A Hybrid Method for Solving Buffer Sizing and Inspection Stations Allocation

Mohamed Ouzineb, Fatima-Zahra Mhada, Robert Pellerin, Issmail El Hallaoui

A hybrid method for solving buffer sizing and inspection stations allocation

Mohamed Ouzineb ^{a*}, Fatima-Zahra Mhada^{b}, Robert Pellerin^{c}, and Issmail El Hallaoui^{c}

^{a} Institut National de Statistique et d'Economie Appliquée and GERAD, B.P.:6217 Rabat-Instituts, Madinat AL Irfane, Rabat, Morocco
^{b} University Mohammed V- Souissi/ENSIAS, B.P. 713, Rabat-Instituts, Madinat Al Irfane, Rabat, Morocco.
^{c} Ecole Polytechnique de Montréal and GERAD, CP 6079 succ Centre-Ville, Montreal, Qc, Canada H3C 3A9
{mohamed.ouzineb@gerad.ca,f.mhada@um5s.net.ma,
robert.pellerin@polymtl.ca,issmail.el-hallaoui@polymtl.ca}

Abstract. The buffer sizing problem in unreliable heterogeneous production lines is an important, indeed, complex combinatorial optimization problem with many industrial applications. These applications include quality, logistics and manufacturing production systems. In the formulation of the problem, the system consists of n machines, n fixed-size buffers and m inspection station in series. The objective is to minimize a combined storage and shortage costs, and also specifying the optimal location of inspection stations in the system. The present paper aims at optimizing a generalization of the model previously proposed in (Mhada, Malhamé, & Pellerin, 2014) using a novel approach. In this approach, we combine Tabu Search (TS) and Genetic Algorithm (GA) to identify search regions with promising locations of inspection stations and an exact method to optimize the assignment of buffer sizes for each location. This approach provides a balance between diversification and intensification. Numerical results on test problems from previous research are reported. Using this approach, we can reduce the solution time by more than 97% in some cases.

Keywords: Inspection, Production lines, Quality, Combinatorial Optimization, Meta-heuristics

1 Introduction

Apart from rare exceptions ((Kim & Gershwin, 2005, 2008), (Colledani & Tolio, 2005, 2006a, 2006b, 2011)), the current and standard quality analysis models tend to consider separately the problem of developing strategies for the quality preservation in production lines (positioning stations inspection for example) and the development of strategies to optimize the production problem (kanban, CONWIP, or others). Yet as noted it so well ((Kim & Gershwin, 2005, 2008) and (Colledani & Tolio, 2005, 2006a, 2006b, 2011)), both whole decision are interdependent. We illustrate this interdependence through the example of the analysis of kanban strategies introduced by Toyota in the sixties and have since become the paradigm of "lean manufacturing": they essentially advocate areas of limited storage between successive machines in a production line. The objective of these storage areas is to allow certain degree of decoupling between machines to increase productivity of the line (limitation of the effects of parts shortage for downstream machines or blocking machines upstream a broken down one). These stocks should be limited because they are associated with a frozen capital, a storage costs and a parts extended transit times in the workshop.

In a "just in time" ideal, there would be no intermediate stock and finished parts would be pulled from the system as they are produced. However, hazards, such machinery breakdowns and lack of raw material or operators, will come to question this idealized vision, unable to ensure a sufficient continuity of the workshop output. The sizes of the storage areas are directly related to the statistics of these hazards, as well as the estimated costs of the service losses that may result.

So if, we choose to size the kanban by ignoring totally the production quality dimension in the

---

problem, we risk likely to overestimate the security service level with a storage areas which may contain important quantities of defective parts. These defective parts play a negative role for at least two reasons: on one hand they correspond to a misused production line time, since they decrease the efficiency of individual machines; they systematically undermine the "efficiency" of the intermediate storage for its ability to help increase the productivity of the line. On the other hand, as the investments associated with production line come from the same source, the costs associated with storage within the line are going to burden the budget associated with the improvement of the quality and vice versa. (Inman, Blumenfeld, Huang, & Li, 2013) present a survey of recent advances on the interface between quality and production systems design, they provide evidence that production design impact quality and quality impact production design. Also the location of the inspection station affects both the expected production cost per item and the production rate of the line.

They are many research (see (Mandroli, Shrivastava, & Ding, 2006)) that focus in determining the optimal inspection station position in n serial production lines with or without: i) scrapping, ii) rework, iii) off-line repair, without considering the concept of buffer sizing.

Like the work of (Kim & Gershwin, 2005, 2008), we propose to work with a continuous production models (fluid models). However, the analogy with (Kim & Gershwin, 2005, 2008) stops there. In (Kim & Gershwin, 2005, 2008), although the models are continuous, quality, remains attached to a discrete part and we see this as a potential source of contradictions in the modeling.

In our case, we consider a tandem machines where every machine has to satisfy a demand rate of good parts per time unit. We consider the problem of sizing the inventory level taking into account that the stock is mix of good and defective parts; we assume at first all outstanding in the production line continuously observed. Here, the modeling becomes more complex given that there is generally elimination of defective parts along the line by positioning a number of inspection stations. A steady effort was made to develop methods of decomposition, to reduce the analysis of the line to a series of equivalent machine that can be isolated and sequentially analyzed (Mhada et al., 2014).

The present paper efficiently solves a generalization of the model proposed in (Mhada et al., 2014) where the question of the optimal position in the production line of a number m of inspection stations will be treated (m also must be optimal). We combine Tabu Search (TS) and Genetic Algorithm (GA) to identify search regions with promising locations of inspection stations and an exact method to optimize the assignment of buffer sizes for each given location. Hybridizing TS and GA has been proven to be very effective for high-dimensional combinatorial optimization problems. Actually, when the number of solutions is huge, which is the case in this paper, this step aims at locating promising search regions. This approach provides a balance between diversification and intensification. First, the selection of solutions by GA allows the identification of promising regions in the search space. One intended role of using GA is to facilitate the exploration by guiding the search to unvisited regions with good potential. This leads to a certain diversification in terms of subspaces to explore. Second, TS intends to search intensively around good solutions found in the past search. This leads to intensification by exploiting the neighborhood of each solution selected by GA.

The exact method of (Ouzineb, Mhada, El Hallaoui, & Pellerin, 2013) has been shown to be efficient when dealing with the small instances of buffer sizing and inspection stations allocation. This approach is an Exhaustive Search Method (ESM) because the search is guaranteed to generate all possible locations of inspection stations. For each fixed location, the problem is reformulated as a network flow optimization problem that can be efficiently solved by a fast polynomial algorithm. Actually, the space that will be searched is finite by very large and the exhaustive search method may need hours for large instances. The approach proposed in this paper finds near optimal solutions in a fraction of time compared to the exhaustive method proposed in (Ouzineb et al., 2013).

The remainder of the paper is organized as follows. In Section 2, the problem statement is presented. A short description of the proposed approach is given in Section 3. Numerical results are reported in Section 4 and some conclusions are drawn in Section 5.

2   Problem Formulation

Figure 1 represents the production system studies, it consists of n machines separated by $n-1$ buffers and contain (m+1) inspection stations. Machines can be either up or down, starved or blocked. A machine M_i is starved if one of the upstream machines is down and all buffers between this machine and the machine M_i are empty. M_i is blocked if one of the downstream machines is down and all buffer between this machine and the machine M_i are full. When a machine is neither starved or blocked and operational, it transfer parts from the upstream buffer to the downstream in a continuous way. We assume that the first machine can never be starved and an inspection station is located after the last machine to ensure the conformity of parts received by the customer.

All machines $M_i, i = 1...n$ could be modeled as a continuous time Markov chain that produce a
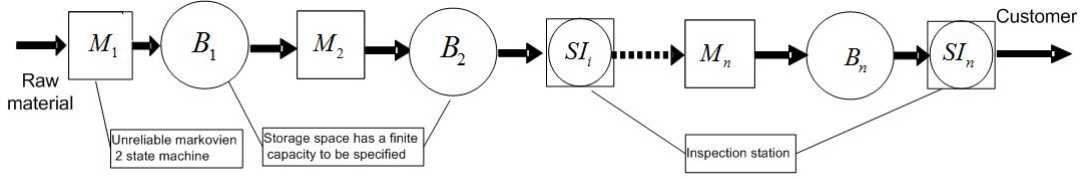

Fig. 1: The production line

single part type with two different quality levels: conforming and non conforming with the same predefined ratio of non-conforming parts to conforming parts β. We consider that all the machines have the same maximal production rate k, failure rate p and repair rate r.

We denote d the demand rate for good parts from the last buffer, x_i the inventory level on the buffer i, $\tilde{d}_i$ the long term average number of parts pulled per unit time from the stock x_i, c_p the storage cost per time unit and per part, c_I the inspection cost per pulled part, and a_{des}^n the required conforming finished parts availability rate.

A binary variable λ_i determine whether or not there is a station before the machine $M_{(i+1)}$ to ensure that all the parts be processed by $M_{(i+1)}$ are conform. We suppose that m inspection stations are dispersed along the line (m is not known) i.e. $\sum_{i=1}^{n-1} \lambda_i = m, \ m \in \{0, 1, ..., (n-1)\}$ and $\lambda_n = 1$.

The optimization problem is the minimizing of the long term per unit time average global cost of storage , production shortages, and inspection. Meaning, the cost to be minimized is:

$$J_{T(\tilde{d}_i, \lambda_i)} = \lim_{T \to \infty} \frac{1}{T} \sum_{i=1}^{n} E[\int_0^T (c_p\, x_i(t) + c_I\, \tilde{d}_i\, \lambda_i) dt] \tag{1}$$

under conditions:
$$\sum_{i=1}^{n-1} \lambda_i = m, \ m \in \{0, 1, ..., (n-1)\} \text{ and } \lambda_n = 1$$

The average long term combined storage, shortage costs and inspection costs in (1) expressions can be written after development and calculation (see article Mhada et al. (2014)) as:

$$J(a,\lambda) \qquad = \qquad \sum_{i=1}^{n-1} T^{(i)}(a,\lambda) \qquad + \qquad T_F(a,\lambda) \qquad + \qquad c_I \sum_{i=1}^{n-1} \lambda_i\, \tilde{d}_i \tag{2}$$

$$T^{(i)}(a,\lambda) = c_p \left(\frac{k\, \frac{(r(1-a_{i-1})+p)}{a_{i-1}}}{\sigma_i(k - \frac{\tilde{d}_i}{a_i})\, \frac{(r+p)}{a_{i-1}}} - \frac{k\,(1-a_i)}{\sigma_i(k - \frac{\tilde{d}_i}{a_i})} - \left[\frac{1}{\sigma_i} - \frac{(1-a_i)\,\frac{(r+p)}{a_{i-1}}}{\sigma_i^2(k - \frac{\tilde{d}_i}{a_i})} \right] \right.$$
$$\left. ln\left[\frac{\frac{(r(1-a_{i-1})+p)}{a_{i-1}}\, \frac{\tilde{d}_i}{a_i}}{r\,(k - \frac{\tilde{d}_i}{a_i})} - \frac{\sigma_i\, \frac{(r(1-a_{i-1})+p)}{a_{i-1}}\, \frac{\tilde{d}_i}{a_i}}{\frac{(r+p)}{a_{i-1}}\, r\,(1-a_i)} \right] \right), \ i = 1, ..., n-1 \tag{3}$$

$$T_F(a,\lambda) = \frac{\rho_n\, c_p\,(\frac{k_n\,(1-\exp(-\mu_n(1-\rho_n)\,z_n(a_n^{des})))}{1-\rho_n} - \frac{(r+p)}{a_{n-1}} z_n(a_n^{des})\exp(-\mu_n(1-\rho_n)\,z_n(a_n^{des})))}{\frac{(r+p)}{a_{n-1}}(1-\rho_n\,\exp(-\mu_n(1-\rho_n)\,z_n(a_n^{des})))} \tag{4}$$

with: $\sigma_i = \dfrac{(\frac{(r+p)}{a_{i-1}})\frac{\tilde{d}_i}{a_i}-k\,r}{(k-\frac{\tilde{d}_i}{a_i})\frac{\tilde{d}_i}{a_i}}$, $\rho_n = \dfrac{r(k-\frac{\tilde{d}_n}{a_n^{des}})}{\frac{(r(1-a_{n-1})+p)}{a_{n-1}}\frac{\tilde{d}_n}{a_n^{des}}}$, $\mu_n = \dfrac{(r(1-a_{n-1})+p)}{a_{n-1}(k-\frac{\tilde{d}_n}{a_n^{des}})}$ and

$z_n(a_n^{des}) = -\dfrac{\ln\left[\frac{1}{\rho_n}\left(1-\frac{(1-\rho_n)}{(1-a_n^{des})(\frac{(r+p)}{(r(1-a_{n-1})+p)})}\right)\right]}{\mu_n(1-\rho_n)}$

- $a_i, i = 1...n-1$ is the total parts wip availability coefficient at buffer x_i ; this coefficient has a lower and an upper bounds (Sadr & Malhamé, 2004):
  The upper bound : the minimum of a_i is achieved when the levels of intermediate stocks are at zero, in this case a_i is therefore equal to $a_{i-1}\frac{r}{(r+p)}$. i.e. $a_i > a_{i-1}\frac{r}{(r+p)}$. Also a_i must be smaller than 1. So $a_{i-1} < min\left(\frac{a_i(r+p)}{r}, 1\right)$

  The lower bound : a_{i-1} must guarantee the feasibility of demand $\tilde{d}_i$ by M_i, i.e $\tilde{d}_i < a_{i-1}k\frac{r}{(r+p)}$ then $\frac{(r+p)\tilde{d}_i}{r\,k} < a_{i-1}$. Also $a_i > a_{i-1}\frac{r}{(r+p)}$ and we suppose that $a_0 = 1$, then $a_{i-1} > \left(\frac{r}{(r+p)}\right)^{i-1}$. So : $a_{i-1} > max\left[\left(\frac{r}{(r+p)}\right)^{i-1}, \frac{(r+p)\tilde{d}_i}{r\,k}\right]$.

- $\tilde{d}_i, i = 1...n$ is the long term average number of parts pulled per unit time from stock x_i. The $\tilde{d}_i, i = 1...n$ expression depends on the selected position along the line of the m inspection stations. So if $e_j, j = 1...m$ the inspection station's position, i.e.

$$\lambda_i = \begin{Bmatrix} 1 \text{ if } i = e_j,\ j=1...m \\ 0 \text{ if not} \end{Bmatrix} \tag{5}$$

then:

$$\tilde{d}_i = \begin{cases} d(1+\beta)^n & \text{if} \quad 1 \le i \le e_1 \\ d(1+\beta)^{n-e_1} & \text{if} \quad e_1 < i \le e_2 \\ . & . \\ . & . \\ d(1+\beta)^{n-e_{m-1}} & \text{if} \quad e_{m-1} < i \le e_m \\ d(1+\beta)^{n-e_m} & \text{if} \quad e_m < i \le n \end{cases} \tag{6}$$

The problem can be formulated as follows: find the minimal average global cost system structure (a, λ, m) that satisfies both constraints. That is,

minimize $\quad J(a, \lambda, m)$ $\hfill (7)$

subject to

$$\sum_{i=1}^{n-1} \lambda_i = m \qquad \forall m,\ 1 \le m \le n-1, \tag{8}$$

$$max\left[\left(\frac{r}{(r+p)}\right)^{i-1}, \frac{(r+p)\tilde{d}_i}{r\,k}\right] < a_{i-1} \le min\left[\frac{a_i(r+p)}{r}, 1\right], \forall i,\ 1 \le i \le n, \tag{9}$$

$$\lambda_n = 1 \ \text{ and } \ a_n = a_n^{des}, \tag{10}$$

$$\lambda_i \in \{0,1\} \qquad \forall i,\ 1 \le i \le n-1. \tag{11}$$

$$a_i \ge 0 \qquad \forall i,\ 1 \le i \le n, \tag{12}$$

This problem is a Mixed Integer Nonlinear Programming (MINLP) problem which refers to a mathematical programming with continuous and discrete variables and nonlinearities in the objective function (7) and constraints ((8), (9), (10), (11), and (12)). This problem simultaneously optimizes the system structure: the location of the inspection stations (discrete) and parameters and the buffer sizing (continuous).

The variables $a_i, 1 \le i \le n$, $\lambda_i \in \{0,1\}\, 1 \le i \le n-1$ and m are the decision variables.

The constraint (10) represents the final conditions set by the customer.

The constraint (9) is a nonlinear constraint function which represents the line configuration, it is a cumulative constraint type that represents a guarantee the functioning of the production line. The vector of binary variables ((8),(11)) is connected to the line structure.

3   Solution methodology

As mentioned earlier, the objective function to minimize is the average long term combined storage and shortage costs, while also specifying the optimal location of inspection stations that meets a specified final condition set by the customer. The goal is to solve the optimization problem given by Eqs. (7-12) for a large test problem with 20 machines.

The exact method of (Ouzineb et al., 2013) has been shown to be efficient when dealing with the small instances of buffer sizing and inspection stations allocation. This approach is an Exhaustive Search Method (ESM) because the search is guaranteed to generate all possible locations of inspection stations.

This approach is an Exhaustive Search Method (ESM) because the search is guaranteed to .

It is clear that while the exhaustive exact method (ESM) of (Ouzineb et al., 2013) finds optimal solutions to small instances in reasonable times, the running time appears to be unreasonable for larger ones. In fact, ESM generate all possible locations of the inspection stations.

If we especially increase the number of machines, this method becomes drastically inefficient. The number of possible locations of the inspection stations, which is exponential, explodes. It is observed that the large amount of these locations are not profitable. For solving the combinatorial optimization problem formulated in 7-12, it is important to have an effective and fast procedure to smartly locate the extra m inspection stations. To do so, we propose a combination of Genetic Algorithm (GA) and Tabu Search (TS), which has been proven to be very effective for high-dimensional combinatorial optimization problems.

For each fixed location, we use a network flow problem, that can be efficiently solved as the standard shortest path problem to find the optimal assignment of buffer sizes. Because of such a combination, this method is said to be hybrid. Hereunder, we will refer to this hybrid method as EM/TG, EM and TG being acronyms of Exact Method and Tabu-Genetic, respectively. This approach finds optimal or near optimal solutions to the same instances in a fraction of time compared to the exhaustive method proposed in (Ouzineb et al., 2013) as will be shown by the experiments. The hybrid method is presented in details in the next section.

3.1   Hybrid optimization method

The genetic algorithm used in this paper is based on the method commonly known as the steady-state GA or GENITOR (Whitley & Kauth, 1988). Starting from a randomly-generated initial population (feasible solutions), the genetic algorithm applies iteratively four operators: selection, crossover, mutation and culling to produce a new population of solutions.

The search space S is composed of all possible λ_i values, for $i \in \{1, ..., n-1\}$. The production line can contain inspection stations located at the exit of buffer B_i and the provisioning point for machine M_{i+1}. The presence or absence of these stations is captured by a binary variable λ_i that takes 1 if there is an inspection station at the exit of the buffer B_i and 0 if not, respectively.

To avoid creating infeasible solutions after the crossover operator, we choose to generate solutions with same address. We divide the solution into two same parts and we create an address for each solution as follows:

The address of λ is defined by: $Address(\lambda) = (\sum_{l=1}^{n/2} \lambda_l, \sum_{l=n/2+1}^{n} \lambda_l)$.

Compared to standard genetic algorithm, the mutation step is replaced in EM/TG by Tabu Search. These operators can be summarized as follows:

1. **Selection operator:** it randomly selects, from the population, two solutions of same address named parent 1 and parent 2 that will "reproduce" to create a new solution.
2. **Crossover operator:** it produces a new solution (child or offspring) from a selected pair of parent solutions. We use the so-called one-point crossover operator, which, as illustrated in Table 1, creates the child string O for the given pair of parent strings λ_1 and λ_2 by:

(a) Copying string elements belonging to the first fragment from λ_1;

(b) Copying the rest of string elements from λ_2.

Parent string λ_1	0	0	0	1	0	0	1	0	0	0	‖	0	1	0	0	0	0	0	1	0	1
Parent string λ_2	0	0	0	0	1	0	1	0	0	0	‖	0	1	1	0	0	0	0	0	0	1
Child string		0	0	0	1	0	0	1	0	0	0	‖	0	1	1	0	0	0	0	0	0	1

Table 1: An example illustrating the 1-point crossover procedure

3. **Tabu Search (TS)** (this replaces the Mutation operator): it looks for the best solution in the subspace associated with the new child using the TS algorithm described previously. The best solution found becomes the new child.

4. **Culling operator**: it generates the new population. Each new solution obtained by TS is decoded to obtain its fitness value. The fitness value, which is in fact a measure of quality, is used to compare different solutions. This fitness is evaluated according to the objective function $J(a, \lambda)$ optimized as a subproblem with an exact method (for more details see Section 3.2). If the new solution obtained by TS is better than the worst solution currently in the population, it replaces it, else it is discarded. After N_{rep} new solutions have been produced, new randomly constructed solutions are generated to replenish the population, and a new genetic cycle begins. The GA is terminated after N_c genetic cycles. The final GA population contains the best solution.

3.2   Flow optimization model

For a fixed location λ, the subproblem used to evaluate the fitness can be reformulated as a network flow problem. Consider the connected network $G(E, V)$ consisting of a set of nodes E and a set of links V as illustrated by Figure 2. Each column i corresponds to a set of possible buffer levels for a machine i, such that:

$$a_{i1} = 100 \times \max((\frac{r}{r+p})^{i-1}, \frac{d}{k}\frac{r+p}{r}) \quad \text{and} \quad a_{in_i} = 100 \times \min(\frac{a_i(r+p)}{r}, 1).$$

Each node is connected to all nodes in the next column. To each link is associated some real number $c_{i,i+1}$ that corresponds to the shortage costs and inspection costs, such that: $c_{0,1} = 1$, $c_{i,i+1} = T^i(a_i, a_{i+1}, \lambda) + C_I \times \lambda_i \times d_i$ and $c_{n-1,n} = T_F(a_{n-1}, a_n)$. The main objective is to find for each λ the shortest path between a_0 and a_n.
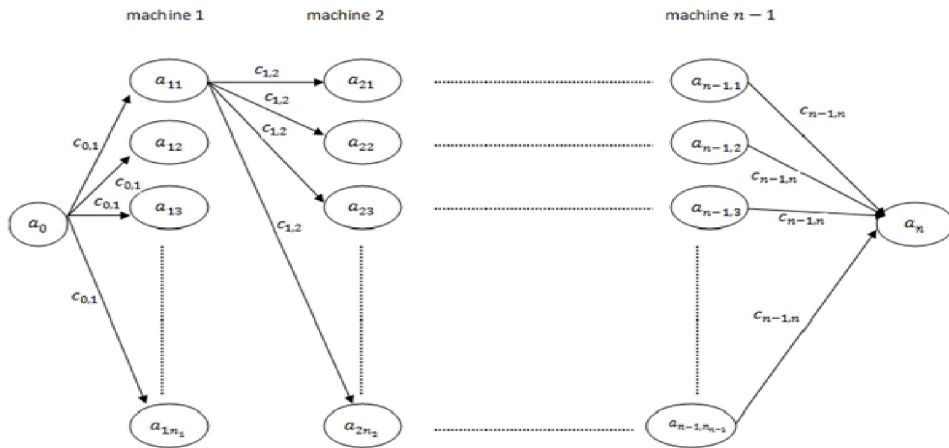

Fig. 2: The production line

4 Numerical Results

To analyze the solution method introduced in this paper, we tested it on a large problem with 20 machines, one of the problems benchmarked in previous research. This problem was first introduced and solved by the exact ESM method in (Ouzineb et al., 2013). The exact method applied to this problem takes hours while the new approach finds optimal or near optimal solutions in a fraction of time. Our algorithm was implemented in $C++$. The numerical tests were performed on an Intel Core i7 at 2.8 GHz with 8 Gbytes of RAM running under Linux.

Table 2 shows the parameters used for the experimentation. To compare the proposed hybrid approach EM/TG to ESM, we re-implemented ESM and re-tested it and EM/TG in the same conditions (computer, programming language, operating systems, etc.). This allows us to determine

Table 2: system parameters

	n	p	r	k	β_i	d	c_p	c_I	a_n^{des}
20 machines	20	0.2	0.9	9	0.1	1	0.1	0.2	0.95

the optimal solutions to compare to. We define two ratios in terms of the objective value and CPU time to determine which solution method performs better:

$$\text{MPCI} = 100\% \times \frac{\text{(Minimal ESM Cost - Minimal EM/TG Cost)}}{\text{Minimal ESM Cost}}. \tag{13}$$

$$\text{MPTI} = 100\% \times \frac{\text{(Minimal ESM Time - Minimal EM/TG Time)}}{\text{Minimal ESM Time}}. \tag{14}$$

Comparison results are given in Table 3 . The column m indicates the number of inspection stations to use. We varied this number between 1 and 10. We see that EM/TG obtains optimal solutions in 90% of the cases in less time. It reduces the solution time by huge factors up to 40 compared to ESM.

	Cost			Running time (s)		
m	ESM	EM/TG	$MPCI$	ESM	EM/TG	$MPTI$
1	**8.3125**	**8.3125**	**0.00**	08.44	424.63	**-98.01**
2	**6.2645**	**6.2645**	**0.00**	75.82	709.54	**-89.31**
3	**5.9802**	**5.9802**	**0.00**	432.83	1610.94	**-73.13**
4	**6.0663**	**6.0663**	**0.00**	1792.58	1535.98	**14.32**
5	**6.2772**	**6.2772**	**0.00**	5402.57	2254.07	**58.28**
6	**6.5230**	**6.5230**	**0.00**	12509.7	1675.02	**86.61**
7	**6.7959**	**6.7959**	**0.00**	23037.5	1714.70	**92.56**
8	**7.0965**	**7.0965**	**0.00**	34423.4	1739.85	**94.95**
9	**7.4298**	7.5000	**-0.94**	41982.9	1322.47	**96.85**
10	**7.7947**	**7.7947**	**0.00**	42057.9	1222.84	**97.09**

Table 3: Comparison results for 20 machines

Table 4 shows the results obtained for the test problems with 20 machines by using the proposed hybrid method and the exact approach limiting the running time to 3600 seconds. In all test cases, EM/TG performs as well as ESM in 40% of the test cases and better in the remaining 60% of the test cases when we limit the solution time to one hour.

m	1	2	3	4	5	6	7	8	9	10
EM/TG	**8.3125**	**6.2645**	**5.9802**	**6.0663**	**6.2772**	**6.5230**	**6.7959**	**7.0965**	**7.5000**	**7.7947**
ESM	**8.3125**	**6.2645**	**5.9802**	**6.0663**	6.2774	6.8639	7.3566	7.8709	8.4458	8.9299

Table 4: The results obtained by the proposed method and exact approach for 20 machines during 3600 seconds

N.B: For instance (m=9) the algorithm has converged locally after 1322 seconds and could not improve this solution even after more time.

5   Conclusions

This paper proposes an efficient hybrid approach based on an exact method and methaheuristics to solve the buffer sizing problem in unreliable heterogeneous production lines with several inspection stations. This is a very hard mixed integer nonlinear program. The proposed approach uses ideas from genetic algorithm and tabu search to identify profitable configurations (locations of the inspection stations). For these locations, we use an exact approach to decide the optimal sizes of the buffers to use. Our final goal is to find an optimal design.

This hybrid approach provides a balance between diversification and intensification that shows to work well on such heterogeneous production lines with up to 20 machines. When compared to the exhaustive exact method, the hybrid approach results are superior in terms of CPU time.

Future research will focus on applying our heuristic to solve larger instances (up to 30 machines and a variable number of inspection stations).

References

Colledani, M., & Tolio, T. (2005). Impact of statistical process control (spc) on the performance of production systems-part 2 (large systems). Zakynthos Island, Greece.

Colledani, M., & Tolio, T. (2006a). Impact of quality control on production system performance. *Annals of the CIRP*, *55*(1), 453-456.

Colledani, M., & Tolio, T. (2006b). Performance evaluation of production systems monitored by statistical process control and offline inspections. *Information Control Problems in Manufacturing*, 317-322.

Colledani, M., & Tolio, T. (2011). Integrated analysis of quality and production logistics performance in manufacturing lines. *International Journal of Production Research*, *49*(2), 485-518.

Inman, R., Blumenfeld, D., Huang, N., & Li, J. (2013). Survey of recent advances on the interface between production system design and quality. *IIE Transactions*, *45(6)*, 557-574.

Kim, J., & Gershwin, S. (2005). Integrated quality and quantity modeling of a production line. *OR Spectrum*, *27*, 287-314.

Kim, J., & Gershwin, S. (2008). Analysis of long flow lines with quality and operational failures. *IIE Transactions*, *40*, 284-296.

Mandroli, S., Shrivastava, A., & Ding, Y. (2006). A survey of inspection strategy and sensor distribution studies in discrete-part manufacturing processes. *IIE Transactions*, *38(4)*, 309-328.

Mhada, F., Malhamé, R., & Pellerin, R. (2014). Joint assignment of buffer sizes and inspection points in unreliable transfer lines with scrapping of defective parts. *Production and Manufacturing Research*, *1*, 79-101.

Ouzineb, M., Mhada, F., El Hallaoui, I., & Pellerin, R. (2013). An exact method for solving the bffer sizing and inspection stations allocations problem. *IESM'2013, Rabat, Morocco*.

Sadr, J., & Malhamé, R. (2004). Unreliable transfer lines: decomposition/ aggregation and optimisation. *Annals of Operations Research*, *125*, 167-190.

Whitley, D., & Kauth, J. (1988). *Genitor: A different genetic algorithm* (Tech. Rep. No. CS-88-101). Colorado State University.