



Decision Support

Increasing the efficiency in integer simulation optimization: Reducing the search space through data envelopment analysis and orthogonal arrays



Rafael de Carvalho Miranda^{a,*}, José Arnaldo Barra Montevechi^a, Aneirson Francisco da Silva^b, Fernando Augusto Silva Marins^b

^aFederal University of Itajubá (UNIFEI), Avenida BPS, 1303 – Caixa Postal: 50 – Itajubá, MG 37500-903, Brazil

^bSao Paulo State University (UNESP), Av. Dr. Ariberto Pereira da Cunha, 333, Guaratinguetá, SP 12.516-410, Brazil

ARTICLE INFO

Article history:

Received 29 December 2015

Accepted 5 April 2017

Available online 11 April 2017

Keywords:

Data envelopment analysis

Simulation

Integer simulation optimization

Super-efficiency

ABSTRACT

The development of various heuristics has enabled optimization in simulation environments. Nevertheless, this research area remains underexplored, primarily with respect to the time required for convergence of these heuristics. In this sense, simulation optimization is influenced by the complexity of the simulation model, the number of variables, and by their ranges of variation. Within this context, this paper proposes a method capable of identifying the best ranges for each integer decision variable within the simulation optimization problem, thereby providing a reduction in computational cost without loss of the quality in the response. The proposed method combines experimental design techniques, Discrete Event Simulation, and Data Envelopment Analysis. The experimental designs called orthogonal arrays are used to generate the input scenarios to be simulated, and super-efficiency analysis is applied in a Data Envelopment Analysis model with variable returns to scale to rank the input scenarios. The use of the super-efficiency concept enables to distinguish the most efficient input scenarios, which allows for the ranking of all the orthogonal array scenarios used. The values of the variables of the two input scenarios that present the highest values of super-efficiency are adopted as the new range of the optimization problem. To illustrate this method's use and advantages, it was applied to real cases associated with integer simulation optimization problems. Based on the results, the effectiveness of this approach is verified because it delivered considerable reductions in the search space and in the computational time required to obtain a solution without affecting the quality.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

In the last 40 years, Discrete Event Simulation has increasingly been used by decision-makers in a wide range of fields (Banks, Carson II, Nelson, & Nicol, 2009; Law, 2015). The flexibility, versatility, and potential for analysis offered by this method have driven the adoption of simulation (Jahangirian, Eldabi, Naseer, Stergioulas, & Young, 2010; Xu, Huang, Chen, & Lee, 2015). Identified as one of the most commonly used research techniques in multiple sectors, simulation enables the study of complex systems in a more rapid, more flexible, and more economic fashion than experimentation with real systems, which consumes a greater amount of resources (Law, 2015; Shen & Wan, 2009).

One drawback to simulation is the fact that it provides statistical estimations for a single unique problem unless a sensitivity analysis is applied (Hillier & Lieberman, 2010). Without optimization software, analysts would be forced to simulate each input scenario individually (with different numerical and logical programming requirements that must be altered manually by the analyst in each simulation run) based on the real-world situation to search for the simulation model with variable arrangements that produces the most desirable results (Law & McComas, 2002).

With the aim of overcoming this hurdle, the concept of simulation optimization appeared in the 1990s as a viable option (Fu et al., 2000; Pasupathy & Ghosh, 2014) and has been widely accepted in the market, as evidenced by the availability of simulation software with integrated optimization routines (Alrabghi & Tiwari, 2015; Banks et al., 2009; Better, Glover, & Kochenberger, 2015; Figueira & Almada-Lobo, 2014; Fu, 2015; Fu et al., 2014; Kleijnen, 2015).

Although advancements have occurred in the realm of optimization, a common criticism that remains is that when these

* Corresponding author.

E-mail addresses: rafael.miranda@unifei.edu.br (R.d.C. Miranda), montevechi@unifei.edu.br (J.A.B. Montevechi), aneirson@feg.unesp.br (A.F. da Silva), fmarins@feg.unesp.br (F.A.S. Marins).

commercial software packages are presented with more than one decision variable, the solution time usually grows significantly (Harrel, Ghosh, & Bowden, 2004; Tenne & Goh, 2010). Furthermore, according to Hillier and Lieberman (2010), despite advancements in optimization software, simulation is still considered a relatively slow and costly means of studying chaotic and dynamic systems. Complex random systems require great investment in terms of money and time for programming and analysis, without even considering computational costs.

In an attempt to move beyond this obstacle, many studies have concentrated on development of methods for optimization, i.e., metaheuristics, which increase optimization efficiency by seeking solutions within acceptable computational time (Bachelet & Yon, 2007; Dellino, Kleijnen, & Meloni, 2010; Lin, Sir, & Pasupathy, 2013; Martins et al., 2013; Siegmund, Bernedixen, Pehrsson, Ng, & Deb, 2012; Willis & Jones, 2008). However, few studies have examined placing a limit on the optimization search space and thus working only with the most significant variables and the best range of values for each variable. Among these investigations, factorial designs stand out in their attempt to identify the significant variables for problems and concentrate the optimization process on these variables (Besseris, 2012; Kleijnen, 1998; Kleijnen, 2017; Montevechi, Miranda, & Friend, 2012).

Therefore, this study aims at defining the best variation ranges for each integer decision variable, thus reducing the search space and the computational time involved in the optimization process through simulation while reaching an optimal solution. In a way to meet its objective, this study proposes a method to integrate Orthogonal Arrays, Discrete Event Simulation, and Data Envelopment Analysis (DEA) with variable returns to scale (DEA BCC) combined with the concept of super-efficiency.

Orthogonal arrays are used in this study for a pre-assessment of the search space of the simulation optimization problems, because these arrays are experimental designs covering several variables with multiple levels; the orthogonal arrays are popular in Taguchian optimization; see Taguchi (1987).

Data Envelopment Analysis (DEA) consists of a methodology for a comparative measurement of the efficiency of Decision Making Units (DMUs) (Cooper, Sieford, & Tone, 2007). The classic DEA models developed by Charnes, Cooper, and Rhodes (1978) consider a constant return to scale, it is named DEA CCR, referring to the first letters of the surnames of their authors. Banker, Charnes, and Cooper (1984) had modified this DEA model to include a variable return to scale, this model is named DEA BCC, referring to the first letters of the surnames of their authors.

The DEA enables the identification of reference DMUs and generate an efficient border. In classic DEA models, DMUs with efficiency indexes equal to one are considered efficient, while those with efficiency indexes lower than one are inefficient. However, several DMUs may have efficiency indexes equal to one.

In this case and with a view to deal with such limitation, Andersen and Petersen (1993) have proposed the super-efficiency concept. The method promotes an alteration in the formulation of the DEA BCC model, allowing efficient DMUs to have efficiency indexes over one, and, this way, enabling to rank the most efficient DMUs. Therefore, based on the super-efficiency ranking, it is possible to reduce the search space of the optimization problem through the identification of the orthogonal array most efficient scenarios and, as consequence, incrementing the process efficiency.

Recently, Miranda, Montevechi, Silva, and Marins (2014) proposed a new Fuzzy-DEA-BCC model to obtain optimum variation intervals for decision variables, to improve DEA's discrimination power under occurrence of uncertainty and seeking for reduction in search space and computational solution times when compared to conventional simulation optimization techniques. They adopted orthogonal array to generate the necessary quantity of DMUs, and

the output variables were generated by simulation. The Fuzzy-DEA-BCC model was applied to two mono- and multi objective problems, and it presented good results which confirmed its reliability and applicability. However, it can be difficult to apply this procedure to large-scale problems because it involves mathematical complexity that requires advanced knowledge of Operations Research techniques from the analyst involved in the simulation.

Here, we proposed the classical DEA BCC model, also using orthogonal array, as simpler and innovative alternative than the Fuzzy DEA BCC model (Miranda et al., 2014) to analyze the super-efficiency. The new procedure reduces search space in the optimization problems without compromising the solution quality, and it allows solving real, complex, and large-scale problems, with substantial decrease in the associated computational time.

To address the proposed objectives, this paper is organized into four more sections. Section 2 presents the literature review upon which the study is based, Section 3 describes the proposed optimization procedure, Section 4 applies the proposal to simulation models and discusses the obtained results, and finally, Section 5 summarizes the study conclusions.

2. Literature review

2.1. Simulation optimization

The overwhelming majority of simulation optimization problems in such fields as manufacturing, logistical, and transport systems; supply chains, medical fields, and other applications are too complex to be modeled analytically. In these cases, simulation provides a useful method for evaluating the performance of these complex systems but can present deficiencies when applied in isolated situations. Thus, in many cases, the coupling of simulation and optimization is necessary (Lee, Chew, Teng, & Chen, 2008).

According to Fu (2002), Simulation Optimization can be defined as the process of seeking out the best combination of the decision variables.

Swisher et al. (2000) goes on to state that the simulation model decision variables domain may be both discrete, continuous, or a mixture of these two. The type of decision variable as well as its variation range will influence the search space and the optimization method that will be used.

Additionally, according to Nelson (2010), many decision variables in simulation optimization problems are discrete, which increases the difficulty in obtaining solutions and a gap exists in the literature on this topic.

Fu (1994) defines a traditional minimization problem with a single objective via simulation as:

$$\min f(\theta) \quad (1)$$

subject to: $\theta \in \Theta$

with $f(\theta) = E[\psi(\theta, \omega)]$ as the expected value for the estimated system performance based on the simulation model, $\psi_j(\theta, \omega)$ indicating the observed performance values derived from the discrete or continuous decision variables belonging to the feasible set Θ , ω represents the pseudorandom numbers (PRNs), θ is a controllable vector of p parameters, and Θ is the constraint set on θ .

Kleijnen, Van Beers, and Van Nieuwenhuysse (2010) wrote that simulation optimization problems are generally difficult to solve or NP-Hard and claim the disadvantages lie in the fact that simulation model output data originate from implicit functions and are exposed to noise caused by the PRNs. Furthermore, depending on the number of decision variables in the simulation model and the ranges of the decision variables, the optimization process can become complex due to the computational demands and the time involved in convergence of these algorithms.

2.2. Data envelopment analysis (DEA)

According to Cook and Seiford (2009), DEA is a methodology used to perform a comparison of the relative efficiencies of DMUs. Another important characteristic of the DEA models is that they allow consideration of incommensurability, that is, the presence of different measurement units for the input and output matrix elements (Tamiz, Jones, & El-Darzi, 1995).

The DEA models enable users to identify DMUs that are benchmarks for the other models under analysis, thus creating an efficiency boundary. The classic DEA CCR and DEA BCC models are defined as follows.

In their original model, Charnes et al. (1978) used Fractional Programming to obtain the input and output variable weights, as shown in (2)–(5):

$$\max w_0 = \sum_{r=1}^s u_r y_{r0} / \sum_{i=1}^m v_i x_{i0} \tag{2}$$

subject to:

$$\left(\sum_{r=1}^s u_r y_{rj} / \sum_{i=1}^m v_i x_{ij} \right) \leq 1, \quad j = 1, 2, \dots, n \tag{3}$$

$$u_r \geq 0, \quad r = 1, 2, \dots, s \tag{4}$$

$$v_i \geq 0, \quad i = 1, 2, \dots, m \tag{5}$$

where j are the DMU indices; n is the total number of DMUs; $r \in \{1, \dots, s\}$ is the output index, where s is total the number of outputs; $i \in \{1, \dots, m\}$ is the input index, where m is the total number of inputs; y_{rj} is the r th output for the j th DMU; x_{ij} is the i th input for the j th DMU; u_r is the weight associated with the r th output; v_i is the weight associated with the i th input; w_0 is the relative efficiency of DMU₀, which is the DMU under evaluation; and y_{r0} and x_{i0} are respectively the outputs and inputs for DMU₀.

It can be observed that when $w_0=1$, DMU₀ is efficient compared with the other units considered in the model; $w_0 < 1$ implies that the DMU is inefficient.

The model given by (2)–(5) is not linear; however, it can be linearized as shown in (6)–(10), thus generating the input-oriented DEA CCR model (Charnes et al., 1978) that presents a constant return of scale:

$$\max w_0 = \sum_{r=1}^s u_r y_{r0} \tag{6}$$

subject to:

$$\sum_{i=1}^m v_i x_{i0} = 1 \tag{7}$$

$$\sum_{r=1}^s u_r y_{rj} - \sum_{i=1}^m v_i x_{ij} \leq 0, \quad j = 1, 2, \dots, n \tag{8}$$

$$u_r \geq 0, \quad r = 1, 2, \dots, s \tag{9}$$

$$v_i \geq 0, \quad i = 1, 2, \dots, m \tag{10}$$

In a similar procedure, it is possible to formulate the output-oriented DEA CCR model (Charnes et al., 1978).

For cases in which it is interesting to consider variable returns of scale, the input-oriented DEA BBC model (Banker et al., 1984) given by (11)–(16) was developed, and the main difference between the DEA BCC and DEA CCR models is the addition of the free variable c_0 :

$$\max w_0 = \sum_{r=1}^s u_r y_{r0} + c_0 \tag{11}$$

subject to:

$$\sum_{i=1}^m v_i x_{i0} = 1 \tag{12}$$

$$\sum_{r=1}^s u_r y_{rj} - \sum_{i=1}^m v_i x_{ij} + c_0 \leq 0, \quad j = 1, 2, \dots, n \tag{13}$$

$$u_r \geq 0, \quad r = 1, 2, \dots, s \tag{14}$$

$$v_i \geq 0, \quad i = 1, 2, \dots, m \tag{15}$$

$$c_0 \text{ free} \tag{16}$$

In a similar procedure, it is possible to formulate the output-oriented DEA BCC model (Banker et al., 1984).

Banker et al. (1984) commented that a DMU that is considered efficient by BCC also would be considered efficient by the CCR model; however, the inverse of this statement will not necessarily be true. Furthermore, according Cooper et al. (2007), the DEA BCC model is invariant to the application of linear transformations to the inputs and/or outputs values involved, which does not occur in the DEA CCR model.

The DEA BCC model was chosen for use with the methodology proposed herein, primarily due to the stochastic and non-linear nature of simulation and the multiple inputs and outputs in the simulation models. These inputs and outputs are the DMUs for the DEA model, as shown in Section 4.1, and have different returns of scale.

Moving forward, due to its property of invariance to linear transformations, the DEA BCC model allows the use of data that present negative values, which can occur in stochastic simulation models, e.g., in the case of a variable associated with profit that could turn negative in the case of financial loss. The solution proposed in the literature for this case is to add a positive value to the inputs and/or outputs values of each DMU.

Finally, according to Azadeh, Moghaddam, Asadzadeh, and Negahban (2011), if the output values are given in the form of ratios, such as production, productivity and waste rates, and these are the type of problems addressed in this work, the DEA BCC model is the most frequently applied.

For Banker, Charnes, Cooper, Swarts, and Thomas (1989), to enable suitable discrimination of the DMUs in a traditional DEA model, one should verify whether the number of DMUs meets (golden rule):

$$n \geq \text{Max} \{ (ms), 3(m + s) \} \tag{17}$$

where n is the total number of DMUs; m is the total number of inputs and s is total the number of outputs.

Classic DEA models consider DMUs with $w_0=1$ as efficient and DMUs with $w_0 < 1$ as inefficient. This situation occurs if there is a possibility that multiple DMUs may have $w_0=1$; i.e., it is difficult to differentiate between DMUs. To address this limitation, Andersen and Petersen (1993) proposed a concept known as super-efficiency, which determines the difference between DMUs that present $w_0=1$. For (Bal et al., 2010) the use of super-efficiency allows an increase in the DMU discrimination. This concept will be used in this study together with the techniques covered in the following sections.

For use of the super-efficiency approach in a DEA BCC model, Andersen and Petersen (1993) suggest the removal of restriction (13) from the model but only for the DMU under analysis, thus enabling the DMUs to attain levels greater than 1, which may facilitate the elaboration of an efficiency ranking for those DMUs that originally presented $w_0=1$.

Table 1
Taguchi's orthogonal array and full factorial design.

Orthogonal array L16 (4^4)				Full factorial (2^4)			
A	B	C	D	A	B	C	D
1	1	1	1	1	1	1	1
1	2	2	2	4	1	1	1
1	3	3	3	1	4	1	1
1	4	4	4	4	4	1	1
2	1	2	3	1	1	4	1
2	2	1	4	4	1	4	1
2	3	4	1	1	4	4	1
2	4	3	2	4	4	4	1
3	1	3	4	1	1	1	4
3	2	4	3	4	1	1	4
3	3	1	2	1	4	1	4
3	4	2	1	4	4	1	4
4	1	4	2	1	1	4	4
4	2	3	1	4	1	4	4
4	3	2	4	1	4	4	4
4	4	1	3	4	4	4	4

2.3. Orthogonal arrays

Ballantyne, Van Oorschot, and Mitchell (2008) note that the process of simulation optimization for complex systems with multiple input variables can require a considerable amount of time and consume extensive resources in terms of personnel and financial impacts. For systems with only a few levels, Antony (2006) recommends the use of factorial strategies, i.e., full, or fractional designs, which typically work with two levels, i.e., one lower and one upper level (Montgomery, 2009).

Complex processes involving four or more input variables with three or more levels require a larger number of experiments for thorough optimization. One alternative approach is the use of orthogonal array (Taguchi, 1987), which reduces the number of necessary experiments and makes it possible to analyze and determine the most important input variables.

Taguchi, Chowdhury, and Wu (2005) define the orthogonal array as a matrix of numbers arranged in lines and columns in a way that each pair of columns is orthogonal to the other. When used in an experiment, each line represents an experimental situation, and each column a specific factor or condition that may be altered. According to Taguchi et al. (2005) an orthogonal matrix enables the evaluation of the impact of each parameter independently.

In fact, this array allows for the consideration of k input variables with l levels while testing each level of each variable in a balanced manner (Ross, 1996). In these arrays, the number of experiments needed is recognized by the index that accompanies the L abbreviation (Latin Square). A L8 experiment, for instance, refers to an experiment with eight experiments, disregarding any replications (Ross, 1996). For Roy (2010), orthogonal arrays usually define other ratings that show the number of factors involved in each array. For instance, the orthogonal array L16 (4^4) foresees the performance of 16 experiments in 4 columns, being each column the combination of 4 levels.

In this paper, orthogonal array is used as a pre-evaluation phase of the search space during optimization to generate a more diversified experimental matrix. The orthogonal arrays are selected as a function of the number of variables involved and their ranges, and the experimental matrix guides the running of the simulation input scenarios. The efficiency for each input scenario is determined by a DEA BCC model.

As an example, Table 1 presents an experimental matrix for a L16 (4^4) orthogonal array and full factorial design (2^4) for four input variables (A, B, C, D) and four levels (1, 2, 3, 4).

The example shows that the orthogonal array explores all four levels for all four input variables that compose the feasible region

in a more diversified manner, whereas the full factorial experimental matrix examines only two levels for each input variable these two levels are the levels 1 and 4 (excluding the levels 2 and 3). The discussion about the power of analysis of both experimental designs is beyond the scope of this paper, since only the experimental arrays are used to represent the feasible region of the optimization problem.

3. Proposed simulation optimization method

The procedure proposed in this paper includes the following assumptions:

- The use of orthogonal arrays permits to explore the feasible region of the simulation optimization problem.
- By ranking of the most efficient DMUs, the application of DEA in the orthogonal array makes it possible to reduce the search space of the simulation optimization problem.
- With the search space reduction, the computational time required for solving the simulation optimization problem will be less than that associated with the full search space to obtain solutions that are statistically equivalent.

The proposed method for effective reduction of search space during simulation optimization in this work makes use of four techniques:

- (1) Orthogonal array is applied to select the experimental matrix and generate input scenarios.
- (2) Simulation is used to obtain the input/output data.
- (3) A DEA BCC model is adopted to analyze the efficiency of each generated input scenario (DMUs) and to rank the DMUs.
- (4) Simulation optimization is used to obtain the solution.

Fig. 1 (adapted from Miranda et al., 2014) presents the proposed step-by-step approach for optimization and assumes that the simulation model has been built, programmed, verified, and validated. Furthermore, all decision variables are integers.

The steps are described as follows:

- (1) Determine the decision variables ($x_1, x_2, x_3, \dots, x_n$) and the input variable ranges for each variable.
- (2) Determine the estimator of the objective to be optimized (y).
- (3) Select the Taguchi's orthogonal array (L4, L9, L12, L16, L25, L32, L54, ...) as a function of the number of decision variables and their ranges as well as the minimum number of DMUs to be analyzed using DEA BCC with the purpose of meeting the basic rule for number of DMUs (Banker et al., 1989).
- (4) Execute the chosen orthogonal array combining original simulation inputs.
- (5) Execute the experiments and store the simulated data for analysis.
- (6) Determine each input scenario's super-efficiency by application of DEA BCC for the simulated results.
- (7) Rank the most efficient DMUs.
- (8) Set new ranges of variation for each decision variable based on two most efficient DMUs. Remove from the optimization process, the variables that possess equal values in both most efficient DMUs.
- (9) Optimize the simulation model using the new ranges for the remaining decision variables.

It should be noted that the first two steps of the proposed method are part of the definition of formulating the optimization problem, while the remaining steps are part of the method of solving the problem. In Fig. 1, the activities established to the right of

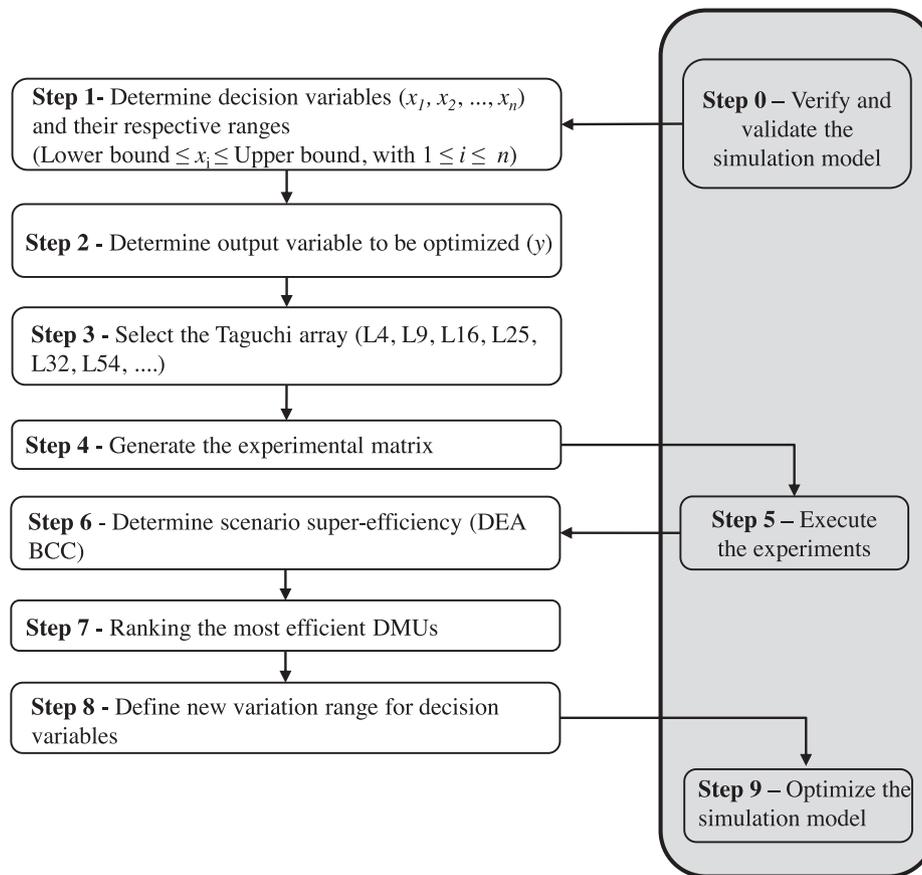


Fig. 1. Proposed optimization method (adapted from Miranda et al., 2014).

the main flow (steps 0, 5 and 9) are performed in the simulator or in the optimization module coupled to the simulator.

To exemplify the application of the procedure from Fig. 1, real cases will be presented in the next section. In accordance with the assumptions of this study, the simulation models were validated and verified and thus are suitable for simulation optimization.

4. Application of the optimization procedure, results, and discussion

According to Ahuja, Magnanti, and Orlin (1993), there are three possible approaches for complexity analysis of the proposed procedure: Empirical Analysis, Average-Case Analysis, and Worst-Case Analysis. This work adopted the Empirical Analysis for testing the performance of the proposed procedure on certain classes of problem instances.

This section describes real cases in which the proposed optimization procedure was implemented. The cases are typical problems that appear in the manufacturing area and present increasing number of scenarios. Note that for design of experiments problems with multiple, and different level for each decision variable, the associated complexity is:

$$\prod_{i=1}^n l_i \quad (18)$$

where l_i is the number of levels for i th decision variable, and n is the number of decision variables.

This study evaluated three classes of simulation optimization problems:

- (1) Integer Simulation Optimization problems with the same number of levels per decision variable.

- (2) Integer Simulation Optimization problems with a different number of levels per decision variable.

- (3) Integer Simulation Optimization problems where the number of decision variables changes while the simulation model remains the same.

The simulation models were implemented using the ProModel software (ProModel, 2016). The case studies represent large-scale problems in which discrete event simulation is often used to support decision-making. The cases exemplify the use of the method for reducing the search space. It should be noted the method developed herein can be applied to any simulation optimization problem if the problem presents the same assumptions already cited.

4.1. First class of simulation optimization problems

4.1.1. Case study 1

The first case study is taken from a Brazilian telecom company that produces transponders sold to several countries. The simulation model was effectively verified and validated such that the model represents the system that both researchers and systems specialists sought to optimize. The model was programed in the ProModel software and consists of eight locations, seven entities and eight resources. The Appendix brings the probability distributions (adherence tests) for the times of the simulated processes and the conceptual model for the case study.

For this first case study, the decision variables were defined as number of workbenches (types 1 and 2) denoted by (x_1, x_2) , and the number of technical personnel (types 1, 2 and 3) denoted by (x_3, x_4, x_5) , which perform different activities that make up

Table 2
Decision variables for original search space - case study 1.

Decision variables (integers)		
x_1	Number of workbenches, Type 1	$1 \leq x_1 \leq 5$
x_2	Number of workbenches, Type 2	$1 \leq x_2 \leq 5$
x_3	Number of technical personnel, Type 1	$1 \leq x_3 \leq 5$
x_4	Number of technical personnel, Type 2	$1 \leq x_4 \leq 5$
x_5	Number of technical personnel, Type 3	$1 \leq x_5 \leq 5$

the production cell. The decision variables are integers. Table 2 presents this information.

The optimization objective was to find the best combination of variables to maximize the estimator of the total production, denoted by y . From Table 2, there are five decision variables, each one with five levels, and the total level number calculated by (18) is: $\prod_{i=1}^5 l_i = 3,125$ scenarios.

Orthogonal array L25 was chosen for the quantity of variables and the variation of the levels for each. This array covers out 25 experiments, which meets the minimum rule (Eq. 17).

Once the array was defined, an experimental matrix was generated and is presented in Table 3. The input scenarios from the experimental matrix were simulated in ProModel. The simulation run length for each input scenario is one month of simulated operations in the production cell, and the data for each estimator of the objective function were stored to calculate the super-efficiency. The number of necessary replications for each scenario was calculated considering the minimum number of data needed for the application of the parametric tests (normality test and t -test) and the calculus proposed by Banks et al. (2009, p. 445–449), and turned out to be 30.

Simulations of the 25 input scenarios and their respective replications were performed on a computer with an Intel Core 2 Duo 1.58-GigaHertz processor, 2 GigaByte of RAM, and the Windows (64-bit) operating platform. For the simulation of the 30 replications of each of the 25 scenarios, common random numbers (CRN) were used. The entire process lasted slightly less than 25 minutes.

The software General Algebraic Modeling – GAMS, version 23.6.5 (Gams, 2015), was used to calculate the efficiency related to each DMU and adapted to calculate the super-efficiency. With these results, it is possible to relate the efficiency value of each DMU to each input scenario. These values are listed in Table 3. The formulation of the DEA model for the case study is in the Appendix.

The analysis of the super-efficiency values (Table 3), ranked the most efficient DMUs, and it was shown that DMU 17 is the most efficient followed by DMU 1. Both DMUs are listed in Table 3. With the identification of the two most efficient DMUs, a new interval for each decision variable can be redefined, thus leading to a reduction of optimization search space. For instance, for the x_1 variable, the analysis of Table 3 showed that in the most efficient DMU, such variable was equal to 4, while the second most efficient DMU was equal to 1, therefore, according to the proposed method, this variable assumes a new variation range given by $1 \leq x_1 \leq 4$. Table 3 shows that variable x_5 presented the same value for its most efficient DMUs; namely, the value 1, reducing the number of variables under evaluation from 5 to 4. The new intervals for the other decision variables are shown in Table 4.

With the reduction of the variation interval for each decision variable, the search space was reduced from 3125 to 120 scenarios, a reduction of 96.16%. The commercial optimizer SimRunner (SimRunner User Guide, 2002) was used to test the efficiency of this proposal. SimRunner is a robust software package that is available with ProModel and uses Genetic Algorithms associated with Evolution Strategies (Banks et al., 2009).

To confirm the results from case study 1, SimRunner was set to the same conditions and the same objectives and with the original decision variable ranges (see Table 2). The results obtained with the optimization by reducing the search space and with the original search space of the problem are presented in Table 5.

The optimal values estimated by the optimizer for the decision variables were equal only for the variable x_3 ; namely, the value 4. For the other decision variables, SimRunner found different values, as shown by the fact that the problem with the original ranges would require researchers to hire a greater number of technical personnel of type 2 and type 3 and purchase additional workbenches of type 2.

In a way to assure the equality of results for the estimator of the objective function, the t -test (Montgomery, 2009) was applied for the difference of the 30 responses regarding each replication of the solutions found by the optimizer, considering the original search space and the reduced search space. To do so, we defined $y(i, j)$ as the optimum estimated by method i with $i=1, 2$ (reduced and original search space) in replication j with $j=1, \dots, 30$. Computed the 30 differences $d(j)=y(1, j) - y(2, j)$, the t -test was applied to these $d(j)$. The p -value of the test was equal to 0.758, evidencing no difference of results, that is, they are equivalent in terms of the estimated optimal simulation output.

Continuing the analysis, it can be observed that the optimization phase completed 69 experiments before converging while working in a reduced range equivalent to 57.5% of the total search space and consumed 2.4 hours of computational time. In comparison, the solution that considered the original range required 243 experiments, which is equivalent to only 7.7% of the feasible region, and consumed approximately 8 hours.

To validate the assumptions applied and the results produced, 3125 input scenarios (related to the total search space) were executed with 30 replications each, i.e., a total of 93,750 simulations were performed. Approximately 106 hours were required to run all the input scenarios and their replications. In this sequence, the efficiency analysis using DEA BCC model was carried out for the 3125 input scenarios (number of DMUs) and obtained 1426 efficient DMUs.

For comparison, the DEA BCC model was also applied to 120 input scenarios limited by the procedure proposed. In fact, considering the 120 input scenarios, 73 were efficient, i.e., the proposed procedure led to a new search space for the simulation optimization problem with approximately 61% of the original efficient input scenarios. In other words, the proposed procedure provided a reduction of the search space from 3125 input scenarios to 120 (a reduction of approximately 96%) without loss of solution quality.

Furthermore, it was observed that with generation of 100% of the possible input scenarios, the optimal solution (the average of the 30 replications) involved production of 1399 products within an interval with 95% confidence given by (1395; 1402). This result is statistically equivalent to that obtained by the proposed method (reduced search space): 1397 products within an interval with 95% confidence given by (1393; 1400).

4.2. Other classes of problems

The same steps, presented for case study 1, were applied to other case studies, described in the Appendix of this paper. The results found for all case studies, with respect to reduction of search space and time, are shown in Table 6.

Applying the method to the case studies led to an average reduction in search space of 97%, and an average reduction in computation time of 40%, with no loss of quality of response reported.

Table 3
Experimental matrix and results obtained for case study 1.

DMU	Decision variables					Estimator of the objective function	Super-efficiency	Ranking
	x_1	x_2	x_3	x_4	x_5			
1	1	1	1	1	1	853.77	2.000	2
2	1	2	2	2	2	1392.97	1.630	6
3	1	3	3	3	3	1391.40	1.000	19
4	1	4	4	4	4	1393.53	1.370	8
5	1	5	5	5	5	1392.97	1.000	20
6	2	1	2	3	4	1252.93	1.159	15
7	2	2	3	4	5	1394.50	1.085	16
8	2	3	4	5	1	1393.47	1.278	10
9	2	4	5	1	2	851.73	1.000	21
10	2	5	1	2	3	1136.77	1.218	13
11	3	1	3	5	2	1394.30	1.345	9
12	3	2	4	1	3	851.57	1.000	22
13	3	3	5	2	4	1396.43	1.875	4
14	3	4	1	3	5	1125.00	1.000	18
15	3	5	2	4	1	1393.43	1.529	7
16	4	1	4	2	5	1395.57	1.917	3
17	4	2	5	3	1	1395.43	2.989	1
18	4	3	1	4	2	1121.83	1.182	14
19	4	4	2	5	3	1394.90	1.687	5
20	4	5	3	1	4	847.63	1.000	23
21	5	1	5	4	3	1394.83	1.039	17
22	5	2	1	5	4	1152.07	1.244	11
23	5	3	2	1	5	852.07	1.000	24
24	5	4	3	2	1	1306.07	1.224	12
25	5	5	4	3	2	1389.67	0.790	25

Table 4
Decision variables for reduced search space – case study 1.

Decision variables (Integers)		
x_1	Number of workbenches, Type 1	$1 \leq x_1 \leq 4$
x_2	Number of workbenches, Type 2	$1 \leq x_2 \leq 2$
x_3	Number of technical personnel, Type 1	$1 \leq x_3 \leq 5$
x_4	Number of technical personnel, Type 2	$1 \leq x_4 \leq 3$

Table 5
Optimization results for case study 1.

Decision variables	Solution	
	Reduced search space (Proposed method)	Original search space
x_1	3	1
x_2	2	5
x_3	4	4
x_4	3	4
x_5	1	5

Estimator of the objective function	Responses	
	Reduced search space (Proposed method)	Original search space
\bar{y}	1397	1398
Confidence interval for \bar{y} (95%)	(1393–1400)	(1394–1401)

Computational time	Reduced search space (Proposed method)	Original search space
		(2.4+0.42) hours

5. Conclusions and recommendations for future studies

The proposed simulation optimization method was integrated with DEA BCC, adapted to the concepts of super-efficiency and orthogonal array, and used to determine the best search space. This method produced excellent results for the problems investigated.

The studied cases, which were associated with real input scenarios were modeled using simulation. With the application of this method, reductions of nearly 97% of the search space were attained without loss of response quality. We also wish to point out that, with the application of the method, average reductions of 40% in computational time were reached in the simulation optimization.

Furthermore, it should be noted that because it is not a parametric analysis tool, DEA enables users to work with data associated with non-normal probability distributions as well as small samples. In addition, we solve the problem of incommensurability; i.e. in the input and output matrix represented by the orthogonal arrays, we could admit different measurement units and different scales between the variables of decision and responses without the need of any data modification.

The use of the SimRunner optimizer allowed for comparison of the optimization results with those from the original search space and from the reduced space proposed in this paper. In all cases, SimRunner reached equivalent solutions in terms of the estimated optimal simulation output (y), thus demonstrating that the reduction in search space leads to optimal solutions for the problem. Another highlight is that in all cases, the reduction in computational time obtained with the optimization procedure was significant.

The reduction of search space allows the feasible region to be better explored by SimRunner. In the studied cases, the coverage of the feasible region after reduction of the search space was significantly greater resulting in estimated values of the objective function that often do not differ significantly to those found by the optimizer when considered all the search space.

In this manner, it can be assumed that, it is possible to represent the feasible region of a simulation optimization problem using orthogonal array. Additionally, using the super-efficiency analysis of the results of this array, it is possible to reduce the search space by ranking the most efficient DMUs using a DEA BCC model.

With respect to recommendations for future studies, the authors suggest:

- Application of the proposed procedure in multi-objectives optimization problems.

Table 6
Results for reducing search space and computational time.

	Class / Case study	Original search space	Reduced search space	Reduction (%)	Time SimRunner (Original search space) Hours	Time method (Reduced search space)	Reduction (%)
1	1	3125	120	96.16	8.00	(2.40+0.42) h	64.79
	2	15,625	480	96.93	78.50	(45.50+3.00) h	38.22
	3	100,000	1200	98.80	11.50	(7.40+0.88) h	27.97
2	4	4320	200	95.40	25.20	(9.60+8.5) h	28.17
	5	2488,320	7776	99.70	10.12	(5.18+0.62) h	42.69
	6	127,776	484	99.60	7.46	(4.70+1.10) h	22.25
3	7	107,811	15,309	85.80	5.15	(3.34+0.81) h	19.42
	8	1331	33	97.50	4.75	(0.70+0.72) h	70.11
	9	215,622	1764	99.20	4.72	(2.88+0.75) h	23.09

- Investigation of other experimental arrays combined with a DEA in a way to improve the efficiency and the effectiveness of the proposed method.
- Testing of the proposed procedure in problems involving continuous variables.
- Testing of the proposed procedure using other optimizers.
- Testing of BiO-MCDEA models to replace the DEA BCC model (Ghasemi et al., 2014).

Acknowledgments

The authors thank CAPES, CNPq (Process 445632/2014-3, 306214/2015-6, and 431758/2016-6), FAPEMIG (Process TEC-PPM-00512-15, and APQ-01188-16) and FAPESP (Process 2015/12711-4, and 2015/24560-0) for supporting this research and the anonymous referees for suggestions that contributed to substantial improvements in the paper.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ejor.2017.04.016.

References

Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). *Network Flows*. New Jersey: Prentice Hall.

Alrabghi, A., & Tiwari, A. (2015). State of the art in simulation-based optimisation for maintenance systems. *Computers & Industrial Engineering*, 82, 167–182.

Andersen, P., & Petersen, N. C. (1993). A procedure for ranking efficient units in data envelopment analysis. *Management Science*, 39, 1261–1264.

Antony, J. (2006). Taguchi or classical design of experiments: A perspective from a practitioner. *Sensor Review*, 26, 227–230.

Azadeh, A., Moghaddam, M., Asadzadeh, S. M., & Negahban, A. (2011). An integrated fuzzy simulation-fuzzy data envelopment analysis algorithm for job-shop layout optimization: the case of injection process with ambiguous data. *European Journal of Operational Research*, 214, 768–779.

Bachelet, B., & Yon, L. (2007). Model enhancement: improving theoretical optimization with simulation. *Simulation Modelling Practice and Theory*, 15, 703–715.

Bal, H., Örkücü, H., & Çelebioglu, S. (2010). Improving the discrimination power and weights dispersion in the data envelopment analysis. *Computers & Industrial Engineering*, 37, 99–107.

Ballantyne, K. N., Van Oorschot, R. A., & Mitchell, R. J. (2008). Reduce optimization time and effort: taguchi experimental design methods. *Forensic Science International: Genetics*, 1, 7–8.

Banker, R. D., Charnes, A., & Cooper, W. W. (1984). Some models for estimating technical and scale inefficiencies in data envelopment analysis. *Management Science*, 30, 1078–1092.

Banker, R. D., Charnes, A., Cooper, W. W., Swarts, J., & Thomas, D. A. (1989). An introduction to data envelopment analysis with some of its models and their uses. *Research in Governmental and Nonprofit Accounting*, 5, 125–163.

Banks, J., Carson II, J. S., Nelson, B. L., & Nicol, D. M. (2009). *Discrete-event system simulation*. Upper Saddle River: Pearson.

Besseris, G. J. (2012). Profiling effects in industrial data mining by non-parametric DOE methods: an application on screening check weighing systems in packaging operations. *European Journal of Operational Research*, 220, 147–161.

Better, M., Glover, F., & Kochenberger, G. (2015). Simulation optimization: improving decisions under uncertainty. In Cox L. A., Jr. (Ed.), *Breakthroughs in decision science and risk analysis* (pp. 59–82). New Jersey: John Wiley & Sons.

Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring the efficiency of decision making units. *European Journal of Operational Research*, 2, 429–444.

Cook, W. D., & Seiford, L. M. (2009). Data Envelopment analysis (DEA) - thirty years on. *European Journal of Operational Research*, 192, 1–17.

Cooper, W. W., Seiford, L. M., & Tone, K. (2007). *Data envelopment analysis: A comprehensive text with models, application, references and DEA-solver software*. New York: Springer Science+Business Media.

Dellino, G., Kleijnen, J. P. C., & Meloni, C. (2010). Robust optimization in simulation: taguchi and response surface methodology. *International Journal of Production Economics*, 125, 52–59.

Figueira, G., & Almada-Lobo, B. (2014). Hybrid simulation-optimization methods: a taxonomy and discussion. *Simulation Modelling Practice and Theory*, 46, 118–134.

Fu, M. C. (1994). Optimization via simulation: a review. *Annals of Operations Research*, 53, 199–247.

Fu, M. C. (2002). Optimization for simulation: theory vs. practice. *Journal on Computing*, 14, 192–215.

Fu, M. C. (2015). *Handbook of simulation optimization*. New York: Springer.

Fu, M. C., Andradóttir, S., Carson, J. S., Glover, F., Harrel, C. R., Ho, Y. C., Kelly, J. P., & Robinson, S. M. (2000). Integrating optimization and simulation: research and practice. In J. A. Joines, R. R. Barton, K. Kang, & P. A. Fishwick (Eds.), *Proceedings of the 2000 Winter Simulation Conference* (pp. 610–616).

Fu, M. C., Bayraksan, G., Henderson, S. G., Nelson, B. L., Powell, W. B., Ryzhov, I. O., & Thengvall, B. (2014). Simulation optimization: a panel on the state of the art in research and practice. In A. Tolk, S. Y. DIALLO, I. O. Ryzhov, I. Yilmaz, S. Buckley, & J. A. Miller (Eds.), *Proceedings of the 2014 Winter Simulation Conference* (pp. 3696–3706).

GAMS. The General Algebraic Modeling. (2015). <http://www.gams.com/> Accessed 11.11.15.

Ghasemi, M. R., Ignatius, J., & Emrouznejad, A. (2014). A bi-objective weighted model for improving the discrimination power in MCDEA. *European Journal of Operational Research*, 233(3), 640–650.

Harrel, C. R., Ghosh, B. K., & Bowden, R. (2004). *Simulation using promodel*. New York: McGraw-Hill.

Hillier, F. S., & Lieberman, G. J. (2010). *Introduction to operations research*. New York: McGraw-Hill.

Jahangirian, M., Eldabi, T., Naseer, A., Stergioulas, L. K., & Young, T. (2010). Simulation in manufacturing and business: a review. *European Journal of Operational Research*, 203, 1–13.

Kleijnen, J. P. C. (1998). Experimental design for sensitivity analysis, optimization, and validation of simulation models. In J. Banks (Ed.), *Handbook of simulation* (pp. 173–223). New-York: Wiley.

Kleijnen, J. P. C. (2017). Regression and Kriging metamodelling with their experimental designs in simulation: a review. *European Journal of Operational Research*, 256, 1–16.

Kleijnen, J. P. C. (2015). *Design and analysis of simulation experiments*. New York: Springer.

Kleijnen, J. P. C., Van Beers, W., & Van Nieuwenhuyse, I. (2010). Constrained optimization in simulation: a novel approach. *European Journal of Operational Research*, 202, 164–174.

Law, A. M. (2015). *Simulation modeling and analysis*. Boston: McGraw-Hill.

Law, A. M., & McComas, M. G. (2002). Simulation-based optimization. In P. A. Farington, H. B. Nembar, D. T. Sturrock, & G. W. Evans (Eds.), *Proceedings of the 1999 Winter Simulation Conference* (pp. 56–59).

Lee, L. H., Chew, E. P., Teng, S., & Chen, Y. (2008). Multi-objective simulation-based evolutionary algorithm for an aircraft spare parts allocation problem. *European Journal of Operational Research*, 189, 476–491.

Lin, R. C., Sir, M. Y., & Pasupathy, K. S. (2013). Multi-objective simulation optimization using data envelopment analysis and genetic algorithm: specific application to determining optimal resource levels in surgical services. *Omega - The International Journal of Management Science*, 41, 881–892.

Martins, M. S. R., Fuchs, S. C., Pando, L. U., Lüders, R., & Delgado, M. R. (2013). PSO with path relinking for resource allocation using simulation optimization. *Computers & Industrial Engineering*, 65, 322–330.

Miranda, R. C., Montevechi, J. A. B., Silva, A. F., & Marins, F. A. S. (2014). A new approach to reducing search space and increasing efficiency in simulation optimization problems via the Fuzzy-DEA-BCC. *Mathematical Problems in Engineering*, 2014, 1–15.

- Montevecchi, J. A. B., Miranda, R. C., & Friend, J. D. (2012). Sensitivity analysis in discrete event simulation using design of experiments. In E. W. C. Lim (Ed.), *Discrete event simulations - development and applications* (pp. 63–102). Rijeka In-Tech.
- Montgomery, D. C. (2009). *Design and analysis of experiments* (7th ed). Hoboken: Wiley.
- Nelson, B. L. (2010). Optimization via simulation over discrete decision variables. In J. J. Hasenbein (Ed.), *Tutorials in operations research: Risk and optimization in an uncertain world* (pp. 193–207). Hanover: INFORMS.
- Pasupathy, R., & Ghosh, S. (2014). Simulation optimization: a concise overview and implementation guide. In H. Topaloglu (Ed.), *Tutorials in operations research: Theory driven by influential applications* (pp. 122–150). Catonsville: Informs.
- ProModel. ProModel Corporation. (2016). <https://www.promodel.com/> Accessed 12.12.16.
- Ross, P. J. (1996). *Taguchi techniques for quality engineering* (2nd ed., p. 329). New York: McGraw-Hill.
- Roy, R. K. (2010). *A primer on the Taguchi method*. Dearborn: Society of Manufacturing Engineers.
- Shen, H., & Wan, H. (2009). Controlled sequential factorial design for simulation factor screening. *European Journal of Operational Research*, 198, 511–519.
- Siegmund, F., Bernedixen, J., Pehrsson, L., Ng, A. H. C., & Deb, K. (2012). Reference point-based evolutionary multi-objective optimization for industrial systems simulation. In C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, & A. M. Uhrmacher (Eds.), *Proceedings of the 2012 Winter Simulation Conference* (pp. 1–11).
- Simrunner User Guide. (2002). ProModel Corporation. Orem: ProModel.
- Swisher, J. R., Hyden, P. D., Jacobson, S. H., & Schruben, L. W. (2000). A survey of simulation optimization techniques and procedures. In J. A. Joines, R. R. Barton, K. Kang, & P. A. Fishwick (Eds.), *Proceedings of the 2000 Winter Simulation Conference* (pp. 119–128).
- Taguchi, G. (1987). *System of experimental design: Engineering methods to optimize quality and minimize costs*. Dearborn: UNIPUB/Kraus International Publications.
- Taguchi, G., Chowdhury, S., & Wu, Y. (2005). *Taguchi's quality engineering handbook*. New Jersey: John Wiley & Sons, Inc.
- Tamiz, M., Jones, D. F., & El-Darzi, E. (1995). A review of goal programming and its applications. *Annals of Operations Research*, 58(1), 39–53.
- Tenne, Y., & Goh, C-K. (2010). *Computational intelligence in expensive optimization problems*. Berlin: Springer.
- Willis, K. O., & Jones, D. F. (2008). Multi-objective simulation optimization through search heuristics and relational database analysis. *Decision Support Systems*, 46, 277–286.
- Xu, J., Huang, E., Chen, C-H., & Lee, L. H. (2015). Simulation optimization: A review and exploration in the new era of cloud computing and big data. *Asia-Pacific Journal of Operational Research*, 32(3), 1–34.