

Paramétrisation de réseaux de régulation biologiques avec l'évolution artificielle et l'apprentissage par renforcement

Denis Pallez, Guillaume Grataloup

Jean-Paul Comet, Gilles Bernot

Université Côte d'Azur, CNRS, I3S

L'objectif de la modélisation des réseaux de régulation biologique (RRB¹) est d'étudier et de comprendre les mécanismes qui permettent aux systèmes biologiques d'accomplir des fonctions essentielles, allant du métabolisme à l'adaptation aux perturbations environnementales. L'étude de la dynamique de ces systèmes ouvre de nouvelles perspectives avec des applications importantes en biologie fondamentale, de la biologie moléculaire à l'écologie, mais aussi en biologie synthétique, en bioproduction, en pharmacologie, en médecine ou en chronothérapie. Bien que de nombreux cadres aient été proposés pour modéliser les RRB, en particulier les réseaux de régulation génétique (GRN), les modèles obtenus sont souvent grossiers et ne tiennent pas compte des valeurs seuils régissant les transitions entre différents niveaux d'expression qualitative des gènes ou le temps passé dans un état donné.

Dans cet article, nous nous concentrons sur ces modèles grossiers, pour lesquels les valeurs précises des seuils sous-jacents sont inconnues. Nous formulons d'abord le problème clé de la modélisation, à savoir l'identification des paramètres, comme un problème d'optimisation sous contraintes. Cette étape d'identification est ensuite résolue à l'aide de métaheuristiques à base

1. En anglais : biological regulatory network — BRN.

de population, capables de gérer ces contraintes. Le modèle est ensuite affiné pour prendre en compte le temps passé dans des états qualitatifs, ce qui est rendu possible par le remplacement des paramètres qualitatifs par des paramètres continus appelés «célérités». Ce nouveau problème d'identification est résolu à l'aide d'un problème d'optimisation sans contraintes, résolu avec succès à l'aide de métaheuristiques standard, surpassant ainsi les solveurs de contraintes continues (cCSP). Enfin, la recherche arborescente Monte Carlo (MCTS) a été améliorée dans le domaine continu afin de réduire le temps d'évaluation des solutions. Dans les deux derniers cas, les algorithmes sont adaptés pour trouver un ensemble diversifié de solutions. Ce travail montre que l'intelligence computationnelle permet de modéliser efficacement des réseaux de régulation à grain fin, et de proposer diverses solutions biologiquement plausibles.

1. Réseaux de régulation

Plusieurs cadres de modélisation ont été développés pour représenter les RRB, chacun offrant une perspective différente sur la dynamique du système. Le *cadre différentiel* utilise des équations différentielles ordinaires pour décrire l'évolution continue de l'expression des gènes dans le temps. Le *cadre stochastique*, quant à lui, prend en compte l'aspect aléatoire des processus biologiques, en modélisant les transitions entre états comme des phénomènes probabilistes. Enfin, le *cadre discret* propose une représentation abstraite en ne considérant qu'un nombre fini d'états, souvent binaire (présence ou absence d'une entité biologique) permettant une analyse qualitative du comportement du système. Toutes ces approches reposent en premier lieu sur la connaissance des activations (+) ou inhibitions (−) potentielles entre les entités biologiques : toutes ces interactions constituent un graphe *de régulation* ou *d'interaction* (figure 1, à gauche).

Malheureusement, le graphe d'interaction global n'est pas connu de manière exhaustive, et le nombre d'entités impliquées est trop grand pour qu'une approche globale puisse voir le jour. Ainsi, d'autres approches ont émergé : elles consistent à inférer un RRB à partir de nombreuses données biologiques expérimentales [13, 24, 9, 8]. La force de ces approches est de mettre au jour des interactions entre les entités biologiques qui n'étaient pas encore connues. Cependant ces méthodes ne permettent pas d'avoir les conditions précises dans lesquelles les activations ou inhibitions ont lieu. Les seuils de déclenchement des interactions constituent encore aujourd'hui une pierre d'achoppement pour la modélisation. Plus généralement, ce sont les *paramètres* qui régissent l'activité de chaque gène en fonction des autres gènes qui sont extrêmement difficiles à identifier car ils sont sensibles, d'une part, aux conditions expérimentales et, d'autre part, au niveau d'abstraction

de l'étude. Comme ces paramètres ne sont généralement pas directement mesurables par des expérimentations biologiques, ils nécessitent des raisonnements subtils et complexes pour déduire leur valeur du comportement global observé du système. Cette identification des paramètres constitue le goulot d'étranglement de la modélisation d'un RRB, tout particulièrement dans le cadre discret. C'est ce sur quoi travaille l'équipe de recherche en bio-informatique du laboratoire I3S² depuis de nombreuses années. L'une des voies envisagées pour cette identification des paramètres consiste à exploiter, non pas les données expérimentales brutes, mais des observations biologiques qualitatives issues de ces données brutes ainsi, bien sûr, que d'autres connaissances biologiques déjà établies par la communauté sur les systèmes biologiques étudiés. Ces observations sont formalisées en utilisant soit des logiques temporelles soit une version modifiée de la logique de Hoare [5] selon qu'elles représentent une connaissance sur la dynamique globale du système ou une suite d'observations issue d'une expérience. Cette formalisation permet de représenter l'ensemble des contraintes que le modèle du réseau de régulation biologique doit satisfaire pour qu'il soit en accord avec l'ensemble des connaissances disponibles. Toutefois, cette démarche est extrêmement fastidieuse et coûteuse en temps : la modélisation fine d'un RRB, qui nécessite la connaissance de son graphe d'interaction et des paramètres expliquant la dynamique, s'étale généralement sur plusieurs thèses en bio-informatique ou en biologie.

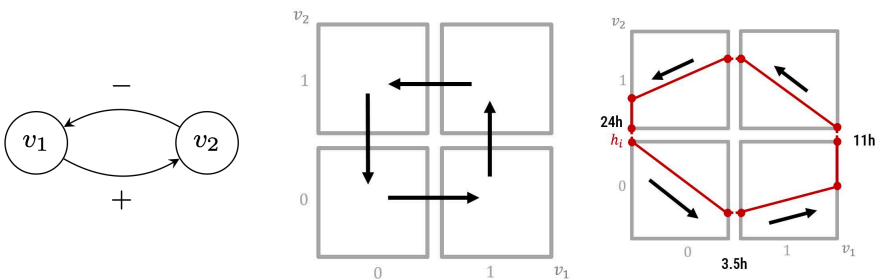


Fig. 1. À gauche : exemple d'un graphe d'interaction représentant le cycle circadien entre les gènes v_1 et v_2 . Au centre : sa représentation discrète sachant que chaque gène possède 2 états discrets 0 ou 1. À droite : une dynamique possible dans le graphe hybride.

Le cadre discret de modélisation est souvent un point de départ intéressant car d'une part, les interactions sont souvent à seuil (sigmoïde) et, d'autre

2. Laboratoire d'informatique, signaux et systèmes de Sophia Antipolis, <https://www.i3s.unice.fr>.

part, l'accent est mis sur des comportements qualitatifs souvent plus facilement observables. Pour construire un modèle discret, il est avant tout nécessaire d'identifier le nombre d'états (discrets) de chacun des gènes du RRB. Ces états discrets sont le résultat d'abstraction de la concentration du produit des gènes : si on considère une activation d'un gène v_1 sur un gène v_2 , en dessous d'un certain seuil du produit du gène v_1 , l'activation n'a pas lieu, et au dessus, elle a lieu. Cette notion de seuil est cruciale, et mène à une discrétisation de l'espace des états (cf. figure 1, au milieu). Vu le nombre de gènes et le nombre d'interactions impliqués, une approche manuelle n'est pas envisageable, et l'équipe de recherche tente de développer des approches permettant d'automatiser l'identification de ces seuils qui caractérisent les états discrets. L'idée de base est de partir des données brutes et de ramener le problème d'identification des seuils à un problème d'optimisation sous contraintes. Ce problème contraint est optimisé grâce à des métaheuristiques à base de population capables de gérer des contraintes exprimées sous forme d'inégalités, que nous expliquons dans la section 2.

Une fois les seuils identifiés, les états qualitatifs pertinents pour le système étudié sont évidents, et il est relativement facile d'expliquer la succession des événements, et par conséquent la succession des états discrets traversés au cours du temps. Toutefois, cela ne permet pas de déterminer le temps passé dans chacun des états, alors que la question temporelle traverse toute la biologie. C'est là qu'une extension *hybride* (RRBh) du cadre de modélisation discret prend tout son sens : le temps passé au sein de chaque état discret est déduit d'une trajectoire affine dont la direction et la vitesse sont représentées par un vecteur *célérité* (figure 1, à droite). Ainsi, les trajectoires du modèle sont des trajectoires linéaires par morceaux dans un espace multidimensionnel. Les paramètres de ces trajectoires, *i.e.* les vecteurs célérité, sont soumis à des contraintes : ils doivent permettre de représenter l'ensemble des observations qualitatives formalisées grâce à la logique de Hoare. Behaegel *et al.* [3] a logiquement utilisé des solveurs de satisfaction de contraintes continues (cCSP) pour échantillonner l'ensemble de toutes les paramétrisations compatibles avec les observations, mais les solveurs n'ont pas réussi à exhiber des solutions dès lors que le nombre de gènes du RRB dépassait trois. Pour exhiber tout de même des solutions, nous avons cherché à transformer ce problème de satisfaction de contraintes en un problème d'optimisation sans contrainte. Nous avons à nouveau adapté et comparé plusieurs métaheuristiques que nous détaillons dans la section 3.

Les paramètres recherchés sont les vecteurs célérité associés aux états traversés par la trajectoire, et cette trajectoire donne un ordre naturel dans l'ensemble des paramètres à rechercher. Cette particularité permet de considérer le problème d'identification des paramètres, dans un troisième temps, comme un enchaînement de décisions à prendre, état discret par état discret.

Pour cela, nous avons utilisé l'apprentissage par renforcement avec un algorithme de recherche arborescente Monte Carlo (MCTS) que nous détaillons dans la section 4.

Chacune des approches présentées dans ce travail a été testée sur un ou plusieurs systèmes biologiques distincts, notamment : un cycle de rétroaction négative à deux gènes (dont on attend un comportement périodique), un modèle abstrait du cycle circadien impliquant trois gènes, ainsi qu'un modèle du cycle cellulaire à cinq gènes. Enfin, nous exposons à la fin nos conclusions et les perspectives envisagées.

2. Optimisation sous contraintes

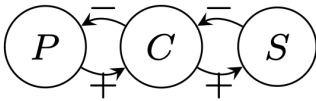


Fig. 2. Graphe d'interaction impliquant les gènes P, C et S.

À cette étape — et comme annoncé précédemment — nous supposons connu un graphe d'interaction (figure 2) ainsi que des données brutes qui représentent le niveau d'expression du produit des gènes (impliqués dans le graphe) au cours du temps (croix dans la figure 3, reliées en

pointillés). L'objectif est de déterminer *automatiquement* les valeurs des seuils de chaque gène du RRB (traits horizontaux), ce qui se faisait jusque-là manuellement. Le seuil noté θ_P^{n+1} désigne une valeur scalaire représentant la concentration du produit du gène P qui sépare le $(n + 1)$ -ième état discret du n -ième. Dans un RRB, les seuils obéissent à des contraintes d'ordre, telles que $\theta_P^1 \leq \theta_P^2$, reflétant une progression croissante de la concentration du produit de P entre les états discrets 0 (en dessous des deux seuils), 1 (entre les deux seuils) et 2 (au dessus des deux seuils).

La connaissance du graphe d'interaction et des seuils de chaque gène ne permet pas, à elle seule, de définir la dynamique complète du réseau : les concentrations des produits des gènes évoluent dans le temps, provoquant potentiellement des transitions entre états discrets. Le passage d'un état à un autre d'un gène x est provoqué par des variations dans la concentration des produits des gènes qui précèdent x dans le graphe d'interaction. Toutefois, comme x peut avoir plusieurs prédécesseurs, il est essentiel de déterminer quels prédécesseurs ont une activité incitant x à changer d'état : cela peut être un activateur actif (+) de x qui l'aide à s'exprimer suffisamment pour que x franchisse son seuil, ou un inhibiteur actif (−) dont la concentration diminue incitant x à s'exprimer, ou la combinaison de plusieurs prédécesseurs... Par la suite, x pourra faire évoluer à son tour d'autres gènes.

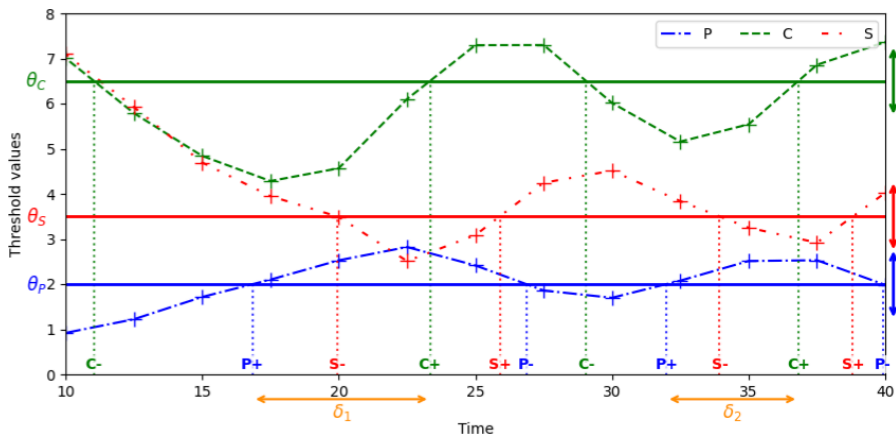


Fig. 3. Évolution de l'expression observée du produit des gènes du graphe de la figure 2 (croix) et les valeurs recherchées des seuils (traits pleins horizontaux) qui permettent de déduire une suite d'événements biologiques ($C-$, $P+$, $S-$, ...).

Ainsi, pour définir finement la dynamique d'un RRB, on a besoin de caractériser à la fois les seuils de chaque gène mais aussi ses ressources : une ressource est un prédécesseur qui aide le gène *cible* à s'exprimer, c'est à dire un activateur (+) qui dépasse son seuil d'activation, ou un inhibiteur (−) qui est en dessous du seuil d'inhibition. D'après le graphe d'interaction, il est possible de déduire :

- le nombre de seuils différents pour chaque gène x , sachant qu'il y a au maximum autant de seuils qu'il y a de régulations contrôlées par x ;
- le nombre de conditions différentes contrôlant le gène x : lorsque x a n prédécesseurs, il faut considérer toutes les situations distinctes où chaque prédécesseur peut être ou non une ressource ; ainsi, il y aura 2^n conditions possibles (les parties d'un ensemble à n éléments).

Pour chacune de ces situations, il est nécessaire de décrire le comportement du gène cible : vers quel état discret x va-t-il être attiré lorsqu'il est soumis à tel ensemble de ressources. Ces valeurs sont données par de nouveaux paramètres, notés $K_{x,\omega}$, qui donnent l'état discret vers lequel x est attiré lorsque ω est l'ensemble des ressources. En plus des contraintes d'ordre sur les seuils, il existe d'autres contraintes, dites de Snoussi [10], qui expriment le fait que la valeur de l'état discret vers lequel un gène est attiré, ne peut pas décroître si ses ressources augmentent.

La figure 3 présuppose connues les valeurs des seuils des gènes du RRB (traits horizontaux). À l'aide des données brutes observées (croix), il est possible de savoir qu'un gène passe un seuil de manière croissante — la concentration de son produit a augmenté — ou décroissante — la concentration de son

produit a diminué. En suivant l'axe horizontal de cette figure représentant le temps, le premier événement pertinent est le fait que le produit du gène C (courbe supérieure de la figure) passe d'une valeur supérieure au seuil fixé de C (ligne horizontale) à une valeur inférieure. Cet événement est noté $C-$ au dessus de l'axe du temps de la figure. De manière similaire, le deuxième événement pertinent est $P+$ lorsque la courbe bleue en pointillé passe le seuil de P en montant. Le troisième est $S-$. La suite des événements est donc $C-, P+, S-, C+, S+, P-, C-, P+, S-, C+, S+, P-$. Notons qu'un changement des seuils (supposés connus) va entraîner un changement de cette suite d'événements.

Intéressons-nous à l'événement $C+$ qui apparaît deux fois dans la suite précédente. Cet événement peut être expliqué soit par l'augmentation de son activateur P , soit par la diminution de son inhibiteur S dans le graphe d'interaction (figure 2) et on constate par deux fois, dans la suite des événements, que $P+$ et $S-$ précèdent l'événement $C+$. On postule que si les seuils sont bien choisis alors le temps qui sépare l'apparition d'une ressource et l'augmentation de la cible est stable au cours du temps. Dans notre cas, il est donc utile d'évaluer la variabilité des délais (indiqués par des flèches orange en dessous de l'axe du temps sur la figure 3) entre les événements ressources ($P+$ ou $S-$) et l'événement conséquence $C+$. La proximité de ces différentes valeurs suggère une bonne adéquation des seuils associés. Il faut bien évidemment faire de même avec les autres gènes : on observe dans les données deux occurrences de $C-, P+$ en sachant que C est ressource de P . Ainsi, nous avons transformé l'identification des valeurs des seuils du RRB en un problème de minimisation de la variabilité des délais (variance) entre des événements biologiques en tenant compte de deux types de contraintes (d'ordre sur les seuils et de Snoussi).

Pour résoudre ce problème d'optimisation, nous avons considéré les 3 meilleures métaheuristiques à base de population issues de la compétition CEC'2020 [17] de la catégorie problèmes d'optimisation sous contraintes à objectif unique dans le monde réel³. Cette compétition a comparé 8 algorithmes d'optimisation gérant des contraintes sur 57 problèmes réels issus de différents domaines (chimie, mécanique, électronique... mais pas bio-informatique), et contenant entre 2 et 158 variables à optimiser, entre 0 et 148 contraintes d'égalité et entre 0 et 91 contraintes d'inégalité. Les algorithmes ont été classés en fonction de la somme pondérée des meilleurs résultats, de la médiane et de la moyenne obtenus sur les 57 problèmes ; une pondération plus importante étant accordée aux dimensions plus élevées. Les gagnants étaient *Self-adaptive spherical search* (SASS) [15], suivi de *Constrained optimization with Lévy flights differential evolution* (COLSHADE) [12] et de *Modified covariance matrix adaptation evolution strategy* (sCMaGES) [15].

3. <https://github.com/P-N-Suganthan/2020-RW-Constrained-Optimisation>.

Nous avons adapté ces algorithmes [11] à notre problème d'identification des seuils. La première étape de cette adaptation consiste à identifier les paramètres représentant une solution candidate Ψ manipulée par les algorithmes. Si x_1, x_2, \dots, x_n est l'ensemble des gènes du RRB et t_i le nombre de seuils de x_i , Ψ contient la valeur du premier seuil de chaque gène du RRB, $\theta_{x_i}^1$, à laquelle s'ajoute une valeur $\delta_{x_i}^1$ pour obtenir la valeur du seuil $\theta_{x_i}^2 = \theta_{x_i}^1 + \delta_{x_i}^1$, à laquelle s'ajoute la valeur $\delta_{x_i}^2$ pour obtenir la valeur du seuil suivant ($\theta_{x_i}^3 = \theta_{x_i}^2 + \delta_{x_i}^2$) et ainsi de suite jusqu'à $\theta_{x_i}^{t_i}$ et cela pour l'ensemble des gènes du RRB ($i \in \llbracket 1, n \rrbracket$). Cela permet de tenir compte indirectement des contraintes de seuils introduites précédemment ($\theta_{x_i}^1 \leq \theta_{x_i}^2 \leq \theta_{x_i}^3$). L'algorithme sera en charge de vérifier uniquement les contraintes de Snoussi.

À ces valeurs, s'ajoutent également les paramètres K des ensembles de ressources $r_i^1 = \emptyset, r_i^2, \dots, r_i^{j_i}$ de x_i pour l'ensemble des gènes. Ainsi, $\Psi = [\theta_{x_i}^1; \delta_{x_i}^1; \dots; \delta_{x_i}^{t_i-1}; K_{x_i, r_i^1}; \dots; K_{x_i, r_i^{j_i}}]_{i=1, \dots, n}$. Il est important de noter que les valeurs $\theta_{x_i}^{j_i}$ et $\delta_{x_i}^{t_i}$ sont réelles alors que les $K_{x_i, r_i^{j_i}}$ sont entières. Pour le RRB de la figure 2 qui considère que les seuils des régulations $C \rightarrow P$ et $C \rightarrow S$ sont identiques, Ψ contiendra 3 réels (les seuils des 3 gènes) et 8 entiers, $\$K_{P, \emptyset}, K_{P, C}, K_{C, \emptyset}, K_{C, P}, K_{C, S}, K_{C, PS}, K_{S, \emptyset}$ et $K_{S, C}$, qui doivent respecter 6 contraintes de Snoussi

$$\begin{aligned} K_{P, \emptyset} &\leq K_{P, C}, & K_{C, \emptyset} &\leq K_{C, P}, & K_{C, \emptyset} &\leq K_{C, S}, \\ K_{C, P} &\leq K_{C, PS}, & K_{C, S} &\leq K_{C, PS}, & K_{S, \emptyset} &\leq K_{S, C}. \end{aligned}$$

La deuxième étape consiste à définir la fonction d'évaluation d'une solution candidate et nous utiliserons la technique expliquée précédemment qui consiste à minimiser la variance des délais des différents passages de seuils, et ce, sur l'ensemble des gènes.

Nous avons exécuté chaque algorithme 50 fois avec à chaque fois, la même population initiale pour chacun des algorithmes, composée de 100 solutions candidates choisies aléatoirement. Le nombre de mesures expérimentales (croix) pour chaque gène est de 30 équitablement réparties dans le temps. À chaque algorithme est alloué un budget maximum de 10 000 évaluations de la fonction à minimiser (*number of function evaluations*, NFE). L'évolution moyenne des 50 exécutions de la qualité des solutions au cours du temps (NFE) est représentée sur la figure 4 pour chaque algorithme. On constate que SASS converge rapidement vers une solution optimale car la variance des délais devient nulle alors que sCMaGES converge vers un optimal local. COLSHADE semble converger mais beaucoup plus lentement que SASS qui obtient un optimum global après 7 000 évaluations. Sur cette figure, on peut également constater que les populations de COLSHADE et sCMaGES conservent une certaine diversité alors que c'est beaucoup moins le cas pour SASS.

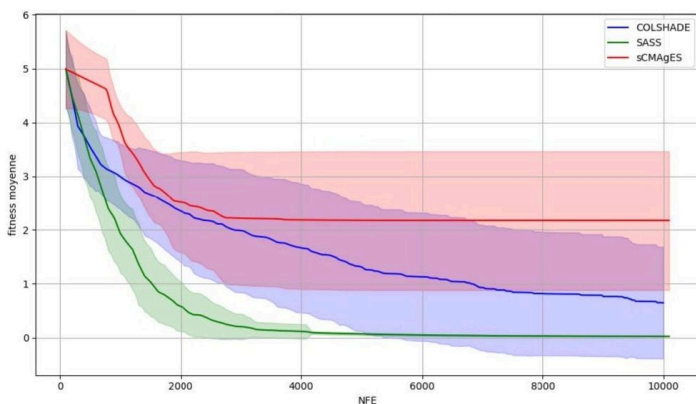


Fig. 4. Courbes de convergence des trois métaheuristiques COLSHADE, SASS et sCMaGES.

La figure 5 présente les fonctions de répartition (*cumulative distribution function*) qui décrivent, pour chaque algorithme, la probabilité (en ordonnée) de trouver une solution ayant une qualité égale ou inférieure à une valeur donnée (en abscisse).

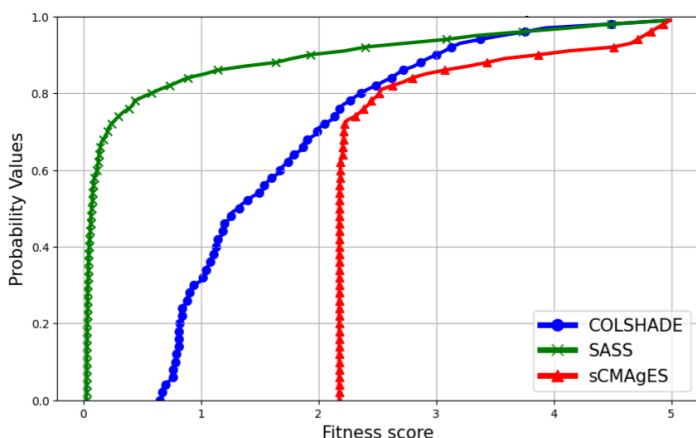


Fig. 5. Fontions de répartition des trois métaheuristiques COLSHADE, SASS et sCMaGES.

Ainsi, on constate que sur 50 exécutions, SASS a une probabilité d'environ 60 % de trouver une très bonne solution d'une qualité inférieure à 0.2 alors que les deux autres algorithmes n'y arrivent jamais. Non seulement, cela confirme les résultats de la compétition CEC'2020 [17] qui avait été gagnée par SASS mais surtout cela confirme qu'il est possible de déterminer automatiquement des seuils à partir de données brutes.

Par la suite, nous devons étudier le passage à l'échelle de notre approche, en l'appliquant à des RRB de plus grande taille, tout en déterminant le nombre minimal de mesures expérimentales nécessaires pour identifier les seuils. On pourra noter que nous avons utilisé des algorithmes d'optimisation conçus pour des variables continues, alors que les paramètres manipulés sont mixtes (entiers et réels). Cela implique que les algorithmes doivent explorer un espace de recherche beaucoup plus vaste que dans le cas d'un problème purement continu, ce qui peut expliquer la faible probabilité d'obtenir une solution satisfaisante. À l'avenir, nous envisageons de recourir à des algorithmes d'optimisation mixte, capables de gérer efficacement différents types de variables, dans l'espoir d'améliorer la convergence et d'augmenter la probabilité d'atteindre une solution optimale.

3. Optimisation sans contraintes

Dans ce nouveau travail, nous considérons les seuils des gènes comme identifiés mais nous passons du cadre de modélisation discret (figure 1, au centre) au cadre de modélisation hybride (figure 1, à droite) qui ajoute au cadre discret le temps passé dans chacun des états. Cela conduit à la recherche de trajectoires linéaires par morceaux dans un espace multidimensionnel. Ces trajectoires doivent respecter une séquence d'événements biologiques observés, qui a été précédemment et manuellement interprétée par les biologistes comme un ensemble de contraintes basées sur la logique de Hoare [5]. Les trajectoires sont caractérisées par une collection de célérités $\{C_{v,\eta}\}$ indexées par un des gènes v et un état discret η (correspondant à la vitesse d'évolution de la concentration du produit du gène v dans l'état η) et par un point de départ h_i . Ainsi, la représentation d'une solution candidate de la figure 1 (à droite) est $\Psi = [h_i; C_{v_1,(0,0)}; C_{v_2,(0,0)}; C_{v_1,(1,0)}; C_{v_2,(1,0)}; C_{v_1,(1,1)}; C_{v_2,(1,1)}; C_{v_1,(0,1)}; C_{v_2,(0,1)}]$ dont toutes les valeurs sont réelles.

Pour identifier les paramètres des trajectoires qui respectent les contraintes biologiques, nous avons tout d'abord transformé le problème de satisfaction de contraintes en un problème d'optimisation sans contrainte (*free optimisation problem*) en définissant une fonction de pénalité à minimiser qui calcule dans quelle mesure une solution candidate satisfait les contraintes [20].

Nous avons ensuite comparé plusieurs métaheuristiques telles que *Differential evolution* (DE), *Particle swarm optimisation* (PSO), *Covariance matrix adaptation evolution strategies* (CMA-ES), *Genetic algorithm* (GA), et *Random optimisation* (RO) qui sert de référence, et nous avons montré qu'elles sont capables d'identifier une solution équivalente à celles trouvées par les solveurs cCSP. Les résultats sous forme de fonction de répartition des différentes métaheuristiques sont présentées dans la figure 6. CMA-ES obtient

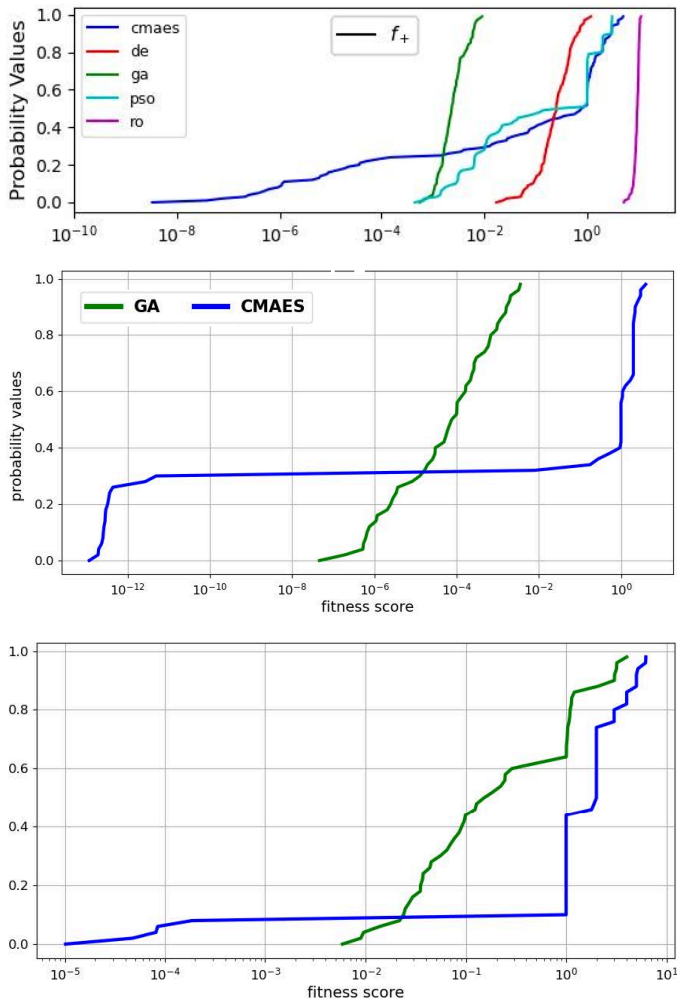


Fig. 6. Fonctions de répartition des différentes métaheuristiques sur un RRB à 2 gènes (en haut), 3 gènes (au centre) et 5 gènes (en bas), pour identifier les paramètres des célérités.

d'excellents résultats (valeur d'évaluation égale à 10^{-8} pour 2 gènes — voir figure 6, en haut — avec un optimum global à 0) mais avec une très faible probabilité (proche de 0) alors que GA obtient des résultats très intéressants

d'un point de vue biologique (une précision de 10^{-2} est suffisante) mais avec une très forte probabilité. Ces métaheuristiques ont même été capables d'identifier une solution pour des RRB de plus grande taille (5 gènes, cf. figure 6, en bas) pour lesquels les cCSP échouaient.

Multiples solutions ?

Les métaheuristiques à base de population considèrent généralement une structure panmictique (qui assure à chaque individu une probabilité identique d'interagir avec tous les autres membres de la population). L'inconvénient de ces approches est une diminution de la diversité de la population qui à terme converge vers un seul bassin d'attraction de l'espace de recherche alors que la connaissance de la diversité des solutions est importante pour le biologiste. À l'inverse, les algorithmes génétiques *cellulaires* [2], inspirés de la modélisation de systèmes complexes à l'aide d'automates cellulaires, impose une structuration de la population suivant une certaine topologie obligeant les échantillons à n'interagir (se reproduire) qu'avec leurs voisins proches dans la structure (figure 7).

Cette restriction d'interaction et le chevauchement de voisinage aident à l'exploration de l'espace de recherche tout en réduisant l'influence qu'un échantillon peut avoir sur l'ensemble de la population. L'intérêt de ce modèle d'évolution est qu'il permet de maintenir une certaine diversité dans la population favorisant ainsi une optimisation multimodale [14] au sein d'une même population.

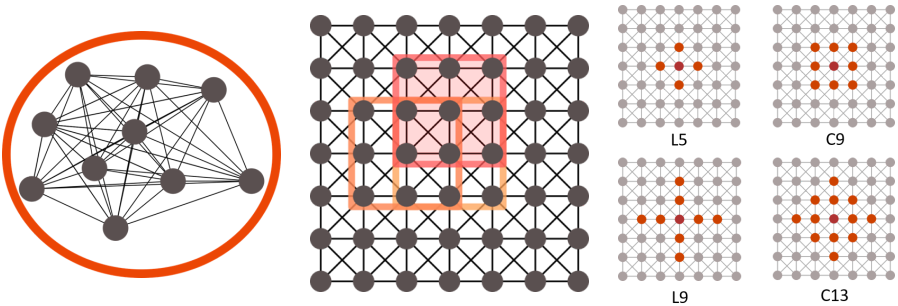


Fig. 7. Population panmictique (à gauche) et population possédant une certaine structure utilisée dans les algorithmes cellulaires (au centre) avec différents voisinages possibles (à droite). Un point correspond à une solution et une arête à une interaction possible.

Dans [19], afin d’avoir une situation comparable produite par les CSP (qui caractérisent une infinité de solutions), nous avons adapté ce schéma d’optimisation à notre problème pour obtenir plusieurs trajectoires optimales pour des RRBh de différentes tailles (2, 3 et 5 gènes). Pour cela, nous avons testé plusieurs structures de voisinage différentes (L5, L9, L29, L41, L13, C9, L pour linéaire et C pour compact, figure 7, à droite). Ces approches ont été comparées à l’algorithme *Repelling subpopulations in covariance matrix self-adaptation evolution strategy II* (RS-CMSA-ESII) [1], vainqueur de la compétition CEC’2020 [17] de la catégorie « techniques de niche pour l’optimisation multimodale⁴ sur une dizaine de problèmes de tests [18]. Il a la particularité d’être basé sur l’algorithme CMA-ES auquel nous nous sommes comparés [20] et qui est capable d’obtenir d’excellents résultats mais avec une très faible probabilité. RS-CMSA-ESII est basé sur un schéma de coévolution visant à trouver des optima distincts (un par population) et utilise des points tabous de l’espace dans chaque sous-population pour construire de nouveaux échantillons qui sont éloignés des points tabous en se basant sur la distance de Mahalanobis.

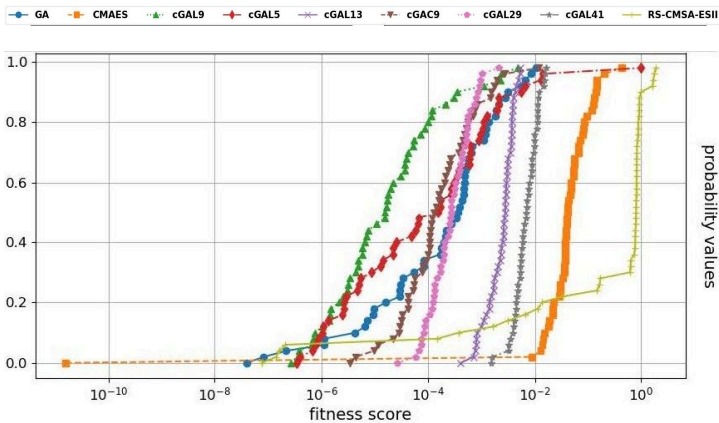


Fig. 8. Fonctions de répartition de plusieurs métaheuristiques avec un RRBh à 2 gènes moyennées sur 50 exécutions.

Nos expérimentations ont été répétées 50 fois avec un budget de 100 000 évaluations pour des RRBh à 2 gènes (figure 8) et 3 gènes (figure 9) et 200 000 évaluations pour un RRBh à 5 gènes (figure 10).

4. <https://github.com/mikeagn/CEC2013/tree/master>, <https://www.epitropakis.co.uk/ieee-mm0>.

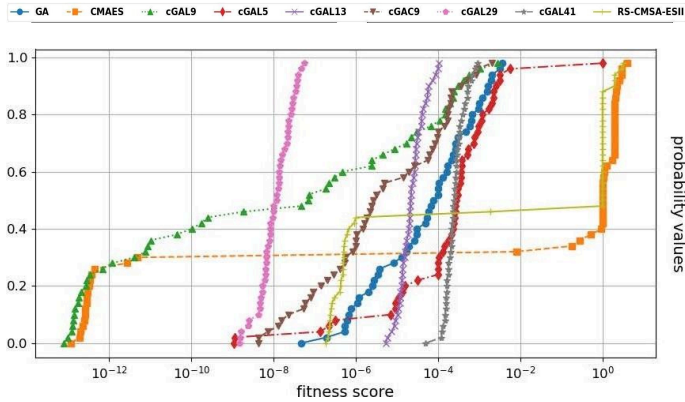


Fig. 9. Fonctions de répartition de plusieurs métaheuristiques avec un RRBh à 3 gènes moyennées sur 50 exécutions.

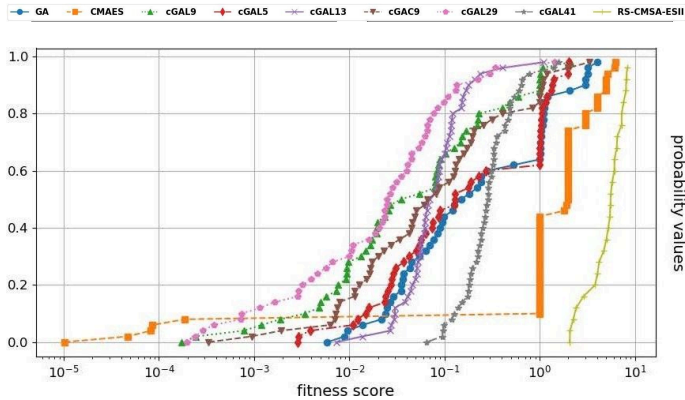


Fig. 10. Fonctions de répartition de plusieurs métaheuristiques avec un RRBh à 5 gènes moyennées sur 50 exécutions.

On peut observer que, comme prévu, les métaheuristiques panmictiques (GA, CMA-ES) sont moins performantes que l'algorithme génétique cellulaire (cGA) dans tous les cas car elles convergent plus rapidement vers un optimum local alors que cGA converge plus lentement mais en maintenant plus de diversité dans la population. Les résultats se confirment pour CMA-ES qui reste la technique qui obtient les meilleurs résultats mais avec une trop faible probabilité. La partie surprenante est que RS-CMSA-ESII obtient de moins bons résultats que CMA-ES et qu'il est souvent à la traîne pour ce qui est de trouver des solutions de qualité.

Nous avons mené une campagne de tests statistiques qui démontrent que sur l'ensemble des trois problèmes (2 à 5 gènes), cGAL9 et cGAL29 sont plus compétitifs que les autres techniques pour trouver plus d'optima avec des échantillons de meilleure qualité. Il est surprenant que RS-CMSA-ESII soit aussi contre-performant sur sa capacité à obtenir une grande diversité d'échantillons alors qu'il a gagné la compétition CEC'2020. En fait, cela est dû au fait que RS-CMSA-ESII ne garde au mieux qu'une seule solution par bassin d'attraction afin de pouvoir conserver d'autres solutions dans d'autres bassins. Or, notre problème d'optimisation est très particulier puisqu'il peut contenir une infinité de solutions optimales dans le même bassin d'attraction (plateau optimal, figure 12).

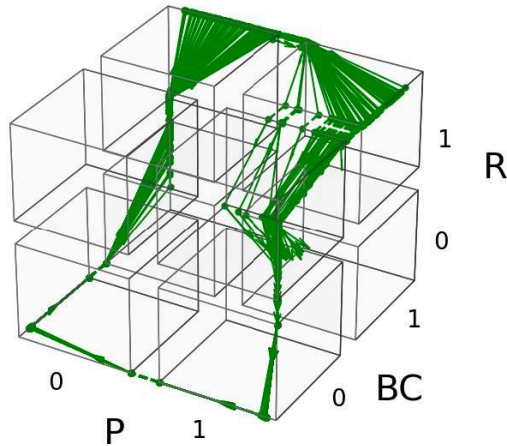


Fig. 11. Trajectoires diversifiées obtenues en une seule exécution avec cGAL9 sur un RRBh à 3 gènes.

Sur la figure 11, on peut observer les trajectoires qui valident les observations biologiques obtenues en une seule exécution de cGAL9 sur un RRBh à 3 gènes.

L'analyse des fonctions de test des compétitions sur les problèmes multi-modaux [18] montre qu'aucune ne correspond aux caractéristiques de notre problème, soulignant ainsi les limites des fonctions de test actuellement utilisées dans la communauté. Ce problème de modélisation de réseaux de régulation a été proposé à la communauté scientifique dans le cadre de notre implication au sein du réseau européen COST *Action randomised optimisation algorithms research network*⁵ et du groupe de travail consacré à la modélisation de problèmes.

5. <https://roar-net.eu>.

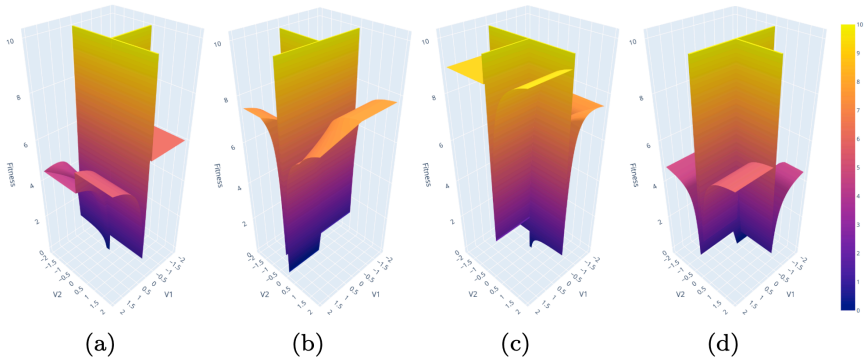


Fig. 12. Visualisation de l'espace de recherche des trajectoires du RRBh à 2 gènes décomposé état discret par état discret qui fait apparaître en violet l'infinité de solutions : $(v_0 = 0, v_1 = 0)$ en (a), $(1, 0)$ (b), $(1, 1)$ (c), $(0, 1)$ (d).

4. Problème de décision séquentielle

Dans l'approche précédemment décrite, chaque solution candidate représente l'ensemble des célérités définissant une trajectoire à simuler. Si cette trajectoire conduit prématurément le système vers un état stationnaire, la simulation initiale ne permet pas d'en évaluer la suite éventuelle, nécessitant ainsi une relance. Ce processus conduit à l'évaluation de solutions non viables, augmentant inutilement le temps de calcul et pénalisant les performances de l'algorithme.

Dans ce nouveau travail, nous adoptons une stratégie alternative complètement différente qui consiste à identifier les célérités de manière séquentielle suivant l'ordre chronologique comme un processus de décision markovien (MDP) à états discrets mais dont les transitions entre les états sont continues et multidimensionnelles (les célérités). L'approche vise à sélectionner, pour le premier état discret du RRBh, une action (célérité) parmi l'ensemble des possibles, de manière à satisfaire les contraintes propres à cet état. Ce choix conditionne le point d'entrée dans l'état suivant, où le processus est reconduit. La succession de ces décisions permet de calculer une récompense cumulative, que l'on cherche à maximiser. Plus grand est le nombre d'actions identifiées respectant les contraintes biologiques, meilleure est la solution. L'intérêt est de construire et évaluer progressivement la solution, célérité après célérité.

Pour cela, nous avons adopté une méthode d'apprentissage par renforcement telle que la recherche arborescente Monte-Carlo (*Monte-Carlo tree search*, MCTS, figure 13) mise en lumière quand cette technique, combinée à de l'apprentissage profond, a battu le champion du monde de Go [25]. Le

principe consiste à construire progressivement un arbre de recherche dont les nœuds représentent les états et les arêtes les actions entreprises pour passer d'un état à un autre; la récompense est obtenue en appliquant des simulations (figure 13.d). Les articles [6] et [26] présentent en détail la technique ainsi que de nombreuses variantes.

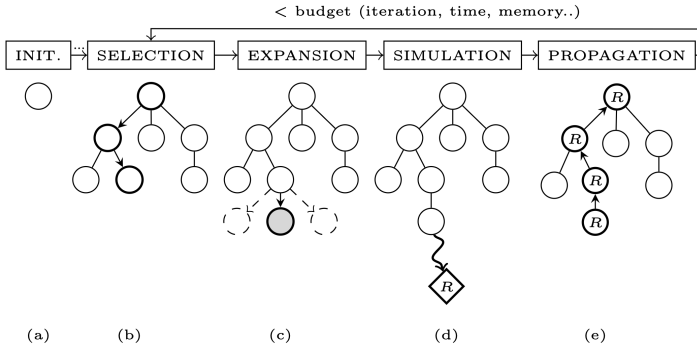


Fig. 13. Principes de la recherche arborescente Monte Carlo (MCTS).

Une des difficultés majeures pour utiliser MCTS est la stratégie à adopter dans la phase de sélection (figure 13.b) où il faut choisir entre un nœud qui est prometteur (exploitation) et un nœud pour lequel peu de simulations ont été réalisées (exploration). Par défaut, MCTS utilise les limites supérieures de confiance appliquées aux arbres (*upper confidence bounds for trees*, UCT) mais l'heuristique *Rapid action value estimation* (RAVE) s'est rapidement imposée à la fois dans le domaine discret et le domaine continu (cRAVE); elle assure un partage de connaissances entre des nœuds apparentés, ce qui permet d'obtenir une estimation rapide mais biaisée des valeurs d'action. Ensuite, une généralisation de RAVE (GRAVE) a été proposée pour recueillir des estimations plus précises près des feuilles de l'arbre mais uniquement dans le cas discret [7]. Ainsi, notre première contribution dans [22] a été d'adapter GRAVE au domaine continu (cGRAVE). Ensuite, nous avons suggéré de décomposer une action multidimensionnelle, une célérité qui est définie sur plusieurs dimensions, en plusieurs actions unidimensionnelles (AD), ce qui a pour conséquence d'augmenter la hauteur de l'arbre. La dernière contribution consiste à utiliser les contraintes pour réduire *a priori* l'espace des possibilités pour les actions (*constraint-based selective policy*, CSP). Nous avons comparé les performances de ces différentes approches (figure 14) en testant à chaque fois l'apport de chaque contribution (CSP, AD ou cGRAVE) à la technique de l'état de l'art (cRAVE).

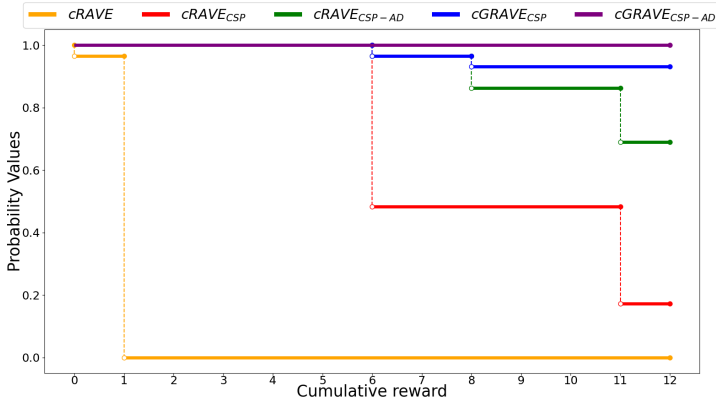


Fig. 14. Fonctions de répartition de plusieurs stratégies d'apprentissage par renforcement continu sur la modélisation d'un RRBh à 5 gènes sur 30 exécutions. Plus le gain est grand, avec un maximum à 12, meilleure est la solution.

On remarque que cRAVE combinée ou non à une stratégie de gestion de contraintes (cRAVE_{CSP}) n'arrive pas à obtenir de solutions comme pour les problèmes de satisfaction de contraintes continues même si la stratégie CSP améliore grandement le gain obtenu, passant de 1 à 6 dans la quasi-totalité des cas. Par rapport à cRAVE_{CSP}, la stratégie de décomposition AD (cRAVE_{CSP-AD}) de faire mieux que tripler la probabilité d'obtenir des solutions viables, passant de 0.2 à presque 0.7. De plus, notre adaptation de GRAVE au domaine continu (cGRAVE_{CSP}) atteint une probabilité proche de 1 d'obtenir une solution viable comparée à cRAVE_{CSP}. C'est seulement la combinaison des trois propositions (cGRAVE_{CSP-AD}) qui permet d'obtenir des échantillons respectant toutes les contraintes à coup sûr (probabilité égale à 1).

Multiples solutions ?

Comme pour les métaheuristiques à base de population, la grande majorité des variantes de l'apprentissage par renforcement basées sur MCTS ne permettent d'obtenir qu'une seule séquence optimale d'actions à réaliser. Afin de proposer un ensemble diversifié de solutions, nous avons proposé trois contributions différentes permettant d'obtenir plusieurs séquences optimales dans le domaine continu [21] en se basant sur les récents travaux de [4] qui formalisent parfaitement la problématique dans le domaine discret (*diverse multi-armed bandit policies*, DMAB). Nous introduisons deux stratégies d'inhibition différentes (*lock* et *diverse progressive widening*, DivPW) renforçant la diversité pendant la construction de l'arbre de recherche ainsi qu'une troisième (*diverse planning with MO-MCTS*, DPMO) qui ajoute un

deuxième objectif de diversité à une variante multiobjectif de MCTS [28]. Nous avons testé ces stratégies sur notre problème d'identification de célérités. Nous pouvons affirmer que sur un RRBh à 3 gènes (figure 15, à gauche) DivPW surpasse largement les autres stratégies testées. DPMO et Lock obtiennent de meilleures performances en matière de diversité que DMAB, mais leurs performances ne sont pas statistiquement différentes. Sur un RRBh à 5 gènes (figure 15, à droite), la stratégie de référence DMAB est également à la traîne par rapport aux stratégies proposées. Cependant, DivPW est surpassée par DPMO. Dans l'ensemble, cela démontre l'intérêt des nouvelles stratégies proposées.

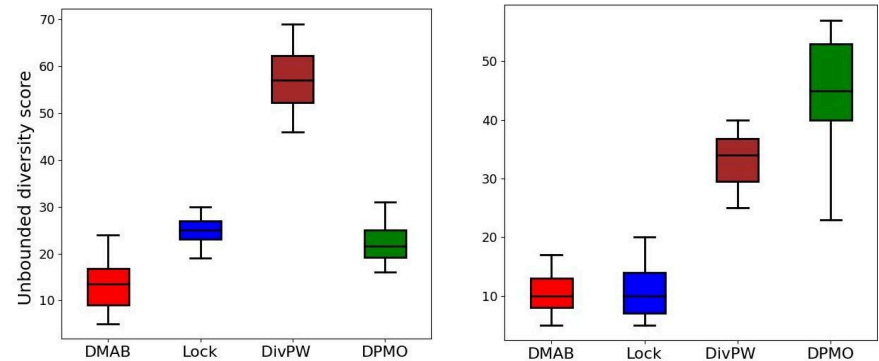


Fig. 15. Boîtes à moustaches du score de diversité moyen de nos stratégies par rapport à DMAB sur un RRBh à 3 (à gauche) et 5 gènes (à droite).

Conclusion et perspectives

Dans ces travaux, nous démontrons l'intérêt de l'évolution artificielle, par le biais des métaheuristiques à base de population, et de l'apprentissage par renforcement, via une recherche arborescente Monte-Carlo, pour résoudre la paramétrisation de réseaux de régulation biologique (RRB) par rapport aux solveurs de contraintes continues (cCSP). Contrairement à de nombreux travaux de recherche qui infèrent des RRB, nous adaptons les deux approches pour construire des RRB discrets et hybrides plus finement. De plus, nous proposons, dans les deux cas, des stratégies qui autorisent plusieurs paramétrisations optimales, intéressantes pour les biologistes. Dans le cas de l'évolution artificielle, nous mettons même en exergue certaines lacunes des fonctions de tests de la compétition CEC'2020 [17] sur l'optimisation multimodale qui considère l'identification d'un seul optimum

par bassin d'attraction alors que notre cadre d'étude exhibe un cas où il y en a une infinité. Dans le cas de l'apprentissage par renforcement, nous proposons même une adaptation d'une récente version de MCTS (GRAVE) au domaine du continu ainsi que plusieurs contributions pour obtenir plusieurs séquences optimales diversifiées (Lock, DPMO, DivPW).

On pourrait être tenté de comparer les métaheuristiques à MCTS sur les mêmes problèmes (RRBh à 2, 3 puis 5 gènes) mais cela ne nous semble pas équitable dans la mesure où une trajectoire complète est évaluée pour l'évolution artificielle alors qu'elle est construite de manière incrémentale pour MCTS. De plus, dans le cas de MCTS, les contraintes biologiques sont utilisées pour réduire l'ensemble des valeurs de la célérité dans chaque espace discret. Une démarche similaire pour les métaheuristiques serait de construire des opérateurs génétiques (croisement, mutation) capables de produire uniquement des trajectoires viables qui respecteraient les contraintes biologiques. Cela pourrait être une nouvelle piste de recherche. Une autre piste de recherche intéressante est de valider nos différentes contributions de MCTS (pour le domaine continu et pour les séquences diversifiées) sur des problèmes plus classiques de la communauté. Enfin, une dernière piste que nous envisageons consiste à adapter les techniques d'optimisation mixte [27] qui gèrent de manière plus appropriée le fait que nous manipulons simultanément des entiers et des réels.

Références

- [1] A. Ahrari, S. Elsayed, R. Sarker D. Essam, et C. C. Coello. 2022. Static and Dynamic Multimodal Optimization by Improved Covariance Matrix Self-Adaptation Evolution Strategy With Repelling Subpopulations. *IEEE Transactions on Evolutionary Computation*. <https://doi.org/10.1109/TEVC.2021.3117116>.
- [2] E. Alba et B. Dorronsoro. 2008. Introduction to cellular genetic algorithms. In *Cellular Genetic Algorithms*. https://doi.org/10.1007/978-0-387-77610-1_1.
- [3] J. Behaegel, J.-P. Comet, et M. Pelleau. 2018. Identification of Dynamic Parameters for Gene Networks. In *ICTAI*. <https://doi.org/10.1109/ICTAI.2018.00028>.
- [4] L. Benke, T. Miller, M. Papasimeon, et N. Lipovetzky. 2023. Diverse, Top-k, and Top-Quality Planning Over Simulators. <https://doi.org/10.3233/FAIA230275>.
- [5] G. Bernot, J.-P. Comet, Z. Khalis, A. Richard, et O. Roux. 2019. A genetically modified Hoare logic. *Theoretical Computer Science*. <https://doi.org/10.1016/j.tcs.2018.02.003>.
- [6] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, et S. Colton. 2012. A Survey of Monte Carlo Tree Search Methods. *IEEE Transactions on Computational Intelligence and AI in Games*. <https://doi.org/10.1109/TCIAIG.2012.2186810>.
- [7] Tristan Cazenave. 2015. Generalized rapid action value estimation. In (IJCAI). Consulté à l'adresse <https://dl.acm.org/doi/abs/10.5555/2832249.2832354>.

- [8] J. Dong, J. Li, et F. Wang. 2024. Deep Learning in Gene Regulatory Network Inference: A Survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. <https://doi.org/10.1109/TCBB.2024.3442536>.
- [9] J. M. Escorcia-Rodríguez, E. Gaytan-Nuñez, E. M. Hernandez-Benitez, A. Zorro-Aranda, M. A. Tello-Palencia, et J. A. Freyre-González. 2023. Improving gene regulatory network inference and assessment: The importance of using network structure. *Frontiers in Genetics*. <https://doi.org/10.3389/fgene.2023.1143382>.
- [10] G. Grataloup, J.-P. Comet, et G. Bernot. 2024. Improving Snoussi constraints in the Thomas framework for Gene Networks. In *Journées ouvertes de biologie, informatique et mathématiques*.
- [11] G. Grataloup, D. Pallez, J.-P. Comet, et G. Bernot. 2024. Single-objective constrained optimization for Gene Regulatory Networks Modeling. In *EA*.
- [12] J. Gurrola-Ramos, A. Hernández-Aguirre, et O. Dalmau-Cedeño. 2020. COLSHADE for Real-World Single-Objective Constrained optimization Problems. In (CEC). <https://doi.org/10.1109/CEC48606.2020.9185583>.
- [13] V. A. Huynh-Thu et G. Sanguinetti. 2019. Gene Regulatory Network Inference: An Introductory Survey. In *Gene Regulatory Networks: Methods and Protocols*. https://doi.org/10.1007/978-1-4939-8882-2_1.
- [14] M. Kronfeld et A. Zell. 2010. Towards scalability in niching methods. In (CEC). <https://doi.org/10.1109/CEC.2010.5585916>.
- [15] A. Kumar, S. Das, et I. Zelinka. 2020. A self-adaptive spherical search algorithm for real-world constrained optimization problems. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion (GECCO)*. <https://doi.org/10.1145/3377929.3398186>.
- [16] A. Kumar, S. Das, et I. Zelinka. 2020. A modified covariance matrix adaptation evolution strategy for real-world constrained optimization problems. In (GECCO). <https://doi.org/10.1145/3377929.3398185>.
- [17] A. Kumar, Guohua Wu, Mostafa Z. Ali, Rammohan Mallipeddi, Ponnuthurai Nagaratnam Suganthan, et Swagatam Das. 2020. A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. *Swarm and Evolutionary Computation*. <https://doi.org/10.1016/j.swevo.2020.100693>.
- [18] X. Li, A. Engelbrecht, et M. G. Epitropakis. 2013. Benchmark Functions for CEC2013 Special Session and Competition on Niching Methods for Multimodal Function Optimization. Consulté à l'adresse <https://titan.csit.rmit.edu.au/~e46507/cec13-niching/competition>.
- [19] R. Michelucci, V. Callegari, J.-P. Comet, et D. Pallez. 2024. Cellular Genetic Algorithms for identifying variables in hybrid Gene Regulatory Networks. In *EvoAPPS*. https://doi.org/10.1007/978-3-031-56852-7_9.
- [20] R. Michelucci, J.-P. Comet, et D. Pallez. 2022. Evolutionary continuous optimization of hybrid Gene Regulatory Networks. In *EA*. https://doi.org/10.1007/978-3-031-42616-2_12.
- [21] R. Michelucci, J.-P. Comet, et D. Pallez. 2024. Comparing diverse planning strategies with continuous MCTS applied to hGRN. In *ICTAI*. <https://doi.org/10.1109/ICTAI62512.2024.00018>.
- [22] R. Michelucci, D. Pallez, T. Cazenave, et J.-P. Comet. 2024. Improving Continuous Monte Carlo Tree Search for Identifying Parameters in Hybrid GRN. In *PPSN*. https://doi.org/10.1007/978-3-031-70085-9_20.
- [23] R. Michelucci, D. Pallez, et J.-P. Comet. 2025. Parametrisation of hybrid Gene Regulatory Networks using Artificial Evolution and Reinforcement Learning. In *ROADEF*. Consulté à l'adresse <https://inria.hal.science/hal-04980785>.

- [24] H. Nguyen, D. Tran, B. Tran, B. Pehlivan, et T. Nguyen. 2020. A comprehensive survey of regulatory network inference methods using single cell RNA sequencing data. *Briefings in Bioinformatics*. <https://doi.org/10.1093/bib/bbaa190>.
- [25] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. Driessche, T. Graepel, et D. Hassabis. 2017. Mastering the Game of Go without Human Knowledge. *Nature*. <https://doi.org/10.1038/nature24270>.
- [26] Maciej Świechowski, Konrad Godlewski, Bartosz Sawicki, et Jacek Mańdziuk. 2023. Monte Carlo Tree Search: A Review of Recent Modifications and Applications. *Artificial Intelligence Review*. <https://doi.org/10.1007/s10462-022-10228-y>.
- [27] El-Ghazali Talbi. 2024. Metaheuristics for variable-size mixed optimization problems: A unified taxonomy and survey. *Swarm and Evolutionary Computation*. <https://doi.org/10.1016/j.swevo.2024.101642>.
- [28] Weijia Wang et Michèle Sebag. 2012. Multi-objective Monte-Carlo Tree Search. In *Proceedings of the Asian Conference on Machine Learning*. Consulté à l'adresse <https://inria.hal.science/hal-00758379>.