Evolutionary continuous optimization of hybrid Gene Regulatory Networks

Romain Michelucci[0000−0001−6107−4394], Jean-Paul Comet[0000−0002−6681−3501], and Denis Pallez[0000−0001−5358−8037]

Université Côte d'Azur, CNRS, I3S, Sophia Antipolis, France
firstname.name@univ-cotedazur.fr

Abstract. The study of gene regulatory networks (GRNs) allows us to better understand biological systems such as the adaptation of the organism to a disturbance in the environment. Hybrid GRNs (hGRNs) are of interest because they integrate the continuous time evolution in GRN modeling which is convenient in biology. This study focuses on the problem of identifying the variables of hGRN models. In a large-scale case, previous work using constraint-based programming has failed to solve the minimal constraints on such variables which reflect the biological knowledge on the system behavior. In this work, we propose to transform a Constraint Satisfaction Problem (CSP) into a Free Optimization Problem (FOP) by formulating an adequate fitness function and validate the approach on an abstract model of the circadian cycle. We compare several continuous optimization algorithms and show that these first experimental results are in agreement with the specifications coming from biological expertise: evolutionary algorithms are able to identify a solution equivalent to the ones found by continuous constraint solvers.

Keywords: Continuous single-objective optimization · Fitness formulation · hybrid GRN · Real-world application · Bio-inspired computation.

1 Introduction

Genetic regulatory network (GRN) modeling aims at studying and understanding the molecular mechanisms that enable the organism to perform essential functions ranging from metabolism to environmental disturbance adaptation. Two types of control rules coexist in these regulatory networks: activations and inhibitions. Their combination allows the system to behave in a large variety of ways and the complexity of these systems comes from the so-called positive and negative feedbacks commonly observed, which respectively lead to multistationarity and homeostasis (ability to maintain a balance). Studying the dynamics of these systems opens new perspectives with crucial applications in fundamental biology, pharmacology, medicine, or chronotherapy for instance, which tries to choose the best time of day to administer the medication in order to limit the side effects while preserving the therapeutic effects.

Numerous modeling frameworks have been proposed for representing GRNs such as differential frameworks (using ordinary differential equations), stochastic ones (considering that transitions between states have a stochastic nature), or discrete ones (modeling the presence or absence of biological entities in the system states). Even if each of them presents their own advantages, they all rely on the identification of the variables that govern the model dynamics and this variable identification remains the limiting step. To address this difficulty, a considerable number of research groups apply evolutionary algorithms to fit GRN models and variables to gene expression data, see e.g. the survey [16].

In the present work, we prefer to consider *hybrid* frameworks [1], called hGRNs, which add to discrete ones [17] the time spent in each of the discrete states. Once more, the variables' identification remains the bottleneck of the modeling process, but one can seek in such a hybrid framework for an automation of this step to build a model in agreement with the experimental observations. Indeed, modeling variations of protein concentration in a biological system can be very hard for numerous proteins. Nevertheless, experimental observations allow us to represent experimental traces by irregularly spaced time series of observable events. From those events, minimal constraints on the hGRN variables can be deduced and the authors attempted to use continuous Constraint Satisfaction Problem (CSP) solvers [2] but faced difficulties in extracting solutions.

In this paper, we show that the constraint problem, which characterizes the set of solutions exhaustively, can be expressed as a FOP [6, 8] by indirectly handling constraints. More precisely, the representation of biological knowledge as a sequence of observable events allows to define a high-dimensional non-trivial continuous optimization problem in which the search space increases exponentially with the number of genes involved in the hGRN.

The work focuses on the FOP formulation, on the fitness characterization and performs some comparisons between several bio-inspired algorithms, leaving out the scalability problem which is out of the scope of the article. We illustrate the approach on a very abstract model of the circadian cycle (subsystem allowing an adaptation of the body to day/night alternation).

The paper is organized as follows: section 2 describes the models used for representing the dynamics of biological systems and the biological knowledge used as an input. Section 3 proposes a method whereby the modeling problem is reformulated as a continuous FOP that can be solved by means of a bio-inspired algorithm. Experimental results are discussed in section 4 and some conclusions are drawn in section 5.

2 Problem description

2.1 Hybrid GRN

To build a digital model of a biological system, it is necessary to know precisely how it works. Such a system is defined as a set of genes performing a biological function and represented in the form of a GRN where vertices V correspond

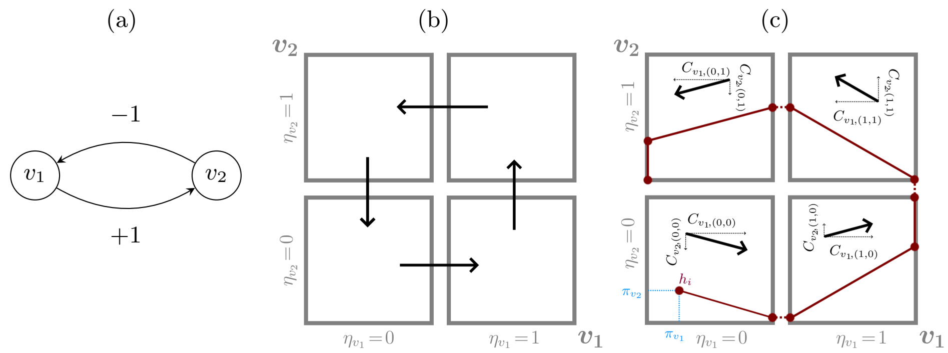

Fig. 1: Interaction graph representing the circadian cycle (a), its discrete state graph (b) and a possible dynamic of its hybrid state graph (c).

to an abstraction of one or more biological genes (within circles) and edges depicting activations (+) or inhibitions (−). It can be statically represented as a labeled directed graph or *interaction graph* (cf. fig. 1a). For studying the GRN evolution, we first need to define the system state as the concentrations vector of the proteins related to genes. Because the regulations take place above particular thresholds, we associate with the sign of the regulation an abstract threshold: $v_1 \xrightarrow{+n} v_2$ (resp. $v_1 \xrightarrow{-n} v_2$) means that v_1 can activate (resp. inhibit) v_2 only if the concentration of v_1 is above its n^{th} threshold (ranked by increasing order). For example, graph of fig. 1a forms a negative feedback loop where each gene ($v_1, v_2 \in V$) has an indirect negative action on itself: when v_1 is active, it is above its first threshold (we note $v_1 = 1$), then, v_1 activates the gene v_2 and v_2 passes from level 0 (under its first threshold) to another level greater than its first threshold ($v_2 = 1$). As v_2 reaches level 1, v_2 inhibits v_1, and so on. This represents a highly abstracted model of regulations piloting the circadian cycle ensuring the cyclic adaptation (day or night) of the organism.

In order to integrate dynamics in the previous model, the first step is to enumerate all possible states: a *discrete state* is defined by the level of all genes contained in the GRN. Thus, if there are n genes, each state η is defined by a vector of n integers $(\eta_{v_1}, ..., \eta_{v_n})$ and $\mathbb{S}$ denotes the set of all possible discrete states of the GRN. For instance, state $(0, 0)$, the bottom left gray square box in fig. 1b, corresponds to the state where discrete levels of genes v_1 and v_2 are both equal to 0. The second step consists in adding transitions between all these states (black arrows). Thus, *state graph* of fig. 1b represents the dynamics associated with the interaction graph of fig. 1a. Such kind of models is very interesting for logically reasoning on regulatory changes. Nevertheless, this qualitative modeling framework totally abstracts time information whereas , for numerous biological systems, time plays a crucial role in the system's fate.

In addition to discrete transitions (dotted red lines in fig. 1c), an hGRN adds continuous evolution of gene product concentration in each discrete state represented by a continuous trajectory (linear, see straight red lines). One point on this trajectory inside a particular discrete state, is given by a precise position inside the square: $\pi = (\pi_{v_1}, ..., \pi_{v_n}) \in [0,1]^n$. Thus, a *hybrid state* h is defined by a discrete state η and its *fractional part* π. For instance, the coordinates of the initial hybrid state h_i are $\left((\eta_{v_1}, \eta_{v_2})^t, (\pi_{v_1}, \pi_{v_2})^t\right) = \left((0,0)^t, (0.25, 0.25)^t\right)$.

Starting from h_i, the hGRN dynamics is given by following the evolution direction of the discrete state $(\eta_{v_1}, \eta_{v_2}) = (0,0)$. This direction is defined by a so-called *celerity vector*. Thus, the celerity of v_1 in $(0,0)$ is denoted $C_{v_1,(0,0)}$ in order to specify that this celerity is associated with v_1, when v_1 and v_2 levels are 0. In a similar way, the celerity of v_2 in $(0,0)$ is denoted $C_{v_2,(0,0)}$. More generally, an hGRN is defined by both a GRN and celerity vectors $C = \{C_{v,\eta}\}$, a family of floated values indexed by (v,η) where $v \in V$ and $\eta \in \mathbb{S}$. $C_{v,\eta}$ is called the *celerity* of v in η. The hybrid state graph of fig. 1c depicts one possible dynamic associated with the interaction graph of fig. 1a. Starting from the initial hybrid state h_i, v_1 concentration increases until it reaches the right border of discrete state $(0,0)$. From this border, the trajectory jumps into the neighbor state $(1,0)$ because the celerity vector of this second state does not oppose the entry of the trajectory (signs of v_1 celerities in both states are the same). In $(1,0)$, the trajectory reaches the right border of this discrete state which corresponds to the maximum admissible concentration of v_1. As there is no discrete state at the right of $(1,0)$, the trajectory evolves on this border in v_2 direction resulting in a so-called *slide* of v_1, noted $slide^+(v_1)$. After sliding, the trajectory enters the state $(1,1)$. This process follows up until the trajectory enters back the initial state $(0,0)$. The complete definition of hGRN dynamics can be found in [1].

Such modeling frameworks are very useful to reason on the GRN trajectories. Nevertheless, as usual, the bottleneck of the modeling process relies on the determination of variable values controlling the trajectories, that is the celerities. The goal of this paper is to automatically determine, from some formalized biological information, all celerity vector values in order to obtain a valid hGRN model of the biological system studied. In the next part, we introduce the *biological knowledge* (BK) from which celerity values can be determined.

2.2 Biological knowledge

As opposed to numerous works that attempt to automatically build a model from raw experimental data [3, 5, 12, 15], the present work takes into consideration already-formalized information analyzed by biologists themselves coming from both biological data and expertise. This complementary approach is preferred because raw data are subject to noisiness and scarcity. A biological experiment consists of (i) putting the biological system in a particular initial state partially defined, (ii) recording the sequence of observable events and (iii) measuring the reached final state of the observed system. While initial and final states are described using their discrete and fractional parts $h_i = (\eta_i, \pi_i)$ and $h_f = (\eta_f, \pi_f)$, a sequence of observable events is formalized by a sequence of triplets of the form $(\Delta t, b, e)$. Each element of each triplet expresses a property on the behavior in

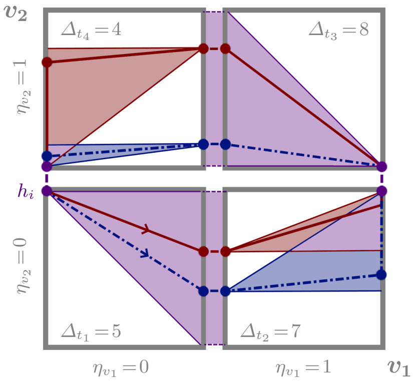

Fig. 2: Visual representation of the infinite set of possible solutions.

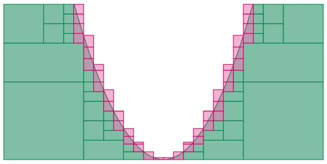

Fig. 3: CSP difficulty to target solutions for constraint $y \leq x^2$.

the current discrete state: Δt delineates the time spent in the current state; b specifies the observed behaviors during the continuous trajectory expressed by $slide(v)$ and $noslide(v)$; finally, e represents the next discrete state transition which is of the form $v+$ (resp. $v-$) specifying that the next discrete event is the increasing (resp. decreasing) of the discrete level of v.

For the interaction graph of fig. 1a, biological expertise can be summarized as follows: there exists a behavior starting from a particular point of coordinates going through four discrete states and coming back to the initial point after 24 hours. More precisely, the time spent in each of the 4 discrete states is approximately 5 hours in $(0,0)$, 7 in $(1,0)$ and so on. See the first properties of each event in the following description of the biological knowledge:

$$\{h_i\}\begin{pmatrix}5.0\\ noslide\,(v_2)\\ v_1+\end{pmatrix};\begin{pmatrix}7.0\\ slide^{+}\,(v_1)\\ v_2+\end{pmatrix};\begin{pmatrix}8.0\\ noslide\,(v_2)\\ v_1-\end{pmatrix};\begin{pmatrix}4.0\\ slide^{-}\,(v_1)\\ v_2-\end{pmatrix}\{h_f\} \quad (1)$$

where $h_i = \left((0,0)^t, (0.0, 1.0)^t\right)$ is the initial hybrid state and h_f (final hybrid state) is equal to h_i. For the first event, v_1+ constrains the trajectory to reach the next discrete state by increasing the concentration level of v_1. The second property $noslide(v_2)$ in $(0,0)$ expresses that the trajectory has to reach the right border of the discrete state without touching the upper or lower borders as explained in section 2.1. The continuous trajectory of fig. 1c satisfies all properties of eq. (1) except for the initial point h_i which is misplaced: it should be located in the top left corner of discrete state $(0,0)$ to allow trajectory to be a cycle.

Figure 2 represents for each discrete state, and one after another all possible trajectories satisfying eq. (1) using colored surfaces. Starting from h_i, the purple surface represents all compatible celerity vectors of $(0,0)$ which lead the trajectory to the next expected state without sliding at the bottom or top border. For illustrative purposes, two instances of compatible trajectories are highlighted in red and blue in the figure.

2.3 Constraint Satisfaction Problem (CSP) approach

Our goal is to identify celerity vectors that define trajectories (cf. section 2.1) satisfying constraints given by the *biological knowledge* BK. An earlier attempt has been developed using constraint-based programming [2]. This CSP formulation led to constraints on celerity vectors which had to be satisfied for the hGRN dynamics to be consistent with BK. However, the exploitation of the constraints generated was not so easy: classical solvers were not able to extract particular solutions. Let us consider a CSP that aims to find all solutions satisfying the constraint $y \leq x^2$. A continuous solver paves the search space in multiple tiles (colored rectangles in fig. 3). Green tiles only contain solutions of the CSP whereas red tiles may contain values that do not satisfy the constraint (i.e. $y > x^2$ above the curve).

The problem that arises from using a continuous solver may be summed up by its inability to extract particular solutions on the function curve. It would be necessary to obtain a tiling of infinitesimal size. That is why we decided to reformulate the hGRN variables' identification as an optimization problem.

2.4 Problem characterization

Finding celerity values that satisfy BK constraints consists of finding a continuous trajectory that (i) goes through the right sequence of discrete states, (ii) spends the right elapsed time in each encountered state, and (iii) satisfies the right behavior in each state by sliding or not. In the case of a trajectory that does not satisfy BK, we measure how much it does not respect this knowledge. For instance, as BK specifies spending 5 hours in $(0, 0)$, a trajectory spending 5 hours and 10 minutes is "better" than a trajectory that only spends 2 hours in the same discrete state. In other words, we use the notion of *distance* between a trajectory and the expected properties expressed by BK: this distance vanishes as soon as all properties of BK are satisfied. Since BK specifies the properties of a sequence of states, we can decompose such distance by computing how a considered trajectory tr inside each state η is far from BK properties of the corresponding state. Thus, the global distance of one property p is defined by summing such distances $d_{p,\eta}$ inside each encountered discrete state $\eta \in \mathbb{S}$ where p is one of the three BK properties Δt, b, or e. Therefore, we define three criteria:

Time criterion. The first criterion $d_{\Delta t}$ is related to the time spent in the current discrete state. It is the Euclidean distance between the expected time t^*_η of BK and the time t_η necessary for the current trajectory to reach the exit point from the current state:

$$d_{\Delta t}(tr, BK) = \sum\nolimits_{\eta} d_{Euclidean}\left(t_\eta, t^*_\eta\right) \tag{2}$$

Slide criterion. Second criterion evaluates the distance between the continuous trajectory behaviors inside each encountered discrete state and the properties of sliding in BK (denoted b in each observable event). Three different cases

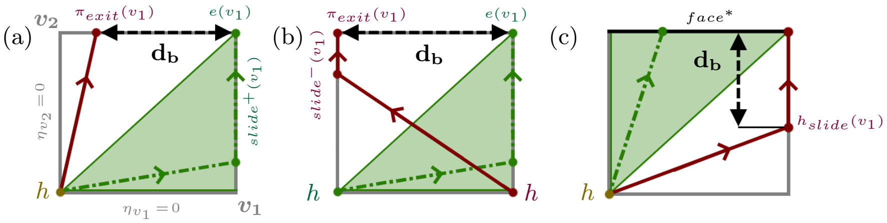

Fig. 4: Illustration of evaluation cases with respect to BK behavior property.

are considered and respectively illustrated in fig. 4 where green color represents BK and black dotted lines with double arrows, the distance d_b: (i) "v should slide according to BK, but the trajectory tr does not". In this case (fig. 4a), we compute the difference between the fractional part of the exit point of v according to tr ($\pi_{exit}(v)$) and $e(v)$ which is the fractional part of the exit point according to the sliding BK property (it either equals to 0 when $slide^-(v)$ or 1 when $slide^+(v)$):

$$d_{b,\eta}(tr, BK) = |\pi_{exit}(v) - e(v)| \tag{3}$$

where v is the gene concerned by the sliding property of the current discrete state η; (ii) "v should slide on max (resp. min) level according to BK, but the given trajectory slides on min (resp. max)". We consider it (see fig. 4b) as a special case of previous item (eq. (3)) where the exit point of the trajectory $\pi_{exit}(v)$ is either equal to 0 (sliding right in fig. 4b) or 1 (sliding left in fig. 4b); (iii) "v should not slide according to BK, but tr does" (fig. 4c). Here we compute the Manhattan distance between the first hybrid state where v begins to slide $h_{slide}(v)$ and the expected exit face noted $face^*$:

$$d_{b,\eta}(tr, BK) = d_{Manhattan}\left(h_{slide}(v), face^*\right) \tag{4}$$

In fig. 4c, the expected exit face is the north one (black line). As for the previous criterion, $d_b(tr, BK)$ is defined as the sum of the different $d_{b,\eta}(tr, BK)$ for each encountered discrete state η.

Discrete criterium. Intuitively, we have to compare the expected next discrete state (according to BK) with the discrete one into which the given trajectory tr enters. Unfortunately, in some cases, it is not possible to compute tr next discrete state because the trajectory can be *blocked* in the current discrete state. Let us take as an example the situation where the celerity vector inside the $(0, 0)$ discrete state points towards the south-west direction (cf. fig. 5a). The trajectory is blocked because the concentration of both gene products vanishes and there are no neighbors in these directions. In order to accurately evaluate tr, following the sequence of discrete states of BK, we evaluate the local distance between the considered trajectory inside the current discrete state and the associated BK. If the given trajectory does not allow the right discrete transition, then we artificially restart the trajectory in the next expected discrete state of BK.

The initial restart point $h_{restart}$ is defined by the new discrete state combined with the same fractional part before it stopped. This step is illustrated by the

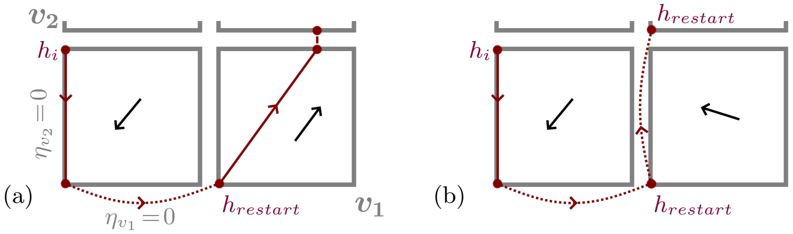

Fig. 5: Illustration of blockage (a) and wrong discrete transition (b).

curved dotted lines (cf. fig. 5). Therefore, d_e has to take into account on the one hand the Manhattan distance between the expected next discrete state η^{+^*} and the next discrete state η^+ according to tr, and on the other hand, the number of detected blockages:

$$d_e(tr, BK) = \sum\nolimits_\eta \left(d_{Manhattan}\left(\eta^+, \eta^{+^*}\right) + \mathbb{1}_{blockage}(\eta) \right) \tag{5}$$

where $blockage(\eta)$ is $True$ if the trajectory is blocked in the current discrete state η, and η^+ (resp. η^{+^*}) is the next discrete state according to tr (resp. to BK). Note that when a blockage occurs η^+ is not defined and in such a case $d_{Manhattan}\left(\eta^+, \eta^{+^*}\right)$ is considered zero (the penalty comes from $\mathbb{1}_{blockage}(\eta)$).

Aggregating criteria. We are focusing on formulating an adequate fitness function by indirectly handling constraints. Constraints are embedded into the three previously described optimization criteria such that all we need to care about is optimizing them. Thus, identifying celerity values consists in minimizing these criteria. One could consider this problem as a multi-criteria optimization problem. However, they are neither conflicting nor invariant: solutions exist that simultaneously optimize each criterion. Therefore, we suggest to combine them into a global distance $g(tr, BK)$ which consists in a combination of $d_{\Delta t}(tr, BK)$, $d_b(tr, BK)$ and $d_e(tr, BK)$ where the criteria weights are equal. Minimizing g leads to a single-objective optimization problem and will be addressed using bio-inspired algorithms. We propose two versions of the aggregation of three different criteria: an additive version defined by $g_+ = d_{\Delta_t} + d_b + d_e$ and a multiplicative version defined by $g_\times = (1 + d_{\Delta_t}) \times (1 + d_b) \times (1 + d_e) - 1$. Although the former is commonly used, the latter is proposed because, intuitively, it could have a greater impact on the convergence rate: errors are amplified and improvements are better controlled thanks to a steeper gradient. As each distance should be as close to 0 as possible, g_+ (resp. $g_\times$) should also be as close to 0 as possible (resp. thanks to the subtraction of 1). That leads to the definition of two fitness functions (knowing that BK is fixed):

$$f_+(x) = g_+(tr, BK) \quad (6) \qquad\qquad f_\times(x) = g_\times(tr, BK) \quad (7)$$

whose domain is $\left(\prod_{v\in V}[0, b_v]\right) \times [0,1]^n \times \mathbb{R}^{|C|}$ and codomain $\mathbb{R}^+$.

3 Bio-inspired hGRN modeling search

This section presents different bio-inspired approaches for identifying celerities of an hGRN. For this purpose, we compare several continuous single-objective bio-inspired algorithms for searching trajectories that satisfy biological knowledge BK as explained in section 2.4.

Representation. As presented in section 2.2, a trajectory is characterized by all celerities of all discrete states $\{C_{v,\eta}\}$ plus the initial hybrid state h_i. Thus, trajectory genotype of fig. 1c is defined by a tuple of 2 integers and 2 float values for h_i and 8 float values for celerities: the genotype is represented by $x = (h_i; C_{v_1,(0,0)}; C_{v_2,(0,0)}; C_{v_1,(1,0)};\ C_{v_2,(1,0)}; C_{v_1,(1,1)}; C_{v_2,(1,1)}; C_{v_1,(1,1)}; C_{v_2,(1,1)})$. Each floated value varies in the interval $[-r; r]$ with r equals 2 by default. In the presented example, the problem of identifying variables of an hGRN may seem trivial, nevertheless, in realistic models, the size of the genome is exponential with respect to the number n of genes: the initial hybrid state $h_i = (\eta_i, \pi_i)$ is described by n integer values for the discrete state and n float values for the fractional part. Because the number of celerities is also equal to n in each state and because the number of states is $|\mathbb{S}| = \prod_{v \in V}(b_v + 1)$, the total number of celerities $|C|$ is at most $n \times |\mathbb{S}| = n \times \prod_{v \in V}(b_v + 1)$ (possibly less in case of equality of a priori different celerities).

Fitness evaluation. Evaluating a candidate solution consists in computing the difference between BK formalized in 2.2 and the given trajectory obtained from celerities contained in the genome. To do so, we simulate the trajectory thanks to the initial state h_i and evaluate, discrete state by discrete state, each of the three introduced criteria $d_{\Delta t}$, d_b, and d_e.

Continuous optimization methods. A baseline random optimization (RO) [11] and the following four continuous meta-heuristic algorithms are compared:
(i) Differential Evolution (DE) [14], a global search heuristic using a binomial crossover and a mutation operator of $DE/rand/1/bin$. The different control parameters are $P_{CR} = 0.3$ and F is selected from the interval $[0.5, 1.0]$ randomly for each difference vector with the dither technique.
(ii) a simple $(\mu + \lambda)$ Genetic Algorithm (GA), used with a binary tournament selection and the following operators: Simulated Binary Crossover and Polynomial Mutation are applied with Fitness Survival. All duplicates are removed.
(iii) Adaptive Particle Swarm Optimization (APSO) [18] which is based on the simulating of social behavior. The algorithm uses a swarm of particles to guide its search. Each particle has a velocity and is influenced by locally and globally best-found solutions. The default parameters are $w = 0.9, c_1 = 2.0, c_2 = 2.0$ with a $max\_velocity\_rate = 0.2$.
(iv) Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) [9], which is a state-of-the-art and self-adaptive EA with the default initial standard deviation in each coordinate $\sigma = 0.1$.

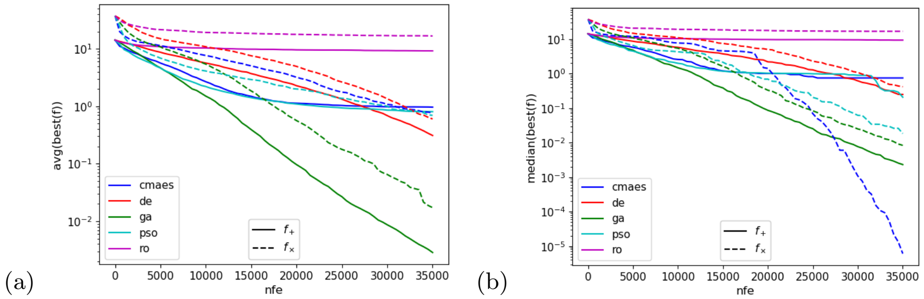

Fig. 6: Comparison of monotonic evolution of (a) mean and (b) median best fitness values by algorithm and fitness function on 100 runs. The y-axis is log scaled.

4 Experimental Study

The four meta-heuristics are implemented in pymoo [4]. To evaluate the algorithms' performance, we execute 100 independent runs for each algorithm and each fitness function. An initial population size of 500 is applied, followed by 35000 function evaluations (NFE). Both experiments are realized on the hGRN of fig. 1c using BK described by eq. (1) with h_i fixed to $((0,0)^t, (0.0, 1.0)^t)$.

Results. For each algorithm and each fitness function, at each generation we compute the best candidate solution so far, repeat 100 times the executions and compute the mean (resp. median) over the 100 runs. Monotonic evolutions of all algorithms are depicted in fig. 6 where straight lines represent f_+ and dotted lines, $f_\times$. It can be observed that (i) as expected, meta-heuristics results are (far) better performing compared to RO algorithm, (ii) decreases of f_+ and $f_\times$ values are done at the same pace (the curves are roughly parallel), except for CMA-ES whose $f_\times$ median evolution has a better convergence rate than with f_+, and (iv) apart from this case, GA convergence of the fitness function is one of the best (with both f_+ and $f_\times$) when focusing on the mean (resp. median).

Table 7a summaries statistics of the results obtained after 100 runs of the five considered algorithms. The best result (column by column) for f_+ (resp. $f_\times$) is bolded. Minimum, average and standard deviation are reported along with the Biological Success Rate (BSR) defined by the number of times an algorithm finds a solution with a fitness close to 0 with a precision error ϵ equal to 10^{-2}. BSR is based on the traditional success rate but introduces an important precision error coherent with biological expertise. For instance, a trajectory which would slide in $\eta = (0,0)$ during a fraction of seconds ($< \epsilon$) very next to the exit point $e(v_1) = 1.0$ before going to the next discrete state is an acceptable trajectory despite BK stating $noslide(v)$. In addition, Cumulative Distribution Function (CDF) curves are constructed in fig. 7b for f_+ (top) and $f_\times$ (bottom). Each CDF curve describes the probability that a solution is found at, or below, a given fitness score. For instance, in $f_\times$ experiment, there is almost 60% probability

(a)

A	FE	avg	stdev	min	BSR
CMA-ES	f_+	0.9644	1.18	**3e-9**	0.41
	$f_\times$	0.7661	2.51	**4e-10**	0.86
DE	f_+	0.3102	0.23	0.0171	0.13
	$f_\times$	0.6004	0.77	0.0373	0.04
GA	f_+	**0.0029**	**2e-3**	6e-4	**1.**
	$f_\times$	**0.0172**	**0.05**	0.0016	**0.98**
PSO	f_+	0.8053	0.98	4e-4	0.48
	$f_\times$	0.6938	1.71	2e-4	0.68
RO	f_+	9.1934	1.11	5.1679	0.
	$f_\times$	16.6763	2.5	7.9144	0.

(b)

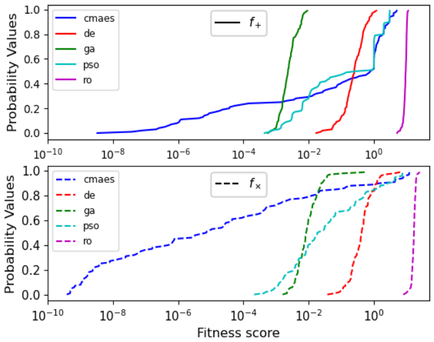

Fig. 7: Summary (a) and CDF curves (b) of overall best results.

that a user obtains a solution at a fitness score less than or equal to 10^{-4} with CMA-ES (given 35000 NFE). From both diagrams, two algorithms stand out: GA has the highest probability to obtain good results and there is a non-zero probability for CMA-ES to perform top results ($< 10^{-8}$).

To statistically validate the observed differences among the algorithms, we conducted a statistical validation campaign on the reported performance values of the two following scenarios: (1) algorithms performances obtained with f_+ objective function and (2) algorithms performances achieved with $f_\times$ one. In addition, a third scenario is suggested as being a comparison of algorithms performances between f_+ and $f_\times$. First, we employ the Friedman rank-sum test [10] to assess whether at least two algorithms exhibit significant differences in the observed performance values. The p-values for the null hypothesis are $p_+ =$ 5e-56 and $p_\times =$ 2e-64 for f_+ and $f_\times$ respectively. At the 0.05 confidence level, the differences among the algorithms are significant. The statistical analysis proceeds with a post hoc analysis to determine which pairs of algorithms show significant differences in performance (for the three scenarios considered). In this step, we proceed to the Wilcoxon signed-rank test (as neither normality nor homoscedasticity conditions required for the application of parametric tests hold [7]) on the performance samples of each pair of algorithms. In addition, to reduce the issue of having Type I errors given multiple comparisons, the Bonferroni correction method is applied.

For all scenarios, Table 1 present tile-plots to illustrate all pairwise differences in the observed performance samples at the 0.05 confidence level. More specifically, the outcomes of the pairwise Wilcoxon-signed rank tests, without and with the application of the Bonferroni correction method, are provided on the left and right-hand side of the table respectively. Each tile corresponds to a pairwise significance test between the algorithms of the corresponding row and column. The color of the tile indicates if the observed performance differences were enough to reject the null hypothesis at the significance level (p-value < 0.05). Light gray tiles indicate significant differences between the pair of algorithms, while dark

■ Fail to reject H0 □ Reject H0 ($p < 0.05$)

(f_+)

	DE	GA	PSO	RO		DE	GA	PSO	RO
PSO				4e-18	PSO				4e-17
GA			4e-16	4e-18	GA			4e-15	4e-17
DE		4e-18	2e-4	4e-18	DE		4e-17	2e-3	4e-17
CMA-ES	3e-5	5e-13	■ 2e-1	4e-18	CMA-ES	3e-4	5e-12	■ 1.0	4e-17

($f_\times$)

	DE	GA	PSO	RO		DE	GA	PSO	RO
PSO				4e-18	PSO				4e-17
GA			1e-6	4e-18	GA			1e-5	4e-17
DE		4e-18	9e-5	4e-18	DE		4e-17	9e-4	4e-17
CMA-ES	2e-7	7e-4	2e-5	4e-18	CMA-ES	2e-6	7e-3	2e-4	4e-17

(f_+ vs. $f_\times$)

CMA	DE	GA	PSO	RO		CMA	DE	GA	PSO	RO
5e-6	3e-5	1e-15	2e-2	5e-18		3e-5	1e-4	5e-15	■ 8e-2	2e-17

Table 1: Pairwise Wilcoxon statistical tests (left) with Bonferroni post hoc analysis (right) for the three considered scenarios.

gray tiles indicate that no significant differences were observed. Analyzing these results, if we base acceptance or rejection of the above hypotheses, we arrive at the following insights: (i) in f_+ scenario PSO performances are not significantly different and (ii) Bonferroni correction reveals that PSO performances are the same whatever fitness function. Nevertheless, the performances of other algorithms depend on the chosen fitness function. Therefore, according to the algorithm considered, the fitness function choice has definitely an impact on the performances: f_+ is preferred when considering DE and GA while $f_\times$ is in the case of CMA-ES and PSO.
Finally, with respect to the conducted experiments, GA and CMA-ES will be investigated in the future as the first one gives good and stable results with high probability, whereas the second performs better overall (the best solutions are obtained using CMA-ES), but is subject to instability (due to exploration phases).

Visualization of the results. The application of bio-inspired algorithms allows us to exhibit different solutions consistent with BK and they seem complementary to the CSP approach. Both diagrams of fig. 8 present in red the overall best trajectory obtained by GA (a) and CMA-ES (b) together with the one in blue, obtained by the CSP approach using the CSP solver Absolute [13] combined with a possible strategy for cutting the search space [2]. The solutions provided by GA and CMA-ES illustrate the diversity of acceptable solutions that are compliant (the structure of the trajectories is similar) with BK. From a modelization perspective, it would be great to exhibit a diverse sampling of possible solutions, in order to reason not only on one possible identification but on a set of sensible identifications.

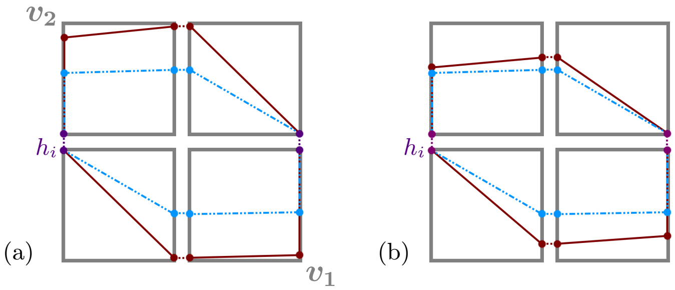

Fig. 8: Best trajectories (red) obtained by GA (a) and CMA-ES (b) with $f_\times$ compared to one of the solutions obtained by the CSP approach (blue).

5 Conclusion

The goal of this paper is to show that the problem of identifying variables in an hGRN, already formalized as a CSP, can be transformed into a bio-inspired optimization problem.

In previous works, many biological experiments have been formalized as constraints on time, behavior, and discrete events with the help of biologists' expertise. From these constraints, our work focused on finding how to model them as an FOP: we proposed a representation of a candidate solution and designed two appropriate fitness evaluation functions. To empirically test our approach, we conducted a study with a random optimization algorithm and four well-known continuous meta-heuristics: the proposed method shows satisfying results as the newly introduced BSR metric is high. In our experiments, CMA-ES obtains the overall best solutions satisfying BK constraints. Nevertheless, for this kind of problem, GA appears to be the best meta-heuristic because of its high probability of getting good results.

The proof-of-concept developed in this paper will shortly be applied to designing a new cell cycle hGRN model where time plays a crucial role in passing through each phase. Although this cell cycle model contains only 5 abstract genes, the number of celerities is about 240. The optimization problem will be challenging and lead us to apply *large-scale* optimization algorithms.

Moreover, when working with biologists, our ability to propose different solutions compliant with BK is of great importance because it leads to considerate new information which would not exhibit otherwise. Diversity in solutions reflects, on the one hand, a plurality of functioning within an observed system and, on the other hand, helps to evaluate the robustness of oscillating biological systems (the more diversity, the more robustness). From such a perspective, future work will focus on *multimodal* approaches that could be able to sample the set of solutions compliant with the formalized biological knowledge.

References

1. Behaegel, J., Comet, J.P., Folschette, F.: Constraint identification using modified Hoare logic on hybrid models of gene networks. In: Proceedings of the 24th Int. Symposium TIME (2017). https://doi.org/10.4230/LIPIcs.TIME.2017.5
2. Behaegel, J., Comet, J.P., Pelleau, M.: Identification of dynamic parameters for gene networks. In: Proceedings of the 30th IEEE Int. Conf. ICTAI (2018)
3. Biswas, S., Acharyya, S.: Neural model of gene regulatory network: a survey on supportive meta-heuristics. Theory in Biosciences (2016). https://doi.org/10.1007/s12064-016-0224-z
4. Blank, J., Deb, K.: pymoo: Multi-objective optimization in python. IEEE Access (2020)
5. Buchet, S., Carbone, F., Magnin, M., Ménager, M., Roux, O.: Inference of Gene Networks from Single Cell Data through Quantified Inductive Logic Programming (2021), https://doi.org/10.1145/3486713.3486746
6. Coello, C.A.C.: Constraint-handling techniques used with evolutionary algorithms. In: Proceedings of GECCO (2021). https://doi.org/10.1145/3449726.3461400
7. Eftimov, T., Korošec, P.: Statistical Analyses for Meta-Heuristic Stochastic Optimization Algorithms: GECCO Tutorial (2020), https://doi.org/10.1145/3377929.3389881
8. Eiben, A.E., Smith, J.E.: Constraint Handling (2015). https://doi.org/10.1007/978-3-662-44874-8_13
9. Hansen, N.: The CMA Evolution Strategy: A Comparing Review (2006). https://doi.org/10.1007/3-540-32494-1_4
10. Hollander, M., Wolfe, D.A., Chicken, E.: Nonparametric statistical methods (2013)
11. Matyas, J., et al.: Random optimization. Automation and Remote control (1965)
12. Mitra, S., Biswas, S., Acharyya, S.: Application of meta-heuristics on reconstructing gene regulatory network: A bayesian model approach. IETE Journal of Research (2021). https://doi.org/10.1080/03772063.2021.1946433
13. Pelleau, M., Miné, A., Truchet, C., Benhamou, F.: A constraint solver based on abstract domains. In: 14th Int. Conf. VMCAI 2013. https://doi.org/10.1007/978-3-642-35873-9_26
14. Price, K., Storn, R.M., Lampinen, J.A.: Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series) (2005)
15. da Silva, J.E.H., Betnardino, H.S., Helio J.C., B., Vieira, A.B., Luciana C.D., C., de Oliveira, I.L.: Inferring gene regulatory network models from time-series data using metaheuristics. In: IEEE CEC (2020). https://doi.org/10.1109/CEC48606.2020.9185572
16. Spirov, A., Holloway, D.: Using evolutionary computations to understand the design and evolution of gene and cell regulatory networks. Methods (2013). https://doi.org/10.1016/j.ymeth.2013.05.013
17. Thomas, R.: Boolean formalization of genetic control circuits. J.T.B. (1973)
18. Zhan, Z.H., Zhang, J., Li, Y., Chung, H.S.H.: Adaptive particle swarm optimization. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) (2009). https://doi.org/10.1109/TSMCB.2009.2015956