An Empirical Study on Parameter Selection for Multiobjective Optimization Algorithms Using Differential Evolution

Gilberto Reynoso-Meza, Javier Sanchis, Xavier Blasco, and Miguel Martínez

Abstract—In this work, an empirical study on multiobjective optimization algorithms based on Differential Evolution (DE) algorithm is performed. The study focuses on getting good initial choices for evolutionary parameters for DE algorithm to tackle the multiobjective problem of PI controller tuning. This is an important issue in the field of control engineering, because the PI-PID controller remains a reliable digital solution for industrial processes. This study will bring an insight to the capabilities of a basic DE algorithm for multiobjective optimization, which can be the basis for more complex approaches using this evolutionary technique.

Index Terms—Multiobjective optimization, controller tuning, Differential Evolution algorithm, parameter selection.

I. INTRODUCTION

By today, PI and PID controllers remain as reliable digital control solutions, due to their simplicity and efficacy [1]. They are a common solution for research on new techniques for robust tuning in Single-Input Single-Output (SISO) and Multiple-Input Multiple-Output (MIMO) environments.

New PI-PID controller tuning techniques mainly search for a trade-off solution between several control requirements as setpoint response, load rejection and robustness. The classical conflict between performance and robustness is well-known: an outstanding performance in the nominal model could lead to an unexpected performance in the real implementation.

Some approaches state the design problem as an analytical/numerical optimization procedure [2]–[6] or as an evolutionary optimization statement [7]–[9]. In both cases, a variety of specifications with several requirements must be faced. When multiple objectives to fulfill are demanded, it is known as a multiobjective problem (MOP).

In an MOP, the designer (control engineer) has to deal with a list of requirements, and searches for a solution with a desired trade-off between objectives. A traditional approach for solving an MOP is to translate it into a single-objective problem using weighting factors, indicating the relative importance among objectives. It is not usually a trivial task to select the right weighting vector to assure a quality solution with a reasonable exchange among objectives [10]. This situation could be more complicated when constraints are considered. More elaborated methods to tackle these issues have been developed [11], such as lexicographic methods, goal programming methods, physical programming and recently global physical programming [10], [12].

Multiobjective optimization (MOO) can handle these issues in a simpler manner, due to its simultaneous optimization approach. In MOO, all the objectives and constraints are significant from the designer point of view, and as consequence, each one is optimized, to get a set of optimal non-dominated solutions.

A MOO problem, without loss of generality 1, can be stated as follows:

\[
\min_{x \in \mathbb{R}^n} J(x) = [J_1(x), \ldots, J_m(x)] \in \mathbb{R}^m
\]

where \(x \in \mathbb{R}^n\) is defined as the decision vector and \(J\) as the objective vector. In general, it does not exist a unique solution because there is not a solution better than other in all the objectives. Therefore a set of solutions, the Pareto set \(X_P\) is defined and its projection into the objective space is known as the Pareto front \(J_P\). Each point in the Pareto front is said to be a non-dominated solution.

Definition (Dominance relation): given that a solution \(x^1\) with cost function value \(J(x^1)\) dominates a second solution \(x^2\) with cost value \(J(x^2)\) if and only if:

\[
\{ \forall i \in [1, 2, \ldots, m] : J_i(x^1) \leq J_i(x^2) \} \wedge \{ \exists q \in [1, 2, \ldots, m] : J_q(x^1) < J_q(x^2) \}
\]

which is denoted as \(x^1 < x^2\).

MOO techniques search for a discrete approximation \(X_P^*\) of the Pareto set \(X_P\) that is capable of generating a good quality description \(J_P^*\) of the Pareto front. In this way, the designer has a set of solutions for a given problem and a high degree of flexibility to choose a particular or desired solution. There are several widely used algorithms for calculating this Pareto front approximation (Normal Boundary Intersection method [13], Normal constraint method [14]–[17] and the successive Pareto front optimization [18]). Recently, multi-objective evolutionary algorithms (MOEA's) have been used due to its flexibility to deal with non-convex and highly

1 A maximization problem can be converted to a minimization problem. For each one of the objectives that have to be maximized, the transformation: \(\max J_i(x) = \min(–J_i(x))\) could be applied.

Instituto Universitario de Automática e Informática Industrial, Universitat Politècnica de València, Camino de Vera s/n, Valencia 46022, Spain (e-mail: gilrym@posgrado.upv.es; http://ctl-predictivo.upv.es)
constrained functions [19], [20]. Some examples are NSGA-II [21], MOGA [22], ev-MOGA [23], pa-MyDE [24], sp-MODE [25], MOEA/D [26] among others. They have been used for SISO PID tuning [27]–[31] as well for MIMO tuning [32].

MOEA’s based on Differential Evolution (DE) algorithm [33] have shown a remarkable performance in a variety of multiojective optimization problems [34], [35]. The DE algorithm has two main operators: mutation and crossover.

A. Mutation

For each target (parent) vector \(x^t_k\), a mutant vector \(v^t_k\) is generated at generation \(k\) according to equation 2:

\[
v^t_k = x^{r_1}_k + F(x^{r_2}_k - x^{r_3}_k)
\]

(2)

Where \(r_1 \neq r_2 \neq r_3 \neq i\) and \(F\) is known as the Scaling Factor.

B. Crossover

For each target vector \(x^t_j\) and its mutant vector \(v^t_j\), a trial (child) vector \(u^t_j\) is created as follows:

\[
u^t_j = \begin{cases} v^t_{j,k} & \text{if } \text{rand}(0,1) \leq Cr \\ x^t_{j,k} & \text{otherwise} \end{cases}
\]

(3)

where \(j \in 1, 2, 3 \ldots n\) and \(Cr\) is the Crossover probability rate.

As any evolutionary algorithm, good results of DE based techniques depend on the right selection of its evolutionary parameter values. To tackle such a problem, some algorithms use some kind of adapting strategy [38]–[40]. However, a knowledge of good initial choices for evolutionary parameters in specific engineering problems can lead to a better overall performance for MOEA’s based on DE algorithm, even for algorithms with adapting strategies.

In this work, an empirical study on good initial choices for evolutionary parameters of a simple MOEA using DE algorithm is driven. The MOP is tackle is the parameter tuning for PI controllers as a preliminary step. Such a study will give a better insight on good initial choices for DE algorithms. The remain of this paper is as follows: in section II the MOP to address will be settled. In section III some examples are used for the MOP definition and the results of the performance of MOEA’s based on DE are discussed. Finally, some concluding remarks are given.

II. PROBLEM DEFINITION

The problem at hand is to tune a PI controller for a given process (see figure 1). For this work, an ISA-PI controller will be proposed:

\[
G_c(s) = k_c \left( b + \frac{1}{T_i s} \right) R(s) + k_c \left( 1 + \frac{1}{T_i s} \right) Y(s)
\]

(4)

where \(k_c\) is the proportional gain, \(T_i\) the integral time, \(b\) the set-point weighting and \(R(s), Y(s)\) the reference and control variable signal respectively.

Fig. 1: SISO PI controller structure.

Two approaches are evaluated for the parameter tuning: allowing only frequency function calculations (a common approach) and allowing time function computations (through simulation).

A. Frequency function computations (FFC)

It will be used as a guideline the non-convex optimization developed by [5] for ISA-PI/SISO controllers. This optimization procedure is analytical and model oriented and it does not require any time function computations. It is focused on getting an exchange among load disturbance rejection, robustness and setpoint response. It defines as a parameter for design a given value of the maximum sensitivity function \(M_s = \max \left| \frac{G_c(\omega)}{1 + G_c(\omega)G_p(\omega)} \right| \in [1.2, 2.0]\) and/or the maximum complementary sensitivity function \(M_p = \max \left| \frac{G_p(\omega)}{1 + G_c(\omega)G_p(\omega)} \right| \in [1.0, 1.5]\), where \(G_p(\omega)\) represents the process transfer function. Then a numerical non-convex optimization is employed, by increasing as much as possible the integral gain \(k_i = k_c/T_i\) subject to the pre-defined \(M_s\) and \(M_p\) values.

The optimization statement is

\[
J(k, k_i, \omega) = \left| C + \left( k_c - \frac{k_i}{\omega} \right) G(\omega) \right|^2
\]

(5)

subject to:

\[
J(k, k_i, \omega) \geq R^2
\]

(6)

where \(C, R\) are the center and the radius respectively on the circle which serves as envelope for a desired value of \(M_s\) and \(M_p\) in the nyquist diagram over a range of frequencies \(\omega \in [\omega_l, \omega_u]\). As pointed in [5], to perform an optimization with desired values of \(M_s\) and \(M_p\) lead to non-convex complications in the optimization procedure, and thus the envelope using \(C, R\) is used. Parameter \(b\) is selected trying
to keep $M_{sp} = \max \left[ b \cdot k_c - \frac{\omega}{2}, \frac{G_r(\omega)}{1 + G_r(\omega)G_p(\omega)} \right]$ as close to 1 as possible, in the peak resonant frequency.

For the SISO case (see figure 1), the MOP statement can be declared as: to find a trade-off solution $x = [k_c, T_i, b]$ for the design objectives $J_1(x) = -k_c/T_i$, $J_2(x) = M_s$ and $J_3(x) = M_{sp}$. That is:

$$
\min_{x \in \mathbb{R}^3} J(x) = [J_1(x), J_2(x), J_3(x)] \quad (7)
$$

However, the equation 7 does not guaranties to give the DM a useful Pareto front, with a good degree of flexibility to pick a reliable and practical solution. It is well-known that certain practical limits to $M_s$ and $M_{sp}$ values are used, to guarantee a minimum of stability margin. Due to this, the equation 7 must considerer the following constraints:

$$
\begin{align*}
1.2 \leq M_s & \leq 2.0 \quad (8) \\
1 \leq M_{sp} & \leq 1.5 \quad (9) \\
k_c + \eta \cdot k_c/T_i & \leq K_u \quad (10)
\end{align*}
$$

Where $\eta$ is the maximum value between the time delay process and 1. Constraint $k_c + \eta \cdot k_c/T_i \leq K_u$ is used to bound the maximum allowed control action effort to the ultimate gain $K_u$. Constraints $1.2 \leq M_s$ and $1 \leq M_{sp}$ are used to avoid controllers with a sluggish performance whilst constraints $M_s \leq 2.0$ and $M_{sp} \leq 1.5$ to guarantee a minimum of stability margin [5]. Due to this, it is needed to punish any unfeasible solution. Following the ideas developed in [41], the MOO statement can take the following form:

$$
\min_{\theta \in \mathbb{R}^3} \mathcal{J}(\theta) = \begin{cases} 
J(\theta) \in \mathbb{R}^3 & \text{if } \sum_{k=1}^{5} \phi_k(\theta) = 0 \\
\text{offset} + \sum_{k=1}^{5} \phi_k(\theta) & \text{otherwise}
\end{cases}
$$

where:

$$
\begin{align*}
\text{offset} & = [J_{sp}^{\max}(x), \ldots, J_{sp}^{\max}(x)], J(x) \in J^p \quad (11) \\
\phi_1(x) & = \max\{0, k_c + \eta \cdot k_c/T_i - K_u\} \quad (12) \\
\phi_2(x) & = \max\{0, 1.2 - M_s\} \quad (13) \\
\phi_3(x) & = \max\{0, 1.0 - M_{sp}\} \quad (14) \\
\phi_4(x) & = \max\{0, M_s - 2.0\} \quad (15) \\
\phi_5(x) & = \max\{0, M_{sp} - 1.5\} \quad (16)
\end{align*}
$$

Moreover, $b = 1$ if $1 \geq M_{sp}$. So, a fixing rule in the population vector is required.

### B. Time function computations (TFC)

If time function computations are possible, meaningful objectives and indexes can be used from the DM point of view. In this work, the integral of the absolute magnitude of the error for set-point response ($J_1(x) = IAE_{sp}$), for load rejection ($J_2(x) = IAE_{lr}$) and the integral of the absolute value of the derivative control signal ($J_3(x) = IADU$) are used.

Given a model, which will be controlled with a sampling time of $T_s$ with $t \in [t_0, t_f]$ and with tuning parameters $x$, the IAE and IADU are defined as:

$$
\begin{align*}
J_{1,2}(x) & = T_s \sum_{k=1}^{N} |r_k - y_k| \quad (17) \\
J_3(x) & = \sum_{k=1}^{N} |u_k - u_{k-1}| \quad (18)
\end{align*}
$$

Where $r_k$, $y_k$ and $u_k$ are respectively the set-point signal, the controlled and manipulated variables at sample $k$; $N$ is the number of samples in $[t_0, t_f]$. As it is not possible to rely only in the model used for time function computations due to robustness issues, some measure for robust stability ($J_3(x)$) is required. It is proposed the value of the maximum sensitivity function $M_s$, defined earlier. As simulations are performed, it will be required to reach an error signal $e_t$ value inside the ±2% error band for the simulation time $T_{sim}$ employed.

The MOO statement takes the following form:

$$
\min_{\theta \in \mathbb{R}^3} \mathcal{J}(\theta) = \begin{cases} 
J(\theta) \in \mathbb{R}^3 & \text{if } \sum_{k=1}^{5} \phi_k(\theta) = 0 \\
\text{offset} + \sum_{k=1}^{5} \phi_k(\theta) & \text{otherwise}
\end{cases}
$$

where:

$$
\begin{align*}
\text{offset} & = [J_{sp}^{\max}(x), \ldots, J_{sp}^{\max}(x)], J(x) \in J^p \quad (19) \\
\phi_1(x) & = \max\{0, k_c + \eta \cdot k_c/T_i - K_u\} \quad (20) \\
\phi_2(x) & = \max\{0, M_s - 2.0\} \quad (21) \\
\phi_3(x) & = \max\{0, 1.2 - M_s\} \quad (22) \\
\phi_4(x) & = \max\{0, M_{sp} - 1.5\} \quad (23)
\end{align*}
$$

These MOP definitions are general, and can be used for any kind of process defined by the control engineer. Next, a numerical study to identify good initial choices for parameter values for DE operators will be shown.

### III. Experimental Setup

Two sets of objectives will be defined for this empirical study, according with their nature: frequency function computations (FFC) and time function computations (TFC), as defined earlier. The empirical study will be performed with two well-known processes, used to validate controller tuning procedures [5]:

$$
\begin{align*}
G_1(s) & = \frac{1}{(s + 1)^3} \quad (25) \\
G_2(s) & = \frac{1}{(s + 1)^3} e^{-15s} \quad (26)
\end{align*}
$$

where $T_{sim} = [30, 300]$ secs are used respectively.
In all cases, 3e3 function evaluations (FEs) are allowed. A factorial study with parameters \( F = [0.1, 0.2, 0.3, \ldots, 1.0] \) and \( Cr = [0.1, 0.2, 0.3, \ldots, 0.9] \) is performed. This brings a total of 90 experiment combinations. Each pair of values for \( F \) and \( Cr \) will be evaluated with eleven independent trials. An initial population of 30 decision vectors is used in all cases.

As the true Pareto front is unknown, a reference pattern will be generated with a random search over the decision space. A total of 3e3 FEs will be used to generate this Pareto front for reference \( J^* \).

To evaluate the performance of each MOEA, the hypervolume indicator (Hyp) and R-indicator (R) will be used [42]. Both indicators are measured and evaluated with the executables available at www.tik.ee.ethz.ch/pisa using their standard parameters. In both cases, they are normalized between 0 and 1. The lower the indicator, the better the performance.

**A. Results**

Due to limitation space the controllers with the best tradeoff among objectives for each case will be presented. Visualization and selection in each case were performed using the Level Diagram (LD) Tool [43].

- \( G_1(s),\text{FFC} \): Five non-dominated solutions are determined. All of them with the same \( \infty \)-norm. The solution with the lowest \( 2 - \text{norm} \) is \( Cr = 0.8, F = 0.1 \).
- \( G_2(s),\text{FFC} \): Nine non-dominated solutions are found. The solution with the lowest \( \infty - \text{norm} \) is \( Cr = 0.9, F = 0.4 \).
- \( G_1(s),\text{TFC} \): Four non-dominated solutions are found. The one with the lowest \( \infty \)-norm is \( Cr = 0.8, F = 0.5 \).
- \( G_2(s),\text{TFC} \): A non-dominated solution is obtained: \( Cr = 0.8, F = 0.7 \).

The performance indicators for such parameter values combinations are shown in tables I and II.

**TABLE I: Median of solutions attained.**

<table>
<thead>
<tr>
<th>( G_1(s),\text{FFC} )</th>
<th>( G_2(s),\text{FFC} )</th>
<th>( G_1(s),\text{TFC} )</th>
<th>( G_2(s),\text{TFC} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 0.8,0.1 )</td>
<td>24</td>
<td>15</td>
<td>14</td>
</tr>
<tr>
<td>( 0.8,0.5 )</td>
<td>24</td>
<td>18</td>
<td>20</td>
</tr>
<tr>
<td>( 0.9,0.4 )</td>
<td>26</td>
<td>25</td>
<td>26</td>
</tr>
<tr>
<td>( 0.8,0.7 )</td>
<td>28</td>
<td>25</td>
<td>26</td>
</tr>
</tbody>
</table>

In all cases, the \( Cr \) value is around 0.8. It is important to remark that the common proposal for non-separable problems (as in this case) is to have a first trial with \( Cr = [0.8, 1.0] \) [44]. Therefore, in the case of \( Cr \) value we found that the rule of thumb works well. This is not the case for \( F \) value, where there are differences in the best initial choice for the value (the usual proposal is to use a \( F = [0.5, 1.0] \)). An overall good choice for control purposes seems to be the combination \( Cr = 0.8, F = 0.7 \). This decision is supported since only this value combination is capable of achieving a reasonable performance in all the experiments. Finally, at figure 2 it is shown that the LD with the performance for this solution for the \( G_1(s) \), TFC case.

**B. Validation**

For validation purposes, the parameter values \( Cr = 0.8 \), \( F = 0.7 \) will be tested against an implementation of the DEMOwSA algorithm [45], based on DE. Roughly speaking, it uses an adaptive parameter technique which includes \( F \) and \( Cr \) as variables in the decision vector, to promote the spread of good parameters along the evolution process. In both cases, a diversity improvement mechanism, it will be used a spherical pruning [25], [29], [46].

The process to be used is a linear model of a high performance drilling process [47], where the cutting force is controlled by the feed rate.

\[
G_3(s) = \frac{1958}{s^3 + 17.98s^2 + 103.3s + 190.8}e^{-0.4s}
\]  

A total of 11 runs for each algorithm (adaptive and non-adaptive) will be used with 3e3 FEs, an initial population of 60 individuals and a subpopulation of 30. As Pareto front for reference, it will be used the front generated by 25 runs of the gamultiobj algorithm provided by MatLab®. This algorithm uses a controlled elitist genetic algorithm (a variant of NSGA-II [21]). Diversity is maintained by controlling the elite members of the population as the algorithm progresses by using a crowding distance index (default parameters are used). Results are listed on table III. The results show the advantage of having a good initial choice on parameter selection. An initialization around these parameters values for an algorithm as DEMOwSA could be helpful.

**IV. Conclusions**

In this paper, an empirical study of MOEA’s based on DE algorithm examining PI parameter tuning is performed. This preliminary study supports the common thumb of rule of selecting a value for \( Cr \) around 0.8 for non-separable problems. Nevertheless, an adequate selection for scaling factor \( F \) is not an easy task, and it must be considered for each specific case. This is possible since each experiment can be easily identified: time function computations vs. frequency time computations, and time process with time delay vs. time process without time delay.

For the four scenarios evaluated, a global parameter selection was identified, which achieves a good overall tradeoff giving good results in all cases. This study will help to improve the performance of MOEA’s based on DE algorithm.

Future work will focus in a larger set of industrial processes, as well as more MOP definition, to have a bigger basis to identify good parameter choices. Also, this study will settle the basis for an MOP benchmark definition for control purposes. This will allow comparisons among MOEAs in the control engineering field.

**ACKNOWLEDGMENT**

This work was partially supported by the FPI-2010/19 grant from the Universitat Politècnica de València and the project DPI2008-02133/DPI from the Spanish Ministry of Science and Innovation.
Fig. 2: Performance of a basic MOEA using DE algorithm for evolutionary parameter combination $C_r = 0.8$, $F = 0.7$ in $G_1(s)$, TFC experiment. The Pareto front approximation calculated with 11 runs using DE (dots) is compared with the Pareto front for reference $J_p^{ref}$ (•).
Table II: Performance on $$J^{exp}_P$$.

<table>
<thead>
<tr>
<th>Test</th>
<th>Mean value</th>
<th>Median value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cr</td>
<td>F</td>
</tr>
<tr>
<td>$G_1(s)$, FFC (98 solutions)</td>
<td>0.8</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.7</td>
</tr>
<tr>
<td>$G_2(s)$, FFC (40 solutions)</td>
<td>0.8</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.7</td>
</tr>
<tr>
<td>$G_1(s)$, TFC (93 solutions)</td>
<td>0.8</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.7</td>
</tr>
<tr>
<td>$G_2(s)$, TFC (110 solutions)</td>
<td>0.8</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.7</td>
</tr>
</tbody>
</table>

Table III: Performance on $$J^{test}_P$$.

<table>
<thead>
<tr>
<th>Test</th>
<th>Mean value</th>
<th>Median value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cr</td>
<td>F</td>
</tr>
<tr>
<td>$G_1(s)$, FFC (130 solutions)</td>
<td>SA</td>
<td>SA</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.7</td>
</tr>
<tr>
<td>$G_2(s)$, TFC (129 solutions)</td>
<td>SA</td>
<td>SA</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.7</td>
</tr>
</tbody>
</table>

References


