Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

SIMULATION BASED OPTIMIZATION PACKAGE FOR PERIODIC REVIEW INVENTORY CONTROL

André F. Santos

Instituto Superior Técnico
Universidade de Lisboa
Avenida Rovisco Pais, 1
Lisbon, PORTUGAL

Carlos F. Bispo

Instituto de Sistemas e Robótica
Instituto Superior Técnico
Universidade de Lisboa
Avenida Rovisco Pais, 1
Lisbon, PORTUGAL

ABSTRACT

This paper presents a simulation based optimization package for periodic review Inventory Control, to study idling and non-idling policies. This package contains three different modules, a Simulink library for system design, an Infinitesimal Perturbation Analysis based simulator, and an optimization procedure based in the Davidon-Fletcher-Powell algorithm. We illustrate the usefulness of the package by means of presenting some preliminary numerical results for a three machine single product tandem flow line. The presented results allow us to extract some very interesting managerial insights and structural properties for multiple machine, multiple products inventory control. Namely, there are systems for which one can achieve significant performance gains by resorting to idling policies, and Local Base Stock policies may be better than Multi-Echelon Base Stock policies. Besides performance gains, by introducing idleness, we are also able to stabilize systems that under non-idling policies are unstable.

1 INTRODUCTION

In this paper we present a simulation based optimization package developed to study alternative inventory control policies. Inventory control policies address the problem of splitting a finite amount of capacity among a set of different products that need processing. In the context of periodic review inventory control, the base stock policy is known to be optimal for single product, single machine systems (capacitated or uncapacitated). Also, for uncapacitated machine flow lines it is known that the Multi-Echelon Base Stock policy – MEBSP – is optimal (see Clark and Scarf 1960). However, when machines are capacitated, little is known about the structure of the optimal inventory policy, except that the optimal final product inventory levels are bounded. Despite knowing this, there are no closed form results when it comes to determining the optimal policy nor these bound values. When using MEBSP one still has to address how to split capacity among the different products when their requirements are above the capacity value. Many authors have used variations of the MEBSP proposing different dynamic rules to address the capacity split. Examples are Priority, Linear Scaling, Equalize Shortfall, and Weighted Equalize Shortfall, among others. One particular feature of all these variations is the fact that they all are non-idling. The main objective of the work described in this paper is to use the developed package to study idling policies for periodic review inventory control and to establish a framework where they can be compared with the non-idling approaches.

The motivation to study idling policies for inventory control comes, mainly but not exclusively, from the queuing networks literature. Inspired by the TPS, (Ohno 1988), a significant amount of enterprises regulate their production decisions by using Kanban based policies. These are structurally idling policies, in the sense that we will define here.

978-1-5090-4486-3/16/$31.00 ©2016 IEEE

In Perkins and Kumar (1989), in the context of queuing networks, the authors propose a series of scheduling policies intended to be used for cyclic and acyclic systems. A cyclic system is such that there may be one or more products that have to visit a server more than once. When faced with the problem of proving the stability of the proposed policies for cyclic systems, the authors could only guarantee that every part entering the system, would eventually leave it. In order to present a stronger result regarding stability of cyclic systems, the authors proposed a modified version of the policies. This modified version included time gaps where the servers could be forced into idleness despite having available work to do. During the study of cyclic systems in the paradigm of inventory control, Bispo (1997) and Bispo (2013) had to include some sort of idleness to the production policies in order to ensure stability in the presence of random yield.

Studying the scheduling problem for large cyclic systems, Lu and Kumar (1991) give an example of a queuing network where a specific choice of job priorities will induce instability. Moreira and Bispo (2002) manage to stabilize a similar system using idling policies. Later, (Moreira and Bispo 2003), the authors also demonstrate that idling policies could perform better than non-idling policies in situations where stability is not an issue. More recently, Bispo (2012) used simulated annealing to optimize these policies and was able improve over the maximum pressure policies of Dai and Lin (2005) and Dai and Lin (2008).

Santos (2016) proposes a new class of policies that should work as a generalization of the MEBSP used by Glasserman and Tayur (1995), Bispo (1997), and Bispo and Tayur (2001). The Idling Multi-Echelon Base Stock Policy – IMEBSP – proposed adds one extra set of parameters to the MEBSP. This set bounds the production quantities at any given period. Under this policy, the system capability of recovering to the multi-echelon base stock levels, in a given decision period, will be limited not only by the upstream inventories and the machine capacity, but also by these production bounds. The MEBSP is a particular instance of the IMEBSP, where these production bound values are considered to be infinite.

This paper reviews the essential components of the work of Santos (2016) and has the objective of presenting the software package created, which will be made available as open source so that the research community may take advantage of it. This package allows to model, simulate, extract metrics, and optimize the proposed policies.

The simulator uses Infinitesimal Perturbation Analysis – IPA – to generate gradient information which is used to determine the optimal values for the policy parameters (see Glasserman 1991). IPA has already been used in the context of inventory control studies. In Bispo (1997), an optimization algorithm using the IPA derivatives information was applied to determine the optimal inventory stock levels for re-entrant systems. In our simulator, the IPA gradient information will also be used to find the optimal values for the production bounds.

The optimization algorithm which was used in this work was the Davidon-Fletcher-Powell – DFP (See Fletcher and Powell 1963). This algorithm is a well known quasi-Newton optimization method, for which some minor modifications had to be included. A brief description of the algorithm as well as the proposed modifications are discussed in Section 3.3.1.

In L'Ecuyer (1994), it is shown that variance reduction and increased optimization convergence rate are achieved by means of using one single sample path during the optimization process. In this paper, the optimization procedure was conducted over a single, long enough, sample path. The obtained optimal parameter values are tested over different sample paths and confidence intervals are calculated to validate the obtained results.

The presented paper is organized as follows. We start by introducing the theoretical model as well as the dynamic equations used to build the simulator and a brief discussion on Infinitesimal Perturbation Analysis, Section 2. In Section 3 we present the architecture of the simulation based optimization package as well as a description of its components. Section 4 presents some simulation results regarding the study of a three machine single product layout where IMEBSP performance is compared to the one of MEBSP. The paper ends, in Section 5 with some conclusions and references to future work.

2 THEORETICAL MODEL

The systems considered during the development of the software package were modelled in the context of periodic review inventory control. This section presents a brief description of the dynamic equations which were coded in the simulator.

The model presented here is an extension of the one introduced in Bispo (1997) and later used in Bispo and Tayur (2001). The new model contemplates a set of bounds capable of inducing idleness in the production polices. These bounds will directly impact equation (1) below.

A summarised description of the IPA approach will also be presented in this section.

2.1 Basic Model

Consider an eventually non-acyclic, multi-stage, multi-product, capacitated production system facing random demand. A capacitated production system is a system whose machines have a limited amount of capacity available at any point of time. Each one of the P products follows a specific routing pattern defined by a set O^p of operations. List of notation:

- P products (indexed by p),
- M machines (indexed by m),
- O^p: Set of operations for product p (indexed by k),
- J^m: Set of operations done by machine m (indexed by j),
- $(k)^+$: Denotes the operation immediately before k,
- $(k)^-$: Denotes the operation immediately after k,
- d_n^p: Demand for product p at period n (last operation - $k = 1$),
- z^{kp}: Echelon base stock level,
- Δ^{kp}: Alternative set of variables accounting for the inventory between stages,
- I_n^{kp}: Inventory in time period n for product p in operation k,
- E_n^{kp}: Echelon inventory at time period n for product p in operation k,
- Y_n^{kp}: Shortfall at time period n for product p in operation k,
- P_n^{kp}: Production amount at period n for product p in operation k,
- $\bar{I}^{kp}$: Forced bound to the production of p in operation k,
- C^m: Capacity of machine m,
- h^{kp}: Holding cost rate for product p in operation k, and
- b^p: Backlogging cost rate for product p.

Every set of operations O^p is indexed from K to 1 being K the first operation (most upstream) and 1 the last operation.

2.1.1 Inventory Dynamic Equation

For the sake of simplicity, in the following set of equations, I_n^{kp} refers to the inventory of operation $O^p(k)$ at the decision period n.

$$I_{n+1}^{kp} = \begin{cases} I_n^{1p} - d_n^p + P_n^{1p} & k = 1 \\ I_n^{kp} - P_n^{(k)^- p} + P_n^{kp} & \text{otherwise.} \end{cases}$$

2.1.2 Echelon Inventory Dynamic Equation

The dynamic evolution of the echelon inventory is described by the following equation:

$$E_{n+1}^{kp} = E_n^{kp} - d_n^p + P_n^{kp}.$$

The echelon inventory will grow with the amount of production at that corresponding stage and will decrease with the amount of products leaving the system.

2.1.3 Shortfall Dynamic Equation

The shortfall is defined as the difference between the echelon base stock and the echelon inventory

$$Y_n^{kp} = z^{kp} - E_n^{kp}.$$

It is possible to write a dynamic equation for the shortfall similar to the one for echelon inventory given by

$$Y_{n+1}^{kp} = Y_n^{kp} + d_n^p - P_n^{kp}.$$

2.1.4 Production net needs

Production net needs represent the production quantities that the system needs when there are no capacity bounds. In order to ensure that the capacity of the machines is not exceeded production rules will be applied to the production needs in a posterior step. The production net needs may be bounded by upstream inventory or by the forced bounds imposed to the system. When there are no bounds, the system will always try to produce the sufficient amount to take the Shortfall of the buffers to zero. The production net needs for a given product and operation are defined by:

$$f_n^{kp} = \begin{cases} \min\left\{ (z^{kp} + d_n^p - E_n^{Kp})^+, \bar{I}^{kp} \right\} & k = K \\ \min\left\{ (z^{kp} + d_n^p - E_n^{Kp})^+, I_n^{(k)^+ p}, \bar{I}^{kp} \right\} & \text{otherwise,} \end{cases} \tag{1}$$

where $(x)^+ = \max\{0, x\}$. In the expression above $\bar{I}^{kp}$ may be finite or infinite. If $\bar{I}^{kp} = \infty$ we have a non-idling policy, otherwise the policy may be idling. Note that the first machine of a production line will never be limited by inventory since raw material is assumed to be always available.

The echelon base stock variables must respect the following rule $z^{kp} \geq z^{(k)^- p}$. As discussed in (Bispo 1997), instead of continuously compare two consecutive multi-echelon base stock levels throughout the optimization procedure, it is preferable to use an alternative set of variables where this rule is simplified.

$$\Delta^{kp} = \begin{cases} z^{kp} & k = 1 \\ z^{kp} - z^{(k)^- p} & \text{otherwise.} \end{cases}$$

During the optimization procedure it is much easier to enforce that every $\Delta^{kp} \geq 0$ instead of making sure that the base stock variables are always non increasingly ordered.

2.1.5 Production Rules

The production rules solve the problem of dividing the total capacity available at a machine among a set of operations, when the sum of the production net needs exceeds this value. One possibility is for all products to get the same percent of production amount if not all can be produced. This is the Linear Scaling Rule – LSR.

For instance, assuming that every machine $m = 1, ..., M$ has a set of operations J^m, with cardinality N^m, the production decisions for all products visiting machine m under the LSR is established as follows:

$$P_n^{J^m(j)} = f_n^{J^m(j)} g_n^m \quad \text{for} \quad j = 1, 2, ..., N^m,$$

$$g_n^m = \min\left\{ \frac{C^m}{\sum_{j=1}^{N^m} \tau^{J^m(j)} f_n^{J^m(j)}}, 1 \right\}.$$

Note: $\tau^{(J^m(j))}$ is the capacity required to produce one unit of the jth operation performed by machine m.

Another possible production rule is the Priority Rule – PR, where the capacity is assigned according a given priority of the products as long as there is capacity to move to the lower priority product.

For every machine $m = 1,...,M$, its J^m is ordered so that $J^m(q)$ for $q = 1,...,N^m$, is the operation which comes in the qth position in the priority list. The production decision for machine m will be:

$$\begin{aligned}
P_n^{(J^m(1))} &= \min\left\{ f_n^{(J^m(1))}, \frac{C^m}{\tau^{(J^m(1))}} \right\}, \\
P_n^{(J^m(2))} &= \min\left\{ f_n^{(J^m(2))}, \frac{C^m - \tau^{(J^m(1))} P_n^{(J^m(1))}}{\tau^{(J^m(2))}} \right\}, \\
&\vdots \\
P_n^{(J^m(q))} &= \min\left\{ f_n^{(J^m(q))}, \frac{C^m - \sum_{j=1}^{q-1} \tau^{(J^m(j))} P_n^{(J^m(j))}}{\tau^{(J^m(q))}} \right\}.
\end{aligned}$$

2.1.6 Operational Cost

The operational cost refers to the cost of stocking inventory and being penalized by backlogs. Let a single stage cost be defined as

$$C_n = \sum_{p=1}^{P} C_n^p, \tag{2}$$

where C_n^p is given by

$$C_n^p = (I_n^{1p})^- b^p + (I_n^{1p})^+ h^{1p} + \sum_{k=2}^{K^p} I^{kp} h^{kp},$$

and K^p represents the number of operations from the set O^p.

With the single period cost established in (2), one is now capable of calculating the finite horizon average operational cost taking into account N simulated periods:

$$C_N = \frac{1}{N} \sum_{n=1}^{N} C_n. \tag{3}$$

2.2 Infinitesimal Perturbation Analysis

This section briefly reviews the basic framework of Infinitesimal Perturbation Analysis, (Glasserman 1991). To show validity of IPA it is necessary to demonstrate that the expected value and derivative are interchangeable operators. If one takes into account the finite horizon average cost equation presented in (3), if the system is simulated during an amount of decision periods long enough, so that it can cover in time the same amount information that it would cover for all sample paths, for stationary and ergodic demand, the following approximation is valid due to the Law of the Large Numbers:

$$C_N = \frac{1}{N} \sum_{n=1}^{N} C_n \to \bar{C}_N = \frac{1}{N} \sum_{n=1}^{N} E[C_n] \text{ for } N >>, \tag{4}$$

where $E[.]$ is the expected value over all sample paths.

The main result concerns the cost derivatives. When IPA is valid, the following relationship holds:

$$\frac{\partial}{\partial\theta}\bar{C}_N = \frac{1}{N}\sum_{n=1}^{N}\frac{\partial}{\partial\theta}E[C_n] = \frac{1}{N}\sum_{n=1}^{N}E[\frac{\partial}{\partial\theta}C_n].$$

For the same reasons as (4), it holds that:

$$\frac{\partial}{\partial\theta}C_N = \frac{1}{N}\sum_{n=1}^{N}\frac{\partial}{\partial\theta}C_n \rightarrow \frac{\partial}{\partial\theta}\bar{C}_N = \frac{1}{N}\sum_{n=1}^{N}E[\frac{\partial}{\partial\theta}C_n] \text{ for } N >> .$$

Therefore, we can get estimates of the local derivatives by simply using one long enough sample path. When IPA is not valid, and one wants to optimize a large number of parameters, the alternatives rely solely on multiple runs to obtain a single gradient approximation.

3 SOFTWARE PACKAGE

This section presents a description of the simulation package, which was created using Matlab and Simulink. The architecture of the package that was designed is presented in Figure 1.

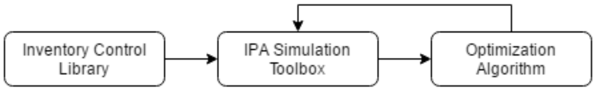

Figure 1: Architecture of the developed package.

The Inventory Control Library is a user friendly graphic interface with the purpose of simplifying the generation and editing of manufacturing layouts. The user designs a manufacturing layout using the Library subsystems, runs the Simulink model, and is provided with a routing matrix which has all the basic information needed to characterize the system. This routing matrix, as well as the mean demand for each product, the respective coefficient of variance, the seed that should be used to extract the demand path, and the system initial conditions are the inputs of the IPA Simulation Toolbox. It should be stressed that the connection between the Inventory Control Library and the IPA Simulation Toolbox is automatic. The most difficult task when setting up an IPA simulator is the codding of the specific equations for a system one wants to study. By constructing a generic IPA Simulation Toolbox that receives the specific information automatically from a graphic interface, this difficulty is significantly reduced. The IPA Simulation Toolbox uses the input information to simulate the system operation throughout an amount of decision periods defined by the user. At the end of the simulation the Toolbox provides the value of the mean operation cost as well as its derivative with respect to all control policy parameters. This information is then processed by the Optimization Algorithm which will calculate the next initial conditions to be used by the IPA Simulation Toolbox. This is an iterative process that will end once the optimization procedure reaches a stopping criteria.

The package is available at (Santos and Bispo 2016). Every module of the package is independent from the others so that any user may have the freedom of using a different optimization algorithm, changing the Simulation Toolbox, creating new production rules or simply use the Inventory Control Library to create routing matrices for a simulation process of his/her choosing. The package has a companion user manual.

3.1 Inventory Control Library

The Inventory Control Library takes advantage of the core capabilities of Simulink to ease the procedure of turning any user created layout into routing matrices capable of being read by a simulator.

The library is composed by three different subsystems, a Raw Material input, the Machine block and a Sink – Figure 2.

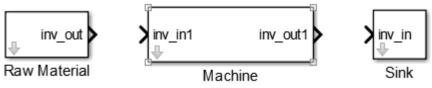

Figure 2: Inventory Control Library subsystems.

Raw Material and Sink mark the beginning and the end of a production line respectively. The Raw Material block will hold the identification of the product correspondent to the flow line that it starts, while the Sink will record the value of the penalty cost for backlogging deliveries.

The machine block is the most complex of the three. This block holds the information regarding its capacity, product priorities (if the production rule is priority based, else these priority values will be ignored), the capacity required to produce one unit of a certain product (τ), and the holding costs for inventory stocking.

The machine block was prepared to dynamically change the number of inputs and outputs so that the layouts could be kept with an elegant presentation. In Figure 3 we present a two machine single product re-entrant flow line.

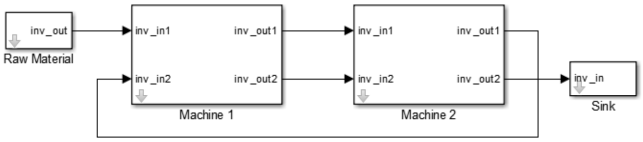

Figure 3: Example of a cyclic system implemented with the library.

The output of the Simulink run is, as already mentioned, a routing matrix for the system that will be made available in the Matlab Workspace.

3.2 IPA Simulation Toolbox

The IPA Simulation Toolbox includes four different kinds of functionalities which were fundamental for the study of the idling policies and their evaluation. These functionalities are separated in the functions that will be described in this section.

The first function is responsible for reading the information regarding the initial conditions of the system as well as the routing matrix from the Inventory Control Library. This function computes all the initial variables needed to run the simulations including the demand values for every simulation period. The demand is calculated using gamma distributions and random numbers generated from a seed provided by the user. The user must provide the mean demand rate and coefficient of variance for each of the products.

The second function is the IPA Simulator. This function uses initial conditions arranged by the previous function and computes the IPA simulation for a number of decision periods defined by the user. In order to improve the computational efficiency of the IPA Simulator, this function was migrated to the C Language. The run times obtained by this migration is typically one hundredth of the original. The simulator must also be provided with the identification of what dynamic production rule should be used. The current version of the package has two pre-programmed rules which are identified as follows:

- Rule 1 – Priority Rule, and
- Rule 2 – Linear Scale Rule.

For future developments, the code is prepared to be merged with additional rules.

After the execution of the simulation the user will be provided with the operation mean cost as well as its derivatives with respect to the policy parameters.

During the optimization procedures some steps do not need derivative information. A more time efficient version of the simulator was created where the derivatives are not calculated. This version of the simulator was used in some optimization steps like a line search using Golden Section. This function may be useful in future work for alternative optimization algorithms, such as, simulated annealing where derivative values are typically not used.

The last functionality is the option to extract information regarding the amount of periods that each machine does not use its full capacity and the inventory tracking. The inventory tracking provides the user with the inventory levels for each buffer at every decision period. This information helps the comparative analysis of the systems behaviour when the production policy changes.

3.3 Optimization Procedure

In the presence of a given system, the optimization procedure should provide the user with the information of what are the optimal parameter values to be used in order to achieve the minimal operational cost.

The optimization procedure was meant to operate over two different families of parameters, the Δ and the $\bar{I}$ variables. Given that these two families of parameters have different behaviour and would be subjected to different constraints, we chose to separate the optimization procedures and let them work in a cyclic manner, one cycle for $\bar{I}$ followed by a cycle for the Δ. After every cycle the difference between the two achieved costs is compared with a stopping criteria (See Figure 4).

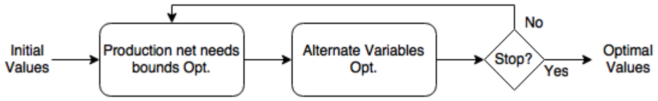

Figure 4: Cyclic optimization scheme.

Both optimization procedures use the Davidon-Fletcher-Powell Algorithm – DFP. The DFP algorithm is a quasi-Newton method (See (Fletcher and Powell 1963)), that uses the gradient information from the previous iterations to make corrections to the new descent direction in order to improve convergence speed. This method uses an estimation of the inverse Hessian matrix, which is continuously updated at every step of the algorithm. This inverse Hessian matrix is used, as well as the IPA derivatives, to calculate the direction that the algorithm should follow in the optimization step. Both optimization procedures follow a structure similar to the one presented in Figure 5

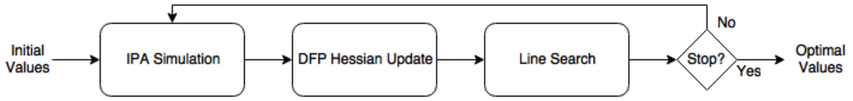

Figure 5: Optimization procedure scheme.

The line search algorithm, in a first approach, finds an interval along the descent direction which contains the minimum, while enforcing the parameter constraints. For example, the Δ values are never allowed to become negative, because that would mean negative inventory levels. A golden section search (Luenberger 1973) is used in this interval in order to find the minimum. Given the fact that the Golden section search does not require gradient information, it uses the functionality of the IPA simulation that does not compute the gradient information.

3.3.1 Optimization Challenges

The DFP algorithm was originally designed for unconstrained problems. In our setting, there are constraints for the parameters. Namely, all Δ parameters have to be non negative and all $\bar{I}$ parameters have to be above the average demand. In order to apply DFP under these circumstances, we had to change the algorithm.

Every time the algorithm would point to a zone out of the constraints, the inverse Hessian matrix would be replaced by an identity matrix and the DFP algorithm would restart from the boundary point.

Note that the original DFP algorithm replaces the Hessian inverse estimate by the identity matrix every K iterations, where K the number of variables being optimized. When the estimate of the Hessian inverse is replaced by the identity matrix, the next iteration is a steepest descent step. This is what the Global Convergence Theorem requires in order to guarantee that the given optimization procedure will converge to the optimal solution. Therefore, despite the alteration of the original algorithm refresh rate, the Global Convergence Theorem is still applicable.

If, after the reset, the DFP would still point to a zone where a parameter would violate a constraint, this parameter is set to the boundary of the constraint, its derivative is masked, and the optimization procedure continues only operating over the other parameters.

4 NUMERICAL RESULTS

In (Santos 2016) an extensive numerical study is presented for a handful of different systems. In this paper we present a sample of those results for one single system. The system that we will present here is a three machine single product tandem. We will show how the IMEBSP will regulate the production on stages upstream of the bottleneck and compare those results with the MEBSP. Results regarding the confidence intervals of the optimized parameters will not be presented in this paper.

Figure 6 displays the layout used for this set of experiments.

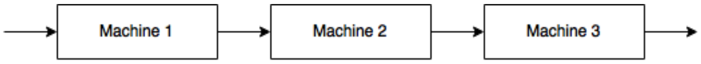

Figure 6: Three state tandem configuration.

Since this is a single product setup, the results obtained while testing with the Priority Rule or the Linear Scaling Rule will be the same. This is due to the fact that each machine capacity does not have to be split among different products. This section presents a study on how the bottleneck positioning along the production line, together with the choice of the policy (IMEBSP or MEBSP), will impact the operational costs as well as the overall variance of the stock levels in the buffers. The tests were conducted using the holding cost setup presented in Table 1.

Table 1: Holding cost structure.

	h^{31}	h^{21}	h^{11}
Cost	2	4	10

Penalty cost for backlogging orders was set to 20.

Throughout these experiments, the machine set as the bottleneck is always working at 80% load, while the other two would operate at 60%. To this end the bottleneck machine has a capacity of 15 and the other machines 20, while the mean demand was set to 12 with a coefficient of variance of 1.

Table 2 presents the average operational costs results for the tandem layout achieved under the studied policies varying the position of the system bottleneck. Looking at the table, one can conclude that the IMEBSP will show larger improvement margins when compared with MEBSP if the bottleneck is located in the most downstream position of the production line. Moreover, the IMEBSP beats the MEBSP in every setup, except when the bottleneck of the tandem line is located at its most upstream machine. In this case the IMEBSP achieves the same performance as its counterpart.

In Figure 7 it is possible to establish a relationship between the bottleneck position and which bounds converge to the value of the bottleneck. As observed from this experiment, one can conclude that all the machines positioned upstream of the bottleneck see their production decisions bounded by the bottleneck capacity.

Table 2: Cost benchmark between the studied policies.

Bottleneck Machine	IMEBSP	MEBSP	Cost Improvement
Machine 1	407.80	407.80	0%
Machine 2	440.16	472.10	6.8%
Machine 3	468,43	528.41	11.4%

Looking at the Δ optimal values, we can conclude that all the Δ variables corresponding to operations upstream of the bottleneck will also converge to the bottleneck capacity value.

Every machine operating under the conditions where both its bound value and its Δ value converged to the bottleneck capacity value is in fact working under a Local Base Stock policy and, in these cases, their inventory buffers have zero variance.

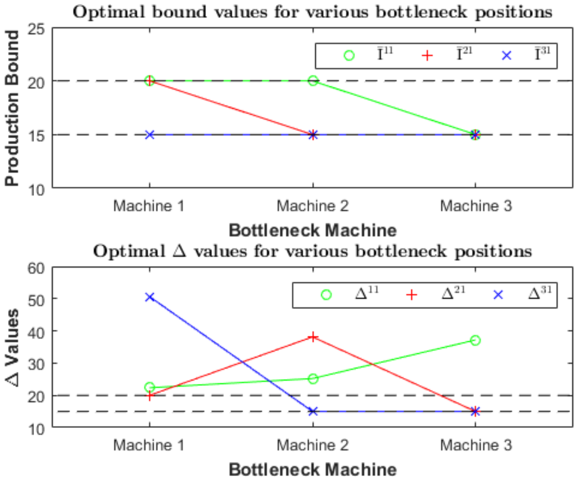

Figure 7: Optimal production bound and Δ values.

When the bottleneck is machine 3, the inventory paths of machine 1 and 2 suffer a radical change when we move from MEBSP to IMEBSP. As a sample, in Figure 8, we display the inventory paths for machine 2. Given the discussion above, the inventory level of the machine under IMEBSP, presented on the top of Figure 8, does not change during the simulation (zero variance), which contrasts with the inventory path under MEBSP on the bottom of this figure. For IMEBSP the inventory level not only does not change but it keeps equal to its Δ parameter as well as its $\bar{I}$. Given that the bounds for machine 1 and 2 converged to the value of the bottleneck capacity, these machines will have forced idleness in every period.

Analysing the obtained results, one can conclude that the introduction of forced idleness in this system will convert into operational cost improvements. Also, these improvements will be higher when the bottleneck is positioned at the end of the production line. Table 3 illustrates how this performance improvement is achieved by means of a variance reduction on the inventory paths. Intuitively, the production bounds are filtering the burstiness of the external demand process.

Table 3: Inventory buffer variance.

Inventory Variance	IMEBSP	MEBSP
I^{11}	33.80	33.80
I^{21}	0	24.01
I^{31}	0	2.33

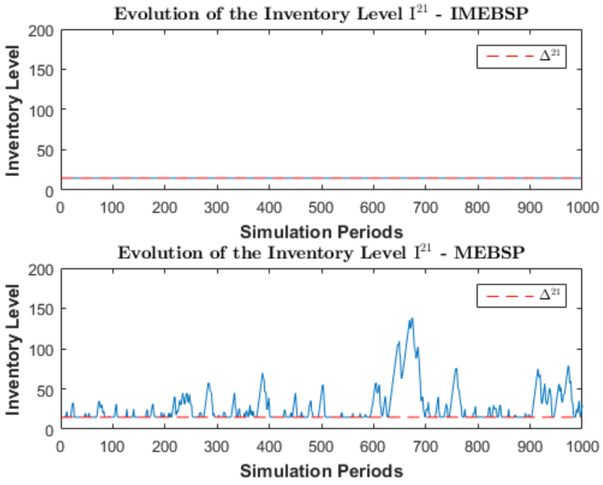

Figure 8: Inventory path for I^{21}.

5 CONCLUSIONS

We presented the simulation based optimization tool proposed for the study of periodic review Inventory Control policies. It should be stressed that all the modules of the package may work independently and will be available as an open source application. A software manual will be available as well. The tool here presented is sufficiently general in terms of the type of systems one may want to model: multiple machines; multiple products; cyclic, acyclic, and non-acyclic flows. The definition of the system to be studied does not require the user to code any equations. However, it is possible to add different production rules as long as in the context of IMEBSP. Note that MEBSP is a particular instance of IMEBSP. Only if one wants to add a different rule will coding be necessary.

The numerical results presented are very encouraging in the sense that significant performance improvements are achieved by means of idling policies. Also, the study allows us to identify some very interesting structural properties on the optimal solutions. We expect these preliminary results to trigger interest from the research community.

Another interesting feature of the results is the fact that the performance improvements are a consequence of an overall reduction on the inventory buffer variances. We also observed a shift in the policies, imposed by the production bounds. The optimal policy for all the machines operating upstream of the bottleneck becomes Local Base Stock, which carries a very interesting connection with the Kanban Based Policies (See (Ohno 1988)).

The validation of the IPA approach was out of the scope of the present paper. This is an issue that will have to be addressed in the future. Also, we assumed no fluctuation on machine capacity, like random yield or failures. Such features will require further work, as well as more appropriate policies to address the challenges posed by them.

ACKNOWLEDGMENTS

This work was partially supported by the Fundação de Ciência e Tecnologia Project UID/EEA/50009/2013.

REFERENCES

Bispo, C. F. 1997, 9. *Re-Entrant Flow Lines*. Ph. D. thesis, Carnegie Mellon University, Pittsburgh, USA.

Bispo, C. F. 2012. “Active Idleness Scheduler with Simulated Annealing”. In *Information Control Problems in Manufacturing*, Volume 14, 1425–1432.

Bispo, C. F. 2013. "Stability of Re-Entrant Flow Lines". In *OR 2013 Conference - The International Conference on Operations Research*. Rotterdam, the Nederlands.

Bispo, C. F., and S. Tayur. 2001. "Managing Simple Re-Entrant Flow Lines: Theoretical Foundation and Experimental Results". *IIE Transactions* 33 (8): 609–623.

Clark, A. J., and H. Scarf. 1960. "Optimal policies for a Multi-Echelon Inventory Problem". *Management science* 6 (4): 475–490.

Dai, J. G., and W. Lin. 2005. "Maximum Pressure Policies in Stochastic Processing Networks". *Operations Research* 53 (2): 197–218.

Dai, J. G., and W. Lin. 2008. "Asymptotic Optimality of Maximum Pressure Policies in Stochastic Processing Networks". *The Annals of Applied Probability* 18 (6): 2239–2299.

Fletcher, R., and M. J. Powell. 1963. "A Rapidly Convergent Descent Method for Minimization". *The Computer Journal* 6 (2): 163–168.

Glasserman, P. 1991. *Gradient Estimation Via Perturbation Analysis*. Springer Science & Business Media.

Glasserman, P., and S. Tayur. 1995. "Sensitivity Analysis for Base-Stock Levels in Multiechelon Production-Inventory Systems". *Management Science* 41 (2): 263–281.

L'Ecuyer, P. 1994. "Efficiency Improvement and Variance Reduction". In *Proceedings of the 1994 Winter Simulation Conference*, edited by J. Tew, S. Manivannan, D. Sadowski, and A. Seila, WSC '94, 122–132. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Lu, S. H., and P. R. Kumar. 1991. "Distributed Scheduling Based on Due Dates and Buffer Priorities". *IEEE Transactions on Automatic Control* 36 (12): 1406–1416.

Luenberger, D. G. 1973. *Introduction to Linear and Nonlinear Programming*, Volume 28. Addison-Wesley Reading, MA.

Moreira, J., and C. F. Bispo. 2002. "Stability or Stabilizability? Seidman's FCFS Example Revisited". In *10th Mediterranean Conference on Control and Automation*. Lisbon, Portugal.

Moreira, J., and C. F. Bispo. 2003. "Performance improvement through active idleness". In *11th Mediterranean Conference on Control and Automation*. Rhoddes, Greece.

Ohno, T. 1988. *Toyota Production System: Beyond Large-Scale Production*. CRC Press.

Perkins, J., and P. R. Kumar. 1989. "Stable Distributed, Real-Time Scheduling Of Flexible Manufacturing / Assembly / Disassembly Systems". *IEEE Transactions on Automatic Control* 34:139–148.

Santos, A. F. 2016. "Idling Policies for Periodic Review Inventory Control". Master's thesis, Instituto Superior Técnico, Lisbon, Portugal.

Santos, A. F. and Bispo, C. F. 2016. "IMEBSP Software Package". Accessed Jul. 14, 2016. http://users.isr.tecnico.ulisboa.pt/~cfb/IMEBSP/.

AUTHOR BIOGRAPHIES

ANDRÉ F. SANTOS holds a MSc (2016) in ECE from the Instituto Superior Técnico – Universidade de Lisboa, Portugal. He specialized his studies in Systems, Decision and Control, focusing in Manufacturing Applications. His email is andre.freixo.santos@gmail.com.

CARLOS F. BISPO holds a first degree (1985) and a MSc (1988) in ECE from the Instituto Superior Técnico – Universidade de Lisboa, Portugal, and a MSc (1993) in Industrial Administration and a PhD (1997) in Management of Manufacturing Automation from the Carnegie Mellon University, Pittsburgh, PA, USA. He is currently an Assistant Professor in Electrical and Computer Engineering at the Instituto Superior Técnico – Universidade de Lisboa, Portugal. His main research interests are Inventory Control, Distributed Scheduling for Queuing Networks, Markov Decision Processes, and Simulation Based Optimization. His email address is carlos.bispo@tecnico.ulisboa.pt.