Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Solving Multi-Objective Linear Control Design Problems Using Genetic Algorithms ⋆

Gustavo Sánchez * **Minaya Villasana** ** **Miguel Strefezza** ***

* *Universidad Simón Bolívar. Departamento de Tecnología Industrial. Sartenejas. Edo. Miranda. Venezuela (gsanchez@usb.ve)*
** *Universidad Simón Bolívar. Departamento de Cómputo Científico. Sartenejas. Edo. Miranda. Venezuela (mvillasa@usb.ve)*
*** *Universidad Simón Bolívar. Departamento de Procesos y Sistemas. Sartenejas. Edo. Miranda. Venezuela (strefeza@usb.ve)*

Abstract: Two multi-objective genetic algorithms, an elitist version of MOGA and NSGA-II, were applied to solve two linear control design problems. The first was a $\mathcal{H}_2$ problem with a PI controller structure, for a first order stable plant. The second was a mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control problem. In both cases, three indicators were used to evaluate each algorithm performance: Set coverage, spread and hypervolume. It was found that NSGA-II shows better performance indicators. Moreover, for the second problem, a new controller representation was proposed with corresponding cross-over and mutation operators. This approach was able to find solutions as good as those previously published. The main advantage is that the stability restriction disappears, allowing to deal with an unconstrained optimization problem.

Keywords: Computer-aided control system design, Genetic algorithms, Multiobjective optimisation

1. INTRODUCTION

Multi-Objective Evolutionary Algorithms (MOEA) have been successfully applied to solve control problems, when a number of design objectives are conflicting (Liu et al, 2002) . One of the first attempts was presented by Fonseca (1995). This author developed an algorithm called "Multiple Objective Genetic Algorithm" (MOGA) and presented several applications to control system design.

Herreros (2000) proposed an algorithm called "Multi-Objective Robust Control Design" (MRCD) and tested it against a "Linear Matrix Inequalities" (LMI) approach for $\mathcal{H}_2/\mathcal{H}_\infty$ problems. An adaptive search space was proposed, motivated by two reasons: the selection of the initial population and the delimitation of the search space. In fact, these remain open subjects.

Herreros et al. (2001) presented an approach for adjusting the parameters of a PID controller based on the MRCD algorithm. The approach was generalized to design multivariable coupled and decentralized PID loops and was successfully validated for a large number of experimental cases.

Takahashi et al. (2004) and Molina-Cristobal et al. (2006a) also compared genetic versus LMI -based approaches to estimate the Pareto set for $\mathcal{H}_2/\mathcal{H}_\infty$ problems. The authors asserted that the genetic approach could find better results compared to the LMI approach.

Tan et al. (2005) presented a comprehensive treatment on the design of multi-objective evolutionary algorithms and their applications in domains covering areas such as control and scheduling.

Despite all this important work, many improvements and analysis are still to be accomplished. This includes issues like space search adaptation, controller representations, variation operators, complex problems tackling, etc. For example, the literature is not rich concerning comparative studies of MOEAs for control design applications: this kind of studies could be useful for the designer to select an algorithm for a given project.

In this paper, two MOEAs are compared: an elitist version of MOGA, here called EMOGA, and NSGA-II proposed by Deb (2000) . Two linear control design problems are discussed. The first is a $\mathcal{H}_2$ problem with a PI controller structure, for a first order stable plant. The second is a more complex $\mathcal{H}_2/\mathcal{H}_\infty$ problem, previously studied by Herreros (2000) and Molina-Cristobal et al. (2006a).

For the first problem, the trade-off effect between perturbation and noise rejection is analyzed. For the second problem, a new controller representation is proposed with corresponding cross-over and mutation operators. In both cases, three indicators were used to evaluate each algorithm performance: Set coverage, spread and hypervolume.

The exposition is organized as follows. In section 2, the optimization algorithms are introduced. In sections 3 and 4 the control problems are explained and numerical results are given. Finally, conclusions are drawn in section 5.

⋆ This work was supported by DID-Universidad Simón Bolívar and FONACIT-Misión Ciencia

2. MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS (MOEA)

Many multi-objective optimization algorithms using evolutionary concepts have been suggested since the pioneering work by Schaffer (1984). In order to obtain effective results, the search process needs to be guided toward the Pareto-optimal front, maintaining diversity to prevent premature convergence and to achieve a well distributed population. The two MOEA which will be compared in this work are briefly described below.

2.1 Elitist Multi-Objective Genetic Algorithm

The algorithm MOGA (Fonseca, 1995) assigns the smallest rank value for all non-dominated individuals. The dominated ones are ranked according to the number of individuals that dominate them. The fitness of each individual is computed by implementing a mapping inversely related to its rank. This value will be degraded based upon a sharing function, according to the distribution density in the feature space. The parameter α regulates the shape of the sharing function. The sharing distance σ_{share} determines the extent of the sharing region for each indivi-dual. Note that the original version of MOGA does not include any elitist mechanism.

In this work an elitist version of MOGA, here called EMOGA, was considered. The implemented elitist mechanism is very simple: after the offspring is created, children and parents are put together and the shared fitness is computed. The survivors selection is based on this last value.

2.2 Elitist Non-dominated Sorting Genetic Algorithm

NSGA-II is an improved, "state of the art", version of the original algorithm proposed by Srinivas & Deb (1994). Note that this algorithm was previously used to design PID structures by Lagunas (2004).

NSGA-II has the following features:

- It uses an elitist principle.
- It uses and explicit diversity preserving mechanism.
- There is no sharing parameter to select.
- The sorting mechanism is faster than MOGA.

The offspring is created using the parent population and usual genetic operators. Thereafter, the two populations are combined together and a non-dominated sorting mechanism is used to classify the entire population.

3. PI STRUCTURES DESIGN USING MOEA

PI is the preferred controller strategy for industrial applications. Therefore, it is important to verify whether any design problem can be solved using a PI structure like the following:

$$K_{PI}(s) = K_p \left(1 + \frac{1}{T_i s}\right) \quad (1)$$

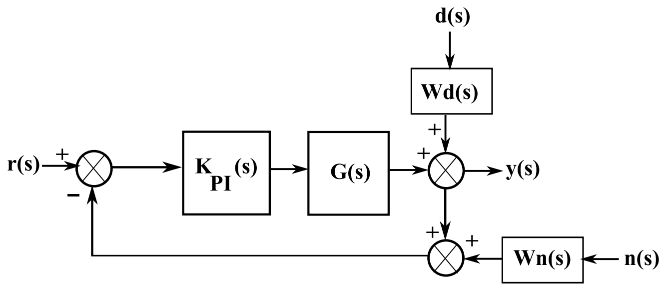

Fig.1 Control system with a PI controller

In this section, the system presented in figure 1 is considered, with:

- $r(s) \equiv$ reference signal
- $d(s) \equiv$ disturbance signal
- $n(s) \equiv$ noise signal

Let $G(s)$ be the plant transfer function, $W_n(s)$ and $W_d(s)$ arbitrary filters with:

$$G(s) = \frac{1}{s+1}, W_n(s) = \frac{1}{s+1}, W_d(s) = 1 \quad (2)$$

Let $r(s) = 0$ (regulation problem). For the closed loop we have:

$$y(s) = -T(s)W_n(s)n(s) + S(s)W_d(s)d(s) \quad (3)$$

with

$$T(s) = \frac{K_{PI}(s)G(s)}{1 + K_{PI}(s)G(s)}, \ S(s) = \frac{1}{1 + K_{PI}(s)G(s)} \quad (4)$$

and

$$S(s) + T(s) = 1 \quad (5)$$

For the condition $\|y(s)\| \approx 0$ to hold, it is necessary that $\|T(s)\| \approx 0$ and $\|S(s)\| \approx 0$ *at the same time*. However, from (5) this can not be accomplished and this is why a trade-off has to be seeked. The two-objective control problem can be stated as follows:

$$\min_{K_{PI} \in \mathcal{R}(s)^{n_u \times n_y}} \begin{pmatrix} \|W_n(s)S(s)\|_2 \\ \|W_d(s)T(s)\|_2 \end{pmatrix} \quad (6)$$

subject to the stability of the closed-loop system.

Before starting the optimization process, the search space for K_p, T_i has to be specified. In this case, to have closed-loop stability, it is sufficient to set

$$K_p > 0, T_i > 0 \quad (7)$$

Note that in this example, it is easy to generate feasible individuals, even after cross-over and mutations: there is no need for any mechanism to cope with the stability restriction. This of course is not true in the general case. The search space was set as:

$$K_p, T_i \in [0.1, 100] \quad (8)$$

3.1 EMOGA results

The EMOGA algorithm was executed using the parameters given in table 1. No special tuning procedure was carried out. The final Pareto front approximation is presented in figure 2.

Table 1. EMOGA parameters

Representation	Real numbers
Cross-Over Recombination	Arithmetic
Cross-Over Rate	0.9
Mutation Operator	Gaussian Perturbation
Mutation Rate	1/50
σ_{share}	1
α	2
Stop Condition	50 generations
Initial Population	Random
Parents Selection	SUS $(s = 2)$
Survivor Selection	$(50 + 25)$

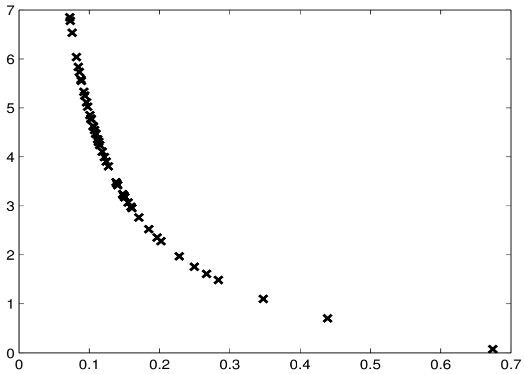

Fig 2. Final Pareto front approximation with EMOGA

3.2 NSGA-II results

The algorithm NSGA-II was executed using the parameters given in table 2. The final Pareto front approximation is presented in figure 3. To illustrate the trade-off between perturbation and noise rejection, three final controllers K1, K2 and K3 were selected (see table 3). A simulation was carried out with each controller. In figures 4, 5 and 6 simulation plots, corresponding to closed-loop responses, are presented. Note that, as expected, K3 achieves a better trade-off compared to K1 and K2.

Table 2. NSGA-II parameters

Representation	Real numbers
Cross-Over Recombination	Arithmetic
Cross-Over Rate	0.9
Mutation Operator	Gaussian Perturbation
Mutation Rate	1/50
Stop Condition	50 generations
Initial Population	Random
Parents Selection	Binary Tournament
Survivor Selection	$(50 + 25)$

Table 3. K1, K2 and K3 parameters. NSGA-II results.

	K_p	T_i	$\Vert W_n(s)S(s)\Vert_2$	$\Vert W_d(s)T(s)\Vert_2$
K1	82.15	67.73	0.08	6.37
K2	0.51	96.76	0.57	0.29
K3	9.98	100	0.21	2.13

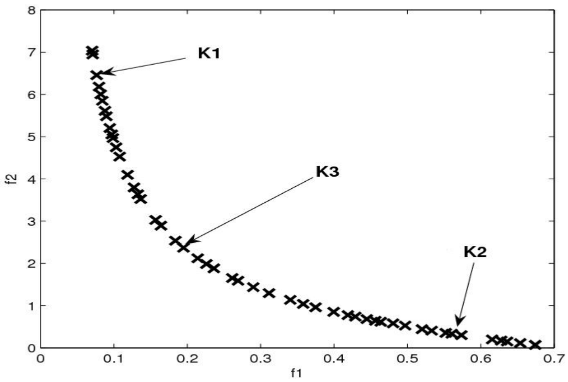

Fig 3. Final Pareto front approximation with NSGA-II

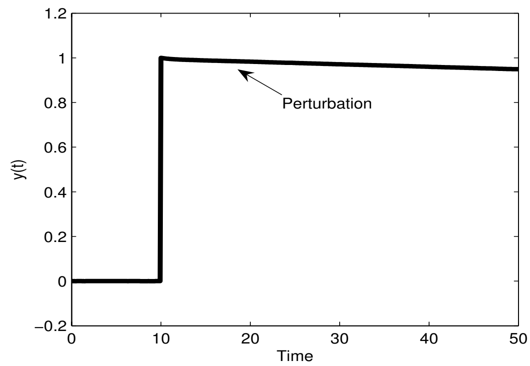

Fig 4. Perturbation and noise rejection with K1

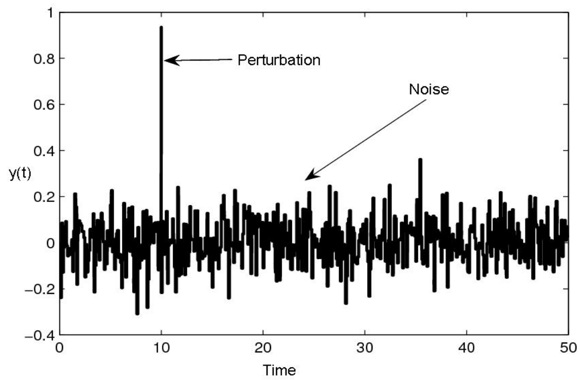

Fig 5. Perturbation and noise rejection with K2

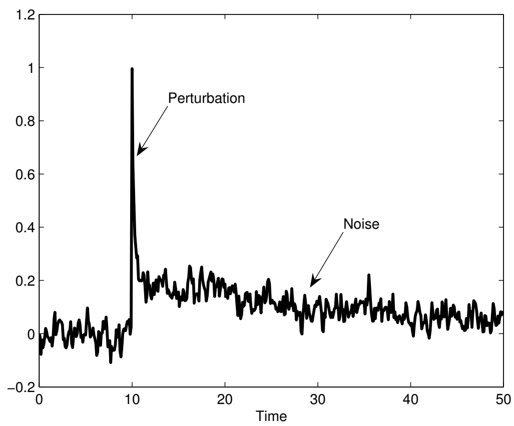

Fig 6. Perturbation and noise rejection with K3

3.3 Comparison between EMOGA and NSGA-II.

Each algorithm was executed 30 times. The mean value and the standard deviation was calculated for each indicator (see table 4):

- Set Coverage (C): it calculates the proportion of dominating solutions (always with respect to other approximation set). It is better when it is greater.
- Spread (Δ): it evaluates the diversity among the solutions. It is better when it is lower.
- Hypervolume (HV): it evaluates both closeness to the Pareto front and diversity among the solutions. It is better when it is greater. For more information about performance indicators, see the reference (Deb, 2001).

Table 4. EMOGA - NSGA-II comparison. Problem 1.

Ind.	**EMOGA**	**NSGA-II**
C	$5.3\times10^{-3}(1.4\times10^{-2})$	$1.2\times10^{-2}(2.7\times10^{-2})$
Δ	$1.1(1.6\times10^{-1})$	$6.8\times10^{-1}(6.0\times10^{-2})$
HV	$2.0(1.4)$	$2.8(0.8)$

4. MULTI-OBJECTIVE POLE PLACEMENT WITH EVOLUTIONARY ALGORITHMS

4.1 The control problem

We focus now on the problem of designing a linear controller $Kc \in \mathcal{R}(s)^{n_u\times n_y}$ for the system shown in figure 7. Matrices $A \in \mathcal{R}^{n\times n}$, $B \in \mathcal{R}^{n\times n_u}$ and $C \in \mathcal{R}^{n_y\times n}$ denote the given plant state matrices.

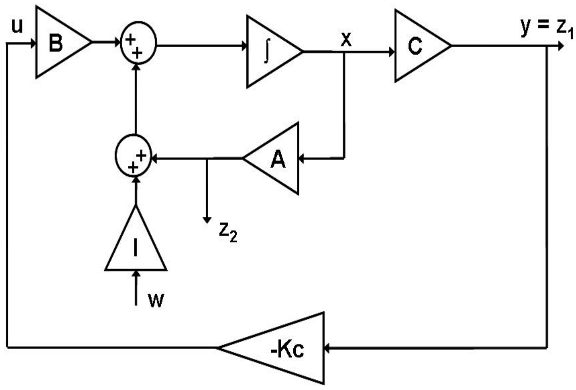

Fig 7. Second control system

As usual, $w \in L_2^{n_w\times1}$denote the exogenous input, $z_1 \in L_2^{n_{z_1}\times1}$ and $z_2 \in L_2^{n_{z_2}\times1}$ represent the outputs to be regulated, while $u \in L_2^{n_u\times1}$ and $y \in L_2^{n_y\times1}$ represent the control input and the measured output respectively. It is assumed that $G(s) = C(sI-A)^{-1}B$ is strictly proper, stabilizable from u and detectable from y. The open-loop state-space equations are:

$$\begin{cases} \dot{x} = Ax + Iw + Bu \\ z_1 = y = Cx \\ z_2 = Ax \end{cases} \tag{9}$$

The state-space equations of the controller are:

$$\begin{cases} \dot{x}_c = A_c x_c - Ly \\ u = Kx_c \end{cases} \tag{10}$$

with $K \in \mathcal{R}^{n_u\times n}, L \in \mathcal{R}^{n\times n_y}$ and

$$Kc(s) = -K(sI-A_c)^{-1}L \tag{11}$$

Let

$$T_1(K_c) = T_{z_1w}(K_c) \tag{12}$$

$$T_2(K_c) = T_{z_2w}(K_c) \tag{13}$$

be the closed-loop transfer function from w to z_1 and z_2 respectively. The mixed $\mathcal{H}_2/\mathcal{H}_\infty$ Objective Control Problem (MOCP) can now be stated as:

$$\min_{Kc\in\mathcal{K}^n\subset\mathcal{R}(s)^{n_u\times n_y}} \begin{pmatrix} \|T_1(K_c)\|_2 \\ \|T_2(K_c)\|_\infty \end{pmatrix} \tag{14}$$

where $\mathcal{K}^n \subset \mathcal{R}(s)^{n_u\times n_y}$ is the set of all stabilizing controllers of degree n.

4.2 The Pole Placement Method

Recall that an output feedback controller can be designed by combining a full information controller with a state observer. The resulting output feedback sub-system is called "observer-based controller" and has the following state-equations:

$$\begin{cases} \dot{\widehat{x}} = (A+BK+LC)\,\widehat{x} - Ly \\ u = K\widehat{x} \end{cases} \tag{15}$$

where $\widehat{x}$ is the estimated state. Thus, the system closed-loop state equations, using the estimation error $\widehat{e} = x - \widehat{x}$ as state variable are:

$$\begin{cases} \begin{pmatrix} \dot{x} \\ \dot{\widehat{e}} \end{pmatrix} = \begin{pmatrix} A+BK & -BK \\ 0 & A+LC \end{pmatrix} \begin{pmatrix} x \\ \widehat{e} \end{pmatrix} + \begin{pmatrix} I \\ I \end{pmatrix} w \\ z_1 = y = (\,C\ \ 0\,) \begin{pmatrix} x \\ \widehat{e} \end{pmatrix} \\ z_2 = (\,A\ \ 0\,) \begin{pmatrix} x \\ \widehat{e} \end{pmatrix} \end{cases} \tag{16}$$

Let $pk \in \mathfrak{C}^{n_k}$ and $pl \in \mathfrak{C}^{n_l}$ be the eigenvalues of $A+BK$ and $A+LC$ respectively. To assure closed-loop system stability, the gain matrix K and L must be calculated in such way that pk and pl belong to $\mathfrak{C}^-$ (open left-half complex plan). This representation allows poles be placed through a classical observer-based feedback controller, based on the information contained within each chromosome. Note that, unlike this representation, usually controllers are coded in terms of real-valued controller parameters. For example, in (Molina-Cristobal et al, 2006b) the decision variables are the parameters of the controller, which has the following form:

$$K(s) = bs + c\frac{s^2+a_1s+a_2}{s^2+b_1s+b_2} \tag{17}$$

The key concept of the proposed design method is using an evolutionary process in order to evolve pk and pl, moving across $\mathfrak{C}^-$ in order to find the best feasible closed-loop poles locations. In this work, the MATLAB *place* function was used to compute K and L from pk, pl. Note that a similar approach, based on Diophantine equations, was previously proposed in Zavala et al. (2002).

The MOCP problem (14) can be stated again as:

$$\min_{pk,pl\in\mathfrak{C}^-} \begin{pmatrix} \|T_1(pk,pl)\|_2 \\ \|T_2(pk,pl)\|_\infty \end{pmatrix} \tag{18}$$

Note that, in this case, the stability restriction $Kc \in \mathcal{K}^n$ has disappeared. This is the main advantage of the proposed method when compared to previous works by Herreros (2000) and Molina-Cristobal et al. (2006a).

Despite the simplicity of this method, problems have been reported when using the pole placement technique: in high-order systems, certain pole locations result in very large gains (Laub & Wette, 1984). This fact suggests caution during the optimization evolutionary process: a penalty mechanism can be used in order to avoid such locations.

4.3 Representation and Operators

In this framework, chromosomes were designed to contain the information about closed-loop poles location. They consist of complex-valued vectors containing the concatenation of *pk* and *pl* (see figure 8).

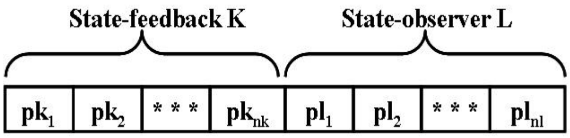

Fig 8. Chromosome structure

Two cross-over operators were implemented. The first (see figure 9) performs a "block" exchange between *pk* and *pl*, belonging to different individuals. The second (see figure 10) performs an uniform random cross-over (Eiben, 2003).

Two mutation operators were also implemented. The first (see figure 11) performs a "block" exchange between *pk* and *pl* belonging to the same individual. The second (see figure 12) slightly moves the poles in random directions.

Note that by construction of all operators, the new children are always feasible (i.e represents a stabilizing controllers).

4.4 A design example

The proposed representation and operators were applied to solve the MOCP problem (18) with EMOGA and NSGA-II and the following state matrices:

$$\left(\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right) = \left(\begin{array}{ccc|c} -21 & -120 & -100 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 150 & 0 \end{array}\right) \quad (19)$$

These algorithms were executed using the parameters given in tables 5 and 6. The final Pareto front approximations are presented in figures 13 and 14.

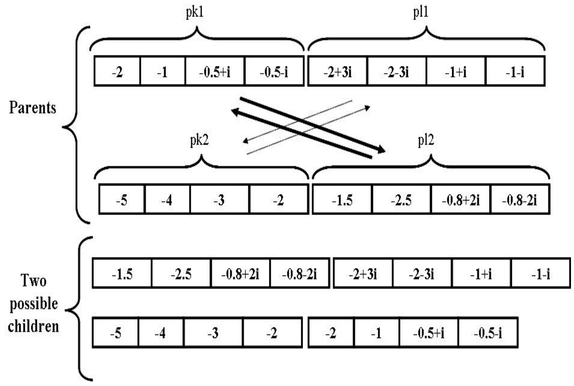

Fig 9. First cross-over operator

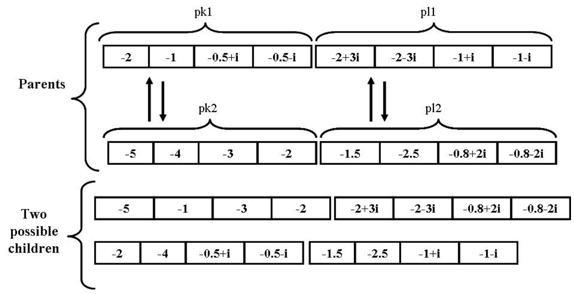

Fig 10. Second cross-over operator

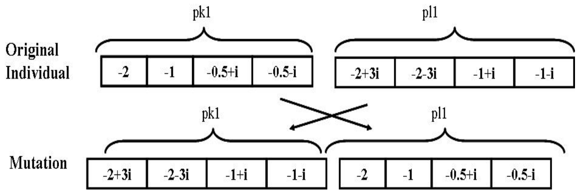

Fig 11. First mutation operator

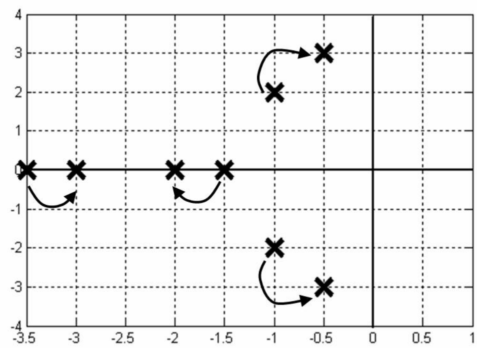

Fig 12. Second mutation operator

Table 5. EMOGA parameters. Problem 2.

Representation	Complex numbers
Cross-Over Recombination	Ops 1,2
Cross-Over Rate	0.9
Mutation Operator	Ops 1,2
Mutation Rate	1/50
σ_{share}	2.7733
α	2
Stop Condition	100 generations
Initial Population	Random
Parents Selection	SUS $(s = 2)$
Survivor Selection	$(50 + 50)$

Table 6. NSGA-II parameters. Problem 2.

Representation	Complex numbers
Cross-Over Recombination	Ops 1,2
Cross-Over Rate	0.9
Mutation Operator	Ops 1,2
Mutation Rate	1/50
Stop Condition	100 generations
Initial Population	Random
Mating Restriction	None
Parents Selection	Binary Tournament
Survivor Selection	$(50 + 50)$

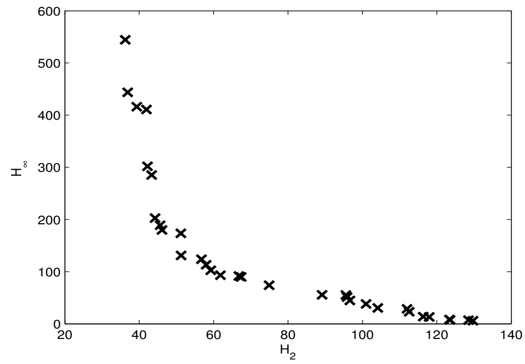

Fig 13. Final Pareto front approximation with EMOGA.

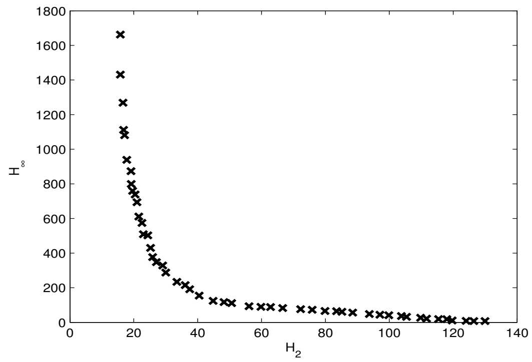

Fig 14. Final Pareto front approximation with NSGA-II

Comparison between EMOGA and NSGA-II Each algorithm was executed 30 times. Table 7 shows the results for each indicator, with mean value and standard deviation between parentheses.

Table 7. EMOGA - NSGA-II comparison. Problem 2.

Ind.	**EMOGA**	**NSGA-II**
C	$3.4 \times 10^{-2}(2.9 \times 10^{-2})$	$7.5 \times 10^{-1}(8.5 \times 10^{-2})$
Δ	$1.0(5.1 \times 10^{-2})$	$9.9 \times 10^{-1}(5.4 \times 10^{-2})$
HV	$4.5 \times 10^{5}(2.8 \times 10^{4})$	$5.4 \times 10^{5}(1.8 \times 10^{4})$

5. CONCLUSIONS AND FUTURE WORK

The goal of this paper was to present two applications of two MOEAs and to compare the performance of each algorithm. It was found that for both problems NSGA-II shows better performance indicators than EMOGA.

For the second problem, a new approach for controller coding, with cross-over and mutation operators, was proposed. This approach was able to find solutions as good as those previously published. However, its main advantage is that the stability restriction disappears, allowing to deal with an unconstrained optimization problem.

Finally, it is clear that more tests are needed. The authors are currently testing other "state-of-the art" MOEAs and other representations, in order to solve more complex control problems.

REFERENCES

K. Deb (2001). *Multiobjective optimization using Evolutionary Algorithms.* Jhon Wiley and Sons Ltd.

K. Deb, et al. (2000). 'A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II'. In *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pp. 849–858.

A. Eiben & J. Smith (2003). *Introduction to Evolutionary Computing.* Springer-Verlag.

C. Fonseca (1995). *Multiobjective Genetic Algorithms with Application to Control Engineering Problems.* Ph.D. thesis, Departement of Automatic Control and Systems Engineering. University of Sheffield.

A. Herreros (2000). 'Diseño de controladores robustos multiobjetivo por medio de algoritmos geneticos'. *Tesis Doctoral, Departamento de Ingenieria de Sistemas y Automatica.Universidad de Valladolid, Espana.* .

A. Herreros, et al. (2001). 'Design of PID-type controllers using multiobjective using genetic algorithms'. *ISA Transactions 41* pp. 457–472.

J. Lagunas (2004). *Sintonización de controladores PID mediante un algoritmo genético multi-objetivo (NSGA-II).* Ph.D. thesis, Departamento de Control Automático. Centro de Investigación y de Estudios Avanzados. México.

A. Laub & M. Wette (1984). 'Algorithms and Software for Pole Assignment and Observers'. *UCRL-15646 Rev. 1, EE Dept., Univ. of Calif., Santa Barbara, CA* .

G. Liu, et al. (2002). *Multiobjective Optimisation and Control.* Research Studies Press, Baldock, UK.

A. Molina-Cristobal, et al. (2006a). 'Linear matrix inequalities and evolutionary computation'. *International Journal of Systems Science. Vol. 37, No. 8* pp. 513–522.

A. Molina-Cristobal, et al. (2006b). 'Multi-Objective Controller Design: Evolutionary Algorithms and BMI for a passive suspension'. *13th IFAC Workshop on Control Applications of Optimization, Cachan-Paris, France* pp. 386–391.

J. Schaffer (1984). *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms.* Ph.D. thesis, Vanderbilt University, Nashville, Tennessee.

N. Srinivas & K. Deb (1994). 'Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms'. *Evolutionary Computation. Vol 2. No 3* pp. 221–248.

R. Takahashi, et al. (2004). 'Estimation of Pareto sets in the mixed H2/Hinf control problem'. *International Journal of System Science, vol. 35(1)* pp. 55–67.

K. Tan, et al. (2005). *Multiobjective Evolutionary Algorithms and Applications.* Springer-Verlag.

J. Zavala, et al. (2002). 'Multiobjective automotive drive by wire controller design'. In *Proceedings of the International Conference on Computer aided Control System Design, Glasgow, Scotland*, pp. 69–73.