Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 49-7 (2016) 821–826

Optimization Alternatives for Robust Model-based Design of Synthetic Biological Circuits ⋆

Y. Boada * **J.L. Pitarch** ** **A. Vignoni** *** **G. Reynoso-Meza** **** **J. Picó** *

* *I.U. de Automática e Informática Industrial (ai2), Universitat Politècnica de Valencia, 46022, Camino de Vera S/N, Valencia, Spain. (e-mail: {yaboa,jpico}@upv.es)*

** *Systems Engineering and Control Department, Universidad de Valladolid, 47011, C/ Real de Burgos S/N, Valladolid, Spain. (e-mail: jose.pitarch@autom.uva.es)*

*** *Center for Systems Biology Dresden and the Max Planck Institute of Molecular Cell Biology and Genetics, Pfotenhauer str. 108, 01307 Dresden, Germany. (e-mail: vignoni@mpi-cbg.de)*

**** *Industrial and Systems Engineering Graduate Program (PPGEPS), Pontifícia Universidade Católica do Paraná, Imaculada Conceião, 1155, 80215-901 Curitiba, PR, Brazil. (e-mail: g.reynosomeza@pucpr.br)*

Abstract: Synthetic biology is reaching the situation where tuning devices *by hand* is no longer possible due to the complexity of the biological circuits being designed. Thus, mathematical models need to be used in order, not only to predict the behavior of the designed synthetic devices; but to help on the selection of the biological parts, i.e., guidelines for the experimental implementation. However, since uncertainties are inherent to biology, the desired dynamics for the circuit usually requires a trade-off among several goals. Hence, a multi-objective optimization design (MOOD) naturally arises to get a suitable parametrization (or range) of the required kinetic parameters to build a biological device with some desired properties. Biologists have classically addressed this problem by evaluating a set of random Monte Carlo simulations with parameters between an operation range. In this paper, we propose solving the MOOD by means of dynamic programming using both a *global* multi-objective evolutionary algorithm (MOEA) and a *local* gradient-based nonlinear programming (NLP) solver. The performance of both alternatives is then checked in the design of a well-known biological circuit: a genetic incoherent feed-forward loop showing adaptive behavior.

Keywords: Biological circuits, Kinetic parameters, Multiobjective optimisations, Nonlinear programming, Computer-aided design

1. INTRODUCTION

Synthetic biology shares some features with civil and mechanical engineering, for instance, in the use of optimization in modelling the whole system-level stresses and traffic flows (Church et al., 2014). Moreover, the lack of well-characterized parts and methods for reliably and robustly composing parts into devices contributes to the current disparity of results between designing systems and synthesising them (Way et al., 2014). Therefore, a precise characterization and predictable part compositions are essential for the efficient creation of sophisticated genetic circuits (Church et al., 2014). However, the development of methodologies to design devices which are functional and whose solution allows biological components to be systematically, reliably, and predictably assembled into a device or circuit (Way et al., 2014), is a current challenge.

The systematic design of complex biocircuits from libraries of standard parts relies on mathematical models describing the circuit dynamics. In this framework, modular modelling tools facilitate the mathematical representation of biological parts and their combinations, providing the description of the reactions taking place inside the different parts and the interfaces to connect them (Medema et al., 2012). In this way, the modular and systematic design of biocircuits allowing to optimally perform a pre-defined function, can be formulated using an optimization framework (Feng et al., 2004).

Advanced optimization-based methods capable of handling high levels of complexity and multiple design criteria are being proposed (Otero-Muras and Banga, 2014). These new approaches combine the efficiency of global mixed-integer nonlinear programming MINLP solvers with multiobjective optimization techniques (Banga, 2008). Along with this, there is an ever-growing appreciation for biological complexity, which requires new circuit design principles and programming paradigms to overcome barriers such as metabolic load, crosstalk, resource sharing, and gene expression noise (Vignoni et al., 2013; Boada et al., 2015). Nevertheless, even the specification of the circuit desired dynamics is most often naturally

⋆ The research leading to these results has received funding from the European Union (FP7/2007-2013 under grant agreement n°604068), the Spanish Government (FEDER-CICYT DPI2011-524 28112-C04-01, DPI2014-55276-C5-1-R, DPI2015-70975-P) and the National Council of Scientific and Technologic Development of Brazil (BJT-304804/2014-2). Yadira Boada thanks also grant FPI/2013-3242 of the Universitat Politècnica de Valencia.

10.1016/j.ifacol.2016.07.291

expressed as a multi-objective problem (MOP), this approach has been seldom used.

For the case of feedforward incoherent circuits presenting adaptive behaviour, Boada et al. (2016) studied the application of a multi-objective optimization design (MOOD) to get model based set guidelines for the selection of its biological parts. Instead, current approaches define independent thresholds a priori for each of the functional goals (accuracy and sensitivity) in adaptive circuits. Then, Monte Carlo-like approaches are used, sampling the parameters space and simulating the circuit time response. The result of these simulations is used to asses the circuit behavior, so as to profile the subset of the parameters space that result in a circuit behavior fulfilling all thresholds (Chiang and Hwang, 2013). However, this approach has a huge computational cost and does not ensure that a satisfactory set of optimal solutions will be found (Chiang et al., 2014), despite of requiring a large set of samples. This problem increases as the thresholds defining the acceptable circuit behavior are tighter.

To build a given functional device with desired dynamic behavior, in Boada et al. (2016) the design criteria are encoded in the formulation of the objectives and optimization setup itself. As a result, the designer obtains regions/intervals of parameters along the Pareto optimal front giving rise to the predefined circuit behavior. Contrarily to the passive search for solutions of Monte Carlo-based approaches, the multi-objective optimization approach point searches to find all possible optimal solutions as a first step. The MOOD framework also naturally provides a classification of the parameters attending to their effect on each of the goals along the Pareto front. In this work, to effectively solve the MOP, we compare different optimization algorithms and took advantage of the recent advances in differential evolution based MOEA tools (Reynoso-Meza et al., 2010) as well as the use of large-scale NLP solvers (Wächter and Biegler, 2006) embedded in simulation and automatic-differentiation frameworks (Andersson et al., 2012).

The rest of the paper is organized as follows: Section 2 describes the design methodology and the different alternatives to approach the optimization problem, Section 3 describes a type-1 feed-forward loop (I1-FFL) circuit and presents a reduced model which will be used later on in the formulation of the multi-objective optimization problem, Section 4 deals with the formulation of the circuit design as a multi-objective problem and presents the used software tools to solve it, the achieved results are shown in Section 5 where the main findings for the different algorithms are presented and, finally, some conclusions and future works are drawn in the last section.

2. METHODOLOGY

The complete methodology, the multi-objective optimization framework to obtain model-based guidelines for tuning biological synthetic devices, is proposed in Boada et al. (2016). Here we assume a starting point when we already have a desired dynamical behavior, a topology for the circuit that can present the mentioned behavior, and a model of the circuit of functional module; and we focus on the different alternatives to solve the multi-objective optimization problem. We refer to (Boada et al., 2016) for the complete methodology.

With the model of the of the functional module is at hand, is possible to formulate the multi-objective optimization design. Typically the objectives will be in conflict, thus requiring to reach a trade-off among the solutions. The ways to approach this are usually:

(1) *Ad-hoc* weighting of the different objectives to transform the problem into a single-objective one (Mattson and Messac, 2005) and use a deterministic optimization algorithm.
(2) Set thresholds on each of the objectives and run multiple single-objective optimization problems (SOOP) through deterministic algorithms to obtain a Pareto front (PF) (Miettinen, 1999): this is building a grid on the objective upper bounds and set it as constraints for the MOOP while minimizing/maximizing just a single one.
(3) Address the problem as a complete multi-objective optimization via evolutionary algorithms.

In this work we did not considered the first option because it does not provide a systematic way to explore all the possible optimal designs, i.e., the obtained solution highly depends on the selected weights and it can hide other optimal solutions (in the Pareto sense) which may better represent the designer preferences. In contrast, given the stochastic nature of multi-objective differential evolution (MODE) algorithms, they are able to search for all the possible solutions in the parameters space along the Pareto front. However, convergence cannot be guaranteed and the tuning the algorithm parameters setting to obtain good performances may be a non-trivial task too.

On the other hand, deterministic algorithms are very robust and can guarantee local convergence. However, they are very sensitive to the initial guess, required to run the optimization, so they may be stuck in a local optimum. Recent developments in gradient-based nonlinear programming, which implement automatic differentiation algorithms, provide a good alternative to compute an approximation of the Pareto front by means of the above listed second alternative. The main advantage of such tools is feeding the NLP solver (SQP-type (Gill et al., 2005) or interior-point ones (Wächter and Biegler, 2006)) with the exact Jacobians and Hessians of the objective function and constraints. This provides a fast and accurate convergence, contrarily to what happens for instance with finite-differences approximations of these derivatives.

3. INCOHERENT TYPE 1 FEED-FORWARD LOOP (I1-FFL)

The desired dynamical behavior for the functional module to be designed is expressed in terms of the desired input-output relationship. Although currently there are no *catalogues* as such for functional modules, there is a vast literature in the systems biology area on network motifs producing a variety of dynamic behaviors (Alon, 2006). Also, many of the functional circuits that are being implemented in synthetic biology draw from the historical work in areas such as electronics and control (bistables, feedback and feedforward structures, switches, etc). See for example Ma et al. (2009) and references therein.

Adaptation is an important property of biological systems, linked to homeostasis (Alon, 2006), and to generation of responses that depend on the fold-change in the input signal, and not on its absolute level (Goentoro et al., 2009). It is defined as the particular ability of biological circuits to respond to a change in its input and return to the value it had prior to the stimulus, even when the input change persists, as depicted in Figure 1A. The case-study genetic circuit here shows adaptation because of its relevance.

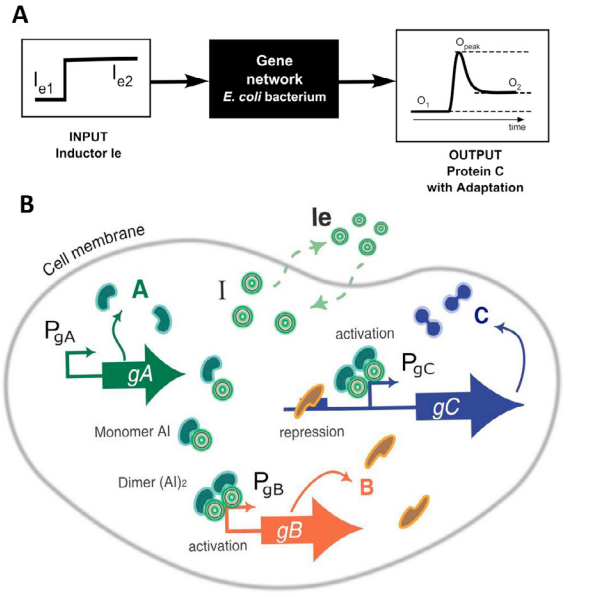

Fig. 1. A. Input-output desired behavior for the functional device. B. Representation of a cell incorporating a three-node incoherent feedforward loop synthetic circuit.

Different three-node topologies are possible (Ma et al., 2009) giving rise to adaptive behavior. Among them, the incoherent type 1 feed-forward loop (I1-FFL) structure is of the most common network motifs. Different implementations are possible, including enzyme reaction networks (Ma et al., 2009; Chiang et al., 2014), and gene networks (Basu et al., 2004) and *in vitro* transcriptional networks (Kim et al., 2014). In the gene network case, a protein A acts as a transcription factor and activates expression of two downstream genes B and C. In turn protein B represses expression of gene C. Figure 1B depicts the gene network topology. An inducer molecule I acts as input to the circuit. The protein A product of gene A bounds to the inducer I, forming a monomer $A \cdot I$. This one dimerizes. The dimer $(A \cdot I)_2$ is the transcription factor that activates expression of gene C directly, and represses it indirectly via activation of the repressor B. As a result, when a signal causes node A to assume its active conformation, C is produced, but after some time B accumulates, eventually attaining the repression threshold for the gene C promoter.

We use the following reduced model proposed in Boada et al. (2016) with nine states. The species in the reduced model are mA, A, I, $(AI)_2$, mB, B, mC, C, and I_e respectively (see Table 1). The resulting reduced dynamical model is:

$$
\begin{aligned}
\dot{x}_1 &= k_{mA} C_{gA} - d_{mA} x_1 \\
\dot{x}_2 &= k_{pA} x_1 - d_A x_2 - k_2 x_2 x_3 + k_{-2} M \\
\dot{x}_3 &= -k_2 x_2 x_3 + k_{-2} M + k_d x_9 - k_{-d} x_3 - d_I x_3 \\
\dot{x}_4 &= k_3 M^2 - k_{-3} x_4 - d_{AI2} x_4 \\
\dot{x}_5 &= K_{mB}\, C_{gB} \frac{x_4}{\gamma_1 + x_4} - d_{mB} x_5 \\
\dot{x}_6 &= k_{pB} x_5 - d_B x_6 \\
\dot{x}_7 &= K_{mC} C_{gC} \frac{x_4 + \beta_1 \gamma_4 x_6 + \beta_2 \gamma_5 x_4 x_6}{\gamma_2 + \gamma_3 x_4 + \gamma_4 x_6 + \gamma_5 x_4 x_6} - d_{mC} x_7 \\
\dot{x}_8 &= k_{pC} x_7 - d_C x_8 \\
\dot{x}_9 &= K_{cells}(k_{-d} x_3 - k_d x_9) - d_{Ie} x_9
\end{aligned}
\tag{1}
$$

with

$$M = \frac{\sqrt{(d_{AI} + k_{-2})^2 + 8k_3(k_2 x_2 x_3 + 2k_{-3} x_4)} - d_{AI} - k_{-2}}{4k_3}$$

being the monomer algebraic relation and

$$K_{cells} = \frac{V_{cell} \cdot N_{cells}}{V_{med}}$$

the volumes relationship in order to correct concentrations outside the cells. In later simulations we set $V_{cell} = 10^{-15}$L (volume of an E.coli cell), $N_{cells} = 2.4 \cdot 10^8$cells/mL $\cdot 0.18$mL (the number of cells in a 180μL culture with $OD = 0.3$) and $V_{med} = 180\mu$L (the culture medium used in a well of a plate reader).

The model has 26 parameters, described in the Table 2.

Table 1. List of variables used in the reduced model

Variable	Description	Units	Symbol
x_1	$mRNA_{gA}$	nM	mA
x_2	A protein	nM	A
x_3	Inducer	nM	I
M	A·I monomer	nM	A·I
x_4	$(A{\cdot}I)_2$ dimer	nM	$(A{\cdot}I)_2$
x_5	$mRNA_{gB}$	nM	mB
x_6	B protein	nM	B
x_7	$mRNA_{gC}$	nM	mC
x_8	C protein	nM	C
x_9	Extracellular inducer	nM	I_e

Table 2. Parameters for the reduced model

Parameter	Description	Unit
$k_{m_A}, k_{m_B}, k_{m_C}$	gA, gB, gC transcription rate	min^{-1}
$k_{p_A}, k_{p_B}, k_{p_C}$	m_A, m_B, m_C translation rate	min^{-1}
k_d, k_{-d}	inducer diffusion rate	min^{-1}
d_{AI}	(AI) degradation rate	min^{-1}
d_{AI2}	$(AI)_2$ degradation rate	min^{-1}
k_2, k_3	(AI) y $(AI)_2$ association rate	min^{-1}
k_{-2}, k_{-3}	(AI) y $(AI)_2$ dissociation rate	min^{-1}
$C_{g_A}, C_{g_B}, C_{g_C}$	gA, gB, gC copy number	mM
γ_1	gB promoter Hill constant	mM
$\gamma_2, \gamma_3, \gamma_4, \gamma_5$	gC promoter coefficients	nM,ad,ad,nM^{-1}
$d_{m_A}, d_{m_B}, d_{m_C}$	m_A, m_B, m_C degradation rate	min^{-1}
d_A, d_B, d_C	A, B, C degradation rate	min^{-1}
d_I, d_{Ie}	inducer degradation rate	min^{-1}
d_{AI}, d_{AI2}	(AI), $(AI)_2$ degradation rate	min^{-1}

4. MULTI-OBJECTIVE OPTIMIZATION APPROACH

Now a multi-objective optimization problem will be stated to search for solutions which fulfill the desired behavior.

4.1 Circuit specifications: MOOP definition.

At this point, the circuit specifications must be formulated as design objectives to be optimized. Recalling the desired input-output behavior for the I1-FFL circuit, depicted in Figure **??**, denote by θ the set of selected parameters of the reduced model (1) for optimization (decision variables):

$$\theta := [k_{mB} C_{gB}, k_{mC} C_{gC}, k_{pB}, k_{pC}, \gamma_1, \gamma_3, \gamma_4, \gamma_5, d_B, d_C]$$

The rest are fixed to the values shown in Table 3.

Two basic objectives are then considered for this circuit:

- **Sensitivity:** after input stimulation a clear transient peak value is desired for the output. Sensitivity can be defined in relative terms as the relationship between the input

Table 3. Fixed parameters.

Parameter	Value	Parameter	Value
k_{pA}	80	k_d	0.06
d_{mA}	0.3624	k_{-d}	0.06
d_{mB}	0.3624	d_I	0.0164
d_{mC}	0.3624	d_{Ie}	0.000282
d_A	0.035	d_{AI}	0.035
k_{-2}	20	d_{AI2}	0.035
k_{-3}	1	γ_2	0.02
k_2	0.1	β_1	0.05
k_3	0.1	β_2	0.05

variation, and the output one during the transient. In our case, we define sensitivity as the ratio between the absolute total variation of the output signal –the C protein concentration x_8–, and the variation of the input signal – the external inducer x_9.

- **Precision:** after the peak transient, the output must go back to its value previous to circuit stimulation. Thus, precision can be defined as the inverse of the normalized output error. The lower the steady state error, the higher the precision.

Our design objectives can be mathematically expressed by means of the indexes:

$$\begin{aligned} J_1(\theta) &= \frac{2(x_9(t_f) - x_9(t_0))}{\int_{t_0}^{t_f} |\frac{dx_8}{dt}| dt} \\ J_2(\theta) &= \frac{x_8(t_f) - x_8(t_0)}{x_9(t_f) - x_9(t_0)} \end{aligned} \tag{2}$$

where t_f is the time length of the experiment. The input stimulus is applied at t_0.

Sensitivity is the inverse of $J_1(\theta)$. Notice the total absolute variation of the C protein concentration is obtained as half the accumulated absolute value of the time derivative of x_8. The lower $J_1(\theta)$ (larger output peak w.r.t. input variation), the higher the sensitivity.

Precision is the inverse of $J_2(\theta)$, i.e. the inverse of the ratio between the variation of the C protein concentration between t_0 and t_f, and that of the the external inducer. If the C protein concentration x_8 at t_f is the same as the initial one at t_0, precision is infinite.

Additionally, other objectives could be considered. For instance, to obtain realistic solutions in our case regarding the values of protein B concentration, its absolute total variation is taken into account as a constraint. This can be expressed as

$$P(\theta) = \int_{t_0}^{t_f} |\frac{dx_6}{dt}| dt$$

together with the constraint $0.1 < P(\theta) < 5000$.

Another relevant issue is the definition of limits for $J_1(\theta)$ and $J_2(\theta)$ beyond which we consider that precision and sensitivity degrade too much (Ma et al., 2009). This is the so-called *pertinency* range of the objectives. The limits established for the pertinency box in this work are: $J_1(\theta) \in [1e^{-3} \quad 100]$ and $J_2(\theta) \in [1e^{-4} \quad 20]$.

Finally, we look for the set of values for the 10 decision variables θ that optimize both objectives. Our MOOP can thus be formulated as follows:

$$\begin{aligned} \min_{\theta \in \mathbb{R}^{10}} J(\theta) = \quad & [J_1(\theta), J_2(\theta)] \in \mathbb{R}^2 \\ \text{subject to:} \quad & \text{System dynamics} \quad (1) \\ & 1 < P(\theta) < 10000 \end{aligned} \tag{3}$$

4.2 Optimization process

The multi-objective optimization process seeks for the best parameters θ_P^* that give the best Pareto-front approximation J_P^*. Solutions with a strong degradation in some objectives (out of the pertinency box) are intentionally disregarded in the search analysis. At this point, several options are available for practitioners to perform the optimization process. Here we will use and compare:

- A MOEA based on the differential evolution (DE) algorithm, which uses a spherical pruning to approximate the Pareto front. It has been already used with success for controller design with several performance objectives and robustness requirements (Reynoso-Meza et al., 2013). The implementation used is the spMODE [1] algorithm (Reynoso-Meza et al., 2010).
- An interior-point NLP solver (IPOPT) together with an automatic-differentiation framework (CasADi [2]) for numerical optimal control which, following the direction of the cost-function gradient, finds an efficient and suitable path from the initial guess to the (possible local) optimum. This has been successfully used in dynamic programming with sequential and simultaneous approaches (Andersson et al., 2012; Martí et al., 2014).

The former has already been used in Boada et al. (2016) for this purpose. It was selected because:

- improves convergence by using an external file to store solutions and include them in the evolutionary process.
- improves spreading by using the spherical pruning mechanism (Reynoso-Meza et al., 2010).
- improves pertinency of solutions thanks to a bound mechanism in the objective space, as described in Reynoso-Meza et al. (2012).

Therefore, the objective here is testing the second alternative in the same problem in order to compare performance, pointing out particular advantages and drawbacks of both alternatives.

5. RESULTS

We carried out the dynamic optimization of (3) using the tools described in subsection 4.2. In order to evaluate the dynamic behavior of (1), we start from the system initially at equilibrium. This is:

$$x(0) = [\frac{k_{mB} C_{gB}}{d_{m_A}}, \frac{k_{mB} C_{gB} \cdot k_{pA}}{d_{m_A} \cdot d_A}, 0, 0, 0, 0, 0, 0, 0]$$

Then we set a step $x_9(0) = 50$ to act as input excitation and let the system relax until reaching again the steady state in order to effectively evaluate sensitivity and precision.

The optimization with spMODE started with an initial population of candidate solutions, chosen randomly within a normal distribution in the parameters search space (provided in Table

[1] Tool available in `http://www.mathworks.com/matlabcentral/fileexchange/39215`

[2] Tool available in `https://github.com/casadi/casadi/wiki`

4). An approximation of the Pareto front with 46 solutions of our MOOP problem was obtained (green curve in Figure 2), together with the Pareto set containing their corresponding kinetic model parameters. These solutions show, as expected, a trade-off between good sensitivity (low values of J_1) and good precision (low values of J_2).

Now, the MOOP (3) is slightly modified: we set an additional constraint with an user-defined upper bound in J_1, denoted by $\bar{J}_1$, and only J_2 is in the objective function. In this way the original MOOP is cast as a set of SOOP (allowing thus a parallel implementation with IPOPT) as follows:

$$\begin{aligned} \min_{\theta \in \mathbb{R}^{10}} \; & J_2(\theta) \in \mathbb{R} \\ \text{subject to:} \quad & \text{System dynamics (1)} \\ & 1 < P(\theta) < 10000 \\ & J_1(\theta) \leq \bar{J}_1 \end{aligned} \tag{4}$$

Hence, provided a common initial guess for all the independent optimizations (4), randomly chosen as

$$\theta_0 = [5, 20, 0.02, 0.02, 150, 0.005, 1, 10, 10, 10],$$

and defining a well distributed grid of 46 points within the pertinency range for $J_1(\theta)$, an approximation of the Pareto front has been found (blue curve in Figure 2). As it can be seen, both solutions found by IPOPT and spMODE are practically the same. Note that, all these dominant solutions are the "optimal ones" in the Pareto sense.

Table 4 shows the current parameters obtained by the two algorithms for two different type of solutions within the Pareto front: one with high sensitivity and other with high precision.

The evolution of the protein C concentration for some characteristic optimal points is also depicted in Figure 2. The different

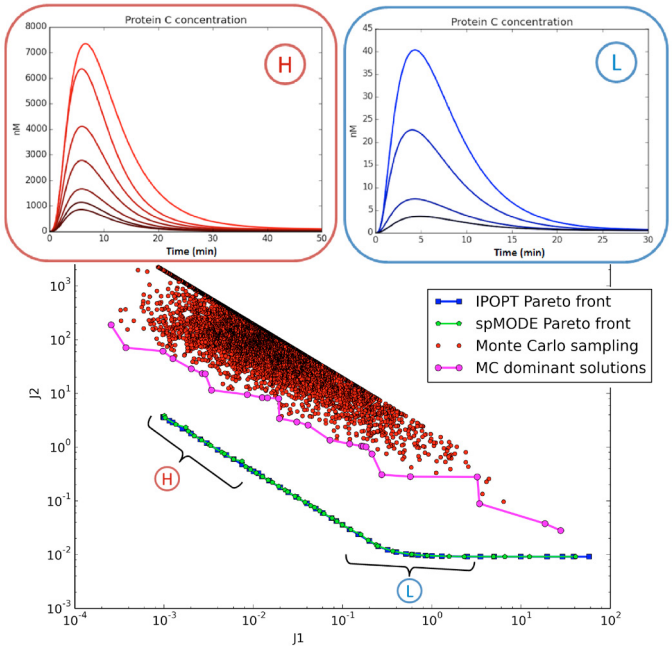

Fig. 2. Bottom: PF estimation obtained with the MC sampling (pink) spMODE (green) and with the NLP solver (blue). Up: Time evolution of the protein C concentration for two sets of Pareto optimal solutions (H≡high sensitivity, L≡high precision).

responses go from high sensitivity-low precision (H peak in red) to low sensitivity-high precision ones (L peak in blue).

Computational cost. The spMODE took 42.7 minutes to obtain its Pareto front, running in parallel in an Intel Core i7-4510U machine, while the IPOPT solver took 2.69 minutes to draw its one, also running in parallel in the same machine (5.84 minutes in a single core).

The number of objective function evaluations performed by the spMODE was 3100 while the IPOPT algorithm did 3624 (in total for all optimizations) plus 2231 evaluations of the objective function gradient, 2277 constraints Jacobian evaluations and 2185 Lagrangian Hessian evaluations.

Table 4. Pareto set optimal results.

Parameter	**Initial**	**Sensitivity**		**Precision**	
	Range	IPOPT	spMODE	IPOPT	spMODE
kmACgA*	[1 200]	13.041	19.123	1.0012	1
kmBCgB	[1 200]	1	1	14.913	1
kmCCgC	[1 200]	13.041	19.123	1.0012	1
k_{pB}	[1 100]	1	1	10.565	19.742
k_{pC}	[1 100]	9.5174	5.8916	1.0012	1
d_B	[0.01 0.3]	0.01	0.01	0.0221	0.01
d_C	[0.01 0.3]	0.2611	0.3	0.2996	0.3
γ_1	[50 200]	161.6	200	146.92	107.16
γ_3	[1e-4 0.5]	0.0001	0.0001	0.0205	0.0001
γ_4	[5e-4 5]	0.8886	0.0005	0.5703	0.0005
γ_5	[1 100]	1	1	4.162	12.121

* kmACgA takes the same value as kmCCgC because gene A and gene C are physically in the same plasmid.

For completeness, a random Monte-Carlo (MC) sampling was performed (red points in Figure 2). In this case, a dominance filter is required in order to select the best solutions which will conform the Pareto front approximation (pink curve in Figure 2). Results show that the MC sampling covers a large region in the objectives space, but sometimes outside of the pertinency box, as expected, because there is no simple way to focus a random search in it. In addition, the Pareto Front approximation obtained from the MC sampling is clearly worse than the ones obtained using optimization algorithms, both in accuracy (in the considered pertinency region) and in computational effort (the MC sampling took 2 hours and 10 min).

6. CONCLUSIONS

In this work, a framework for obtaining a set of guidelines to aid the design of synthetic genetic devices with a desired behavior is proposed, based on a multi-objective optimization (MOOD) procedure. The result of the optimization is already a set of parameters that optimally achieve the desired function and dynamics, as it is encoded in the objective indexes. Therefore, a proper definition of the optimization indexes representing the desired behavior is a key point: an objective which is not properly representing the actual desired behavior will lead the optimization in a wrong direction, thus returning a parameters set that will give misleading design guidelines. This is a drawback, but still easier to handle with than defining the acceptable circuit behavior after a *random* search, like Monte Carlo sampling, which normally will give *suboptimal* solutions in complex systems.

After optimization, the obtained solutions, i.e. the design objectives together with the respective parameter sets, may be clustered hierarchically, or post-processed with any multivariate analysis statistical tool in order to get further insight into

the role of the different parameters. Indeed, further statistical processing is very efficient, as only a small set of data has to be processed (the solutions at the Pareto front), and this set is already ordered. This will allow us to reveal and understand associations of parameters and functionality.

Here, we tested two different tools for the optimization step in the design of a biological functional device. The results in our case study show that using an NLP solver with automatic differentiation to estimate the actual Pareto front is more efficient than a MOEA algorithm. Nevertheless, an evolutionary algorithm is a *global* optimizer, which means that it may obtain better approximations of the PF in other cases (or allowing a higher number of objective function evaluations), while the performance of a gradient-based optimizer depends a lot on the provided initial guess and on the particular system "smoothness", so it may be stuck in a *local* optimum. If this is the case, a combined evolutionary gradient-based approach, where the fast NLP solver computes a preliminary set of suboptimal solutions to be used later as the initial population for the MOEA, may be a good option.

We foresee that the presented approach can be extended to the analysis of interconnection of several devices. Nevertheless, this will be led in further work, as evident difficulties arise when dealing with larger networks.

REFERENCES

Alon, U. (2006). *An Introduction To systems biology. Design Principles of Biological Circuits*. Chapman & Hall/ CRC Mathematical and computational Biology Series.

Andersson, J., kesson, J., and Diehl, M. (2012). Casadi: A symbolic package for automatic differentiation and optimal control. In S. Forth, P. Hovland, E. Phipps, J. Utke, and A. Walther (eds.), *Recent Advances in Algorithmic Differentiation*, volume 87 of *Lecture Notes in Computational Science and Engineering*, 297–307. Springer Berlin Heidelberg.

Banga, J.R. (2008). Optimization in computational systems biology. *BMC Systems Biology*, 2, 47.

Basu, S., Mehreja, R., Thiberge, S., Chen, M.T., and Weiss, R. (2004). Spatiotemporal control of gene expression with pulse-generating networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(17), 6355–6360.

Boada, Y., Vignoni, A., Navarro, J.L., and Picó, J. (2015). Improvement of a cle stochastic simulation of gene synthetic network with quorum sensing and feedback in a cell population. In *2015 European Control Conference (ECC)*.

Boada, Y., Reynoso-Meza, G., Pico, J., and Vignoni, A. (2016). Multi-objective optimization framework to obtain model-based guidelines for tuning biological synthetic devices: an adaptive network case. *BMC Systems Biology*, 10(1), 27. doi: 10.1186/s12918-016-0269-0.

Chiang, A.W.T. and Hwang, M.J.J. (2013). A computational pipeline for identifying kinetic motifs to aid in the design and improvement of synthetic gene circuits. *BMC Bioinformatics*, 14 Suppl 16, S5.

Chiang, A.W.T., Liu, W.C.C., Charusanti, P., and Hwang, M.J.J. (2014). Understanding system dynamics of an adaptive enzyme network from globally profiled kinetic parameters. *BMC Syst Biol*, 8, 4.

Church, G.M., Elowitz, M.B., Smolke, C.D., Voigt, C.A., and Weiss, R. (2014). Realizing the potential of synthetic biology. *Nature Reviews. Molecular Cell Biology*, 15(4), 289–294.

Feng, X.j..J., Hooshangi, S., Chen, D., Li, G., Weiss, R., and Rabitz, H. (2004). Optimizing genetic circuits by global sensitivity analysis. *Biophysical journal*, 87(4), 2195–2202.

Gill, P.E., Murray, W., and Saunders, M.A. (2005). SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, 47(1), 99–131.

Goentoro, L., Shoval, O., Kirschner, M.W., and Alon, U. (2009). The incoherent feedforward loop can provide fold-change detection in gene regulation. *Molecular cell*, 36(5), 894–899.

Kim, J., Khetarpal, I., Sen, S., and Murray, R. (2014). Synthetic circuit for exact adaptation and fold-change detection. *Nucleic acids research*, 42(9), 6078–6089.

Ma, W., Trusina, A., El-Samad, H., Lim, W.A., and Tang, C. (2009). Defining network topologies that can achieve biochemical adaptation. *Cell*, 138(4), 760–773.

Martí, R., Rodriguez, T., Pitarch, J.L., Sarabia, D., and De Prada, C. (2014). Dynamic optimization by automatic differentiation using EcosimPro CasADi. In *XXXV Jornadas de Automática*, 354–361. CEA-IFAC.

Mattson, C.A. and Messac, A. (2005). Pareto frontier based concept selection under uncertainty, with visualization. *Optimization and Engineering*, 6(1), 85–115.

Medema, M.H., van Raaphorst, R., Takano, E., and Breitling, R. (2012). Computational tools for the synthetic design of biochemical pathways. *Nature Reviews Microbiology*, 10(3), 191–202.

Miettinen, K. (1999). *Nonlinear Multiobjective Optimization*, volume 12. Kluwer Academic Publishers, Boston.

Otero-Muras, I. and Banga, J. (2014). Multicriteria global optimization for biocircuit design. *BMC Systems Biology*, 8(1), 113.

Reynoso-Meza, G., García-Nieto, S., Sanchis, J., and Blasco, X. (2013). Controller tuning using multiobjective optimization algorithms: a global tuning framework. *IEEE Transactions on Control Systems Technology*, 21(2), 445–458.

Reynoso-Meza, G., Sanchis, J., Blasco, X., and Herrero, J.M. (2012). Multiobjective evolutionary algortihms for multivariable PI controller tuning. *Expert Systems with Applications*, 39, 7895 – 7907.

Reynoso-Meza, G., Sanchis, J., Blasco, X., and Martínez, M. (2010). Design of continuous controllers using a multiobjective differential evolution algorithm with spherical pruning. *Applications of Evolutionary Computation*, 532–541.

Vignoni, A., Oyarzún, D., Picó, J., and Stan, G.B. (2013). Control of protein concentrations in heterogeneous cell populations. In *Control Conference (ECC), 2013 European*, 3633–3639. IEEE.

Wächter, A. and Biegler, L.T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1), 25–57.

Way, J., Collins, J., Keasling, J., and Silver, P. (2014). Integrating biological redesign: Where synthetic biology came from and where it needs to go. *Cell*, 157(1), 151–161.