizadeh. A bi-objective multi-item capacitated lot-sizing model: Two Pareto-based meta-heuristic algorithms. *International Journal of Management Science and Engineering Management*, 11(4):279–293, 2016.

Z. Michalewicz, R. Hinterding, and M. Michalewicz. Evolutionary Algorithms. In *Fuzzy Evolutionary Computation*, pages 3–31. Springer US, 1997.

M. Närhi and K. Nordström. Manufacturing, regulatory and commercial challenges of biopharmaceuticals production: a Finnish perspective. *European Journal of Pharmaceutics and Biopharmaceutics*, 59(3):397–405, 2005.

M. Nourelfath. Service level robustness in stochastic production planning under random machine breakdowns. *European Journal of Operational Research*, 212 (1):81–88, 2011.

D. Ouelhadj and S. Petrovic. A survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling*, 12(4):417–431, 2009.

L. Özdamar and Ş. İ. Birbil. Hybrid heuristics for the capacitated lot sizing and loading problem with setup times and overtime decisions. *European Journal of Operational Research*, 110(3):525–547, 1998.

S. S. Ozturk. Oportunities and Challenges for the Implementation of Continuous Processing in Biomanufacturing. In *Continuous Processing in Pharmaceutical Manufacturing*, pages 457–476. Wiley Blackwell, 2015.

S. M. Paul, D. S. Mytelka, C. T. Dunwiddie, C. C. Persinger, B. H. Munos, S. R. Lindborg, and A. L. Schacht. How to improve R&D productivity: the pharmaceutical industry's grand challenge. *Nature Reviews Drug Discovery*, 9(3): 203–214, 2010.

C. W. Pickardt, T. Hildebrandt, J. Branke, J. Heger, and B. Scholz-Reiter. Evolutionary generation of dispatching rule sets for complex dynamic scheduling problems. *International Journal of Production Economics*, 145(1):67–77, 2013.

G. S. Piperagkas, I. Konstantaras, K. Skouri, and K. E. Parsopoulos. Solving the stochastic dynamic lot-sizing problem through nature-inspired heuristics. *Computers & Operations Research*, 39(7):1555–1565, 2012.

R. Pitakaso, C. Almeder, K. F. Doerner, and R. F. Hartl. A MAX-MIN ant system for unconstrained multi-level lot-sizing problems. *Computers & Operations Research*, 34(9):2533–2552, 2007.

J. Pollock. *Evaluating the Economic and Operational Feasibility of Continuous Processes for Monoclonal Antibodies*. PhD thesis, University College London, 2013.

J. Pollock, G. Bolton, J. Coffman, S. V. Ho, D. G. Bracewell, and S. S. Farid. Optimising the design and operation of semi-continuous affinity chromatography for clinical and commercial manufacture. *Journal of Chromatography. A*, 1284: 17–27, 2013a.

J. Pollock, S. V. Ho, and S. S. Farid. Fed-batch and perfusion culture processes: Economic, environmental, and operational feasibility under uncertainty. *Biotechnology and Bioengineering*, 110(1):206–219, 2013b.

J. Pollock, J. Coffman, S. V. Ho, and S. S. Farid. Integrated continuous bioprocessing: Economic, operational and environmental feasibility for clinical and commercial antibody manufacture. *Biotechnology Progress*, 33(4):854–866, 2017.

C. N. Potts and L. N. V. Wassenhove. Integrating Scheduling with Batching and Lot-Sizing: A Review of Algorithms and Complexity. *Journal of the Operational Research Society*, 43(5):395–406, 1992.

B. S. Purohit and B. Kumar Lad. Production and maintenance planning: an integrated approach under uncertainties. *International Journal of Advanced Manufacturing Technology*, 86(9-12):3179–3191, 2016.

R. A. Rader. What is a Biopharmaceutical? Part 1: (Bio)Technology-Based Definitions. *BioExecutive International*, pages 60–65, 2005.

R. Ramya, C. Rajendran, and H. Ziegler. Capacitated lot-sizing problem with production carry-over and set-up splitting: mathematical models. *International Journal of Production Research*, 54(8):2332–2344, 2016.

T. C. Ransohoff. Considerations Impacting the Make vs. Buy Decision. *American Pharmaceutical Outsourcing*, 5(2), 2004.

C. Renotte and A. V. Wouwer. Stochastic Approximation Techniques Applied to Parameter Estimation in a Biological Model. In *Second IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2003. Proceedings*, pages 261–265, 2003.

S. Rouf, M. Moo-Young, J. Scharer, and P. Douglas. Single versus multiple bioreactor scale-up: economy for high-value products. *Biochemical Engineering Journal*, 6(1):25–31, 2000.

A. A. Shukla and U. Gottschalk. Single-use disposable technologies for biopharmaceutical manufacturing. *Trends in Biotechnology*, 31(3):147–154, 2013.

A. A. Shukla, B. Hubbard, T. Tressel, S. Guhan, and D. Low. Downstream processing of monoclonal antibodies—Application of platform approaches. *Journal of Chromatography B*, 848(1):28–39, 2007.

C. C. Siganporia, S. Ghosh, T. Daszkowski, L. G. Papageorgiou, and S. S. Farid. Capacity planning for batch and perfusion bioprocesses across multiple biopharmaceutical facilities. *Biotechnology Progress*, 30(3):594–606, 2014.

E. A. Silver and H. C. Meal. A heuristic for selecting lot size quantities for the case of a deterministic time-varying demand rate and discrete opportunities for replenishment. *Production and Inventory Management*, 14(2):64–74, 1973.

G. Sofer and D. Lister. Virus inactivation in the 1990s - and into the 21st century. *Biopharm International*, (April):42–52, 2003.

G. K. Sofer and L. E. Nyström. *Process chromatography: a guide to validation*. Academic Press, 1991. ISBN 0126542678.

C. R. Sox, P. L. Jackson, A. Bowman, and J. A. Muckstadt. A review of the stochastic lot scheduling problem. *International Journal of Production Economics*, 62 (3):181–200, 1999.

T. G. Stützle. *Local Search Algorithms for Combinatorial Problems - Analysis, Algorithms and New Applications*. PhD thesis, Darmstadt University of Technology, 1998.

H. Tempelmeier. A column generation heuristic for dynamic capacitated lot sizing with random demand under a fill rate constraint. *Omega*, 39(6):627–633, 2011.

H. Tempelmeier. Stochastic Lot Sizing Problems. In J. MacGregor Smith and B. Tan, editors, *Handbook of Stochastic Models and Analysis of Manufacturing System Operations*, volume 192, chapter 10, pages 313–344. Springer, New York, NY, 2013.

H. Tempelmeier and T. Hilger. Linear programming models for a stochastic dynamic capacitated lot sizing problem. *Computers & Operations Research*, 59: 119–125, 2015.

P. Troein. Market Development in Europe and Globally—A Perspective on Biologicals and Biosimilars, 2017.

R. Verma, E. Boleti, and A. J. T. George. Antibody engineering: Comparison of bacterial, yeast, insect and mammalian expression systems. *Journal of Immunological Methods*, 216:165–181, 1998.

M. Vieira, T. Pinto-Varela, and A. P. F. D. Barbósa-Póvoa. Planning and Scheduling in the Biopharmaceutical Industry. In T. Majozi, E. R. Seid, and J.-Y. Lee, editors, *Synthesis, Design, and Resource Optimization in Batch Chemical Plants*, volume 32, pages 107 –126. CRC Press, 2015.

M. Vieira, T. Pinto-Varela, S. Moniz, A. P. Barbosa-Póvoa, and L. G. Papageorgiou. Optimal planning and campaign scheduling of biopharmaceutical processes using a continuous-time formulation. *Computers & Chemical Engineering*, 91: 422–444, 2016.

P. A. Vikhar. Evolutionary algorithms: A critical review and its future prospects. In *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*, pages 261–265, IEEE, 2016.

H. M. Wagner and T. M. Whitin. Dynamic Version of the Economic Lot Size Model. *Management Science*, 5(1):89–96, 1958.

M. Wagner and S. R. Smits. A local search algorithm for the optimization of the stochastic economic lot scheduling problem. *International Journal of Production Economics*, 90(3):391–402, 2004.

J. P. Walser, R. Iyer, and N. Venkatasubramanyan. An integer local search method with application to capacitated production planning. In *Proceedings of AAAI-98*, pages 373–379, 1998.

G. Walsh. Biopharmaceutical benchmarks 2010. *Nature Biotechnology*, 28(9): 917–24, 2010.

V. Warikoo, R. Godawat, K. Brower, S. Jain, D. Cummings, E. Simons, T. Johnson, J. Walther, M. Yu, B. Wright, J. Mclarty, K. P. Karey, C. Hwang, W. Zhou, F. Riske, and K. Konstantinov. Integrated continuous production of recombinant therapeutic proteins. *Biotechnology and Bioengineering*, 109(12):3018–3029, 2012.

M. Weise, M.-C. Bielsky, K. De Smet, F. Ehmann, N. Ekman, G. Narayanan, H.-K. Heim, E. Heinonen, K. Ho, R. Thorpe, C. Vleminckx, M. Wadhwa, and C. K. Schneider. Biosimilars—why terminology matters. *Nature Biotechnology*, 29 (8):690–3, 2011.

R. G. Werner. Economic aspects of commercial manufacture of biopharmaceuticals. *Journal of Biotechnology*, 113(1-3):171–82, 2004.

E. M. M. Winands, I. Adan, and G. J. van Houtum. The stochastic economic lot scheduling problem: A survey. *European Journal of Operational Research*, 210 (1):1–9, 2011.

Appendix A

Chapter 4 Appendix

A.1 Figures

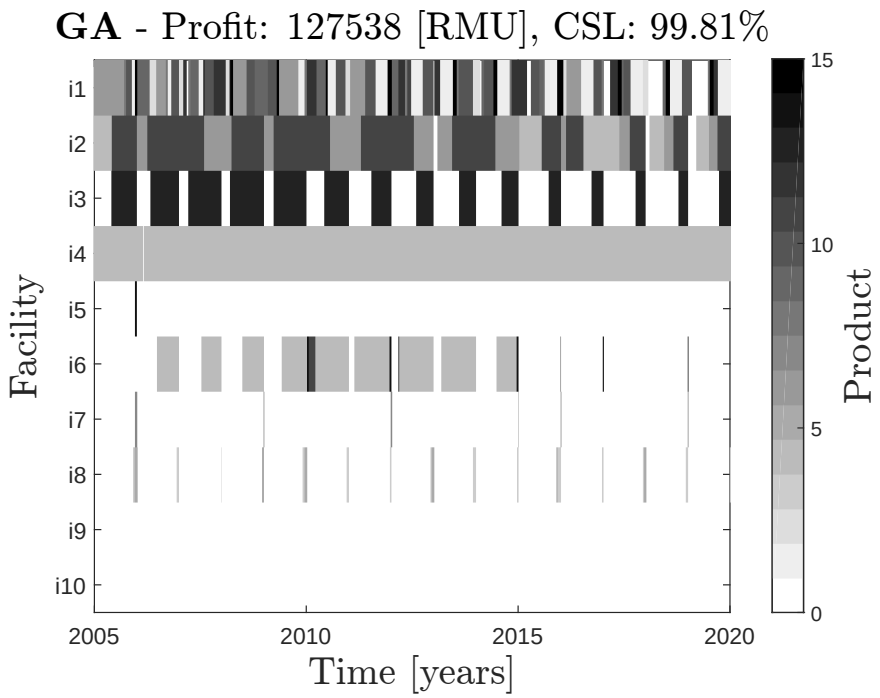


Figure A.1: Exemplary Gantt chart of a schedule generated by the GA for the 2 × demand case. The profit and customer service level (CSL) is also indicated..

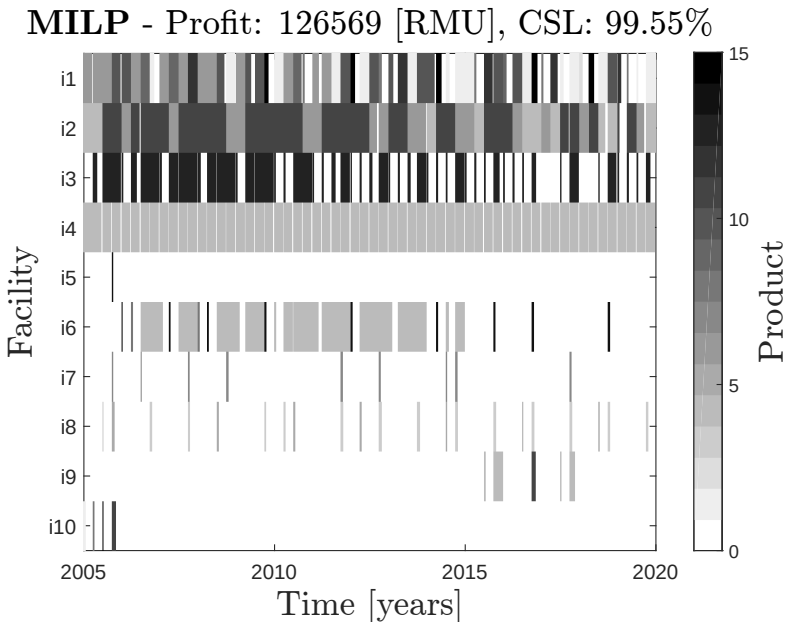


Figure A.2: Exemplary Gantt chart of a schedule generated by the MILP, for the 2 $\times$ demand case. The profit and customer service level (CSL) is also indicated.

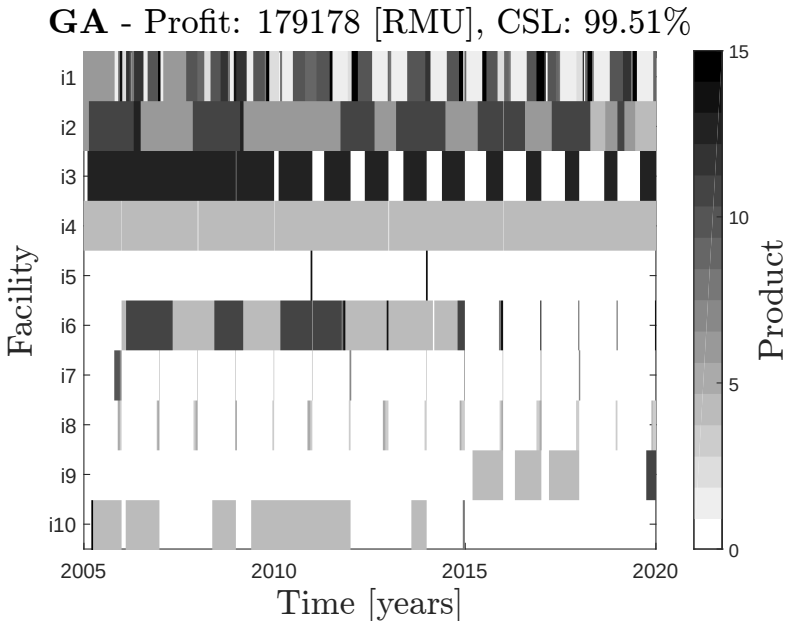


Figure A.3: Exemplary Gantt chart of a schedule generated by the GA, for the 3 $\times$ demand case. The profit and customer service level (CSL) is also indicated.

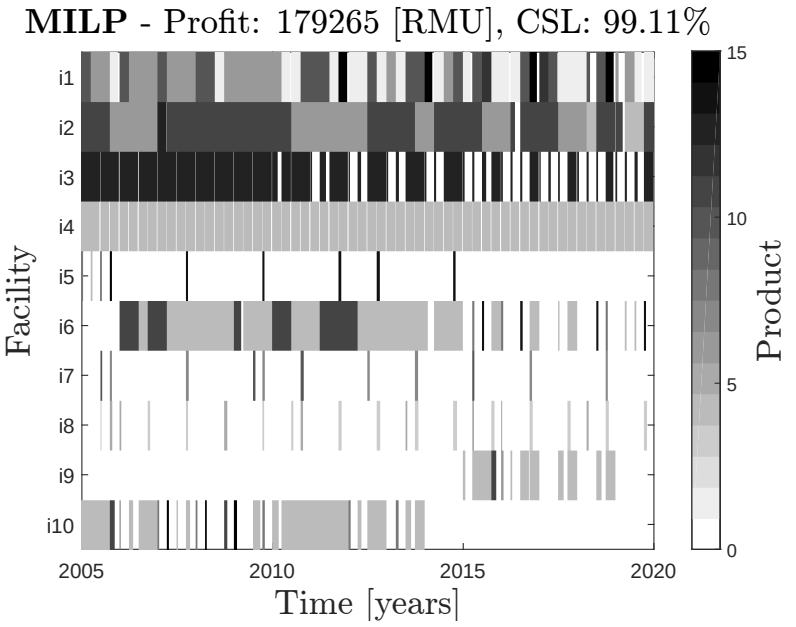


Figure A.4: Exemplary Gantt chart of a schedule generated by the MILP, for the 3 $\times$ demand case. The profit and customer service level (CSL) is also indicated.

Appendix B

Chapter 5 Appendix

B.1 Bioprocess Model Parameters

Calculation of Bioprocess Economics Parameters

The basis and rationale for determining the cost parameters for different aspects of the process is laid out in this section. Figures, estimates and assumptions are based on guideline values used by or suggested by Felo et al. (2013), Farid et al. (2014), Pollock et al. (2013b), or Pollock (2013). The dollar amounts have been transformed such that 1,000 USD $\approx$ 1 RMU.

There are six different types of costs attributed to the bioprocess as defined in Section 5.2.1. These costs are attributable to direct costs including materials and consumables and do not take into account labour, indirect or other fixed investment costs unless otherwise stated or indicated. It is assumed that the process utilises disposable/single-use equipment and is capable of meeting an annual demand of ca. 450 kg with a 1:1 USP:DSP train ratio and whole process yield of 69%.

Seed Train Costs This is representative of the direct material costs accrued from operating the seed train (Seed #1 and Seed #2 in Figure 5.3). To estimate this, the costs for a 100L Wave bag, a 200L SUB (Single Use Bioreactor), and media costs are summed up — it is assumed that there are no additions during the batch and

the volumetric efficiency is 80% for both reactors. Media costs $ 3.15/L.

$$\begin{aligned} & \text{100L Wave} + \text{200L SUB} + 80\% \text{ of 300L} \times \text{Media Cost} \\ = \quad & 420 + 4200 + 240 \times 3.15 \\ = \quad & \$\ 5376 \end{aligned}$$

So, about 5.4 RMU.

Cell Culture Setup Costs This takes into account the ATF filter, the SUB and the initial media within it. The bioreactor used is a 2000L SUB with a volume efficiency of 50% utilising an ATF 10 system.

$$\begin{aligned} & \text{1000L} \times \text{Media Cost} + \text{ATF filter} + \text{2000L SUB} \\ = \quad & 1000 \times 3.15 + 16300 + 9800 \\ = \quad & \$\ 29250 \end{aligned}$$

This comes to roughly 29.3 RMU.

Cell Culture Daily Costs Media costs for daily perfusion are the only component represented here; the perfusion rate is 1 vessel volumes per day (vv/day) in terms of the liquid working volume.

$$\begin{aligned} & \text{Perfusion rate} \times \text{working volume} \times \text{Media Cost} \\ = \quad & 1 \times 1000 \times 3.15 \\ = \quad & \$\ 3150 \end{aligned}$$

Cell culture daily costs are therefore estimated to be 3.2 RMU.

ATF Filter Replacement Cost Simply the cost for a new filter for the ATF 10 system. This is stated to be $ 16300 per unit and therefore a cost of 16.3 RMU is used.

DSP Batch Costs Includes the process steps from the Protein A step to the final UF/DF. The costs of the chromatography resins are amortised over the number of batches it takes to meet five years' worth of demand which take into account repurchases when the resin has exceeded its lifetime (re)uses. The parameters of which can be found in Table B.1.

$$\begin{aligned} &\text{Amort. ProA} + 2 \times \text{Amort. IEX} + \text{VRF} + \text{Amort. UF} + \text{Bags \& Liners} \\ &= \quad 1742 + 2 \times 302 + 6500 + 100 + 2 \times 453 + 7 \times 360 \\ &= \quad \$\ 12372 \end{aligned}$$

As such, the Cost of each DSP batch is estimated to be 12.4 RMU.

Campaign/Changeover Costs Estimated from QCQA batch release costs being $ 35000 (Pollock, 2013) to therefore be 35 RMU.

B.2 Tables

Table B.1: Parameters, inputs and outputs for estimating costs of the Protein A and the two ion-exchange (IEX) resins.

Parameters	ProA	IEX	Unit
Inputs			
Columns	1	1	
Resin DBC	50	30	g/L
Bed height	20	20	cm
No.of cycles	1	1	
Linear velocity	450	600	cm/h
Lifetime reuses	120	180	
Resin cost	8000	2000	$/L
Outputs			
Column diameter	40	40	cm
Cycle time	173	130	minutes
Amortised resin cost	1742	302	$/batch

Table B.2: Summary of GA evolved (s,B) policy parameters for the various USP:DSP process configurations. Units for parameter s and annual demand are in kg and in days for parameter B.

	1 reactor		2 reactors		3 reactors		4 reactors	
	s	B	s	B	s	B	s	B
mean	34.8	110.9	38.3	105.0	41.7	109.4	47.4	110.0
s.d.	1.86	4.55	1.93	3.16	2.89	3.04	5.2	2.56
median		111		105		109		110
mode		111		107		106		110

B.3 Figures

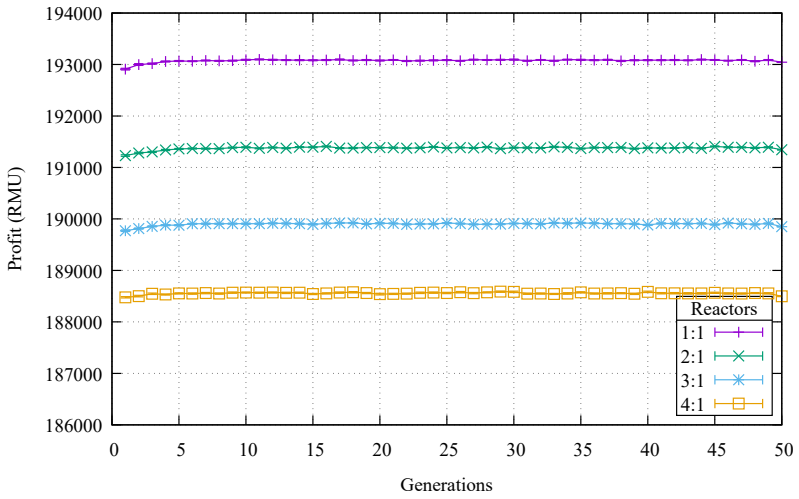


Figure B.1: Mean convergence over 50 GA runs for the (s,B) scheduling strategies at the four process configurations investigated.

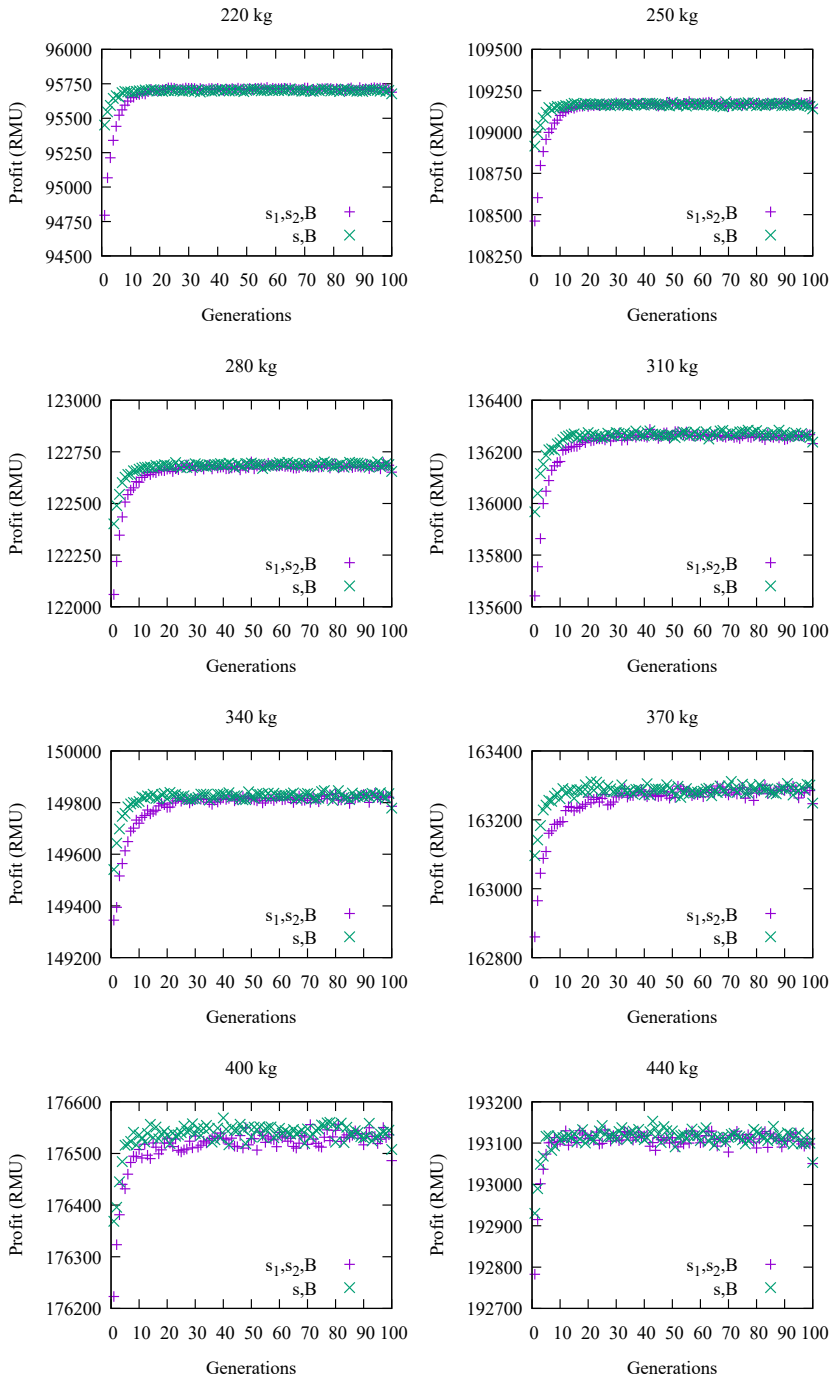


Figure B.2: Mean convergence over 50 GA runs for the (s,B) and (s_1,s_2,B) scheduling strategies at various demand targets.

Appendix C

Chapter 6 Appendix

C.1 Tables

Table C.1: Sensitivity of the optimised policies to different demand scales at monthly demand frequency. Figures reported are expected profit values (in RMU) ± std. err. Best performance for each demand case is in **bold**.

Demand case	BSP1B	BSP2B	BSP3B	ANN
50%	90985 ± 4.5	**91833 ± 4.7**	91489 ± 4.6	67738 ± 5.7
75%	140314 ± 5.9	**140592 ± 5.9**	140552 ± 6.0	128846 ± 8.1
80%	150060 ± 6.3	**150409 ± 6.4**	150217 ± 6.3	141164 ± 8.4
85%	159970 ± 6.8	**160058 ± 6.8**	160016 ± 6.7	153263 ± 8.6
90%	169796 ± 7.0	**169911 ± 7.2**	169869 ± 7.1	165227 ± 9.1
95%	179480 ± 7.6	**179574 ± 7.5**	**179576 ± 7.5**	177260 ± 9.3
100%	189225 ± 8.1	189261 ± 8.2	**189343 ± 8.0**	188540 ± 11.7
105%	197807 ± 10.8	**197899 ± 10.8**	197589 ± 11.1	189258 ± 32.6
110%	**202071 ± 17.9**	**202073 ± 17.8**	201003 ± 18.2	182174 ± 33.9
115%	**203112 ± 20.8**	203071 ± 20.6	202013 ± 20.4	174785 ± 34.0

Table C.2: Profit, costs, customer service level (CSL), and other performance characteristics for the ANN and BSP3B, tuned at monthly demand frequency and ±10% demand scales. Mean ± std. err. are listed for 50 runs each and values reported are in RMU apart from CSL values.

	−10% demand		Base demand		+10% demand	
	ANN	BSP3B	ANN	BSP3B	ANN	BSP3B
Profit	168465 ± 99.5	170223 ± 9.6	188373 ± 45.1	189288 ± 7.3	206240 ± 94.5	207621 ± 10.3
Revenue	200032 ± 85.1	200475 ± 7.1	222103 ± 30.3	222547 ± 9.2	242760 ± 82.8	243717 ± 12.8
Total costs	31566 ± 38.9	30252 ± 11.3	33730 ± 25.0	33259 ± 9.4	36520 ± 24.5	36096 ± 10.6
Seed	248.2 ± 2.08	178.1 ± 0.48	201.4 ± 0.82	182.9 ± 0.67	155.3 ± 0.47	154.9 ± 0.28
USP	8360 ± 30.0	7479 ± 6.8	8368 ± 8.9	8130 ± 6.5	8394 ± 4.7	8379 ± 3.0
Replacement ATF filters	7.8 ± 0.17	9.6 ± 0.05	11.2 ± 0.09	11.2 ± 0.06	17.8 ± 0.16	16.1 ± 0.06
Cell culture setup	1436.7 ± 15.86	1030.5 ± 3.53	1163.9 ± 4.16	1063.4 ± 3.20	903.3 ± 2.66	905.6 ± 1.90
DSP	19092 ± 19.6	19075 ± 4.6	21197 ± 10.6	21191 ± 5.3	23298 ± 15.1	23303 ± 6.4
Changeover	1559.9 ± 17.14	1211.8 ± 3.46	1364.1 ± 10.29	1221.3 ± 3.91	1055.6 ± 3.32	1000.5 ± 1.78
Storage	1992 ± 18.3	2134 ± 7.2	2192 ± 14.3	2283 ± 9.9	2686 ± 16.6	2677 ± 8.9
Backlog penalties	282.7 ± 21.63	134.6 ± 1.89	361.9 ± 9.66	203.8 ± 2.47	854.8 ± 22.81	512.1 ± 3.59
Wastage	31.4 ± 0.59	39.5 ± 0.13	46.2 ± 0.31	46.7 ± 0.28	76.2 ± 0.88	69.0 ± 0.23
CSL	99.55% ± 0.044	99.74% ± 0.004	99.49% ± 0.013	99.64% ± 0.004	98.87% ± 0.035	99.17% ± 0.006

Table C.3: Optimised parameters for $p1$-$p3$ in each of the scheduling policies. The mean $\pm$ std. deviation of the 50 best solutions (i.e., the best solution in each EA run) are reported.

	$p1$					
	$Y^{(1)}$	$Y^{(2)}$	$Y^{(3)}$	$Y^{(4)}$	$Y^{(5)}$	B
BSP0	6.2	52.5	-	-	-	60
BSP1A	16.4 ± 1.0	18.7 ± 2.2	-	-	-	60
BSP1B	19.4 ± 1.6	21.3 ± 1.9	-	-	-	43.4 ± 3.1
BSP2A	10.8 ± 2.4	24.3 ± 5.1	15.6 ± 1.1	17.7 ± 2.0	-	60
BSP2B	12.4 ± 3.9	24.8 ± 3.2	18.1 ± 1.8	20.1 ± 1.9	-	44.1 ± 8.6
BSP3A	-	-	-	-	16.0 ± 0.9	60
BSP3B	-	-	-	-	17.4 ± 1.2	43.2 ± 0.9
ANN	–	–	–	–	–	34.1 ± 1.88
	$p2$					
	$Y^{(1)}$	$Y^{(2)}$	$Y^{(3)}$	$Y^{(4)}$	$Y^{(5)}$	B
BSP0	11.1	93.6	-	-	-	60
BSP1A	27.6 ± 1.8	29.5 ± 2.2	-	-	-	60
BSP1B	26.6 ± 2.9	28.6 ± 3.2	-	-	-	58.4 ± 4.5
BSP2A	22.0 ± 4.8	33.6 ± 4.0	26.5 ± 2.1	27.8 ± 5.0	-	60
BSP2B	20.5 ± 6.3	34.0 ± 5.0	26.7 ± 3.3	28.2 ± 2.9	-	58.4 ± 6.4
BSP3A	-	-	-	-	25.2 ± 1.6	60
BSP3B	-	-	-	-	30.1 ± 2.0	51.7 ± 1.9
ANN	–	–	–	–	–	51.3 ± 2.75
	$p3$					
	$Y^{(1)}$	$Y^{(2)}$	$Y^{(3)}$	$Y^{(4)}$	$Y^{(5)}$	B
BSP0	10.7	77.5	-	-	-	60
BSP1A	23.8 ± 0.7	23.9 ± 0.7	-	-	-	60
BSP1B	20.3 ± 2.6	20.9 ± 2.5	-	-	-	72.4 ± 7.2
BSP2A	14.6 ± 4.9	38.3 ± 5.1	21.4 ± 1.3	21.5 ± 1.3	-	60
BSP2B	10.7 ± 4.7	31.0 ± 6.0	19.2 ± 1.6	19.9 ± 1.5	-	75.5 ± 6.6
BSP3A	-	-	-	-	28.1 ± 1.6	60
BSP3B	-	-	-	-	20.6 ± 2.0	77.0 ± 4.7
ANN	–	–	–	–	–	73.2 ± 6.14

Table C.4: Matrices of observed significance level of Mann-Whitney tests comparing the mean profit of the best solution of each policy at different demand scales. The p-values are for the test that $\langle \text{row} \rangle > \langle \text{column} \rangle$.

	Demand case $\times 0.85$		
	BSP1B	BSP2B	BSP3B
BSP1B	-	1.00E+00	1.00E+00
BSP2B	4.71E-33	-	1.01E-10
BSP3B	8.78E-09	1.00E+00	-

	Demand case $\times 0.90$		
	BSP1B	BSP2B	BSP3B
BSP1B	-	1.00E+00	1.00E+00
BSP2B	2.25E-76	-	7.17E-02
BSP3B	1.23E-65	9.28E-01	-

	Demand case $\times 0.95$		
	BSP1B	BSP2B	BSP3B
BSP1B	-	1.00E+00	1.00E+00
BSP2B	9.63E-21	-	3.06E-01
BSP3B	5.87E-19	6.94E-01	-

	Demand case $\times 1.00$		
	BSP1B	BSP2B	BSP3B
BSP1B	-	1.00E+00	1.00E+00
BSP2B	4.63E-17	-	9.95E-01
BSP3B	1.76E-28	5.10E-03	-

	Demand case $\times 1.05$		
	BSP1B	BSP2B	BSP3B
BSP1B	-	1.00E+00	9.90E-01
BSP2B	3.40E-110	-	4.29E-90
BSP3B	9.59E-03	1.00E+00	-

	Demand case $\times 1.10$		
	BSP1B	BSP2B	BSP3B
BSP1B	-	1.00E+00	5.58E-35
BSP2B	1.94E-24	-	1.36E-111
BSP3B	1.00E+00	1.00E+00	-

	Demand case $\times 1.15$		
	BSP1B	BSP2B	BSP3B
BSP1B	-	1.79E-30	6.15E-01
BSP2B	1.00E+00	-	1.00E+00
BSP3B	3.85E-01	9.56E-40	-

C.2 Figures

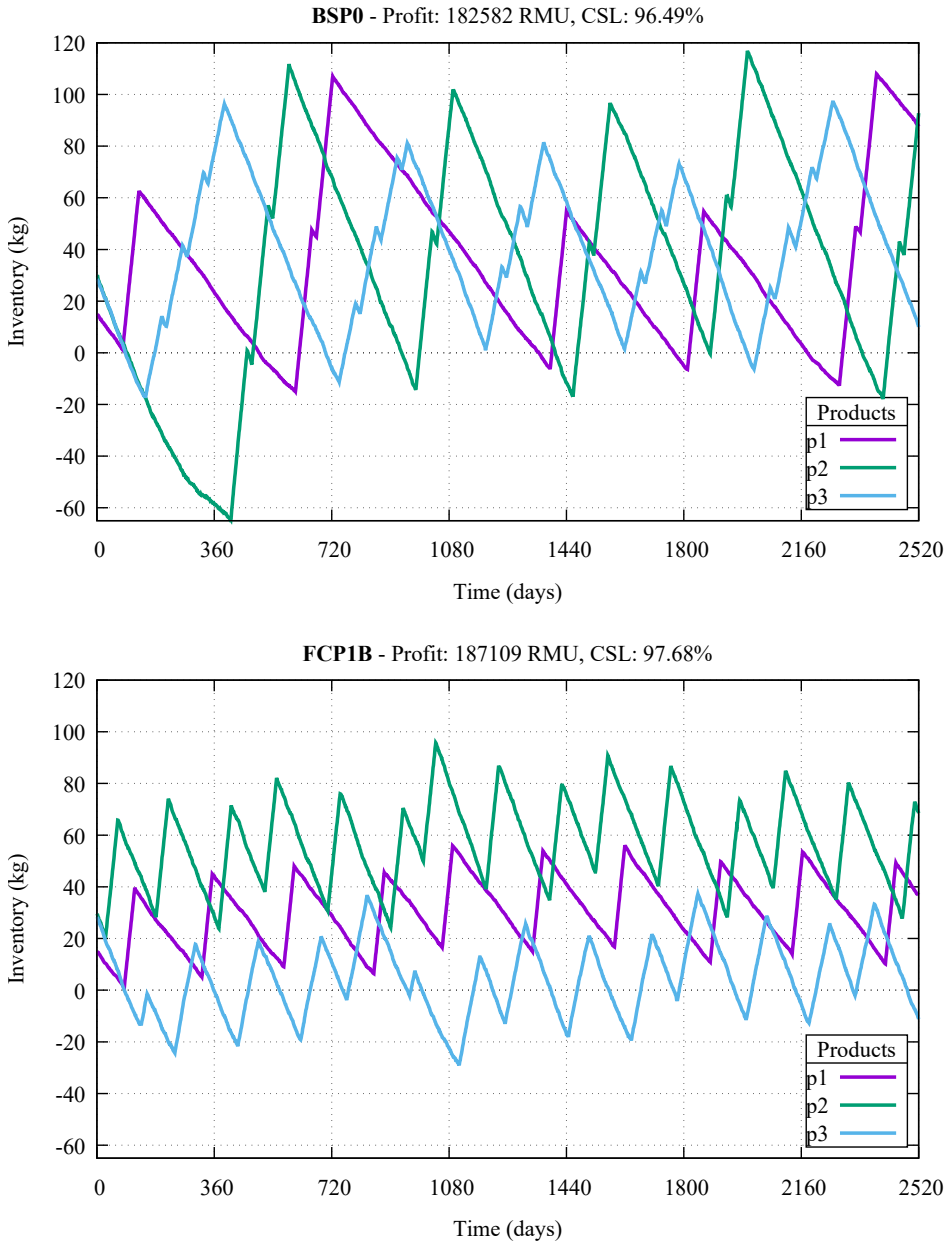


Figure C.1: Exemplar inventory profiles over scheduling horizon from the simulation run in Figure 6.3 on page 128 for the standard and FCP1B scheduling policies.

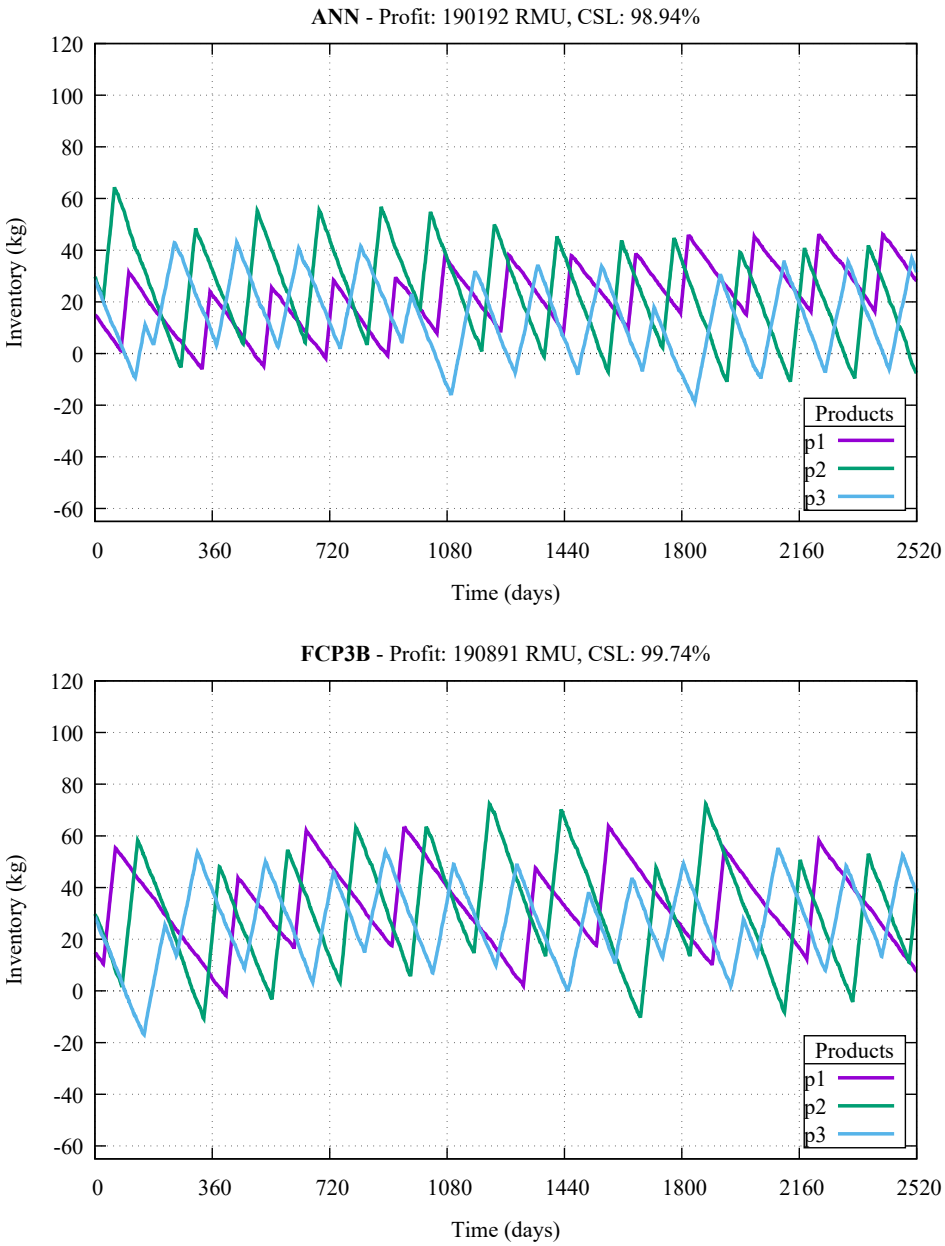


Figure C.2: Exemplar inventory profiles over scheduling horizon from the simulation run in Figure 6.3 on page 128 for the ANN and FCP3B scheduling policies.

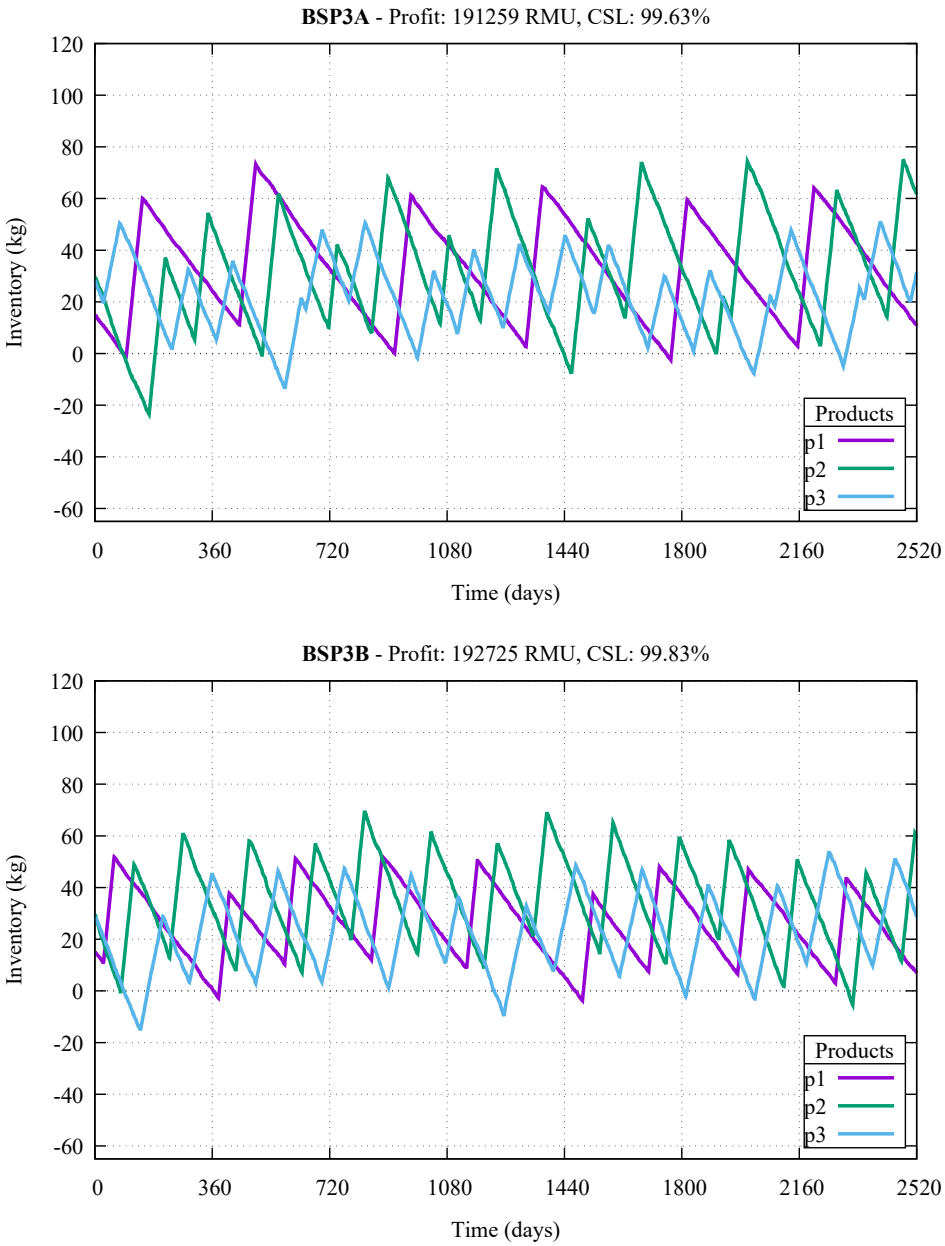


Figure C.3: Exemplar inventory profiles over scheduling horizon from the simulation run in Figure 6.3 on page 128 for the BSP3A and BSP3B scheduling policies.

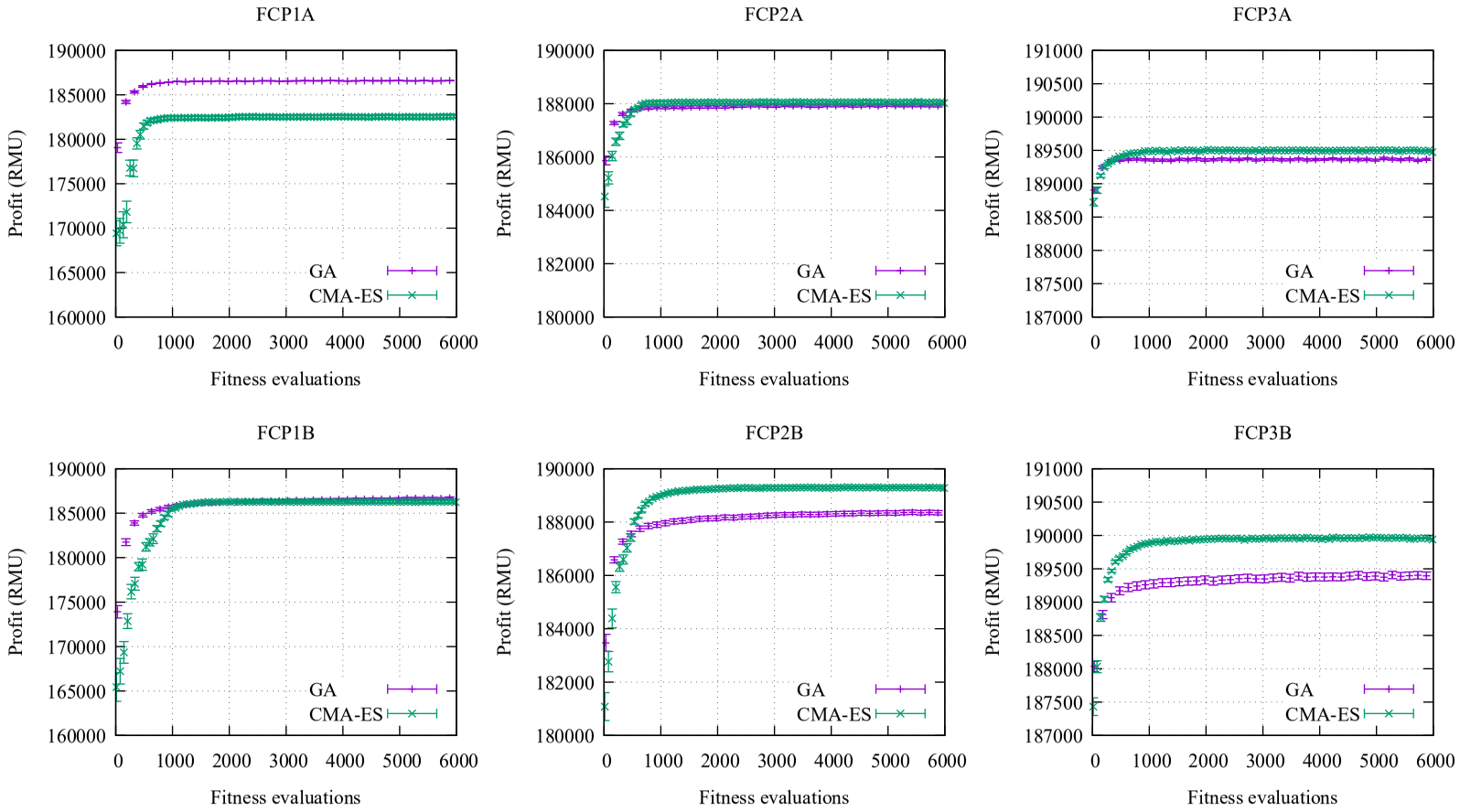


Figure C.4: Convergences of the fixed cycle policy optimisation runs with the genetic algorithm (GA), and CMA-ES.

Appendix D

Publication(s) by the Author

The following are paper(s) published by the author:

Oyebolu, F. B., van Lidth de Jeude, J., Siganporia, C. C., Farid, S. S., Allmendinger, R., and Branke, J. **(2017)**. A new lot sizing and scheduling heuristic for multi-site biopharmaceutical production. *Journal of Heuristics*, 23(4), 231–256. 10.1007/s10732-017-9338-9

Oyebolu, F. B., Allmendinger, R., Farid, S. S., and Branke, J. **(2019)**. Dynamic Scheduling of Multi-Product Continuous Biopharmaceutical Facilities: a Hyper-Heuristic Framework. *Computers & Chemical Engineering*, 125, 71–88. 10.1016/j.compchemeng.2019.03.002