Qual Quant (2012) 46:665–678
DOI 10.1007/s11135-010-9420-y

RESEARCH NOTE

An improved genetic algorithm for robust design in multivariate systems

Enrique Canessa · Christian Droop · Hector Allende

Published online: 1 January 2011

Abstract In a previous article, we presented a genetic algorithm (GA), which finds solutions to problems of robust design in multivariate systems. Based on that GA, we developed a new GA that uses a new desirability function, based on the aggregation of the observed variance of the responses and the squared deviation between the mean of each response and its corresponding target value. Additionally, we also changed the crossover operator from a one-point to a uniform one. We used three different case studies to evaluate the performance of the new GA and also to compare it with the original one. The first case study involved using data from a univariate real system, and the other two employed data obtained from multivariate process simulators. In each of the case studies, the new GA delivered good solutions, which simultaneously adjusted the mean of each response to its corresponding target value. This performance was similar to the one of the original GA. Regarding variability reduction, the new GA worked much better than the original one. In all the case studies, the new GA delivered solutions that simultaneously decreased the standard deviation of each response to almost the minimum possible value. Thus, we conclude that the new GA performs better than the original one, especially regarding variance reduction, which was the main problem exhibited by the original GA.

Keywords Robust design · Taguchi methods · Genetic algorithms · Desirability functions · Research article

E. Canessa (✉) · C. Droop · H. Allende
Facultad de Ingeniería y Ciencias, Universidad Adolfo Ibáñez, Balmaceda 1620, Viña del Mar, Chile
e-mail: ecanessa@uai.cl

C. Droop
e-mail: chdroop@gmail.com

H. Allende
e-mail: hallende@uai.cl

H. Allende
Departamento de Informática, Universidad Técnica Federico Santa María, CP 110-V, Valparaíso, Chile

1 Introduction

One of the most important, yet controversial contributions of Taguchi's methods to process and quality improvement has been robust design (Allende et al. 2005; De Mast 2004; Leon et al. 1987; Kackar and Shoemaker 1986). Robust design tries to adjust controllable input variables of a system (control factors), so that system's outputs stay as close as possible to their corresponding target values and with minimum variability, even in the presence of noise factors, which cannot be controlled. To simultaneously achieve those two objectives, robust design determines dispersion factors, which mainly affect the variability of the responses, and adjustment factors, which can be used to adjust the means of the responses to their corresponding target values. This is a two-stage process, in which one first finds the dispersion factors and sets their values to minimize variability of the outputs. Then, the process determines the values of the adjustment factors, which locate the mean of the outputs as close as possible to the corresponding target values. However, as the number of outputs of the system increases, the objectives of robust design become harder to achieve (Maghsoodloo and Chang 2001). The same happens when the number of control and noise factors is large (Allende et al. 2005; Kackar and Shoemaker 1986).

To overcome the difficulties of applying robust design to multivariate systems with many control and noise factors, we proposed the use of a Genetic Algorithm (GA) (Allende et al. 2008). Figure 1 shows a simple diagram of this GA and gives an idea about its application in robust design.

Using the data collected in a typical robust design experiment, the GA searches for the combinations of levels of control factors, which tend to minimize variability of the responses of the system and achieve a good fit between the value of each response and its corresponding target value. The performance of the GA was good for getting the responses close to their target values, but the reduction in variability was only partially met. For univariate systems, the decrease in variability attained was acceptable, but for multivariate systems, it was modest (Allende et al. 2008). Thus, the current article presents some changes to the original GA to improve variability reduction.

The rest of the article is organized as follows: section two presents some details of the original GA and the changes made to it. Then, section three shows the results obtained with the new algorithm, both for univariate as well as multivariate systems. Additionally, section three compares the performance of the original GA with the new GA for univariate and multivariate systems. The article ends with a summary of the results and their implications for the use of the GA in robust design and for future developments of it.

2 Changes to the original algorithm

In this study we use our original GA, as described in Allende et al. (2008), except (a) we use a new fitness function and corresponding desirability limits, and (b) we apply a new crossover operator, which corresponds to a uniform crossover. We will present a brief description of the original GA and explain those two changes.

The original GA (Allende et al. 2008) represents the combinations of k control factors that may take s different levels (values) of a robust design experiment using an integer codification. One chromosome will be composed of a combination of different levels of each factor, which corresponds to a particular treatment of the experiment.

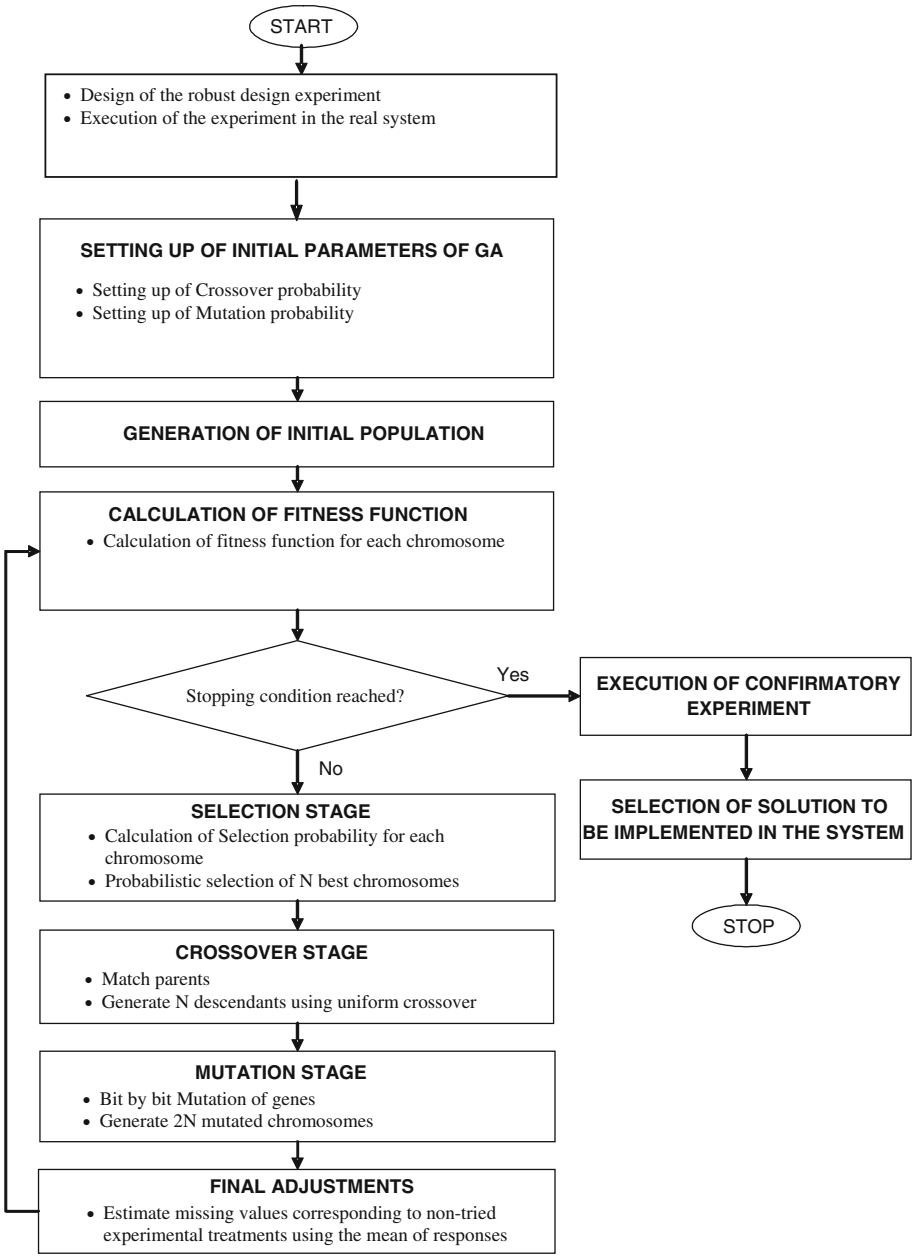

Fig. 1 Application of the genetic algorithm to robust design in multivariate systems (modified from Allende et al. 2008)

Let f_{lj} be the factor j of chromosome l, with $j = 1, 2, \ldots, k$ and $l = 1, 2, \ldots, N$. Each f_{lj} can take the value of a given level of the factor j, that is 1,2, …, s. One chromosome (or solution) is expressed as a row vector (see Eq. 1). The matrix representing the total population of solutions X will be composed of N chromosomes (see Eq. 2).

$$x_l = [f_{l1}, f_{l2}, ..., f_{lk}] \tag{1}$$

$$X = [x_1, x_2, ..., x_N]^T \tag{2}$$

To assess the fitness of each of the chromosomes, the original GA used a fitness function based on the mean square deviation (MSD) or the signal to noise ratio (SNR) for the nominal the better quality characteristic (NTB) (Taguchi 1991). However, Pignatiello (1988) concluded that for the NTB quality characteristic in robust design, a more reasonable performance statistic would be the one presented in Eq. 3, and thus we apply it in the new GA:

$$Fitness\ function \quad = \phi\,(x_l) = -\left\{\, s^2\,(x_l) + [T - \overline{y}\,(x_l)\,]^{\,2}\right\} \tag{3}$$

where: T, target value for NTB quality characteristic; $\overline{y}\,(x_l) = \frac{1}{n}\sum_{i=1} y(x_l)$, mean of the response $y(x_l)$ of chromosome x_l; $s^2\,(x_l) = \frac{1}{n}\sum_{i=1}[y_{li} - \overline{y}\,(x_l)]^2$, variance of the response $y(x_l)$ of chromosome x_l

In a multivariate system, we will have more than one response (in general, R responses) and hence, one fitness function for each of them. Following the work of Ortiz et al. (2004) and Del Castillo et al. (1996), the total fitness function for multivariate systems will be expression (4), which consists of a desirability function $D(\phi(x_l))$ and a penalty function $P_l((y(x_l))$:

$$D_l\,(x_l) = D_l\,(\phi_r\,(x_l)) - P_l\,(y\,(x_l)) \tag{4}$$

where $y\,(x_l)$ is the generic form of designating all the responses of the treatment combination x_l. Thus, the penalty function will depend on all the replications of all the responses of the system.

Each element of expression (4) can be decomposed into expressions (5) and (6) according to Ortiz et al. (2004):

$$D_l\,(\phi\,(x_l)) = (d_{1l}\,(\phi_1\,(x_l)) \times\ d_{2l}\,(\phi_2\,(x_l)) \times \ldots \times d_{Rl}\,(\phi_R\,(x_l)))^{\frac{1}{R}} \tag{5}$$

$$P_l\,(y\,(x_l)) = \left[(p_{l1}\,(y_l\,(x_l)) \times p_2\,(y_2\,(x_l)) \times \ldots \times p_R\,(y_R\,(x_l)))^{\frac{1}{R}} - c\right]^2 \tag{6}$$

where $y\,(x_l)$ is the generic form of designating all the responses of the treatment combination x_l (for an explanation of constant c, see below).

Moreover, each element of the penalty function (6) can be expressed as (Ortiz et al. 2004):

$$p_{r1}\,(y_l\,(x_l)) = \begin{cases} c + \left(\frac{L_r - \overline{y}_r(x_l)}{H_r - L_r}\right), & \overline{y}_r\,(x_l) \le L_r \\ c, & L_r \le \overline{y}_r\,(x_l) \le H_r \\ c + \left(\frac{\overline{y}_r(x_l) - H_r}{H_r - L_r}\right), & \overline{y}_r\,(x_l) \ge H_r \end{cases} \tag{7}$$

Each response y_r has a target value T_r and a lower and upper limit given by L_r and H_r respectively, in which $L_r < T_r < H_r \quad \forall r, r = 1, 2, \cdots, R$. For chromosome x_l to be feasible, the corresponding response must be within those limits ($L_r \le \overline{y}_r\,(x_l) \le H_r$). The constant c avoids the penalty function from becoming zero if infeasible cases arise and is assigned a value of 0.0001, which does not influence the value of the final solution (see Ortiz et al. 2004).

The desirability function must also have a lower and upper limit for each of the elements $d_{rl}(\phi_r(x_l))$ (see expression (5)), called the desirability limits (Ortiz et al. 2004). Expression (8) presents the desirability function, where b_r corresponds to the most desirable case and a_r to the least desirable case:

$$d_{rl}\left(\phi_r\left(x_l\right)\right) = \begin{cases} 0, & \phi_r\left(x_l\right) \leq a_r \\ \frac{\phi_r(x_l)-a_r}{b_r-a_r}, & a_r \leq \phi_r\left(x_l\right) \leq b_r \\ 1, & \phi_r\left(x_l\right) \geq b_r \end{cases} \tag{8}$$

For the new fitness function (3), parameter b_r may be expressed by Eq. 9, using the minimum variability attained in the treatment combination of the initial experiment performed:

$$b_r = -s^2_{\min}\left(x_l\right), \quad \forall r, l \tag{9}$$

The value of a_r corresponds to the least desirable case, where the variance of the response is a maximum and the mean of it is at any of the two limits (H_r or L_r). Thus, parameter a_r may be calculated using expression (10):

$$a_r = -\left(s^2_{\max}\left(x_l\right) + \left(\max\left\{\left(H_r - T_r\right), \left(T_r - L_r\right)\right\}\right)^2\right), \quad \forall r, l \tag{10}$$

As noted in Allende et al. (2008), in the calculation of the fitness value for each chromosome, we need to know the response corresponding to the experimental treatment, which each chromosome represents. However, some of those treatments might not have been part of the experiment that we conducted to gather the data. Thus, we need to estimate those responses. For estimating the variance of the response for a chromosome (treatment) not tried, we will continue using the method presented in Allende et al. (2008), which consists of calculating the variance considering all the replications of all the treatments tried in the original experiment. In the case of the mean, we will use the mean of the observations corresponding to the experiment that was carried out (Yuan 2006), since the results show that we obtained a good performance when employing it (Allende et al. 2008; Droop 2008; Bravo 2005).

As in any GA, after the algorithm computes the fitness value of the chromosomes of a generation, it must build a new generation. To do so, we need to use the genetic operators of selection, crossover and mutation (Holland 1974). Regarding selection and mutation, we kept the same operators used in the original GA, which correspond to a roulette or probabilistic selection and a bit by bit (factor by factor) mutation (see Allende et al. 2008 for details). In the case of crossover, we changed it from a one-point crossover to a uniform one, since it has been shown that this type of crossover can be more effective than others (Syswerda 1989). This crossover works as follows:

- Generate a vector C of N 0's and 1's random numbers sampled from a uniform distribution, where N equals the number of genes of the chromosome.
- When the number c_i in vector C is 0, the gene f_i from parent 1 passes to child 1 and the corresponding gene from parent 2 passes to child 2.
- When the number c_i in vector C is 1, the gene f_i from parent 1 passes to child 2 and the corresponding gene from parent 2 passes to child 1.
- Store both children and parents for use in the next generation.
- Repeat the above steps for all the selected couples of parents.

To be able to compare the performance of the original GA with the new one, we kept the stopping criterion of reaching between 3000 and 4000 chromosomes in the population. For the same reason, we continued using a crossover probability of 0.3 and a mutation probability of 0.05.

3 Application of the new algorithm

To evaluate the performance of the new GA and also compare it to the original one, we used three case studies. The first one corresponds to a real application of robust design to adjust the

automatic body paint process in a car manufacturing plant (Vandenbrande 2000). The second case study uses a multivariate process simulator with four responses, ten control factors and five noise factors. This simulator is described in Allende et al. (2008) and was used to test the original GA. Since these two case studies were used to evaluate the original GA, we also employ them to compare the original GA with the new one. Finally, we implemented a second multivariate process simulator with the same number of responses, control and noise factors, but having interactions between control and noise factors. This change increases the similarity of the simulator to a real system, since real systems generally exhibit those interactions (Gunter 1987).

3.1 Results obtained for the univariate real system

In this case, a robust design was carried out to adjust the width of the painted strip of a car painting system to a nominal width of 40.0 cm. The design of the experiment consisted of an orthogonal array $L_9(3^4)$ for the four control factors and a $L_4(2^3)$ for the three noise factors. More details and the data may be found in Vandenbrande (2000). Given the small size of the experiment, one can manually compute the best solution, which corresponds to the treatment (combination of control factors) $A = 2, B = 3, C = 1$ and $D = 2$ (chromosome $x = [2312]$). For that combination, the width of the painted strip is 41.025 cm with a standard deviation of 1.439 cm. Both the new and original GA found that solution under the same conditions and settings of the GA's (for details see Allende et al. 2008). The main variation observed between both GA's was the difference in execution time. Using a Dell Notebook 1410, Intel Centrino processor at 1.66 GHz, 1.24 GB RAM memory and running Windows XP Professional SP2 and Matlab ver. 6.5.0.180913, the original GA took a mean time of 80.1 seconds to reach the stopping condition, whereas it took only 41.47 seconds to the new GA. This is a 48% decrease in mean execution time. For this case we stopped both GA's when they reached nine iterations, so that we could precisely measure the stopping time. The decrease in execution time is due to the change in crossover operator (from a one-point crossover to a uniform one), since it has been shown that a uniform crossover tends to run faster than a uniform one (Syswerda 1989).

3.2 Results obtained for the univariate complex systems, original simulator

To test both GA's under a more complex situation, we built a simulator which is described in detail in Allende et al. (2008). The robust design for this situation considers using an inner array $L_{64}(4^{10})$ for the ten control factors and an outer array $L_{16}(4^5)$ for the five noise factors. The four responses of the simulator are optimized independently of each other, so that we deal with a univariate system.

Table 1 shows the best solutions obtained by the original and new GA. For each response (y_1, y_2, y_3 and y_4), the table lists the chromosome (treatment combination of control factors) that achieved the closest match of the response to its target value and the minimum variability (standard deviation). Since some chromosomes deliver a better match of the response to its target value than others, but achieve a smaller reduction in variability and vice-versa, the table shows more than one chromosome for some of the responses. Additionally, we should note that all of the best solutions obtained by the original GA were also found by the new GA, but are not repeated in the table as part of the chromosomes delivered by it.

We can see that the variability reduction for all of the responses achieved by the new GA is much bigger than the one obtained by the original GA. Almost all of the ratios of the

Table 1 Comparison of means and standard deviations of best solutions for the univariate complex systems, original simulator

Resp.	Chromosome	Response mean			Standard deviation		
		Target value	Obtained	Diff. (%)	Minimum possible (s_{min})	Obtained (s_{obt})	s_{obt}/s_{min} (times)
Original GA							
y_1	1-4-4-4-4-4-4-4-4-1	201.5	200.3	−0.60	2.1	5.883	2.80
y_1	1-3-3-3-3-1-1-1-1-3	201.5	187.1	−7.15	2.1	4.176	1.99
y_2	1-2-2-2-2-1-1-1-1-2	50.8	50.1	−1.38	0.8	2.167	2.71
y_2	1-2-2-2-2-2-2-2-2-1	50.8	52.4	3.15	0.8	1.712	2.14
y_3	3-3-1-2-4-4-2-1-3-2	965.0	924.2	−4.23	36.0	108.958	3.03
y_4	3-2-4-3-1-3-1-2-4-4	512.0	482.8	−5.70	28.0	98.892	3.53
y_4	4-4-1-3-2-2-3-1-4-3	512.0	593.3	15.88	28.0	91.530	3.27
New improved GA							
y_1	2-1-2-3-4-4-3-2-1-4	201.5	215.9	7.15	2.1	2.13	1.01
y_2	1-1-1-1-1-3-3-3-3-3	50.8	47.70	−6.11	0.8	0.84	1.05
y_3	4-1-4-2-3-1-4-2-3-1	965.0	975.1	1.04	36.0	36.16	1.004
y_4	2-3-4-1-2-4-3-2-1-2	512.0	426.4	−16.75	28.0	38.98	1.39

obtained to the minimum possible standard deviation (s_{obt}/s_{min}) for the solutions calculated by the new GA are 1.0, except for y_4. Even in that case, the ratio is less than half the one obtained by the best solutions delivered by the original GA. Thus, we can say that the new GA achieved the objective of improving variance reduction. Regarding the adjustment of the mean, we can see that the new GA behaves well compared to the original one. For responses y_1 and y_3, the percentage difference in mean adjustment is smaller than or equal to the one obtained by the original GA. In the case of y_2, that difference is twice the one achieved by the original GA, but given the large variance reduction that the new chromosome obtains, it is acceptable. For response y_4, the percentage difference is bigger for the new GA than for the original one, but again that is offset by the large difference in variance reduction.

3.3 Results obtained for the multivariate complex system, original simulator

In this case, we used the same simulator and the same experimental design as before, but we optimized the four responses at the same time. This means that we are optimizing a 4-dimensional multivariate system. Table 2 presents the solutions for this case.

To make Table 2 more readable, we present only the best global solution for the original GA, which tries to reduce variability and adjust the mean of the four responses at the same time. This chromosome was chosen as the best solution, since it represents a rather good compromise between both objectives for all responses (Allende et al. 2008). For the new GA, the table shows a new chromosome, which was not found by the original GA. We must note that the solution delivered by the original GA was also found by the new one. We can see that the solution delivered by the new GA is much better in variance reduction than the one of the original GA. The ratios of the obtained to the minimum possible standard deviation (s_{obt}/s_{min}) for the new GA are between 1.01 and 1.25, whereas the ones for the original GA are between 3.01 and 9.91. However, the differences in mean adjustment for responses y_2, $y_3$and y_4 are smaller for the solution of the original GA than the ones corresponding to the chromosomes of the new GA, especially for response y_4.

Table 2 Comparison of means and standard deviations of responses for the multivariate system, original simulator

Resp.	Chromosome (solution)	Response mean			Standard deviation		
		Target value	Obtained	Diff. (%)	Minimum possible (s_{min})	Obtained (s_{obt})	s_{obt}/ s_{min} (times)
Original GA							
y_1	3-4-2-1-3-3-1-2-4-2	201.5	195.0	−3.23	2.10	6.33	3.01
y_2		50.8	51.7	1.77	0.80	2.52	3.15
y_3		965.0	910.0	−5.70	36.0	356.82	9.91
y_4		512.0	508.2	−0.74	28.0	117.63	4.20
New improved GA							
y_1	3-3-1-2-4-3-1-2-4-1	201.5	200.7	−0.41	2.10	2.13	1.01
y_2		50.8	47.2	−7.20	0.80	0.81	1.01
y_3		965.0	782.2	−18.9	36.0	36.29	1.01
y_4		512.0	612.3	19.6	28.0	34.90	1.25

As we did with the solutions of the original GA, we need to select the better solution between the two shown in Table 2, since we can implement only one of them in the system. We should remember that the solution delivered by the original GA was also found by the new one, thus we must also consider that chromosome. To do so, we need to look at the difference in mean and standard deviation among the four responses and try to find a good balance between the goals of mean adjustment and variance reduction. It seems that if our primary goal is to achieve variance reduction, chromosome [3-3-1-2-4-3-1-2-4-1] might be the best choice. This solution decreases the standard deviation of responses y_1, y_2 and y_3 to almost the minimum possible one and for y_4 to a reasonable value.

Now, if we look at mean adjustment, solution [3-3-1-2-4-3-1-2-4-1] adjusts the mean of response $y_1$very well, but not for the other three outputs, whereas chromosome [3-4-2-1-3-3-1-2-4-2] achieves a better adjustment. Note also that for response y_3, neither of the two solutions obtains a rather good adjustment. It seems that chromosome [3-4-2-1-3-3-1-2-4-2] might be the best choice, since it adjusts very well the mean of responses y_1, y_2 and y_4, and rather well for y_3 compared with the other chromosome. Regarding the achievement of both goals, it is unclear which would be the better solution of the two. If our primary objective is variance reduction, chromosome [3-3-1-2-4-3-1-2-4-1] might be the best choice, whereas if we favor mean adjustment, it might be [3-4-2-1-3-3-1-2-4-2].

3.4 Results obtained for the univariate complex systems, new simulator

Since the new GA generated all of the solutions found by the original GA and also some new and better ones, at least for variance reduction, we continued analyzing the performance of only the new GA. For doing that, we built a new simulator with four responses, ten control factors, five noise factors and having interactions between some of the control and noise factors. Appendix A presents the details of the simulator and the settings used in the GA and for doing the robust design experiment. This experiment used a $L_{64}(4^{10}) \times L_{16}(4^5)$ design, which is the same design used in the optimization of the outputs of the old simulator. We decided to include interactions between control and noise factors, since this situation is more similar to real systems (Gunter 1987; Lorenzen and Anderson 1993). One generally encounters such interactions in systems, especially if one analyzes many control and noise

Table 3 Comparison of means and standard deviation of best solutions for the univariate complex systems, new simulator

Resp.	Chromosome	Response (Y) mean			Standard deviation		
		Target value	Obtained	Diff. (%)	Minimum possible (s_{min})	Obtained (s_{obt})	s_{obt}/s_{min} (times)
y_1	3-4-2-1-3-3-1-2-4-2	200	190.5	−4.77	2.1	2.1	1.0
y_1	3-3-1-2-4-3-1-2-4-1	200	197.4	−1.31	2.1	2.1	1.0
y_2	4-3-2-4-1-4-1-3-2-2	50	52.1	4.13	0.8	0.8	1.0
y_2	2-3-4-1-2-1-2-3-4-3	50	47.26	−5.49	0.8	0.8	1.0
y_3	2-2-1-4-3-3-4-1-2-4	1000	1028.8	2.88	36.0	37.1	1.03
y_3	3-3-1-2-4-4-2-1-3-2	1000	981.0	−1.90	36.0	38.1	1.06
y_4	1-4-4-4-4-1-1-1-1-4	500	521.9	4.38	28.0	28.0	1.0
y_4	3-2-4-3-1-2-4-3-1-2	500	474.9	5.02	28.0	28.0	1.0

factors. Additionally, these interactions allow finding regions of the parameter space where one can achieve good results in diminishing the impact of noise factors on the responses of the system, thus effectively reducing the variance of the outputs (Gunter 1987; Lorenzen and Anderson 1993). In this subsection we present the results for the four univariate systems, which correspond to the optimization of the four outputs of the simulator independently of each other.

Table 3 shows the best two solutions found by the GA. We can see that the GA found good solutions regarding variance reduction and mean adjustment. Selecting the appropriate control factor settings, as indicated by the corresponding chromosomes, the experimenter may achieve the lowest possible standard deviation for almost each response and obtain a mean which will be located no more than a 5.5% apart from the corresponding target values.

The only situation in which one generates a slightly bigger standard deviation than the minimum possible one is for response y_3. However, even in this case, the standard deviation is only 3–6% bigger than the minimum one.

3.5 Results obtained for the multivariate complex system, new simulator

After using the new simulator as four univariate systems, we analyzed the corresponding multivariate system, which means that we tried to optimize the four responses at the same time. Table 4 shows the best results found by the GA.

As indicated by the ratios of the obtained to the minimum possible standard deviation (s_{obt}/s_{min}), all of the obtained solutions achieved a very good variance reduction. Regarding mean adjustment, we have a less favorable situation. Some chromosomes obtained a very close match with the corresponding target value, but others achieved a mean that is located 10% or more apart from the corresponding target value. Thus, it seems that the selection of the best solution, among the ones shown, should be based mainly on mean adjustment. Even so, the decision is rather unclear. Since chromosome [2-2-1-4-3-3-4-1-2-4], achieves an adjustment of the mean that is no more than a 12.6% apart from its corresponding target value, it seems that solution might be the best one. Of course, that decision depends on the tolerance limits for each response and economic factors involved in the implementation of the settings in the real system. This means that a given setting (solution) might be less expensive to implement than others, and thus might be favored over other solutions.

Table 4 Comparison of means and standard deviation of best solutions for the multivariate complex system, new simulator

Resp.	Chromosome (solution)	Response mean			Standard deviation		
		Target value	Obtained	Diff. (%)	Minimum possible (s_{min})	Obtained (s_{obt})	s_{obt}/ s_{min} (times)
y_1	3-3-1-2-4-4-2-1-3-2	200	189.0	−5.53	2.1	2.16	1.03
y_2		50	54.0	7.93	0.8	0.81	1.01
y_3		1000	981.0	−1.90	36.0	38.1	1.06
y_4		500	570.0	14.0	28.0	28.5	1.02
y_1	2-2-1-4-3-3-4-1-2-4	200	174.9	−12.6	2.1	2.10	1.00
y_2		50	48.1	−3.73	0.8	0.80	1.00
y_3		1000	1028.8	2.88	36.0	37.1	1.03
y_4		500	553.8	10.80	28.0	28.3	1.01
y_1	1-3-3-3-3-2-2-2-2-4	200	200.3	0.17	2.1	2.10	1.00
y_2		50	43.0	−14.1	0.8	0.82	1.03
y_3		1000	791.2	−20.1	36.0	36.6	1.02
y_4		500	491.2	−1.76	28.0	28.0	1.00

4 Conclusions

The case studies we used in analyzing the performance of the new GA and its comparison with the original one show that the former achieves a better variance reduction than the latter. This is especially noticeable for multivariate complex systems. Regarding mean adjustment, both GA's obtain a good result. Here, we must remember that the best solutions for mean adjustment found by the original GA were also generated by the new GA. However, for both GA's, still one has to analyze multiple solutions delivered by the GA and then select the best one. As we did in this article and in Allende et al. (2008), one may select the best solution by trying to reach a compromise between variance reduction and mean adjustment for all the responses. This is a rather complicated and time consuming process, especially when the system has many responses. For conciseness, in this article we presented only a very shorten version of the whole analysis. Moreover, as already noted, if we also need to take into account economic factors in making our decision, that analysis may become even harder. Thus, instead of manually dealing with the analysis, one could first compute an efficient frontier with a Pareto GA (Schaffer 1984; Foneseca and Flemming 1993). This efficient frontier might involve the two objectives of robust design, variability reduction and adjustment of the mean for all of the responses. Having that frontier, the experimenter could more easily select the solution, which should lie on that frontier and should also suit his/her other needs. One can also consider economic factors in the calculation of the efficient frontier. In any case, practitioners' literature suggests that variability reduction should be the primary goal of robust design (Drickhamer 2002; Roy 2001), since one must first have the system under control before attempting to adjust the mean of the outputs and/or trying to achieve other improvements. Thus, the efficient frontier will expose the trade-off between variability reduction and mean adjustment, so that the experimenter will consciously select a solution that efficiently deals with that balance, instead of just analyzing many solutions and choosing one that might not be an efficient one, i.e. one that might not lie on the efficient frontier.

Acknowledgements We would like to thank Rick L. Riolo, Center for the Study of Complex Systems, The University of Michigan, and Claudio Moraga, European Centre for Soft Computing and FB Informatik Universität, Dortmund, for their valuable comments regarding the original version of this article.

Appendix A

The simulator was built considering 10 control factors, 5 noise factors and 4 responses. The equations which define the simulator are the following:

$$y_{Tr} = y_r + N_r \tag{11}$$

$$\begin{aligned} y_r = K_r &+ \sum_{i=1}^{10} \left[A_{ir}(X_{ir} - O_{ir}) + B_{ir}(X_{ir} - O_{ir})^2\right] \\ &+ \sum_{i=1}^{4} \sum_{j=1, i<j}^{4} C_{ijr}\,(X_{ir} - O_{ir})\left(X_{jr} - O_{jr}\right) \\ &+ \sum_{i=5}^{8} \sum_{j=5, i<j}^{8} C_{ijr}\,(X_{ir} - O_{ir})\left(X_{jr} - O_{jr}\right) \end{aligned} \tag{12}$$

$$\begin{aligned} N_r = \mu_r + \sqrt{-2\ln(U_1)}\cos(2\pi U_2)\,\sigma_r \Bigg| &\sum_{i=1}^{5} Q_{ir}(W_{ir} - Z_{ir})^2 \\ &+ \sum_{i=6,\, j=1}^{i=10,\, j=5} R_{ijr}\,(X_{ir} - O_{ir})\,(W_{ir} - Z_{ir}) + 1 \Bigg| \end{aligned} \tag{13}$$

where: y_{Tr}, total value of response $r =$ 1,2,3,4; y_r, value of r response without noise; N_r, value of noise normally distributed $\mathrm{N}(\mu_r, \sigma_r^2)$ of each response r; K_r, constant of each response r; A_{ir}, linear coefficient of control factor i of response r; X_{ir}, value of control factor i of response r; O_{ir}, optimum value of control factor i of response r, B_{ir}, quadratic coefficient of control factor i of response r; C_{ijr}, interaction coefficient of the double interaction between control factor i and control factor j of response r; μ_r, mean of the noise of response r; U_1, random number uniformly distributed between 0 and 1; U_2, random number uniformly distributed between 0 and 1; σ_r, standard deviation of the noise of response r; Q_{ir}, quadratic coefficient of noise factor i of response r; W_{ir}, value of noise factor i of response r; Z_{ir}, optimum value of noise factor i of response r; R_{ijr}, interaction coefficient of the interaction between control factor i and noise factor j of response r

The characteristics of the simulator defined by those equations are the following:

- The following double interactions exist between the control factors: 12, 13, 14, 23, 24, 34, 56, 57, 58, 67, 68, 78.
- There are no higher order interactions between control factors.
- The following double interactions exist between the control and noise factors: 61, 72, 83, 94, 10 5
- There are no interactions between noise factors.

Table 5 shows the values we set for the coefficients for obtaining the data we used in the present study.

Table 5 Values of coefficients set in the simulator

	y_1	y_2	y_3	y_4		y_1	y_2	y_3	y_4
Constant and linear coefficients					Quadratic coefficients				
K_r	200	50	1000	500	B_{1r}	0.002	−0.0003	0.008	0.005
A_{1r}	0.3	−0.08	2.0	1.5	B_{2r}	−0.05	0.001	0.09	0.002
A_{2r}	1.5	0.5	−2.0	0.8	B_{3r}	0.001	0.0002	0.008	0.005
A_{3r}	−0.15	0.08	1.0	0.3	B_{4r}	0.0005	0.00001	−0.0009	0.0008
A_{4r}	0.2	0.01	−0.5	0.1	B_{5r}	0.0024	−0.0002	−0.009	0.009
A_{5r}	0.5	−0.2	3.5	2.9	B_{6r}	0.00018	0.00008	0.0005	0.0001
A_{6r}	0.1	0.1	1.9	−0.5	B_{7r}	−0.0017	0.0005	0.0006	0.005
A_{7r}	−0.5	0.1	4.5	1.8	B_{8r}	0.0008	0.000015	0.005	0.0015
A_{8r}	0.8	−0.3	2.8	0.9	B_{9r}	−0.0003	0.00006	0.0008	0.001
A_{9r}	−0.3	0.09	−2.8	2.0	B_{10r}	0.028	0.003	−0.07	0.05
A_{10r}	2.2	−0.8	5.5	1.5					
Double interactions coefficients					Double interactions coefficients				
C_{12r}	0.0005	0.0001	0.002	−0.002	C_{56r}	0.00008	0.00002	0.0002	0.0001
C_{13r}	−0.0001	0.00005	0.00006	0.0005	C_{57r}	0.00001	0.000001	0.00008	0.00006
C_{14r}	0.0002	−0.000005	0.005	0.0005	C_{58r}	0.0009	−0.0001	0.006	−0.004
C_{23r}	0.0009	0.0002	0.002	0.0006	C_{67r}	0.0005	0.0003	0.0003	0.00001
C_{24r}	−0.00005	0.00004	0.0006	0.0003	C_{68r}	−0.002	0.0002	0.004	0.004
C_{34r}	0.0001	−0.00005	−0.0005	0.00008	C_{78r}	0.00004	0.0005	0.0008	−0.00007
Optimum value of control factor					Optimum value of control factor				
O_{1r}	75	45	12	60	O_{6r}	221	190	128	185
O_{2r}	12	12	18	8	O_{7r}	16	40	60	51
O_{3r}	18	50	95	72	O_{8r}	25	25	50	15
O_{4r}	110	75	25	65	O_{9r}	71	45	28	62
O_{5r}	50	55	82	40	O_{10r}	12	9	5	8
Noise factor coefficient					Optimum value of noise factor				
Q_{1r}	0.005	0.001	0.009	0.0008	Z_{1r}	8	12	10	9
Q_{2r}	0.0006	0.0002	0.0009	0.006	Z_{2r}	45	15	65	58
Q_{3r}	0.04	0.08	0.005	0.02	Z_{3r}	4	5	8	6
Q_{4r}	0.001	0.0004	0.008	0.0005	Z_{4r}	60	80	28	55
Q_{5r}	0.0003	0.0001	0.00005	0.0009	Z_{5r}	112	145	55	80
Double interactions coefficients between control and noise factors									
R_{61r}	−0.002	0.0005	−0.003	0.001	R_{94r}	−0.0009	0.0004	0.004	0.0015
R_{72r}	−0.008	−0.0009	0.006	0.0009	R_{105r}	0.022	0.009	−0.009	0.05
R_{83r}	0.008	0.00015	−0.005	0.0018					
Mean of the noise					Standard deviation of the noise				
μ_r	0.0	0.0	0.0	0.0	σ_r	2.1	0.8	36	28

Table 6 Target values and tolerance limits set in the algorithm

	y_1	y_2	y_3	y_4
Target value T_r	200	50	1000	500
Lower tolerance limit L_r	170	40	750	400
Upper tolerance limit H_r	230	60	1250	650

Table 7 Values for levels of control and noise factors

Level of control factor	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**
1	10	5	0	20	35	120	10	5	25	4
2	35	10	35	55	50	165	30	25	40	8
3	55	15	65	85	70	205	45	35	60	12
4	80	20	100	120	85	250	65	55	75	16
Level of noise factor	**P**	**Q**	**R**	**S**	**T**					
1	10	5	0	20	35					
2	35	10	35	55	50					
3	55	15	65	85	70					
4	80	20	100	120	85					

We must note that the response surfaces we used contain regions where all of the objectives for the responses can be simultaneously achieved. This is a necessary condition to be able to find feasible solutions in the multivariate system.

Table 6 shows the target values (T_r) and the lower (L_r) and upper (H_r) tolerance limits of the desirability function used in the algorithm, according to expression (7). We set those values around the value of the constant of each response and they might be changed to analyze the impact of them on the behavior of the algorithm.

The experimental design we selected corresponds to a product array $L_{64}(4^{10}) \times L_{16}(4^5)$, thus we used four levels for the control and noise factors. If someone decides to perform a full factorial robust design experiment using four levels, he/she must run approximately 10^9 experiments, which is impossible to do in any real productive system. Table 7 shows the values that correspond to each level.

References

Allende, H., Canessa, E., Galbiati, J.: Diseño de Experimentos Industriales. Edit. Universidad Técnica Federico Santa María, Valparaíso (2005)

Allende, H., Bravo, D., Canessa, E.: Robust design in multivariate systems using genetic algorithms. Qual. Quant. J. (2008). doi:10.1007/s11135-008-9201-z

Bravo, D.: Desarrollo de una Herramienta basada en Algoritmos Genéticos para resolver problemas de Diseño Robusto en Sistemas Multivariados. Master thesis, Universidad Adolfo Ibáñez (2005)

De Mast, J.: A methodological comparison of three strategies for quality improvement. Int. J. Qual. Reliab. Manag. **21**(2), 198–213 (2004)

Del Castillo, E., Montgomery, D.C., McCarville, D.R.: Modified desirability functions for multiple response optimization. J. Qual. Technol. **28**, 337–345 (1996)

Drickhamer, D.: BASF breaks through with statistics. Ind. Week, June (2002)

Droop, C.: Diseño Robusto en Sistemas Multivariados utilizando Algoritmos Genéticos guiados por la media y varianza de las respuestas del sistema. Master thesis, Universidad Adolfo Ibáñez (2008)
Foneseca, G., Flemming, P.J.: Genetic algorithms for multi-objective optimization: formulation, discussion, and generalization. In: Proc. 5th Intern. Conf. on Genetic Algorithms, pp. 416–423. Morgan Kaufmann, San Francisco (1993)
Gunter, B.: A perspective on the Taguchi methods. Qual. Prog. 44–52, June (1987)
Holland, J.: Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor (1974)
Kackar, R.N., Shoemaker, A.C.: Robust design: a cost effective method for improving manufacturing processes. AT&T Tech. J. **65**(2), 39–50 (1986)
Leon, R., Schoemaker, A.C., Kackar, R.N.: Performance measures independent of adjustment: an explanation and extension of Taguchi's signal to noise ratios (with discussions). Technometrics **29**(3), 253–285 (1987)
Lorenzen, T., Anderson, V.L.: Design of Experiments, a No-Name Approach. Marcel Dekker Inc., New York (1993)
Maghsoodloo, S., Chang, C.: Quadratic loss functions and SNR for a bivariate response. J. Manuf. Syst., **20**(1), 1–12 (2001)
Ortiz, F., Simpson, J., Pigniatello, J. Jr., Heredia-Langner, A.: A genetic algorithm approach to multiple-response optimization. J. Qual. Technol. **36**(4), 432–449 (2004)
Pignatiello, J.: An overview of the strategy and tactics of Taguchi. IIE Trans. **20**(3), 247–254 (1988)
Roy, R.K.: Design of Experiments Using the Taguchi Approach. Wiley, New York (2001)
Schaffer, J.D. Some experiments in machine learning using vector evaluated genetic algorithm. PhD Dissertation. Vanderbilt University, Nashville (1984)
Syswerda, G. Uniform crossover in genetic algorithms. In: J.D. Schaffer (ed.) Proc. of the 3rd Intern. Conf. on Genetic Algorithms. Van Nostrand Reinhold, New York (1989)
Taguchi, G.: Systems of experimental design. American Supplier Institute, Dearborn (1991)
Vandenbrande, W.: Make love, not war: combining DOE and Taguchi. In: ASQ's 54th Annu. Qual. Congr. Proc., pp. 450–456. (2000)
Yuan, Y.: Multiple Imputation for Missing Data: Concepts and New Development. SAS Institute Inc., Rockville (2006)