Multi-objective biopharma capacity planning under uncertainty using a flexible genetic algorithm approach

Karolis Jankauskas, Suzanne S. Farid

University College London, Department of Biochemical Engineering, Gower Street, London, WC1E 6BT, UK

Corresponding author:

Prof. Suzanne S. Farid

University College London, Department of Biochemical Engineering, Gower Street, London, WC1E 6BT, UK.

s.farid@ucl.ac.uk.

+44 20 7679 4415

Abstract

This paper presents a flexible genetic algorithm optimisation approach for multi-objective biopharmaceutical planning problems under uncertainty. The optimisation approach combines a continuous-time heuristic model of a biopharmaceutical manufacturing process, a variable-length multi-objective genetic algorithm, and Graphics Processing Unit (GPU)-accelerated Monte Carlo simulation. The proposed approach accounts for constraints and features such as rolling product sequence-dependent changeovers, multiple intermediate demand due dates, product QC/QA release times, and pressure to meet uncertain product demand on time. An industrially-relevant case study is used to illustrate the functionality of the approach. The case study focused on optimisation of conflicting objectives, production throughput and product inventory levels, for a multi-product biopharmaceutical facility over a 3-year period with uncertain product demand. The advantages of the multi-objective GA with the embedded Monte Carlo simulation were demonstrated by comparison with a deterministic GA tested with Monte Carlo simulation post-optimisation.

Keywords: multi-objective; uncertainty; biopharmaceutical; capacity planning; scheduling; genetic algorithm

1. Introduction

The business landscape of the biopharmaceutical sector is defined by expensive and lengthy research and development (R&D) processes with high risks of clinical failure. Biopharmaceutical products are immensely sophisticated requiring substantial investment of capital and human resources, and technical expertise to produce them. Biopharmaceutical facilities are reported to typically take 3-5 years to build and to cost in the order of $40-650M depending on the scale (Farid, 2007). Nevertheless, the success of biopharmaceutical products in treating complex diseases has meant that the industry has experienced constant and significant growth since its inception in 1980s. For example, the global revenues of biopharmaceuticals were reported to be over $100B in 2010 (Walsh, 2010) and over $150B in 2014 (Otto et al., 2014). The overall annual industry growth has been estimated at 14-15% (Langer & Rader, 2017).

Managing manufacturing facility assets for these growing and dynamic biopharmaceutical portfolios requires careful capacity planning and scheduling. However, it is complicated by the unique features of biopharmaceutical production. Based on the report by Langer (2009), key factors quoted as leading to biopharmaceutical production capacity constraints include lack of financing for production expansion, inability to optimise the overall system and general inability to meet demands for finished products.

Meeting product demand in the biopharmaceutical industry is a highly sensitive issue owing to the high value and clinical importance of the products. However, the market demand is often not known in advance and must be estimated. For example, the demand for Enbrel® for rheumatoid arthritis was higher than anticipated when it was launched and could not be accommodated even after increasing volume with an existing contract manufacturing organisation (CMO) (Kamarck, 2006). Malik et al. (2002) estimated that the lack of manufacturing capacity for Enbrel cost the companies involved more than $200M in lost revenue in 2001.

The area of planning and scheduling in the biopharmaceutical industry has not received as much attention as bioprocess design and optimisation (Majozi et al., 2015). Discrete-event simulation-based approaches have been especially popular at modelling the impact of uncertainties within the biopharmaceutical manufacturing environment for more effective use of resources and improved economic performance. Examples include the impact of uncertainties in factors such as process yields and equipment failure on biopharmaceutical decisions related to the following domains: stainless steel versus single-use technologies (Farid, 2007; S. S. Farid et al., 2005), batch versus perfusion culture (Lim et al., 2006; Pollock et al, 2013), and the robustness of legacy purification facilities to increasing cell culture titres. Stonier et al. (2012) and Yang et al. (2014) further leveraged the stochastic datasets generated from such tools with datamining tools (e.g. principal component analysis, clustering algorithms, decision trees) to predict the root cause of facility fit issues.

Mathematical programming approaches to bioprocess design have included mixed-integer linear programming (MILP) (Vasquez-Alvarez & Pinto, 2004), mixed-integer non-linear programming (MINLP) (Asenjo et al., 2000; Montagna et al., 2000) and hybrid simulation and mixed-integer optimization approaches (Brunet et al. (2012). Simaria et al. (2012) and Allmendinger et al. (2014) proposed multi-objective GA-based approaches for the optimisation of bioprocess design under uncertainty with a focus on decisions related to purification sequences and equipment sizing strategies under uncertainty. A similar problem was also addressed by Liu et al. (2016) using a stochastic MILP model with chance constrained programming (CCP).

One of the first frameworks for biopharmaceutical capacity planning and scheduling was developed by Samsatli and Shah (1996). They addressed the design and short-term scheduling of biopharmaceutical processes using MILP and State Task Network (STN) formulations. The first medium-term capacity planning model for a multi-product, multi-suite biopharmaceutical facility was introduced by Lakhdar et al. (2005). The work was later extended to tackle planning under uncertainty (Lakhdar et al., 2006) as well as multiple objectives and facilities, and a longer planning horizon (Lakhdar et al., 2007). Siganporia et al. (2014) proposed a large-scale discrete-time MILP model to optimise long-term capacity plans for a portfolio of biopharmaceutical products, with either batch or perfusion bioprocesses, across multiple facilities to meet quarterly demand. Jankauskas et al. (2017) developed a variable-length GA-based optimisation approach for medium-term capacity planning of a multi-product, multi-suite biopharmaceutical facility using a continuous-time representation. Taking inspiration from GA-based approaches to job-shop scheduling, Oyebolu et al. (2017) proposed a problem-tailored construction heuristic for scheduling demands of multiple products sequentially across several facilities to generate a long-term manufacturing schedule. Jankauskas et al. (2018) developed fast genetic algorithm approaches for generating near optimal solutions to medium- and long-term biopharmaceutical planning problems formulated originally as discrete-time MILP models.

The shortcomings of discrete-time based models (e.g. inaccuracies due to the approximation of the time horizon, and increases in the overall problem size, due to the introduction of a large number of binary variables associated with each discrete time interval) have been reported in the literature (Floudas & Lin, 2004). To address these drawbacks, methods based on continuous-time representations have received recent attention. The Lakhdar et al. (2005) discrete-time model for multi-period scheduling of a multi-stage, multi-product process has been compared with continuous-time MILP models based on an STN (Kabra et al. (2013) and based on a Resource Task Network (RTN) (Vieira et al. (2016) , with reported improved objective function values.

One of the most attractive features of heuristics compared to mathematical programming is that they can be easily integrated with other methods such as Monte Carlo simulation which can be used to represent complex problem features, multiple objectives, and uncertainties that cannot be straightforwardly modelled by mathematical equations. A general simulation-based optimization method comprises an optimization part that guides the search process and a simulation part used to evaluate performances of candidate solutions. Compared with mathematical programming techniques, simulation-based optimization methods replace the analytical objective function and constraints by one or more simulation models. Iteratively the output of the simulation is used by the underlying optimisation algorithm, such as GA, to guide the search for the optimal solution(s). Blum et al. (2011) provides a survey of some of the most important hybrid metaheuristics in combinatorial optimisation. A more recent survey by Gutjahr and Pichler (2016) includes reviews of non-scalarising mathematical programming- and heuristic-based stochastic multi-objective optimisation. For example, multi-objective GAs combined with simulation procedures have been developed for supply chain optimization applications in the automative and textile sectors (Amodeo et al., 2009; Ding et al., 2006; Syberfeldt et al., 2009). Senties et al. (2010) embedded an artificial neural network embedded into a multiobjective GA for multi-decision scheduling problems in a semiconductor wafer fabrication environment. Dong and Wang (2012) presented a hybrid permutation-based differential evolution for scheduling of large-scale zero-wait batch processes with setup times. Costa (2015) developed a hybrid genetic optimisation approach with two stage encoding and a local search for solving a real-world parallel machines scheduling problem from the pharmaceutical environment. Lau and Srinivasan (2016) presented a GA and simulation based GPU-accelerated approach for process monitoring and optimisation. Frazzon et al. (2018) combined MILP, discrete event simulation, and a GA for integrated scheduling of production and transport processes along supply chains. Engin and Güçlü (2018) proposed a hybrid ant colony algorithm for no-wait flow shop scheduling problems. Koller et al. (2018) identified optimal design, control, and scheduling decisions for multiproduct continuous stirred tank reactor subject to stochastic uncertainty and disturbance using a Monte Carlo simulation-based back-off algorithm. This paper presents the first application of a novel variable-length multi-objective GA combined with Monte Carlo simulation for capacity planning of a multi-product biopharmaceutical facility.

In this paper, the deterministic single-objective variable-length GA presented in Jankauskas et al. (2017) is extended with multi-objective and Monte Carlo simulation components for the generation of medium-term production schedules that are robust to the variations in product demand. For the sake of brevity, the integrated Monte Carlo simulation and multi-objective GA approach will be referred to as the *stochastic GA* while the multi-objective GA without the embedded Monte Carlo simulation will be referred to as the *deterministic GA*. The advantages of the stochastic GA over the deterministic one will be demonstrated by comparing the results generated when the uncertainty in demand is ignored, by using only the most likely demand values, and when it is accounted for, by characterising it with a triangular probability distribution.

The paper is organised as follows: Section 2 provides the input data and defines the biopharmaceutical scheduling problem with uncertain product demand. Section 3 describes the variable-length chromosome structure, the key parts of the GA, the continuous-time scheduling heuristic for decoding chromosomes into production schedules and evaluating the objective values, and how Monte Carlo simulation is integrated with the GA to generate production schedules under demand uncertainty. Additionally, the section explains how the performance of the stochastic GA is improved by accelerating the computationally expensive Monte Carlo simulations using a GPU. The results and discussion of the comparison between the stochastic GA (GA with Monte Carlo simulation embedded in the optimisation) and the deterministic GA are given in Section 4.

2. Problem Definition

The statement of the medium-term multi-objective biopharmaceutical capacity planning and scheduling problem in this paper is as follows:

- *Given:*
  - A start date (1-Dec-2016) and a planning horizon of 3 years.
  - A set of biopharmaceutical products *{ A, B, C, D }.*
  - Upstream processing (*USP*) and downstream processing (DSP) times.
  - Product-dependent manufacturing yields.
  - Rolling product sequence-dependent changeovers.
  - Varying amounts of product stock available at the beginning of the schedule.
  - Desired minimum and maximum number of batches per individual product campaign.
  - Unique manufacturing requirements to produce the batches in multiples of a specified number.
  - QC/QA approval times.

    - 3-year profile of strategic product inventory targets.
    - 3-year profile of uncertain monthly product demand.
- *Determine:*
    - A set of production schedules.
    - The number and length of manufacturing campaigns.
    - Production quantities along with inventory profiles.
- *So as to (constrained stochastic multi-objective problem):*
    - Maximise the total production throughput.
    - Minimise the median total inventory deficit, i.e. cumulative differences between the monthly product inventory levels and the strategic inventory targets.
- *Subject to:*
    - The median total backlog (amount of missed orders) being no greater than 0 kg.

It is assumed that the biopharmaceutical facility is available during the entire 3-year (1096-day) planning horizon. The product demand is assumed to be due on the first day of each calendar month. The products must undergo a 90-day QC/QA process before they can be delivered which must be taken into consideration when meeting the product demand. For example, if a demand for a certain product is due on 31 March 2018, then the material must be manufactured by 31 December 2017. Product sequence-dependent changeover time (Table 1) is incurred only when there is a switch between different product campaigns. Each product has a different manufacturing yield which determines how many kilograms are produced in a single batch. Due to the QC/QA approval process, there is a certain amount of product stock made available at the beginning of the schedule to meet the initial product demand. Additionally, due to specific *DSP* requirements, product D needs to be produced in multiples of 3 batches. The complete process data for the industrial case study is provided in Table 2. The strategic product inventory monthly targets are listed in Table 3.

In certain cases, the exact pricing and cost data can be difficult to obtain. Under such circumstances, other non-economic objectives can be considered, e.g. minimisation of makespan, earliness or lateness, maximization of throughput, and minimisation of backlogs. Furthermore, economic-based measures are commonly reported in the literature. This paper provides a unique contribution to the processes scheduling field by demonstrating the optimisation of multiple non-economic objectives. High throughput (approximates capacity utilization of the facility), well-maintained inventory (for contingency planning), and minimisation of backlogs (meeting demand on time) are very important not only in the biopharmaceutical but other industries as well. The proposed scheduling optimisation approach is not dependent on the problem objectives. The same approach can be effectively applied to similar biopharmaceutical scheduling problems with economic-based objectives.

One of the objectives is to minimise the total inventory deficit which is defined as the cumulative sum of the differences between the product inventory levels and the corresponding strategic monthly targets whenever the latter are greater than the former. The product demand profile in Table 4 is set based on the discussions with the industrial sponsor. Triangular distribution is used to approximate the outcomes where the minimum, most likely, and maximum amounts were available whereas 0.0 kg is applied on due dates where there were no orders in the past. The total inventory deficit and total backlog will have a corresponding distribution of different values depending on the randomly generated product demand scenarios during Monte Carlo simulation. Overall, the goal of the stochastic GA is to generate a set of schedules that would maximise the total production throughput and minimise the median total inventory deficit subject to the median total backlog being no greater than 0 kg.

3. Methods

This paper is an extension of the earlier work (Jankauskas et al., 2017) which presented a variable-length chromosome structure and a set of new genetic operators to automatically determine the optimal order, number, and length of production campaigns for a single-objective biopharmaceutical capacity planning and scheduling problem. In this section, the key components of the GA are described. The details of the continuous-time scheduling heuristic for evaluating the fitness of each chromosome and constructing schedules, the implementation details of Monte Carlo simulation, and the steps taken to improve the performance of the stochastic multi-objective GA-based framework are also discussed.

In Section 4, Mann-Whitney U tests are applied to evaluate the stochastic optimisation results and compare them with the deterministic outcome. Mann-Whitney U test is a non-parametric alternative to independent samples *t-test* with a null hypothesis H_0 that it is equally likely that a random observation from a distribution X will be less than or greater than a random observation from distribution Y*.* The effect size can be evaluated by calculating the point estimate of the Hodges-Lehmann’s median difference Δ_{XY} which is equal to the median of all pairwise differences between the two distributions X and Y (Hodges Jr & Lehmann, 1963).

Using unary performance indicators to assess the performance of multi-objective algorithms can be problematic (Zitzler et al., 2003). Nevertheless, the hypervolume indicator is often used for assessing the performance of many multi-objective evolutionary algorithms (Fonseca et al., 2006; Knowles et al., 2003; Zitzler & Künzli, 2004). In this work, an improved dimension-sweep algorithm proposed by (Fonseca et al., 2006) and provided by the Distributed Evolutionary Algorithms in Python (DEAP) framework (Fortin et al., 2012) is used to estimate the hypervolume indicator. Python was also used for data input/output, and visualization.

The proposed approach has been implemented using C++ programming language and CUDA 8.0 API and compiled with NVCC v8.0. The industrially-relevant capacity planning and scheduling problem of medium-term multi-product biopharmaceutical manufacture under uncertainty has been solved on Intel i5-6500 (CPU) and NVIDIA GTX-1060 (GPU) based Windows 10 system with 16GB of RAM and 6 GB of VRAM.

3.1. Chromosome Structure

Due to the variable and more flexible nature it becomes more difficult to model scheduling problems using a continuous-time representation. Such MILP models often tend to have more complicated formulations than the discrete-time based alternatives. Moreover, the usefulness and computational efficiency of the continuous-time formulation depend on the number of predefined *event points* (Méndez et al., 2006). If the global optimum of the scheduling problem requires at least *n* points then fewer points will lead to sub-optimal or even infeasible solutions whereas a large number of points will result in long computation times. Since the number of points is not known in advance, it is usually determined iteratively by increasing it until there is no improvement in the objective function. In certain cases, a substantial number of model instances need to be solved for each scheduling problem. Furthermore, this stopping criterion does not guarantee the optimality of the schedule and may terminate with a sub-optimal solution.

In this work, a variable-length chromosome structure is utilised to explore the decision space by simultaneously varying both the total number as well as the length of individual product campaigns without specifying the number of product campaigns in advance. Every variable-length chromosome encodes a continuous-time production schedule as a 1-D vector of unit elements known as *genes*. Every gene in the vector is a minimal representation of a production campaign which is encoded as a combination of a product label and an integer number of batches. Figure 1 displays an example of what a variable-length chromosome is like at the start (GEN 0) and after 100 generations (GEN 100) of the GA have elapsed. While it is possible to set how many genes within each chromosome would be generated at the beginning of the GA, the algorithm presented in this work is designed to evolve the candidate solutions from a single gene. This is accomplished by modifying traditional genetic operators, e.g. uniform crossover, as well as introducing a few new ones to add a new random gene at the end of every GA generation and to mutate the old ones. The order of the genes (from left to right) defines the timing of each manufacturing campaign, e.g. the second gene in the chromosome encodes the second manufacturing campaign in the production schedule. The initial population is created by generating a pool of random chromosomes containing a single gene. With the aid of special genetic operators, the chromosomes grow and shrink in length over the course of the GA.

3.2. Crossover

The traditional uniform crossover is adapted to suit the variable-length chromosome structure. Before the crossover is applied, the chromosomes are sorted according to the number of genes they contain. This way the crossover operator is performed on similar individuals. Provided that both parent chromosomes have at least 3 genes, the genes are exchanged with a rate of 0.5 until the end of the shorter chromosome is reached. The extra genes from a longer parent are copied to the shorter one with a rate of 0.5. The crossover operator is illustrated by Figure 2.a.

3.3. Mutation

Several special gene- and chromosome-level mutation operators are used to perform the following in an order (see Figure 2.b):

1. To mutate a product label with a rate of *pMutP.*
2. To increase or decrease the number of batches by one with a rate of *pPosB and pNegB*, respectively.
3. To add a new random gene to the end of the chromosome (unconditionally).
4. To swap two genes within the same chromosome once with a rate of *pSwap*.

3.4. Genetic Algorithm

The multi-objective GA in this work is based on NSGA-II (Deb et al., 2002) which is well-known for its effectiveness at solving a wide variety of multi-objective problems, e.g. see Raisanen and Whitaker (2005) and Hamdy et al. (2016). The multi-objective variable-length GA employs a generational reproduction scheme using two populations (*parents* and *offspring*) with a fixed number of chromosomes. Parent population is used to keep track of the best solutions found, i.e. provides elitism, while the offspring population is a result of crossover, mutation, and selection operators. Figure 3 displays a schematic of the key steps of the multi-objective GA developed in this work. After the initial population of single-gene chromosomes is created and evaluated, the steps are performed continuously until the maximum number of generations is reached.

The scheduling problem of this chapter is a constrained multi-objective optimisation problem. Production schedules that are not able to meet all product demands on time are considered to be *infeasible*. Constraint handling and representation in heuristic-based optimisation is a difficult issue (Harjunkoski et al., 2014). Simpler constraints such as the fact that a valid schedule has to be a permutation of jobs or product demands can be mapped into the problem representation and into the choice of genetic operators. However, such implicit representation becomes harder with the increasing number and complexity of constraints. The most basic methods of constraint handling are to discard all infeasible solutions or to apply a penalty function. More sophisticated methods include the use of repair mechanisms to convert infeasible solutions into feasible ones during the search process or the handling of only some of the degrees of freedom by the meta-heuristic search strategy and fixing the remaining ones during the evaluation of the solution, e.g. by using local priority rules (Piana & Engell, 2010).

In this work, repairing infeasible schedules was deemed to be computationally inefficient. The penalty-based constraint handling was rejected to avoid introducing additional parameters into the model. Moreover, according to Sand et al. (2008), incorrectly applied penalty, e.g. too large, may prevent the heuristic from traversing infeasible sub-regions in disjoint search spaces. There have been several other constraint-handling approaches for the multi-objective problems reported in the literature, e.g. Fonseca and Fleming (1998) and Ray et al. (2001). For its simplicity and computational efficiency, a constraint-handling approach proposed by Deb et al. (2002) is used together with a binary tournament selection to choose more optimal, non-dominated solutions. The pseudocode for this procedure is listed in Algorithm 1. Using this approach, the solutions which do not satisfy the constraints of the problem, i.e. with a median total amount of backog greater than 0 kg, will not be selected and will be ranked lower by the NSGA-II ranking algorithm, even if the values of the objectives are better than those of the solutions which fully satisfy the constraints. Therefore, the GA initially selects the chromosomes based on the extent of constraint statisfaction.

3.5. Scheduling Heuristic

Figure 4 explains the scheduling heuristic with an illustrative example of how a two-gene chromosome is decoded into a production schedule of two manufacturing campaigns. In Figure 4.a, the chromosome contains two genes: one represents a manufacturing campaign of one batch of product A and another – a manufacturing campaign of one batch of product C. The length of each production campaign is determined based on the number of batches within each gene and the number of *USP* and *DSP* days for the corresponding product. For example, it takes 52 days in total (45 for *USP* and 7 for *DSP*) to produce 1 batch of product A. As it was mentioned previously, the order of the genes within the variable-length chromosome determines the timings of the manufacturing campaigns. Hence, the campaigns are scheduled in sequence one after another. At the first glance, it might seem that the two manufacturing campaigns in Figure 4.b overlap with each other. However, it only looks so because of the rolling product sequence-dependent changeovers. Figure 4.c illustrates how they are implemented. For example, once the Inoculation stage of product A is complete, a changeover process can begin to prepare the stage for product C while product A is in Seed stage. The product sequence-dependent changeover time is used to determine the start date of the new campaign. This is illustrated by the black and white striped box which separates the *DSP* stages of products A and C in Figure 4.c. Production stage of product C manufacturing campaign is scheduled to end 16 days (the number of changeover days) after the end of product A manufacturing campaign.

The scheduling heuristic maintains that each chromosome encodes a production schedule which starts and ends within the set planning horizon. Genes encoding production campaigns beyond the planning horizon are removed from the chromosome. The crossover and mutation operators can sometimes cause multiple, consecutive genes encode manufacturing campaigns of the same product. After the schedule has been constructed, the heuristic combines the consecutive genes encoding the campaigns of the same product into one. Figure 5 illustrates an example of this.

3.6. GPU-accelerated Monte Carlo Simulation

After the continuous-time scheduling heuristic has been applied to decode the variable-length chromosomes into production schedules, their robustness to the variations in monthly product demand is then tested by conducting Monte Carlo simulation trials. Hundreds of demand scenarios are generated for each individual production schedule based on the provided triangular probability distributions for each product and its demand due dates (Table 4). The performance of a production schedule is evaluated on each randomly generated demand scenario by calculating the total amount of inventory deficit and backlog. The objective of the total production throughput maximisation remains unchanged as the throughput from each individual manufacturing campaign is the same regardless of the product demand scenario. For each Monte Carlo simulation trial *t*, the values of inventory deficit and backlog are stored in | *t* |-dimensional arrays (see Lines 9 and 10 in Figure 3). After the simulation trials are completed, the medians of the total inventory deficit and total backlog distributions are assigned to the corresponding chromosome as the objective and constraint values.

One of the major drawbacks of Monte Carlo simulation is the associated computational overhead. The average time elapsed for a single run of stochastic GA with Monte Carlo simulation embedded into the optimisation was approximately 100-fold longer than that of a deterministic GA without Monte Carlo simulation (Figure 6.a.). Reducing the number of Monte Carlo simulation trials to improve the performance is not ideal as the error of the simulation estimates is inversely proportional to the number of trials. The larger the number of trials is, the more confident the estimates are. Hence, it was necessary to find a way to improve the performance, i.e. execution speed, without sacrificing the accuracy and confidence of the results.

Since the individual Monte Carlo simulation trials are independent form each other in this study, the overall simulation process can be made more efficient through parallelisation. Modern GPUs are optimised for Single Instruction Multiple Data (SIMD) type processing with massive parallelism. For example, compared to an average consumer-grade Central Processing Unit (CPU) which typically has from 4 to 8 cores, a single GPU can have over 2000 cores (Vanek et al., 2017). Each individual Monte Carlo trial can be assigned to a single core on a GPU thus enabling hundreds of trials to be performed in parallel with substantial savings in computational power and time.

In this work, only the Monte Carlo simulation component from the stochastic multi-objective GA-based framework was made to run on a GPU since it was found to be the biggest performance bottleneck compared to other components. The execution of the program was transferred from CPU to GPU every time

MonteCarloSimulationKernel (see Lines 3, 8-18 in Figure 3) was invoked during the objective function evaluation. Once the simulation finished, the execution of the program was transferred back to CPU to continue running the GA.

Accelerating Monte Carlo simulation with a GPU reduced the mean running time of a single stochastic GA run by approximately 30 times (see Figure 6.b). The ability to achieve solutions in a timely manner is very valuable as it would enable the production schedulers to test more scenarios and perform more case studies with different inputs in less amount of time.

4. Results and Discussion

In this section, the validity of the stochastic multi-objective GA approach outlined earlier is demonstrated on an industrially-relevant case study of medium-term multi-objective biopharmaceutical capacity planning and scheduling. The problem requires to produce a set of optimal 3-year schedules for a multi-product biopharmaceutical facility manufacturing 4 products with uncertain monthly demand. The objectives of the capacity planning and scheduling problem are to maximise the total kilogram throughput and to minimise the median total kilogram inventory deficit. The optimisation problem is also subject to the constraint of 0.0 kg median total backlog.

First, Section 4.1 defines the objective space of the stochastic optimisation problem using a single-objective GA with integrated Monte Carlo simulation. Additionally, the results obtained using the stochastic multi-objective GA are discussed by comparing the trade-offs between two non-dominated solutions selected from the extreme ends of the best Pareto front. Section 4.2 strengthens the argument for stochastic optimisation with a comparison of the production schedules generated using the stochastic and deterministic GAs. The schedules generated with the deterministic GA were tested using Monte Carlo simulation post-optimisation to show the impact of the optimisation using only the most likely values, ignoring the uncertainty aspect.

4.1. Stochastic Objective Space

In this section, the scheduling problem with uncertain demand is first solved using a single-objective GA that also has Monte Carlo simulation embedded into the objective function evaluation. The highest total production throughput and the lowest median total inventory deficit values are used to create an ideal point (see Table 5) to define the boundaries of the stochastic objective space for the scheduling problem presented in this paper (see Figure 7.a). Knowing the total hypervolume of the objective space makes it convenient to gauge the performance of the algorithm using a hypervolume indicator normalised to 0.0-1.0 range (the higher, the better).

The stochastic multi-objective GA generates a Pareto front of unique non-dominated solutions at the end of each run. However, the GA is an optimisation technique that is not guaranteed to converge on the same solution(s) every run. Therefore, the top Pareto front at the end of each individual GA run is saved. After all 50 runs are completed, the fronts are combined and sorted again using the non-dominated sorting algorithm described by Deb et al. (2002) to create the best Pareto front containing a set of top non-dominated solutions found during those 50 GA runs. Such a front of non-dominated solutions with a hypervolume of 0.997 is displayed in Figure 7.b alongside the single-objective solutions and the ideal point.

The production schedule of solution X (Figure 8.a) has a greater number of manufacturing campaigns than solution Y (Figure 8.b). The average production time per campaign of solution X is 117 days compared to 161 days for solution Y. The proposed approach predicts that more frequent but shorter manufacturing campaigns scheduled according to a recurring pattern would lead to better optimised product inventory levels. However, this is at the expense of lower total production throughput (539.3 kg vs 601.5 kg) because of the lost production time due to more frequent product changeovers. According to the Mann-Whitney U test, the difference between the total inventory deficit distributions of the solutions X and Y is statistically significant with a two-tailed *p-value* of 0.0. The value of Δ_{XY} between the total inventory deficit distributions of the solutions X and Y is -126 kg, i.e. the total inventory deficit of the solution Y is 126 kg higher on average than that of the solution X. Based on the Mann-Whitney U test, the difference between the total backlog distributions is also statistically significant (two-tailed *p-value* of 0.0) though the value of Δ_{XY} is only 0.1 kg. However, solution X has a greater probability of meeting the product demand compared to the solution Y (0.82 vs 0.50). The comparison summary is provided in Table 6.

The results of the stochastic multi-objective GA show that depending on the chosen objectives and constraints, there can be multiple alternative solutions to a scheduling problem even in the presence of uncertainty. The stochastic GA generates a set of equally good alternative production schedules. Depending on the business strategy, decision-makers can decide whether it is more acceptable to choose a production schedule that would result in higher total throughput but also a higher risk of not being able to meet the set inventory targets and/or product demands on time or vice versa. If there was more data available, e.g. costs of production and changeovers, then the decision-making process could be more effective. Solution X might be more expensive due to more frequent changeovers whereas the probable profitability of Solution Y might offset lower inventory levels. Without such data a single production schedule could be selected from the non-dominated solutions using a weighted sum method or Euclidean distance (finding a production schedule that is closest to the ideal point in the assumed objective space).

One of the key aims of this paper is to present a flexible and efficient way to create continuous-time production schedules under uncertainty. In the next section, using a comparison with the Deterministic GA, a case will be made for the importance of factoring the uncertainty as a variable directly into the optimisation which is often either overlooked or evaluated post-optimisation using sensitivity analysis.

4.2. Comparison with the Deterministic GA

This section will discuss the merits of integrating Monte Carlo simulation into the multi-objective variable-length GA for creating production schedules under product demand uncertainty.

The scheduling problem was solved again but the uncertainty in product demand was ignored and instead of the probability distributions only the most likely product demand values were used as an input. First, the objective space was defined using a deterministic single-objective GA without Monte Carlo simulation. Then, a multi-objective GA, also without Monte Carlo simulation, was used to generate the best Pareto front of deterministic solutions in a similar way that was described earlier in the last paragraph of Section 4.1. Table 7 lists the best single-objective values obtained with a deterministic single-objective GA (without Monte Carlo simulation) whereas Figure 7.c shows the boundaries of the deterministic objective space defined by the reference and ideal points.

To highlight the differences between the stochastic and the naïve, deterministic optimisation approaches, the solution X from the best Pareto front generated with the *deterministic* GA (Figure 7.d), i.e. without Monte Carlo simulation, is compared to the solution X from the best Pareto front generated using the *stochastic* GA (Figure 7.b). For convenience, the two solutions are referred to as the *deterministic solution* and *the stochastic solution*, respectively. To be able to compare the deterministic solution with the stochastic one, Monte Carlo simulation is used to conduct a stochastic analysis to assess its robustness to the variability of product demand. Both solutions are taken from the extreme ends of the respective Pareto fronts favouring the objective of total inventory deficit minimisation.

Using only the most likely demand values, the deterministic solution achieved the total throughput and total inventory deficit values of 498.5 kg and 175.4 kg respectively. The production schedules of the deterministic (Figure 9.a) and stochastic solution (Figure 9.b) look very similar: both contain short but frequent recurring product campaigns. Therefore, at the first glance, it might seem like the production schedule generated deterministically would perform similarly to the production schedule generated using a stochastic GA under uncertain product demand, i.e. have a similar median total inventory deficit and a median total backlog equal to 0.0 kg. However, after Monte Carlo simulation was applied post-optimisation to evaluate the robustness of the deterministic solution to the variability of demand, it was found that deterministically generated production schedule had a significantly lower probability of meeting product demands on time.

Figure 10 and Table 8 illustrate and list the results of the Monte Carlo simulation-based sensitivity analysis of the deterministic solution and the comparison with the stochastic solution generated using the multi-objective GA with Monte Carlo simulation embedded into the objective function evaluation. The production schedule generated using the deterministic multi-objective GA was capable of meeting all product demands in only 14 randomly generated product demand scenarios out of

1000 in total (1.4%). Moreover, the median total inventory deficit level is also higher (501.0 kg vs 424.4 kg) than that of the production schedule generated using the stochastic GA with integrated Monte Carlo simulation. According to the Mann-Whitney U-test, the difference between the total inventory distributions of the deterministic and stochastic solutions is statistically significant with a two-tailed *p-value* of 0.0 and Δ_{XY} of -71.4 kg. The difference between the total backlog distributions is also statistically significant with a two-tailed *p-value* of 0.0 and Δ_{XY} of -6.4 kg. According to Figure 11.a, the deterministic production schedule is expected to be unable to meet the demand on time for products A and C on 3 separate due dates. In comparison, the stochastic solution meets all product demands on time on average (Figure 11.b).

Based on the comparison between the deterministic and stochastic solutions, the advantages of the stochastic GA are evident. The difference between the total inventory deficit and total backlog distributions of the two solutions is statistically significant. The stochastic solution is more likely to meet all product demands on time despite the monthly variations. Moreover, the monthly product inventory levels of the stochastic solution are expected to be closer to the set strategic targets.

5. Conclusion

In this paper, the continuous-time GA-based scheduling optimisation approach was extended with Monte Carlo simulation to address an inherent and very important feature of biopharmaceutical industry – uncertainty in product demand. The monthly demand for each product was characterised with a triangular distribution defined by the minimum, most likely, and maximum kg quantities. Integrating Monte Carlo simulation into the multi-objective GA permitted the identification of more robust production schedules better suited to handle product demand fluctuations. The benefits of an integrated GA and Monte Carlo simulation approach were demonstrated by comparing it with a deterministic approach. The production schedules generated with a deterministic GA were based on the most likely demand values and did not account for the variability in product demand during each month. Hence, in scenarios where the product demand was higher than expected the solution was shown as not able to meet all product demands on time on average.

The continuous-time scheduling heuristic presented in this paper assumed that the biopharmaceutical facility was available for the entire planning horizon. However, biopharmaceutical companies often have to regularly shut down their facilities for maintenance or inspection. The variable-length chromosome could include several special genes encoding a facility shut-down taking place. This way the GA could generate production schedules with optimised start and end dates of the facility shut-down(s) without compromising the objectives and constraints of the problem. Moreover, an integrated Monte Carlo simulation and GA approach could be applied to evaluate the impact of unplanned facility shut-downs which can occur, for example, due to contamination or equipment breakdown.

For certain objectives, such as the minimisation of the total production time, the GA would need to not only determine the timings and durations of the manufacturing campaigns but also when the facility can remain idle. In this work, each gene corresponded to an actual manufacturing campaign. The start and end dates of every manufacturing campaign were inferred from the order of the genes in the variable-length chromosome and the product-dependent process durations and changeovers. The encoding strategy could be improved by adding dummy genes that do not have a product label associated with them. The number of batches in a dummy gene would encode the duration of idle time in the facility. The timings of real and dummy manufacturing campaigns can still be implicitly encoded by the order of genes within the chromosome.

Finally, the variable-length chromosome could be extended to address capacity planning and scheduling of multiple biopharmaceutical facilities. For example, each gene could be modified to encode biopharmaceutical facility and product labels, and the number of batches to be manufactured.

The stochastic multi-objective scheduling optimisation approach demonstrated in this paper provides the foundation for future incorporation of other uncertainties inherent in the biopharmaceutical manufacturing process such as variable fermentation titres and process yields, contamination risks, and QC/QA rejection rates. The integrated Monte Carlo simulation and multi-objective variable-length GA could be used to generate production schedules that have the highest probabilities of meeting the specified objectives and constraints under the aforementioned uncertainties and risks.

Acknowledgements

Financial support from the UK Engineering and Physical Sciences Research Council (EPSRC) and Eli Lilly & Co. is gratefully acknowledged. UCL Biochemical Engineering hosts the Future Targeted Healthcare Manufacturing Hub in collaboration with UK universities and with funding from the UK Engineering & Physical Sciences Research Council (EPSRC) and a consortium of industrial users and sector organisations. This is funded by an Industrial Case Award supported by EPSRC and Eli Lilly. The grant code is EP/L505717/1.

References

Allmendinger, R., Simaria, A. S., & Farid, S. S. (2014). Multiobjective evolutionary optimization in antibody purification process design. Biochemical Engineering Journal, 91, 250-264.

Amodeo, L., Prins, C., & Sánchez, D. R. (2009). Comparison of metaheuristic approaches for multi-objective simulation-based optimization in supply chain inventory management. In Workshops on Applications of Evolutionary Computation (pp. 798-807): Springer.

Asenjo, J. A., Montagna, J. M., Vecchietti, A. R., Iribarren, O. A., & Pinto, J. M. (2000). Strategies for the simultaneous optimization of the structure and the process variables of a protein production plant. Computers & Chemical Engineering, 24, 2277-2290.

Blum, C., Puchinger, J., Raidl, G. R., & Roli, A. (2011). Hybrid metaheuristics in combinatorial optimization: A survey. Applied Soft Computing, 11, 4135-4151.

Brunet, R., Guillén-Gosálbez, G., Pérez-Correa, J. R., Caballero, J. A., & Jiménez, L. (2012). Hybrid simulation-optimization based approach for the optimal design of single-product biotechnological processes. Computers & Chemical Engineering, 37, 125-135.

Costa, A. (2015). Hybrid genetic optimization for solving the batch-scheduling problem in a pharmaceutical industry. Computers & Industrial Engineering, 79, 130-147.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on evolutionary computation, 6, 182-197.

Ding, H., Benyoucef, L., & Xie, X. (2006). A simulation-based multi-objective genetic algorithm approach for networked enterprises optimization. Engineering Applications of Artificial Intelligence, 19, 609-623.

Dong, M.-G., & Wang, N. (2012). A novel hybrid differential evolution approach to scheduling of large-scale zero-wait batch processes with setup times. Computers & Chemical Engineering, 45, 72-83.

Engin, O., & Güçlü, A. (2018). A new hybrid ant colony optimization algorithm for solving the no-wait flow shop scheduling problems. Applied Soft Computing, 72, 166-176.

Farid. (2007). Process economics of industrial monoclonal antibody manufacture. Journal of Chromatography B, 848, 8-18.

Farid, S. S., Washbrook, J., & Titchener-Hooker, N. J. (2005). Decision-support tool for assessing biomanufacturing strategies under uncertainty: Stainless steel versus disposable equipment for clinical trial material preparation. Biotechnology Progress, 21, 486-497.

Floudas, C. A., & Lin, X. (2004). Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. Computers & Chemical Engineering, 28, 2109-2129.

Fonseca, C. M., & Fleming, P. J. (1998). Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 28, 26-37.

Fonseca, C. M., Paquete, L., & López-Ibánez, M. (2006). An improved dimension-sweep algorithm for the hypervolume indicator. In Evolutionary Computation, 2006. CEC 2006. IEEE Congress on (pp. 1157-1163): IEEE.

Fortin, F.-A., Rainville, F.-M. D., Gardner, M.-A., Parizeau, M., & Gagné, C. (2012). DEAP: Evolutionary algorithms made easy. Journal of Machine Learning Research, 13, 2171-2175.

Frazzon, E. M., Albrecht, A., Pires, M., Israel, E., Kück, M., & Freitag, M. (2018). Hybrid approach for the integrated scheduling of production and transport processes along supply chains. International Journal of Production Research, 56, 2019-2035.

Gutjahr, W. J., & Pichler, A. (2016). Stochastic multi-objective optimization: a survey on non-scalarizing methods. Annals of Operations Research, 236, 475-499.

Hamdy, M., Nguyen, A.-T., & Hensen, J. L. (2016). A performance comparison of multi-objective optimization algorithms for solving nearly-zero-energy-building design problems. Energy and Buildings, 121, 57-71.

Harjunkoski, I., Maravelias, C. T., Bongers, P., Castro, P. M., Engell, S., Grossmann, I. E., Hooker, J., Méndez, C., Sand, G., & Wassick, J. (2014). Scope for industrial applications of production scheduling models and solution methods. Computers & Chemical Engineering, 62, 161-193.

Hodges Jr, J. L., & Lehmann, E. L. (1963). Estimates of location based on rank tests. The Annals of Mathematical Statistics, 598-611.

Jankauskas, K., Papageorgiou, L. G., & Farid, S. S. (2017). Continuous-Time Heuristic Model for Medium-Term Capacity Planning of a Multi-Suite, Multi-Product Biopharmaceutical Facility. In Computer Aided Chemical Engineering (Vol. 40, pp. 1303-1308): Elsevier.

Jankauskas, K., Papageorgiou, L. G., & Farid, S. S. (2018). Fast genetic algorithm approaches to solving discrete-time mixed integer linear programming problems of capacity planning and scheduling of biopharmaceutical manufacture. Computers & Chemical Engineering.

Kabra, S., Shaik, M. A., & Rathore, A. S. (2013). Multi-period scheduling of a multi-stage multi-product bio-pharmaceutical process. Computers & Chemical Engineering, 57, 95-103.

Kamarck, M. E. (2006). Building biomanufacturing capacity—the chapter and verse. Nature biotechnology, 24, 503-505.

Knowles, J. D., Corne, D. W., & Fleischer, M. (2003). Bounded archiving using the Lebesgue measure. In Evolutionary Computation, 2003. CEC'03. The 2003 Congress on (Vol. 4, pp. 2490-2497): IEEE.

Koller, R. W., Ricardez-Sandoval, L. A., & Biegler, L. T. (2018). Stochastic back-off algorithm for simultaneous design, control, and scheduling of multiproduct systems under uncertainty. AIChE journal, 64, 2379-2389.

Lakhdar, K., Farid, S., Savery, J., Titchener-Hooker, N., & Papageorgiou, L. (2006). Medium term planning of biopharmaceutical manufacture under uncertainty. Computer Aided Chemical Engineering, 21, 2069-2074.

Lakhdar, K., Savery, J., Papageorgiou, L., & Farid, S. (2007). Multiobjective Long-Term Planning of Biopharmaceutical Manufacturing Facilities. Biotechnology Progress, 23, 1383-1393.

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

Langer, E. (2009). Trends in capacity utilization for therapeutic monoclonal antibody production. In MAbs (Vol. 1, pp. 151-156): Taylor & Francis.

Langer, E., & Rader, R. A. (2017). Top Trends in Biopharmaceutical Manufacturing, 2017. Pharmaceutical Technology, 41.

Lau, M. C., & Srinivasan, R. (2016). A hybrid CPU-Graphics Processing Unit (GPU) approach for computationally efficient simulation-optimization. Computers & Chemical Engineering, 87, 49-62.

Liu, S., Farid, S. S., & Papageorgiou, L. G. (2016). Integrated optimization of upstream and downstream processing in biopharmaceutical manufacturing under uncertainty: a chance constrained programming approach. Industrial & Engineering Chemistry Research, 55, 4599-4612.

Majozi, T., Seid, E. R., & Lee, J.-Y. (2015). Synthesis, Design, and Resource Optimization in Batch Chemical Plants: CRC Press.

Malik, A., Pinkus, G., & Sheffer, S. (2002). Biopharma's capacity crunch. McKinsey Quarterly. In.

Méndez, C. A., Cerdá, J., Grossmann, I. E., Harjunkoski, I., & Fahl, M. (2006). State-of-the-art review of optimization methods for short-term scheduling of batch processes. Computers & Chemical Engineering, 30, 913-946.

Montagna, J. M., Vecchietti, A. R., Iribarren, O. A., Pinto, J. M., & Asenjo, J. A. (2000). Optimal design of protein production plants with time and size factor process models. Biotechnology Progress, 16, 228-237.

Otto, R., Santagostino, A., & Schrader, U. (2014). Rapid growth in biopharma: Challenges and opportunities. McKinsey & Company.

Oyebolu, F. B., van Lidth de Jeude, J., Siganporia, C., Farid, S. S., Allmendinger, R., & Branke, J. (2017). A new lot sizing and scheduling heuristic for multi-site biopharmaceutical production. Journal of heuristics, 23, 231-256.

Piana, S., & Engell, S. (2010). Hybrid evolutionary optimization of the operation of pipeless plants. Journal of heuristics, 16, 311-336.

Raisanen, L., & Whitaker, R. M. (2005). Comparison and evaluation of multiple objective genetic algorithms for the antenna placement problem. Mobile Networks and Applications, 10, 79-88.

Ray, T., Tai, K., & Seow, C. (2001). An evolutionary algorithm for multiobjective optimization. Eng. Optim, 33, 399-424.

Samsatli, N., & Shah, N. (1996). An optimization based design procedure for biochemical processes: Part II: Detailed scheduling. Food and bioproducts processing, 74, 232-242.

Sand, G., Till, J., Tometzki, T., Urselmann, M., Engell, S., & Emmerich, M. (2008). Engineered versus standard evolutionary algorithms: A case study in batch scheduling with recourse. Computers & Chemical Engineering, 32, 2706-2722.

Senties, O. B., Azzaro-Pantel, C., Pibouleau, L., & Domenech, S. (2010). Multiobjective scheduling for semiconductor manufacturing plants. Computers & Chemical Engineering, 34, 555-566.

Siganporia, C. C., Ghosh, S., Daszkowski, T., Papageorgiou, L. G., & Farid, S. S. (2014). Capacity planning for batch and perfusion bioprocesses across multiple biopharmaceutical facilities. Biotechnology Progress.

Simaria, A. S., Turner, R., & Farid, S. S. (2012). A multi-level meta-heuristic algorithm for the optimisation of antibody purification processes. Biochemical Engineering Journal, 69, 144-154.

Stonier, A., Simaria, A. S., Smith, M., & Farid, S. S. (2012). Decisional tool to assess current and future process robustness in an antibody purification facility. Biotechnology Progress, 28, 1019-1028.

Syberfeldt, A., Ng, A., John, R. I., & Moore, P. (2009). Multi-objective evolutionary simulation-optimisation of a real-world manufacturing problem. Robotics and Computer-Integrated Manufacturing, 25, 926-931.

Vanek, J., Michálek, J., & Psutka, J. (2017). A Comparison of Support Vector Machines Training GPU-Accelerated Open Source Implementations. arXiv preprint arXiv:1707.06470.

Vasquez-Alvarez, E., & Pinto, J. (2004). Efficient MILP formulations for the optimal synthesis of chromatographic protein purification processes. Journal of Biotechnology, 110, 295-311.

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

Walsh, G. (2010). Biopharmaceutical benchmarks 2010. Nature biotechnology, 28, 917.

Yang, Y., Farid, S. S., & Thornhill, N. F. (2014). Data mining for rapid prediction of facility fit and debottlenecking of biomanufacturing facilities. Journal of Biotechnology, 179, 17-25.

Zitzler, E., & Künzli, S. (2004). Indicator-based selection in multiobjective search. In International Conference on Parallel Problem Solving from Nature (pp. 832-842): Springer.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & Da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transactions on evolutionary computation, 7, 117-132.

List of Tables

using the deterministic multi-objective GA without the embedded Monte Carlo simulation-based optimisation.

Table 8. A comparison between the stochastic and the deterministic solutions.

Table 1. Product sequence-dependent changeovers [days].

		To product			
		A	**B**	**C**	**D**
From product	**A**	0	10	16	20
	B	16	0	16	20
	C	16	10	0	20
	D	18	10	18	0

Table 2. Process data for the industrial case study.

	Product			
	A	**B**	**C**	**D**
USP duration [days]	45	36	45	49
DSP duration [days]	7	11	7	7
QC/QA duration [days]	90	90	90	90
Yield per batch [kg]	3.1	6.2	4.9	5.5
Opening stock [kg]	18.6	0.0	19.6	33.0
Minimum batch throughput per campaign	2	2	2	3
Maximum batch throughput per campaign	50	50	50	30
Produce batches per campaign in multiples of	1	1	1	3

Table 3. Strategic inventory targets [kg].

	Product			
Due date	**A**	**B**	**C**	**D**
1-Jan-17	6.2	0.0	0.0	22.0
1-Feb-17	6.2	0.0	4.9	27.5
1-Mar-17	9.3	0.0	9.8	27.5
1-Apr-17	9.3	0.0	9.8	27.5
1-May-17	12.4	0.0	9.8	27.5
1-Jun-17	12.4	0.0	9.8	33.0
1-Jul-17	15.5	0.0	19.6	33.0
1-Aug-17	21.7	0.0	19.6	27.5
1-Sep-17	21.7	0.0	14.7	27.5
1-Oct-17	24.8	0.0	19.6	27.5
1-Nov-17	21.7	0.0	19.6	38.5
1-Dec-17	24.8	0.0	19.6	33.0
1-Jan-18	27.9	0.0	14.7	33.0
1-Feb-18	21.7	0.0	19.6	33.0
1-Mar-18	24.8	0.0	19.6	33.0
1-Apr-18	24.8	0.0	14.7	33.0
1-May-18	24.8	0.0	14.7	27.5
1-Jun-18	27.9	6.2	19.6	33.0
1-Jul-18	27.9	6.2	19.6	33.0
1-Aug-18	27.9	6.2	9.8	33.0
1-Sep-18	31.0	6.2	19.6	38.5
1-Oct-18	31.0	6.2	19.6	33.0
1-Nov-18	34.1	6.2	19.6	38.5
1-Dec-18	34.1	6.2	19.6	33.0
1-Jan-19	27.9	6.2	24.5	33.0
1-Feb-19	27.9	6.2	34.3	33.0
1-Mar-19	27.9	6.2	24.5	33.0
1-Apr-19	27.9	6.2	29.4	44.0
1-May-19	34.1	6.2	39.2	33.0
1-Jun-19	34.1	6.2	39.2	33.0
1-Jul-19	31.0	6.2	29.4	33.0
1-Aug-19	31.0	6.2	19.6	33.0
1-Sep-19	21.7	6.2	19.6	22.0
1-Oct-19	15.5	6.2	14.7	11.0
1-Nov-19	6.2	6.2	4.9	11.0
1-Dec-19	0.0	6.2	0.0	5.5

Table 4. Product demand uncertainty for a 3-year period.

	Product			
Due date	**A**	**B**	**C**	**D**
1-Jan-17	0.0	0.0	0.0	0.0
1-Feb-17	0.0	0.0	0.0	Tr(4.5, 5.5, 8.25)
1-Mar-17	Tr(2.1, 3.1, 4.65)	0.0	0.0	Tr(4.5, 5.5, 8.25)
1-Apr-17	0.0	0.0	0.0	0.0
1-May-17	0.0	0.0	0.0	Tr(4.5, 5.5, 8.25)
1-Jun-17	Tr(2.1, 3.1, 4.65)	0.0	0.0	Tr(4.5, 5.5, 8.25)
1-Jul-17	0.0	0.0	Tr(3.9, 4.9, 7.35)	Tr(4.5, 5.5, 8.25)
1-Aug-17	Tr(2.1, 3.1, 4.65)	0.0	Tr(3.9, 4.9, 7.35)	Tr(4.5, 5.5, 8.25)
1-Sep-17	Tr(2.1, 3.1, 4.65)	0.0	0.0	Tr(4.5, 5.5, 8.25)
1-Oct-17	Tr(2.1, 3.1, 4.65)	0.0	0.0	0.0
1-Nov-17	0.0	0.0	0.0	Tr(10, 11, 16.5)
1-Dec-17	Tr(5.2, 6.2, 9.3)	0.0	Tr(8.8, 9.8, 14.7)	Tr(4.5, 5.5, 8.25)
1-Jan-18	Tr(5.2, 6.2, 9.3)	0.0	Tr(3.9, 4.9, 7.35)	0.0
1-Feb-18	Tr(2.1, 3.1, 4.65)	0.0	0.0	Tr(4.5, 5.5, 8.25)
1-Mar-18	Tr(5.2, 6.2, 9.3)	0.0	Tr(3.9, 4.9, 7.35)	Tr(4.5, 5.5, 8.25)
1-Apr-18	0.0	0.0	0.0	Tr(10, 11, 16.5)
1-May-18	Tr(2.1, 3.1, 4.65)	0.0	0.0	Tr(4.5, 5.5, 8.25)
1-Jun-18	Tr(8.3, 9.3, 13.95)	0.0	Tr(3.9, 4.9, 7.35)	Tr(4.5, 5.5, 8.25)
1-Jul-18	0.0	0.0	Tr(8.8, 9.8, 14.7)	0.0
1-Aug-18	Tr(5.2, 6.2, 9.3)	0.0	0.0	Tr(4.5, 5.5, 8.25)
1-Sep-18	Tr(5.2, 6.2, 9.3)	0.0	0.0	Tr(4.5, 5.5, 8.25)
1-Oct-18	0.0	0.0	0.0	Tr(4.5, 5.5, 8.25)
1-Nov-18	Tr(5.2, 6.2, 9.3)	Tr(5.2, 6.2, 9.3)	Tr(3.9, 4.9, 7.35)	Tr(10, 11, 16.5)
1-Dec-18	Tr(8.3, 9.3, 13.95)	0.0	Tr(3.9, 4.9, 7.35)	Tr(4.5, 5.5, 8.25)
1-Jan-19	0.0	0.0	0.0	0.0
1-Feb-19	Tr(8.3, 9.3, 13.95)	0.0	Tr(8.8, 9.8, 14.7)	Tr(10, 11, 16.5)
1-Mar-19	Tr(5.2, 6.2, 9.3)	0.0	0.0	0.0
1-Apr-19	Tr(2.1, 3.1, 4.65)	0.0	0.0	Tr(10, 11, 16.5)
1-May-19	Tr(5.2, 6.2, 9.3)	Tr(5.2, 6.2, 9.3)	Tr(3.9, 4.9, 7.35)	Tr(4.5, 5.5, 8.25)
1-Jun-19	Tr(2.1, 3.1, 4.65)	0.0	Tr(8.8, 9.8, 14.7)	Tr(4.5, 5.5, 8.25)
1-Jul-19	0.0	0.0	Tr(8.8, 9.8, 14.7)	0.0
1-Aug-19	Tr(8.3, 9.3, 13.95)	0.0	0.0	Tr(10, 11, 16.5)
1-Sep-19	Tr(5.2, 6.2, 9.3)	0.0	Tr(3.9, 4.9, 7.35)	Tr(10, 11, 16.5)
1-Oct-19	Tr(8.3, 9.3, 13.95)	0.0	Tr(8.8, 9.8, 14.7)	0.0
1-Nov-19	Tr(5.2, 6.2, 9.3)	0.0	Tr(3.9, 4.9, 7.35)	Tr(4.5, 5.5, 8.25)
1-Dec-19	0.0	Tr(5.2, 6.2, 9.3)	0.0	Tr(4.5, 5.5, 8.25)

Note: Tr(x, y, z) denotes a triangular distribution where x, y, and z are the minimum, mode (most likely), and maximum values in kg.

Algorithm 1. Procedure for binary tournament multi-objective selection based on constrained-domination (Deb et al., 2002). DetermineDominance procedure returns an integer flag of 1 if solution q dominates p, -1 if p dominates q, and 0 if both solutions are non-dominated.

```
procedure Select(q, p)
    flag = DetermineDominance(q, p)
    if flag == 1
        return q
    else if flag == -1
        return p
    end if
    if q.d > p.d ▻ if both q and p are non-dominated select the solution with a larger crowding distance
        return q
    else if p.d > q.d
        return p
    end if
    Randomly select between q and p if both solutions have the same crowding distance
end procedure

procedure DetermineDominance(q, p)
    if q.constraints != p.constraints          ▻ constraints variable is equal to the sum of all constraint violations
        if q.constraints < p.constraints
            return 1
        return -1
    end if
    q_dominates = false
    p_dominates = false
    for each objective                          ▻ all objectives are assumed to be minimised
        if q.objective < p.objective
            q_dominates = true
        else if p.objective < q.objective
            p_dominates = true
        end if
    end for
    if q_dominates == true and p_dominates == false
        return 1
    else if p_dominates == true and q_dominates == false
        return -1
    end  if
    return 0
end procedure
```

Table 5. The best values of each objective (bold) obtained with the stochastic single-objective GA and the boundary solutions X and Y of the best Pareto front generated using the stochastic multi-objective GA (hypervolume of 0.997).

	Stochastic single-objective solution		**Stochastic Pareto solution**	
	Max total throughput	**Min median total inventory deficit**	**X**	**Y**
Total throughput [kg]	**602.1**	514.3	539.3	**601.5**
Median total inventory deficit [kg]	555.2	**423.1**	**424.4**	551.7
Median total backlog [kg]	0.0	0.0	0.0	0.0
No. Monte Carlo simulation trials[1]	1000			
No. runs	50			
No. generations	1000			
No. chromosomes	100			
Starting length[1]	1			
pC	0.11			
pMutP	0.04			
pPosB	0.61			
pNegB	0.77			
pSwap	0.47			
Run time[2] [s]	8.9	9.1	10.8	

[1] The starting number of genes per chromosome in the initial population.

[2] Mean run time of a single GA run.

Table 6. Comparison of the solutions X and Y from the best Pareto front after 50 GA runs generated using the stochastic GA.

	Stochastic Pareto front solution	
	X	**Y**
Total throughput [kg]	539.3	601.5
Max total backlog [kg]	8.2	16.0
Mean total backlog [kg]	0.2 ± 0.6	7.1 ± 4.3
Median total backlog [kg][1]	0.0	0.0
Min total backlog [kg]	0.0	0.0
P(total backlog ≤ 0 kg)	0.82	0.50
Δ_{XY} (total backlog) [kg]	-0.1	
Max total inventory deficit [kg]	683.4	786.0
Mean total inventory deficit [kg][1]	432.6 ± 58.6	558.6 ± 59.0
Median total inventory deficit [kg]	424.4	551.7
Min total inventory deficit [kg]	259.4	355.6
Δ_{XY} (total inventory deficit) [kg]	-126.0	

[1] Mean ± 1 standard deviation.

Table 7. The best values of each objective obtained with a deterministic single objective GA and the boundary solutions X and Y of the best Pareto front generated using the deterministic multi-objective GA without the embedded Monte Carlo simulation-based optimisation.

	Deterministic single-objective solution		**Deterministic Pareto solution**	
	Max total throughput	**Min total inventory deficit**	**X**	**Y**
Total throughput [kg]	**630.4**	488.2	498.5	**630.4**
Total inventory deficit [kg]	464.3	**174.8**	**175.4**	461.4
Total backlog [kg]	0.0	0.0	0.0	0.0
No. runs	50			
No. generations	1000			
No. chromosomes	100			
Starting length[1]	1			
pC	0.11			
pMutP	0.04			
pPosB	0.61			
pNegB	0.77			
pSwap	0.47			
Run time[2] [s]	0.8		3.1	

[1] The starting number of genes per chromosome in the initial population.

[2] Mean run time of a single GA run

Table 8. A comparison between the stochastic and the deterministic solutions after 50 GA runs.

	Solution	
	Stochastic	**Deterministic**
Total throughput [kg]	539.3	498.5
Max total backlog [kg]	8.2	27.1
Mean total backlog [kg][1]	0.2 ± 0.6	7.1 ± 4.3
Median total backlog [kg]	0.0	6.0
Min total backlog [kg]	0.0	0.0
P(total backlog ≤ 0 kg)	0.82	0.01
Δ_{XY} (total backlog) [kg]	-6.4	
Max total inventory deficit [kg]	683.4	776.5
Mean total inventory deficit [kg][1]	432.6 ± 58.6	504.8 ± 74.1
Median total inventory deficit [kg]	424.4	501.0
Min total inventory deficit [kg]	259.4	238.4
Δ_{XY} (total inventory deficit) [kg]	-71.4	

[1] Mean ± 1 standard deviation.

Note: the stochastic solution was generated using a multi-objective GA with Monte Carlo simulation embedded into the optimisation of the objectives. The deterministic solution was obtained using a multi-objective GA without the integrated Monte Carlo simulation. Instead, Monte Carlo simulation was used to perform a post-optimisation sensitivity analysis of the solution was performed using it.

List of Figures

Figure 5. Correction of the mapping of genes to the production campaigns. In (a), the genes 2 and 3 correspond to the same product. The continuous-time scheduling heuristic combines them into (b) one contiguous manufacturing campaign and re-maps it to (c) a single a gene.

Figure 6. Average elapsed time for each of the 50 GA runs with 100 chromosomes for 1000 generations (the total runtime of 50 runs is divided by 50):

(a) Deterministic GA vs. CPU-only stochastic GA.

(b) Stochastic GA with Monte Carlo simulation performed on a GPU vs. CPU-only stochastic GA.

Note: fitness evaluations of both deterministic and CPU-only stochastic GAs were performed in parallel using the OpenMP API.

Figure 7. (a) Stochastic objective space. (b) The best Pareto front generated using the stochastic multi-objective GA (hypervolume of 0.997). The gray shaded area is used for illustrative purposes to show the area of the objective space that is dominated by the Pareto front solutions. (c) Deterministic objective space. (d) The best Pareto front generated using the deterministic multi-objective GA (hypervolume of 0.996).

Figure 8. Production schedules of (a) solution X and (b) Y from the best Pareto front after 50 runs generated using the stochastic GA. The numbers in the boxes show how many kilograms are being manufactured, followed by the production time (days).

Figure 9. Production schedules of (a) the deterministic solution X and (b) stochastic solution X from the respective best Pareto fronts. The numbers in the boxes show how many kilograms are being manufactured, followed by the production time (days).

Figure 10. A comparison of:

(a) The total inventory deficit.

(b) Total backlog distributions between the stochastic and deterministic solutions. after the stochastic analysis with Monte Carlo simulation.

Figure 11. Individual product (🟩 A 🟪 B 🟫 C 🟥 D) inventory profiles of:

a) The deterministic solution after the stochastic analysis with Monte Carlo simulation.

b) The stochastic solution.

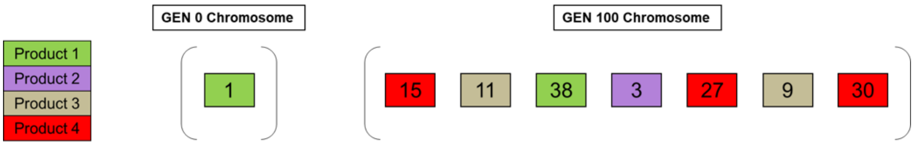


Figure 1. Variable-length chromosome at the start (GEN 0) and end of the GA (GEN 100). The values in the boxes correspond to the number of batches produced. The product label is denoted by the color.

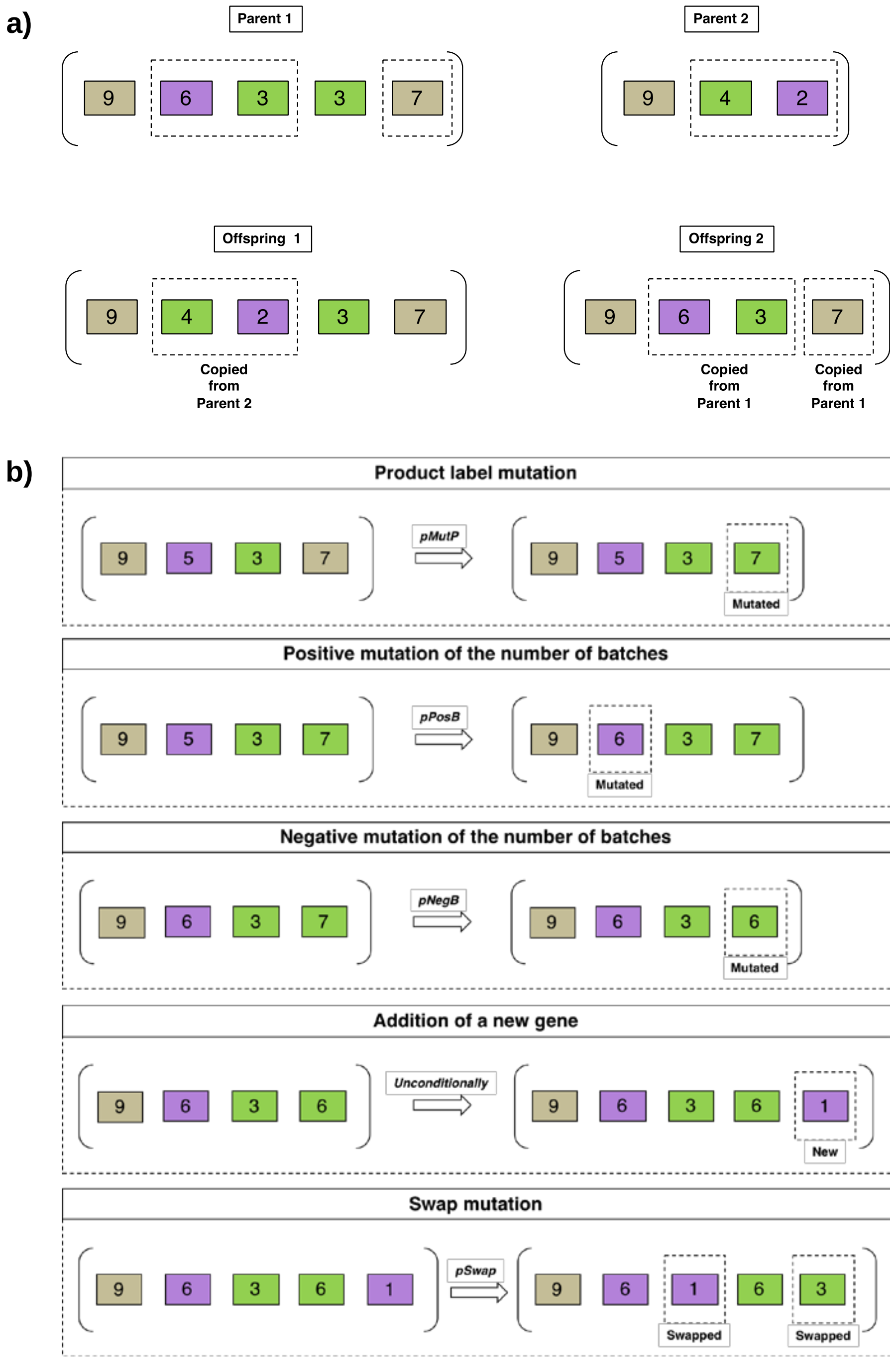
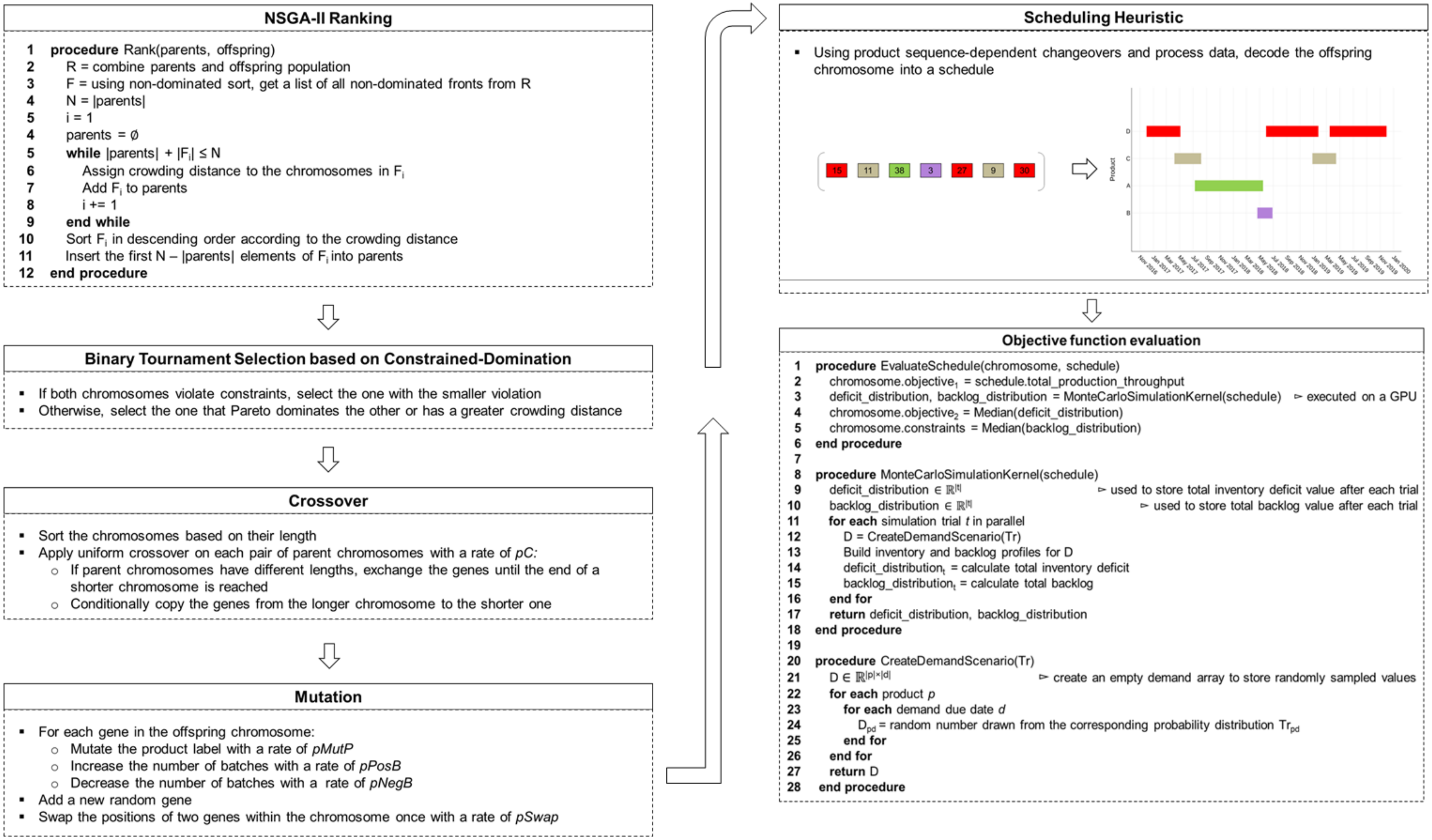


Figure 2. Crossover and mutation of variable-length chromosomes:
a) An example of a modified uniform crossover between two variable-length chromosomes: genes 2 and 3 are exchanged between the parent chromosomes and gene 5 from the first parent chromosome is copied to the second one.
b) Variable-length mutation steps. pMutP, pPosB, and pNegB denote the rate of each gene undergoing the corresponding mutation. The addition of a new gene and swap mutation occur once per chromosome.



NSGA-II Ranking

```
procedure Rank(parents, offspring)
    R = combine parents and offspring population
    F = using non-dominated sort, get a list of all non-dominated fronts from R
    N = |parents|
    i = 1
    parents = ∅
    while |parents| + $|F_i|$ ≤ N
        Assign crowding distance to the chromosomes in $F_i$
        Add $F_i$ to parents
        i += 1
    end while
    Sort $F_i$ in descending order according to the crowding distance
    Insert the first N – |parents| elements of $F_i$ into parents
end procedure
```

Binary Tournament Selection based on Constrained-Domination

- If both chromosomes violate constraints, select the one with the smaller violation
- Otherwise, select the one that Pareto dominates the other or has a greater crowding distance

Crossover

- Sort the chromosomes based on their length
- Apply uniform crossover on each pair of parent chromosomes with a rate of *pC:*
  - If parent chromosomes have different lengths, exchange the genes until the end of a shorter chromosome is reached
  - Conditionally copy the genes from the longer chromosome to the shorter one

Mutation

- For each gene in the offspring chromosome:
  - Mutate the product label with a rate of *pMutP*
  - Increase the number of batches with a rate of *pPosB*
  - Decrease the number of batches with a rate of *pNegB*
- Add a new random gene
- Swap the positions of two genes within the chromosome once with a rate of *pSwap*

Scheduling Heuristic

- Using product sequence-dependent changeovers and process data, decode the offspring chromosome into a schedule

Objective function evaluation

```
procedure EvaluateSchedule(chromosome, schedule)
    $chromosome.objective_1$ = schedule.total_production_throughput
    deficit_distribution, backlog_distribution = MonteCarloSimulationKernel(schedule)   ▻ executed on a GPU
    $chromosome.objective_2$ = Median(deficit_distribution)
    chromosome.constraints = Median(backlog_distribution)
end procedure

procedure MonteCarloSimulationKernel(schedule)
    deficit_distribution ∈ $\mathbb{R}^{|t|}$          ▻ used to store total inventory deficit value after each trial
    backlog_distribution ∈ $\mathbb{R}^{|t|}$          ▻ used to store total backlog value after each trial
    for each simulation trial t in parallel
        D = CreateDemandScenario(Tr)
        Build inventory and backlog profiles for D
        $deficit\_distribution_t$ = calculate total inventory deficit
        $backlog\_distribution_t$ = calculate total backlog
    end for
    return deficit_distribution, backlog_distribution
end procedure

procedure CreateDemandScenario(Tr)
    D ∈ $\mathbb{R}^{|p|\times|d|}$          ▻ create an empty demand array to store randomly sampled values
    for each product p
        for each demand due date d
            $D_{pd}$ = random number drawn from the corresponding probability distribution $Tr_{pd}$
        end for
    end for
    return D
end procedure
```

Figure 3. Schematic of the core steps of the stochastic multi-objective GA with integrated Monte Carlo simulation. Assuming the initial population has been created and evaluated, the steps are looped through until the maximum number of generations is reached.

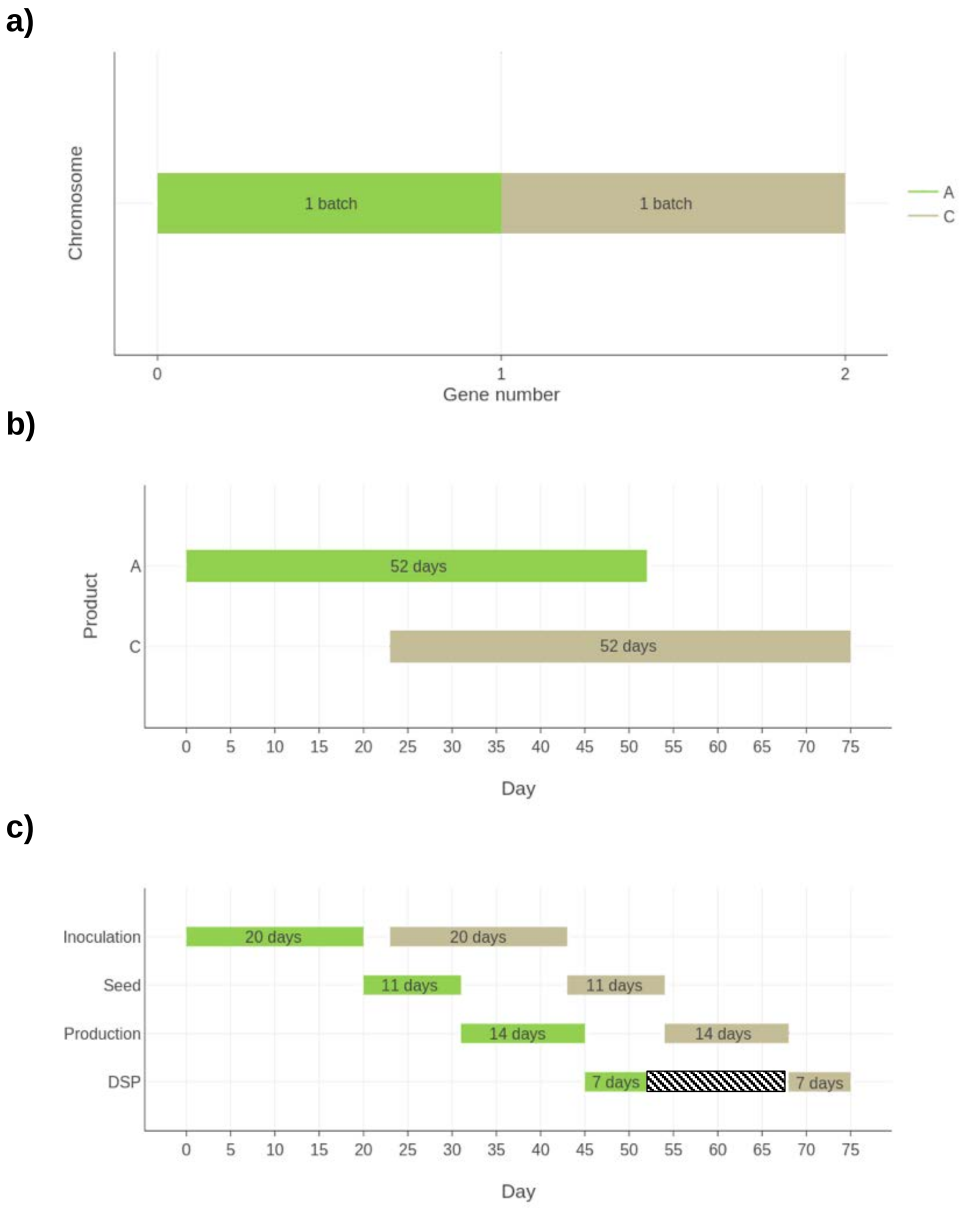


Figure 4. An example of the relationship between (a) the genes and (b) the decoded production schedule displayed at a product campaign level and (c) at a manufacturing stage level. The black and white striped box in (c) denotes product sequence-dependent changeover time used to determine the start of a new product campaign.

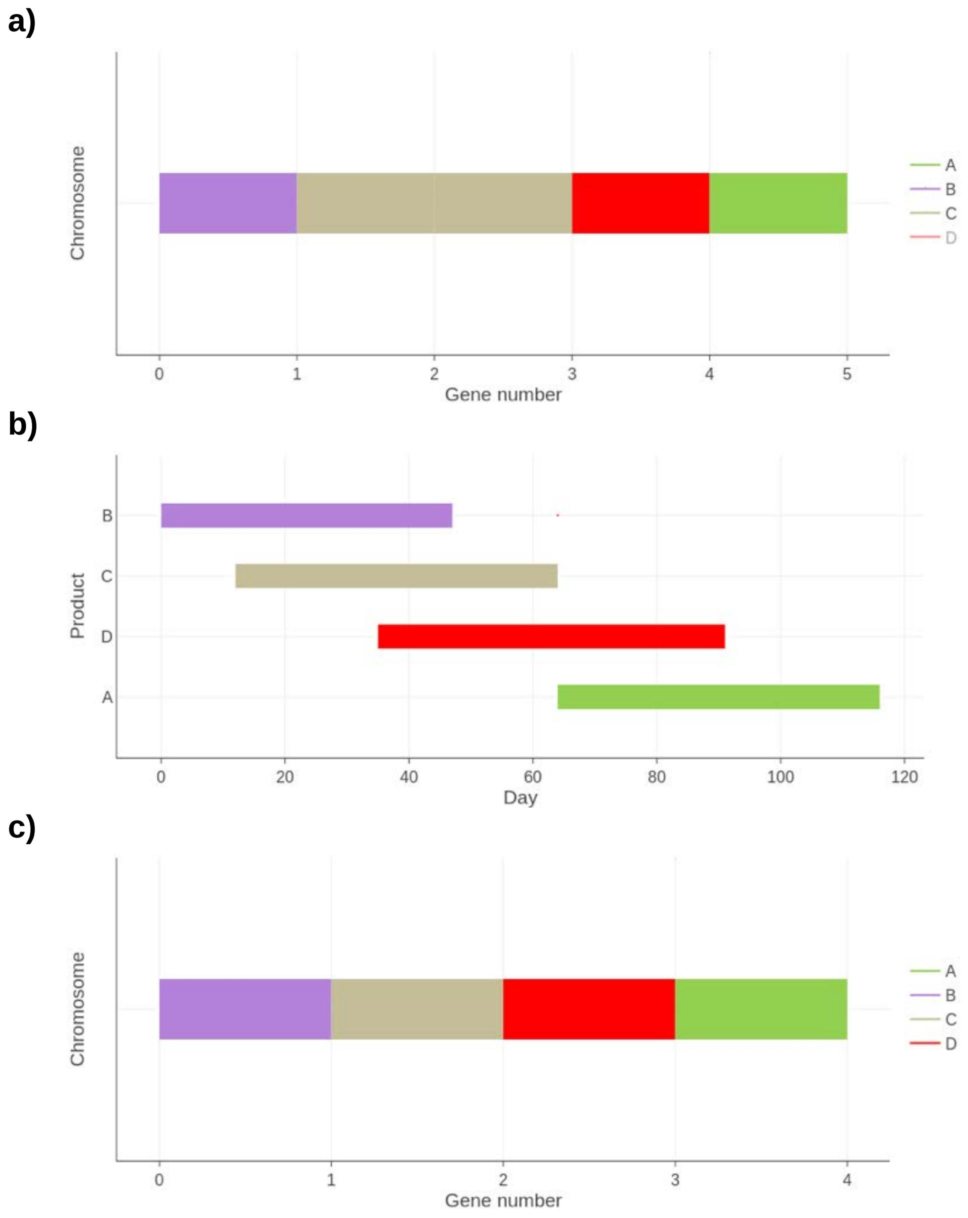


Figure 5. Correction of the mapping of genes to the production campaigns. In (a), the genes 2 and 3 correspond to the same product. The continuous-time scheduling heuristic combines them into (b) one contiguous manufacturing campaign and re-maps it to (c) a single a gene.

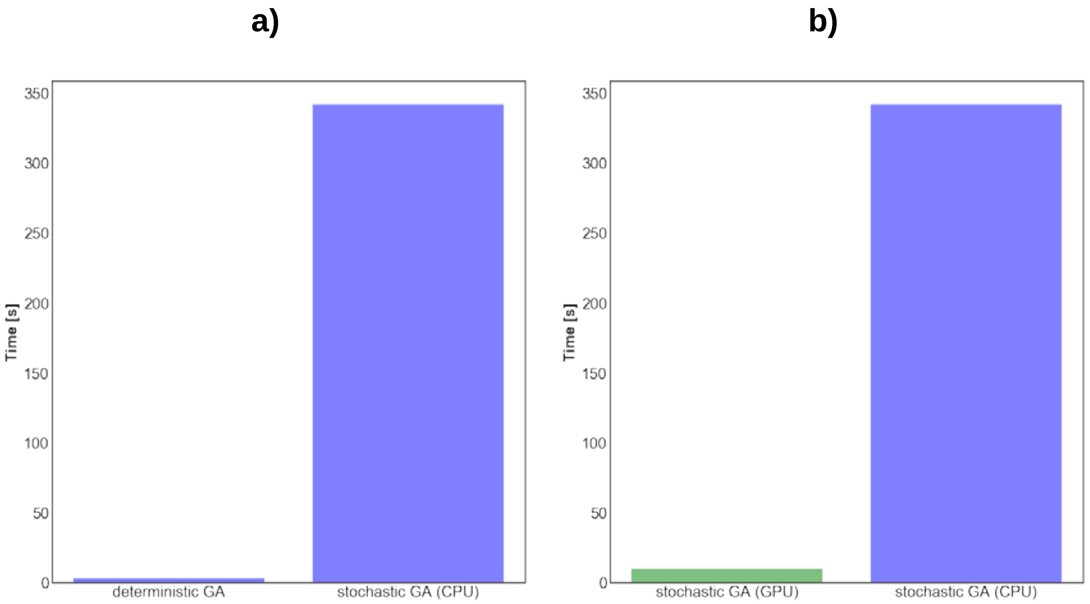


Figure 6. Average elapsed time for each of the 50 GA runs with 100 chromosomes for 1000 generations (the total runtime of 50 runs is divided by 50):

(a) Deterministic GA vs. CPU-only stochastic GA.

(b) Stochastic GA with Monte Carlo simulation performed on a GPU vs. CPU-only stochastic GA.

Note: fitness evaluations of both deterministic and CPU-only stochastic GAs were performed in parallel using the OpenMP API.

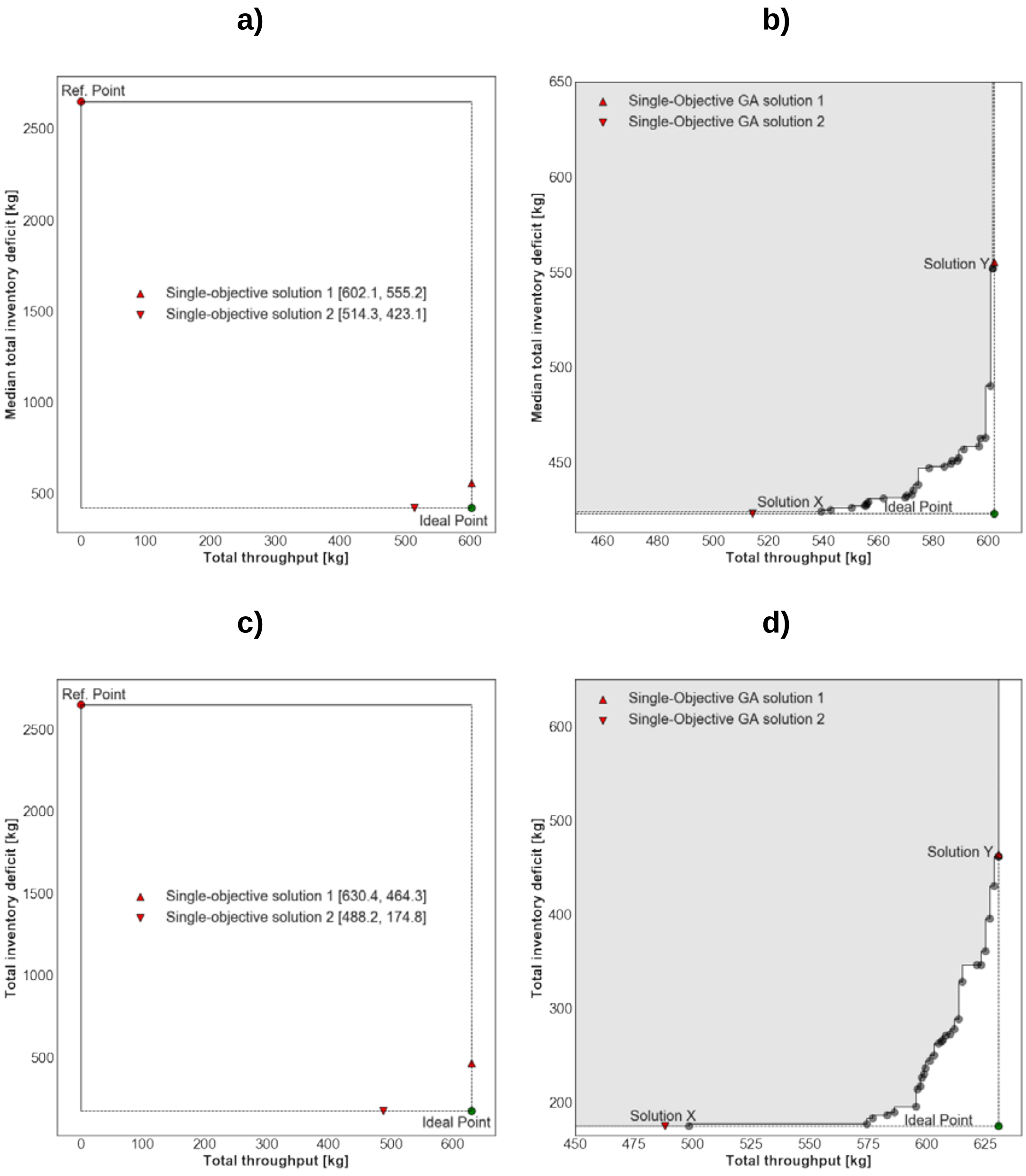


Figure 7.
(a) Stochastic objective space.
(b) The best Pareto front generated using the stochastic multi-objective GA (hypervolume of 0.997). The gray shaded area is used for illustrative purposes to show the area of the objective space that is dominated by the Pareto front solutions.
(c) Deterministic objective space.
(d) The best Pareto front generated using the deterministic multi-objective GA (hypervolume of 0.996).

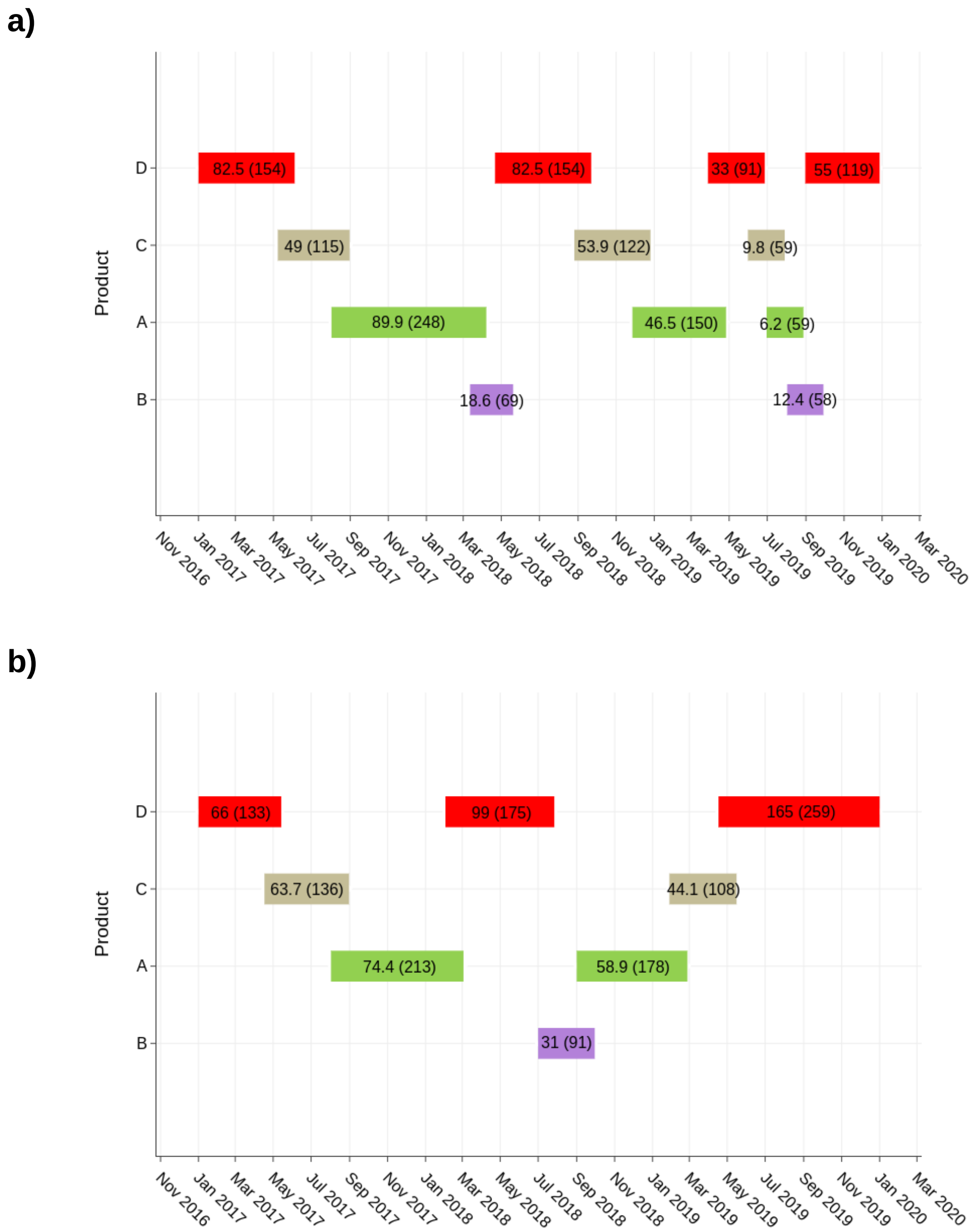


Figure 8. Production schedules of (a) solution X and (b) Y from the best Pareto front after 50 runs generated using the stochastic GA. The numbers in the boxes show how many kilograms are being manufactured, followed by the production time (days).

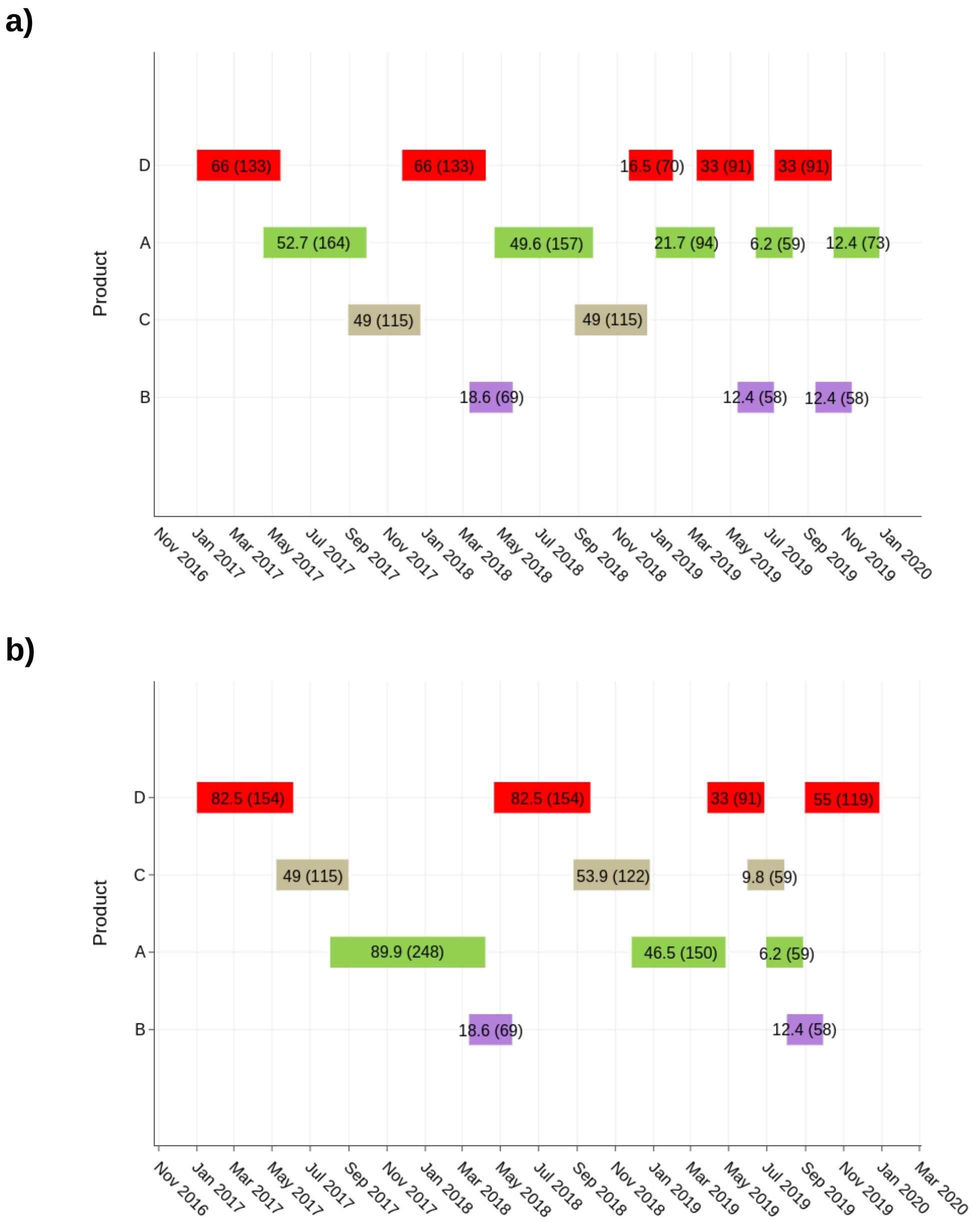


Figure 9. Production schedules of (a) the deterministic solution X and (b) stochastic solution X from the respective best Pareto fronts. The numbers in the boxes show how many kilograms are being manufactured, followed by the production time (days).

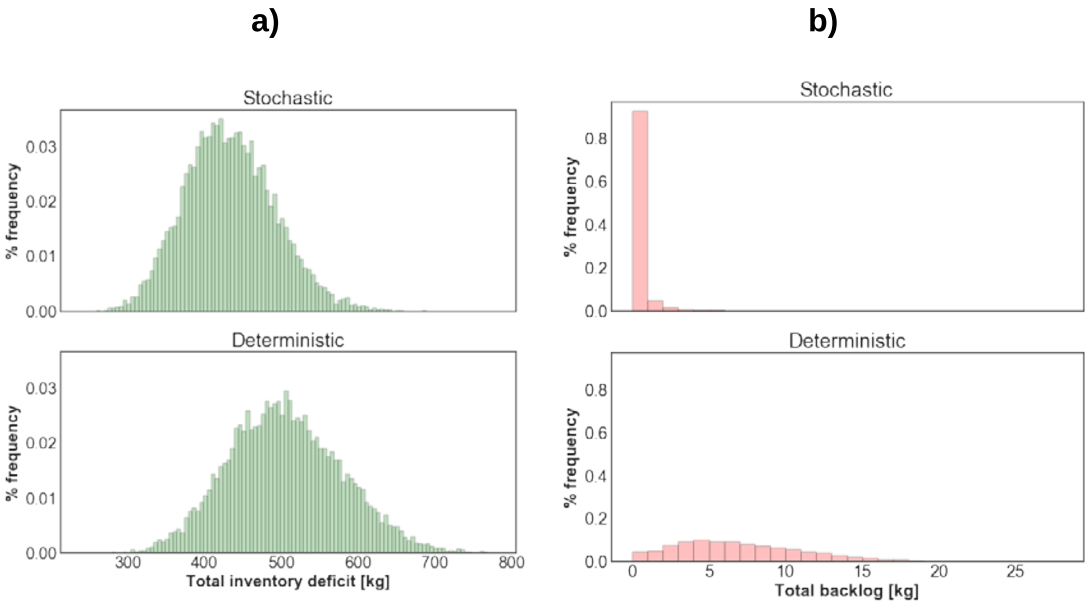


Figure 10. A comparison of:

(a) The total inventory deficit.

(b) Total backlog distributions between the stochastic and deterministic solutions. after the stochastic analysis with Monte Carlo simulation.

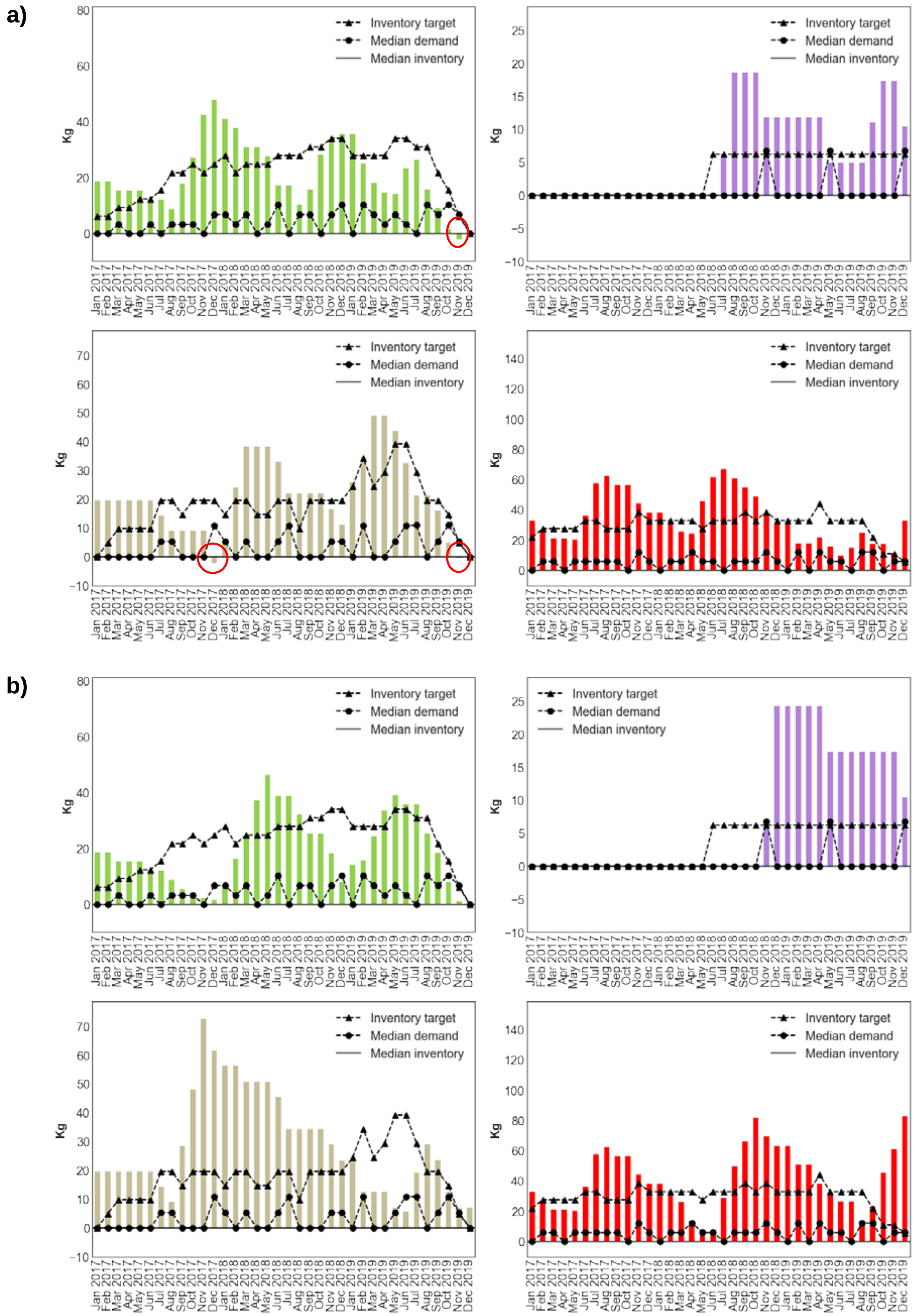


Figure 11. Individual product (🟩A 🟪B 🟫C 🟥D) inventory profiles of:
a) The deterministic solution after the stochastic analysis with Monte Carlo simulation.
b) The stochastic solution.