

Algorithme génétique ou automate cellulaire : le cas d'une optimisation multicritère sous contraintes pour la conception d'une enveloppe

Genetic algorithm or cellular automaton: The case of a multi-criteria optimization under constraints for an envelope design

Claire Duclos-Prévet^{*,1}, François Guéna¹, et Mariano Efron²

¹MAP-MAACC, ENSA Paris la Villette, 75019 Paris, France

²Architecture Studio, 75012 Paris, France

Résumé. Il existe différentes méthodes génératives qui permettent une conception performancielle. Les algorithmes génétiques apparaissent comme la solution la plus populaire dans la littérature scientifique. Néanmoins, en pratique, les problèmes rencontrés nécessitent souvent une intégration de contraintes strictes que d'autres techniques génératives fonctionnant avec des règles, comme les automates cellulaires, sont plus aptes à intégrer. Dans cette étude, nous comparons l'efficacité de ces deux techniques appliquées à un même cas d'étude issu de la pratique professionnelle. Il s'agit de la conception d'une enveloppe pour une tour de bureaux à Madagascar qui intègre à la fois le confort thermique et la qualité de vue. Il ressort de cette expérience que l'approche la plus efficace est une méthode hybride qui utilise à la fois un automate cellulaire pour intégrer les contraintes, et un algorithme génétique conduisant l'optimisation multicritère.

Mots clés. Design génératif, conception intégrée, automate cellulaire, confort visuel et thermique, optimisation sous contraintes, algorithme génétique.

Abstract. Several generative methods can be suited for performance-based design. Genetic algorithms seem to be the most popular in the literature. However, in practice, designers often face issues that require the integration of strict constraints that are better handled by other generative techniques based on rules, such as cellular automata. In this study, we compare the efficiency of genetic algorithms and rule based generation methods in a case study drawn from professional practice consisting in the design of an envelope for an office tower in Madagascar that integrates both thermal comfort and view quality criteria. This experiment shows that

* Corresponding author: claire.duclos-prevet@paris-lavillette.archi.fr

the most efficient approach is a hybrid method that combines (i) a cellular automaton to treat the constraints, and (ii) a genetic algorithm driving the multicriteria optimization.

Keywords. Generative design, integred design, cellular automata, visual and thermal confort, constrained optimization, genetic algorithm.

1 Introduction

Un modèle génératif est à distinguer d'un simple modèle paramétrique. Caetano et ses co-auteurs [1] définissent le design génératif comme « un paradigme de conception qui emploie des descriptions algorithmiques qui sont plus autonomes que les modèles paramétriques ». Dans un modèle génératif, des instructions sont effectuées en boucle jusqu'à ce que le critère d'arrêt soit atteint. Bien que les deux types de modélisation requièrent des connaissances en programmation, le design génératif implique l'usage de la récursivité qui est difficile à implémenter avec les langages de programmation visuelle[†], tel que Grasshopper® (GH), le plus utilisé en pratique.

L'algorithme génétique (AG) est la technique générative la plus couramment employée en conception architecturale et performancielle [2]. Ces algorithmes d'optimisation inspirés de la théorie de l'évolution ont donné lieu à de nouvelles méthodes de conception qui permettent d'explorer une grande quantité de solutions architecturales. Ils ont été expérimentés sur de nombreux cas d'études, principalement théoriques [3]. Ces méthodes partagent toutes le même fonctionnement qui consiste à élaborer un modèle paramétrique capable de générer un espace de solutions à partir d'un ensemble de paramètres dont on peut faire varier les valeurs. Ces paramètres constituent le génome des solutions et le mécanisme d'optimisation consiste à retrouver les combinaisons des valeurs des paramètres permettant de générer les meilleures solutions vis-à-vis d'une ou plusieurs fonctions d'optimisation. A cette fin, le modèle paramétrique peut être relié à des outils de simulation environnementale dont les résultats servent au processus d'évolution de l'algorithme génétique.

Toutefois, ces méthodes ont des limites. En particulier, il est difficile d'intégrer des règles de conception ou des contraintes strictes avec de tels modèles. D'autres techniques génératives permettent de faire de la génération de façade comme les grammaires de formes (GF) [4], les modèles à base d'agents (MBA) [5], les automates cellulaires (AC) [6] et plus récemment des algorithmes de Deep Learning, Generative Adversal Network (GAN) [7]. Cet article a pour objectif d'apporter une nouvelle contribution sur l'usage des techniques génératives en comparant AG et AC sur la base du même cas d'étude issu d'un projet réel et en proposant une méthode hybride qui mixte les deux approches.

[†] En programmation visuelle, les données se propagent généralement dans un graphe acyclique. Définir des cycles nécessite l'usage de plugin permettant le contrôle des itérations.

2 Techniques génératives pour la conception d'enveloppe

De nombreuses revues de littérature font état de l'usage d'algorithmes d'optimisation, principalement des algorithmes génétiques, pour la conception des enveloppes des bâtiments [8,9,10]. Cette technique est utilisée pour intégrer des critères de performance notamment pour optimiser le confort visuel et thermique [11]. L'exploration peut aussi bien porter sur le système de percement (allocation des fenêtres, ratio de surface vitrée), le système de protection solaire, ou encore la couverture [12]. Cette technique générative peut aussi être utilisée pour la conception de façades adaptatives [13,14].

Les techniques génératives ne sont pas nécessairement utilisées à des fins environnementales. Les GAN qui mettent en concurrence deux réseaux de neurones sont depuis peu utilisés pour générer des images 2D de façades, par exemple à partir d'un style architectural donné, comme le fait D. Newton avec des cathédrales gothiques [15]. Avant les GAN, les GF qui permettent de faire émerger des formes à partir de règles étaient utilisées pour ce type de procédures. Elles étaient à l'origine utilisées pour faire de l'analyse historique des styles avant de servir à la production architecturale ou comme outil d'aide à la conception [16]. Ainsi, les GF peuvent être utilisées pour générer automatiquement des variantes de façades à partir d'un modèle [17]. Certains utilisent les GF pour la conception de façades performantes [18], d'autres pour la conception de systèmes de protection solaire intelligents [19, 20, 21].

Un MBA est une autre technique qui permet de faire de l'optimisation. Un MBA est un processus qui permet de résoudre des systèmes complexes constitués de multiples entités en interaction appelées agents. En exécutant des règles en fonction d'informations locales, les agents peuvent atteindre un objectif global en coopérant entre eux et en fonction de leur environnement [4]. Dans la littérature, certains utilisent cette technique pour optimiser la panélisation d'un système de couverture [22], la performance énergétique d'une façade en intégrant les préférences des futurs usagers [23], la conception d'une enveloppe piézoélectrique [24], ou encore la couverture d'un stade de tennis [25]. S. Parasho et ses co-auteurs [6] ont mené une étude comparative entre AG et MBA, mais appliquée à deux cas d'études distincts [22].

Les AC sont des MBA particuliers dans lesquels les agents ne se déplacent pas spatialement, mais changent d'état. Ils disposent de caractéristiques distinctes et fonctionnent selon des règles très simples. Un AC [6] est constitué d'une grille d'une, deux ou trois dimensions (D), faite de « cellules ». Chaque cellule peut prendre un nombre d'états (K) préalablement définis. L'évolution de l'état d'une cellule impacte l'état des cellules voisines situées dans un rayon (R). A partir des paramètres D, K et R, on définit un automate cellulaire pour lequel il existe un nombre fini de règles applicables.

Les AC ont été utilisés pour générer automatiquement des façades. Cependant, la plupart des auteurs les utilisent pour leur propriété d'émergence et pour l'esthétique que ces méthodes sont capables de générer plutôt que pour intégrer des objectifs environnementaux. Ainsi, M. Zawidski propose en 2009 une façade adaptative à partir d'un AC où des films polarisants permettent de contrôler les apports solaires [26]. À plusieurs reprises, les AC ont été utilisés pour la conception de façade, notamment pour contrôler l'éclairage naturel [27, 28, 29].

Dans notre cas, la trame orthogonale de l'AC est parfaitement adaptée pour la conception de notre tour de bureaux, aussi nous avons défini un AC qui permet d'intégrer des contraintes et des objectifs environnementaux.

3 Optimisation sous contraintes avec des algorithmes génétiques

Si les méthodes d'exploration fondées sur l'usage d'algorithmes génétiques sont populaires auprès des chercheurs, elles le sont beaucoup moins dans la pratique [8], notamment car les problèmes rencontrés en agence peuvent être très contraints. Ces contraintes doivent être intégrées au processus d'optimisation pour générer des solutions pouvant être retenues par les architectes. Dans le cas de l'enveloppe, on observe deux types d'exploration. L'exploration peut porter sur un détail de l'enveloppe [30, 31, 32], ou sur l'ensemble de la façade [33, 34, 35]. Lorsque toute la façade doit être considérée, la combinatoire est souvent très grande et le nombre de contraintes à intégrer au processus d'optimisation peut être important.

Il existe deux grandes approches pour intégrer des contraintes dans un problème d'optimisation : (i) soit on utilise une méthode interne intégrée à l'algorithme comme l'outil « Constraint-NGSAlI » qui est intégré dans NSGAlI (Non dominated Sorting Genetic Algorithm II) [36], (ii) soit on utilise une méthode de gestion externe à l'algorithme [37].

Parmi ces méthodes externes, la plus populaire repose sur l'intégration de fonctions de pénalisation [38]. Ces fonctions dégradent le score des solutions ne satisfaisant pas les contraintes en y appliquant un malus, une pénalisation. Cette méthode est facilement répliquable mais peut être inefficace. En effet, si le sous-ensemble de solutions qui respectent les contraintes (la « région faisable ») est proportionnellement très étroit comparé à l'ensemble de solutions total, alors la recherche risque de stagner rendant les fonctions de pénalisation peu efficaces [39].

Une meilleure alternative repose sur des fonctions de réparation qui corrigent les mauvaises solutions [40], au lieu de les pénaliser. Cette technique impose de déterminer une heuristique de réparation spécifique au problème, ce qui n'est pas toujours possible dans le temps de conception d'un projet réel en agence. Il existe deux types d'algorithme de réparation : (i) l'approche Baldwinienne dans laquelle le génome utilisé lors du croisement génétique est celui des solutions originales ; (ii) et l'approche Lamarcienne où le génome des solutions originales est remplacé par celui des solutions réparées, pour tout ou partie de la population [41]. Pour les deux approches, seules les solutions réparées sont évaluées. Lors de la phase de sélection, les scores des solutions réparées sont utilisés pour trier et sélectionner les meilleurs individus avant la phase de croisement et de mutation.

4 Cas d'étude

4.1 Description du cas d'étude

Notre cas d'étude est issu de la pratique professionnelle de l'agence Architecture Studio. Il s'agit de la conception d'une enveloppe, en phase esquisse, pour une tour de bureaux située en climat tropical. Le design doit permettre de réduire les apports solaires tout au long de l'année et d'assurer le confort thermique en limitant les consommations en refroidissement. De plus, cette protection ne doit pas être atteinte au détriment du confort visuel. Des vues non obstruées vers l'extérieur doivent être conservées pour chaque bureau. Ce type de problèmes imposant un arbitrage entre confort thermique et qualité de vue a déjà été traité dans la littérature à l'aide d'algorithme génétique [42].

Inspirée par l'art vernaculaire de la région et la géographie locale, l'équipe d'architectes a proposé une enveloppe double peau constituée d'un système de protection solaire avec des éléments verticaux alternant des vides et des éléments opaques disposés de façon aléatoire. Soucieux d'optimiser la disposition de ces éléments en fonction de

l'ensoleillement, nous leur avons proposé d'épaissir la première peau en faisant varier la profondeur des bureaux, comme le montrent le plan et la coupe en Figure 1. La forme de la façade génère des ombres propres et peut s'autoprotéger des rayons du soleil sans protection supplémentaire qui impacterait négativement la qualité des vues vers l'extérieur.

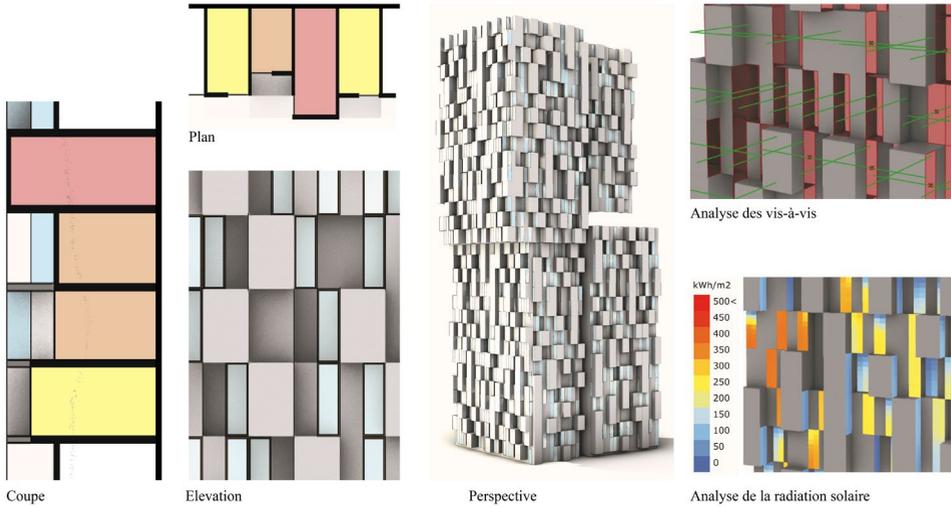


Fig. 1. Représentation graphique du concept de l'enveloppe.

Une trame adaptée aux bureaux est utilisée pour diviser l'enveloppe en modules rectangulaires. Comme l'illustre la Figure 2, chaque module peut être disposé en retrait (C), en saillie (A) ou rester en l'état (B). Ainsi, un module peut avoir trois formes. Par ailleurs, chaque module dispose d'un élément vitré. Cet élément peut être placé à deux ou quatre endroits différents, selon la forme du module. Au total, il y a 10 variantes de modules possibles.

Attribut 1 : trois formes possibles		Attribut 2 : deux ou quatre emplacements du vitrage possibles			
		Emplacement 0	Emplacement 1	Emplacement 2	Emplacement 3
forme A	plan	[Diagram]	[Diagram]	[Diagram]	[Diagram]
	élévation développée	[Diagram]	[Diagram]	[Diagram]	[Diagram]
forme B	plan	[Diagram]	[Diagram]	[Diagram]	[Diagram]
	élévation développée	[Diagram]	[Diagram]	[Diagram]	[Diagram]
forme C	plan	[Diagram]	[Diagram]	[Diagram]	[Diagram]
	élévation développée	[Diagram]	[Diagram]	[Diagram]	[Diagram]

Fig. 2. Représentations des 10 modules classés selon les attributs "forme" (A, B, C) et "emplacement du vitrage" (0, 1, 2, 3).

Pour optimiser la disposition des modules, plusieurs expérimentations ont été conduites à l'aide d'un modèle paramétrique Grasshopper® (GH), sur un morceau de façade composé de 150 modules, donc 150 variables, soit un ensemble de solutions de 1×10150 solutions au total. Les expérimentations ont été réalisées avec une façade exposée plein Sud, avec les masques solaires du contexte urbain et un fichier météo de la station de l'aéroport d'Antananarivo (<https://www.ladybug.tools/epwmap/>).

Les températures à Antananarivo étant élevées toute l'année et le climat très ensoleillé, il est essentiel pour assurer le confort thermique de réduire au maximum les apports solaires sur le vitrage quel que soit le jour de l'année. Le plugin Ladybug® [43] a été utilisé pour faire une évaluation annuelle de la radiation solaire directe et diffuse des parties vitrées.

Une méthode d'analyse des vis-à-vis entre les bureaux de l'immeuble a été programmée directement dans GH avec un système de lancers de rayons (voir Figure 1). Un rayon est lancé depuis le centre de chaque vitrage, il s'agit d'un vecteur normal au vitrage. Le paramètre à optimiser correspond au nombre de vitrages qui comptent un vis-à-vis sur un autre bureau de l'immeuble.

4.2 Les différentes méthodes d'exploration

Le modèle paramétrique ne permet pas d'intégrer des règles d'allocation des modules. De fait, des aberrations peuvent être générées, par exemple des modules privant des bureaux de lumière naturelle. Pour intégrer cette contrainte, nous avons d'abord essayé d'utiliser des méthodes génériques d'intégration des contraintes avec un algorithme génétique, à savoir Constraint-NSGAI [36], mais sans succès. Une méthode reposant sur des fonctions de pénalisation où le nombre d'aberrations générées est utilisé comme coefficient de violation des contraintes n'a pas donné de meilleurs résultats. Définir la contrainte comme un critère d'optimisation à part entière n'a pas, non plus, permis de résoudre le problème. Quelle que soit la méthode courante utilisée, la quantité de solutions faisables est trop faible, aucune de ces trois méthodes ne parvient à générer des solutions qui respectent les contraintes.

4.2.1. Automate cellulaire pour intégrer des contraintes

Face à cet échec des méthodes les plus communes, nous nous sommes tournés vers d'autres techniques génératives qui permettent de réduire le nombre d'aberrations tout en satisfaisant les exigences de confort thermique et visuel. La trame orthogonale régulière de cette façade constituée de modules pouvant s'interpréter comme des cellules capables de changer d'état permettait la mise en œuvre d'un AC à 2 dimensions.

Le principe consiste à parcourir l'ensemble des cellules et à déterminer leur état en fonction de l'état des cellules voisines. On produit ainsi une nouvelle génération de cellules qui sont de nouveau analysées. Ce processus est appliqué récursivement sur les populations jusqu'à ce que l'état des cellules cesse d'évoluer et que toutes les règles de voisinage sont respectées. Pour intégrer toutes les règles permettant d'éviter les aberrations, les cellules voisines sont sélectionnées en fonction de leurs positions et des règles choisies. En outre, à chaque cellule sont associés deux attributs : la forme, qui compte 3 états possibles (A, B ou C décrite en figure 2), et l'emplacement du vitrage, qui compte 2 ou 4 états possibles (0,1 ou 0,1,2,3 décrits en figure 2), selon la forme. Pour traiter les aberrations, trois types de règles ont été définis et sont présentés dans la Figure 3, où le point représente la cellule analysée. Prises ensembles, ces règles assurent que chaque module comporte un élément vitré.

Les trois types de règle sont les suivants :

- (i) La forme des cellules voisines impose une modification de la forme de la cellule.
- (ii) La forme des cellules voisines impose une modification de l'emplacement du vitrage de la cellule.
- (iii) La forme et l'emplacement du vitrage des cellules voisines imposent une modification du vitrage de la cellule.

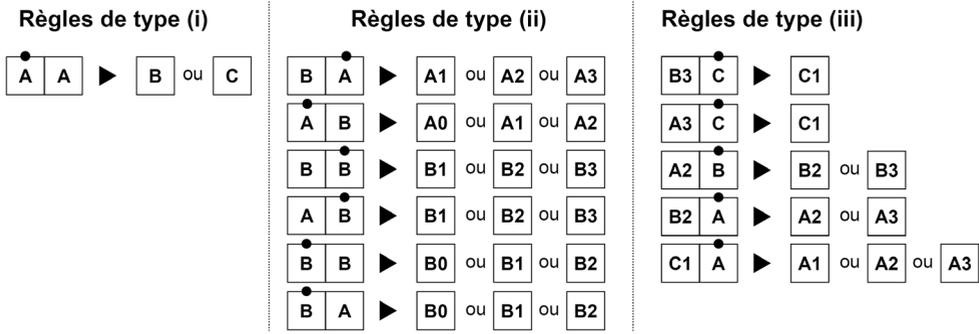


Fig. 3. Ensemble des règles de l'automate cellulaire permettant de supprimer les aberrations.

Afin de faciliter l'implémentation de toutes ces règles dans GH nous avons imaginé une méthode qui permet d'ajouter et de retirer des règles facilement à l'aide d'un unique composant pour chaque type de règle. Le fonctionnement du modèle GH pour les règles de type (ii) est présenté en exemple dans la Figure 4.

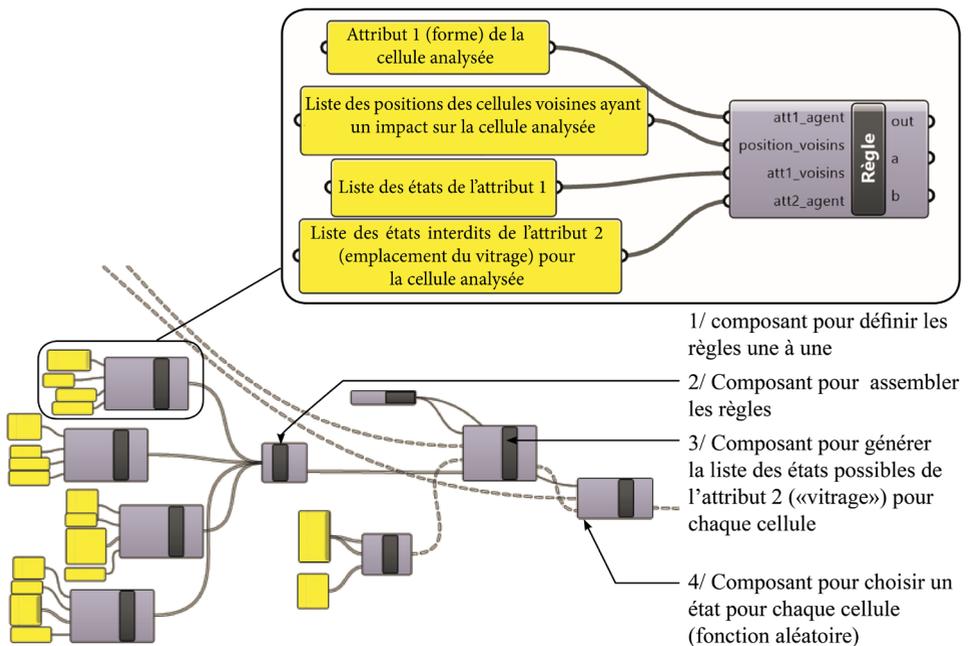


Fig. 4. Méthode d'implémentation des règles dans Grasshopper.

4.2.2. Optimisation avec un automate cellulaire

Aux règles permettant de supprimer les aberrations, nous avons ajouté des règles pour augmenter la performance du confort thermique et visuel des solutions générées. Il est relativement aisé d'établir des corrélations entre la géométrie locale et les performances de radiation solaire et de qualité de vue. Nous avons donc élaboré un algorithme dont le fonctionnement est présenté ci-dessous :

- 1) Si la cellule analysée crée suffisamment d'ombre pour ses voisines, alors ses attributs (forme et vitrage) restent inchangés et l'algorithme passe à la cellule suivante.
- 2) Si tel n'est pas le cas, et si le score de radiation de la cellule analysée est supérieur à un certain seuil ou si la cellule analysée est en vis-à-vis avec une cellule voisine, l'emplacement du vitrage de la cellule analysée est modifié.
- 3) Après plusieurs modifications du vitrage, si les critères ne sont toujours pas satisfaits, alors la forme de la cellule analysée est aussi modifiée.

Le modèle complet intègre le mécanisme de gestion des aberrations et celui de la poursuite des objectifs. Son fonctionnement est décrit dans la Figure 5 et peut être synthétisé comme suit. En premier lieu, une proposition originale est créée à partir du modèle paramétrique. Cette proposition est ensuite évaluée grâce aux outils de simulation avant que chaque module ne soit traité par l'algorithme appliquant les règles de confort. Enfin, la proposition est traitée par l'algorithme supprimant les aberrations. Lorsque la proposition ne contient plus d'aberrations, elle est considérée comme la première solution générée et servira de proposition pour constituer la seconde. Nous avons généré 1 500 solutions faisables avec cet algorithme. L'automate cellulaire a été entièrement codé en Python directement sur GH.

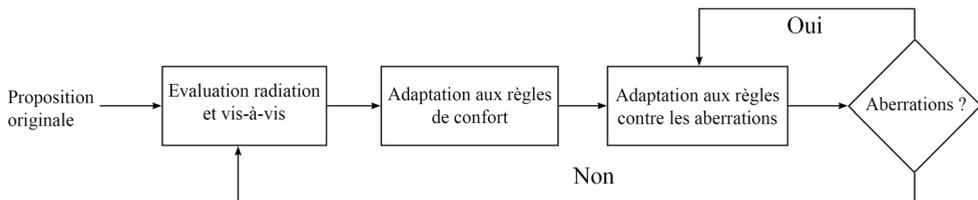


Fig. 5. Fonctionnement de l'automate cellulaire avec toutes les règles.

4.2.3. Optimisation avec une méthode hybride

Enfin, nous avons testé une approche mixte, parfois appelée méthode hybride [41] dont le fonctionnement est décrit en Figure 6. Cette méthode utilise à la fois un algorithme génétique multicritère (NSGAI) et un automate cellulaire qui sert de fonction de réparation. En somme, l'algorithme de suppression des aberrations a été utilisé comme fonction de réparation pour modifier les solutions créées par le modèle paramétrique avant qu'elles ne soient évaluées. Les scores sont ensuite transmis à l'algorithme génétique, ainsi que les nouveaux paramètres (le génome) des solutions réparées. Deux approches ont été testées : l'approche Balwinienne et l'approche Lamarckienne (voir section 2.1). Le taux de remplacement utilisé dans le second cas est de 5 %. Il s'agit du taux préconisé pour les problèmes d'optimisation combinatoire [44].

L'algorithme évolutionnaire à multi-objectifs NSGA-2 [36] a été utilisé. La librairie pymoo (<https://pymoo.org/>) a été utilisée pour implémenter cet algorithme sur GH en Python. Les calculs sont lancés pour 30 générations de 50 individus chacune. Une fois les

simulations terminées, les résultats ont été exportés et assemblés dans un fichier csv et analysés avec le logiciel R.

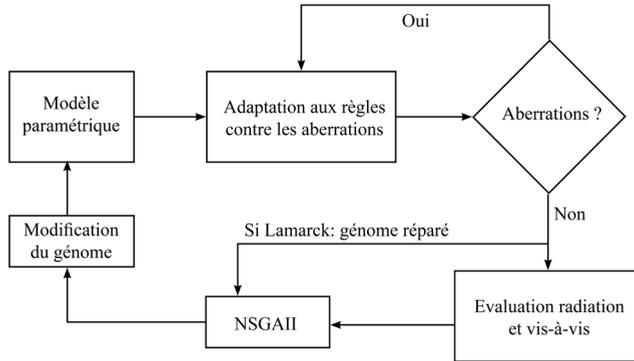


Fig. 6. Fonctionnement de la méthode hybride.

5 Résultats

Afin de vérifier l’efficacité des deux méthodes, nous avons utilisé un groupe contrôle de 1 500 solutions qui ont préalablement été créées à l’aide du modèle paramétrique, avant d’être « réparées » avec la fonction de réparation qui permet la suppression des aberrations. Les résultats présentés dans la Figure 7 montrent que la méthode avec un AC permet de générer des solutions optimisées. Cependant, dans notre cas précis, c’est la méthode hybride qui utilise à la fois un AG et un AC servant de fonction de réparation qui est la plus performante. Elle l’est d’autant plus avec une approche Lamarckienne où 5 % des génomes originaux sont remplacés par les génomes de solutions réparées.

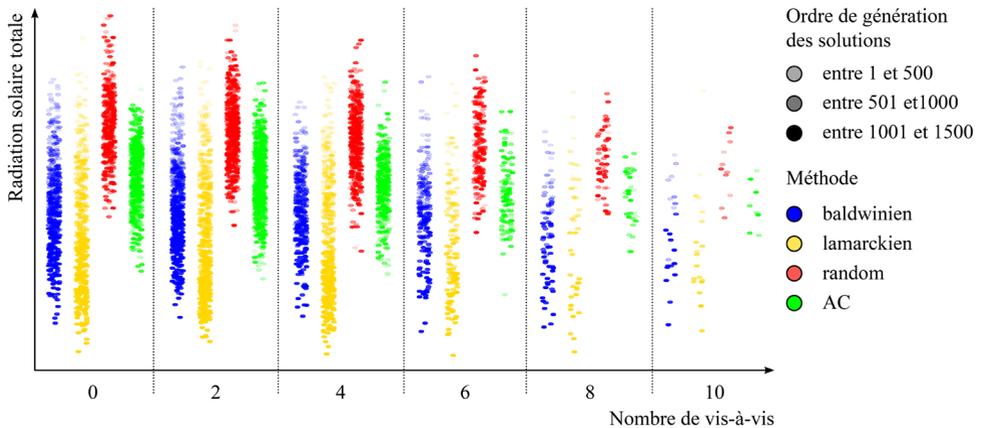


Fig. 7. Graphique comparant les résultats avec différentes méthodes d’optimisation.

6 Conclusions

Les résultats de cette étude comparative confirment que les méthodes d’intégration des contraintes génériques pour les AG, comme celles reposant sur une fonction de pénalisation, ne sont pas adaptées pour des problèmes très contraints. Lorsqu’il est possible de déterminer des règles permettant d’optimiser les critères, un AC peut être utilisé pour

intégrer des règles simples qui permettent à la fois de gérer des contraintes et des critères environnementaux. Aussi, le cas particulier de la façade, élément architectural dont le design est souvent tramé, est un cadre d'application très adapté pour un AC 2D. De manière intéressante, il apparaît que des techniques génératives à base de règles peuvent être utilisées efficacement comme fonctions de réparation afin d'intégrer des contraintes dans un processus d'optimisation évolutionnaire. D'ailleurs, c'est finalement une méthode hybride associant à la fois AG et AC que nous recommandons pour le problème traité.

Finalement, il semble que les techniques génératives tel que l'AC puissent constituer un apport intéressant au design performant. Cependant, il reste compliqué d'implémenter de tels systèmes en utilisant des règles personnalisées sans connaissance en programmation. Ainsi, des outils dédiés de programmation visuelle restent à imaginer pour démocratiser l'usage de ces techniques génératives dans les agences d'architecture. Si nous avons évoqué dans cette étude une piste pour les automates cellulaires, d'autres expérimentations sont à conduire avec des modèles à base d'agents plus complexes, notamment avec des agents sont en mouvement.

Références

1. I. Caetano, L. Santos, A. Leitão, Computational design in architecture: Defining parametric, generative, and algorithmic design. *Frontiers of Architectural Research*, **9(2)**, 287-300 (2020)
2. V. Singh, N. Gu, Towards an integrated generative design framework. *Design studies*, **33(2)**, 185-207 (2012)
3. S. Li, L. Liu, C. Peng, A review of performance-oriented architectural design and optimization in the context of sustainability: dividends and challenges. *Sustainability*, **12(4)**, 1427 (2020)
4. G. Stiny, Introduction to shape and shape grammars. *Environment and planning B: planning and design*, **7(3)**, 343-351 (1980)
5. C.M. Macal, Everything you need to know about agent-based modelling and simulation. *Journal of Simulation*, **10(2)**, 144-156 (2016)
6. J. Von Neumann, A.W. Burks, Theory of self-reproducing automata. *IEEE Transactions on Neural Networks*, **5(1)**, 3-14 (1966)
7. A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015)
8. S. Li, L. Liu, C. Peng, A Review of Performance-Oriented Architectural Design and Optimization in the Context of Sustainability: Dividends and Challenges. *Sustainability* **12(4)**, 1427 (2020)
9. Y. Huang, J.L. Niu, Optimal building envelope design based on simulated performance: History, current status and new potentials. *Energy and Buildings* **117**, 387-398 (2016)
10. A.T. Nguyen, S. Reiter, P. Rigo, A review on simulation-based optimization methods applied to building performance analysis. *Applied Energy* **113**, 1043-1058 (2014)
11. A. Eltaweel, S.U. Yuehong, Parametric design and daylighting: A literature review. *Renewable and Sustainable Energy Reviews* **73**, 1086-1103 (2017)
12. S. Zhao, E.de Angelis, Performance-based Generative Architecture Design: A Review on Design Problem Formulation and Software Utilization. *Journal of Integrated Design and Process Science* **22(3)**, 55-76 (2018)

13. A. Tabadkani, M.V. Shoubi, F. Soflaei, S. Banihashemi, Integrated parametric design of adaptive facades for user's visual comfort. *Automation in Construction* **106**, 102857 (2019)
14. P. Jayathissa, S. Caranovic, J. Hofer, Z. Nagy, A. Schlueter, Performative design environment for kinetic photovoltaic architecture. *Automation in Construction* **93**, 339-347 (2018)
15. D. Newton, Generative deep learning in architectural design. *Technology, Architecture+ Design*, **3(2)**, 176-189 (2019)
16. B. Tepavčević, V. Stojaković, Shape grammar in contemporary architectural theory and design. *Facta Universitatis-series: Architecture and Civil Engineering*, **10(2)**, 169-178 (2012)
17. F. Bao, M. Schwarz, P. Wonka, Procedural facade variations from a single layout. *ACM Transactions on Graphics (TOG)*, **32(1)**, 1-13 (2013)
18. N. Ashrafi, J. Pinto Duarte, A shape-grammar for double skin facades. *SharingofComputableKnowledge!*, 471 (2017)
19. S.D. Kotsopoulos, F. Casalegno, G. Carra, W. Graybil, B. Hsiung, A visual-performative language of façade patterns for the connected sustainable home. In *Proceedings of the 2012 Symposium on Simulation for Architecture and Urban Design* (p. 5). Society for Computer Simulation International. (2012)
20. R. Tomasowa, F.U. Sjarifudin, F. U., Adaptive Façade: Variant-Finding using Shape Grammar. In *IOP Conference Series: Earth and Environmental Science*, **109(1)**, 012039 (2017)
21. B. Ceranic, T. Nguyen, C. Callaghan, Shape grammar and kinetic façade shading systems: a novel approach to climate adaptive building design with a real time performance evaluation. *International Conference Geomapplica* (2018)
22. S. Parascho, M. Baur, J. Knippers, A. Menges, Design Tools for Integrative Planning. *eCAADe 31th Conference*, **2**, 18, 237-246 (2013)
23. D.J. Gerber, E. Pantazis, A. Wang, A multi-agent approach for performance based architecture: design exploring geometry, user, and environmental agencies in façades. *Automation in construction*, **76**, 45-58 (2017)
24. A.H. Zarrabi, M. Azarbayjani, M. Tavakoli, Generative Design Tool: Integrated Approach toward Development of Piezoelectric Façade System.
25. E. Pantazis, D.J. Gerber, Behavioral Form Finding: A Multi Agent Systems Framework for Environmental Aware Form Finding of Shell Structures. In *Design Modelling Symposium Berlin*, Springer, Cham, 146-158 (2019, September)
26. M. Zawidzki, Implementing cellular automata for dynamically shading a building facade. *Complex Systems*, **18(3)**, 287 (2009)
27. H. Alshoubaki, T. Rawashdeh, K. Alomari, R. Hammad, Innovative design strategy to develop facades opening for a commercial building in Amman, Jordan. *International Journal of Applied Engineering Research*, **11(7)**, 5288-5292 (2016)
28. J. Kim, Adaptive façade design for the daylighting performance in an office building: the investigation of an opening design strategy with cellular automata. *International Journal of Low-Carbon Technologies*, **10(3)**, 313-320 (2013)
29. F. Fathy, Y. Mansour, H. Sabry, S. Abdelmohsen, A. Wagdy, (2015). Cellular Automata for Efficient Daylighting Performance: Optimized Façade Treatment. In *Proceedings of the 14th Conference of the International Building Performance Simulation Association (IBPSA)*, 2705-2711 (2015)

30. F. Fathy, H.A. Fareed, Performance-driven Façade Design Using an Evolutionary Multi-Objective Optimization Approach. In *International Conference for Sustainable Design of the Built Environment-SDBE London*, 217 (2017)
31. A. Tabadkani, M.V. Shoubi, F. Soflaei, S. Banihashemi, Integrated parametric design of adaptive facades for user's visual comfort. *Automation in Construction*, **106**, 102857 (2019)
32. X. Shen, Environmental parametric multi-objective optimization for high performance facade design. *CAADRIA 23rd Conference*, **2**, 103-112 (2018)
33. Y.K.Yi, Building facade multi-objective optimization for daylight and aesthetical perception. *Building and Environment*, **156**, 178-190 (2019)
34. I. Chatzikonstantinou, M. Turrin, C. Cubukcuoglu, A. Kirimat, S. Sariyildiz, A Comprehensive Optimization Approach for Modular Facades: The Case of PULSE Sunshading. *International Journal of Design Sciences & Technology*, **23(2)**, 159-185 (2019)
35. Negendahl, T.R. Nielsen. «Building energy optimization in the early design stages: A simplified method.» *Energy and Buildings*, **105**, 88-99 (2015)
36. K. Deb, A. Pratap, S. Agarwal, T.A.M.T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, **6(2)**, 182-197 (2002)
37. C. Duclos-Prévet, F. Guéna, M. Efron, Constrained Multi-Criteria Optimization for Integrated Design in Professional Practice. In *SIGraDi 2021 Designing possibilities*, 29-40 (2021)
38. Z. Michalewicz, M. Schoenauer, Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary computation*, **4(1)**, 1-32 (1996)
39. C.A.C. Coello, Constraint-handling techniques used with evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 675-701 (2017)
40. C.A.C. Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer methods in applied mechanics and engineering*, **191(11-12)**, 1245-1287 (2002)
41. S. Salcedo-Sanz, A survey of repair methods used as constraint handling techniques in evolutionary algorithms. *Computer science review*, **3(3)**, 175-192 (2009)
42. M. Showkatbakhsh, S. Kaviani, Homeostatic generative design process: Emergence of the adaptive architectural form and skin to excessive solar radiation. *International Journal of Architectural Computing*, 1478077120951947 (2020)
43. M.S. Roudsari, M. Pak, A. Smith, Ladybug: a parametric environmental plugin for grasshopper to help designers create an environmentally-conscious design. In *Proceedings of the 13th international IBPSA conference held in Lyon, France Aug* , 3128-3135 (2013, August)
44. D. Orvosh, Shall we repair? Genetic algorithms, combinatorial optimization, and feasibility constraints. In *Proc. 5th International Conf. on Gas*, 650 (1993)