Simulation Optimization Approach to Solve a Complex Multi-objective Redundancy Allocation Problem

Carlos Henrique Mariano and Carlo Alessandro Zanetti Pece

Abstract This chapter addresses the problem of redundancy and reliability allocation in the operational dimensioning of an automated production system. The aim of this research is to improve the global reliability of the system by allocating alternative components (redundancies) that are associated in parallel with each original component. By considering a complex componential approach that simultaneously evaluates the interrelations among subsystems, conflicting goals, and variables of different natures, a solution for the problem is proposed through a multi-objective formulation that joins a multi-objective elitist genetic algorithm with a high-level simulation environment also known as simulation optimization (SIMO) framework.

1 Introduction

The simulation/optimization framework (SIMO) is an iterative and stochastic technique for generating multiple optimization scenarios where the optimization process occurs simultaneously with system simulation analysis. In general, there are two approaches to SIMO framework. On the first the simulation process is used as a validation tool or test for the effectiveness of any optimization method as seen in [9, 32, 46]. The second presents an optimization process through the simulation applied to the resolution of a complex problem as we see in the texts [22, 36, 44].

The whole process can be summarized in two questions. What happens if? and How do I get? The first is applicable to the analysis of multiple simulation possibilities or scenarios. The second is applicable to optimization analysis, where we can maximize or minimize important criteria or objectives to streamline the effectiveness of the system [30].

C.H. Mariano (✉)
Department of Electrical Engineering - DAELT, Federal Technological University of Paraná - UTFPR, Curitiba, PR, Brazil
e-mail: mariano@utfpr.edu.br

C.A.Z. Pece
Department of Electrical Engineering - DAELT - Postgraduate Program in Biomedical Engineering, Federal Technological University of Paraná - UTFPR, Curitiba, PR, Brazil

M. Mujica Mota et al. (eds.), *Applied Simulation and Optimization*,
DOI 10.1007/978-3-319-15033-8_2

2 Considerations About the Process

In this chapter, the idealized case under consideration assumes the existence of an automated system characterized by an operational scenario (e.g., machine configuration, maintainers, costs, and individual reliabilities). To obtain an optimal scenario by considering the inherent analytical difficulties, random factors, and conflicting multiple objectives, a simulative process combined with an optimization process is proposed.

This process makes intensive use of computational power and can be defined as an iterative stochastic technique also known as simulation optimization framework. The technique generates multiple operational scenarios and those that best meet the problem objectives and constraints are chosen. A generation of new operational scenarios is evaluated after each iteration and, after many iterations, a set of scenarios and solutions can be selected from many possible combinations. The evaluation process starts at the simulator. In the idealized automated system—a discrete complex system—the following relationships are observed:

(1) Nonlinear relationships;
(2) Relationships with feedback loops;
(3) Interchanged relationships (input/output) with the environment;
(4) Past-state dependent relationship;
(5) Concatenated relationships; and
(6) Relationships consisting of different types of variables with different natures.

The modeling of such a system requires an appropriate simulation environment. Among the several options available, the ARENA commercial simulation software was chosen because it is oriented to discrete events. The present study adopted a variant of the simulation optimization process, where the simulator is the core of the process [1]. Due to the high dimensional aspect and complexity of the system, it is virtually impossible to represent it as a single optimization function (conventional case). Therefore, the simulation model enables representation of complex system relationships.

In this variant of the simulation optimization process, system performance is tested by using indicators and variables captured from the simulation model [1, 37]. Thus, the simulator produces inputs required by the optimizer to guide the selection of the operational scenarios (solutions) that best fit the objectives and constraints of the problem.

The term “optimization”, which is commonly used in the computer science field, is used broadly in this chapter. Optimization can be defined as an iterative process of global search, where the optimum is approximated through stochastic processes.

The elitist multi-objective genetic algorithm (MOGA) that was chosen for the present study is the NSGA-II (Nondominated Sorting Genetic Algorithm II), whose ordination principle is based on the notion of Pareto dominance [6, 7]. In this algorithm, the simulation optimization process is completed once a set of optimal Pareto solutions is obtained.

In addition to the optimal Pareto set, a procedure to aggregate quantitative and qualitative information is necessary to determine the best solution. The qualitative information usually reflects the opinion of the specialists involved in system operation and maintenance. The quantitative information is obtained from technical and administrative specifications [5].

3 Some Aspects of Simulation Model

It is worth highlighting the stochastic nature of the simulation model. This characteristic is the basis of the main argument of this study, particularly the unlikelihood of building a single function that represents all of the stochastic system relationships. Therefore, this study proposes a different approach. Significant random aspects of the model, such as the failure and repair process, the loss of machine production speeds after a certain number of failures have occurred, and the processing of lines produced in the machines, are ruled by several variables:

(1) Machine operating time;
(2) Waiting time for a part;
(3) Waiting time for another operation;
(4) Machine failure time; and
(5) Machine repair time.

The events related with failure and repairs of each machine or robot were exponentially distributed with different means. Exponential distributions feature risk functions with constant failure rates throughout the life span of the system. Therefore, they are frequently used to represent failure or repair of complex systems. Different times for scheduled maintenance inspection (preventive) were assigned to each machine. Such times depend on the particular type of inspection, which is determined by a certain number of parts produced.

The topology and assignment of values in the simulation model are extremely dependent on the experience of the modeler. Consequently, a detailed description of the simulation model used in this study would be lengthy and beyond the scope and purpose of this paper.

The automatized system adopted and represented by the idealized case under consideration is discrete. The approach for this type of system requires specific languages and modeling tools. In the present study, Petri networks were adopted to specify the conceptual model of the production process.

The Petri conceptual model facilitated the planning of all involved operations and clearly established system transitions and operations. The model also allowed visual and facilitated changes to be made in the simulation model and defined hierarchies in the sequences of operations.

The advantage of this method, as compared with other existing methods, is that its formalism is based on a simplified graphic model with few syntactic rules [28, 31, 33]. Another interesting characteristic of the Petri model is that its logical development resembles the block-oriented programming logic of current simulation environments such as ARENA, PROMODEL, and SIMUL8.

It is worth highlighting the stochastic nature of the simulation model. This characteristic is the basis of the main argument of this study, particularly the unlikelihood of building a single function that represents all of the stochastic system relationships. Therefore, this paper proposes a different approach. Significant random aspects of the model, such as the failure and repair process, the loss of machine production speeds after a certain number of failures have occurred, and the processing of lines produced in the machines, are ruled by several variables:

- Machine operating time;
- Waiting time for a part;
- Waiting time for another operation;
- Machine failure time; and
- Machine repair time.

The events related with failure and repairs of each machine or robot were exponentially distributed with different means. Exponential distributions feature risk functions with constant failure rates throughout the life span of the system. Therefore, they are frequently used to represent failure or repair of complex systems. Different times for scheduled maintenance inspection (preventive) were assigned to each machine. Such times depend on the particular type of inspection, which is determined by a certain number of parts produced.

The topology and assignment of values in the simulation model are extremely dependent on the experience of the modeler. Consequently, a detailed description of the simulation model used in this study would be lengthy and beyond the scope and purpose of this chapter.

4 Multi-objective Optimization Models

A multi-objective optimization model strives to minimize or maximize a group of functions that are usually in mutual conflict. The existence of multiple objective functions suggests a fundamental difference between multi-objective and mono-objective optimizations. Therefore, there is no one single solution to the problem, but a cluster of solutions that invoke different compromises among the values of the functions to be optimized. Within a set of efficient solutions, the desired solution is one that exhibits the best compromise for the objectives (e.g., Pareto-optimal and nondominated solutions). The identification of the best compromise solution requires that the decision-maker establish a preference because many objectives are not only in conflict with each other but also represent different phenomena described by deterministic and stochastic mathematical functions.

The classic multi-objective optimization methods, according to [8] and [5], can be divided into two categories: complete enumeration methods and preference-based methods. The former category presents a full group of nondominated solutions to the decision-maker, which allows him to choose the solution that best suits his goals. The latter category is based on choosing a solution that offers the best compromise from a group of nondominated solutions, through the use of an explicit or implicit criterion. Such compromises indicate that the solution must simultaneously satisfy the analytical functions of the objectives and the degrees of preference established by the decision-maker. Deb [5] presents another classification that also divides the methods into two categories: preference and nonpreference. Deb [5] expands the preference-based methods into three subcategories: posteriori selection methods, priori methods, and interactive methods. The following is a list of distinct preference-based methods:

a. Weighted sum method;
b. Utility function method;
c. Constraint method—ϵ (where ϵ is the upper limit of the scenario function inserted as a constraint to the problem);
d. Weighted metrics method;
e. Benson's method;
f. Value function method;
g. Goal programming method;
h. Pareto method; and
i. Lexicographic method.

The natural complexity inherent to multi-objective optimization problems poses a challenge to exact algorithms.[1] For this reason, evolutionary algorithms are becoming increasingly popular as robust and effective methods to solve single- and multi-objective optimization problems [4].

The operation of Evolutionary Algorithms (EA) is based on the simulation of natural evolution [11]. The EA use an iterative technique that applies stochastic operators to a group of individuals (population) to improve their levels of adaptation to the problem, performance, or fitness. In most applications, this measure is related to the objective functions of the problem being considered.

Multi-objective evolutionary algorithms (MOEA) are capable of treating multiple objectives naturally because they operate naturally in parallel on the group of solutions and develop a set of solutions similar to the Pareto boundary or frontier at each execution [5]. Such characteristics allow MOEA to approach problems with a large solution space.

According to Nesmachnow [29], MOEA exhibit two special operators that do not appear in the generic structure of EAs: a diversity operator and a fitness attribution operator. The diversity operator represents a technique used to avoid premature convergence to a sector of the Pareto boundary (e.g., niche, fitness, sharing, and crowding) and assess diversity. The fitness attribution operator is aimed at ensuring the permanence of individuals with the best characteristics for future generations by considering the values of the objective functions and the results of the metrics.

[1] Enumerative local search algorithms based on gradients or that use standard techniques of deterministic programming, such as greedy algorithms, or branch and bound techniques [14, 29].

For these reasons above, the MOEA are very suitable to apply in the majority of actual optimization problems in engineering. In recent decades multi-objective optimization models have gained a crescent acceptance for multiple applications in various areas of engineering, Coello and Lamont [4], Zio and Zille [48], Çunkas [25], Zhuo et al. [47], Tian et al. [42], Tzu-Chieh and Kuei-Yuan [43]. The difficulty in these cases lies in representing the complex interrelations contained in such systems, as reported by the following authors below, which are better defined on high-level simulation environment:

Mattila and Virtanen [23]—Airplane maintenance workshop—To maximize the availability of airplanes to ensure fleet operation capability, while minimizing the disparity between scheduled and nonscheduled maintenance operations.

Merkuryeva and Napalkowa [26]—Supply/logistics chain—To minimize total average cost corresponding to the total cost of storage, production, and procurement and maximize customer service satisfaction.

Hani et al. [10]—Train maintenance workshop—To maximize the production rate represented by the number of vehicles that leave the workshop per year, minimize the waiting time of vehicles, and the occupation rate of maintainers in the different groups.

5 Reliability Optimization—Redundancy Allocation Problem to Improve System Reliability—A Review

Analysis of the reliability of engineering systems, particularly industrial systems, can be performed by two ways: by the classical/binary approach, where, according to [2], a system subjected to failure can assume only two different states, total standstill or fully functional. And the multistate approach, where, according to [21], a system might assume several states between fully functional and total failure.

The multistate approach better describes actual systems because they are subjected to a series of factors that diminish their life span [21]. The occurrence of failure in such systems may not necessarily cause complete standstill of the activity despite the increasing tendency for such standstill.

Whether the approach adopted for the analysis and optimization of the reliability of a system is binary or multistate, the optimization techniques are typically classified by the following four methods [17, 18]:

1. Redundancy allocation (RAP)—where the decision variables represent the number of redundancies;
2. Reliability allocation—in this case, the decision variables are the reliabilities of the components or subsystems;
3. Redundancy/reliability allocation—the decision variables are a combination of the number of redundancies and reliabilities of the components or subsystems; and
4. Components or subsystems allocation—where the system configuration (component arrangement) represents the decision variables.

Redundancy allocation problems (RAP) are problems related to the search for specific combinations of alternative components that, when properly associated with the components whose reliabilities needs to be improved, enhance the global reliability of the system. Their nature is eminently combinatorial; they can be considered to be nonlinear mixed integer programming problems or a special case of integer programming, and they can be solved by the traditional cut or search methods, or by a combination of both, as presented by Jianping and Xishen [12], Misra and Sharma [27]. The problems are usually solved by mono- or multi-objective formulations and, more recently, exploration of multilevel and multistate RAP has been observed, as presented by [16, 41, 45]. Consequently, these problems have become a focus of many studies, which adds an even greater degree of complexity to traditional problems.

The use of RAP-solving methods such as algorithm-hybridization processes, which combine heuristic methods, neuronal networks, fuzzy techniques, and local search methods with metaheuristic types, are increasing. The processes are utilized alone to improve computational efficiency or with exact methods to reduce the search space. Their use creates the potential for combining two metaheuristics, annealing-genetic (AG) algorithms and simulated annealing, as indicated by [3, 13, 15, 18, 19, 24, 35, 39].

EA exhibit interesting characteristics that enable them to manage RAP in mono-objective, multi-objective, and multilevel formulations. They can control noncontinuous, nonconvex, and/or nonlinear spaces, as well as unknown objective functions. Although genetic algorithms are frequently used, current RAP-solving is moving toward the application of EA that differ from genetic algorithms, as presented by Salazar et al. [34], Taboada and Coit [38], Lins and Droguett [20], Taboada et al. [40].

To test the algorithms applied to multi-objective RAP-solving, a general formulation is utilized that maximizes the reliability of the system, minimizes costs associated with the allocation of components, minimizes the weight and volume of the system, and it is subjected to constraints in cost, volume, weight, number of components to be allocated, and system reliability limits. Mathematically, this formulation is depicted in Fig. 1
where

$\boldsymbol{m}$—number of subsystems or stages
$\boldsymbol{i}$—subsystem index, $i = 1, 2, \ldots, m$
$\boldsymbol{j}$—index of the components of each subsystem, $j = 1, 2, \ldots, n$
$\boldsymbol{r}_{ij}$—reliability of component j in subsystem i
$\boldsymbol{c}_{ij}$—cost of component j in subsystem i
$\boldsymbol{w}_{ij}$—weight of component j in subsystem i
$\boldsymbol{R}_s$—total reliability of the parallel-series system
$\boldsymbol{C}_s$—total cost of the parallel-series system
$\boldsymbol{W}_s$—total weight of the parallel-series system
$\boldsymbol{C}_o$—allowed system cost
$\boldsymbol{W}_o$—allowed system weight
$\mathbf{a}_i$—number of component choices available for subsystem i

Fig. 1 RAP—general multi-objective formulation

$$Max \quad R_s = \prod_{i=1}^{m} \left(1 - \prod_{j=1}^{a_i} (1 - r_{ij})^{xij}\right)$$

$$Min \quad C_s = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$Min \quad W_s = \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} x_{ij}$$

$$s.t. \quad \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \leq C_o$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} x_{ij} \leq W_o$$

$$\sum_{j=1}^{a_i} x_{ij} \leq n_{\max}, \quad i = 1,2,\ldots,m$$

$$\sum_{j=1}^{a_i} x_{ij} \geq n_{\min}, \quad i = 1,2,\ldots,m$$

$$x_{ij} \in Z^+, \quad i = 1,2,\ldots,m; \quad j = 1,2,\ldots,n$$

$\mathbf{x}_{ij}$—amount of component j used in subsystem i
$\boldsymbol{n}_i$—total number of components that might be in parallel
$\boldsymbol{n}_{max}$—maximum number of components that can be in parallel
$\boldsymbol{n}_{min}$—minimum number of components that can be in parallel.

Thus, RAP can be classified according to the type of redundancy, such as cold standby, warm standby, hot standby, parallel redundancy, or active redundancy. The type of components to be allocated can be classified as identical or nonidentical components; however, nonidentical components are more complex and similar to the actual systems. The redundancy levels of items can include the component redundancy level (individual component level), modular redundancy level (subsystem level), or system redundancy level, which produces a multilevel RAP. Based on its functioning status, a component can be classified as binary, where the component might assume one of two possible states (e.g., zero for total failure or one for fully functional), or multistate, where the component might assume multiple states between total failure and total functioning. The RAP can also be classified according to the type of system and whether its nature features repairable or nonrepairable components.

In the present study, classification was performed with either a basic componential attribute, where all components were described in terms of three attributes: weight, volume, and individual reliability, or a complex componential attribute including other attributes such as failure and repair distributions, inspections, and process speeds. The last feature increases the degree of complexity of the problem in terms of computational effort because the number of possible combinations increases.

This chapter considers RAP as linked with subsystems characterized by attributes not typically found in combinatory problem solving instead of simple components.

In basic and complex componential approaches, a component with an adequate set of attributes is required that, when combined with the remaining allocated components, will result in the total final reliability of the system. The total final reliability value depends on the structure of the system. A review of current research indicates that analysis in this area is primarily focused on a classical parallel–series structure. The present study approached the analysis similar to that of [18], whose general statement is represented by the following equations:

$R_i = -\prod_{j=1}^{n_i}(1 - r_{ij})$ and the system reliability is defined by $R_s = \prod_{i=1}^{k} R_i$ where

$\boldsymbol{R}_i$—reliability of subsystem i
$\boldsymbol{r}_{ij}$—reliability of a component j, $1 \leq j \leq n_i$ of subsystem i
$\boldsymbol{n}_i$—components in parallel
$\boldsymbol{k}$—subsystems in series.

Joining both expressions yields the following equation:

$$R_s = \prod_{i=1}^{k} \left[1 - \prod_{j=1}^{n_i}(1 - r_{ij})\right].$$

The reliability value only acquires its full meaning when it is associated with a definite timepoint; however, a literature review reveals that this aspect is seldom addressed. This chapter is rooted in this context and adds an appropriation of time to the complex componential approach by considering that the values of reliability are measured in time. The component is replaced by a subsystem whose main attributes are not only related to weight, volume, and individual reliability but also to time.

Conversely, the component is now a machine that performs a certain process within a production line characterized by the following attributes:

(a) operation speed;
(b) repair time;
(c) production cost;
(d) individual reliability; and
(e) production capacity.

The present study considers realistic problems of superior complexity. The modeling of such problems considers different types of variables (real, binary, stochastic, and nonstochastic) concomitantly as more than one objective is optimized. Given the high dimension and complexity of the system and the impracticality of representing the system analytically, a simulative and evolutionary optimizing process combined with remarkable characteristics was selected to solve this type of problem [7].

To dimension a group of machines in an idealized automated production line, the process was applied to the project stage as a function of its inherent advantages and the control of the conjunctural variables of the system (e.g., production standstill due to strikes and lack of raw materials). The expected practical objective is to demonstrate the feasibility of the process rather than to describe an empirical implementation example. The result is a pragmatic method that can be applied to actual manufacturing, especially in automated environments.

6 Formulation of the Variables, Objectives, and Constraints of the Multi-objective Model

The variables of the simulation optimization process can be classified as both exogenous and endogenous to the main code, i.e., the MOGA. The exogenous variables are processed inside the simulator and based on a given operational scenario. The elements of each scenario (e.g., configuration, maintainers, costs, and reliabilities) are represented in the model by a set of endogenous variables that form a chromosome or individual of a specific age.

The objectives of optimization are calculated from exogenous and endogenous variables. In the idealized case under consideration, four conflicting objectives, which must be optimized concurrently, are used:

(1) *TCSM*—Total cost of system maintenance (minimization)—exogenous variable;
(2) R_s—Total reliability of the system (maximization)—exogenous variable;
(3) *TOC*—Total cost of system operation (minimization)—exogenous variable; and
(4) *Numan*—Number of maintainers (minimization)—endogenous variable.

Certain variables are either part of the optimization objectives or represent the objectives themselves, as is the case of TMFC (total maintenance fixed cost) and numan (number of maintainers), respectively. It must still be emphasized that variable $\boldsymbol{a}_{ij}$ represents the allocation of a machine/robot i in stage j of the system. Such a variable plays a crucial role in the production variability of the operational scenarios and can be presented as follows:

$$\boldsymbol{a}_{ij}$$

$$\begin{cases} 0, \textit{for non allocated machine/robot} \\ 1, \textit{for allocated machine/robot} \end{cases}$$

Thus, the functioning of the full process is based on the correct definition of the constraint limits. This task is highly dependent on the degree of familiarity of the analyst with the system; without such knowledge appropriate limits would not be imposed on the process. In other words, the limits represent the horizons of the solution spaces and the problem variables that guide the search process in the desired direction. So, below is the list of variables used on the models:

(1) ($cmttpm_{ij}$)—*cost of materials, tools, and outsourced manpower applied to the maintenance of machine i in subsystem j*
(2) ($cinst_{ij}$)—*cost of allocation of redundancy maintenance in machine i in subsystem j*
(3) ($cplf_{ij}$)—*cost of production loss due to failure of i in subsystem j*
(4) *General management cost (GMC) corresponding to 10% of the sum of costs* $cmttpm_{ij}$,$cinst_{ij}$ *and* $cplf_{ij}$. $\left[\sum_{i=1}^{k}\sum_{j=1}^{n} a_{ij} * (cinst_{ij} + cmttpm_{ij} + cplf_{ij})\right]$
(5) dtm_{ij}—*total standstill time of machine i in subsystem j*
(6) $TTPS_{ij}$—*total standstill time of machine i in subsystem j,*
(7) ot_{ij}—*operation time of machine i in subsystem j*
(8) $csmm_{ij}$—*cost of specialized manpower to maintain machine i in subsystem j.*
(9) *crama—cost of applied raw material*
(10) *cmpol—cost of manpower for operating the line*
(11) *ceeoc—cost of electricity and other fuel*
(12) *cpsat—cost of packaging, storing, and transport*
(13) *TPPS—total number of products produced by the system* (*exogenous variable*)
(14) *TNSF—total number of system failures*
(15) *TSA—total system availability*
(16) *SDT—system downtime*
(17) *TGPPS—total of good products produced by the system*
(18) *TMFC—total fixed costs of maintenance*
(19) *CMTTPM—total cost of maintenance tools, parts, and manpower of the machines* (*exogenous variable*)
$CMTTPM = \sum_{i=1}^{n}\sum_{j=1}^{m} cmttpm_{ij} * a_{ij} \le UB$
(20) *CINST—total cost of machine installation* (*exogenous variable*)

$$CINST = \sum_{i=1}^{n}\sum_{j=1}^{m} cinst_{ij} * a_{ij} \le UB$$

(21) *CCPLF—total cost of production loss by the machines due to standstill caused by failure (exogenous variable)*

$$CPLF = \sum_{i=1}^{n}\sum_{j=1}^{m} cplf_{ij} * a_{ij} \leq UB$$

(22) *cost of production loss by the machine i in subsystem j due to standstill caused by failure*

$$cpfl_{ij} = \left(ocm_{ij} \times \left(\frac{ppm_{ij}}{eppm_{ij}} \right) \times rt_{ij} \right) * a_{ij}$$

(23) ocm_{ij}*—operation cost of machine i in subsystem j*
(24) ppm_{ij}*—total products produced by machine i in subsystem j*
(25) $eppm_{ij}$*—expected total of products produced without occurrence of failures of machine i in subsystem j*
(26) rt_{ij}*—repair time of machine i in subsystem j.*
(27) *total General Maintenance Cost, GMC, corresponding to 10 % of the sum of costs* $cmttp_{mij}$, $cins_{tij}$, *and* csp_{lij}. *(exogenous variable)*

$$GMC = \sum_{i=1}^{n}\sum_{j=1}^{m} (cmttp_{ij} + cinst_{ij} + cplf_{ij}) * a_{ij} \leq UB$$

The complete multi-objective model is shown as

$$\boldsymbol{MIN\,TCSM} =$$

$$\left[\frac{\left[\sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij} * \left[(numan * e^{(\frac{1}{70} * TTPS_{ij})} + (ot_{ij} + dtm_{ij})) * csmm_{ij} \right] * a_{ij} \right] + (\sum_{i}^{n}\sum_{j}^{m} CPLF_{ij} * a_{ij}) + 1.1 * (\sum_{i=1}^{k}\sum_{j=1}^{n} a_{ij} * (cinst_{ij} + cmttpm_{ij} + cplf_{ij}))}{TSA} + TMFC \right]$$

$$\boldsymbol{MIN\,TOC} = TPPS * 0,05 * crama + TPPS * 0,15 * cmpol + TPPS * 0,10 * ceeoc + TPPS * 0,06 * cpsat + TMVC$$

$$\boldsymbol{MAX\,R_s} = \prod_{i=1}^{k} \left[1 - \prod_{j=1}^{n_j} (1 - r_{ij} * a_{ij}) \right]$$

where:

$$CMTTPM = \sum_{i=1}^{n}\sum_{j=1}^{m} cmttpm_{ij} * a_{ij} \leq UB;$$

$$CINST = \sum_{i=1}^{n}\sum_{j=1}^{m} cinst_{ij} * a_{ij} \leq UB;$$

$\sum_{i=1}^{n} \sum_{j=1}^{m} cplf_{ij} * a_{ij} \leq UB$

$CGM = \sum_{i=1}^{n} \sum_{j=1}^{m} (cmttp_{ij} + cinst_{ij} + cplf_{ij}) * a_{ij} \leq UB;$

$LB \leq R_s \leq UB; LB \leq TOC \leq UB; LB \leq TMC \leq UB; LB \leq numan \leq UB;$
$LB \leq TMFC \leq UB$

$LB \leq SDT \leq UB; LB \leq TNSF \leq UB; LB \leq TSA \leq UB.$

LB—lower bound—UB—Upper bound

6.1 Declaration of the Nonlinear, Nonstochastic, and Mono-Objective Test Model

In addition to the multi-objective problem, a mono-objective model was included for the purpose of comparison to the full model, which was constructed to reduce its complexity by removing several stochastic features present in the full model.

Thus, this model was developed on the grounds of *ϵ-constraint methods or approach*, according to [5]. Three of the main objectives were transformed into constraints to maximize the system reliability through allocation of redundancies. The costs related to the use of tools, parts, and manpower in the maintenance of machine j in subsystem I ($cmttpm_{ij}$); with the installation of a machine j in subsystem i ($cinst_{ij}$), and with the loss of production of machine j in subsystem i due to standstill caused by failure ($cppm_{ij}$) were predefined for each machine, thus forming a fixed cost matrix.

The amount of products produced by the system per period of system operation (720, 1,440, and 2,160 h) was calculated and supplied. Events related to failure or repair time of the system were not considered. A total system availability (TSA) of 0.90 was established for each simulation period. This model ensures that at least one machine will be allocated to each subsystem. Constant *A* is a fixed value in each simulation period, corresponding to 3,973.99, 7,959.12, and 11,989.40 currency units (CU) for 720, 1,440, and 2,160 h, respectively. The model was encoded in A Mathematical Programming Language (AMPL) and the KNITRO 6.0 solver was used. A total of forty-five binary and five integer variables with thirteen linear constraints was used.

The maximized model is shown as

$$\boldsymbol{Max}\ R_s = \prod_{i=1}^{k} \left[1 - \prod_{j=1}^{n_j} (1 - r_{ij} * a_{ij})\right]$$

where

$$\boldsymbol{TCSM} = \left[\frac{\boldsymbol{numan*BESETTINGA + 1.1*(\sum_{i=1}^{k} \sum_{j=1}^{n} a_{ij}*(cinst_{ij} + cmttpm_{ij} + cplf_{ij}))}}{\boldsymbol{TSA}} + \boldsymbol{TMFC}\right] \leq \mathbf{3500 \times 10^3}$$

$\boldsymbol{TOC} = TPPS * 0,05 * crama + TPPS * 0,15 * cmpol + TPPS * 0,10 * ceeoc + TPPS * 0,06 * cpsat + TMVC \leq \mathbf{8000 * 10^3}$

$\sum_{i=1}^{k} \sum_{j=1}^{n} a_{ij} \leq n$

$CPPROS = \sum_{i}^{k} \sum_{j}^{n} (cppm_{ij} * a_{ij}) \leq 900 \times 10^3$ Currency units

$CINST = \sum_{i=1}^{k} \sum_{j=1}^{n} cinst_{ij} * a_{ij} \leq 900 \times 10^3$ Currency units

$CPPM = \sum_{i=1}^{k} \sum_{j=1}^{n} cppm_{ij} * a_{ij} \leq 10 \times 10^3$ Currency units

$CGM = \sum_{i=1}^{k} \sum_{j=1}^{n} a_{ij} * (cinst_{ij} + cmttpm_{ij} + cppm_{ij}) \leq 260 \times 10^3$ Currency units

$$
\begin{aligned}
&5 \leq \boldsymbol{numan} \leq 20 \\
&0.1 \leq \boldsymbol{crama} \leq 1.0 \\
&1.0 \leq \boldsymbol{cmpol} \leq 3.0 \\
&1.0 \leq \boldsymbol{ceeoc} \leq 7.0 \\
&1.0 \leq \boldsymbol{cpsat} \leq 5.0 \\
&250 \times 10^3 \leq TMFC \leq 350 \times 10^3 \text{ Currency units (CU)}
\end{aligned}
$$

7 Process of Interaction Between Simulator and Optimizer

Unlike the traditional optimization process, which is based on the construction of objective functions by [17, 18], the optimization process described in this chapter does not employ such functions in the usual manner. The full process is represented by a set of variables defined in both MOGA and the simulator. Certain variables are incorporated in the constraints whereas other variables become the objectives of optimization.

Evolution is guided by a strategy that selects the individuals best adapted for the solution of the problem and best satisfies the constraints and objectives of the problem. The MOGA is responsible for producing variability in the operational scenarios, which is performed automatically with crossover, mutation, and selection operators.

Each operational scenario, or each MOGA individual, is randomly generated and consists of the following endogenous variables:

(1) Number of maintainers (number of persons devoted to tasks related to system maintenance)—*numan* $\in \mathbb{Z}$;
(2) Allocation of machines or robots—binary variable a_{ij};

(3) Expected reliability of each machine, regardless of whether it is allocated or nonallocated in the system—real variable between 0 and 1;
(4) The four components of the total cost of system operation—real variables *crama, cmpol, ceeoc, and cpsat*; and
(5) TMFC—real variable *TMFC.*

Thus, the operational scenario for MOGA is a chromosome whose encoding represents a list of vectors containing:

(1) Forty-five binary variables for the allocation of machines: $a_{ij}, for\, i = 1, \ldots, n$ $components\, and\, j = 1, \ldots, k\, stages\, or\, subsystems$, which can assume a value of 0 or 1 and that correspond to the nonallocated and allocated machines, respectively.
(2) Forty-five individual reliabilities, real variables that represent, $r_{ij}\, for\, i = 1, \ldots, n$ *components*, and $j = 1, \ldots,$ *k stages or subsystems*;
(3) Four real variables, c_p, $p = 1, \ldots, z$ basic production costs;
(4) One real variable *numan* that represents the number of the system maintainers;
(5) One real variable representing the *TMFC* of the system for a given operational scenario.

Thus, a vector with a 96-variable longitude is obtained a follows:

$$\upsilon = \boldsymbol{a}_{ij}, \ldots, \boldsymbol{a}_{nk}/\boldsymbol{r}_{ij}, \ldots, \boldsymbol{r}_{nk}/\boldsymbol{c}_p, \ldots, \boldsymbol{c}_z/\boldsymbol{TMFC/numan}.$$

The interface between the tools is established by a function inserted in the MOGA code, which coordinates the full process of reading and exchange of endogenous and exogenous variables between the programs. The entire simulation is a single process that is coordinated within the main routine of MOGA. The simulation process plays an important role in the testing of the fitness of the operational scenario because it supplies the variables that represent the dynamics and randomness of the process.

The simulator generates partial inputs for the assessment of the scenario; however, it is not a function of fitness. The stop criterion introduced in the simulation model is the functional period of the automated system, i.e., the number of hours that the system must function during simulation.

The position of the simulator in the flowchart of the optimizer (NSGA-II), as explained by [6], has paramount importance for the employed simulation process. As Fig. 2 indicates, the MOGA is responsible for sequentially producing operational scenarios. The simulator receives one scenario at a time and simulates its operation more realistically the more detailed its respective model is. In a later stage, the MOGA assesses the fitness of the scenario according to the nondominance criterion mentioned above.

When an operational scenario is created and sent to the simulator, the configuration of the machines that will be activated in the process is established, and the following system exogenous variables are calculated as the system evolves over time: *TMVC,* R_s, *TPPS, SDT, TSA, TNSF, TGPPS, CPLF, CINST, CMTTPM, CMODC,* and *CGM.*

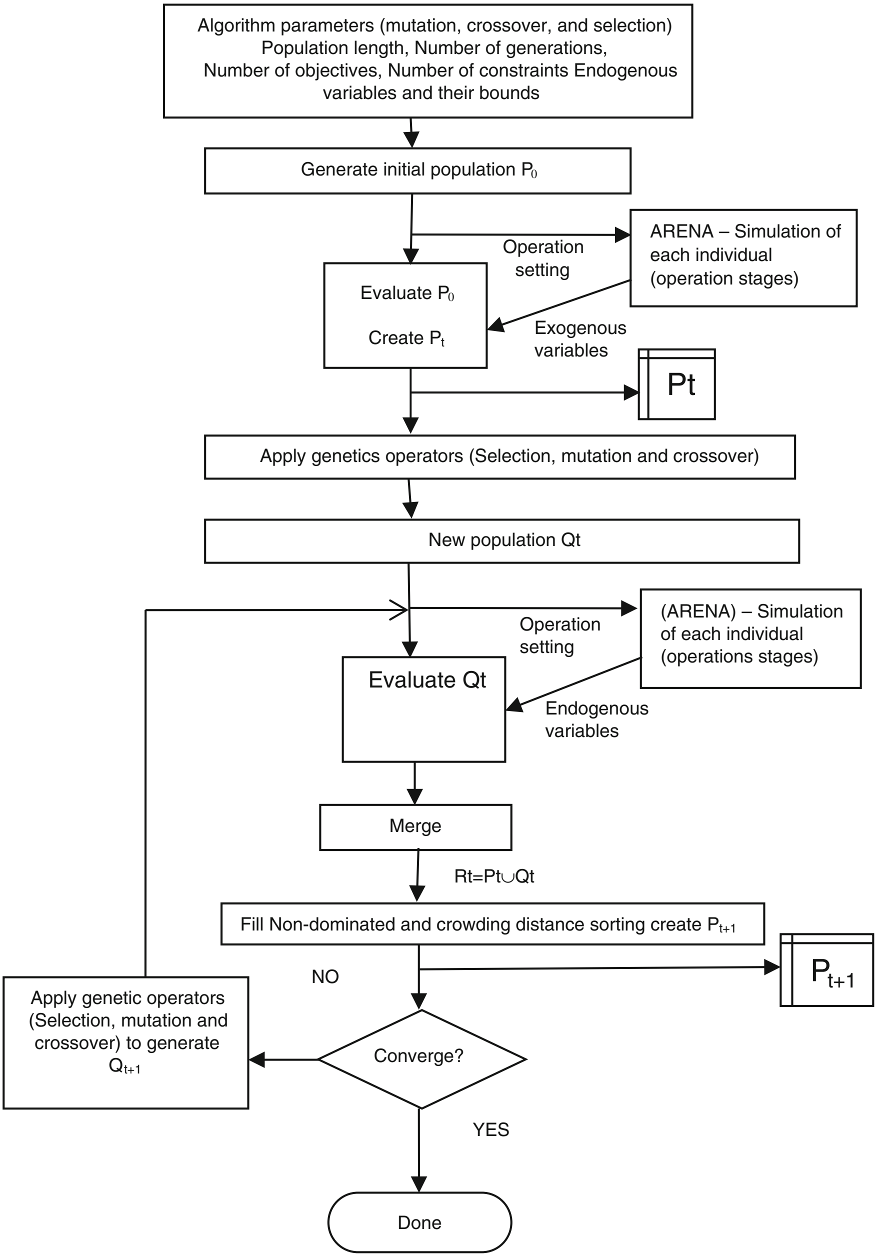


Fig. 2 Flowchart of simulation optimization framework

Some of these variables are included in the calculation of the objectives to be optimized, whereas other variables describe features related to system efficiency and the configuration of machines established in the operational scenario. Nevertheless, all of the variables are either directly or indirectly included in the MOGA-routine of the verification of the problem constraints.

In the routine, the values of the exogenous and endogenous variables described above are compared to limits that are preset by the analyst in the problem constraints. The satisfaction of the limits defines the fitness of the scenario under consideration. Scenarios are discarded when one or more restrictions are violated and admitted when all constraints and optimization objectives are satisfied.

Once the fitness of a scenario is established, the simulation process restarts with a new individual or operational scenario (previously generated in the MOGA). The global stop criterion of the process is the number of generations, where each generation consists of a set of individuals or operational scenarios. The MOGA is capable of handling only one generation at a time. This characteristic requires the inclusion of a waiting mechanism in the MOGA to allow for the simulator to perform the simulation of each individual or scenario of the corresponding generation.

8 Description of the Idealized Case

Actual systems, especially automated ones, are characterized by a wide interrelation of subsystems, such as pneumatic, hydraulic, microelectronic, and electrical systems. Actual systems are essentially complex sets with low human interference (automated). Due to such complexity, these systems exhibit several types of failure modes (e.g., electronic and mechanical) with random failure rates. Failure causes production loss or complete standstill.

Considering the high productivity of these systems, failure mitigation becomes a crucial activity that justifies the use of redundant operations. A well-allocated set of redundancies may ensure availability and high levels of systemic reliability.

To dimension a group of machines in an idealized automated production line, the process was applied to the design stage. Therefore, the present study considered an automated system represented by a generic and continuous production process. The process may contain as many as forty-five operations; nine of these operations form the core of the system functioning. There are also two warehouses; one warehouse contains production parts (left), and the other warehouse stores finished products (right), as shown in Fig. 2. There is also a conveyor that interconnects the system with a set of robots.

Each machine in the system corresponds to a different operation, such as milling, erosion, assembly, and adjustment, which must be performed in a specific sequence. The robots perform the transference and storage operations of the parts and finished products. The system as a whole (machines and robots) performs nine operations, as shown in Fig. 1.

The system can be represented schematically by a set of nine stages in series, corresponding to the nine basic operations of the system, as illustrated in Fig. 3.

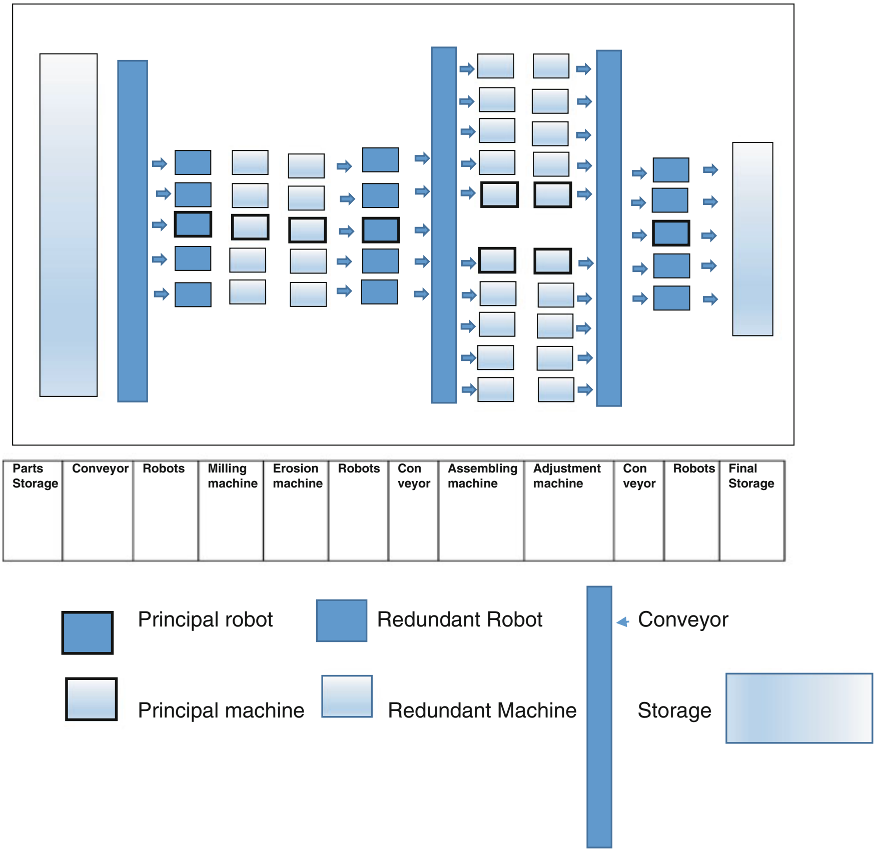


Fig. 3 Schematic diagram of the system

The series of stages represents a parallel system commonly used in RAP. Each stage or subsystem can be performed by as many as five machines that each have distinct operational characteristics. At least one of the five machines is assumed necessary for the basic functioning of the subsystem, whereas the remaining machines are redundancies that may or may not be allocated in the subsystem. The utilization and allocation of these redundancies is precisely what makes it possible to increase the entire availability of the system. The system can have a minimum configuration of nine machines and a maximum configuration of forty-five machines. The resulting system contains a large number of combinations, which warrants evaluation.

Each configuration or set of machines and robots chosen for operation has an associated number of maintainers, their corresponding unitary production costs,[2]

[2] The costs of production include

- CRAMA—cost of the raw materials applied
- CMPOLl—cost of the manpower of the operation of the line

individual reliability, and fixed maintenance costs corresponding to the identified configuration. This set of variables is called the operational scenario (configuration of machines, number of maintainers, individual production costs, and total fixed maintenance costs of the system). Operational scenarios with the same characteristics differ due to random events such as failures.

Therefore, the problem addressed in this study might be expressed with the following questions:

What is the optimal scenario of operation that maximizes total system reliability and minimizes simultaneously the use of maintainers, and the costs of maintenance and operation of the system?

9 Experiments

A total of six experiments were conducted, of which five were multi-objective and one was mono-objective:

- Three multi-objective experiments with 720, 1,440, and 2,160 simulated hours, respectively, and 50 generations, each comprised of 20 individuals (total of 1,000 individuals).
- One multi-objective experiment with 720 simulated hours and 100 generations, each comprised of 20 individuals (total of 2,000 individuals).
- One multi-objective experiment with 720 simulated hours and 150 generations, each comprised of 20 individuals (total of 3,000 individuals).
- One mono-objective experiment, which followed the model described in Sect. 6, with 720, 1,440, and 2,160 h of uninterrupted system operation.

Each individual corresponds to an operational scenario and each 720 h set corresponds to one month of uninterrupted system operation. The individuals were subjected to the process of evolution for the age that is guided by the reproduction operators. The probability of occurrence for the operators was 0.7 for the crossover operator and 0.02 for the mutation operator (real variables).

Such operators are responsible for the magnitude and diversity of the search and their occurrence probabilities were kept constant to ensure some homogeneity in the search across the experiments. Thus, substantial differences between the results of the five multi-objective experiments were avoided.

By keeping the occurrence probabilities of the genetic operators and the number of individuals per generation (20) in each experiment constant, the present study sought to demonstrate the influence of the increase in generations and the influence of the duration of the simulations within the boundaries of the experiments conducted.

(Footnote 2 continued)

- CEEOC—cost of electricity and other combustibles
- CPSAT—cost of packaging, storing, and transport.

In the case of multi-objective optimization, the algorithms used in the problems are expected to exhibit a set of solutions that satisfy the objectives. However, because there is a set of solutions, the final selection of a solution depends on a process to aggregate information, which can be performed *a priori*, *during the process*, or *a posteriori* [5].

In the *a priori* procedure, the criteria are established before the process is developed and guide the construction of the optimal Pareto set.

In the *during-the-process* procedure, the criteria are inserted during the development of the process and may include variables resulting either from previous iterations or inserted by the user.

In the *a posteriori* procedure, the process develops without interference by the decision-maker and, in the end, the information required to select a solution is added to the obtained Pareto set.

A procedure that aggregates information can exhibit a technical quantitative nature or a nontechnical qualitative nature. The former type of procedure treats the technical data as process specifications and variables related to the adjustment of the entire system, whereas the latter type of procedure expresses the opinions of the individuals who participate in decision-making.

In the study, the *a posteriori* procedure was chosen because the resulting Pareto set, which was prepared for the application of the selection criteria, became available at the end of the all experiments, as presented on the tables at Sect. 10.

10 Results—Analysis and Discussion

The main analysis of results is organized into two subanalysis: the firsts subanalysis concerns the convergence of the simulation optimization process and the second subanalysis concerns the process of selecting the best operational scenario. The latter subanalysis also sought to answer the questions posed at the end of section two:

> What is the optimal operational scenario for the idealized case under consideration? Is the system that maximizes total system reliability while minimizing operation and maintenance costs and the use of maintainers the ideal scenario?

Four objectives were assessed in each experiment. The assessment included indexes of reliability, cost, and number of maintainers. The reliability indexes varied between 0 and 1 whereas cost and number of maintainers could have any value, although the maintainers are integer variables. To compare the objectives, which have different natures, adjustments were needed. As a result, the following normalization process was adopted:

$$x_n \sqrt{\frac{\bar{x}_i}{\sum_{i=1}^{n} \bar{x}_i^2}}$$

where

x_n—value of a normalized point

$\bar{x}_i$—value of an average individual of generation i with $i = 1 \ldots n$
n—total number of generations.

10.1 Analysis of AGE Convergence

Figures 4c, 5b, 6b, 7b, and 8b were plotted by using the averages of the objectives for each generation to draw a line of tendency toward convergence, which illustrates reduction of the variability of the responses as a function of the evolution of the process. The MOGA approached its stop criterion (which in this case was the number of generations). The plotted graphs of the resulting Pareto boundaries' are shown in Figs. 4a,b, 5a, 6a, 7a, and 8a for each experiment and reveal an increase in the concentration of individuals and the number of generations. The experiments did not result in a larger number of feasible scenarios. An increase in the number of MOGA generations and computation times resulted in enhanced responses, similar to scenarios with lower costs and higher reliability indexes. The convergence of the genetic process is related to the number of generations, the probabilities of the genetic operators, the type of selection, and the stop criterion. Thus, the convergence of the process is certain, and a larger number of generations guarantees better results.

By keeping constant the number of generations under various simulation times, greater oscillation in the values of the objectives was observed in experiments with 1,440 and 2,160 h as compared with the experiment with 720 h, as shown in Figs. 7b and 8b, respectively. The outcome was due to the presence of undesirable stochastic events, such as failures, which altered the average availability of the system between simulations. Oscillation was expected because some modalities of failures were defined by exponential models with averages greater than 1,000 h. This finding indicates that the system exhibits natural degradation as a function of longer functioning periods, which is due to the presence of failures and other stochastic events that degenerate the system (e.g., reduced operation times and more frequent lines). Longer periods also cause degradation of the reliability of the system, as shown by the boundaries in Figs. 4b, 7a, and 8a, which represent the global system variables of reliability versus maintenance cost.

10.2 Analysis of the Selecting Process of the Best Operational Scenario

The mono-objective model (MOOM) was surpassed by the multi-objective model (MUOM) at SIMO framework for several reasons. The main reason is the impracticality of representing the stochastic characteristics inherent to the investigated system using the mono-objective model, although it produced reliability results similar to the averages of the optimal boundaries obtained by MOGA in each of the multi-objective experiments performed, as shown in Table 1. Reliability tended to decrease with longer simulation times because it is a function that indirectly reveals the aging

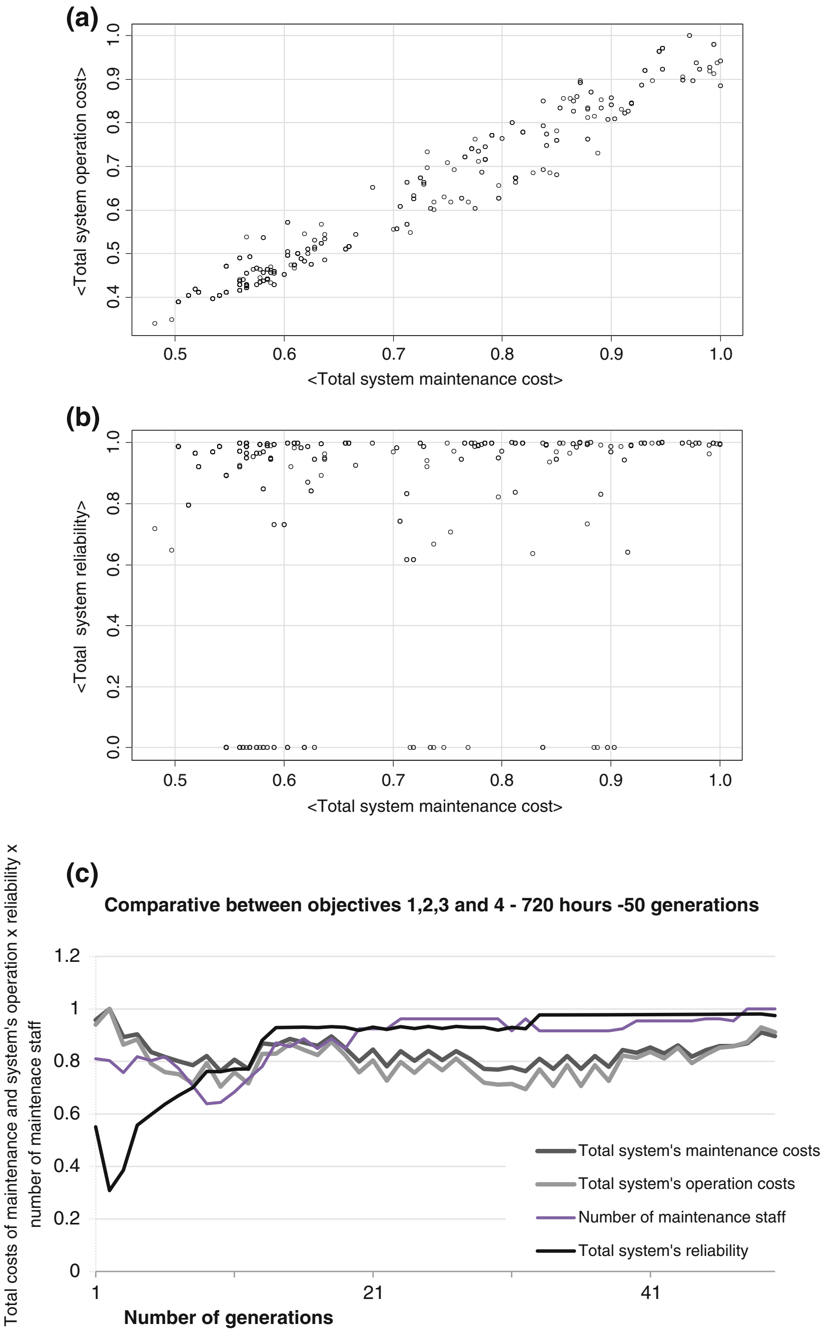


Fig. 4 **a** Pareto boundary objective 1 versus objective 3 for 720 h—50 generations; **b** Pareto boundary objective 2 versus objective 1 for 720 h—50 generations; **c** Experiment (simulation) for 1,720 h—50 generations

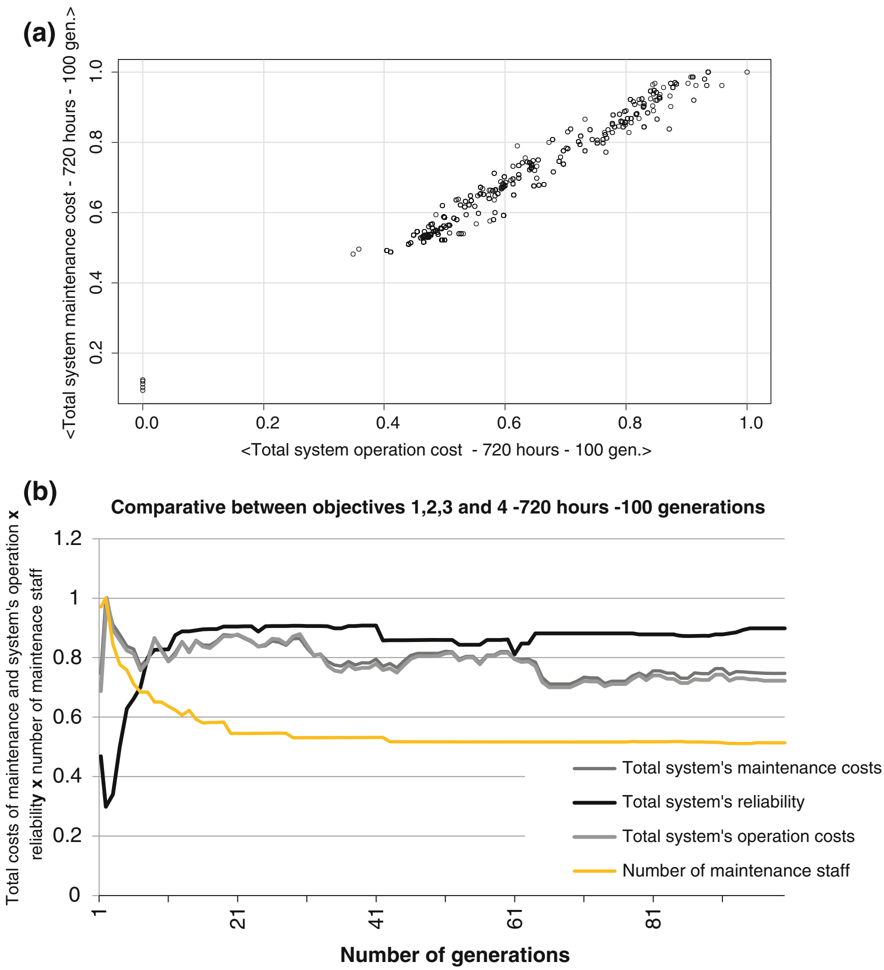
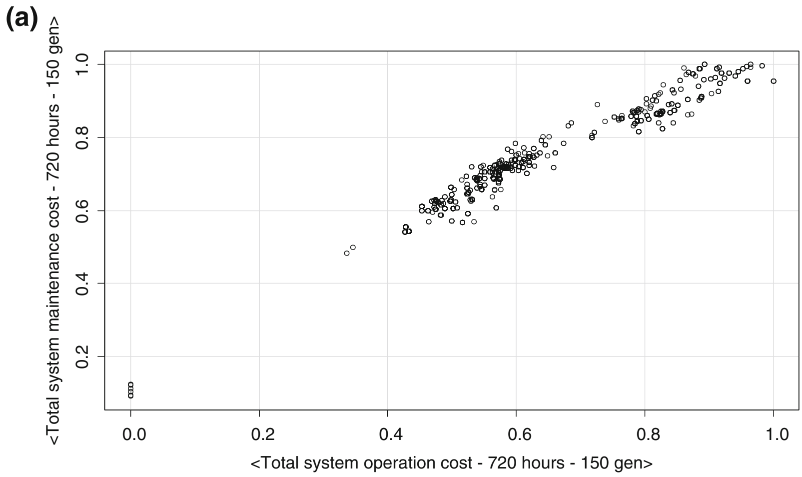


Fig. 5 **a** Pareto boundary objective 1 versus objective 3 for 720 h—100 generations; **b** Experiment (simulation) for 4,720 h—100 generations

of the system. The system is at a higher exposure to future risk of failure with a reduction in performance because the probability of failure increases with the passage of time. This situation can be prevented by adding redundancies to the system and improving the global reliability of the system as a function of the increase of alternative pathways for the process.

The costs obtained in the simplified mono-objective model were higher than the averages of the multi-objective simulation, as shown in Table 2.

Although a simplified model is easier to solve, it is not necessarily less complex. If the development of such a model required many simplifications, the accuracy of its responses would be impacted and the model would not serve its purpose. The consistency of the simplified model, as described in Sect. 6.1, is based on the



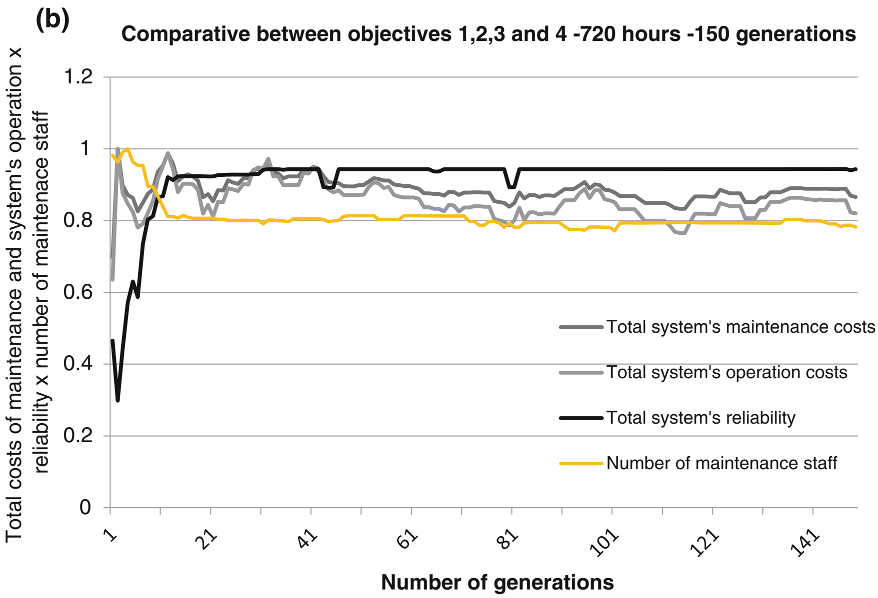


Fig. 6 **a** Pareto boundary objective 1 versus objective 3 for 720 h—150 generations; **b** Experiment (simulation) 5 for 720 h—150 generations

appropriation of the same cost formulation applied to the simulation model; however, without the influence of the full stochastic conjuncture (e.g., failures and reduction of process speeds), which the simulation sought to emulate. Therefore, the process costs in the simplified model originated exclusively from one specific configuration of machines and solely from the number of allocated machines. Figure 9 represents the average values of the main objectives by generation and illustrates that the results of

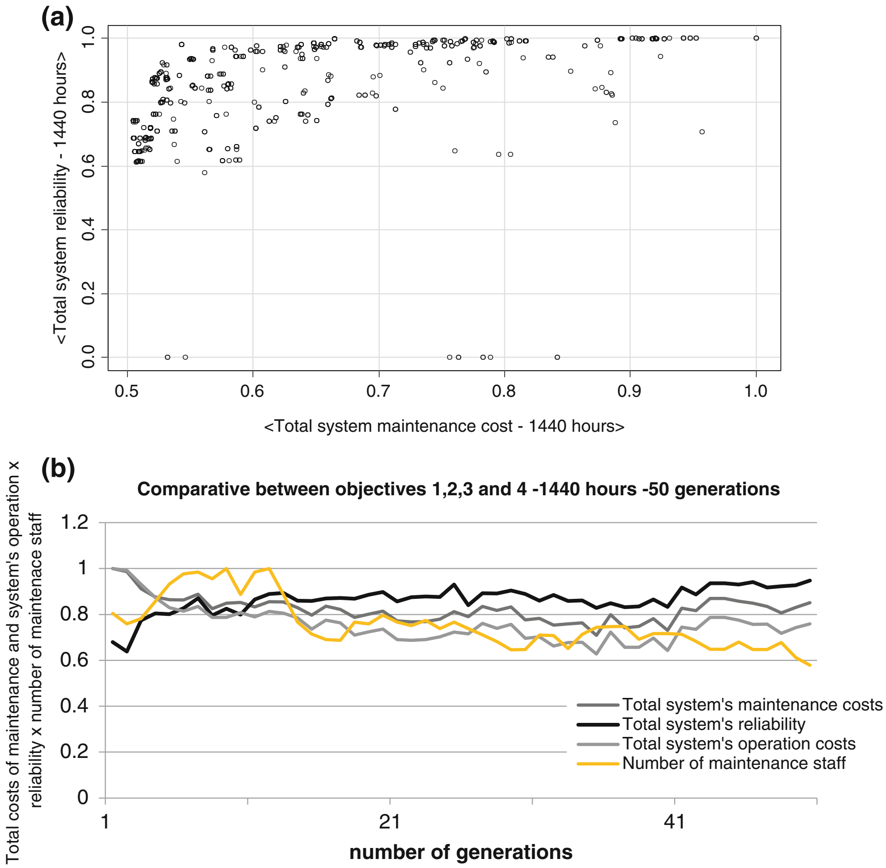


Fig. 7 **a** Pareto boundary objective 1 versus objective 3—1,440 h—50 generations; **b** Experiment (simulation) 2 for 1,440 h—50 generations

the simulation optimization process, which included the above-mentioned stochastic context, surpass the results of the simplified model (mono-objective) in terms of costs. The results accurately reflect the evolution of the responses not only by the level of randomness in the model but also by the level of randomness in the evolutionary process itself.

Of the 8,000 operational scenarios generated in the study, forty-four scenarios were selected by the MOGA that satisfied all objectives and constraints of the problem, as shown in Tables 3 and 6. In spite of the low number of simulations, satisfactory results were achieved. The selected scenarios were initially clustered according to the number of machines allocated in the process. According to that criterion, the minimum number of allocations was fifteen and the maximum number of allocations was forty-two, within a universe that ranged from a total of nine to forty-five

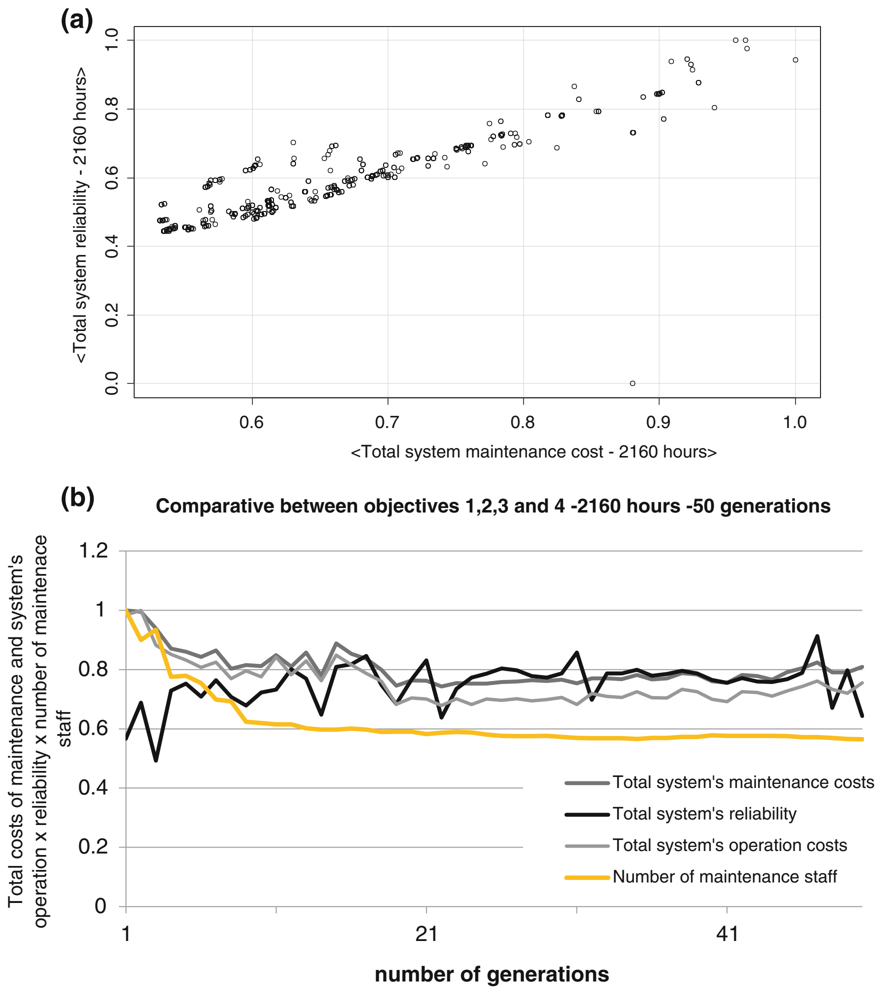


Fig. 8 **a** Pareto boundary objective 1 versus objective 3 for 2,160 h—50 generations; **b** Experiment (simulation) 3 for 2,160 h—50 generations

machines. By itself; however, this criterion did not facilitate the selection of the most appropriate operational scenario.

Another criterion that was considered was the reliability of the scenario. The reliability of the selected scenarios oscillated slightly, with a tendency toward improvement as the number of allocated machines increased (greater redundancy). All selected experiments exhibited high reliability (>0.98), which makes that criterion insufficient for decision-making.

The decision-maker might consider other variables of interest such as cost/product, maintenance costs (objective 1), and operational costs (objective 3), thus creating a novel criterion. Upon application of this criterion, the resulting minimum

Table 1 Reliabilities—multi-objective versus mono-objective experiments

	MUOM—SIMO FRAMEWORK		MOOM	
		Means*	720/1,440 h	2,160 h
Objective 2	720/50	0.870737	0.88	0.78
	1440/50	0.863031	0.88	0.78
	2160/50	0.752815	0.78	0.78
	720/100	0.999761	0.88	0.78
	720/150	0.999959	0.88	0.78

Table 2 Maintenance (objective 1) and operation (objective 3) costs of the system—multi- and mono-objective experiments

	MUOM—SIMO framework (mean values) Multi-objective model		MOOM Mono-objective model	
Simulations	Objective 1	Objective 3	Objective 1	Objective 3
720/50	2,071,950.00	2,257,797.00	2,522,403.00	4,123,482.00
720/100	2,014,479.00	2,166,809.00	2,522,403.00	4,123,482.00
720/150	2,371,427.00	2,592,107.00	2,522,403.00	4,123,482.00
1,440/50	1,962,392.00	2,022,221.00	2,700,065	5,902,222.00
2,160/50	1,892,339.00	2,010,936.00	2,930,879.00	7,734,114.00

cost was 5.4 C.U. (experiment 2 720/100/3 and 720/100/4), as shown in Table 2, and the resulting maximum cost was 11.19 C.U. (experiment 1 720/50/6), as displayed in Table 3, corresponding to 36 and 32 allocated machines, respectively (C.U. = currency units). When 36 machines were allocated, the six maintainers

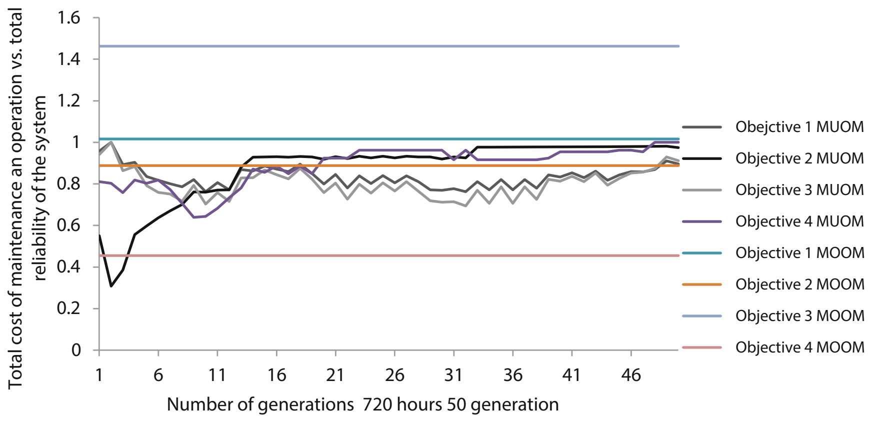


Fig. 9 Comparison between the average fitness of the objectives 1, 2, 3, and 4—multi-versus mono-objective simulation

Table 3 Experiment (simulation) for 1,720 h—50 generations

Experiment 1 (run 1)		720 h—50 generations Objective 1—Total system's maintenance cost Objective 2—Total system's reliability Objective 3—Total system's operation cost Objective 4—Total number of maintenance staff				
Operation set	Obj. 1	Obj. 2	Obj.3	Obj. 4	Machines assignments	Cost/ product
1	2,820,206	0.998928	3,624,749	16	35	6.45
2	2,792,698	0.998885	3,709,403	16	35	5.92
3	1,959,979	0.997946	2,079,463	16	35	8.23
4	1,606,541	0.985626	1,622,457	5	31	9.76
5	1,733,869	0.987398	1,676,912	16	31	10.39
6	1,809,541	0.996716	1,770,991	16	32	11.19
7	1,786,686	0.996919	1,775,866	16	33	10.11
8	2,468,443	0.998576	3,081,645	16	34	6.69
9	2,585,959	0.998631	3,326,377	16	35	6.28
Means	2,167,227.2	0.995818	2,481,908	12	33.5	8.33

were found in objective 4, as compared with sixteen maintainers found when 32 machines were allocated.

By adding to this criterion the requirement that the lowest desired number of maintainers must be less than seven with cost per product lower than 6.00 C.U. Considering the solutions described in Tables 3, 4, 5 and 6 and striving to answer the questions in Sect. 2, it was concluded that the most appropriate (optimal) scenarios for the idealized case were as shown in Table 7.

The costs differed for similar and identical numbers of allocated machines. This finding reveals an interesting characteristic of the process; specifically, that the quantity of machines, the position of each machine in the system, and the consequences of the random events to which they were subjected (failures) were all considered. This consideration causes alterations in production and exposes the system to failures that might result in longer standstill of an operation. In such cases, despite their quantitative accuracy, the results of the combination of machines or of the operational scenario to which they belong differ; consequently, their performance depends on the stochastic conjuncture that was produced.

11 Remarks

This chapter discusses the problem of redundancy and reliability allocation in an automated production system. The purpose of the study is to improve the global reliability of the system by allocating alternative components (redundancies) that

Table 4 Experiment (simulation) for 4, 720 h—100 generations

Experiment 4 (run 4)		720 h—100 generations Objective 1—Total system's maintenance cost Objective 2—Total system's reliability Objective 3—Total system's operation cost Objective 4—Total number of maintenance staff				
Operation set	Obj. 1	Obj. 2	Obj. 3	Obj. 4	Machines assignment	Cost/ product
1	1,967,423	0.999746	2,162,364	6	36	6.23
2	1,641,490	0.999063	1,801,222	5	35	5.65
3	1,744,774	0.999701	1,955,186	6	36	5.40
4	1,696,361	0.999691	1,893,462	6	36	5.40
5	1,691,796	0.999581	1,885,273	6	36	5.42
6	2,014,479	0.999761	2,166,809	6	35	7.00
Means	1,792,721	0.999591	1,977,386	6	35.6	5.85

Table 5 Experiment (simulation) 2–1,440 h—50 generations

Experiment 2 (run 2)		1440 h—50 generations Objective 1—Total system's maintenance cost Objective 2—Total system's reliability Objective 3—Total system's operation cost Objective 4—Total number of maintenance staff				
Operation set	Obj.1	Obj. 2	Obj. 3	Obj. 4	Machines assignment	Cost/ product
1	2,914,198	0.9993065	3,147,688	6	27	5.81
2	2,043,527	0.9985866	2,074,511	7	27	6.81
3	2,742,798	0.9987716	2,901,673	6	27	6.13
4	1,669,170	0.9797631	1,641,688	6	20	7.11
5	1,992,993	0.9837161	2,040,098	15	21	6.30
6	2,010,620	0.9800117	1,993,446	7	20	7.19
7	2,863,146	0.9991417	3,064,383	6	29	5.83
8	2,771,984	0.9990900	2,931,642	6	28	6.18
9	2,896,160	0.9993100	3,130,280	6	29	5.77
10	2,788,534	0.9988900	2,915,434	6	27	7.10
Means	2,469,313	0.9936590	2,584,084	7.1	25.50	6,42

are associated in parallel with each original component and to minimize the use of maintainers and the associated system maintenance and operation costs.

The simulation optimization process employed uses a model constructed with variables and constraints. The process differs from conventional optimization problems because there is no function (or set of functions) that represents the system.

Therefore, the model can assume various degrees of complexity that allow it to accurately represent actual industrial situations.

The method used is intimately related to the constraints imposed on the model, which result from the judgment of the modeler. The process benefits from and depends on the accuracy (quality) of the work of the modeler.

The setting of the operational scenarios and the choice of the best results is entirely automatic within an evolutionary process that is part of a genetic algorithm and based on the production of a large amount of simulations. The optimizing evolutionary process that was considered revealed few simulations in the discrete system (total of 8,000) that achieved satisfactory results. Of these 8,000 simulations, forty-four feasible scenarios were chosen by the MOGA, according to the imposed constraints. Thus, in spite of the small number of simulations, the proposed approach facilitates decision-making by permitting the generation of a reduced and feasible set of solutions, which is sufficient to alter the constraint limits and minimize the number of operational scenarios.

The method used was efficient in developing feasible solutions, in spite of the existence of multiple conflicting objectives, constraints, systemic interrelations, and random factors. For this reason, and considering the resulting high degree of reliability of the solutions (>0.98) for 720, 1,440, and 2,160 h of system functioning and the degree of generalization of the developed model, one might assert that the suggested method exhibits a wide scope of potential applications for actual industrial situations.

Table 6 Experiment (simulation) for 3–2,160 h—50 generations

Experiment 3 (run 3)		2160 h—50 generations				
		Objective 1—Total system's maintenance cost Objective 2—Total system's reliability Objective 3—Total system's operation cost Objective 4—Total number of maintenance staff				
Operation set	Obj.1	Obj. 2	Obj. 3	Obj. 4	Machines assignment	Cost/ product
1	1,658,352	0.981308	1,623,417	6	23	10.11
2	1,923,583	0.985386	1,899,560	6	20	11.15
3	2,157,052	0.993085	2,287,982	6	24	7.84
4	1,965,154	0.989701	2,058,740	6	21	7.91
5	2,217,531	0.994180	2,399,669	6	25	7.24
6	2,405,210	0.995831	2,726,515	6	25	6.10
7	2,145,893	0.993032	2,287,960	6	24	7.56
8	2,031,733	0.991144	2,119,967	6	22	8.29
9	2,239,416	0.994185	2,418,995	6	25	7.30
10	2,404,594	0.995893	2,728,048	6	25	6.10
Means	2114856	0.991330	2,255,058	6	23.4	7.96

Table 7 Final set of solutions after application of the decision-maker preference criteria

Simulation time/generations	Operation set number	Obj.1	Obj. 2	Obj. 3	Obj. 4	Machines assignment	Cost/product
720/100	2	1,641,490	0.999063	1,801,222	5	35	5.65
720/100	3	1,744,774	0.999701	1,955,186	6	36	5.40
720/100	4	1,696,361	0.999691	1,893,462	6	36	5.40
720/100	5	1,691,796	0.999581	1,885,273	6	36	5.42
1440/50	1	2,914,198	0.9993065	3,147,688	6	27	5.81
1440/50	7	2,863,146	0.999142	3,064,383	6	29	5.83

Although the results are satisfactory given the constraints and the small number of simulations used, the developed process and model requires further analysis due to the multiple possibilities that are feasible with more specific variables (to be included in the model).

The use of production systems with more complex structures or the selection of alternate objectives should be considered in future studies, such as:

- Optimization of the stock of maintenance parts;
- Optimization of the number of maintenance inspections to reduce periods of standstill;
- Optimization of the sequencing of tasks; and
- Optimization of the operational parameters to handle changes in productivity levels.

Acknowledgments To Dr. Deb and his team at the Kanpur Genetic Algorithms Laboratory (KanGAL) for providing the source code of the MOEA NSGA-II used in this work.

References

1. Azadivar, F. (1992). A tutorial on simulation optimization. In *Proceedings of the, (1992). 24th winter simulation conference*, pp. 198–204. doi:10.1145/167293.167332.
2. Billiton, R. A., Ronald N. (1992). Reliability evaluation of engineering systems: Concepts and techniques (2nd ed). Springer. ISBN-10 0306440636; ISBN-13 978–0306440632.
3. Coelho, L. S. (2009). Reliability-redundancy optimization by means of a chaotic differential evolution approach. *Chaos, Solitons and Fractals, 41*(2), 594–602. doi:10.1016/j.chaos.2008.02.028.
4. Coello, C. A., Lamont, G. B. (2004). Applications of multi-objective evolutionary algorithms. Advances in Natural Computation (Vol. 1). World Scientific Publishing. ISBN 981-256-106-4.
5. Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. England: Wiley. ISBN 047187339X, 9780471873396.
6. Deb, K., Pratap, A., & Agarwal, S. (2002). Fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE-Transactions on Evolutionary Computation, 6*(2), 182–197. doi:10.1109/4235.996017.
7. Ergott, M. & Gandibleux, X. (2004). Approximate solution methods for multi-objective combinatorial optimization. Sociedad de Estadística e Investigación Operativa. Madrid. Spain TOP Vol. 12. No. 1. 1–90 June. doi:10.1007/BF02578918.
8. Gen Mitsuo, Cheng Runwei (2000). Genetic Algortihms & Engineering Optimization. Canada. Wiley. A Wiley-interscience publication. ISBN 0471315311, 9780471315315.
9. Ghiani Gianpaolo, Legato Pasquale, Musmanno Roberto, Vocaturo Francesca. A combined procedure for discrete simulation-optimization problems based on the simulated annealing framework. Computational Optimization and Applications, September 2007, Volume 38, Issue 1, Pages 133.145. doi:10.1007/s10589-007-9010-7
10. Hani, Y., et al. (2008). Simulation base optimization of a train maintenance facility. *Journal of Intelligent Manufacturing, 19*(3), 293–300. doi:10.1007/s10845-008-0082-8.
11. Holland, John Henry (1975). Adaptation in natural and artificial systems: an introductory analysis eith applications to biology, control and artificial intelligence. University of Michigan Press. ISBN 0472084607, 9780472084609.
12. Jianping, L. I., & Xishen, J. (1997). A new partial bound enumeration technique for solving reliability redundancy optimization. *Microelectronics Reliability, 37*(2), 237–242. doi:10.1016/S0026-2714(96)00109-6.

13. Khalili-Damghani, K., & Maghsoud, A. (2012). Solving binary-state multi-objective reliability redundancy allocation series-parallel problem using efficient epsilon-constraint, multi-start partial bound enumeration algorithm, and DEA. *Reliability Engineering and System Safety, 103*, 35–44. doi:10.1016/j.ress.2012.03.006.
14. Kulturel-Konak, S., & Coit, D. W. (2007). Determination of Pruned Pareto sets for the multi-objective system redundancy allocation problem. *IEEE Symposium on computational intelligence in multicriteria decision making*, pp. 390–394. doi:10.1109/MCDM.2007.369118.
15. Kulturel-Konak, S., Smith, A. E., & Norman, B. A. (2006). Multi-objective tabu search using a multinomial probability mass function. *European Journal of Operational Research, 169*, 918–931. doi:10.1016/j.ejor.2004.08.026.
16. Kumar, R., Izui, K., Yoshimura, M., & Nishiwaki, S. (2009). Multi-objective hierarchical genetic algorithms for multilevel redundancy allocation optimization. *Reliability Engineering and System Safety*, *94*(4), 891–904. doi:10.016/j.ress.2008.10.002.
17. Kuo, W., Prasad, V. R., Tillman, F. A., & Hwang, C.-L. (2001). Optimal reliability design fundamentals and applications. Cambridge University Press. ISBN 0521781272, 9780521781275.
18. Kuo, W., Zuo, M. J. (2003). Optimal reliability modeling. Principles and applications. New Jersey: Wiley. ISBN 047139761X, 9780471397618.
19. Liang, Y.-C., & Lo, M.-H. (2010). Multi-objective redundancy allocation optimization using a variable neighborhood search algorithm. *Journal of Heuristics, 16*, 511–535. doi:10.1007/s10732-009-9108-4.
20. Lins, Isis Didier, & Droguett, Enrique López. (2011). Redundancy allocation problems considering systems with imperfect repairs using multi-objective genetic algorithms and discrete event simulation. *Simulation Modelling Practice and Theory, 19*, 362–381. doi:10.1016/j.simpat.2010.07.010.
21. Lisnianski, A., & Levitin, G. (2003). Multi-state system reliability Assessment optimization and application (Vol. 6). Singapore: World Scientific Publishing. Series on Quality reliability and Engineering statistics. ISBN 981-238-306-9.
22. Loo Hay Lee, Ek Peng Chew, Kee Hui Chua, Zhuo Sun, Lu Zhen. A Simulation Optimisation Framework for Container Terminal Layout Design. Multi-objective Evolutionary Optimisation for Product Design and Manufacturing 2011, Pages 385–400. doi:10.1007/978-0-85729-652-8_14.
23. Mattila, V., Virtanen, K. (2005). A simulation-based optimization model to schedule periodic maintenance of a fleet of aircraft. In *Proceedings 2005. European simulation and modeling conference*, Porto, Portugal, October 24–26, pp. 479–483.
24. Meedeniya, I., Aleti, A. & Buhnova, B. (2009). Redundancy allocation in automotive systems using multi-objective optimization. Jacobs school UCSD symposium on automotive/avionics systems engineering SAASE 2009. http://www.jacobsschool.ucsd.edu/GordonCenter/g_leadership/l_summer/s_saase/schedule.shtml
25. Mehmet, Çunkaş. (2010). Intelligent design of induction motors by multiobjective fuzzy genetic algorithm. *Journal of Intelligent Manufacturing, 21*(4), 393–402. doi:10.007/s10845-008-0187-0.
26. Merkuryeva, G., & Napalkova, L. (2008). Development of multi-objective simulation-based genetic algorithm for supply chain cyclic planning and optimization. In *International conference 204th EURO mini conference continuous optimization and knowledge-based technologies (EurOPT) May 20.23*. Neringa, Lithuania. ISBN 978-9955-28-283-9.
27. Misra, K. B., Sharma, U. (1991). An efficient approach for multiple criteria redundancy optimization problems. *Microelectronics Reliability, 31*(2/3), 303–321. doi:10.016/0026-2714(91)90216.
28. Murata. T. (1989). Petri Nets: Properties, Analysis and applications. Proceedings of the IEEE. Vol. 77. No 4. pp 541-580. doi: 10.1109/5.24143.
29. Nesmachnow, Sergio (2004). A parallel version of evolutionary algorithm for multiobjective optimization NSGA-II. Proceedings of X Argentine Computation Science Congress, Buenos Aires, Argentina, 2004.

30. Olafsson Sigurdur, Kim Jumi. Simulation Optimization. Proceedings of the IEEE Winter Simulation Conference, Pages 79–82, 2002.
31. Peterson. J.L. (1981) Petri Net Theory and the Modeling of Systems. Prentice-Hall. N.J. ISBN 0136619835.
32. Robin F, Orzati A, Moreno E, Homan O J, Bachtold W. Simulation and evolutionary optimization of electron-beam lithography with genetic and simplex-downhill algorithms. IEEE Transactions on Evolutionary Computation, Volume 7, Issue 1, February 2003, Pages 69–82.
33. Rozenberg Grzegorz, Reisig Wolfgang, Desel Jörg (2004). Lectures on concurrency and Petri Nets. Advances in Petri Nets. Lecture notes in Computer Science. Vol. 3098. doi: 10.1007/b98282.
34. Salazar, D., Rocco, C. M., & Galvan, B. J. (2006). Optimization of constrained multiple-objective reliability problems using evolutionary algorithms. *Reliability Engineering and System Safety, 91*(9), 1057–1070. doi:10.1016/j.ress.2005.11.040.
35. Shelokar, P. S., Jayaraman, V. K., & Kulkami, B. D. (2002). Ant algorithm for single and multi-objective reliability optimization problems. *Quality and Reliability Engineering International, 18*, 497–514. doi:10.1002/qre.499.
36. Soundharajan, B; Sudheer K.P. Sensitivity analysis and auto-calibration of ORYZA2000 using simulation-optimization framework. Paddy and Water Environment, January 2013 - Volume 11, Issue 1–4, Pages 59–71. doi:10.1007/s10333-011-0293-z.
37. Swisher, J. R., Hyden, P. D., Jacobson, S. H., & Schruben, L. W. (2000). A survey of simulation optimization techniques and procedures. In *Proceedings of 2000. Winter simulation conference* (Vol. 1, No. 10–13, pp. 119–128). Dec 2000. doi:10.1109/WSC.2000.899706.
38. Taboada, H. A. & Coit, D. W. (2008). Development of multiple objective genetic algorithm for solving reliability design allocation problems. In J. Fowler, & S. Masons (Eds), *Proceedings of the 2008 industrial engineering research conference*. Available at: http://ie.rutgers.edu/resource/research_paper/paper_08-004.pdf
39. Taboada, H. A., Baheranwala, F., Coit, D. W., & Wattanapongsakorn, N. (2007). Practical solutions for multi-objective optimization: An application to system reliability design problems. *Reliability Engineering and System Safety, 92*, 314–322. doi:10.1016/j.ress.2006.04.014.
40. Taboada, H. A., & Coit, D. W. (2012). A new multiple objective evolutionary algorithm for reliability optimization of series-parallel systems. *Applied Evolutionary Computation, 3*(2), 1–18. doi:10.4018/jaec.2012040101.
41. Tian, Shigang, & Suo, Ming J. (2006). Redundancy allocation for multi-state systems using physical programming and genetic algorithms. *Reliability Engineering and System Safety, 91*, 1049–1056. doi:10.1016/j.ress.2005.11.039.
42. Tian, Z., Lin, D., & Bairong, W. (2012). Condition based maintenance optimization considering multiple objectives. *Journal of Intelligent Manufacturing, 33*(2), 333–340. doi:10.1007/s10845-009-0358-7.
43. Tzu-Chieh, H. & Kuei-Yuan, C. (2011). Uncertainity quantifications of Pareto optima in multiobjective problems. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-011-0602-9.
44. Ullrich, Christian. Exchange Rate Hedging in a Simulation/Optimization Framework. Forecasting and Hedging in the Foreign exchange Markets. Lectures Notes in Economics and Mathematical Systems, Volume 623, 2009, Pages 185–188. doi:10.1007/978-3-642-00495-7_17.
45. Wang, Z., & Chen T., Tang K., Yao X. (2009). A multi-objective approach to redundancy allocation problem in parallel-series systems. *IEEE Congress on evolutionary computation–CEC*, pp. 582–589. doi:10.1109/CEC.2009.4982998.
46. Yang Taho, Chou Pohung. Solving a multiresponse simulation-optimization problem with discrete variables using a multiple-attribute decision-making method. Mathematics and Computers in Simulation, Volume 68, Issue 1, 3 February 2005, Pages 9–21.

47. Zhou, L., Yoke San, W., & Kim Seng, Lee. (2011). A manufacturing-oriented approach for multi-platforming product family design with modified genetic algorithm. *Journal of Intelligent Manufacturing*, *22*(6), 891–907. doi:10.1007/s10845-009-0365-8.
48. Zio, E. & Zille, V. (2007). Multi-objective optimization of network systems by using ANT algorithms. *Quality Technology of Quantitative management. 4*(2), 211–224. Available at: http://web.it.nctu.edu.tw/qtqm/qtqmpapers/2007V4N2/2007V4N2_F4.pdf

http://www.springer.com/978-3-319-15032-1

Applied Simulation and Optimization
In Logistics, Industrial and Aeronautical Practice
Mujica Mota, M.; Flores De La Mota, I.; Guimarans
Serrano, D. (Eds.)
2015, XV, 319 p. 138 illus., 80 illus. in color., Hardcover
ISBN: 978-3-319-15032-1