POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

The Optimization of a manufacturing line affected by the operator's interference issue

Tesi di Laurea Magistrale in
Mechanical Engineering - Ingegneria Meccanica

Author: **Gianluca Palombelli**

Student ID: 953043
Advisor: Prof. Tullio Antonio Maria Tolio
Co-advisors: Eng. Maria Chiara Magnanini
Academic Year: 2021-22

Abstract

The evolving industrial scenario affects the routine of manufacturing companies: internal and external dynamic mutations affect their key production parameters, that undergo fast and sudden modifications. The on-time delivery of the products, together with a strong emphasis on their final quality, are decisive factors for customers' approval and for competitiveness on the market. Companies look for the highest production efficiency in order to operate following a sustainable resource consumption. It is evident the importance of taking right and fast decisions: that's the reason why many optimization algorithms have been conceived in the last years.

There is a clear necessity of a method that, whenever decisions have to be taken, automatically and rapidly finds the best solution. Manufacturing companies face a relevant risk when taking strategic decisions about system resources to be acquired. These risks can be mitigated thanks to performance evaluation models. In literature, it can be found an analytical method supporting the optimization of manufacturing systems configuration: a stochastic analytical model for performance evaluation is integrated into a mixed integer programming problem to provide robust solutions, by means of performance linearization.

The objective of this thesis is to focus on a specific issue - the achievement of the target throughput - and create an optimization algorithm to satisfy this requirement. It has to be integrated a performance evaluation model to analyse the behaviour of a line. In addition, a particular constraint is set: since the line is connected to a Digital Twins system, characteristic of the Industry 4.0 scenario, a frequent situation is to have a single operator dedicate to the whole line management. Having several tasks to accomplish, it frequently arises a worker's interference issue.
A method to solve this problem is conceived and then applied to the real-life industrial case, regarding a company producing train axles.

Keywords: Manufacturing system, Performance evaluation, Digital Twin, Evolution planning, Optimization, Production planning and control

Abstract in lingua italiana

Lo scenario industriale in continua evoluzione impatta sulla routine delle aziende manifatturiere: mutazioni dinamiche interne ed esterne incidono sui parametri di produzione, che subiscono modifiche rapide e improvvise. La puntualità nella consegna dei prodotti, unita ad una forte enfasi sulla loro qualità finale, sono fattori determinanti per la soddisfazione dei clienti e per la competitività sul mercato. Le aziende ricercano la massima efficienza produttiva per operare seguendo un consumo sostenibile delle risorse. È evidente l'importanza di prendere decisioni giuste e veloci: ecco perché negli ultimi anni sono stati ideati molti algoritmi di ottimizzazione.

Vi è una chiara necessità di un metodo che, ogni volta che devono essere prese delle decisioni, trovi automaticamente e rapidamente la soluzione migliore. Le aziende manifatturiere affrontano un rischio rilevante quando prendono decisioni strategiche sulle risorse di sistema da acquisire. Questi rischi possono essere mitigati grazie a modelli di valutazione delle prestazioni. In letteratura si trova un metodo analitico che supporta l'ottimizzazione della configurazione dei sistemi di produzione: un modello analitico stocastico per la valutazione delle prestazioni viene integrato in un problema di programmazione mista integrale per fornire soluzioni robuste, attraverso la linearizzazione delle prestazioni.

L'obiettivo di questa tesi è di concentrarsi su una questione specifica - il raggiungimento di una specifica domanda di prodotto - e di creare un algoritmo di ottimizzazione per soddisfare questa esigenza. Deve essere integrato un modello di valutazione delle prestazioni per analizzare il comportamento della linea. Inoltre, si pone un vincolo particolare: poiché la linea è collegata a un sistema Digital Twins, caratteristico dello scenario Industry 4.0, una situazione frequente è quella di avere un unico operatore dedicato alla gestione dell'intera linea. Avendo diversi compiti da svolgere, si verifica spesso un problema di interferenza del lavoratore.
Un metodo per risolvere questo problema viene ideato e poi applicato al caso industriale reale, riguardante un'azienda che produce assili per treni.

Parole chiave: Sistema Manifatturiero, Metodo Valutazione Prestazioni, Digital Twin, Piano di Sviuppo, Ottimizzazione, Piano e Controllo della Produzione

Contents

Abstract **i**

Abstract in lingua italiana **iii**

Contents **v**

Introduction **1**

1 Industrial Case-Study **7**
1.1 The problem 7
1.2 Hypotheses 9

2 The Method **15**
2.1 Objectives and Model Formalization 15
2.2 The optimization algorithm 17

3 Results **31**
3.1 Validation 31
3.1.1 Model vs Simulink: Results comparison 33
3.1.2 Exhaustive Research 35
3.2 Sensitivity Analysis 36

4 Conclusions **49**

Bibliography **51**

List of Figures **53**

List of Tables **55**

Introduction

Nowadays, the continuously changing and evolving industrial scenario affects the day-by-day routine of manufacturing companies. Dynamic mutations are brought not only by the external environment, but also by the internal one: the demand may face weekly variations and innovations in the automation and technological field are not so sporadic; at the same time, the working conditions of the companies and the key parameters affecting their production undergo fast and sudden modifications.

As a matter of fact, the degradation of a machine leads to an increase in its failure rate; moreover, operators may develop their abilities thanks to upskilling or to a growth in experience: thus, the repair time of the machines would decrease [1]. Furthermore, great attention has to be put on the on-time delivery of the products requested by the customers, together with a strong emphasis on their final quality. Companies look for the highest production efficiency in order to operate following a sustainable resource consumption. That's the reason why many optimization algorithms have been conceived in the last years.

Knowing that, it is evident the importance of taking right and fast decisions. There are two main decision areas: tactical decisions and strategical decisions. On the one hand, the system includes operational decisions, as the workforce allocation, the machines availability and the production planning strategy [1]. In this case, it is easy to adapt the manufacturing system to frequent changes in operating conditions. On the other hand, strategical decisions have a great impact on the system configuration, since they affect the management of resources: the selection of machines and their eventual substitutions, variations in the buffer capacity, new workforce recruitment and training.

Each time the company opts for a strategical optimization there is not only a significant increase in the operating costs, but also a delay in the time schedule. That's why changing configuration is less frequent, while tactical decisions belonging to different production areas are continuously taken and optimized according to evolving operating conditions [1].

In the field of manufacturing systems production managers, at the beginning, always try to opt for tactical decisions first: if they result to be useless in optimizing the configuration, managers are forced to consider the strategical ones. Companies dealing with multi-product production base the planning of the activities on aspects like the product-type prioritization, the scrap rate, the Buffer Allocation Problems, the workers' level of skill, which influences the repair time of the machines and their set-up times [2]. However, when facing random and unplanned events or variations of the customers' requests, they may have to face a revolution regarding the previous ranking of the products: this means that the actual operating plan, which was the best alternative in the previous situation, characterized by different boundary conditions, would result to be useless, or rather, counter-productive. A new optimized production plan needs to be quickly conceived and implemented.

There is a clear necessity of a method that, whenever tactical and strategical decisions have to be taken due to a sudden mutation in the external environment or an internal modification of the working condition, automatically and rapidly finds the best configuration and organization solution.

State of the art

In order to identify that method, available researches concerning the problems of design and operation of production systems must be addressed. As previously underlined, the competitive and unpredictable industrial scenario forces the manufacturing companies to continuously upgrade their working areas and take right and fast decisions.

Papadopoulos et al. [3] suggest the application of both evaluative and optimization algorithms in order to achieve this goal. The first ones are employed to get data concerning the line performances while the second ones to achieve the optimal configurations. Gershwin [4] looks at Manufacturing Systems Engineering as the design and operation of factories. The design of a manufacturing system consists of three steps:

- the choice of the system architecture
- the selection of the machines
- the design of the spaces to hold work-in-process inventory, known as buffers allocation

The complexity of this process depends on the actual status of the plant: if a line already exists and it must be modified only a section of it, the effort will be much lower than the case considering a green area to start.

Manufacturing systems have to be robust to face frequent product changes in the production scenario, required by mass customization. Each time the company opts for a strategic decision a risk rate must be considered. Potential negative consequences can be lightened by addressing performance evaluation models, e.g. analytical models and Discrete Event Simulation, capable of figuring out the performance of possible system configurations. However, the methods employed for optimization must be strictly linked to performance evaluation models, in order to guarantee a good optimization of the system.

Magnanini et al. [5] present an analytical methodology to support the optimization of manufacturing systems configuration and reconfiguration subject to evolving production requirements. A stochastic analytical model for performance evaluation of manufacturing lines is integrated into a mixed integer programming problem, by means of performance linearization. The proposed method is an iterative algorithm that integrates a stochastic analytical model for performance evaluation of serial manufacturing lines into a linear programming problem by means of performance linearization [6].The method is applied to an industrial case of a line configuration focused on the optimization of the machines selection and of the buffer allocation.

The optimization must satisfy the objective function established by the company: in many cases it is a cost minimization and the throughput of the line is always under analysis. All the system performance measures do not vary linearly with respect to the system parameters. The issue related to traditional optimization methods, grounding on linear programming, is that the performance evaluation is treated as linear with respect to the changing parameters thus undermining the results, that are approximated, and the trustworthiness of the optimal configuration got. In order to provide rigorous performance evaluation methods, the optimization model must include the evaluation of the configuration, following an iterative approach.

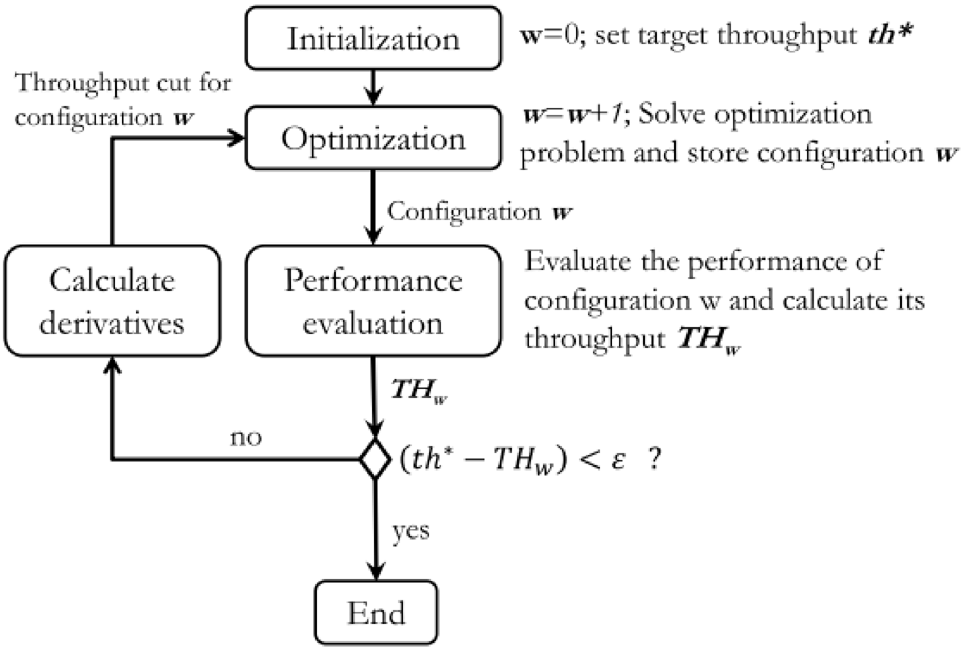


Figure 1: Graphical representation of the proposed algorithm. Picture taken from [5]

First of all, the target throughput (th*) must be set and initialized. Through an iterative cycle, the configuration of the line is optimized: an approximation grounding on linear constraints allows an estimation of the throughput. The candidate optimal configuration, got through approximation, has to be checked in order to be validated: its data are given as input to the performance evaluation model and the real throughput of the line is calculated. The algorithm proceeds iteratively until the real throughput and the target throughput differ less than an arbitrary margin, defined a priori. If the range of tolerance isn't respected, the performance evaluation model evaluates and uses the first-order derivatives to add the throughput cut constraint, needed to linearize the performance, to the optimization problem.

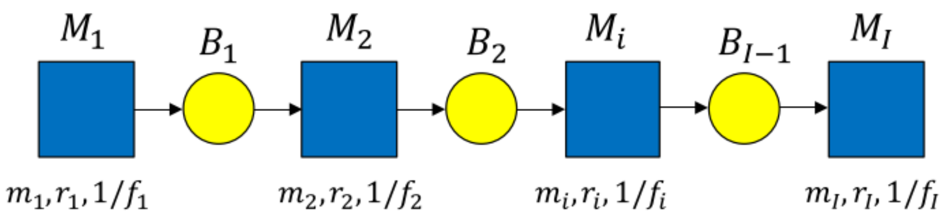


Figure 2: The manufacturing serial line. Picture taken from [5]

Only through analytical model the explicit relation between input parameters and output performance is obtained; decomposition equations model the propagation of blocking and starvation phenomena along the line. A linear system of differential equations is solved by a numerical algorithm in order to evaluate the system performance [5]. The first derivatives of the system throughput can be derived and used to write the first-order approximation of the throughput with respect to the system parameters.

Thus, linear approximation increases the accuracy for each tangent line added to the set:

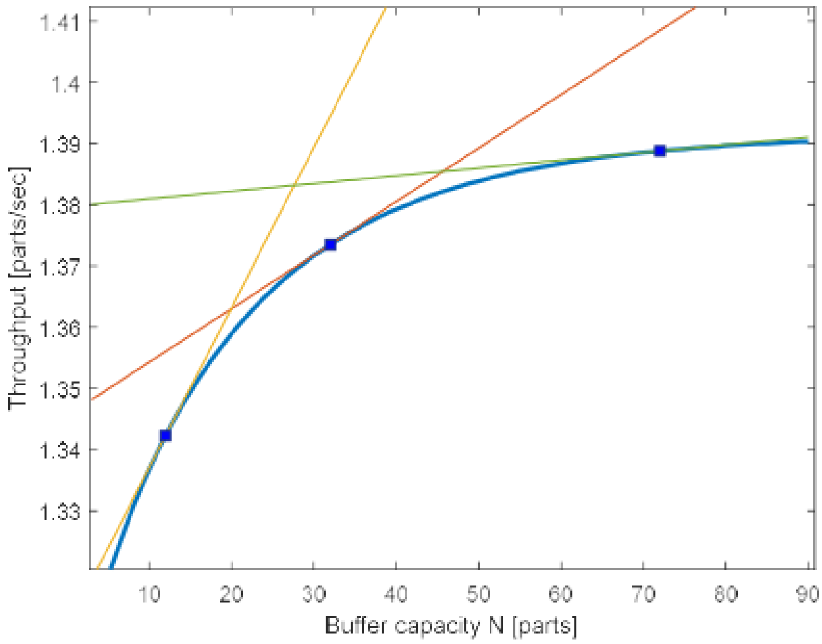


Figure 3: Throughput as a function of buffer capacity with first-order linearization. Picture taken from [5]

The optimization model, instead, is formulated as a Mixed-Integer Linear Problem (MILP). Most of the times, the goal is to minimize the overall resources cost, while satisfying the target throughput. An upper bound for the throughput can be calculated thanks to a preliminary analysis of the problem. The throughput cut constraint is added after each iteration based on the linearization of the throughput.

1 | Industrial Case-Study

1.1. The problem

The proposed methodology is applied to a specific industrial company in the railway sector, dedicated to the production of train axles.

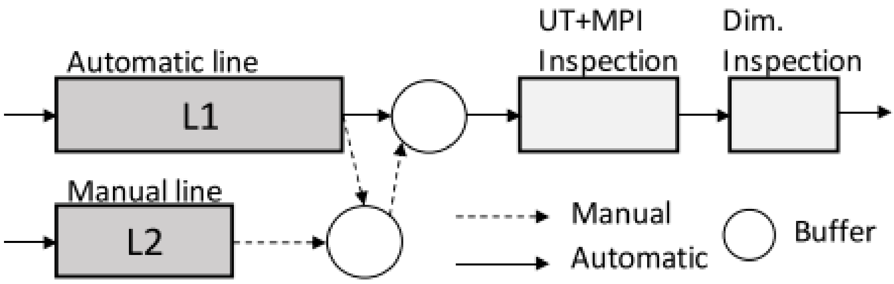


Figure 1.1: The Machining Department. Picture taken from [1]

The manufacturing system consists of several departments: the focus of this work is on the machining area. This section is composed by two parallel manufacturing lines. These two lines are conceived in different ways. The first one (L1) is an automatic production line: a gantry system is in charge of transferring the axles from the first station to the others and no manual intervention is needed [1]. The line is built up by three machines, respectively performing rough turning, finish turning and stone grinding [1]. Between each couple of stations there are buffers of finite capacity. At the end of the line there is the inspection machine, which is directly linked to L1: a small buffer, having only two slots, receives the completed parts from the stone grinding machine [1].

The second line (L2) shows the same machines of L1 but the transportation system is done through hoist cranes requiring manual intervention [1]. This implies that large buffers can be set, since parts are gathered on the floor in between the machines [1]. Having these characteristics, the line can be thought as something similar to a Job shop.

Even if the target weekly throughput is not so demanding (from 160 to 200 axles), having only two slots for the pre-inspection machines buffer can be an issue. The blocking probability of the stone grinding machine can not be neglected: the company is evaluating the possibility in the future to enlarge the buffer capacity, but, for the moment, the idea is to employ the operators in the manual unloading of the buffer, when it is full [1]. Parts are stored on the floor and then reloaded as soon as possible. The reloading is completed only when the line stops due to a failure of a machine or to a setup [1].

The workforce employed on L1 is in charge also of repairing failed or degrading machines, making inspections on the critical ones, to enhance preventive maintenance, and properly performing set-ups when there is a change in the lot to produce. Operators in L2 are not engaged in the manual management of the final buffer but have to provide the material handling till the inspection machine [1].

Once defined the main characteristics of the manufacturing system in analysis, the focus is now on the study of Digital Twin integrated to it. The continuous and live updated flow of information coming from the machining department is made possible through an Industrial Internet of Things sensor network set along the whole lines: data regarding the overall status of the production system and its resources, as a blocking or starvation of a station, a degrading machine, the actual level of the buffers, updated MTTRs, are constantly monitored [1].

Having the possibility to instantly know the evolution of the system, every day the production manager of the machining department faces situations where he has to opt for a technical decision or, less frequently, for a strategical one: indeed, since he has to attain the constraint on the throughput to reach, a change in the production parameters and policies is needed, with the aim of improving the line.

Several alterations can be applied to the line in order to get a specific goal as, for example [1]:

- Since the axles coming from the lines are all stored in the small buffer located pre-inspection machine, at occurrence it is possible to prioritize one stream instead of the other one;
- The workforce employed in Line1 is in charge also of the unloading operations on the final buffer: anytime the line requires manual intervention, i.e. when a machine fails or is degrading, when a set-up is needed or when an inspection machine is planned, it can be decisive to have a prioritization policy to order the sequence of

actions to apply;

- Having the operators several tasks to be accomplished, it is better to address them where their skills are more emphasized;
- If an Energy-saving policy is established, the top management has to deal with strategies to decrease the overall scrap rate, in order to reduce the amount of parts sent to the re-work line;
- Training courses for the upskilling of the operators should be considered if new machines are hard to be managed.

As the throughput can very weekly, also the lots can have different sizes: this implies that the weekly number of lots often changes. Varying the number of lots, the directly influenced parameter is the number of set-ups that has to be done. The more the set-ups are, the less is the availability of the line and the more the operators have to spend time to accomplish this task.

It is now evident the great innovation represented by Digital Twin Networks: tactical and strategical decisions can be immediately taken thanks to the tangent hyperplanes algorithm, thus reducing the waste of time and possible wrong assessments on production planning. The implementation of this new methodology paves the way for new interfaces between fields that once where treated independently [1]. As a matter of fact, improvements in a specific area (e.g maintenance) may lead negative effects to other variables (quality ad logistics).

1.2. Hypotheses

The manufacturing system in analysis, if considering the resources involved and their interconnections, can be described following the Markov Chain representation. Indeed, it consists of service machines separated by buffers: the parts to be processed go through the machines and buffers according to a predefined flow direction.

'The typical Multi-stage Manufacturing System (MMS) is characterized by a series of assumptions [7]:

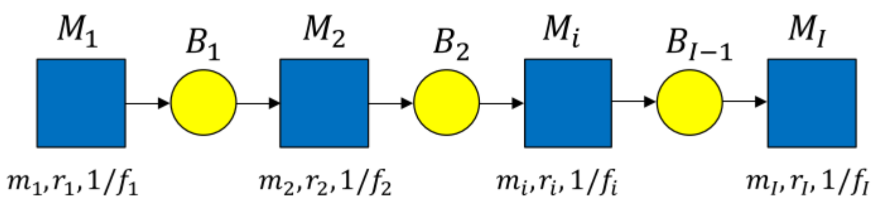


Figure 1.2: Graphical representation of a multi-stage manufacturing system. Picture taken from [8]

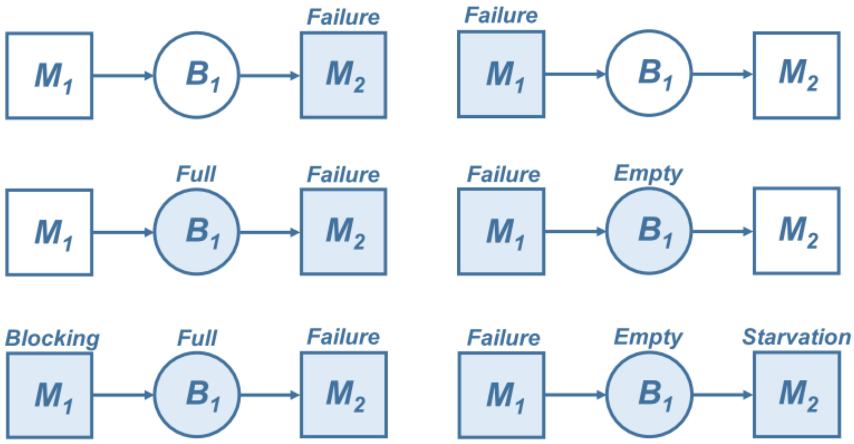
- The first machine is never starved and the last machine is never blocked;
- Processing times of the machines are deterministic and may be different among machines;
- Machines are unreliable and may fail in different modes, with operating dependent failures;
- Time to failure and time to repair have a general distribution;
- Load and unload times are negligible;
- Parts arrive from outside and leave the system after being processed;
- The buffer capacity is finite;
- The system is asynchronous, i.e., each machine can start or finish at any time without synchronization with the other machines;
- The dispatching policy is Fist In First Out (FIFO);
- Parts are not scrapped or reworked.

For the two-machine line an exact solution exists: referred to as "Building Block", it is the basis for the analysis of long lines or more complex systems.

Figure 1.3: The Building Block. Picture taken from [8]

A machine can be operational ('up' state) or non operational / failed ('down' state). Transition rates define timings to switch from one state to another: the mean time to failure and to repair (MTTF and MTTR) are assumed to be exponentially distributed. The machine is described by a state-based representation: in operational states the machine has a production rate greater than 0, whereas in non-operational states its production rate is 0.

When the upstream machine is "up" and the downstream "down", once the buffer is full the first one can't process parts and becomes idle. The upstream machine is "blocked" by the downstream machine. Similarly, when the downstream machine is "up" and the upstream "down", once the buffer is empty the first one can't operate and becomes idle. In this case, the downstream machine is "starved" by the upstream machine. The system remains in this states until the machines are repaired.



(a) *Blocking phenomenon* (b) *Starvation phenomenon*

Figure 1.4: Blocking and Starvation phenomena. Picture taken from [8]

Thus, it can be affirmed that the behaviour of each machine is affected by the other ones.

Having more than two machines the line can't be modelled as seen before since it becomes a "long" line and exact analytical solutions do not exist for long lines. There is the need of approximate analytical methods to calculate the performance: the Decomposition Method is perfectly suited. This method decomposes the line in subsystems composed by two machines and one buffer.

'The main idea of the Decomposition Method is to use the exact analytical results of two-machine lines and to obtain the performance of a long line considering the impact of each machine on the others and by imposing the conservation of flow between each Building Block' [8]. The line can be split in several Building Blocks made up by a buffer having the same size of the original one, an upstream pseudo machine, representing the part of the line upstream the buffer, and a downstream pseudo machine, representing the part of the line downstream the buffer.

Line configuration and operator's interference

This thesis focuses on the optimal configuration of the manufacturing line in analysis, starting from green field. The production system is made up by four stages having a rough turning, a finish turning, a stone grinding and, at the end, an inspection machine. The company doesn't set upper limits on the buffer sizes, but a constraint on the final buffer: its capacity is fixed at 2 slots.

The industrial scenario sees a wide spread of the Industry 4.0. New technologies have been conceived and up-to-date production facilities as the Industrial Internet of Things, the machine learning and the Artificial Intelligence are available. These innovations change the typical production management, since the human role is being put aside: in the future, the amount of workers employed by the company is going to decrease. The high level of automation and the use of optimization algorithms decrease the necessity of many operators managing the line.

Thus, a strong hypothesis can be set: only one operator is in charge of managing the whole line and must fulfill all the tasks. Indeed, in the Industry 4.0 context, it frequently happens to have a single worker managing a whole line. As a result, an issue related to the operator's interference arises. As stated by [1], about the production line in analysis: "Given the space in the automated buffer between L1 and inspection (buffer capacity = 2), lines may occur in blocking. In the future the size of the automatic buffer will be increased but for the moment the policy is that operators create extra buffer space by unloading parts to avoid blocking of the lines and reloading them when the lines stop due to failures or setups."

Therefore, the duties of the workforce for the system under investigation consist in "maintenance activities on the line, the set-up of the stations for product changes, and the manual unloading and reloading of the final buffer to manage the flow of parts" [1]. So, for what concerns this industrial case study, it is clear that the management of workers is a demanding task. Having many tasks to accomplish, the operator can't satisfy them simultaneously. The operator is seen as a server and the tasks as his clients: an algorithm for the choice of the proper maintenance policy must be formalized and solved, in order to get the service time related to the different requests.

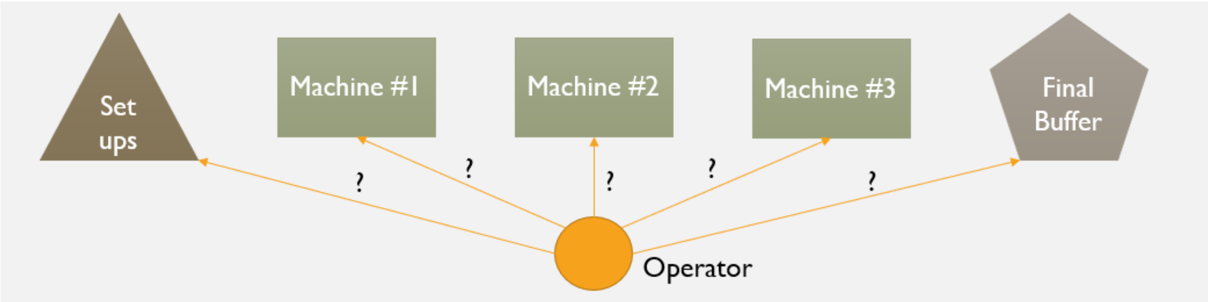


Figure 1.5: Evaluation of the Situation "As is"

The company must opt for a prioritization decision among maintenance, set-ups, and unloading operations in L1. By hypothesis, set-ups and operations on the final buffer come first, whereas maintenance activities on machines follow. Moreover, maintenance activities themselves must be ranked. It must be selected the best ranking policy for the machines, in order to make the worker serve first the most critical ones. Being the machines 3, since the inspection machine is assumed as perfectly reliable, the 6 ranking possibilities in terms of combination must be evaluated to select the best one, for a specific scenario.

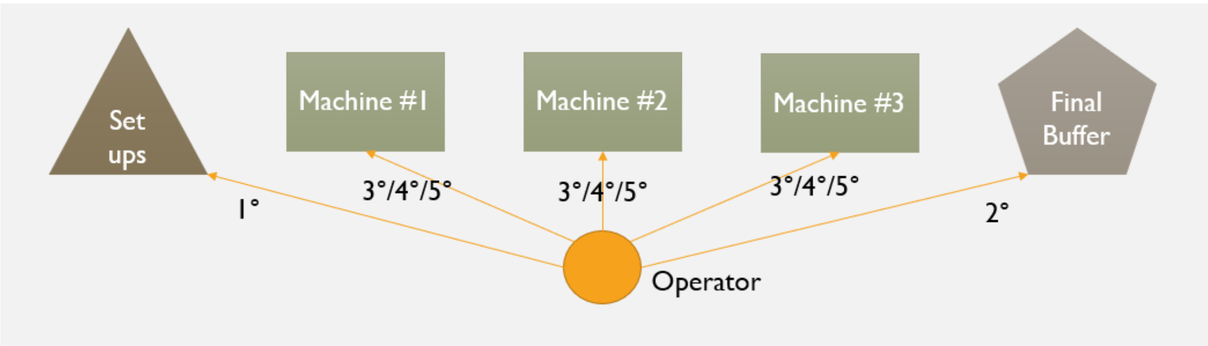


Figure 1.6: Evaluation of the Situation "To be"

As a result, the duty of this thesis is to propose an optimization method able to provide the best configuration for the line taking into account the negative consequences given by the operator's interference.

2 | The Method

2.1. Objectives and Model Formalization

The objective of this thesis is the joint optimization in terms of configuration and control policies of the manufacturing line in analysis. The optimization of the configuration affects the design of the manufacturing system: it is requested the best line arrangement, the proper selection of machines and the best buffer allocations. The aim of the company is to satisfy the weekly throughput demand of train axles, minimizing the related production costs.

Having a single worker managing the line, an operator's interference issue arises. The company must select the control policy that mitigates at the best the negative consequences of the operator's interference on the line performances, since the MTTRs of the machines inevitably increase. The control policy consists in a prioritization decision among the tasks the operator must fulfill, since he can't accomplish more than one of them simultaneously. Given these considerations, an optimization algorithm must be formalized to satisfy the weekly throughput demand for the line applying the control policy that minimizes at the best its related production costs.

For what concerns the configuration, the company must decide now the machine to install in the line, while buffer slots can vary in the future, when the demand will be known without uncertainty. Since two different decision stages are considered, a 2-Stage Stochastic Programming Model must be formalized. For the 2-Stage Stochastic Programming Model Resolution it has been selected the Here&Now approach: this means that the objective function must contain information on future possible demand rates, considering the associated realization probabilities.

The First-Stage decision variable is the choice of the machine to implement, while the Second-Stage decision variable is the set of buffer slots to allocate in the future and the policies to follow. At Second-Stage, the possible future scenarios and their related realization probabilities are selected combining:

- The different demand rates foreseen by the company (from 160 to 200 parts/week)
- The average lot sizes (10 or 60 pieces per lot)

How to optimize

For what concerns the control policy, instead, it must be treated the issue related to the operator's interference, that affects the behaviour of the line. The waiting time of the tasks for the operator's availability increases the MTTRs of the machines and this worsens the performances of the line: optimization is needed to mitigate these negative consequences. The proposed method must select, for each scenario, the policy that makes the line worsen the less: when multiple failures of the machines occur, the policy selects, for each scenario, the machines being prioritized with respect to the others.

The operator has to go repairing first the prioritized ones and, only once finished, provide the maintenance to the less critic machines, that can wait more for its arrival. Once compared the different resulting behaviour of the line considering the possible ranking policies of the machines (ABC/ACB/BAC/BCA/...), the algorithm chooses the one that minimizes the costs associated to the line, needed to attain a specific throughput demand.

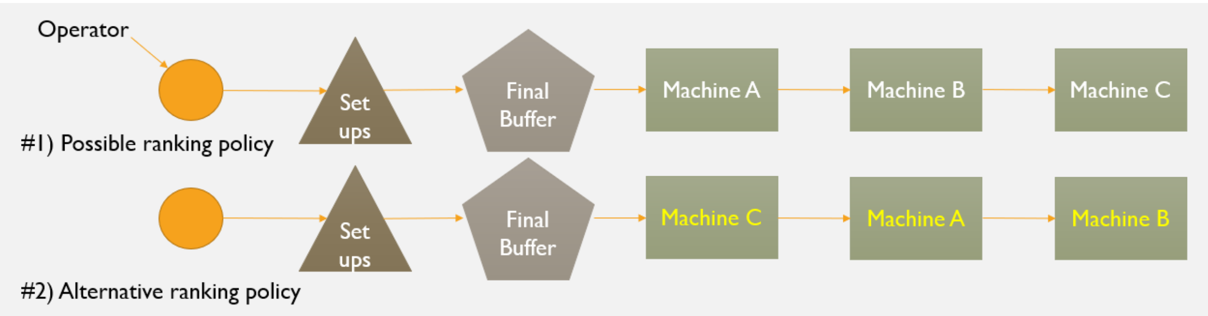


Figure 2.1: Examples of ranking policies

Moreover, two further optimization options are possible:

1. An additional operator can be hired: The waiting time for the availability of one of the two operators decreases, thus reducing the MTTRs of the machines
2. Flexibility on the control point policy: Buffer slots can be added to the line in order to improve the production performances

For each scenario, the algorithm must evaluate whether to optimize:

1. Just selecting the best ranking policy for the machines
2. Selecting the best ranking policy once added buffer slots
3. Selecting the best ranking policy once added an operator to the line and, if needed, buffer slots

The algorithm must work iteratively, looking for the best configuration for each considered scenario.

2.2. The optimization algorithm

The proposed method grounds its basis on the algorithm conceived by Tolio et al. [5]. It is an optimization algorithm designed to take into account the performance of a manufacturing system. As seen before, a stochastic analytical model, addressed for performance evaluation, is integrated to a Stochastic Programming Model, addressed for optimization. In particular, a linear programming model is developed grounding on the assumption that the system response is captured by linear constraints. The method works focusing on the iterative search for the optimal solution.

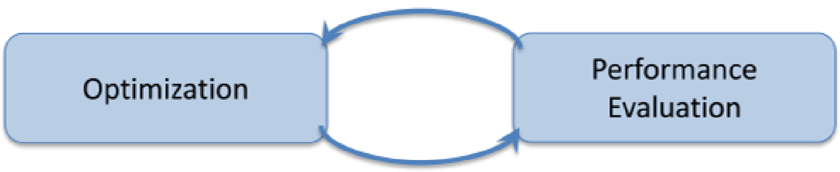


Figure 2.2: Interchanges between optimization and performance evaluation

A description in broad terms of the method follows. The Stochastic Programming Problem is initialized: inputs, decision variables and constraints are defined. Moreover, the objective function is declared: for the case in analysis, the target throughput (th*) is set. Then, the iterative loop can begin, taking into account a first attempt configuration. The approximation based on linear constraints allows the optimization of the configuration: the candidate configuration is given as input to the performance evaluation model for an accurate estimation of the throughput.

The algorithm proceeds iteratively until convergence, that is verified when the throughput estimated by the performance evaluation differs with respect to the target throughput less than a margin, defined a priori. If convergence is not reached, the first-order derivatives are extracted thanks to the performance evaluation and used to generate an additional constraint for the optimization problem, the throughput cut, that is essential to linearize the performance.

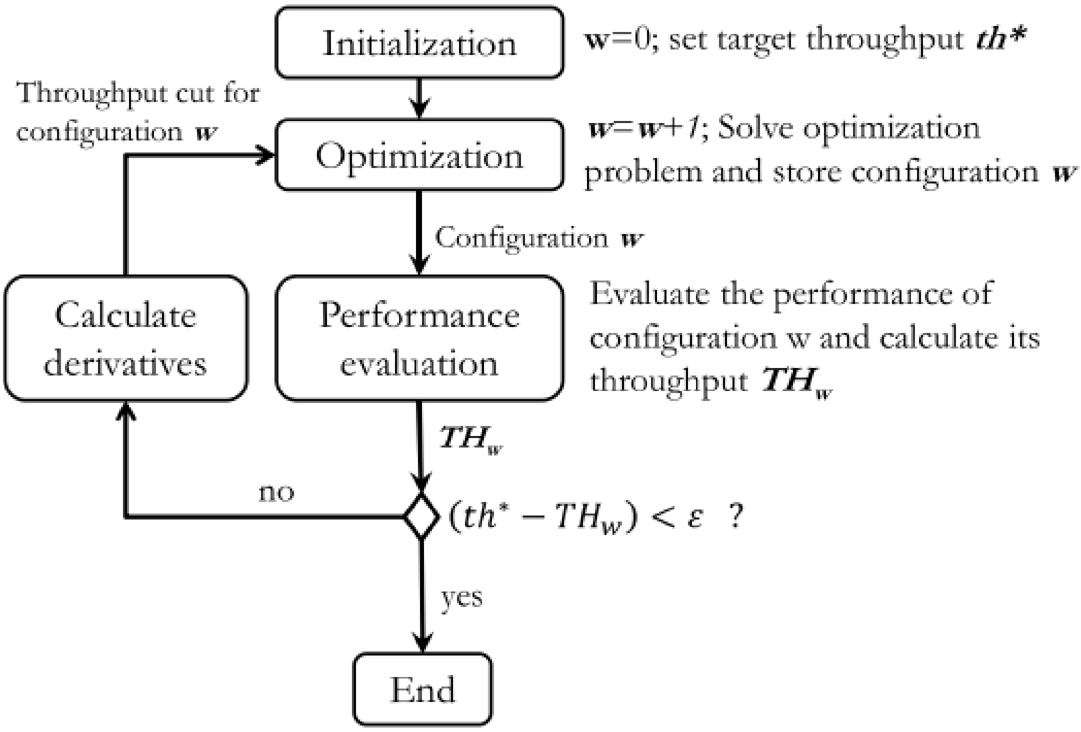


Figure 2.3: Graphical representation of the original algorithm. Picture taken from [5]

The original algorithm is an optimal tool when dealing with the configuration or re-configuration of a production line. The new algorithm proposed in this thesis instead copes with an additional issue to solve while configuring the line: the operator's interference. As a consequence, the iterative cycle must contain a new section dedicated to the evaluation of the behaviour of the line considering all the possible ranking policies of the machines (ABC/ACB/BAC...). Indeed, the MTTRs of the machines change depending on the considered policy.

The optimization selects the one that best mitigates the negative consequences of the operator's interference on the considered configuration, for each cycle.

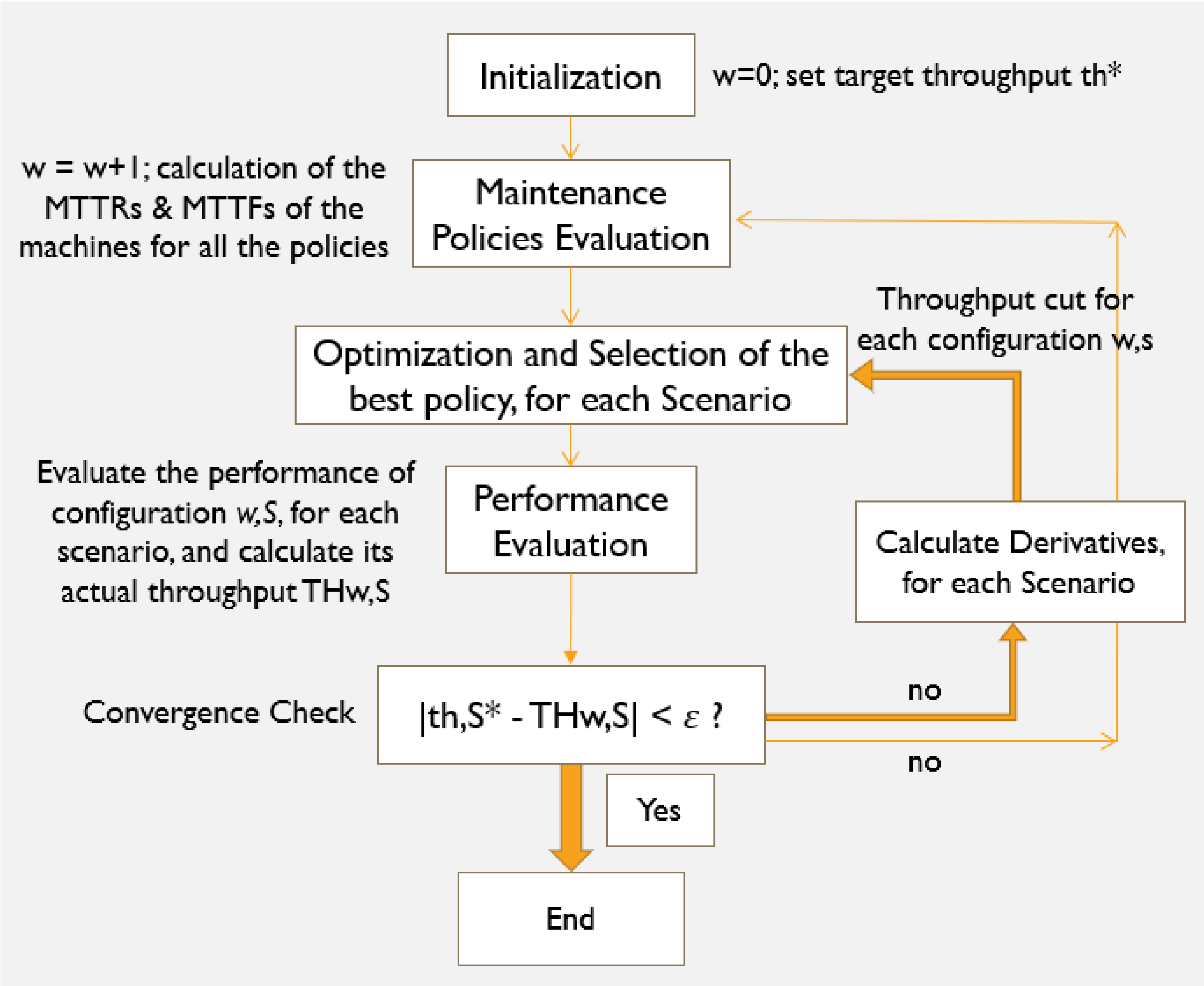


Figure 2.4: Graphical representation of the new algorithm

The 2-stage Stochastic Programming model

The 2-stage Stochastic Programming Model of the proposed algorithm has to take as inputs by the user:

- The set of machines Mj that can be implemented in the line;
- The matrix of assignment of the machines to the related stages.

  The line is made of 4 stages. Indeed, analytical models suggest the third stage as the critical one. Alternative improved machines on other stages don't influence so much the line performances, that stay nearly unvaried. The choice of the machine at the third stage, instead, is significant. That's why two options of machines are available. They differ in terms of production rate, since one is 20% greater than the other one. The machine having better performances is more expensive.

- The costs related to the machine types, the buffer slots and operators.

  The buffer positioned before the inspection machine has no costs related to the slots since no additional spots can be actually implemented, but only virtually by the downloading of the parts from the buffer to the ground made by the operator;

- The data regarding the Second Stage: the target throughput for each scenario "th2,s" and the realization probability of each scenario "p,s".

Then, the optimization problem has to be set: in particular, the minimization of the objective function. It is necessary to define the decision variables of the problem, specifying the related dimensions and, if present, lower and upper bounds.

The algorithm proceeds in the formalization of the objective function. It consists in a summation of all the variables that are a source of costs for production: the machines selected to be installed in each stage, the slots allocated for the buffers and operators working on the line.

$$minTotCost = \sum_k cm_j \cdot x_{i,j} + \sum_k cb_k \cdot xn_k + \sum_{k,s} p_s \cdot cb_k \cdot xn2_{k,s} + \sum_s p_s \cdot cop \cdot op_s$$

Figure 2.5: The Objective Function

- xi,j = machines
- xnk = buffer slots
- xn2k,s = additional buffer slots, for eac scenario s
- ops = number of operators
- ps = scenarios probabilities
- cmj = cost of the machines
- cbk = cost of the buffer slots
- cop = cost of the operators

Before going through the iterative algorithm implementation, the list of constraints that characterize the problem has to be transposed into mathematical formulation:

- Satisfy target throughput: the number of parts produced in one week has to be at least equal to the demand, in order to fully satisfy customers' requirements;

- Maximum throughput: the obtained throughput can not overcome the technical and mechanical limits characteristic of the line;

- Compatibility of machine type j to production stage i: a machine can be assigned only to its related stage;
- Assignment of exactly one machine type to production stage i: no more than one machine can be addressed to a single stage;
- Calculation of production rate in production stage i: the production rate of the selected machine becomes the production rate of the stage;
- Assignment of the selected machines to the 3D policy decision variable: the excluded machine must not be considered in the policy choice;
- Calculation of repair rate in production stage i;
- Calculation of failure time in production stage i;
- Assignment of the selected machines to the policy decision variable, for each Scenario: the excluded machine must not be considered in the policy choice.

Once initialized the 2-Stage Stochastic Programming Model, the optimization iterative algorithm is ready to start.

The iterative cycle

The cycle includes a section dedicated to:

- The choice of the best Maintenance Policy and the consequent update of the machines MTTRs and MTTFs, considering a specific configuration
- The 2-Stage Stochastic Programming Model resolution, to get the candidate optimal configuration solutions.
- The Performance Evaluation: the optimization algorithm works iteratively, looking for the best configuration for the case in analysis, interacting, for each cycle, with the performance evaluation model. The last takes as inputs the values of the actual candidate optimal configuration and gives as outputs the related throughput and the average level of the buffers. In order to check if the approximated throughput solution given by the algorithm is equal to the real one, it has to be fixed a margin of tolerance on the difference between the solutions: if the difference is smaller than the margin, iterations stop.

The algorithm must be performed for all the scenarios, taking into account the machines selection performed at First-Stage.

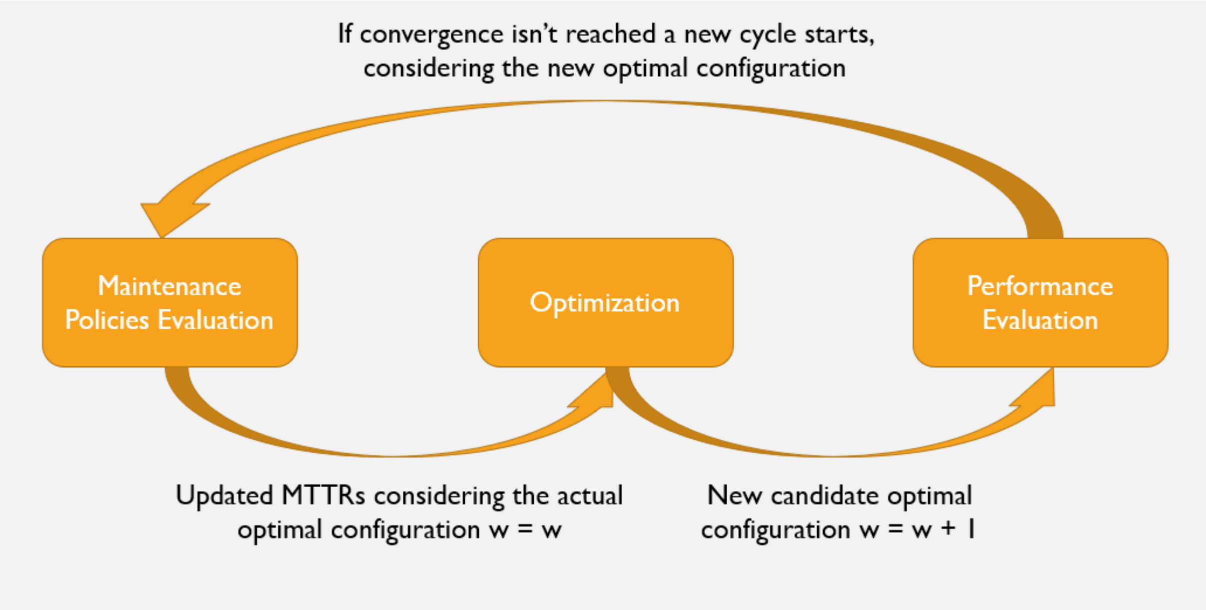


Figure 2.6: Interactions between sections

The iterative algorithm proceeds in that way:

- Primarily, a target throughput th* is fixed and a first attempt configuration ("w" = 0) is tested to check if it is the optimal one;
- The optimal Maintenance Policy is chosen, considering the data of the actual configuration in analysis;
- The optimization model is solved and the configuration w is stored;
- The performance of the system in configuration "w" is analysed focusing on the throughput value;
- The value of the actual throughput is compared to the one of the target throughput: if their difference is less than an arbitrary margin, the iterative algorithm stops, since the optimal solution has been found. Otherwise:
- The partial derivatives of the throughput are evaluated with respect to the configuration parameters and the hyperplane related to the configuration w is added thanks to the throughput cut: a new constraint for the optimization problem is added, since the actual throughput of the next configuration has to be smaller than the summation of the old throughput plus its partial derivatives.
- A new cycle starts and a new configuration ("w" = w + 1) is analysed.

The policy having a single operator

Before solving the problem, the proper control policy has to be selected, considering the actual configuration.

The line is made up of 3 machines, plus a final inspection machine, and three buffers. The first and the second buffer don't require the human intervention, while the final one, having 2 slots, needs the workforce to download and reload the parts in excess, when the buffer is full. The inspection machine is reliable and its availability is high. Instead, the other machines, performing rough turning, finish turning and stone grinding, undergo failures.

Having a single operator to repair them, a prioritization policy has to be applied, since more than one machine could be failed in a specific moment. Consequently, the mean time to repair of the machines change and become equal to their old MTTR plus an average waiting time for the operator's availability. Each of the 6 possible combinations has to be explored: "ABC / ACB / BAC / BCA / CAB / CBA". Since the stone grinding station can be occupied by two different machines and the optimization algorithm has to select the best one that guarantees the minimization of the overall costs, also the combinations considering the other machine type have to be calculated and stored. Changing the MTTRs of the machines, the availability of the machines and the MTTFs change.

At each iteration of the cycle the only input parameter that changes is the vector of the optimal slots for the three buffers. Dependently on the availability of the buffer slots, the starvation and blocking probabilities of the machines vary as well as all the production parameters related to the line. Therefore, at each cycle, the final matrices of the MTTRs and the MTTFs change and new input parameters are given to the optimization problem, that finds out a new possible optimal solution.

The operator is modelled as a server and the requests coming from the machines are compared to the arrivals of new clients to be served. So, the problem can be seen as a "Multi-Class Queue following a Prioritization Policy". It is a Multi-Class Model since the coming requests are not all equal but need different service time; it is an Open Model since the total number of jobs in the network is not a constant N, but varies according to the machine failures and to the downloading or reloading needs of the final buffer.

Iterations stop when no significant variations (differences under the arbitrary margin chosen) are certified, with respect to the values stored during the previous iteration, for what concerns the MTTRs and the MTTFs, as well as the operator's percentage of occupation on the final buffer. At each iteration, the behaviour of the line is evaluated, considering the updated data of the machines.

So, for each cycle:

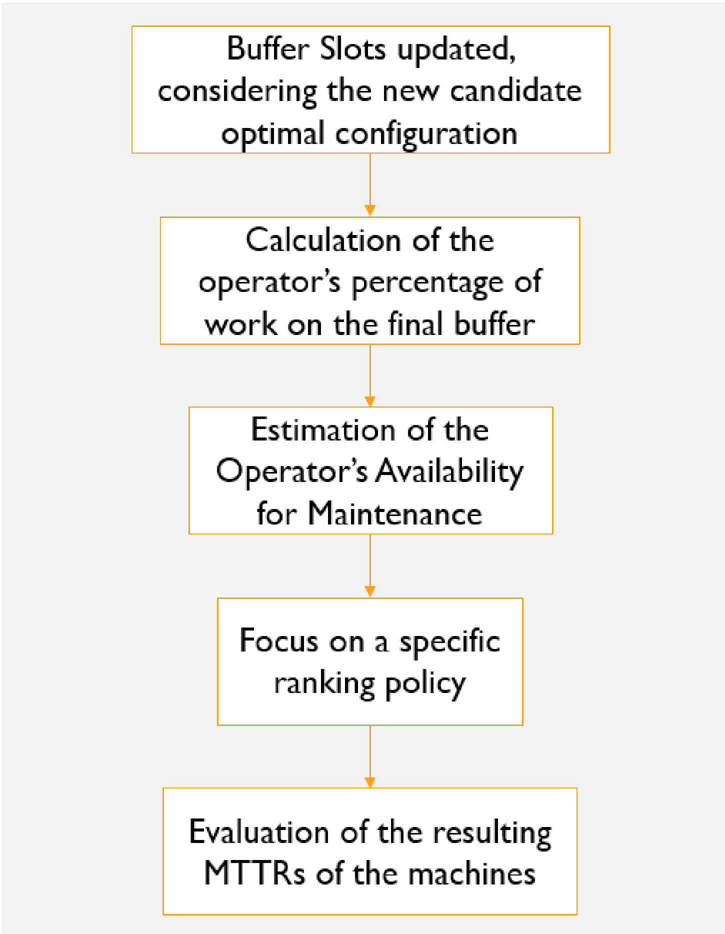


Figure 2.7: The Iterative cycle for the MTTRs update

For what concerns the operator's percentage of occupation in the final buffer, the value under analysis is the blocking probability of the third machine, the one just before the buffer itself, considering the number of slots equal to 2, that are the ones actually installed. This quantity has to be adjusted, since the operator does not work any time the buffer is full. It has to be taken into consideration the optimal value ("Noptimal") for the slots of the final buffer given as an output by the "solve" function, in the previous iteration. If this value is less or equal to 2, the percentage of occupation of the operator is equal to 0. Otherwise, if this value is greater than 2, the operator has to work only when the summation of the parts in the buffer plus the ones downloaded on the ground is in the range between 2 and Noptimal. Therefore, considering Noptimal as the number of slots of the final buffer, it is calculated the blocking probability of the third machine: this amount is subtracted from the value obtained at the beginning, considering 2 slots on the buffer, and thus the operator's right percentage of occupation is obtained.

The procedure to calculate the new MTTRs of the machines depends on the considered prioritization option. The MTTR1 of the first machine does not change so much: the difference is given by the possible operator's unavailability, if he is managing the situation on the final buffer. The new MTTR1 is equal to the operator's availability percentage times the original MTTR1.

The second machine sees a delay in the operator's readiness whenever he is dealing with the final buffer or with the top-ranked machine. The new MTTR2 consists in a weighted average between the original MTTR2, times the probability not to have a simultaneous down-state of the first and second machine, and an MTTR2 equal to the original MTTR2 plus half the actual MTTR1, times the probability to have a simultaneous down-state of the first and second machine, that corresponds to the additional average waiting time for the operator's availability. This quantity, as done before for the MTTR1, has to be multiplied by the operator's percentage of time free from duties on the final buffer.

The third machine, instead, sees a delay in the operator's readiness whenever he is dealing with the final buffer or with the first two ranked machines. The new MTTR3 consists in a weighted average between 4 components:

- The original MTTR3, times the probability not to have: a simultaneous down-state of the first and third machine, or a simultaneous down-state of the second and third machine, or a simultaneous down-state of the three machines together;
- An MTTR3 equal to the original MTTR3 plus half the actual MTTR1, that corresponds to the additional waiting time for the operator's availability, times the probability to have a simultaneous down-state of the first and third machine minus the probability of a simultaneous down-state of the three machines together;
- An MTTR3 equal to the original MTTR3 plus half the actual MTTR2, that corresponds to the additional waiting time for the operator's availability, times the probability to have a simultaneous down-state of the second and third machine minus the probability of a simultaneous down-state of the three machines together;

- An MTTR3 equal to the original MTTR3 plus half the actual MTTR1 and half the actual MTTR2, that corresponds to the additional average waiting time for the operator's availability, times the probability to have a simultaneous down-state of the three machines together.

After having achieved all the needed updated values of the MTTFs and MTTRs, their value is stored inside the dedicated matrices, in the column related to the policy in analysis. The following iteration is now ready to start, taking into account the new data of the machine structs.

The policy considering two operators

Adding one operator to the line changes the procedure for the choice of the maintenance policy just described. Indeed, the workforce employed is now able to satisfy simultaneously two different tasks. Since the downloading of the final buffer, when full, come first, one of the operators will satisfy its requirements, dedicating a percentage of time equal to the already defined opbusy. The second operator enters the scene as a request from a failed machine comes when the first worker is busy on the buffer.

The difference is that now the first-prioritized machine does not see any alterations in its MTTR, since it has no waiting time. For what concerns the second-prioritized machine, instead, its waiting time is greater than 0 only when its down-state happens at the same time of a down-state of the first-prioritized machine and of a downloading request on the final buffer.

The last-prioritized machine, instead, sees a delay in the operators' readiness whenever they are dealing with the final buffer and one of the two first ranked machines or with the two first ranked machines. The new MTTR3 consists in a weighted average between 4 components:

- The original MTTR3, times the probability not to have: a simultaneous down-state of the first and third machine plus a request from the final buffer, or a simultaneous down-state of the second and third machine plus a request from the final buffer, or a simultaneous down-state of the three machines together;
- An MTTR3 equal to the original MTTR3 plus half the actual MTTR1 or half the actual opbusy (depending on which one is smaller), that corresponds to the additional waiting time for the operator's availability, times the probability to have a simultaneous down-state of the first and third machine minus the probability of a simultaneous down-state of the three machines together;

- An MTTR3 equal to the original MTTR3 plus half the actual MTTR2 or half the actual opbusy (depending on which one is smaller), that corresponds to the additional waiting time for the operator's availability, times the probability to have a simultaneous down-state of the second and third machine minus the probability of a simultaneous down-state of the three machines together;
- An MTTR3 equal to the original MTTR3 plus half the actual MTTR1 and half the actual MTTR2, that corresponds to the additional average waiting time for the operator's availability, times the probability to have a simultaneous down-state of the three machines together.

Performance Evaluation

Once got the updated MTTRs and MTTFs of the machines, the optimization problem is solved considering the decision variables, objective function and constraints already fixed at the beginning: a candidate incumbent optimal configuration solution is achieved and stored. It is necessary to assess the system performance of the new configuration to check its effective optimality: the approximated throughput, got through the optimization problem, has to be compared to the real one. A performance evaluation model, taking as inputs the data of the configuration, specifically the selected machines and buffer slots, and giving as outputs its throughput and the associated average buffer level, must be carried out.

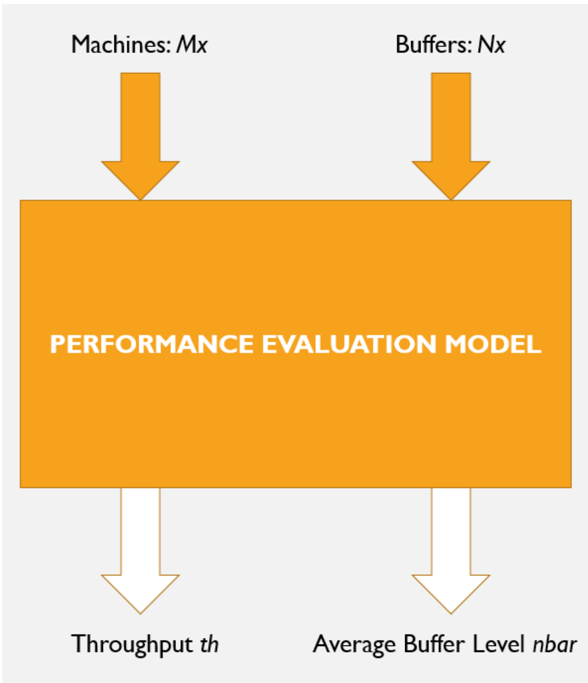


Figure 2.8: The Performance Evaluation Model

The actual throughput is thus compared to the approximated one and if their difference is lower than a margin, arbitrary defined a priori, the convergence has been reached.

Otherwise, a new iteration is ready to start. Before moving to the next cycle, data of the actual system configuration have to be stored in Matlab, in particular the approximated throughput and its partial derivatives with respect to production parameters, i.e. the buffer levels, the production rates, the failure rates, the repair rates and the mean time to failures.

Indeed, a throughput cut constraint has to be set, using the hyperplane method. A piece-wise linear approximation of the throughput curve is found by calculating hyperplanes in the neighborhood of the configuration. The set of hyperplanes is obtained from the first-order analytical derivatives of the performance, with respect to the configuration parameters, which guide the optimization method in promising search areas. The hyperplanes are tangent in a specific point of the performance curve function and they approximate it. This way it is possible to operate a linearization to the problem, making possible a fast resolution through the existing methods.

Once the convergence is reached, after few or many iterations, the optimal configuration is found together with its related throughput and optimal buffer allocation. This iterative procedure must be performed considering all the scenarios in analysis: what changes among them is not the choice on the machines to implement, defined at First-Stage, but the optimal buffer allocation and the selection of the best maintenance policy, ranking operator's tasks.

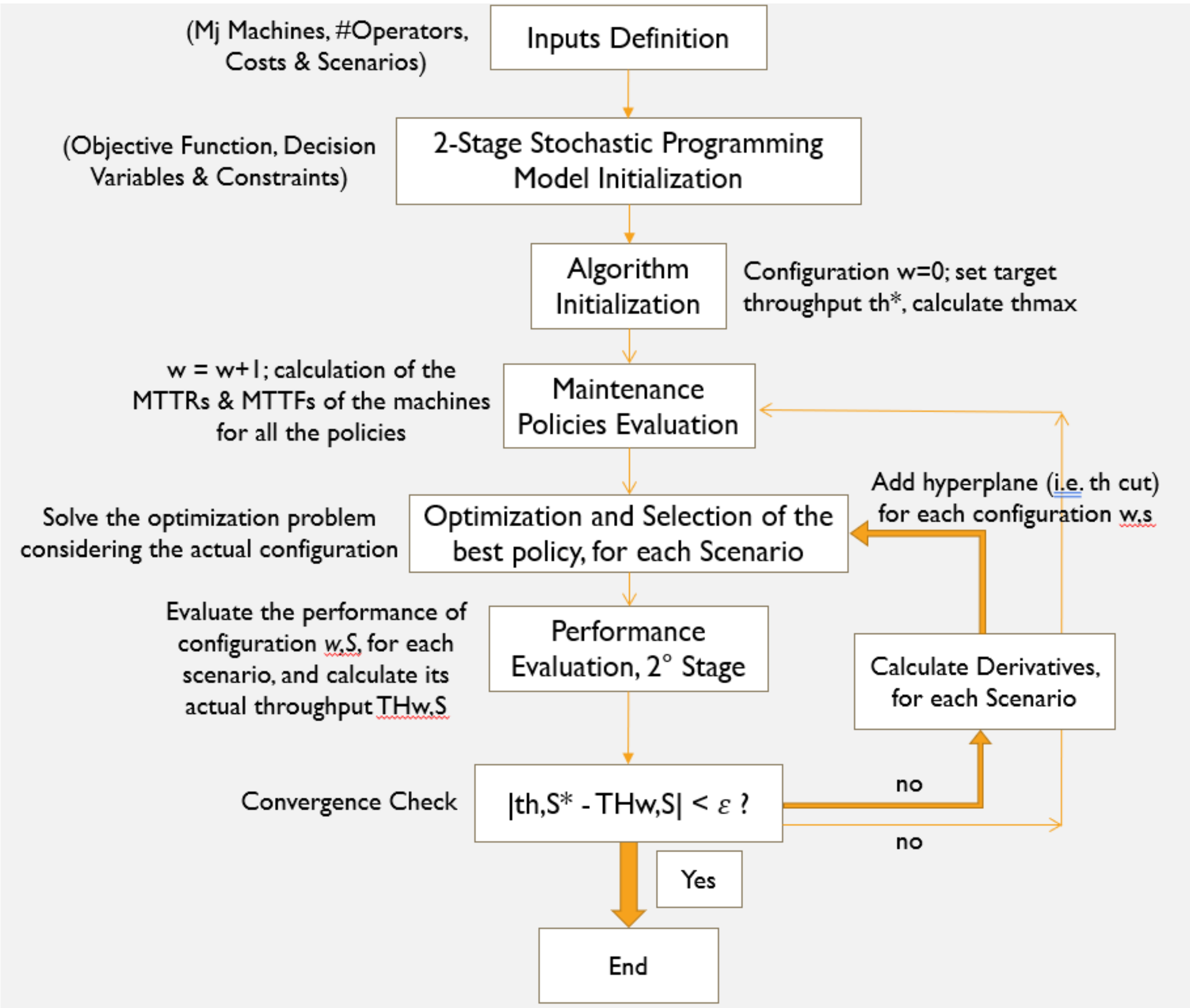


Figure 2.9: Block Diagram of the Method

3 Results

Once formalized the optimization algorithm, run it and got the outputs, results have to be checked and validated.

The algorithm optimizes the manufacturing line: for each future scenario, it is selected the best policy to mitigate the consequences of the operator's interference issue on the line performances. By this way it is set the minimal total buffer capacity which satisfies a given demand rate requirement, minimizing the related production costs.

3.1. Validation

Following the Here&Now approach, the company must decide now the machines to install, that constitutes the decision variable at first-stage.

The result got by running the algorithm, in terms of machine selection, is the following one:

	First-Stage Decision
Selected Machines	A B Cbis D

e

Table 3.1: First-Stage Decision

Therefore, the machine selected at the third stage is the one having the highest production rate, even if its cost impacts more on the objective function to minimize.

For what concerns the Second-Stage for each demand rate, two different average lot sizes are considered: the one having 10 pieces and the one having 60 pieces. The optimization algorithm gives as output the best configurations for each investigated scenario, from 160 to 200 parts/week, considering only the decades.

The consequent list of scenarios to analyze results to be the following:

	th160	th170	th180	th190	th200	Lot10	Lot60
Sc.1	X	-	-	-	-	X	-
Sc.2	X	-	-	-	-	-	X
Sc.3	-	X	-	-	-	X	-
Sc.4	-	X	-	-	-	-	X
Sc.5	-	-	X	-	-	X	-
Sc.6	-	-	X	-	-	-	X
Sc.7	-	-	-	X	-	X	-
Sc.8	-	-	-	X	-	-	X
Sc.9	-	-	-	-	X	X	-
Sc.10	-	-	-	-	X	-	X

Table 3.2: The scenarios in analysis

For what concerns the first four scenarios, having 160 and 170 parts/week as throughput, the related optimal configurations don't require additional buffer slots with respect to the default ones ([3 3 3]), independently from the policy chosen; only one operator is needed.

	Sel. Policy	2° op	Th.	Buffers
Scenarios #1,2,3,4	All are good	No	0.787	[3 3 3]

Table 3.3: Optimal Configurations

When facing a target throughput equal to 180 parts/week, instead:

	Sel. Policy	2° op	Th.	rA	rB	rC	Buffers	k€
10	B Cbis A	No	0.0855	0.02147	0.03209	0.05367	[6 5 4]	225.753
60	Cbis B A	No	0.0797	0.02064	0.03106	0.05633	[4 3 3]	222.753

Table 3.4: Optimal Configurations, Throughput 180

The weekly throughput of the first configuration is of 181 parts/week, whereas the second one reaches 182 parts/week. The difference in the Time Unit throughput is due to the higher weekly availability of time in the second case, since less set-ups are needed.

Two different policies are chosen for a target throughput equal to 180, depending on the average lot size (10 or 60) There is no need of an additional second operator working on the line.

Here the results for a target throughput equal to 190 parts/week:

	Sel. Policy	2° op	Th.	rA	rB	rC	Buffers	k€
10	B Cbis A	No	0.0903	0.02011	0.03136	0.05164	[10 4 7]	230.253
60	B A Cbis	No	0.0841	0.02080	0.03135	0.04462	[6 4 5]	224.253

Table 3.5: Optimal Configurations, Throughput 190

Two different policies are chosen also for a target throughput equal to 190, depending on the average lot size. There is no need of an additional second operator working on the line.

Finally, the results for a target throughput equal to 200 parts/week:

	Sel. Policy	2° op	Th.	rA	rB	rC	Buffers	k€
10	B A Cbis	No	0.0953	0.02087	0.03135	0.04887	[15 6 8]	240.753
60	B A Cbis	No	0.0887	0.02068	0.03135	0.4185	[8 4 6]	227.253

Table 3.6: Optimal Configurations, Throughput 200

3.1.1. Model vs Simulink: Results comparison

Once got the outputs from the Optimization Algorithm, it is necessary to verify them through simulations in order to legitimate their validity. This procedure has been performed using Simulink: the line in analysis is modelled, characterized by 3 machines performing rough turning, finish turning and stone grinding, plus a final inspection machine, and 3 buffers, the last one having two slots.

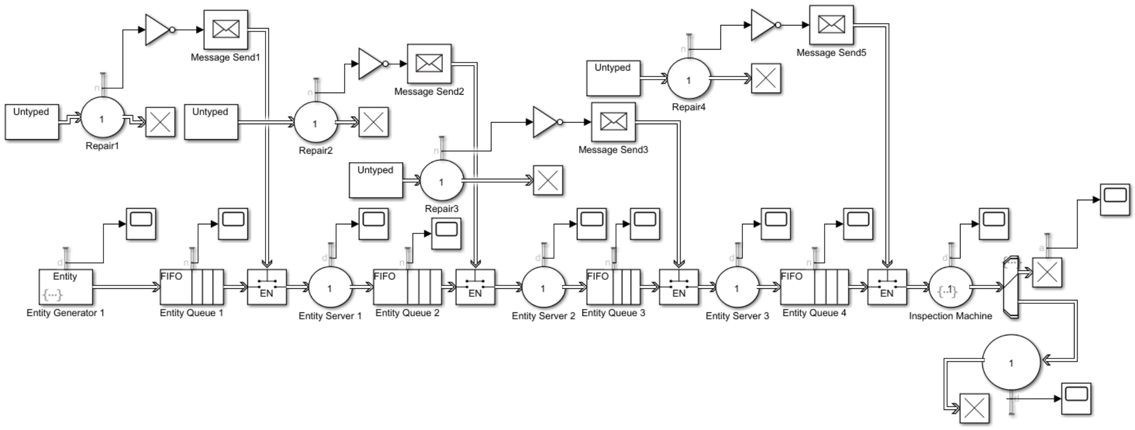


Figure 3.1: The Manufacturing System Modelled in Simulink

The same inputs of the performance evaluation model are given, in particular: the selected machines, the optimal buffer slot allocation and the optimal number of operators, that influences the repair rates of the machines. Then the results are compared setting a margin of tolerance on the difference equal to 2%.

The actual throughput given by the Model has to be verified through the Simulink Model, considering the same inputs for both the cases.

	Model (T.U. th)	**Model (Total th.)**	**Simulink (th.)**	**% Diff**
Th 180, Size 10	0.0855	181 parts	180 parts	0.55 %
Th 180, Size 60	0.0797	182 parts	180 parts	1.11 %
Th 190, Size 10	0.0903	191 parts	190 parts	0.52 %
Th 190, Size 60	0.0841	191 parts	190 parts	0.52 %
Th 200, Size 10	0.0953	202 parts	205 parts	1.48 %
Th 200, Size 60	0.0887	201 parts	200 parts	0.49 %

Table 3.7: Sixth Simulink Check, TH=220

Scenarios considering an average lot size of 10 pieces/lot require a time unit throughput higher than the ones having 60 pieces/lot, since the total available time is lower due to more frequent set-ups.

The confidence interval chosen a priori is ±2%: the comparison of results shows differences that satisfy requirements. Results are validated.

3.1.2. Exhaustive Research

Once given the optimized results regarding all the scenarios, it is appropriate to focus on a specific one and demonstrate that the chosen configuration is actually the one that minimizes the related expenditures. In order to do that, all the possible buffer combinations have to be taken into account and the associated cost and throughput have to be listed. The selected scenario is the one having as target throughput 190 and an average lot size equal to 10. The resulting configuration given by the algorithm is the following:

	Sel. Policy	2° op	Th.	rA	rB	rC	Buffers	k€
10	B Cbis A	No	0.090343	0.02011	0.03136	0.05164	[10 4 7]	230.253

Table 3.8: Optimal Configuration: Throughput 190, Average Lot Size 10

In order to check if the algorithm works well, in the table below, all the main configurations, following the same policy and having less or equal costs with respect to the selected one, are reported.

	Sel. Policy	Actual th.	rA	rB	rC	Buffers	k€
10	B Cbis A / No op	0.07854	0.02242	0.03209	0.05545	[3 3 3]	222.753
10	B Cbis A / No op	0.07981	0.02223	0.03209	0.05539	[4 4 4]	222.753
10	B Cbis A / No op	0.08294	0.02216	0.03209	0.05507	[5 5 5]	224.253
10	B Cbis A / No op	0.08331	0.02214	0.03209	0.05496	[4 10 12]	230.253
10	B Cbis A / No op	0.08426	0.02211	0.03209	0.05487	[5 9 12]	230.253
10	B Cbis A / No op	0.08642	0.02147	0.03209	0.05367	[6 6 6]	227.253
10	B Cbis A / No op	0.08761	0.02101	0.03201	0.05282	[6 8 12]	230.253
10	B Cbis A / No op	0.08797	0.02098	0.03204	0.05272	[7 7 7]	230.253
10	B Cbis A / No op	0.08882	0.02089	0.03203	0.05261	[8 6 7]	230.253
10	B Cbis A / No op	0.08917	0.02088	0.03182	0.05260	[8 6 12]	230.253
10	B Cbis A / No op	0.09003	0.02065	0.03153	0.05271	[10 3 12]	228.753
10	B Cbis A / No op	0.09021	0.02056	0.03146	0.05169	[11 3 12]	230.253
10	B Cbis A / No op	0.09022	0.02033	0.03136	0.05164	[10 4 5]	230.253
10	B Cbis A / No op	0.09025	0.02094	0.03122	0.05169	[9 5 12]	230.253
10	B Cbis A / No op	0.09029	0.02024	0.03136	0.05164	[10 4 6]	230.253

Table 3.9: Exhaustive Reasearch, Considering a Single Policy

It is evident that no configuration is better than the one chosen by the algorithm, since the actual throughput needed to reach 190 parts/week, having an average lot size equal to 10, is equal to 0.0903.

The same procedure is done considering the main configurations following the other policies and having less or equal costs with respect to the selected one.

	Sel. Policy	Actual th.	rA	rB	rC	Buffers	k€
10	A B Cbis / No op	0.08754	0.02246	0.02701	0.04463	[9 5 12]	230.253
10	A Cbis B/ No op	0.08473	0.02246	0.02583	0.04916	[9 5 12]	230.253
10	B Cbis A/ No op	0.09023	0.02080	0.03134	0.04463	[9 5 12]	230.253
10	Cbis A B / No op	0.08695	0.02003	0.02583	0.05569	[9 5 12]	230.253
10	Cbis B A / No op	0.09022	0.02012	0.03063	0.05568	[9 5 12]	230.253
10	A B Cbis / No op	0.08797	0.02247	0.0270	0.04462	[10 4 7]	230.253
10	A Cbis B/ No op	0.08491	0.02248	0.02582	0.0492	[10 4 7]	230.253
10	B Cbis A/ No op	0.09028	0.02079	0.03136	0.04458	[10 4 7]	230.253
10	Cbis A B / No op	0.08713	0.02201	0.02581	0.05571	[10 4 7]	230.253
10	Cbis B A / No op	0.09029	0.02024	0.03136	0.05164	[10 4 7]	230.253

Table 3.10: Exhaustive Research, Comparison Between Policies

Since neither one of these solutions respects the requirements, it can be definitely stated that the algorithm actually gives as output the best configuration possible, reaching the target throughput with the minimum cost.

3.2. Sensitivity Analysis

A sensitivity analysis is carried out to determine how changes in input variables affects target variables. It is useful to predict effects on outputs given by future decisions on production parameters. The focus is on the system performance and on the relationships among the decision variables. In particular it is investigated:

- How a change in the maintenance policy chosen affects the performance of the line
- How a change in the target throughput or a change in the average lot size affect costs related to the line
- How an additional operator increases the performance of the line

- How the configuration of the line affects the repair rates of the machines

The evaluated scenarios cover the range of throughput between 160 and 200, considering only the decades: it is interesting to see what happens also in the middle of each step (i.e. target throughput equal to 165, 175, 185 and 195 parts/week).

The sensitivity analysis must stress the importance of optimizing through a maintenance policy, showing how the selection of a not optimal configuration or control policy negatively impacts the performance of the line and increases the production costs.

Maintenance Policy vs Throughput, @ N = N*

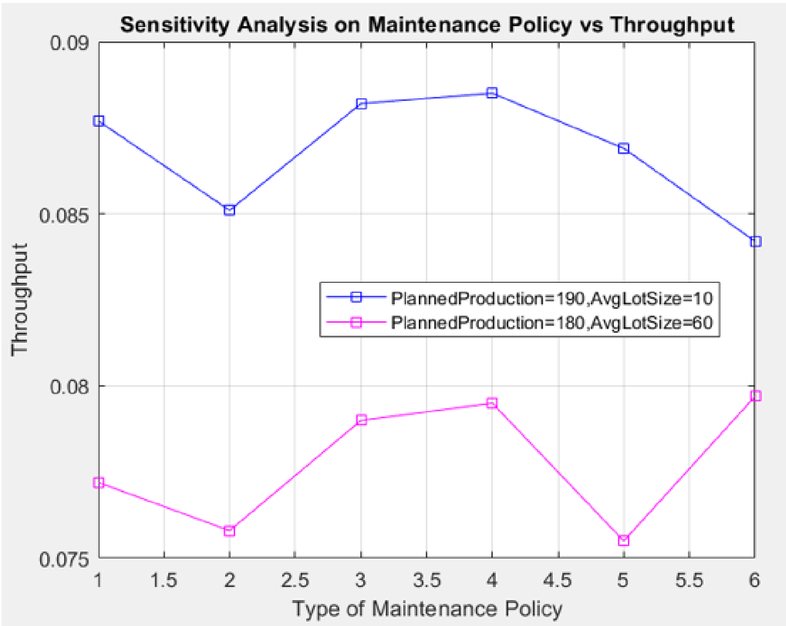


Figure 3.2: Maintenance Policy vs Throughput, @ N = N*

Having three machines to manage, in terms of maintenance, since the last one, the inspection machine, is assumed to be perfectly reliable, six typologies of maintenance policies are possible: ABC, ACB, BAC, BCA, CAB, CBA. Looking at this graph, it is evident the importance of optimizing the line according to a specific policy. Given a fixed buffer allocation, it can be seen how much the different prioritization of machines, for the operator in charge of the repair, affects the overall performance of the line.

Dependently on the target throughput and on the related buffer slots allocation, a specific policy may result to be the best in one scenario and the worst in another one, as happens for CBA in this case.

Maintenance Policy vs Buffer Slots, Th = th*

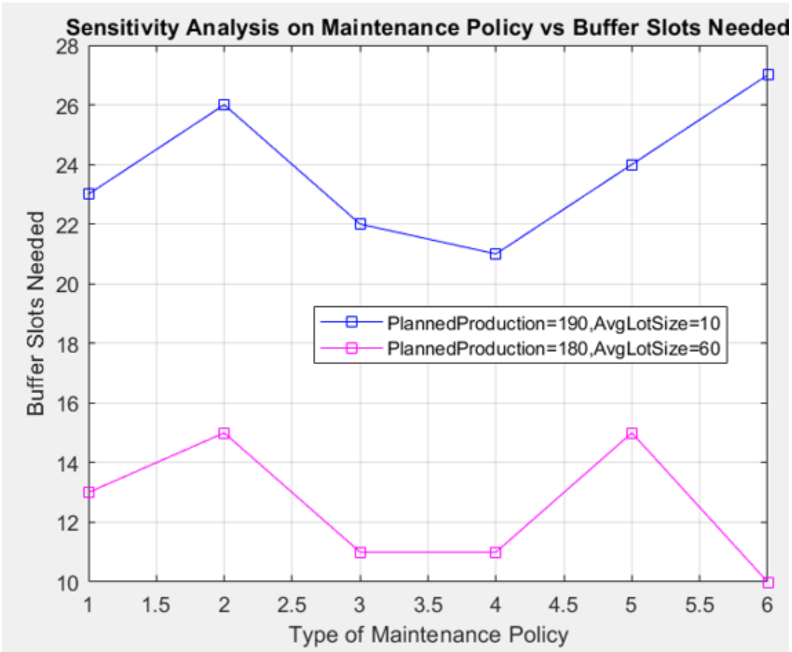


Figure 3.3: Maintenance Policy vs Buffer Slots, Th = th*

A further analysis can be performed, considering the same previous cases. Since the company must attain the weekly target throughput, the line configuration changes in terms of buffer slots allocated. The selection of the best maintenance policy in the considered scenario, as can be seen in the graph, results to be decisive in order to achieve a target system performance requiring fewer buffer slots to be allocated.

Buffer Slots vs Repair Rates

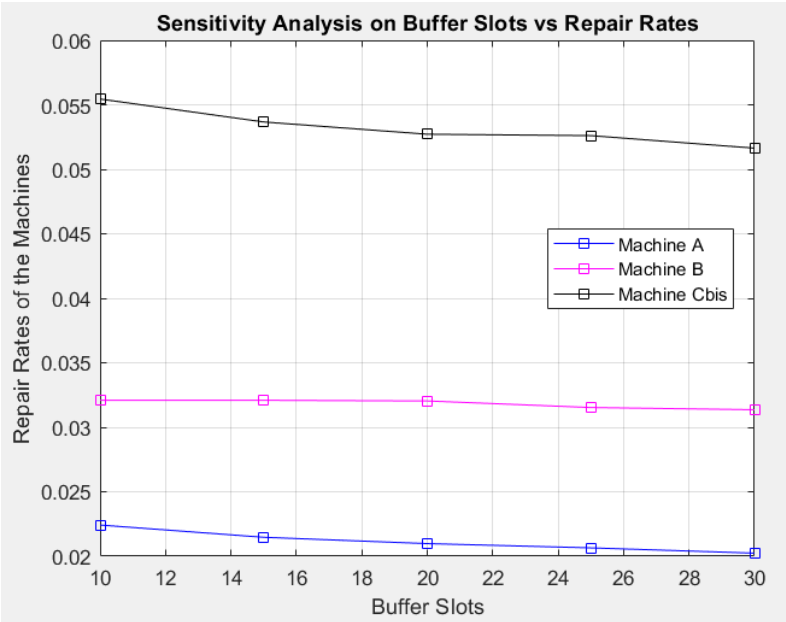


Figure 3.4: Buffer Slots vs Repair Rates

The relevance of optimizing is stressed when evaluating the consequences of an increase of the number of buffer slots on the Repair Rates of the Machines: as buffer slots grow, the Repair Rates of the machines decrease.

This is due to the fact that if the buffer positioned before the inspection machine needs to be virtually enlarged, the operator working on the line should spend more time loading and downloading it. His availability for the requests of the machines would decrease as his readiness to accomplish their tasks. Therefore, the performance of the line worsens as buffer slots increase.

Indeed, the duty of the optimization algorithm is to decrease as much as possible the number of slots on the final buffer.

Number of Operators vs Repair Rates

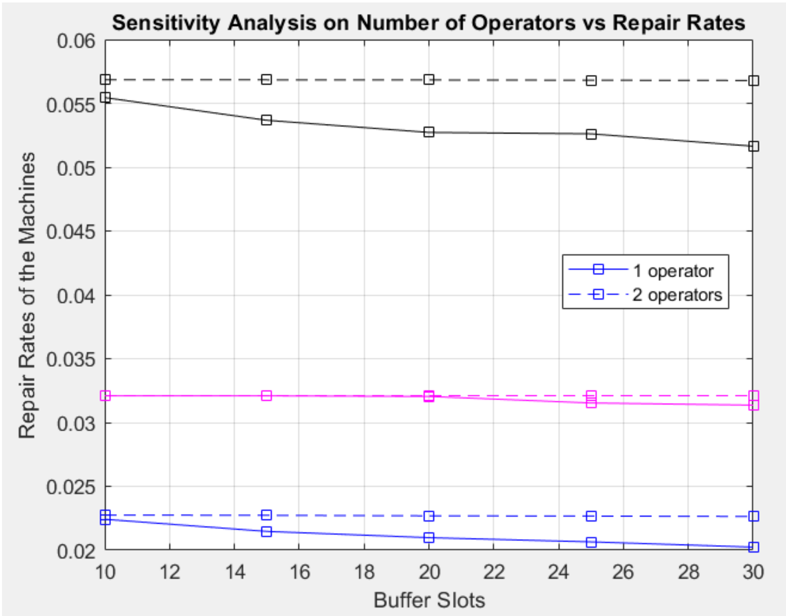


Figure 3.5: Number of Operators vs Repair Rates

Hiring a new operator working on the line implicates great improvement for the line performance. Having an additional worker means that two tasks can be performed simultaneously. Since the line in analysis has four possible tasks to commission, i.e. the loading and downloading of the final buffer and the three of the machines to repair, it is very unlikely that more than two happen together.

That's why the repair rates of the machines don't decrease in case of 2 operators working o the line, even if slots on the final buffer increase.

In conclusion, hiring a new operator significantly improves the performances of the line but it has to be taken into account its related cost. If the line can reach equal performances employing only operator and saving costs, the algorithm opts for the less expensive solution.

Throughput vs Maintenance Policy

	ABC	ACB	BAC	BCA	CAB	CBA
Throughput 180, Lot Size 10	-	-	-	X	-	-
Throughput 180, Lot Size 60	-	-	-	-	-	X
Throughput 190, Lot Size 10	-	-	-	X	-	-
Throughput 190, Lot Size 60	-	-	X	-	-	-
Throughput 200, Lot Size 10	-	-	X	-	-	-
Throughput 200, Lot Size 60	-	-	X	-	-	-
Throughput 210, Lot Size 10	-	-	X	-	-	-
Throughput 210, Lot Size 60	-	-	X	-	-	-
Throughput 220, Lot Size 10	-	-	X	-	-	-
Throughput 220, Lot Size 60	-	-	X	-	-	-

Figure 3.6: Throughput vs Maintenance Policy

This table proves the fact that it does not exist a single maintenance policy that is the best in each scenario. Dependently on the actual configuration and on the required system performance, what they were once considered the critic machines can become the safest, not requiring anymore a prioritization over the others when performing the repair.

Machine Selection vs Throughput

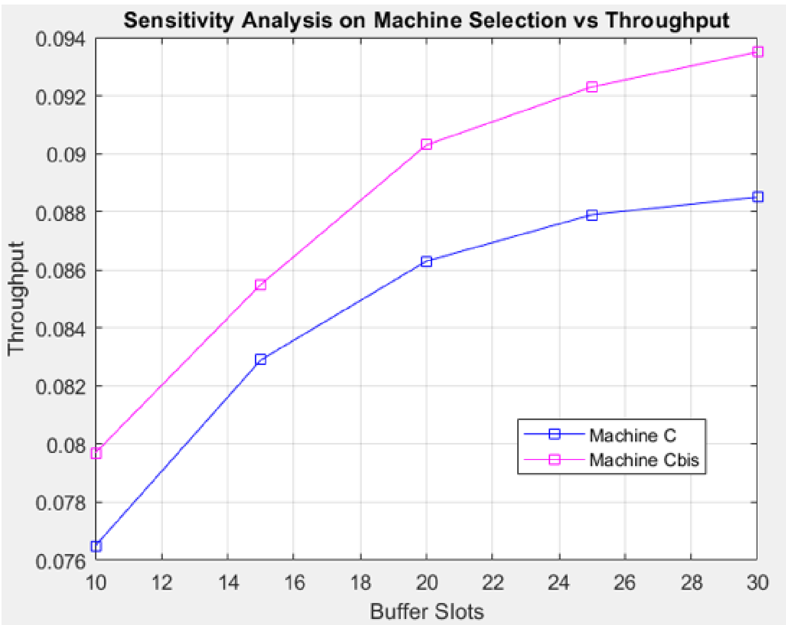


Figure 3.7: Machine Selection vs Throughput

The 2-Stage Stochastic Programming Model sees at First-Stage, as decision variable, the machine selection at the third stage. Here can be noticed how much a 20% increase of the production rate affects the time unit throughput performance of the line. The better performing machine costs more but guarantees higher system performance considering the same number of buffer slots allocated.

Furthermore, the THmax for the line having the cheaper machine is smaller with respect to the one having the more expensive one installed. Thus, it can happen that the first configuration can't satisfy the highest weekly demand rates of the company.

Machine Selection vs Buffer Slots

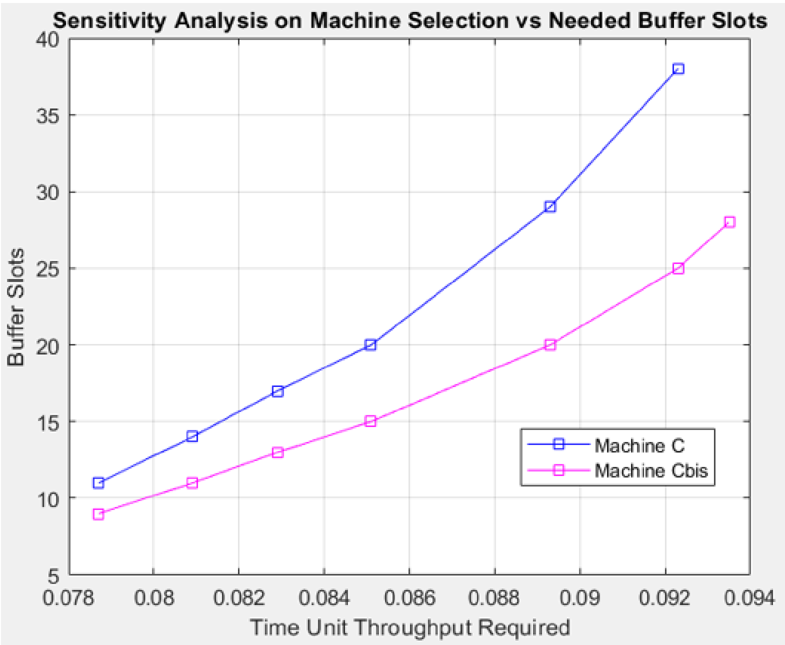


Figure 3.8: Machine Selection vs Buffer Slots

In order to conclude the considerations done at the previous step, it is interesting to see the difference, in terms of buffer slots requirements, between the configurations considering the 2 different machine possibilities. It can be noticed that, as the throughput requirement increases, the additional buffer slots needed by the configuration having the less performing machine increases. Therefore, even if the first configuration is less expensive, significant additional costs may be required in the future.

Buffer Slots vs Throughput

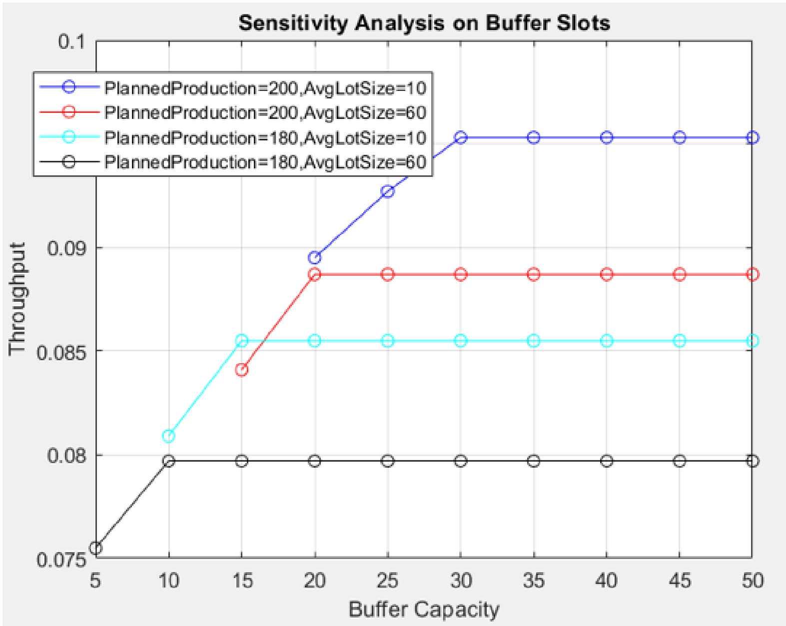


Figure 3.9: Buffer Slots vs Throughput

When the planned production is high and lots are small (blue case), a quite large buffer capacity is needed to satisfy the target throughput (n=30). When lot sizes are larger, the minimum buffer capacity reduces to n=15 (cyan case). Indeed, the buffer absorbs the variability caused by disruptions, which is smaller when operators are less busy with setups. The same is valid for the cases considering lower planned production: having small lots the needed buffer capacity is n=20, whereas having larger slots it decreases to n=10.

Operators vs Throughput

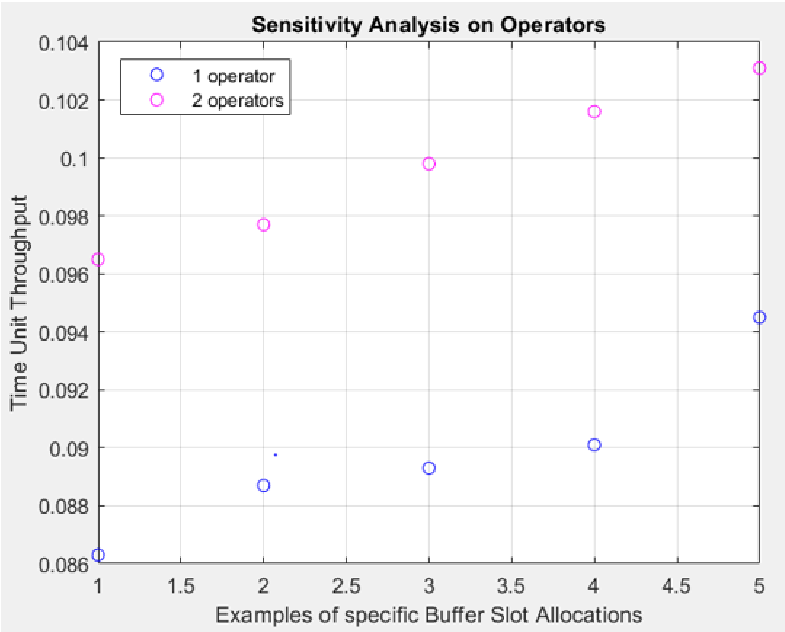


Figure 3.10: Operators vs Throughput

Different Buffer Slot Allocations are taken into account in this diagram, in particular: [7 3 4], [8 4 4], [9 6 6], [11 7 4], [14 7 8].

What changes as input is the number of operators working on the line. The operator's interference issue decreases by adding a new worker on the line. Consequently, indeed, the waiting time for one of the operator's availability decreases, thus reducing the MTTRs of the machines. The related machine down-states decrease in time, thus increasing the time unit throughput of the line.

Throughput vs Cost, @ Lot Size 10

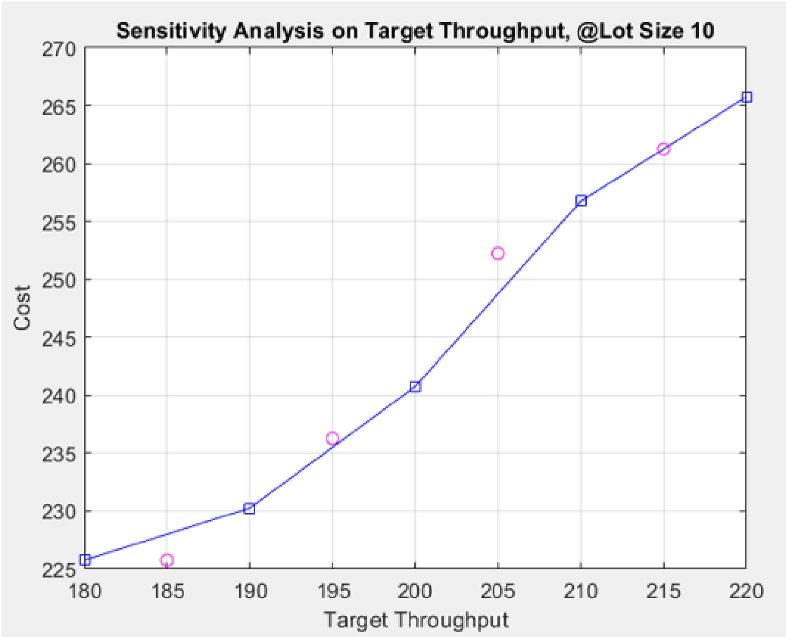


Figure 3.11: Throughput vs Cost, @ Lot Size 10

The cost of the line increases almost linearly as the target throughput increases. Middle configurations, the ones at 185/195/205/215, require expenditures that are nearly the medium of the ones related to the extreme ones. A strong increment of the cost function can be seen between 200 and 210 and is due to the hiring of an additional operator working on the line, in order to attain throughput requirements.

Throughput vs Cost, @ Lot Size 60

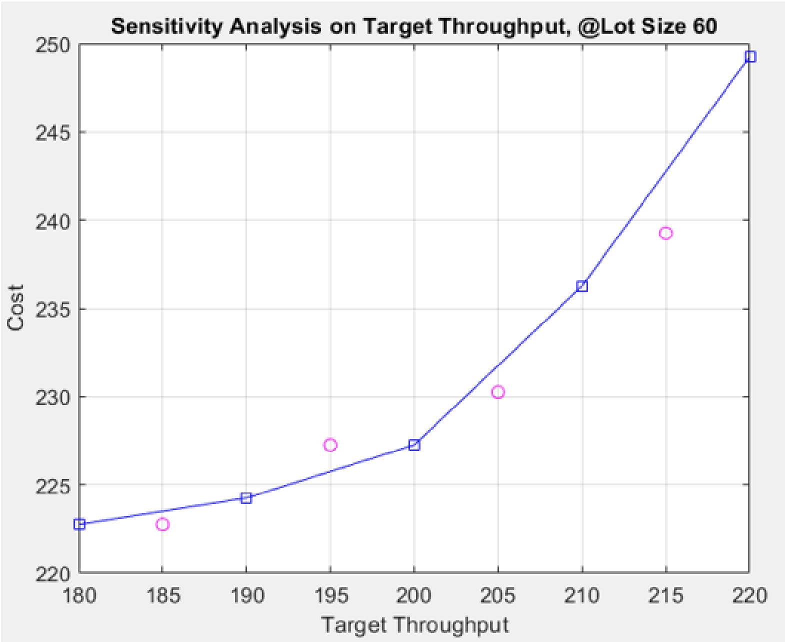


Figure 3.12: Throughput vs Cost, @ Lot Size 60

The cost function changes behaviour with respect to the previous case. At the beginning, indeed, few more buffer slots are needed as throughput increases, thus causing few added costs. A significant increase in costs is seen between 210 and 220, due also in this case to the additional operator. Costs decrease with respect to the case considering an average lot size equal to 10. The reason is investigated in the following diagram.

Average Lot Size vs Cost

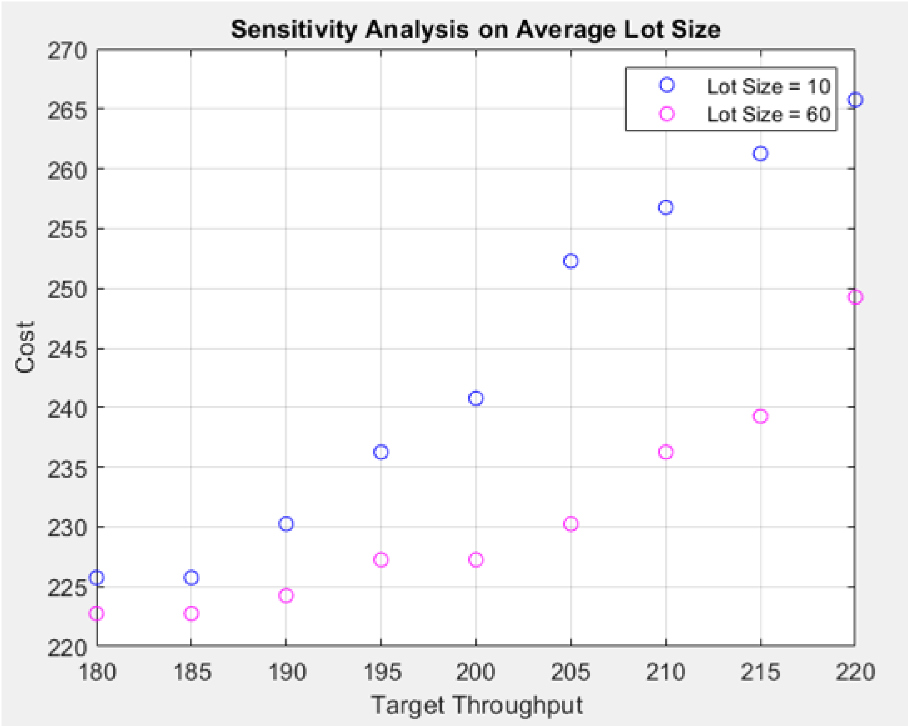


Figure 3.13: Average Lot Size vs Cost

This diagram takes into consideration the two possibilities, planned by the company, concerning the average lot size. Having a greater average lot size means having less set-ups to perform. Set-ups require time, so decreasing their number increases the total available production time. This means that, considering the same target throughput for both the cases, the required time unit throughput will be greater for the configuration having an average lot size = 60.

4 | Conclusions

This thesis proposed an optimization method, including a stochastic analytical model for performance evaluation, useful to get the joint optimization, in terms of configuration and control policies, of the manufacturing line in analysis. In particular, the considered production was affected by the issue related to the operator's interference, that worsened the line performances: the applied control policy concerned the maintenance area, ranking the worker's tasks to accomplish. the operator was modelled as a server and the machines to be repaired as its clients. The optimization algorithm took as inputs the parameters of the machines, operators, costs and data related to the future scenarios. Then the optimization problem was formalized, aiming at minimizing the overall costs of the line. The optimization algorithm worked iteratively looking for the best configurations for the case in analysis and interacting, for each cycle, with the performance evaluation model.

Once got the optimal configurations and the related throughputs, simulations were run using Simulink, to check the validity of the results obtained. Since the values differed less than 1%, the method was legitimated. In order to deeply examine its efficacy an exhaustive research was conducted on a specific considered scenario, that gave the same output of the optimization algorithm. A sensitivity analysis on middle set configurations and on mutual relationships between the key performance indicators of the line was also conducted to check the robustness of the method, which was verified.

As a result it can be stated that, generally, each time a company has to cope with a change in the value of the weekly demand, this algorithm has to be used in order to get the best configuration for the actual specific boundary conditions. What changes among the inputs is the target throughput to satisfy and the list of possible alternative machines. This optimization algorithm properly suits the Industry 4.0 case, where not so much workforce is employed in the production line management. However, a similar situation could arise also in the contest of traditional companies, where all the operators dedicated to a manufacturing system cooperate on a single task moving all together where needed.

Bibliography

[1] Maria Chiara Magnanini and Tullio AM Tolio. A model-based digital twin to support responsive manufacturing systems. *CIRP Annals*, 70(1):353–356, 2021.

[2] Maria Chiara Magnanini, Oleksandr Melnychuk, Anteneh Yemane, Hakan Strandberg, Itziar Ricondo, Giovanni Borzi, and Marcello Colledani. A digital twin-based approach for multi-objective optimization of short-term production planning. *IFAC-PapersOnLine*, 54(1):140–145, 2021.

[3] Chrissoleon T Papadopoulos, Michael EJ O'Kelly, Michael J Vidalis, and Diomidis Spinellis. *Analysis and design of discrete part production lines*. Springer, 2009.

[4] Stanley B Gershwin. The future of manufacturing systems engineering. *International Journal of Production Research*, 56(1-2):224–237, 2018.

[5] Maria Chiara Magnanini, Walter Terkaj, and Tullio Tolio. Robust optimization of manufacturing systems flexibility. *Procedia CIRP*, 96:63–68, 2021.

[6] Maria Chiara Magnanini, Walter Terkaj, and Tullio AM Tolio. A novel algorithm for optimal buffer allocation in automated asynchronous unreliable lines. *The International Journal of Advanced Manufacturing Technology*, 122(11):4405–4419, 2022.

[7] Maria Chiara Magnanini and Tullio AM Tolio. Robust improvement planning of automated multi-stage manufacturing systems. In *Selected Topics in Manufacturing*, pages 61–75. Springer, 2022.

[8] RICCARDO ROSSIGNOLI. An analytical tangent hyperplane method to support the design and operation of asynchronous production lines. 2019.

List of Figures

1 Graphical representation of the proposed algorithm. Picture taken from [5] 4
2 The manufacturing serial line. Picture taken from [5] 4
3 Throughput as a function of buffer capacity with first-order linearization. Picture taken from [5] 5

1.1 The Machining Department. Picture taken from [1] 7
1.2 Graphical representation of a multi-stage manufacturing system. Picture taken from [8] 10
1.3 The Building Block. Picture taken from [8] 10
1.4 Blocking and Starvation phenomena. Picture taken from [8] 11
1.5 Evaluation of the Situation "As is" 13
1.6 Evaluation of the Situation "To be" 13

2.1 Examples of ranking policies 16
2.2 Interchanges between optimization and performance evaluation 17
2.3 Graphical representation of the original algorithm. Picture taken from [5] . 18
2.4 Graphical representation of the new algorithm 19
2.5 The Objective Function 20
2.6 Interactions between sections 22
2.7 The Iterative cycle for the MTTRs update 24
2.8 The Performance Evaluation Model 27
2.9 Block Diagram of the Method 29

3.1 The Manufacturing System Modelled in Simulink 34
3.2 Maintenance Policy vs Throughput, @ N = N* 37
3.3 Maintenance Policy vs Buffer Slots, Th = th* 38
3.4 Buffer Slots vs Repair Rates 39
3.5 Number of Operators vs Repair Rates 40
3.6 Throughput vs Maintenance Policy 41
3.7 Machine Selection vs Throughput 41
3.8 Machine Selection vs Buffer Slots 42

3.9 Buffer Slots vs Throughput 43
3.10 Operators vs Throughput 44
3.11 Throughput vs Cost, @ Lot Size 10 45
3.12 Throughput vs Cost, @ Lot Size 60 46
3.13 Average Lot Size vs Cost 47

List of Tables

3.1 First-Stage Decision 31
3.2 The scenarios in analysis 32
3.3 Optimal Configurations 32
3.4 Optimal Configurations, Throughput 180 32
3.5 Optimal Configurations, Throughput 190 33
3.6 Optimal Configurations, Throughput 200 33
3.7 Sixth Simulink Check, TH=220 34
3.8 Optimal Configuration: Throughput 190, Average Lot Size 10 35
3.9 Exhaustive Reasearch, Considering a Single Policy 35
3.10 Exhaustive Research, Comparison Between Policies 36