Automated Design Framework for Synthetic Biology exploiting Pareto Optimality

Irene Otero-Muras* and Julio R. Banga

BioProcess Engineering Group, IIM-CSIC, Spanish National Research Council, Vigo, Spain

E-mail: ireneotero@iim.csic.es

Phone: +34 986231930. Fax: +34 986292762

Abstract

In this work we consider Pareto optimality for automated design in synthetic biology. We present a generalized framework based on a mixed-integer dynamic optimization formulation that, given design specifications, allows the computation of Pareto optimal sets of designs, i.e. the set of best trade-offs for the metrics of interest.

We show how this framework can be used for (i) forward design, i.e., finding the Pareto optimal set of synthetic designs for implementation, and (ii) reverse design i.e. analyzing and inferring motifs and/or design principles of gene regulatory networks from the Pareto set of optimal circuits.

Finally, we illustrate the capabilities and performance of this framework considering four case studies. In the first problem we consider the forward design of an oscillator. In the remaining problems, we illustrate how to apply the reverse design approach to find motifs for stripe formation, rapid adaption, and fold-change detection, respectively.

Keywords

Pareto optimality, global optimization, multiobjective, mixed-integer, motifs, design principles, regulatory networks

1 Introduction

One challenge of synthetic biology is the design and implementation of transcriptional regulatory circuits that can mimic desirable properties found in naturally occurring systems. This entails handling increasing levels of complexity, both in the number of regulatory regions and in the nature of the tasks that the circuits can accomplish. A number of computer-aided approaches and tools for synthetic biology has been presented during the last decade (*1–11*). A recent breakthrough in the field is the development of Cello by Nielsen et al. (*12*), a hardware language for programming living cells applying principles of electronic design automation (EDA) that allowed the authors to design circuits that double the plateau first noted by Purnick and Weiss (*13*) in the number of regulated promoters.

The high levels of complexity found in natural regulatory networks could be in part explained by the fact that they perform multiple tasks, and they are therefore subject to fundamental trade-offs (*14, 15*). Mimicking an evolutionary process through a multiobjective optimization procedure will lead to find those features or evolutionary traits that show the best trade-offs between a set of advantageous properties (selected as optimization objectives) in what is called Pareto front of optimal solutions. From the perspective of synthetic circuits, a multiobjective formulation will push the design of synthetic circuits towards more complex functionalities, and in general terms, improved characteristics. For example, multicriteria optimization could be used in combination with recent advances (*16*) to find designs with good trade-offs between performance and protein burden.

In this work we present a generalized framework for automated design that allows the computation of Pareto optimal sets of designs and can be applied to tackle both forward

and reverse design problems (we focus on the first stages of the so called circuit engineering process (17)). The aim in the forward problem is to find synthetic designs for implementation that comply with *a priori* defined target performance requirements, whereas in reverse design the purpose is finding motifs, or patterns of interconnections/parameters, leading to a particular functionality in gene regulatory networks. We use mixed-integer optimization methods to search simultaneously topology and parameter spaces with high computational efficiency.

The use of optimization-based approaches for automated design in synthetic biology have been proposed, among others, by Francois and Hakim (18), Rodrigo et al. (19), Dasika and Maranas (20), Cao et al. (21) and Huynh et al. (22, 23). These studies considered single-objective optimization formulations. Here we will instead adopt a multiobjective (or multicriteria) perspective. In this framework, the optimal solution is not unique, but a set of optimal trade-offs (usually called the Pareto set).

Multiobjective optimization is widely used in engineering design (24–28), and is being increasingly used to analyze and design complex systems in many areas of science (14, 29–35). It is worth noting that the use of the multiobjective optimization (Pareto optimality) concept in biosystems has a relatively long history. The role of trade-offs and multicriteria optimization in the evolutionary adaptation of biochemical networks was already studied by a number of authors in the 1990s (36), with pioneering works dating back to the 1980s (37, 38). Initial applications of multicriteria optimization in metabolic engineering were presented by Vera et al. (39) using linear programming and by Sendin et al. (40) for the nonlinear case. By the same time, applications in bioprocess engineering were reported (41, 42). Initial multiobjective formulations for flux balance analysis were developed by Vo et al. (43) and Sendin et al. (44), with subsequent studies providing experimental confirmation (45, 46). During the last decade, a number of authors have considered this kind of approach for different biological applications, including molecular biology (47), neuroinformatics (48) conservation biology (49), RNA design (50), bacterial adaptability (51, 52), microbial strain

design (53, 54), metabolic network analysis (44, 55–60), tuning of biological devices (61), synthetic DNA library assembly (62) and forward design of genetic networks (63–65).

Here we have a double objective: (i) to extend and generalize the multiobjective optimization methodology for forward design introduced by Otero-Muras and Banga (64) to the reverse design of gene regulatory networks, and (ii) to illustrate the capabilities of the framework for both reverse and forward problems with four case studies. First, starting from a library of biological parts we design an oscillator with optimal stability and tunability of the period (a good property for entrainment purposes). The second example applies the reverse design approach to find motifs capable of stripe formation in a morphogen gradient. In the third and fourth case studies we search for motifs for rapid adaption fold-change detection, respectively.

2 Results and Discussion

2.1 Forward problem: finding Pareto optimal synthetic circuits

Synthetic biology aims to use a systematic forward-engineering design cycle (66, 67). Here we consider the problem of forward design as finding, among all the possible combinations of devices from a library of standard parts, the circuit (or circuits) capable of achieving an *a priori* defined functionality.

We approach this problem by means of a formulation based on Multiobjective Mixed-Integer Nonlinear Programming Optimization (see section 3). To this aim, it is first necessary to model the circuits' dynamics in a suitable mixed-integer framework.

In the context of forward design, models need to ensure modularity and easy translation into an implementable circuit, also for an effective transfer from/to experimental data (68, 69). Here we use the formalism of biological components (70) and consider four minimal constitutive parts: promoters, ribosome binding sites (RBS), protein coding regions (containing the sequence information needed to create a functional protein chain) and termi-

nators. We adopt the formalism by Pedersen and Phillips (71) to associate a set of reactions to each constitutive part and obtain the reaction network of a device by combination of the reactions of its constitutive parts (we consider devices constituted by a promoter, a ribosome binding site, a protein coding region and a terminator). In this way, the reaction network of any circuit (or combination of devices (70)) is built automatically starting from the reactions of the constitutive parts. This modeling framework facilitates further connections to the Synthetic Biology Open Language SBOL (72) by means of recently developed annotation methodologies (73). Moreover, the libraries can be extended in an easy manner to incorporate information from improved characterization of parts and devices (74).

Figure 1: Design of an oscillator from a library of standard parts.

Starting from a given library with nG promoters, nR ribosome binding sites, nP protein

coding regions and nT terminator sequences, the number of possible device configurations is $M = nG \times nR \times nP \times nT$. If we label every possible device with an integer index $i = 1, \dots, M$ and build a vector $y \in \mathbb{Z}^M$ of binary variables such that y_i is one if the device i is part of the circuit and zero otherwise (75), the structure of a gene regulatory circuit is completely defined by the binary vector y . Although the parameters (kinetic rate constants associated to each minimal part) are in principle fixed, any parameter(s) can be selected to be tuned as part of a vector x of real decision variables in the optimization problem.

Assuming mass action kinetics for all the reactions and taking into account the dynamics of all the species involved in the network including promoters, mRNA, proteins and intermediate complexes, the dynamics of the a circuit, in terms of x, y (the vectors of real and binary variables previously introduced) reads:

$$\dot{z}(t) = N(y)v(x, y, k) \quad (1)$$

where $N(y)$ is the stoichiometric matrix and $v(x, y, k)$ is the vector of rates of the reaction network (depending nonlinearly on the species in accordance with the mass action law). The vector k contains all the parameters (kinetic rate constants) that remain fixed during the optimization process. Here it is important to remark that, since we assume mass action kinetics, it is possible also to perform stochastic simulations of the dynamics using the SSA algorithm (76).

Once the dynamics is encoded in a mixed-integer framework, the forward design problem is formulated as a multiobjective optimization problem and solved as indicated in detail in the section 3.

Design of an oscillator with optimal properties. Next we apply our method to find, starting from a library of standard parts, the best oscillator for implementation according to *a priori* defined criteria. In order to ensure that the response of the system is a sustained oscillation, we use as a constraint the first peak of the normalized autocorrelation function

(77, 78) (which needs to be above a threshold for an oscillation). Our first design criterion is the tunability (79) of the period, i.e., the capacity of the oscillator to adapt the period according to the cell demands. This property, postulated as an evolutionary aim by Tsai et al. (80), increases the capacity of an oscillator to be successfully entrained by an external oscillatory signal. As a second objective we select the leading Floquet exponent of the monodromy matrix (77) to ensure that the stability of the limit cycle is not compromised.

We start from a database of biological parts adapted from Pedersen and Phillips (71) (we have extended the library to incorporate the degradation of bound repressor) containing 4 promoters, 1 ribosome binding site, 1 terminator and 11 protein coding regions for the proteins *cIR*, *tetR*, *araC*, *lacI*, *luxI*, *luxR*, *lasR*, *lasI*, *ccdB*, *ccdA*, *ccdA2* (See Fig. 1 A). This makes a total of 44 possible devices. Labeling each device with a number from 1 to 44, the structure of a circuit is completely defined by a vector y with 44 binary entries.

We solve the multiobjective optimization problem constraining the maximum number of devices allowed in the circuit to 3. The obtained Pareto front, depicted in Fig. 1 C is constituted by three ring oscillators. Circuit P1 shows the highest period tunability, whereas circuit P3 is the one with greater stability of the limit cycle (this property is shown to be correlated to the robustness of an oscillator with respect to molecular intrinsic noise (77)). Following the criterion proposed in section 3, i.e. the shortest distance to the utopia point U , the circuit P2 is the best circuit for implementation, i.e. the one that shows the best compromise between period tunability and stability of the limit cycle. For comparative purposes we depict the Repressilator oscillator (81), corresponding to the NP circuit in Fig. 1 C. As it can be deduced from the figure, the Repressilator oscillator is dominated by circuits P1, P2 and P3, i.e., it is not Pareto optimal with respect to the period tunability and stability of the limit cycle.

2.2 Reverse Design: Inferring design principles of gene regulatory networks

Synthetic biology can help us to infer design principles, increasing our insights and understanding of the organization and function of naturally occurring genetic networks (82–85). In the context of automated design, we define the problem of reverse design as finding, from a weighted N -dimensional hypergraph, those motifs or patterns of interconnections/parameters, leading to a functionality defined *a priori*.

Here we tackle the reverse design of gene regulatory networks design using the Multi-objective Mixed-Integer Nonlinear Programming Optimization approach included in section 3). To this aim, we first accommodate the dynamics of gene regulatory networks into a mixed-integer nonlinear framework.

Transcriptional networks can be represented by weighted directed graphs in which nodes are genes and connections represent regulation. Within this framework the regulation from gene G_i to gene G_j is characterised by two numbers: an integer $y_{ij} \in \{-1, 0, 1\}$, coding for inhibition (-1), no action (0), or activation (1), and a strictly positive weight $x_{ij} \in \mathbb{R}_{>0}$. A network with N genes is therefore completely characterized by a vector of N^2 integer and N^2 real variables (see the Supporting Information for details).

Following Munteanu et al. (86) the effective regulating input to a gene G_i is given by:

$$\chi_i = \sum_{j=1}^n \omega_{ji} z_j + \alpha_i I$$

where $\omega_{ji} = y_{ji} x_{ji}$, and the term $\alpha_i I$ reflects the effect of external inputs (in case the gene G_i is only affected by internal gene-gene interactions, the coefficient $\alpha_i = 0$). The transcription rate is proportional to the sigmoidal-filtering of the total contribution, such that the balance for the protein z_i encoded by G_i reads:

$$\dot{z}_i = \frac{1}{1 + \exp(a - b(\chi_i))} - \delta z_i \quad (2)$$

where parameters a and b control the steepness and location of the threshold value of the regulation function, and δ is the protein degradation rate constant (see the Supporting Information for more details).

For a 3-dimensional network with genes A , B and C , the ODE system describing the dynamics of the network reads:

$$\begin{aligned}\dot{A} &= \frac{1}{1 + \exp(a - b(I + \omega_{AA}A + \omega_{BA}B + \omega_{CA}C)))} - \delta A \\ \dot{B} &= \frac{1}{1 + \exp(a - b(\omega_{AB}A + \omega_{BB}B + \omega_{CB}C))} - \delta B \\ \dot{C} &= \frac{1}{1 + \exp(a - b(\omega_{AC}A + \omega_{BC}B + \omega_{CC}C))} - \delta C.\end{aligned}\tag{3}$$

This model representation has been biologically-verified, extensively employed in the study of developmental gene networks (87), and it is particularly suitable here due to its generality and easy accommodation into the mixed-integer framework. Simplified mechanistic models (88) could be a good alternative in case that we have more detailed information about the kinetics of the systems under study.

Importantly, our framework can easily accommodate a stochastic description of the dynamics. Recently developed methods (89) allow the efficient stochastic simulation of gene regulatory networks with general kinetics under mild assumptions. In order to consider the effect of molecular noise in the design, the incorporation of these methods for efficient stochastic simulation within our framework is subject of ongoing work.

After encoding the dynamics into the mixed-integer framework, the design problem is formulated as a multiobjective optimization problem and solved as indicated in section 3.

Finding motifs for stripe formation. Stripe formation is a key spatial feature in the early stages of the embryogenesis (86). Next we apply the reverse design method to find stripe-forming motifs, i.e. patterns conferring to a tissue of N isogenic cells the capability to form a stripe of gene expression in response to a morphogen gradient. Previous studies in tissue development at *Drosophila* embryos (90) revealed that the feedforward motifs

Figure 2: Finding stripe-forming motifs by multiobjective optimization automated based design. A) Search space and decision variables, B) Objective functions, C) Pareto front of optimal trade-offs (blue square dots), red square dots represent dominated solutions.

are capable of this patterning function, and synthetic morphogen sensors have been implemented using feedforward architectures (91). In a computational work, Munteanu et al. (86) explored the capacity of the four types of incoherent feedforward loops for stripe formation. Using exhaustive exploration through the topology and parameter spaces (performing simulations for a large sampling of the parameter space for each possible topology), they

found that both incoherent type 1 (IFF1) and incoherent type 3 (IFF3) feedforward loops are capable to form an easily detectable single-stripe pattern under a morphogen gradient.

Here we search for stripe-forming motifs in 3-dimensional gene networks applying optimization instead of exhaustive exploration. Our search space is the weighted hypergraph in Fig. 2 A, with decision variables $y_{AB}, y_{BB}, y_{CB}, y_{AC}, y_{BC}$ and y_{CC} in $\{-1, 0, 1\}$ and $x_{AB}, x_{BB}, x_{CB}, x_{AC}, x_{BC}$ and x_{CC} in $\mathbb{R}_{>0}$.

As a first objective to optimize we consider the quality of the stripe, using a score to measure the distance to an *ideal stripe* represented in Fig. 2 B (centered in the origin and optimal in terms of height and width). As a second objective we consider the protein production cost, as defined by Zaslaver et al. (92). Recent works (93) indicate that the economy of protein production is important to cell physiology supporting the selection of protein cost as a meaningful objective.

The morphogen gradient is simulated, following Munteanu et al. (86) by an input into gene A , which takes the form $I = Md^c$, with M being the morphogen concentration at the left boundary of the tissue, d is the reduction of morphogen concentration in each subsequent cell of the morphogen gradient and c is the increasing cell index $c = 1, 2, \dots, N$. In order to obtain the system response to the morphogen gradient, we compute the steady state levels of the proteins in every cell using Eq. 3. The profiles of the proteins A , B and C are obtained by plotting the corresponding steady state levels as a function of the cell index c .

The complete Pareto front is illustrated in Fig. 2, where the topology and parameters for each circuit $P1, \dots, P8$ are depicted together with the associated system's response.

Examination of the set of optimal trade-off solutions leads to the following observations. First, both IFF1 and IFF3 feedforward structures appear among the non-dominated solutions. This result is in agreement with that reported by Munteanu et al. (86), where analytic and computational approaches were employed to establish IFF1 and IFF3 as core topologies for stripe formation (it is important to remark that our optimization method requires orders of magnitude less computational effort). In addition, the Pareto analysis reveals that,

whereas IFF3 provides the best stripe quality at the highest production cost, IFF1 provides good compromises between both objectives. In fact, the topology of the optimal circuits varies following a structured logic as we move along the Pareto front: the IFF3 structure provides high values of protein cost, at intermediate values of protein cost we find IFF1 structures reinforced with negative regulation from B to C , and at low values of protein cost we find IFF1 structures with very low regulation from B to C .

For comparison purposes, we also depict four circuits found by exhaustive exploration (86) corresponding to dominated points $NP1, \dots, NP4$. As expected, circuits found by optimization outperform in terms of the objectives selected (quality of the stripe and protein cost) those found by other methods that do not take optimality into account.

Finding motifs for *rapid* adaptation. Adaptation is the ability of a system to re-set after responding to a stimulus, allowing cells to maintain homeostasis in presence of perturbations. In the context of enzyme networks, Ma et al. (94) described formally the adaptive response in terms of its sensitivity (the height of output response relative to the initial steady-state value) and precision (the difference between pre- and post-stimulus steady states). Perfect adaptation is achieved when the system's response returns exactly to the same level before stimulation. In the same work, they explored the full range of simple architectures (and performing an exhaustive sampling in the parameter space for each architecture) taking into account all the possible directed links among three nodes. Considering Michaelis-Menten kinetics, they found two major core topologies that can achieve robust adaptation: a negative feedback loop with a buffering node and an incoherent feedforward loop with a proportioner node.

Here we are interested in finding motifs in gene regulation that can achieve a fast adaptation after sensing a signal, returning rapidly to the prestimulus steady state.

We start from 3 gene networks with an input into gene A . The system's response (profiles of the proteins A , B and C upon induction) is given by Eq. 3, and the dynamics of the target protein C is evaluated. The search space is in this case the weighted hypergraph in

Figure 3: Finding motifs for rapid adaptation. A) Search space and decision variables. B) Objectives to optimize. C) Pareto front of solutions for a maximum number of two active connections. C) Pareto front of optimal trade-offs for a maximum of three active connections.

Fig. 3 A, with 9 integer decision variables $y_{AA}, y_{BA}, y_{CA}, y_{AB}, y_{BB}, y_{CB}, y_{AC}, y_{BC}$ and y_{CC} in $\in \{-1, 0, 1\}$ and 9 real decision variables $x_{AA}, x_{BA}, x_{CA}, x_{AB}, x_{BB}, x_{CB}, x_{AC}, x_{BC}$ and x_{CC} in $\mathbb{R}_{>0}$. As objectives to optimize we consider the sensitivity (or amplitude) of the response (see Fig. 3 B) and a measure that indicates the speed of the response, aiming to find circuits that rapidly arrive to their state before perturbation. For this purpose, we find that the area below protein C dynamics provides an effective objective to minimize (note that this area is related to the cost C protein production as defined by (92)).

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

In order to find minimal network structures leading to adaptation, we first solve the multiobjective optimization problem imposing a maximum number of 2 active connections. Under this constraint, we do not found any structure capable of adaptation (the corresponding Pareto front is depicted in Fig. 3 C). In a second step, we solve the multiobjective optimization problem for a maximum of 3 active connections, obtaining the Pareto front in Fig. 3 D, showing how the two objectives are in this case clearly in a strong trade-off. We have found different structures with only three active connections that achieve perfect adaptation showing optimal trade-offs between sensitivity and velocity. The Pareto front depicted corresponds to the type 1 incoherent feedforward motif (IFF1) (circuits from P1 to P20 are of the IFF1 type and the values of the regulation strengths vary as we move along the Pareto front).

26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

Interestingly, the Pareto front does not improve by increasing the number of maximum number of connections, leading to the conclusion that adding more connections will not have any beneficial effect in the adaptive response.

Finding motifs for fold-change detection. Certain cellular sensory systems display what is called fold-change detection (FCD), i.e. they respond to stimuli depending only on fold changes in the input, not on absolute levels (*14*). In a recent work, Adler et al. (*95*) performed an exhaustive analytical screen of all three-node circuits, comprising around 500000 topologies, searching for circuits capable of fold-change detection (FCD circuits). They found that two recurrent circuits in biological systems, the incoherent feedforward and the nonlinear integral-feedback loop, are among the very few minimal FCD circuits that trade-off speed and response amplitude. The authors arrive to this conclusion after an exhaustive exploration (no optimization is performed), in which the speed and amplitude values of different circuits are plotted in the objective space, and both IFF1 and NLIF motifs are found to lie in the Pareto boundary. Here, we apply our multiobjective optimization methodology to search for FCD circuits with optimal response in terms of speed and response amplitude.

As Adler and coauthors remark (95), exact (or perfect) adaptation is a pre-requisite for fold-change detection. Here we search for circuits displaying perfect adaptation that trade-off speed and amplitude of the response (as in the previous case study), but in this case imposing FCD condition as a constraint (see Fig. 4 A). We set the maximum number of connections to 4. In a first step, we solve the multiobjective optimization problem with 9 integer decision variables $y_{AA}, y_{BA}, y_{CA}, y_{AB}, y_{BB}, y_{CB}, y_{AC}, y_{BC}$ and y_{CC} in $\{-1, 0, 1\}$ and 9 real decision variables $x_{AA}, x_{BA}, x_{CA}, x_{AB}, x_{BB}, x_{CB}, x_{AC}, x_{BC}$ and x_{CC} in $\mathbb{R}_{>0}$ (see Fig. 3 A). As in (95), both IFF1 and NLIF circuits (see Fig. 4 B) are found to display fold-change detection with optimal speed and amplitude values. In comparison with exhaustive exploration, the number of function evaluations required by our approach is orders of magnitude lower (this translates into drastically lower computation times). In a second step, we fix the structural variables and solve the multiobjective optimization problem with decision variables $x_{AA}, x_{BA}, x_{CA}, x_{AB}, x_{BB}, x_{CB}, x_{AC}, x_{BC}$ and x_{CC} for both structures IFF1 and NLIF. In both cases we obtain the Pareto front depicted in Fig. 4 C (the values of the regulation strengths vary as we move along the Pareto front from P1 to P20).

Figure 4: Motifs for Fold-Change Detection. A) Change Fold Detection Property. B) Motifs for Fold-Change Detection C) Pareto front of solutions for IFF1 and NLIF motifs. The complete Pareto front has been obtained in less than 20 min CPU time using an Intel 2.8 GHz Xeon.

3 Methods

3.1 Multiobjective mixed-integer nonlinear programming formulation of the automated design problem

Optimization-based automated design of transcriptional regulatory networks aims to find the circuit (or circuits) with the best overall performance among the set of all possible designs (search space). Since we are interested in searching simultaneously topology and parameter spaces, we employ a mixed-integer description of the gene regulatory network dynamics such that a circuit in the search space is completely characterized by a vector of integer (and/or binary) variables (characterizing the topology or structural configuration) and a vector of real variables (the tunable parameters). Within the mixed-integer modeling framework, the dynamics of gene regulatory networks is encoded in a system of Ordinary Differential Equations (ODEs) of the form:

$$\dot{z}(t) = f(z, y, x, k), \quad z(0) = z_0 \quad (4)$$

where $z \in \mathbb{R}^N$ is the vector of dynamic state variables containing the levels of the species involved; $x \in \mathbb{R}^R$ is the vector of continuous decision variables containing the tunable parameters; $y \in \mathbb{Z}^M$ is the vector of integer (or binary) decision variables encoding the circuit model structure; $k \in \mathbb{R}^K$ is a vector of fixed parameters. Note that this representation is generic (neither constrained to specific kinetics, nor to a particular granularity), and can be easily accommodated to upcoming advances in the modeling of synthetic circuits to incorporate, for example, complex aspects of mammalian regulation.

Design targets can be encoded in objective functions of the form $J(\dot{z}, z, x, y, k)$ such that the predefined behavior is achieved when J reaches its minimum. We can define multiple design criteria by a vector of objective functions of the form $J = (J_1, J_2, \dots, J_S)$.

The automated design of a gene regulatory network is formulated as finding a vector

$x \in \mathbb{R}^R$ of continuous variables and a vector $y \in \mathbb{Z}^M$ of integer variables that minimize the vector $J = (J_1, J_2, \dots, J_S)$ of objective functions:

$$\min_{x,y} J_1(\dot{z}, z, x, y, k), J_2(\dot{z}, z, x, y, k), \dots, J_S(\dot{z}, z, x, y, k) \quad (5a)$$

subject to the circuit dynamics in the form of ODEs or Differential Algebraic Equations (DAEs) with the state variables z and additional parameters k :

$$\xi(\dot{z}, z, x, y, k) = 0, \quad z(t_0) = z_0, \quad (5b)$$

to additional requirements in the form of equality and inequality constraints:

$$h(z, x, y, k) = 0, \quad (5c)$$

$$g(z, x, y, k) \leq 0, \quad (5d)$$

and to upper and lower bounds for the real and integer decision variables:

$$x_L \leq x \leq x_U, \quad (5e)$$

$$y_L \leq y \leq y_U. \quad (5f)$$

The solution of the multiobjective optimization mixed-integer nonlinear programming (MO-MINLP) problem is a set of Pareto optimal points (the Pareto front) (96). The Pareto front represents the optimal trade-offs among the design criteria (97). A feasible circuit defined by (x^*, y^*) is a Pareto optimal solution of the multiobjective optimization problem if it is not dominated by other feasible circuits (we say that $J(x^*, y^*)$ dominates $J(x^{**}, y^{**})$ if $J(x^*, y^*) \leq J(x^{**}, y^{**})$ for all J_i ($i = 1, \dots, S$) with at least one strict inequality).

3.2 Computing the Pareto front of optimal trade-offs

The efficiency in solving the optimization MO-MINLP problem is crucial for the ability to handle high levels of complexity. We need to remark that both reverse and forward design problems involve usually large search spaces combining real and integer variables. Due to the high nonlinearity and the presence of integer variables, the expected Pareto front might be discrete and/or non-convex. Global optimization of single-objective MINLPs is a very active research area (98).

We use the ε -constraint strategy because, on the one hand, it allows to compute non-convex regions in Pareto front (unlike for example the weighted-sum method (96)), and on the other hand it allows to compute discrete Pareto fronts (other methods like Normal Boundary Intersection will fail in this case (96)). By means of the ε -constraint strategy we transform the multiobjective MINLP into a set of single objective MINLPs.

The proposed optimization process is composed of the following steps, considering two objective functions J_1 and J_2 :

1. Search for the optima of each of the individual objectives:

$$(x_1^*, y_1^*), (x_2^*, y_2^*)$$

2. Compute the individual objective bounds as:

$$\underline{J}_1 = J_1(x_1^*, y_1^*), \quad \overline{J}_1 = J_1(x_2^*, y_2^*),$$

$$\underline{J}_2 = J_2(x_2^*, y_2^*), \quad \overline{J}_2 = J_2(x_1^*, y_1^*).$$

At this step we can already check whether the objectives are actually competing. Note that, if the objectives are competing, $[\underline{J}_1 \quad \overline{J}_2]$ and $[\overline{J}_1 \quad \underline{J}_2]$ are the extremes of the Pareto front (see the Supporting Information for details).

3. Select the objective function to be minimized, denoted in what follows as the primary objective (without loss of generality let us take J_1 as the primary objective).

4. Define m discretisation intervals on the axis corresponding to the secondary objective J_2 . To this purpose, for the non-minimized objective J_2 , generate a vector

$$\varepsilon = [\varepsilon_1, \dots, \varepsilon_i, \dots, \varepsilon_{m+1}]$$

such that $\varepsilon_1 \leq \underline{J}_2$, $\varepsilon_{m+1} \geq \overline{J}_2$ and $\varepsilon_1 < \varepsilon_2 < \dots < \varepsilon_{m+1}$.

5. Solve the MINLP:

$$\min_{w,y} J_1(\dot{z}, z, x, y, k)$$

subject to:

$$\varepsilon_k \leq J_2(\dot{z}, z, x, y, k) < \varepsilon_{k+1}$$

for $k = 1, \dots, m$ by means of a MINLP solver. In this way, we solve a MINLP for each of the m intervals (see Fig. 5 A). In order to practically implement the strict inequality $J_2(\dot{z}, z, x, y, k) < \varepsilon_{k+1}$ we can perturb it slightly to transform it into a non strict inequality (i.e. $J_2(\dot{z}, z, x, y, k) \leq \varepsilon_{k+1} - \epsilon$) where $\epsilon > 0$ is small enough.

6. Evaluate the solutions obtained (classifying them into dominated and non-dominated) and construct the Pareto front with the non-dominated ones (see Fig. 5 B-C).

Note that only one solution can be found in each interval. With an adequate discretisation, this technique ensures a good distribution of the Pareto optimal points.

Each of the MINPLs to be solved can be regarded as an outer mixed-integer nonlinear problem subject to an inner initial value problem (that is, the dynamics of the encoded

system). This class of problems is in fact a particular case of Mixed-Integer Dynamic Optimization (MIDO). In general terms, MIDO problems can be solved using either deterministic or stochastic global optimization approaches. The advantage of deterministic methods for MIDO (e.g. (99–101)) is that in principle they can guarantee global optimality of the solutions. However, in practice, the current state of the art is that they do not scale up well with problem size, so the computation times associated usually explode as the size of the problem increases.

Figure 5: Multiobjective Mixed Integer Nonlinear Optimization framework for automated design of gene circuits. A) Using the ϵ -constraint strategy we transform the multiobjective MINLP into a set of single objective MINLPs that we solve with efficient hybrid methods to compute the Pareto front of optimal trade-off solutions. B) Possible scenario for a discontinuous Pareto using ϵ -constraint strategy (some intervals might contain dominated solutions or not contain any feasible solutions). C) Typical scenario for a continuous Pareto using ϵ -constraint strategy (higher resolution will be obtained by decreasing the size of the discretization intervals). D) As an additional criterion for forward design, we select among the Pareto optimal solutions the one showing the minimum distance to the utopia point, denoted as knee point. E) To evaluate how the topology evolves along the Pareto front, we can cluster the solutions in the objective space by standard techniques (see section 3.3).

Therefore, currently the pragmatic alternative is to use stochastic global optimization

methods, which typically can produce solutions in the vicinity of the global one in rather reasonable computation times. The price to pay is that we loose guarantees of global optimality. In this context, purely stochastic methods (such as simulated annealing, genetic algorithms, etc.) are very costly since they require a very large number of evaluations of the cost function, and we need to solve an initial value problem for each evaluation of the cost function. Fortunately, a number of hybrid metaheuristic methods (102–106) have been recently developed combining global stochastic methods with efficient (local) deterministic optimization methods.

For our particular problem instances, we have extensively compared the efficiency of several hybrid MINLP solvers, combining three different types of stochastic global search with an efficient local mixed-integer solver, (MISQP) by Exler and Schittkowski(107): (i) the enhanced scatter search algorithm (eSS) by Egea et al. (108), the mixed-integer tabu search algorithm (MITS) by Exler et al. (104) and the mixed-integer ant colony optimization (ACOm) by Schlueter et al. (105). As an alternative to MISQP, we also considered another competitive local MINLP solver, NOMAD (109). From these comparisons we have concluded that the enhanced scatter search algorithm (eSS) by Egea et al. (108), combined with MISQP, systematically results in the most robust and efficient approach. Therefore, all the results presented here have been obtained with eSS-MISQP. A more detailed discussion on the computational efficiency and scalability of the method can be found in section 3.4.

3.3 Knowledge discovery from the Pareto front

3.3.1 Selection of best circuit(s) for implementation in forward design.

In the forward design problem, we have to select the best circuit(s) for implementation. In absence of prior preferences and/or additional information, all the solutions in the Pareto front are equally good (110). Therefore, the decision maker (designer) must introduce some preference information, and evaluate these additional criteria along the Pareto front, in order to select the best solution to be implemented (111). In a multiobjective optimization

problem, the utopia point is the point that optimizes all objective functions simultaneously as if they were considered in isolation (here we use the notation by Kim and de Weck (112), see the Supporting Information for more detail). In absence of prior or posterior preferences, a compromise solution is to select the solution closest to the utopia point. This solution is usually called the knee point (see Fig. 5 D). The knee point is furthest from the extreme points of the Pareto front, and represents a good compromise among the various objectives (97).

3.3.2 Inferring design principles in forward and reverse design.

The main aim of the reverse problem is to infer motifs of interconnections leading to particular functionalities so as to extract design principles of gene regulatory networks. Extracting information from solutions generated during the multiobjective optimization problem might also be of great help in the forward design problem, e.g. as a way to provide a deeper insight about the problem to the designer or decision maker (113). In some cases (low dimensional solution data-sets, few design objectives) it is easy to directly extract knowledge from the the solutions generated during the optimization without the need of automated techniques. However, problems involving large solution data-sets would require of data mining methods, in the same vein of automated innovization approaches recently used in engineering design (114). Here we introduce a number of techniques adequate to the nature of the (forward and reverse) design problems in synthetic biology. For a general review on data mining methods in multiobjective optimization, we recommend the recent papers by Bandaru et al. (113, 115).

Clustering of solutions in the decision space. In a multiobjective optimization problem, the $R + M$ dimensional space formed by the variables (note that R and M are the number of real and integer or binary variables, respectively) is called decision space, while the S dimensional space formed by the objectives is called objective space (an illustrative scheme is depicted in Fig. 6).

Figure 6: Decision and objective spaces: mapping of solutions from a two-dimensional integer decision space to a two-dimensional objective space, adapted from (113).

In order to extract patterns of interconnections/parameters leading to optimal trade-offs among the design objectives, we suggest to cluster the non-dominated solutions in the decision space using standard techniques. Then, we can visualize these clusters in the objective space to reveal how the patterns evolve through the Pareto front (see Fig. 5 E). To illustrate the approach, let us get back to the Pareto front obtained in the example of stripe-forming motifs in Fig 2. There are 8 Pareto optimal points corresponding to the circuits P1 to P8 in the figure. Each circuit is characterized by a 6-dimensional real vector $yx = (y_{AB}x_{AB}, y_{BB}x_{BB}, y_{CB}x_{CB}, y_{AC}x_{AC}, y_{BC}x_{BC}, y_{CC}x_{CC})$. We use the k-means algorithm (116) to partition the vectors in k clusters. In this case we find 2 clusters corresponding to two different motifs: IFF3 (P1) and IFF1 (P2 to P8), revealing how the motif structure varies as we move along the Pareto front (from high cost-good performance to low cost-poor performance). For integer variables we can define the distance between two circuits as the Hamming distance between their vectors (116), (i.e. the number of regulations that differ in the two networks), and use the distance matrix for clustering, as it is done for example in (94). More sophisticated techniques can be used to overcome some of the drawbacks of standard clustering algorithms (117).

Multiobjective optimization problems involving discrete variables tend to have feasible

solutions in disjoint regions of the decision space, and this is often translated into disjoint (and arbitrarily shaped) regions in the objective space (115). While most distance-based clustering algorithms can only find spheroidal clusters, density-based clustering algorithms use a more complex measure and do not have this restriction. The clustering algorithm DBSCAN has been developed by Ester et al. (118) to detect arbitrarily shaped regions in the objective space (see the Supporting Information for an illustrative example).

Heat maps and cp lines. Chichakly and Epstein (119) showed for the first time how feasible dominated solutions of a multiobjective optimization problem can be used for knowledge discovery. For problems involving only real variables, they introduce two different techniques: heat maps and *ceteris paribus* (cp) lines. On the one hand, they use heatmaps of dominated solutions to visualize how selected relevant variables vary with respect to the objectives throughout the feasible region. On the other hand, they introduce the so called cp lines to show how the objective values vary when a given design variable is modified with all the remaining variables held constant.

For problems involving integer or binary variables we suggest an analysis parallel to that of cp lines to assess how a given parameter affect the objective values. Starting from a given feasible solution (not necessarily in the Pareto front), we perturb incrementally the (real) parameter of interest (note that in this case is not necessarily a decision variable) while keeping the remaining parameters and decision variables fixed to obtain the corresponding objective values, which we visualize in the objective space. In this way we can show for example how decreasing or increasing a manipulable parameter gives a better or worse design (see the Supporting Information for an illustrative example).

Flexible pattern mining. Inspired on the sequential pattern mining used in market basket analysis, Bandaru et al. (115) recently proposed a method to find patterns in the solution data set (including non-dominated solutions) of multiobjective optimization problems. Basically, the method generates rules in the decision space that are translated afterwards into patterns in the objective space (more details can be found in (115)). Next we illustrate

with a simple example the kind of knowledge that can be gained within our design context by means of this and similar approaches.

In Fig. 7 we depict in the objective space the feasible points of an optimization problem in which we search for circuits with switch-like response upon induction with two design objectives (cost of protein production and switch-performance). In this problem, we start from a library containing 8 promoters and 4 transcripts. The number of allowed regulatory regions is 3. The decision vector is constituted by 32 binary variables, and the two design objectives are the switch performance (rated from 0 to 1) and the protein production cost. More details about the problem can be found in the Supplementary Information. In Fig. 7 (where black circles represent the feasible points, and the Pareto optimal ones are highlighted in green), we show two different patterns obtained with flexible pattern mining. The first pattern (points in yellow) shows the feasible circuits that fulfill the following rule: “The device containing $P_{tet_2}Lac_I$ is active, and one of the devices $P_{\lambda}Lac_I$ or $P_{\lambda}araC$ is active” (we remind here that when a device is present or active in the circuit, the corresponding binary decision variable is one, whereas if the device is not present the corresponding binary decision variable is zero). We observe that the circuits fulfilling this rule lie in the upper left quadrant of the objective space, i.e., they show a good performance (with a score higher than 0.8) at a high protein production cost.

The second pattern (points in magenta) shows the feasible circuits that fulfill the following rule: “The device containing $P_{tet_2}Lac_I$ is active. The device containing $P_{lac_1}tetR$ is active, whereas the remaining devices producing tetR are inactive. The devices $P_{\lambda}Lac_I$ and $P_{\lambda}araC$ are inactive”. We observe that all the circuits fulfilling this rule show a very good performance (with a score higher than 0.9) at medium-low protein production cost. In fact, some of these circuits lie in the Pareto front of solutions.

Figure 7: Feasible points of the multiobjective switch design example. The Pareto optimal solutions are highlighted in green. We show two different patterns generated by flexible pattern mining (together with two representative circuits of each pattern). The points highlighted in yellow and magenta correspond to the first and second patterns described in the text, respectively.

3.4 Computational efficiency and scalability of the method

In the proposed optimization process, we discretise the axis corresponding to one of the design objectives in m intervals, and solve m MINLP problems (one per discretisation interval). The overall computational time will be the product of the time spent per MINLP multiplied by the number of intervals. Therefore, the computational efficiency of the MO-MINLP strategy is determined by the computational efficiency of the selected MINLP solver.

Here, we illustrate the computational efficiency, robustness and scalability of eSS-MISQP in comparison with MITS-MISQP and ACO-MISQP using a paradigmatic MINLP design problem, consisting of finding a sustained oscillator from a library of standard parts. To this aim, we maximize the first peak of the normalized autocorrelation function (minimizing the negative of the first peak). Details on the problem can be found in the Supplementary Information.

First, we start from library with 4 promoters and 4 transcripts (denoted as library 1), leading to 16 binary decision variables, and we solve three different problems:

- i) Design of an oscillator with 3 active devices (or equivalently, 3 regulatory regions).
- ii) Design of an oscillator with 5 active devices (or equivalently, 5 regulatory regions).
- iii) Design of an oscillator with 9 active devices (or equivalently, 9 regulatory regions).

Each problem is solved using the three methods: eSS-MISQP, ACO-MISQP and MITS-MISQP (30 runs per instance starting from different random initial guesses). The three methods show a very good performance in this case (finding in all runs a solution in less than 25 seconds). The average CPU times spent by each solver are depicted in Fig 8 A (detailed CPU times for eSS-MISQP are included in the Supporting Information).

Now we repeat the analysis for an extended library with 4 promoters and 11 transcripts (denoted as library 2), leading to 44 binary decision variables. While eSS-MISQP shows again a very good performance (finding in all runs a solution in less than 600 seconds), MITS-MISQP and ACO-MISQP do not find a solution in reasonable time. The average CPU times obtained with eSS-MISQP are 462, 451 and 468 seconds for circuits with 3, 5 and 9 active devices, respectively (detailed CPU times are included in the Supporting Information). All the computations were carried out in Matlab 2015b using an Intel 2.8 GHz Xeon.

With this example we illustrate what we have experienced in an extensive range of problems in terms of computational performance, robustness and scalability of the methods. Our findings can be summarized as follows:

- all the solvers (eSS-MISQP, MITS-MISQP, ACO-MISQP) are rather efficient for problems with low-medium number of decision variables, but ACO-MISQP requires a larger computational effort.
- the performances of MITS-MISQP and ACO-MISQP show higher dependency on the nature of the problem and initial guesses, while that of eSS-MISQP is more robust across problems and initial guesses.

- the performances of the methods do not directly depend on the complexity of the obtained circuits, but on the number of decision variables (and the related encoding strategy). This can be seen in Fig 8, in which the CPU time does not increase significantly with the number of active pairs of the circuits (see 8 A), it increases with the number of decision variables, as we can deduce by comparing average CPU times for $N=16$ and $N=44$.
- eSS-MISQP is the only method that scales up well as we increase the problem dimension (i.e. as the number of decision variables increase).

In summary, our results clearly indicate that eSS-MISQP is more robust, outperforming the other metaheuristics for problems of realistic size.

Figure 8: Average CPU time in seconds needed for eSS-MISQP, MITS-MISQP and ACO-MISQP to find an oscillator among all possible combinations of components of Library 1 (16 decision variables), for target circuits with 3, 5 and 9 regulatory regions.

4 Author Information

4.1 ORCID

Irene Otero-Muras: 0000-0003-2895-997X

Julio R. Banga: 0000-0002-4245-0320

4.2 Author Contribution

JRB and IOM conceived the project and methodology. IOM carried out the computations and analysis of results. IOM and JRB wrote the manuscript.

Acknowledgement

The authors acknowledge funding from Spanish MINECO (and the European Regional Development Fund) project SYN BIOFACTORY (grant number DPI2014-55276-C5-2-R).

Supporting Information Available

- Supp_Info_ACS.pdf: pdf file with supporting contents.
- Matlab_code_ACS.zip: (zip file with Matlab code).

This material is available free of charge via the Internet at <http://pubs.acs.org/>.

References

1. Marchisio, M. A., and Stelling, J. (2009) Computational design tools for synthetic biology. *Curr Opin Biotechnol* 20(4), 479–485.

2. MacDonald, J. T., Barnes, C., Kitney, R. I., Freemont, P. S., and Stan, G. B. (2011) Computational design approaches and tools for synthetic biology. *Integr Biol* 3(2), 97–108.
3. Marchisio, M. A., and Stelling, J. (2011) Automatic design of digital synthetic gene circuits. *PLoS Comput Biol* 7(2), e1001083.
4. Barnes, C. P., Silk, D., Sheng, X., and Stumpf, M. P. H. (2011) Bayesian design of synthetic biological systems. *Proc Nat Acad Sci USA* 108(37), 15190–15195.
5. Medema, M. H., Van Raaphorst, R., Takano, E., and Breitling, R. (2012) Computational tools for the synthetic design of biochemical pathways. *Nat Rev Microbiol* 10(3), 191–202.
6. Rollié, S., Mangold, M., and Sundmacher, K. (2012) Designing biological systems: systems engineering meets synthetic biology. *Chem Eng Sci* 69(1), 1–29.
7. Lux, M. W., Bramlett, B. W., Ball, D. A., and Peccoud, J. (2012) Genetic design automation: engineering fantasy or scientific renewal? *Trends Biotechnol* 30(2), 120–126.
8. Rodrigo, G., and Jaramillo, A. (2012) AutoBioCAD: full biodesign automation of genetic circuits. *ACS synthetic biology* 2(5), 230–236.
9. Oberortner, E., and Densmore, D. (2015) Web-based software tool for constraint-based design specification of synthetic biological systems. *ACS Synth Biol* 4(6), 757–760.
10. Leon, M., Woods, M. L., Fedorec, A. J. H., and Barnes, C. P. (2016) A computational method for the investigation of multistable systems and its application to genetic switches. *BMC Syst Biol* 10, 130.

11. Bradley, R. W., Buck, M., and Wang, B. (2016) Recognizing and engineering digital-like logic gates and switches in gene regulatory networks. *Curr Opin in Microbiol* 33, 74–82.
12. Nielsen, A. A. K., Bryan, S. D., Shin, J., Vaidyanathan, P., Paralanov, V., Strychalsky, E. A., Ross, D., Densmore, D., and Voigt, C. A. (2016) Genetic circuit design automation. *Science* 352(6281), aac7341.
13. Purnick, P. E., and Weiss, R. (2009) The second wave of synthetic biology: from modules to systems. *Nat Rev Mol Cel Biol* 10(6), 410–422.
14. Shoval, O., Sheftel, H., Shinar, G., Hart, Y., Ramote, O., Mayo, A., Dekel, E., Kavanagh, K., and Alon, U. (2012) Evolutionary trade-offs, Pareto optimality, and the geometry of phenotype space. *Science* 336(6085), 1157–60.
15. Szekely, P., Sheftel, H., Mayo, A., and Alon, U. (2013) Evolutionary tradeoffs between economy and effectiveness in biological homeostasis systems. *PLoS Comput Biol* 9(6), e1003163.
16. Ceroni, F., Algar, R., Stan, G. B., and Ellis, T. (2015) Quantifying cellular capacity identifies gene expression designs with reduced burden. *Nat Methods* 12(5), 415–418.
17. Goñi-Moreno, A., Carcajona, M., Kim, J., Martínez-García, E., Amos, M., and de Lorenzo, V. (2016) An Implementation-Focused Bio/Algorithmic Workflow for Synthetic Biology. *ACS Synt Biol* 5(10), 1127–1135.
18. Francois, P., and Hakim, V. (2004) Design of genetic networks with specified functions by evolution in silico. *Proc Natl Acad Sci USA* 101(2), 580–585.
19. Rodrigo, G., Carrera, J., and Jaramillo, A. (2007) Genetdes: automatic design of transcriptional networks. *Bioinformatics* 23(14), 1857–1858.

20. Dasika, M. S., and Maranas, C. D. (2008) OptCircuit: An optimization based method for computational design of genetic circuits. *BMC Syst Biol* 2, 24.
21. Cao, H., Romero-Campero, F. J., Heeb, S., Cámara, M., and Krasnogor, N. (2010) Evolving cell models for systems and synthetic biology. *Syst Synth Biol* 4(1), 55–84.
22. Huynh, L., Tsoukalas, A., Köppe, M., and Tagkopoulos, I. (2013) SBROME: A scalable optimization and module matching framework for automated biosystem design. *ACS Synth Biol* 2(5), 263–273.
23. Huynh, L., and Tagkopoulos, I. (2015) Fast and accurate circuit design automation through hierarchical model switching. *ACS Synth Biol* 4(8), 890–897.
24. Bhaskar, V., Gupta, S. K., and Ray, A. K. (2000) Applications of multiobjective optimization in chemical engineering. *Rev Chem Eng* 16, 1–54.
25. Marler, R. T., and Arora, J. S. (2004) Survey of multi-objective optimization methods for engineering. *Struct Multidiscip Opt* 26(6), 369–395.
26. Fleming, P. J., Purshouse, R. C., and Lygoe, R. J. (2005) Many-objective optimization: An engineering design perspective. In: Coello Coello, C. A., Hernández Aguirre, A., Zitzler E. (eds) Evolutionary Multi-Criterion Optimization. EMO 2005. *Lect Notes Computer Sc* 3410, 14–32.
27. Azizi, O., Mahesri, A., Lee, B. C., Patel, S. J., and Horowitz, M. (2010) Energy-performance tradeoffs in processor architecture and circuit design: a marginal cost analysis. *ACM SIGARCH Computer Architecture News* 38(3), 26–36.
28. Donoso, Y., and Fabregat, R. (2016) *Multi-objective optimization in computer networks using metaheuristics*, CRC Press.
29. Pardalos, P. M., Migdalas, A., and Pitsoulis, L. (eds) (2008) *Pareto optimality, game theory and equilibria*; Springer Science & Business Media, Vol. 17.

30. Handl, J., Kell, D. B., and Knowles, J. (2007) Multiobjective optimization in bioinformatics and computational biology. *IEEE/ACM Trans Comput Biol Bioinform* 4(2), 279–292.
31. Schuster, P. (2012) Optimization of multiple criteria: Pareto efficiency and fast heuristics should be more popular than they are. *Complexity* 18(2), 5–7.
32. Goñi, J., Avena-Koenigsberger, A., de Mendizabal, N. V., van den Heuvel, M. P., Betzel, R. F., and Sporns, O. (2013) Exploring the morphospace of communication efficiency in complex networks. *PLoS ONE* 8(3), e58070.
33. Seoane, L. F., and Solé, R. V. (2015) Phase transitions in Pareto optimal complex networks. *Phys Rev E* 92, 032807.
34. Moses, M., Bezerra, G., Edwards, B., Brown, J., and Forrest, S. (2016) Energy and time determine scaling in biological and computer designs. *Phil Trans R Soc B* 371(1701), 20150446.
35. Seoane, L. F., and Solé, R. V. (2016) Multiobjective optimization and phase transitions. In: Battiston, S., De Pellegrini, F., Caldarelli, G., Merelli, E. (eds) Proceedings of ECCS 2014, 259–270 *Springer Proceedings in Complexity*.
36. Heinrich, R., and Schuster, S. *The regulation of cellular systems*; Springer Science & Business Media, 1996.
37. Majewski, R. A., and Domach, M. M. (1985) Consideration of the gain, enzymatic capacity utilization, and response time properties of metabolic networks as a function of operating point and structure. *Biosystems* 18(1), 15–22.
38. Liljenström, H., and Blomberg, C. (1987) Site dependent time optimization of protein synthesis with special regard to accuracy. *J Teor Biol* 129(1), 41–56.

39. Vera, J., De Atauri, P., Cascante, M., and Torres, N. V. (2003) Multicriteria optimization of biochemical systems by linear programming: application to production of ethanol by *Saccharomyces cerevisiae*. *Biotechnol Bioeng* 83(3), 335–343.
40. Sendin, O. H., Vera, J., Torres, N. V., and Banga, J. R. (2006) Model based optimization of biochemical systems using multiple objectives: a comparison of several solution strategies. *Math Comput Model Dyn Syst* 12, 469–487.
41. Halsall-Whitney, H., Taylor, D., and Thibault, J. (2003) Multicriteria optimization of gluconic acid production using net flow. *Bioprocess Biosyst Eng* 25(5), 299–307.
42. Sendin, J. O. H., Otero-Muras, I., Alonso, A. A., and Banga, J. R. (2006) Improved optimization methods for the multiobjective design of bioprocesses. *Ind Eng Chem Res* 45(25), 8594–8603.
43. Vo, T. D., Greenberg, H. J., and Palsson, B. O. (2004) Reconstruction and functional characterization of the human mitochondrial metabolic network based on proteomic and biochemical data. *J Biol Chem* 279(38), 39532–39540.
44. Sendin, J. O. H., Alonso, A. A., and Banga, J. R. (2009) Multi-objective optimization of biological networks for prediction of intracellular fluxes. *Advances in Soft Computing* 49, 197–205.
45. Oberhardt, M. A., Goldberg, J. B., Hogardt, M., and Papin, J. A. (2010) Metabolic network analysis of *Pseudomonas aeruginosa* during chronic cystic fibrosis lung infection. *J Bacteriol* 192(20), 5534–5548.
46. Schuetz, R., Zamboni, N., Zampieri, M., Heinemann, M., and Sauer, U. (2012) Multi-dimensional optimality of microbial metabolism. *Science* 336, 601–604.
47. Soleimani-Damaneh, M. (2011) An optimization modelling for string selection in molecular biology using Pareto optimality. *Appl Math Model* 35(8), 3887–3892.

48. Santana, R., Bielza, C., and Larrañaga, P. (2011) Optimizing brain networks topologies using multi-objective evolutionary computation. *Neuroinformatics* 9(1), 3–19.
49. Soleimani-Damaneh, M. (2011) On some multiobjective optimization problems arising in biology. *Int J Comput Math* 88(6), 1103–1119.
50. Taneda, A. (2015) Multi-objective optimization for RNA design with multiple target secondary structures. *BMC Bioinformatics* 16, 280.
51. El Samad, H., Khammash, M., Homescu, C., and Petzold, L. (2005) Optimal Performance of the Heat-Shock Gene Regulatory Network. *IFAC Proceedings Volumes* 38(1), 19–24.
52. Angione, C., and Lio, P. (2015) Predictive analytics of environmental adaptability in multi-omic network models. *Sci Rep* 5, 15147.
53. Constanza, J., Carapezza, G., Angione, C., Lio, P., and Nicosia, G. (2012) Robust design of microbial strains. *Bioinformatics* 28(23), 3097–3104.
54. Patane, A., Santoro, A., Constanza, J., Carapezza, and Nicosia, G. (2015) Pareto Optimal Design for Synthetic Biology. *IEEE Trans Biomed Circuits Syst* 9(4), 555–571.
55. Oh, Y. G., Lee, D. Y., Lee, S. Y., and Park, S. (2009) Multiobjective flux balancing using the NISE method for metabolic network analysis. *Biotechnol Prog* 25(4), 999–1008.
56. Higuera, C., Villaverde, A. F., Banga, J. R., Ross, J., and Morán, F. (2012) Multi-criteria optimization of regulation in metabolic networks. *PLoS ONE* 7(7), e41122.
57. Costanza, J., Carapezza, G., Angione, C., Lio, P., and Nicosia, G. (2012) Multi-objective optimisation, sensitivity and robustness analysis in FBA modelling. *Computational Methods in Systems Biology*, 127–147.

58. Angione, C., Carapezza, G., Costanza, J., Lio, P., and Nicosia, G. (2013) Pareto optimality in organelle energy metabolism analysis. *IEEE/ACM Trans Comput Biol Bioinform* 10(4), 1032–1044.
59. Carapezza, G., Umeton, R., Costanza, J., Angione, C., Stracquadanio, G., Papini, A., Lio, P., and Nicosia, G. (2013) Efficient behavior of photosynthetic organelles via Pareto optimality, identifiability, and sensitivity analysis. *ACS Synth Biol* 2(5), 274–288.
60. de Hijas-Liste, G. M., Klipp, E., Balsa-Canto, E., and Banga, J. R. (2014) Global dynamic optimization approach to predict activation in metabolic pathways. *BMC Syst Biol* 8, 1.
61. Boada, Y., Reynoso-Meza, G., Picó, J., and Vignoni, A. (2016) Multi-objective optimization framework to obtain model-based guidelines for tuning biological synthetic devices: an adaptive network case. *BMC Syst Biol* 10, 27.
62. Blakes, J., Raz, O., Feige, U., Bacardit, J., Widera, P., Ben-Yehezkel, T., Shapiro, E., and Krasnogor, N. (2014) Heuristic for maximizing DNA reuse in synthetic DNA library assembly. *ACS Synth Biol* 3(8), 529–542.
63. Warmflash, A., Francois, P., and Siggia, E. D. (2012) Pareto Evolution of Gene Networks: An Algorithm to Optimize Multiple Fitness Objectives. *Phys Biol* 9(5), 056001.
64. Otero-Muras, I., and Banga, J. R. (2014) Multicriteria global optimization for biocircuit design. *BMC Syst Biol* 8, 113.
65. Otero-Muras, I., Henriques, D., and Banga, J. R. (2016) SYNBADm: a tool for optimization-based automated design of synthetic gene circuits. *Bioinformatics* 32(21), 3360–3362.
66. Heinemann, M., and Panke, S. (2006) Synthetic biology—putting engineering into biology. *Bioinformatics* 22(22), 2790–2799.

67. Arpino, J. A. J., Hancock, E. J., Anderson, J., Barahona, M., Stan, G. B., Papatristodoulou, A., and Polizzi, K. (2013) Tuning the dials of synthetic biology. *Microbiology* 159, 1236–1253.
68. Sainz de Murieta, I., Bultelle, M., and Kitney, R. I. (2016) Toward the First Data Acquisition Standard in Synthetic Biology. *ACS Synt Biol* 5(8), 817–826.
69. Davidsohn, N., Beal, J., Kiani, S., Adler, A., Yaman, F., Li, Y., Xie, Z., and Weiss, R. (2015) Accurate Predictions of Genetic Circuit Behavior from Part Characterization and Modular Composition. *ACS Synt Biol* 4(6), 673–681.
70. Endy, D. (2005) Foundations for engineering biology. *Nature* 438, 449–453.
71. Pedersen, M., and Phillips, A. (2009) Towards programming languages for genetic engineering of living cells. *J R Soc Interface* 6, S437–S450.
72. Galdzicki, M. et al. (2014) The Synthetic Biology Open Language (SBOL) provides a community standard for communicating designs in synthetic biology. *Nat Biotechnol* 32, 545–550.
73. Roehner, N., and Myers, C. J. (2014) A methodology to annotate systems biology markup language models with the synthetic biology open language. *ACS Synt Biol* 3(2), 57–66.
74. Mısırlı, G., Hallinan, J., Pocock, M., Lord, P., McLaughlin, J. A., Sauro, H., and Wipat, A. (2016) Data Integration and Mining for Synthetic Biology Design. *ACS Synt Biol* 5(10), 1086–1097.
75. Zomorodi, A. R., and Maranas, C. D. (2014) Coarse-grained optimization-driven design and piecewise linear modeling of synthetic genetic circuits. *Eur J Oper Res* 237(2), 665–676.

76. Gillespie, D. T. (1977) Exact Stochastic Simulation of Coupled Chemical Reactions. *J Phys Chem* 81(25), 2340–2361.
77. d'Eysmond, T., de Simone, A., and Naef, F. (2013) Analysis of precision in chemical oscillators: implications for circadian clocks. *Phys Biol* 10(5), 056005.
78. Otero-Muras, I., and Banga, J. R. (2016) Design Principles of Biological Oscillators through Optimization: Forward and Reverse analysis. *PLoS ONE* 11(12), e0166867.
79. Süel, G. M., Kulkarni, R. P., Dworkin, J., Garcia-Ojalvo, J., and Elowitz, M. B. (2007) Tunability and noise dependence in differentiation dynamics. *Science* 315(5819), 1716–1719.
80. Tsai, T. Y, Choi, Y. S., Ma, W., Pomerening, J. R., Tang, C., and Ferrell, J. E. Jr. (2008) Robust, tunable biological oscillations from interlinked positive and negative feedback loops. *Science* 321(5885), 126–129.
81. Elowitz, M. B., and Leibler, S. (2000) A synthetic oscillatory network of transcriptional regulators. *Nature* 403, 335–338.
82. Guantes, R., and Poyatos, J. F. (2006) Dynamical principles of two-component genetic oscillators. *PLoS Comput Biol* 2(3), e30.
83. Elowitz, M., and Lim, W. A. (2010) Build life to understand it. *Nature* 468, 889–890.
84. Khalil, A. S., and Collins, J. J. (2010) Synthetic biology: applications come of age. *Nat Rev Genet* 11(5), 367–379.
85. Garcia-Ojalvo, J., Khalil, A. S., and McCarthy, J. (2016) Biological Insights from Synthetic Biology. *Integr Biol* 8, 380–382.
86. Munteanu, A., Cotterell, J., Solé, R. V., and Sharpe, J. (2014) Design principles of stripe-forming motifs: the role of positive feedback. *Sci Rep* 4, 5003.

87. Mjolsness, E., Sharp, D. H., and Reinitz, J. (1991) A connectionist model of development. *J Theor Biol* 152(4), 429–453.
88. Hancock, E. J., Stan, G. B., Arpino, J. A. J., and Papachristodoulou, A. (2015) Simplified mechanistic models of gene regulation for analysis and design. *J R Soc Interface* 12(108), 20150312.
89. Pájaro, M., Alonso, A. A., Otero-Muras, I., and Vázquez, C. (2017) Stochastic modeling and numerical simulation of gene regulatory networks with protein bursting. *J Theor Biol*, DOI: 10.1016/j.jtbi.2017.03.017.
90. Ashe, H. L., and Briscoe, J. (2006) The interpretation of morphogen gradients. *Development* 133(3), 385–394.
91. Entus, R., Aufderheide, B., and Sauro, H. M. (2007) Design and implementation of three incoherent feed-forward motif based biological concentration sensors. *Syst Synth Biol* 1(3), 119–128.
92. Zaslaver, A., Mayo, A. E., Rosenberg, R., Bashkin, P., Sberro, H., Tsalyuk, M., Surette, M. G., and Alon, U. (2004) Just-in-time transcription program in metabolic pathways. *Nat Genet* 36(5), 486–491.
93. Kafri, M., Metzl-Raz, E., Jona, G., and Barkai, N. (2016) The cost of protein production. *Cell Rep* 14(1), 22–31.
94. 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.
95. Adler, M., Szekely, P., Mayo, A., and Alon, U. (2017) Optimal Regulatory Circuit Topologies for Fold-Change Detection. *Cell Syst* 4(2), 171–181.
96. Miettinen, K. (2012) *Nonlinear multiobjective optimization*; Springer Science & Business Media..

97. Das, I. (1999) On characterizing the knee of the Pareto curve based on Normal-Boundary Intersection. *Structural Optimization* 18(2), 107–115.
98. Boukouvala, F., Misener, R., and Floudas, C. A. (2016) Global optimization advances in mixed-integer nonlinear programming, MINLP, and constrained derivative-free optimization, CDFO. *Eur J Oper Res* 252(3), 701–727.
99. Chachuat, B., Singer, A. B., and Barton, P. I. (2006) Global methods for dynamic optimization and mixed-integer dynamic optimization. *Ind Eng Chem Res* 45(25), 8373–8392.
100. Guillén-Gosálbez, G., Miró, A., Alves, R., Sorribas, A., and Jiménez, L. (2013) Identification of regulatory structure and kinetic parameters of biochemical networks via mixed-integer dynamic optimization. *BMC Syst Biol* 7, 113.
101. Sager, S., Claeys, M., and Messine, F. (2015) Efficient upper and lower bounds for global mixed-integer optimal control. *J Global Optim* 61(4), 721–743.
102. Banga, J. R., Balsa-Canto, E., Moles, C. G., and Alonso, A. A. (2005) Dynamic optimization of bioprocesses: Efficient and robust numerical strategies. *J Biotechnol* 117(4), 407 – 419.
103. Balsa-Canto, E., Peifer, M., Banga, J. R., Timmer, J., and Fleck, C. (2008) Hybrid optimization method with general switching strategy for parameter estimation. *BMC Syst Biol* 2, 26.
104. Exler, O., Antelo, L. T., Egea, J. A., Alonso, A. A., and Banga, J. R. (2008) A tabu search-based algorithm for mixed-integer nonlinear problems and its application to integrated process and control system design. *Comput Chem Eng* 32(8), 1877–1891.
105. Schlueter, M., Egea, J. A., and Banga, J. R. (2009) Extended ant colony optimization

- for non-convex mixed integer nonlinear programming. *Comput Oper Res* 36(7), 2217–2229.
106. Sendin, J. O. H., Exler, O., and Banga, J. R. (2010) Multi-objective mixed integer strategy for the optimisation of biological networks. *IET Syst Biol* 4(3), 236–248.
107. Exler, O., and Schittkowski, K. (2007) A trust region SQP algorithm for mixed integer nonlinear programming. *Optim Lett* 1(3), 269–280.
108. Egea, J. A., Marti, R., and Banga, J. R. (2010) An evolutionary method for complex-process optimization. *Comput Oper Res* 37(2), 315–324.
109. Abramson, M. A., Audet, C., Chrissis, J. W., and Walston, J. G. (2009) Mesh adaptive direct search algorithms for mixed variable optimization. *Optim Lett* 3, 35.
110. Bentley, P. J., and Wakefield, J. P. (1997) Finding Acceptable Pareto-Optimal Solutions using multiobjective genetic algorithms. *Soft Computing in Engineering Design and Manufacturing* 231–240.
111. Thiele, L., Miettinen, K., Korhonen, P. J., and Molina, J. (2009) A Preference-Based Evolutionary Algorithm for Multi-Objective Optimization. *Evol Comput* 17(3), 411–436.
112. Kim, I. Y., and de Weck, O. L. (2005) Adaptive weighted-sum method for multi-objective optimization: a new method for Pareto front generation. *Struct Multidiscip Opt*, 29(2), 149–158..
113. Bandaru, S., Ng, A. H. C., and Deb, K. (2017) Data mining methods for knowledge discovery in multi-objective optimization: Part A-Survey. *Expert Syst Appl* 70, 139–159.
114. Deb, K., Bandaru, S., Greiner, D., Gaspar-Cunha, A., and Tutum, C. C. (2014) An integrated approach to automated innovization for discovering useful design principles: Case studies from engineering. *Applied Soft Computing* 15, 42–56.

- 1
2
3
4 115. Bandaru, S., Ng, A. H. C., and Deb, K. (2017) Data mining methods for knowledge
5 discovery in multi-objective optimization: Part B-New developments and applications.
6 *Expert Syst Appl* 70, 119–138.
7
8
9
10 116. Quinlan, J. (1993) *C4.5: Programs for machine learning*; Morgan Kaufmann, .
11
12
13 117. Roth, V., Laub, J., Buhman, J., and Muller, K. (2003) Going metric: denoising pairwise
14 data. *Adv Neural Inf Process Syst* 15, 817–824.
15
16
17
18 118. Ester, M., Kriegel, H. P., Sander, J., and Xu, X. (1996) A density-based algorithm for
19 discovering clusters in large spatial datasets with noise. *Proceedings of International*
20 *Conference on Knowledge Discovery and Data Mining* 226–231.
21
22
23
24
25 119. Chichackly, K., and Eppstein, K. (2013) Discovering design principles from dominated
26 solutions. *IEEE Access* 1, 275–289.
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

MULTIOBJECTIVE AUTOMATED DESIGN FRAMEWORK

Global Multiobjective
Mixed Integer Optimization

(optimize topology and parameters
simultaneously)

Topologies that optimally *trade-off* objectives 1 and 2:

FORWARD DESIGN

Library of Parts

Best circuit for implementation

Example: Synthetic oscillator with optimal
stability and tunability of the period.

REVERSE DESIGN

All possible N-gene networks

Patterns/motifs with given function

Example: Fold-Change detection architectures
with optimal speed and amplitude