

## Résumé :

Ce mémoire traite de la question de la mobilité piétonne urbaine, telle qu'elle a été transposée dans les outils numériques comme aspect quantifiable de l'environnement urbain. Nous abordons, pour cela, les théories qui ont permis l'émergence du concept de la marchabilité et puis sa transposition comme une mesure définie par les propriétés de la configuration urbaine.

A travers l'usage des algorithmes d'optimisation, intégrés dans la plateforme paramétrique, nous "déconstruisons" cette mesure en y intégrant d'autres critères pour vérifier leur influence sur la relation mathématique établie actuellement et donc sur la façon dont le problème à optimiser est formulé.

Les expériences d'optimisation servent, dans un premier temps, à définir des scénarios différents selon des critères introduits et à vérifier leur influence sur la mesure de la mobilité, selon la règle de calcul actuelle. Dans un second temps, les résultats obtenus nourrissent à leur tour des questionnements autour du processus de l'optimisation et de la nécessité ou non de l'approche du multi-objectif pour des problèmes de conception simples.

## Mots-clés :

Optimisation informatique, Algorithme bio-inspirés, Métaheuristiques, Marchabilité, Multi-critères, Approche paramétrique.

# Optimisation de la marchabilité par les algorithmes bio-inspirés

Amira CHAGHAL

Sous la direction de :

Pr François GUÉNA

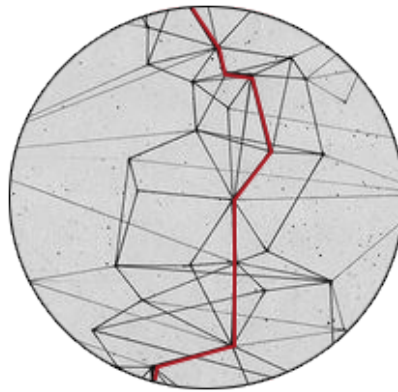
Pr Joaquim SILVESTRE

Pr Anne TÜSCHER DOKIC

École Nationale Supérieure d'Architecture de Paris La  
Villette

Séminaire Activités et Instrumentation de la conception

Mémoire de master



Optimisation de la marchabilité par les  
algorithmes bio-inspirés

Sous la direction de :

Pr François GUÉNA

Pr Joaquim SILVESTRE

Pr Anne TÜSCHER DOKIC

Rédigé par :

Amira CHAGHAL

2019/2021

## Remerciements

Je tiens à témoigner de ma profonde reconnaissance envers mes tuteurs académiques, Monsieur François GUÉNA, Monsieur Joaquim SILVESTRE et Madame Anne TÛSCHER DOKIC pour tout leurs enseignements, leur patience et leur conseils judicieux tout au long de ce master. Leur belles qualités d'enseignants m'ont permise de développer mon savoir-faire et mon savoir-être.

J'exprime également ma gratitude à tous les professeurs de mon cursus universitaire pour le savoir qu'ils m'ont transmis ainsi qu'à mes amis, de mes deux écoles, qui ont enrichi ce parcours.

Enfin, je remercie mes parents et mon frère de m'avoir accompagnée sur cette voie avec autant de patience et de bienveillance.

## Résumé

Ce mémoire traite de la question de la mobilité piétonne urbaine, telle qu'elle a été transposée dans les outils numériques comme un aspect quantifiable de l'environnement urbain. Nous abordons, pour cela, les théories qui ont permis l'émergence du concept de la marchabilité et puis sa transformation comme un aspect quantifiable par les propriétés de la configuration urbaine.

A travers l'usage des algorithmes d'optimisation, intégrés dans la plateforme paramétrique, nous déconstruisons cette mesure en y intégrant d'autres critères pour vérifier leur influence sur la relation mathématique établie actuellement et donc sur la façon dont le problème à optimiser est formulé.

Les expériences d'optimisation servent, dans un premier temps, à construire des scénarios différents selon des critères introduits et à vérifier leur influence sur la mesure de la mobilité, selon la règle de calcul actuelle. Dans un second temps, les résultats obtenus nourrissent à leur tour des questionnements autour du processus de l'optimisation et de la nécessité ou non du multi-objectif pour des problèmes de conception simples.

**Mots-clés :** Optimisation informatique, algorithme bio-inspirés, méta-heuristiques, marchabilité, mobilité urbaine, multi-critères, approche paramétrique.

---

# Sommaire

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Introduction Générale . . . . .	3
1.2	Problématique et objectifs . . . . .	4
1.3	Hypothèses et Méthodes de travail . . . . .	6
1.4	Structure du mémoire . . . . .	8
1.5	Visuel du mémoire . . . . .	9
<b>2</b>	<b>Marchabilité : Théories et émergence d’une nouvelle mesure de la ville</b>	<b>11</b>
2.1	La marchabilité : Définition d’un concept émergent : . . . . .	11
2.2	Théories et Outils autour de la marchabilité : . . . . .	12
2.2.1	Le Transit Oriented Developpement : De l’optimisation du transport à celle de la marche : . . . . .	12
2.2.2	Une lecture systémique de l’espace urbain à travers la Space Syntax : . . . . .	13
2.3	Des théories analytiques à la mesure de la marchabilité . . . . .	24
2.3.1	Les indicateurs de marchabilité : Des métriques pour évaluer l’espace urbain : . . . . .	24
2.3.2	La Space Syntax Walkability :(SSW) : Modèle de substitution des indicateurs de Marchabilité . . . . .	30
2.4	Résumé et conclusion partielle . . . . .	32
<b>3</b>	<b>Walkscore : Des indicateurs théoriques à l’algorithme informatique :</b>	<b>33</b>
3.1	Outil de mesure automatisé : . . . . .	33
3.2	Mode opératoire de l’algorithme . . . . .	34
3.2.1	Echelle du calcul et « Heat Maps » : . . . . .	34
3.2.2	Les variables et la fonction de calcul : . . . . .	36

3.3	Limites de l'algorithme . . . . .	43
3.4	Résumé et conclusion partielle . . . . .	44
<b>4</b>	<b>Modélisation et Optimisation de la marchabilité dans l'outil paramétrique</b>	<b>45</b>
4.1	La démarche paramétrique : Nouveau paradigme de la conception . . . . .	45
4.1.1	L'outil paramétrique . . . . .	46
4.2	L'outil paramétrique pour la modélisation de l'urbain : . . . . .	48
4.2.1	L'extraction des données géographiques par l'outil paramétrique . . . . .	49
4.2.2	La modélisation et simulation du modèle de calcul . . . . .	51
4.2.3	L'optimisation dans l'outil paramétrique par les algorithmes bio-inspirés . . . . .	55
<b>5</b>	<b>Expériences et Analyse</b>	<b>75</b>
5.1	Méthodologie du protocole expérimental . . . . .	75
5.1.1	Que calculons-nous ? Le score de marchabilité . . . . .	76
5.1.2	Origine et limite physiques du calcul . . . . .	77
5.1.3	Définition des variables à introduire . . . . .	83
5.2	Application du protocole : Les expériences d'optimisation . . . . .	90
5.2.1	Choix du contexte de l'étude . . . . .	91
5.2.2	Optimisations mono-objectif selon les critères : . . . . .	92
5.2.3	Optimisations multi-objectif . . . . .	102
5.3	Conclusion partielle . . . . .	111
<b>6</b>	<b>Limites de l'étude et prospectives</b>	<b>113</b>
6.1	Limites et regard critique sur les expériences réalisées : . . . . .	113
6.1.1	Limites relatives à la nature des outils utilisés : . . . . .	113
6.1.2	Limite relative à la taille du problème formulé et au processus conceptuel : . . . . .	114
6.2	Les alternatives : vers d'autres modes de représentation : . . . . .	115
6.2.1	Représentation multi-variée d'un paysage de recherche abstrait multi-dimensionnel pour l'aide à la décision de l'architecte . . . . .	115
6.2.2	Une autre méthode de représenter la marchabilité pour une nouvelle formule de calcul du score : . . . . .	117
6.3	Résumé et conclusion partielle : . . . . .	119
<b>7</b>	<b>Conclusion générale</b>	<b>121</b>
	<b>Annexe A Configuration et simulation</b>	<b>123</b>
	<b>Annexe B Données Brutes</b>	<b>125</b>



# Partie introudctive





# Introduction

## 1.1 | Introduction Générale

Parmi les paradigmes émergents autour de la méthodologie conceptuelle, celui du "Performance Based Design" constitue une voie d'exploration et de réflexion de plus en plus empruntée [Oxman, 2009]. Cet engouement pour ce modèle de conception serait corrélé, à la fois, au concept du développement durable mais aussi avec la nouvelle culture numérique qui se l'approprie.

Pourtant, dans le processus conceptuel, il ne s'agit pas uniquement de concrétiser des solutions potentielles face à un problème formulé, mais aussi de le recarder et de réconcilier des objectifs conflictuels [Kolarevic, 2004, Yu et al, 2014]. Parfois, le design intuitif n'aboutit pas nécessairement à la solution optimale et, qui plus est, il est important de se poser la question de la pluralité des réponses qui pourraient être générées, d'où l'importance d'enrichir la phase exploratoire dans le processus conceptuel.

Le processus d'optimisation permet justement d'explorer des variantes, mais il n'arrive que comme une conclusion aux diverses phases conceptuelles et il s'agit, souvent, d'un travail relégué aux ingénieurs et aux bureaux d'études. Il existe donc une certaine forme d'isolement ou de fracture entre le travail de conception de l'architecte et les travaux d'optimisations réalisés par les autres corps d'états, et rares sont les fois où l'architecte intervient dans le processus de l'optimisation [Dissaux, 2018].

Cette fracture émerge du fait que l'optimisation, à l'origine, est un processus ancré dans le domaine des sciences mathématiques et de l'ingénierie. Un autre facteur accentuant cette brèche est que l'abondance des informations et paramètres à prendre en compte pour l'opti-

misation. Ces derniers n'étaient pas, auparavant, possibles à modéliser sur les premiers outils de CAO. C'est surtout le développement des outils paramétriques, que nous présenterons davantage dans le chapitre [4.1], qui a permis la "décomposition" d'un problème de conception spatiale. De cette décomposition a découlé la possibilité de l'enrichissement de l'exploration inhérente au processus conceptuel, évoqué ci-dessus.

Des architectes, comme David Rutten et Thomas Wortmann, ont notamment participé à cette hybridation entre l'optimisation informatique et l'outil paramétrique de l'architecte. Leur collaboration avec des mathématiciens et ingénieurs a permis de transposer les algorithmes d'optimisation vers des interfaces plus accessibles pour l'architecte en usant de ce qui est appelé « la programmation visuelle ». Ceci a permis le développement de l'ADO : Architectural Design Optimisation, une sous-discipline de l'optimisation relative à l'ingénierie.

Thomas Dissaux évoque Terzidis (2003) et Barbesian (2013) pour souligner cette transition qui permet justement à l'architecte de mettre à profit des algorithmes d'optimisation informatique dans son exploration du design optimal, sans pour autant avoir les connaissances d'un informaticien.<sup>1</sup>

Cette nouvelle méthode d'opérer a suscité un large engouement dans le domaine architecture et la littérature scientifique a vu proliférer un nombre de recherche et d'expérimentations avec ces algorithmes d'optimisation dans les outils relatifs à la conception. Cependant, comme évoqué ci-dessus, l'intérêt était ancré principalement dans le domaine architectural, d'où l'approfondissement de la discipline de l'ADO [Wortmann et Cichoka, 2018].

En revanche, peu d'expérimentations ont été effectuées à l'échelle urbaine avec la démarche paramétrique. Les travaux existants sur l'optimisation, à cette échelle, illustrent surtout une exploration de solutions optimales formelles du bâtiment face à des objectifs climatiques, étant donné que les données environnementales sont facilement modélisées dans l'outil paramétrique et que cela s'inscrit également dans les démarches du développement durable.

## 1.2 | Problématique et objectifs

Dans ce travail, nous nous proposons justement de nous intéresser à **l'optimisation à l'échelle urbaine et particulièrement à l'optimisation de la marchabilité**, aussi

---

1. Dissaux, T. dans son MFE : *Optimisation dans la conception architecturale*, 2018

---

désignée par "potentiel piétonnier urbain". Un chapitre est dédié à l'explication de ce concept [Se référer au chapitre 2] et dans cette partie, nous la définirons sommairement comme étant le degré d'accessibilité dans un espace urbain. Un paramètre également participant, à une conception durable de la ville, notamment, après 2007, où plus de la moitié de la population mondiale représentait une population urbaine.

L'accessibilité étudiée a fait l'objet de recherches, notamment au début des années 2000, qui ont permis de modéliser une mesure ou une quantification de la marchabilité, sur la base de critères relatifs à l'environnement bâti et à la configuration urbaine. Nous estimons néanmoins que ces critères restent déficients, ou du moins pourraient être affinés, et que la relation qui construit l'équation de la marchabilité, sur la base de ces critères, n'est donc pas toujours valide ou possible à prédire sans simulation informatique. Le fait que cette mesure soit jugée déficiente, dans ce travail, est particulièrement dû à la réduction des modèles simulés et aux approches ou sources sur lesquels ces modèles se basent pour le calcul.

Nous ajouterons qu'actuellement la marchabilité est optimisée par rapport à une référence absolue or la complexité du système urbain ne pourrait être réduite à un seul critère, et ce, quelle que soit la thématique traitée dans l'optimisation. Rittel et Webber (1973) parlent des dilemmes dans la théorie de planification en désignant les problèmes architecturaux et urbains comme « wicked problems »<sup>2</sup>. Par conséquent, capturer l'entière et toutes les dimensions d'un problème de planification n'est évidemment pas faisable, mais l'optimisation multi-objectif représenterait une tentative d'approche de la complexité des problèmes de conception urbaine.[Burry,2012]

Il s'agit donc aussi de se questionner, à travers une série d'expériences, sur principalement deux éléments globaux (détaillés dans la section : 1.3 Hypothèses et Méthodes de travail) : Le premier est la formulation du problème actuel de la marchabilité optimale. Le second est la nécessité du multi-objectif dans les problèmes de conception, même avec des formulations simples. Cet élément de questionnement est relativement basé sur les discours construits autour de l'optimisation multi-objectif dans les problèmes architecturaux en premier lieu.<sup>3</sup>

---

2. Rittel, H et Webber, M dans *Dilemmas in a general theory of planning*, 1973

3. Wortmann, T dans *Does architectural design optimization require multiple objectives? A critical analysis*, 2020 et Luke, S dans *Essential of Metaheuristics*, 2018 en parlant de l'approche naïve (voir hypothèse 2).

## 1.3 | Hypothèses et Méthodes de travail

1

**Hypothèse principale :** L'équation de base de la marchabilité est déficiente, car elle omettrait des critères dans le calcul. Si les critères actuels sont suffisants, alors même en introduisant d'autres, le résultat serait le même puisque les critères introduits n'auraient aucune influence et si au contraire, nous avons d'autres résultats, cela validera l'hypothèse.

**Hypothèses secondaires :**

-La formulation du problème d'optimisation de la marchabilité ,actuellement, est erronée (Distance minimale – Marchabilité optimale).

-Cette relation, « distance-marchabilité » , est donc jugée prédictible, au sens ou si nous procédons à réduire la distance de la marche, celle-ci conduirait à une meilleure marchabilité, et donc ne nécessiterait pas de simulation informatique. Nous pensons, que non, il ne s'agit pas d'une relation toujours vraie et donc la simulation informatique est nécessaire pour vérifier.

-Des configurations graphiques/urbaines peu avantageuses pourraient nous donner un meilleur résultat(optimum) numérique.

**Expériences afférentes à l'hypothèse :** L'optimisation mono-objectif nous servira à isoler les critères hypothétiques que nous proposons d'introduire et de vérifier, un à un, leur influence sur le score et donc de démontrer, sur cette base, comment ceux-ci modifieraient ou non le résultat.

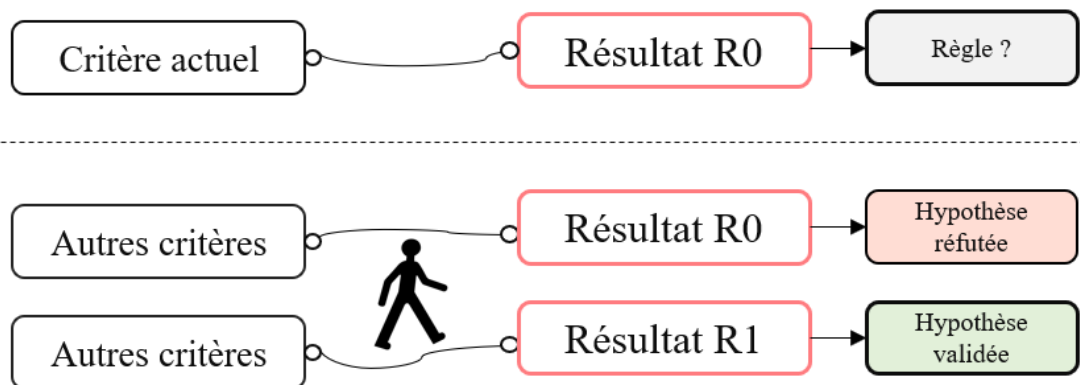


Fig 1.1 – schéma illustrant les hypothèses 1

2

**Hypothèse principale :** La réduction de l'optimisation de la marchabilité par rapport à un seul objectif, tel qu'opéré actuellement, est fausse. La première hypothèse, justifiait l'usage du mono-objectif pour pouvoir tester l'influence des critères. Dans cette hypothèse, nous posons que l'objectif de « *minimiser la distance* » à lui seul, n'est pas suffisant et que la superposition de plusieurs optimisations mono-objectif n'est pas faisable, notamment dans les problèmes urbains.

**Hypothèses secondaires :**

-La notion du « compromis » ou de la « négociation », présente dans les algorithmes multi-objectif, fera que la superposition des résultats mono-objectif n'est pas équivalente à leur interaction.

-Dans cette hypothèse, c'est cette même notion du compromis, dans les algorithmes, qui fera que certaines configurations peu avantageuses aboutiront à des résultats meilleurs

**Expériences afférentes à l'hypothèse :** L'optimisation multi-objectif : construction de deux types de scénarios : Soit par l'interaction des mêmes critères utilisés dans le mono-objectif, soit par d'autres scénarios urbains annexes qui vérifient l'influence de « plusieurs critères à la fois » sur le score de la marchabilité.

Recourir à une optimisation mono-objectif (à un seul critère) pour satisfaire un problème multi-objectif est désigné par l'approche naïve. Il s'agit d'une méthode proposée par Sean Luke [Luke, 2018] pour se soustraire du multi-objectif, en formulant le problème de telle sorte à trouver une relation entre tous les objectifs et les réduire à un seul. L'hypothèse est que, même pour des problèmes simples, la relation n'est pas toujours linéaire et que le multi-objectif est nécessaire.

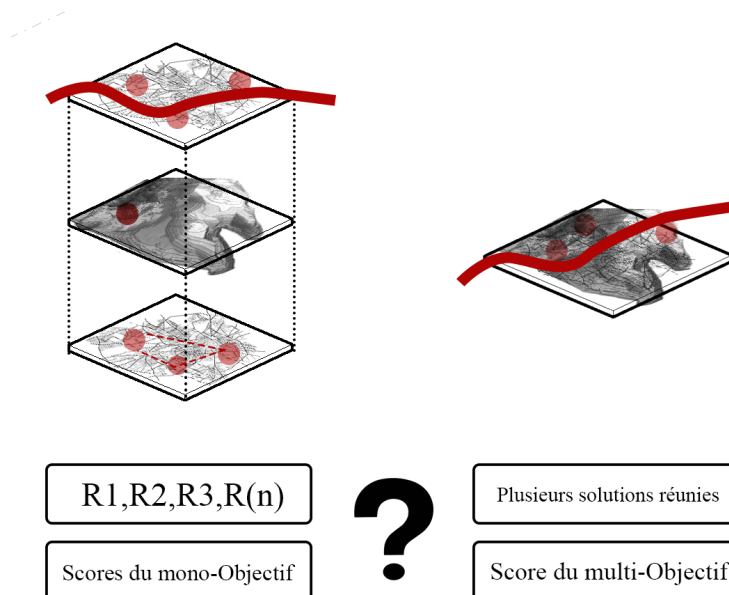


Fig 1.2 – schéma illustrant les hypothèses 2

## 1.4 | Structure du mémoire

### Partie théorique : État de l’art et de la recherche

#### Chapitre 1 : La marchabilité : Théories et émergence d’une nouvelle mesure de la ville

Pour optimiser la marchabilité, nous devrions, au préalable, comprendre ce qu’elle implique. C’est la visée de ce chapitre. Nous expliquerons le terme un peu plus et nous détaillerons les théories analytiques ainsi que les premiers outils permettant la représentation de celle-ci. Ensuite, nous présenterons le passage de ce concept vers une mesure à travers la notion des indicateurs de marchabilité. Nous expliquerons comment ces derniers ont permis de modéliser une relation mathématique entre la pratique de la marche et l’espace urbain.

#### Chapitre 2 : Walkscore : des indicateurs théoriques à l’algorithme informatique

La relation mathématique, évoquée ci-dessus, a été transposée vers un algorithme informatique. Dans ce chapitre, nous détaillerons le mode opératoire de cet algorithme et sur quels critères il se base principalement pour informer l’utilisateur sur la mesure de la marchabilité.

#### Chapitre 3 : Modélisation et optimisation de la marchabilité dans l’outil paramétrique

L’algorithme, que nous avons abordé au chapitre précédent, a été récemment transposé à l’outil paramétrique. Dans ce chapitre, nous détaillerons ce qu’implique la modélisation paramétrique, notamment sur l’application urbaine. Ensuite, nous présenterons les étapes de la construction d’un modèle de calcul de marchabilité, par l’intermédiaire entre l’algorithme et l’outil paramétrique, et enfin, nous présenterons les travaux qui ont été réalisés, sur cette question, et relèverons leur limites.

### Partie expérimentale : Expériences, Analyse et Prospective

#### Chapitre 4 : Expériences et Analyse

Après avoir relevé les limites des études précédentes et comment la marchabilité pouvait être introduite dans l’outil paramétrique, nous détaillerons dans ce chapitre, la mise au

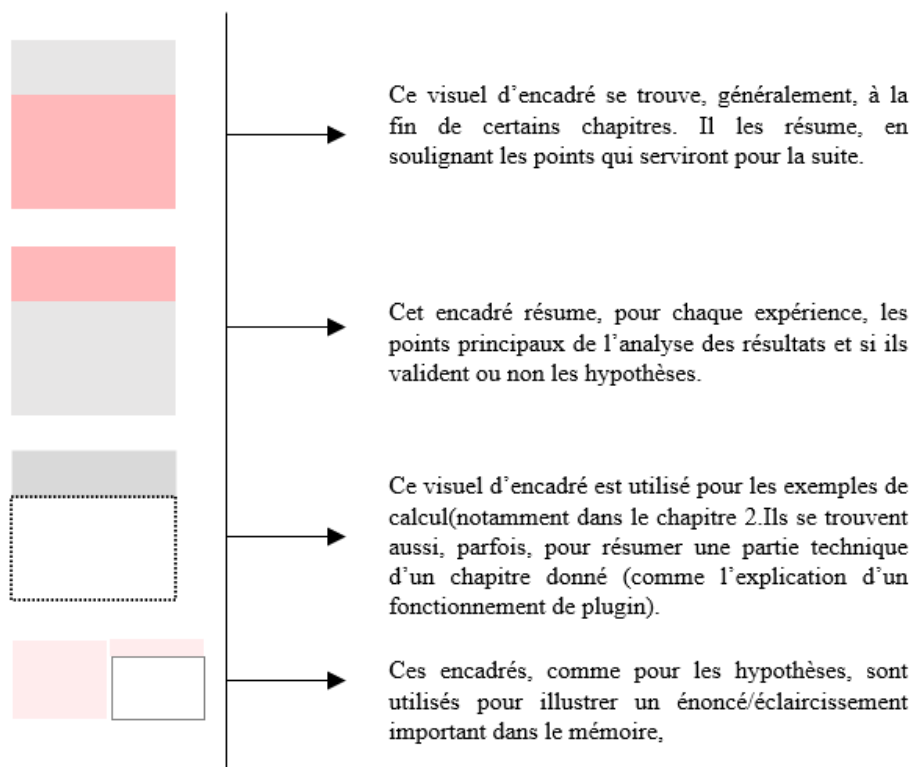
---

point du protocole expérimental. Celui-ci expliquera non seulement la méthode mais aussi un travail pré-optimisation, afin de corriger certaines limites antérieures du modèle. Enfin, nous présenterons les expériences, réparties selon la méthode d'optimisation et leur affiliation aux hypothèses données. L'analyse des résultats nous permettra de relever les limites de la méthode.

## Chapitre 5 : Limites de l'étude et prospectives

Dans ce chapitre, nous proposerons des alternatives ou des corrections face aux limites que nous avons décelées dans l'analyse du résultat, au chapitre précédent. Ces alternatives concerneront, à la fois, le processus d'optimisation et le traitement des résultats mais également la marchabilité elle-même. Ces alternatives concerneront surtout le mode de représentation des résultats d'optimisation ainsi que les cartes de marchabilité, ce qui pourra donner lieu à d'autres méthodes de calcul.

### 1.5 | Visuel du mémoire



# Partie théorique



## Marchabilité : Théories et émergence d'une nouvelle mesure de la ville

Explication du concept, des théories et des outils autour de ce concept ainsi que son passage vers une mesure de la ville, à travers des métriques urbains

### 2.1 | La marchabilité : Définition d'un concept émergent :

Quantifier le mouvement à l'échelle urbaine, telle est la voie explorée par la marchabilité. Si la marche est une pratique qui révèle l'espace, la marchabilité, elle, serait une mesure de cet espace et de sa propension à favoriser la marche [Litman,2003 et Fisher,2011].

Emergent au courant des années 2000, le terme *walkability* puise ses origines dans les études américaines sur l'activité physique et la santé. Evoluant ensuite vers l'Europe, notamment en France sous le terme de *marchabilité* ou *potentiel piétonnier*, cette mesure s'inscrit comme un facteur de l'ère post-carbone. Progressivement, elle devient une unité de conception de certaines villes et évolue vers un outil déterminant dans le développement durable de celles-ci.[Raulin et al,2012] .



Fig 2.1 – Collage,espace marchable

## 2.2 | Théories et Outils autour de la marchabilité :

### 2.2.1 | Le Transit Oriented Development : De l'optimisation du transport à celle de la marche :

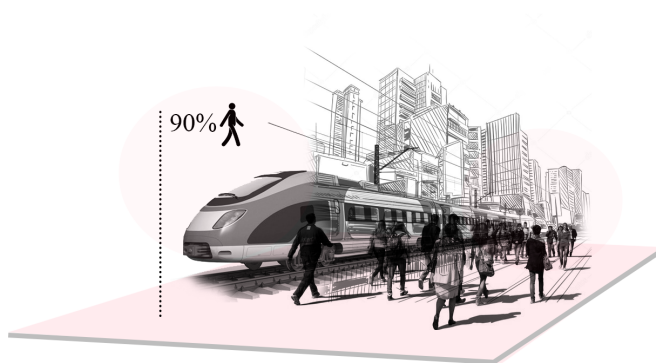
Cet attrait de l'évaluation de l'espace urbain par les études américaines, remonte également au concept du T.O.D , le *Transit Oriented Development*. Cette application, d'une conception orientée sur le transport, découlerait à l'origine, des théories de Howard en Europe, notamment sur les cités jardins, qui impliquaient la réalisation des plans en étoile avec des villes satellites, reliées par les réseaux de transport.



**Fig 2.2** – abstraction,réseau avec villes satellites

C'est, en effet, la majeure caractéristique des quartiers dits T.O.D : Leur densification, tout autant d'espaces résidentiels que commerciaux, autour d'un pôle de transport, permettant ainsi la réduction de l'usage de l'automobile et donc la marche. Le T.O.D est donc une approche de planification urbaine, à laquelle la marchabilité, serait un facteur déterminant. [Calthrope,1993].

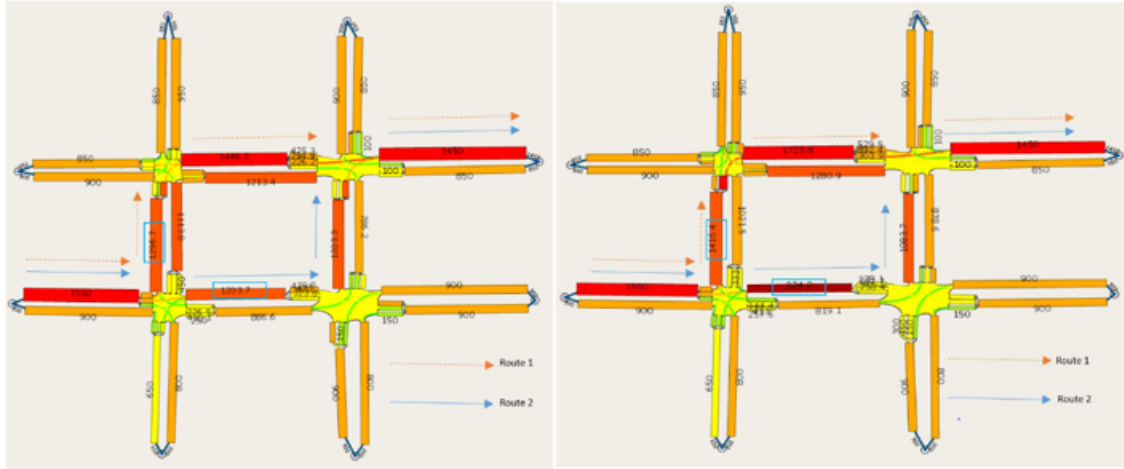
La marchabilité avait donné lieu à des études plus anciennes qui étaient étroitement liées à ce concept du TOD. Strangel[référence de l'article,année] décrit alors des travaux sur lesquels il y aurait eu une transposition de méthode de l'évaluation du transport sur le comportement du piéton, et de ce fait, ces études étaient assez critiquables [Lo, 2008] ,puisque les piétons furent considérés, à base de métriques qui les dénuiaient de leur aspect social ou cognitif.



**Fig 2.3** – Collage,marchabilité en lien étroit avec le T.O.D

A la lumière de ceci, nous citerons le travail de Tuo Mao,docteur de l'université de Sydney [Mao et al,2007], qui, sur la base d'un algorithme d'optimisation informatique, avait exploré les moyens de réduire la distance de coût<sup>1</sup> , sous des conditions particulières, en l'occurrence lorsqu'un accident routier se produit.

1. coût du transport pour des humains ou des marchandises pour franchir une distance donnée



**Fig 2.4** – plan pré et post-optimisation informatique du travail cité

Le début de l'optimisation du transport a fait émerger l'étude sur les flux et cette même notion a été reprise comme moyen de quantifier et d'optimiser le flux humain. Cependant, ce rapport au T.O.D ou à l'automobile lui-même n'a pas été perpétué, en vue de la raison précédemment évoquée et qui rejetait l'assimilation du mouvement piéton à celui de l'automobile.

## 2.2.2 | Une lecture systémique de l'espace urbain à travers la Space Syntax :

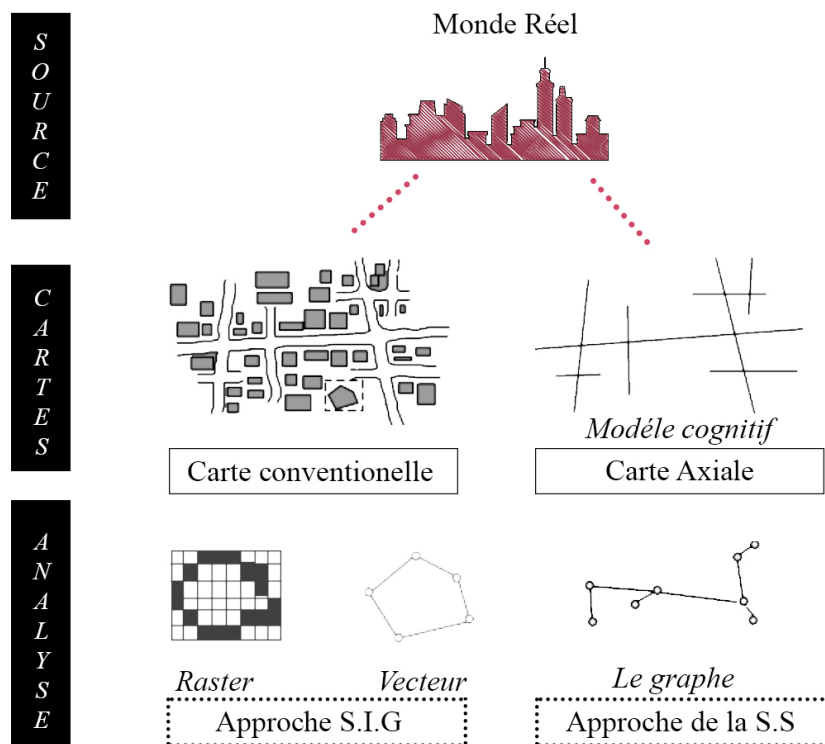
Ce que Kevin Lynch avait établi comme éléments de lisibilité de la ville (L'image de la cité, 1960) ou ce que l'architecte et anthropologue Christopher Alexander avait dressé comme inventaire des schèmes de celles-ci (A Pattern Language, 1977), auraient servi de socle à la méthode de la syntaxe spatiale (1984). [Gohari, 2019]

A l'instar des lectures proposées, émerge la space syntax (S.S). Énoncée par les professeurs de l'UCL, Bill Hillier et Julienne Hanson, elle constitue un corpus de théories décrivant l'interaction entre l'espace et l'homme. Les concepts de la S.S s'attachent à illustrer et expliciter le comportement engendré et forgé par un lieu, notamment à travers ses aspects configurationnels. Elle se distingue de l'approche SIG<sup>2</sup> par la dimension cognitive qu'elle reprend, dans ses métriques, notamment dans la navigation spatiale.

C'est ainsi que la représentation des relations élémentaires devient le sujet fondamental de

2. Systèmes d'Informations Géographiques : Dans ce système, l'approche analytique traite les données spatiales sur la base de distances euclidiennes et à travers les vecteurs/rasters

la SS. A travers cette description des interactions, la grille urbaine permet ,à titre d'exemple, de « peindre » les comportements socio-économiques en relation à l'espace de la ville lui-même.[Emo et Al.2012]



**Fig 2.5** – Comparatif entre l’approche conventionnelle d’analyse et l’approche syntaxique

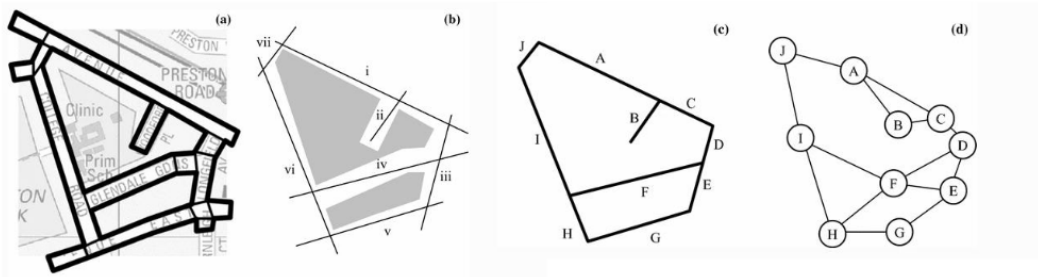
Dans ce langage morphologique qu’est le graphe, la syntaxe spatiale permet donc également de construire des modèles capables de décrire mais aussi de prédire l’évolution de la structure spatiale (urbaine ou architecturale) ainsi que le comportement au sein de cette structure. Particulièrement, cette méthode exploratoire permet de relever les potentialités de la ville, non pas uniquement par ses formes bâties mais par le vide et la manière dont il est planifié ou vécu.<sup>3</sup>

### 2.2.2.1 | Les outils analytiques de la Syntaxe spatiale urbaine :

La thèse soutenue par Hansson et Hillier [Hillier et al,1993] permet de retrouver au sein de plusieurs configurations ou typologies de villes différentes des aspects communs, expliquant

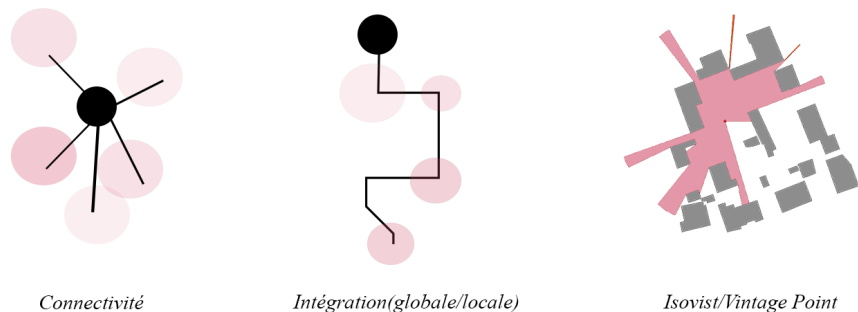
3. Hillier, B dans *Space is the machine*, 1996.

ainsi toute un paradigme dans les sciences de planification urbaine et du comportement humain. La quantification de ces aspects relationnels, tant sur le point spatial que cognitif, est rendue possible par une analyse mathématique de la ville. En effet, à partir de la théorie des graphes<sup>4</sup>, Hillier parvient à traduire l'espace urbain ou architectural, en termes d'un réseau d'arêtes et de nœuds, respectivement, dans le cas de l'urbanisme, la rue et l'intersection, principaux éléments générateurs du tissu urbain.



**Fig 2.6** – Passage de la représentation conventionnelle à la représentation S.S de gauche à droite (a),(b),(c) et(d) tels qu'expliqué ci-dessous

Cette transposition du plan vers une forme de représentation mathématique passe d'abord par **(a)** la traduction de la carte vers des espaces convexes, **(b)** la génération de carte axiale à travers la reconnaissance des segments passant par ces espaces convexes **(c)** les lignes axiales sont ensuite raccourcies et enfin **(d)** la représentation sous la forme des graphe avec chaque intersection comme étant un nœud ou vertex. Cette notion de relation est notamment accentuée par l'émergence du degré de connectivité, un des métriques fondamentale de cette méthode d'évaluation. D'autres critères, afférents à cette méthode, sont les indicateurs d'intégration, à échelle globale ou locale, ainsi que la visibilité traduite par le concept d'Isovist.



**Fig 2.7** – Certains métriques de la Syntaxe Spatiale.

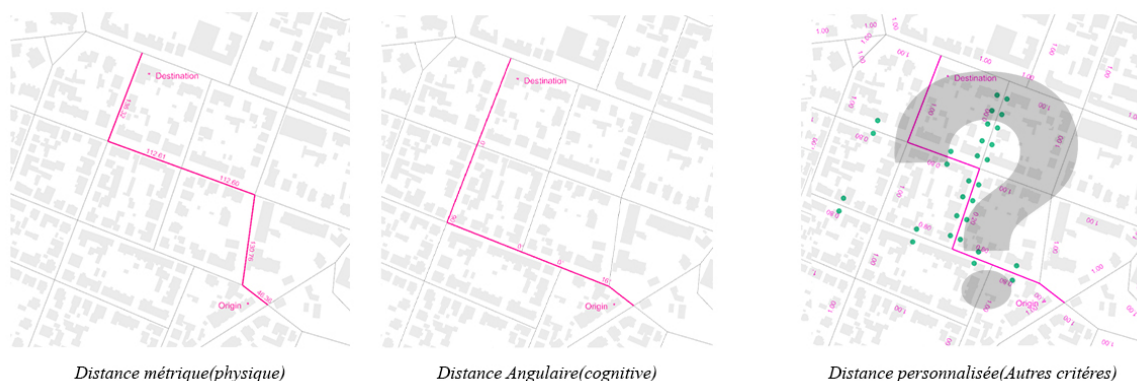
4. Théories des mathématiques et informatique qui permet la représentation en réseau.

Métrique de la syntaxe spatiale	Définition
<b>Connectivité</b>	Mesure combien d'espaces sont directement connecté au nœud/segment sélectionné (un espace autre)
<b>Intégration</b>	Mesure normalisée d'une distance d'un espace donné, pris comme origine, jusqu'à tous les autres espaces dans le système étudié. Elle calcule la proxémie de l'espace à l'origine jusqu'à tous les autres
<b>Isovist</b>	Aire tracée reliant tous les points visibles à partir d'un « vintage point/point de vue » déterminé pré-calcul, dans un environnement donné.

**Tableau 2.1** – Tableau des métriques de la S.S illustrés ci dessus et leur définition, telle qu'établie par le labo de l'UCL-spacesyntax

A partir de ces métriques, il est possible de construire des cartes qui décrivent une certaine logique de l'organisation humaine et spatiale. Il est à préciser que l'outil analytique, qu'est la space syntax, relève plus de résultats graphiques que d'indicateurs numériques. Il est bien évidemment possible de construire des coefficients ou des résultats numériques du flux pédestres sur les co-relations entre les diverses cartes produites mais, la marchabilité a été l'objet d'études spécifiques des indicateurs de marchabilité. Ces derniers seront étudiés dans la section[2.3.1]

Afin d'établir les indicateurs graphiques de la SS, il serait nécessaire de spécifier qu'en terme de mouvement urbain, cette méthode permet de calculer deux types de distances : la distance conventionnelle métrique, sur la base du chemin le plus court. La seconde distance est dite "angulaire", soit le chemin le plus droit possible. Il serait donc pertinent de se questionner, par rapport à la modélisation d'autres types de distances, non étudiées par les graphes de la S.S.



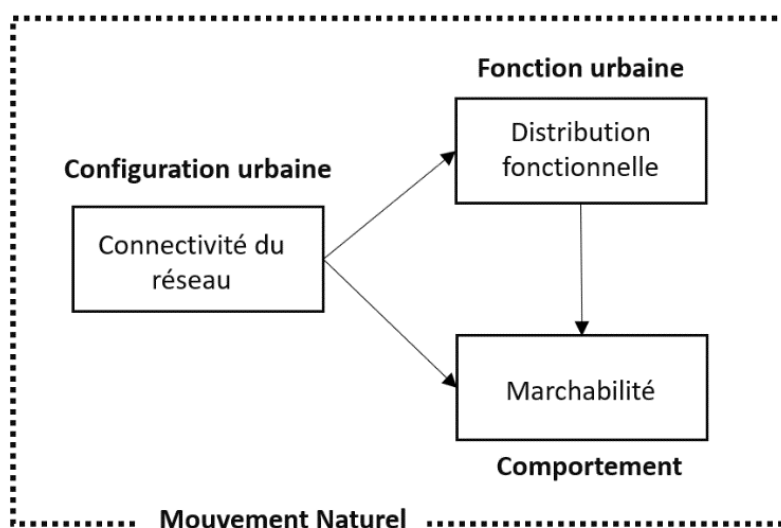
**Fig 2.8** – Les types de distances dont les deux premières calculables par la S.S et la troisième, une distance autre

La question de la mobilité urbaine apparaît, pourtant, comme fondamentale dans l'étude de la syntaxe spatiale. D'ailleurs, le fondateur de cette méthode analytique y consacre une partie de ses travaux, qui feront émerger la théorie du mouvement naturel. [Koenig et al, 2019]

### 2.2.2.2 | La théorie du mouvement naturel de la S.S

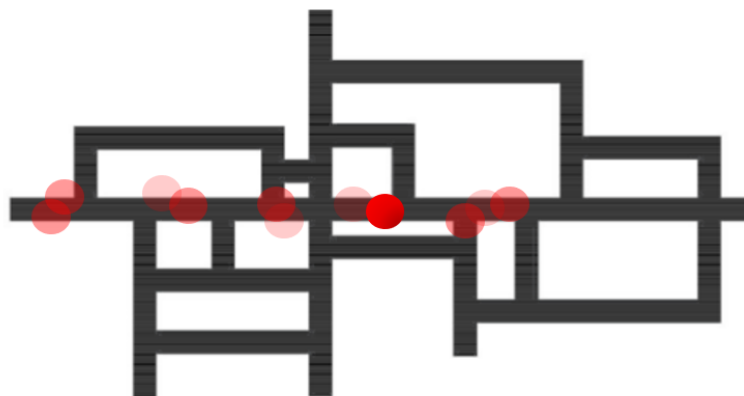
Les outils analytiques de la space syntax, précédemment évoqués, peuvent constituer un socle d'étude du mouvement ou du flux pédestre au sein de l'espace urbain. Bill Hillier énonce que le fondement même de la syntaxe spatiale met en lumière cette dualité entre la production de la ville, en termes de réseaux et d'équipements, et le mouvement qui y sévit. Nous rappellerons, cependant, que les cartes axiales, produites par sa méthode analytique, ne mettent pas en valeur la distribution fonctionnelle. [Ratti, 2004]

Sur ce parti pris, le fondateur de cette analyse définit la théorie du mouvement naturel, qu'il est possible d'illustrer par le schéma suivant :



**Fig 2.9** – Schéma de la théorie évoquée-relation du mouvement par rapport à la forme et configuration urbaine

Afin de motiver cette théorie, Hillier [Hillier, 1999] propose une figure hypothétique d'un réseau viaire ainsi que l'expérience suivante :



**Fig 2.10** – Schéma de l'expérience de pensée proposée par Bill Hillier dans l'énoncé de la théorie étudiée

Le réseau illustré est constitué par la rue principale (long segment horizontal) ainsi que par des proliférations secondaires et d'autres "ramifications" à une échelle moindre. Toutes ces rues sont bordées par des espaces résidentiels. Il est alors demandé d'imaginer le flux de personnes parcourant cette configuration, d'un espace à un autre, en se soumettant à une certaine logique exploratoire. L'exploration de cette figure, devrait se faire, sur la base d'une distance angulaire moindre, comme nous l'avions évoquée dans la figure, soit un chemin avec le plus droit possible.

Sur cette expérience de pensée, nous pourrions classer l'artère principale comme étant le noyau des flux, hébergeant le plus de mouvement, contrairement aux espaces périphériques, qui eux, impliquent nécessairement plus de parcours tournant soit une distance angulaire supérieure. Ceci réduirait la propension à y marcher. Il s'agit là, également d'une observation cognitive, qui mettrait l'accent sur les préférences humaines à mouvoir en ligne droite.

A partir de cet exemple et de sa déduction, Hillier énonce que les principales activités, soit l'implantation des équipements, seraient plus enclines à se proliférer le long de cet axe principal. Cette règle, bien perceptible en termes d'aménagement urbain, serait pourtant bien prône à la critique et assez discutable comme nous le verrons dans la section [2.2.2.4].

### 2.2.2.3 | Modèles informatiques de prédiction du mouvement basés sur la théorie de Hillier :

Martin Bielik et Reinhard Koenig, professeurs à l'université du Bauhaus, en sciences informatiques, publient dans le douzième symposium de la S.S, une tentative d'un modèle prédictif, sur le flux piétonnier, en intégrant, dans la représentation de la ville, un paramètre omis dans les cartes produites par la S.S, à savoir la distribution des équipements. Le mo-

dèle consiste à affirmer, s'il est possible ou pas, de prédire, sur une période temporelle, le changement du flux, en fonction de la distribution fonctionnelle.



Multiplier effect - correlation between movement potential at first and last iteration of the simulation  $\text{LogN} (MP_1 \sim MP_N)$

**Fig 2.11** – Simulation sur l'outil paramétrique de la prédictibilité du mouvement par rapport à la distribution fonctionnelle dans le temps

Pour cela, les deux chercheurs mesurent l'évolution de deux métriques, relatifs à l'analyse des graphes, à savoir l'intégration(globale et locale) et à la connectivité. Le coefficient R, perçu sur la figure, permet de trancher sur la prédictibilité du mouvement ou pas selon les scénarios étudiés.

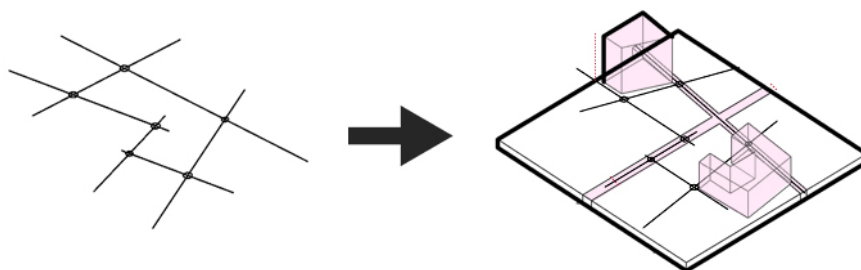
Lorsque  $R$  est proche de 1, comme dans le quatrième scénario, la relation entre l'évolution de la distribution fonctionnelle-mouvement est linéaire et prévisible et ce, selon une métrique donnée. Dans les autres cas, cette relation requiert une simulation informatique.

#### 2.2.2.4 | Limites et critiques de la space syntax

Si la syntaxe spatiale se veut comme un langage qui explique l'espace à travers la pratique humaine et non « à vol d'oiseau », la construction du précédent modèle est sujette à questionnement. De toute évidence, ce modèle se base, paradoxalement, sur une étude d'un système de graphes. Donc, l'analyse se fait à travers des équations ou des algorithmes, traitant uniquement l'unité du segment et du point.

L'architecte et ingénieur italien, Carlo Ratti s'aligne d'ailleurs à cette critique de la S.S concernant la dimension étudiée. Ce créateur du Sensable City Lab au MIT questionnerait la validité des cartes de la Syntaxe Spatiale, qui retracent uniquement la bi-dimensionnalité de l'espace, soit une étude sur plan, omettant la troisième dimension de l'espace : La réalité physique du terrain.

Dans son étude critique, il explique qu'il serait impossible de construire des modèles de simulation de flux, en omettant tant de critères dont il cite, par exemple, la hauteur du bâtiment ou encore la largeur de la rue, qui seraient des facteurs tout aussi déterminants dans la mesure de la pratique de l'espace urbain.

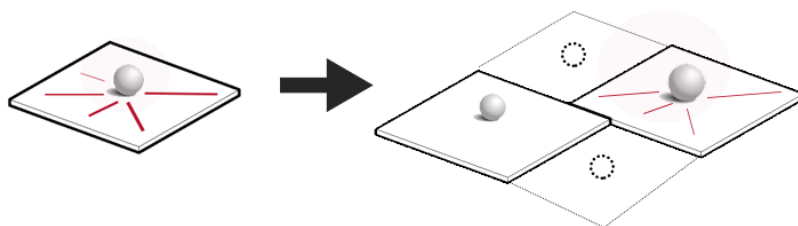


**Fig 2.12** – schéma de la carte Axiale de la S.S et manque d'informations par rapport à la troisième dimension

#### **A - Le paradoxe de la centralité à travers les modèles locaux :**

La carte axiale, reposant principalement sur des unités d'analyse de l'ordre du segment ou du nœud, ne permet pas d'extraire les critères que Ratti cite. A cet effet, l'architecte illustre,

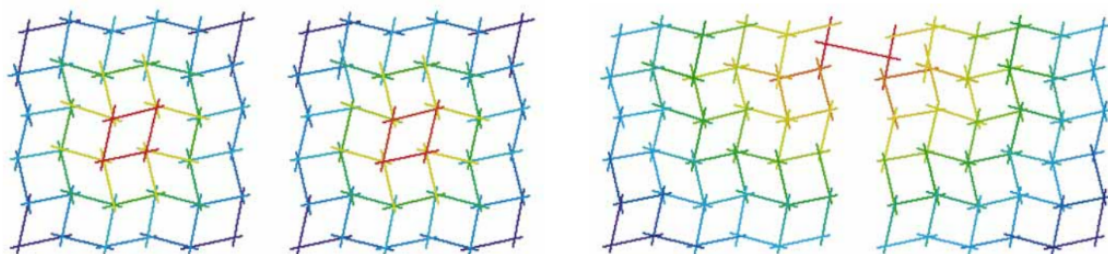
par l'exemple de La Défense, ses propos. Il effectue, en effet, une analyse de l'intégration du secteur de la Défense à travers la génération d'une carte axiale qui permet d'argumenter les défaillances de la traduction de la ville en graphes.[Ratti,2004].



**Fig 2.13** – schéma, paradoxe de la centralité

L'auteur relève alors les défaillances de l'analyse de la SS. Ainsi, le secteur étudié, traduit par les outils de la SS comme un espace peu intégré ou connecté aux espaces urbains adjacents, est pourtant, dans sa pratique sur terrain, perçu comme un espace favorisant la marchabilité, voire, un espace qui est principalement caractérisé par cette dimension piétonne, lorsque nous nous représentons l'esplanade.

Au-delà de cette analyse, Ratti propose de recentrer le regard et questionne les erreurs de calcul relatives à l'étude partielle des cartes. En effet, l'une des restrictions de la démarche numérique, est la réduction de l'étude à une zone locale. Bien que cela soit ,généralement, pertinent pour des études ou des expérimentations mineures, l'architecte italien rejette la S.S comme une méthode scientifique approuvée en illustrant, à travers l'outil Axman<sup>5</sup>, le « paradoxe de la centralité » ou encore le problème de la localité tel que l'illustre la figure suivante :



**Fig 2.14** – Paradoxe de la centralité et mesure de l'intégration globale par Ratti : D'abord dans deux système urbain séparés, puis mis en communication - sur l'outil Axman PPC

5. Outil informatique développé par le laboratoire S.S de l'UCL où la théorie a émergé.

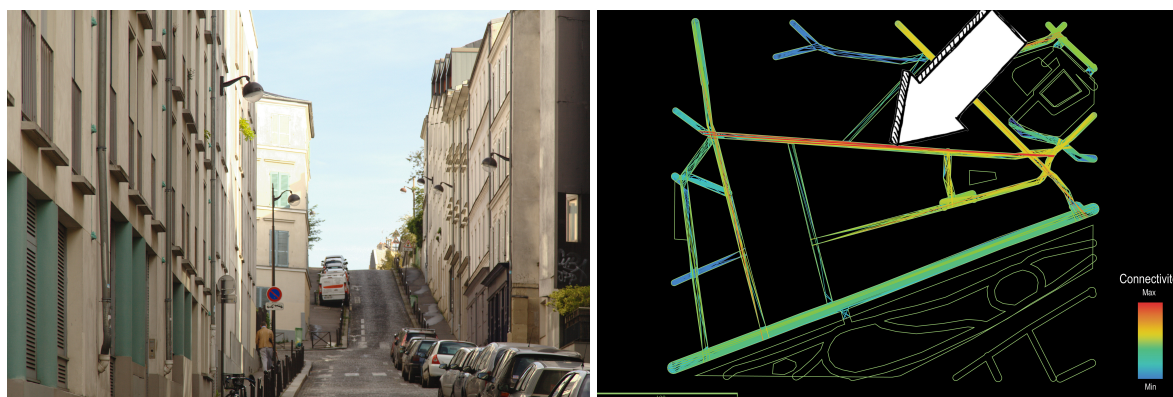
Le calcul de l'intégration du tissu urbain change simplement par la mise en communication des deux grilles urbaines. Le problème du local est commun, à pleines d'applications informatiques et ce, en raison du coût de simulation de grandes étendues géographiques.

### **B- De la nécessité d'intégrer la troisième dimension :**

Dans ce travail de mémoire, nous avons aussi proposé de constater cette différence entre le terrain physique et le décalage du calcul syntaxique et la théorie du mouvement naturel. Pour cela, nous avons mis en application le métrique relatif à la S.S : La connectivité. Comme nous l'avions évoqué, cette mesure a été jugée par Hillier comme un paramètre fondamental dans la marchabilité. La démonstration est réalisée à l'aide de l'outil DepthMapX, également développé dans les laboratoires de l'UCL comme Axman PPC.

Pour cela, nous avons effectué une simple expérience d'observation sur la rue de Gasnier Guy au vingtième arrondissement de Paris. Sa particularité réside dans sa pente s'élevant à 17%, faisant d'elle la rue la plus pentue de la capitale française.

Sur terrain, il s'agit d'une rue résidentielle peu fréquentée. En réalisant l'analyse syntaxique par l'outil dédié, nous trouvons qu'en raison de la restriction graphique et des données, relatives aux cartes axiales de la S.S, la rue est calculée comme étant hautement connectée. Si nous nous remettons à cette simulation et à la théorie de Hillier évoqué à la section [2.2.2.2], alors la rue serait support des activités commerciales et donc favoriserait la marche, ce qui n'est pas le cas. Nous avons également à travers, l'Isovist2D, montré la défaillance de ce type d'analyse. Dans l'annexe B, la figure B.1 illustre ceci.



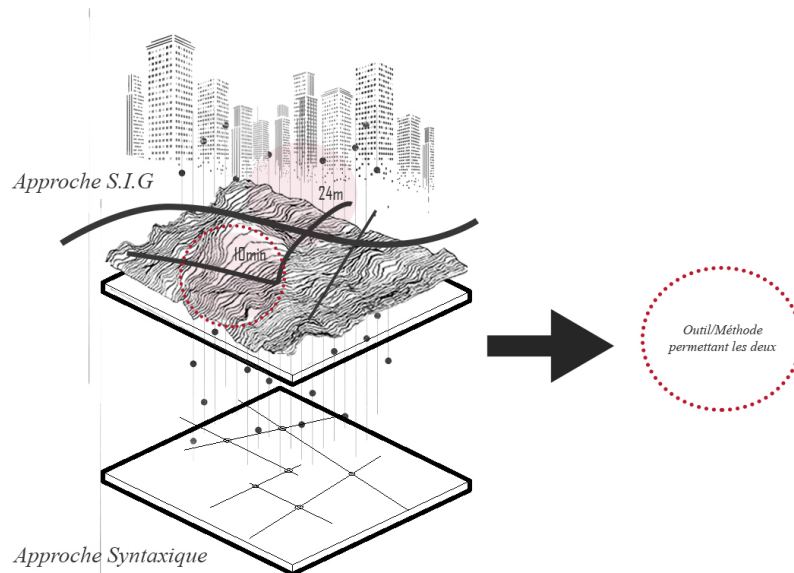
**Fig 2.15** – Rue Gasnier Guy avec l'analyse de la connectivité sur DepthMapX : La rue est désignée comme bien connectée mais cela ne suffit pas pour impliquer sa marchabilité

**C - La complémentarité des méthodes comme correction des limites :** Pour la correction de ces exemples d'erreurs, nous pouvons supposer que l'introduction d'une certaine

sémantique et d'un système de Tag ou de classes, pour répertorier l'importance des rues (traitée toujours en tant que'un simple segment) permettrait de outrepasser ces décalages et c'est ce que s'attache à faire certains systèmes de cartographies informatiques tels que Open Street Map<sup>6</sup>.

Enfin, la proposition de Carlo Ratti implique la réintégration de la dimension euclidienne dans la S.S, à travers des données, telles que le temps de déplacement. Il propose l'extraction de ces données-là à travers des D.E.M (digital elevation model)<sup>7</sup> [Ratti, 2004].

Il n'y a pas de réelle opposition entre les méthodes, il serait surtout question de compléter l'étude de la space syntax par des données nécessaires et facilement extraites d'outils spécialisés, tels que les outils des systèmes d'informations géographiques (S.I.G).



**Fig 2.16** – schéma complémentarité des méthodes : approche SIG et approche syntaxique, quel outil ?

6. Projet mondial de cartographie en ligne, qui réunit une bande de données géographiques et qui fonctionne par un système de Tag affilié à la géométrie.

7. Modèle numérique de terrain qui permet de récupérer la topographie pour être incluse dans le calcul informatique.

## 2.3 | Des théories analytiques à la mesure de la marchabilité

Dans cette section, nous allons aborder les variables définies par les chercheurs et spécifiques à la marchabilité. Ces dernières, liées par une relation mathématique, aboutiront à l'indicateur de marchabilité.

### 2.3.1 | Les indicateurs de marchabilité : Des métriques pour évaluer l'espace urbain :

La littérature scientifique met en lumière cette dualité d'espace-corps, tout autant que l'environnement bâti et le tissu urbain, comme les moteurs de la marchabilité. C'est notamment le projet international IPEN<sup>8</sup>, lancé au début des années 2000, qui s'est attaché à construire des métriques ou des variables permettant de quantifier ou de mesurer la marchabilité d'un lieu donné. Ces variables constituent un « score » désigné par l'indicateur global de marchabilité ou PLOS, en anglais, pour qualifier le « Pedestrian Level Of Service ».

#### 2.3.1.1 | Classification des variables :

Du fait de la richesse des études élaborées sur la question, les indicateurs de marchabilité sont multiples et plusieurs auteurs : architectes, urbanistes ou théoriciens de la question ont établi un indicateur propre à leur démarche analytique. Néanmoins avant d'énumérer les plus importants, il conviendrait de les classer, selon la nature des variables ou des données, ainsi que la méthode d'extraction de ces dernières. En effet, les variables s'étalent depuis la structure bâtie jusqu'au spectre des perceptions subjectives. Le tableau 2.1 ,ci-dessous, illustre un exemple de classification[Maghelal et al, 2011] :

---

8. International Physical Activity and the Environment Network : Programme lancé en 2004 aux USA pour la recherche sur le lien entre l'environnement physique et l'activité.

Nature/Type de la variable	Définition de la variable	Méthodes/Outils d'extraction/
<b>Objective</b>	Quantification/mesure selon une méthode généraliste pouvant, donc, être répliquée à d'autres études.	GIS / Audit
<b>Subjective</b>	La méthode de mesure peut-être répliquable ou pas.	Questionnaire
<b>Distinctive</b>	La méthode de quantification ne peut être généralisée, elle est issue et appliquée sur un seul cas d'étude (le contexte duquel elle émerge).	Observation sur terrain

**Tableau 2.2** – Tableau de classification des types de variables et la méthode afférente

Il conviendrait, bien évidemment, de privilégier les variables objectives car celles-ci permettent de construire un indicateur ou une méthode de calcul reproductible et applicable à d'autres espaces et qui émergerait d'un consensus commun [Baumann et al, 2002, Frank et al.2005].

### 2.3.1.2 | Les indicateurs de marchabilité

Le tableau, ci-dessous, recense les indicateurs les plus connus dont particulièrement celui du professeur urbaniste Lawrence D.Frank(2005) et celui de la docteure en urbanisme à l'EPFL, Anne Vernez Moudon(2005). Ces deux indicateurs constitueront, par la suite, le socle de l'algorithme informatique et ce, vu leur pertinence et leur objectivité dans les variables qui les constituent. Les unités d'analyses dénotent le recours, à la fois, à l'approche SIG et de celle syntaxique, comme nous avons évoqué, dans la partie précédente .

Auteur	Indicateur de marchabilité	Unité D'analyse	Méthode/Outils	Type de variables
Allan [2001]	Walking Permability Indices	Aire	G.I. S	Objective
Bradshaw [1993]	Walkability index	Aire	Questionnaire	O/S
Dixon [1996]	Pedastrian Performances Mesure	Segment	Audit	Objective
Frank et al [2005]	Walkability Index	Aire	GIS	Objective
Moudon [2001]	Pedastrian Location Identifier 1 et 2	Aire	GIS	Objective
Kim et al [2006]	Level of Service Indicator (LOS)	Segment	Questionnaire	Subjective
Wellar [2003]	Basic Walking Security Index	Intersection	Audit	O/S

**Tableau 2.3** – Tableau de certains indicateurs avec les unités d'analyse et méthodes afférentes

Ces indicateurs sont le résultat d'un ensemble de variables, qui ont été sélectionnées, soit à travers des enquêtes soit par le biais d'observations réalisées sur terrain.

### 2.3.1.3 | Méthodologie de choix et de calcul des variables

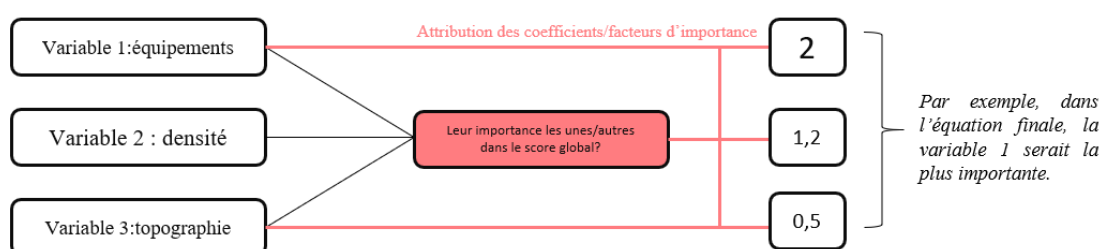
Les index de marchabilité sont généralement construits selon la méthodologie suivante :

- La sélection d'un ensemble d'indicateurs évaluant la marche.
- Quantification de ces indicateurs.
- Normalisation des indicateurs afin d'avoir un calcul homogène.(vue la différence des unités de chaque variable)
- Affectation d'un facteur/coefficient d'importance (Weight factor)(Que vaut une variable par rapport à une autre ?)
- Agrégation de l'indicateur de marchabilité et composition de l'indicateur global (selon une équation)

La première étape peut se révéler comme la plus difficile [Castillo, Pitfield, 2010]. Les chercheurs prônent alors, l'utilisation de variables dont l'obtention ou l'extraction est la plus aisée.

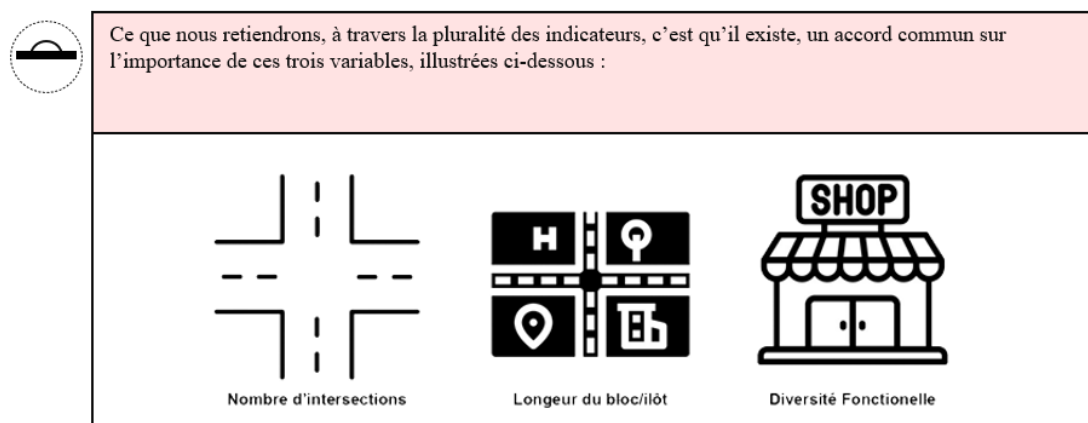
Ceci, bien que tout à fait justifiable, permet cependant de soulever des questions quant à la valeur des données accessibles.

Une fois les variables choisies, l'affectation d'un facteur d'importance s'opère selon la méthode de hiérarchie multicritère (AHP : Analytical hierarchy process)<sup>9</sup>. Il n'est donc pas question d'exactitude, les chercheurs comparent, surtout, les critères les uns aux autres, en hiérarchisant leur importance, de façon à pouvoir procéder à leur analyse du contexte, selon leur propre méthode.



**Fig 2.17** – schéma d'un exemple de la notion des coefficients d'importance dans les variables de marchabilité

Parmi ces variables, nous relèverons les plus caractéristiques des études futures et notamment ceux qui vont servir de base aux algorithmes informatiques. Comme nous l'avions évoqués, c'est ceux de Frank et de Moudon qui ont alimenté la continuité de la recherche.

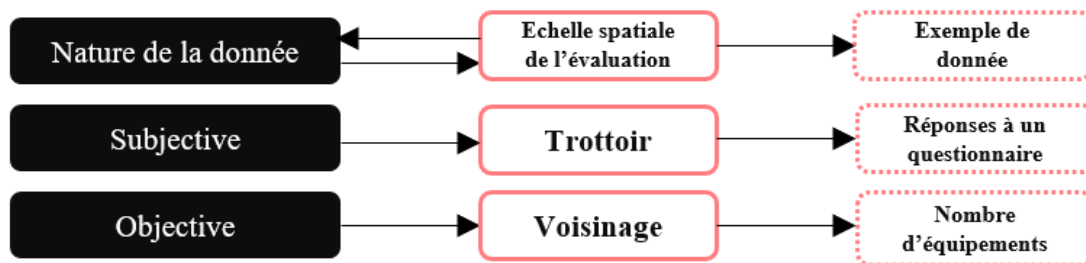


9. Processus d'analyse décisionnelle qui permet de quantifier la valeur d'une décision ou d'un critère dans un ensemble complexe

Présente dans tous les indicateurs, cette triade de variables constitue la colonne vertébrale du calcul de la marchabilité. D'autres variables peuvent se greffer, selon l'étude de l'auteur, au calcul mais demeurent d'une importance moindre, voire omises totalement, surtout par les algorithmes que nous étudierons dans le chapitre suivant [3.1].

#### 2.3.1.4 | Échelle et portée de l'évaluation

Outre la classification de la nature des variables, la question de l'évaluation de l'espace urbain soulève un rapport d'échelle. A quelle échelle ou rayon, ces variables mesurent-elle la marchabilité d'un lieu ?



**Fig 2.18** – schéma du rapport : nature de la donnée-échelle spatiale d'évaluation

La rue, le voisinage ou la ville ont tous été étudiés et ce, selon les diverses approches des auteurs. Ces échelles de mesure sont, également, en corrélation avec la nature des variables. A titre d'exemple, si l'indicateur de marchabilité regroupe des variables subjectives, qui impliquent une observation et un questionnaire sur terrain, celui-ci serait plus enclin à nous procurer une mesure à l'échelle fine, donc à l'échelle de la rue ou du trottoir.

Tandis que, les mesures dites objectives et extraites à partir des outils numériques relèveront, généralement, de l'échelle du voisinage et de la ville. Cette différence d'échelle est fonction de la disponibilité des données à l'échelle micro, parfois onéreuse en termes de coût computationnel ou inexistante. Ci-dessous, un tableau présentant trois cas d'étude avec des échelles différentes. Nous noterons qu'il existe donc d'autres méthodes, autres que celles précitées dans le tableau 2.1, comme celle de l'estimation par noyau (Kernel Density Estimation).<sup>10</sup>

10. Méthode utilisée en statistiques qui permet d'estimer la densité de probabilité d'une variable aléatoire

Comparatif/Chercheurs	Frank et al (2005)	Sundquist et al (2011)	Buck et al (2011)
<b>Discipline de recherche</b>	Planification urbaine, psychologie et médecine	Géo-informatique et santé	Santé, médecine et sciences sportives
<b>Echelle spatiale</b>	Aire à 1km autour des maisons des participants	Voisinage déterminé à partir des limites administratives	Entourage d'une école
<b>Application numérique</b>	GIS.Euclidien network buffers	A partir de l'index de Frank	Estimation par noyau (KDE)
<b>Ville/Pays d'étude</b>	Atlanta, USA	Stockholm, suède	Delmenhorst, Allemagne

**Tableau 2.4** – Tableau de comparaison des échelles et des méthodes d'évaluation de trois chercheurs dont L.D Frank

#### 2.3.1.5 | L'équation ou la fonction d'évaluation de la marchabilité

Du fait de la différence des unités dans ces variables, il est nécessaire de normaliser les valeurs et de réaliser un calcul homogène. Le chercheur Lawrence D.Frank, procède alors, par quatre variables, selon l'équation suivante[Frank et al, 2009] :

$$\text{Indicateur de Marchabilité} = [(2 \times z\text{-Densité d'intersection}) + (z\text{-densité résidentielle}) + (z\text{-COS}) + (z\text{-mixité fonctionnelle})] \quad (1)$$

Où le Z est désigné comme « Z-score », ou cote Z, une valeur exprimant l'écart type par rapport à la valeur moyenne.

Le cas d'application de l'équation du chercheur cité, émane de son étude sur Atlana, pré-cité dans le tableau ci-dessus, où l'auteur procède à un calcul sur un rayon de 1Km, donc à une échelle relativement globale.

Plus généralement, il a été convenu, selon la méthodologie commune des auteurs, que l'équation d'évaluation de la marchabilité pourrait être notée ainsi :

$$WI = \sum W_{ij} X_{ij} \quad (2)$$

où : *WI = Walkability Index*, *W = coefficient d'importance d'une variable*  
*et X = variable/critère donné*

Ces équations seront ensuite intégrées dans l'outil numérique. Elles sont dites des fonctions d'évaluation et pourront, par la suite, être réadaptées sur un algorithme, comme des fonctions « objectif »<sup>11</sup> afin de procéder à des opérations d'optimisations informatiques.

### 2.3.2 | La Space Syntax Walkability :(SSW) : Modèle de substitution des indicateurs de Marchabilité

L'apport de la S.S, au sein de l'étude comportementale de la marchabilité, est enrichi par certaines tentatives de transposer les notions et de reformuler l'équation de Lawrence Frank. A cet effet, par analogie aux concepts de la syntaxe spatiale de Hillier, évoqués dans la section [2.2.2.1], il est possible de reformuler l'équation et de se baser sur des données autres, lorsque ceux de la marchabilité sont indisponibles [Kockelman, 1997 et Koohsari et al, 2016].

Concepts des indicateurs de Marchabilité	Concepts de la Space Syntax Walkability (SSW)	Les 3D de la planification urbaine [Koohsari et al,2016]
<b>Mixité Fonctionnelle</b> <b>Nombre d'intersection</b> <b>COS</b> <b>Longueur de l'îlot urbain</b>	L'intégration (score obtenu à travers un calcul de logiciel Lié à la SS :DepthMapX)	Diversité Design
<b>Densité Résidentielle</b>	Densité Résidentielle Brute (DRB=Nbre de logements/Superficie totale du quartier)	Densité

11. fonction, utilisée dans les opérations d'optimisations, sur laquelle nous nous basons pour déterminer la solution optimale.

---

**Tableau 2.5** – Tableau de transposition des concepts de la SS et des 3-D de la marchabilité développés par l'ingénieur Kara Kockelman et repris par Kooshari pour le modèle de substitution.

A travers cette analogie, l'équation de l'indicateur de marchabilité de Frank (1) pourrait être réécrite ainsi :

$$\mathbf{SSW} = \mathbf{Z} \times [\mathbf{Z} \text{ (DRB)} + 2 \times \mathbf{Z} \text{ (Intégration)}] \quad (3)$$

Cette création d'un « modèle de substitution », pour le calcul de la marchabilité, témoigne des possibilités d'assimilation de concepts "annexes", forgés par les théoriciens sur la question de l'espace urbain.

Ainsi, nous pourrions supposer qu'en cas d'impossibilité d'extraire certains critères nécessaires dans une démarche de calcul donnée, la transposition des méthodes et donc des variables afférentes, pourrait permettre de créer des modèles de calculs approximatifs.

## 2.4 | Résumé et conclusion partielle



### Pour résumer...

Nous avons présenté la marchabilité, depuis son avènement avec les conceptions du TOD (quartiers à développement orienté vers le transport), jusqu'à sa concrétisation comme **une nouvelle mesure de la ville**, dans les recherches théoriques. Nous avons, également, abordé des théories annexes comme « La syntaxe spatiale », qui, par ses propres métriques et représentations, permettait de calculer la marchabilité.

Cette mesure, qui nous intéresse, est connue sous le nom **d'indicateur de marchabilité (ensembles de variables)**. Elle désigne à quel point un **espace est favorable à la marche**. Les chercheurs se sont attachés à désigner certaines **variables objectives** permettant, justement, de « quantifier » ce potentiel piétonnier de l'espace urbain.

Nous retiendrons, principalement, **les indicateurs de Lawrence D. Frank et les travaux d'Anne Vernez Moudon**, deux professeurs urbanistes, qui par leur recherches, ont mis en lumière principalement **3 variables/métriques constituant l'indicateur de marchabilité** :

- La diversité des équipements (nombre/type)
- L'intersection des réseaux viaires (densité des nœuds)
- La longueur marchable d'un îlot.

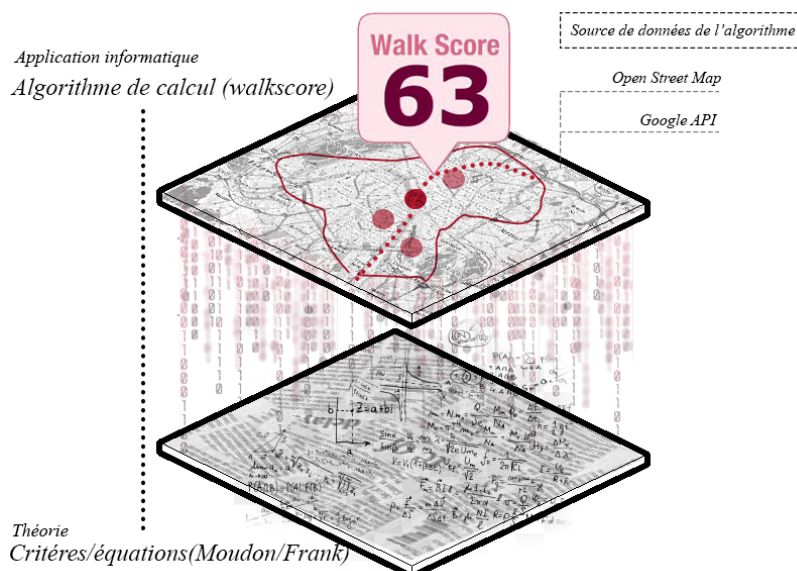
Afin d'affiner la variable « diversité des équipements », Moudon effectue une recherche sur **les classes des équipements** aux USA ainsi que leur importance (par fréquentation). Cette classification **sera ensuite reprise par l'algorithme « Walkscore »**, sous le nom de « **coefficients d'importance** » de l'équipement (**Amenity Weights** = score de l'équipement donné)

La marchabilité, dans les recherches théoriques, peut-être calculée selon diverses échelles (du voisinage jusqu'à l'échelle d'un segment de rue.)

## Walkscore : Des indicateurs théoriques à l'algorithme informatique :

### 3.1 | Outil de mesure automatisé :

Comme pour la space syntax, les recherches sur les métriques spécifiques à la marchabilité, notamment ceux de A.V Moudon et L.D Frank ont esquissé la base des algorithmes de calcul dédiés à cette question. A cet effet, ont émergé plusieurs outils d'aide à la simulation dont un algorithme de calcul particulier, base des recherches actuellement [Gortmaker, 2011, Hall et al, 2018], Walkscore. L'algorithme est également disponible à travers le site : [walkscore.com](http://walkscore.com)



**Fig 3.1** – Walkscore : de la théorie à l'algorithme

Lorsque l'informaticien Matt Lerner met au point cet algorithme en 2007, sa visée était fondamentalement la promotion des voisinages marchables. Il est considéré comme un modèle d'efficacité automatisé (A.E.M), en outre, un modèle mathématique permettant d'estimer la valeur d'une propriété ou d'un espace, et de le mettre en relation avec d'autres. Il s'agissait alors d'une vitrine au visiteur de s'informer sur son prochain espace de vie. Pour les créateurs de l'algorithme, connaître la marchabilité de son voisinage, est un paramètre tout aussi important que l'espace architectural lui-même.

Cette évaluation est rendue possible par l'avancée technologique des méthodes SIG. A cet effet, Walkscore se base principalement sur les données de Google/Open Street Map et d'autres sources, telles que les agences de transport, pour mettre en application la relation mathématique, développée par les théoriciens. En 2014, La compagnie immobilière Redfin s'approprie l'algorithme et il devient à portée économique.

## 3.2 | Mode opératoire de l'algorithme

### 3.2.1 | Echelle du calcul et « Heat Maps » :

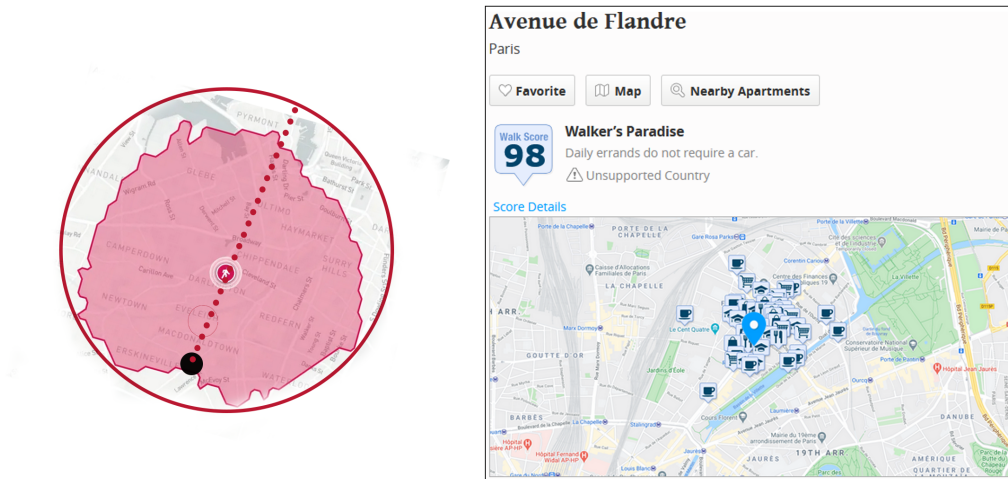
Comme son nom l'indique, ce qu'offre walkscore à l'utilisateur, est principalement un score de marchabilité, gradé de 0 à 100, et ce, par rapport à une adresse donnée. A partir de ce point d'origine, l'algorithme calcule des isochrones<sup>1</sup> et dresse donc une certaine limite géographique au calcul.

Intervalle de Score	Description spatiale
<b>90- 100</b>	<b>Le paradis du marcheur</b> : le recours à l'automobile n'est pas nécessaire
<b>70-89</b>	<b>Très marchable</b> : arpentable pour la plupart des destinations
<b>50-69</b>	<b>Assez marchable</b> : certaines destinations sont accessibles à pieds
<b>0-25 et 25-49</b>	<b>Dépendance à l'automobile</b>

**Tableau 3.1** – Tableau des intervalles de score de l'algorithme et de la description correspondante.

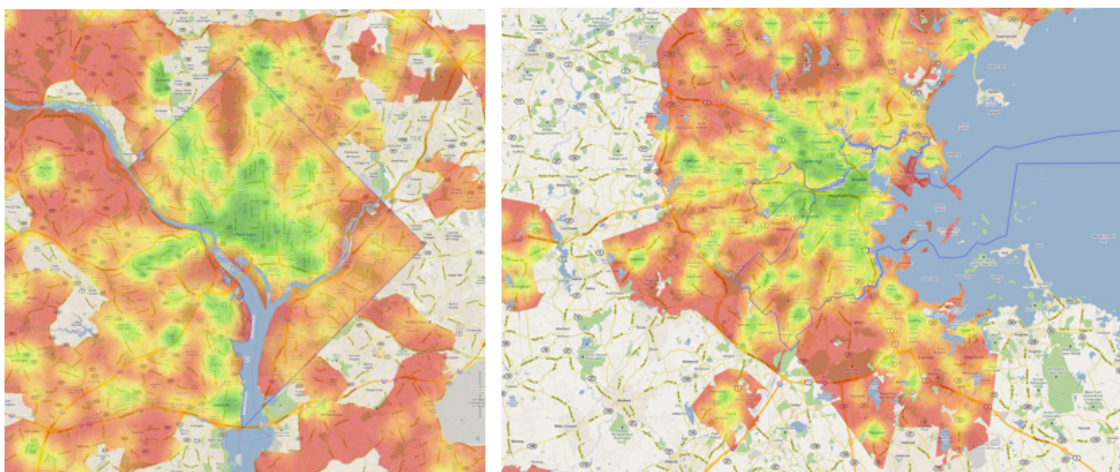
1. En géographie, se dit de cartes qui permettent de délimiter des aires accessibles par un moyen de déplacement pendant un temps donné.

Le score est donc local et n'est pas, à titre exemple, attribuable à des zones géographiques très étendues. Cette règle connaît certaines exceptions notamment pour le Canada ainsi que les États Unis, et ce, du fait que l'algorithme a émergé, initialement, dans ces régions précitées.



**Fig 3.2** – Limites du calcul et application sur l'Avenue de Flandre, à Paris.

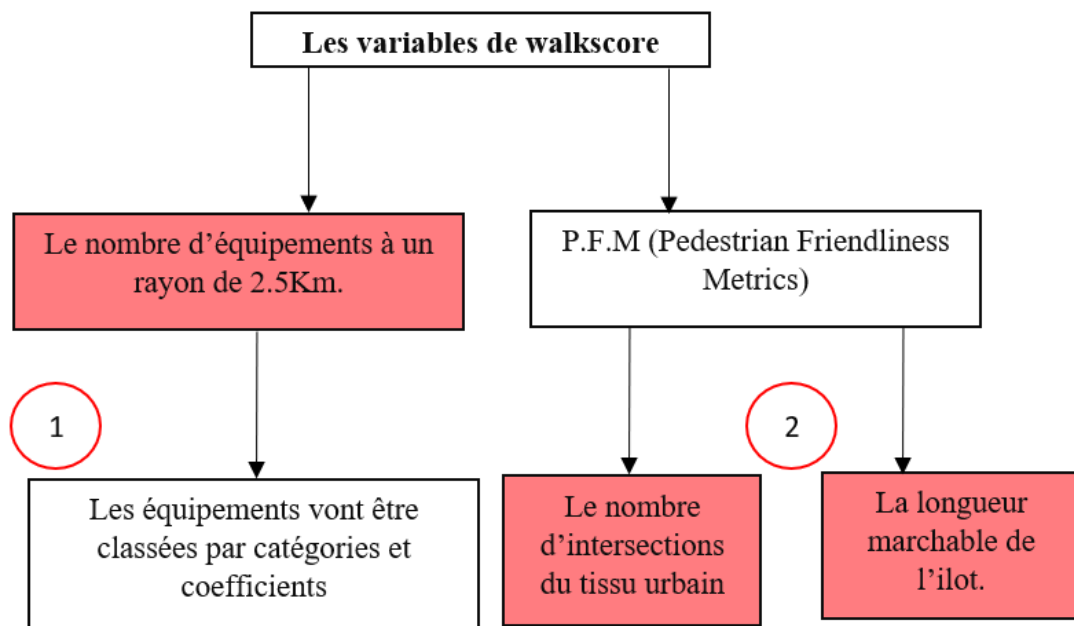
Ainsi à titre d'exemple, il est possible de se limiter à une recherche aussi globale que « New York » pour obtenir un score général, et en vue de l'échelle fine des données accessibles sur cet emplacement. L'algorithme établit même pour ces zones globales, des heat maps : Des cartes, dites de fréquentation, permettant de représenter l'intensité donnée de la valeur recherchée, par des nuanciers de couleur.



**Fig 3.3** – Les Heat Maps de Walkscore : Un autre moyen de représenter le score

### 3.2.2 | Les variables et la fonction de calcul :

Comme nous l'avons abordé dans la section [2.3.1.3], l'algorithme a repris la triade « équipements-intersection-longueur » consentie par les chercheurs de la marchabilité, comme un set de variables à introduire dans sa fonction de calcul. Cette dernière, bien qu'étroitement liée à l'équation de L.D Frank, n'est pas exactement la même. Il est, par ailleurs, possible d'exprimer cela comme suit :



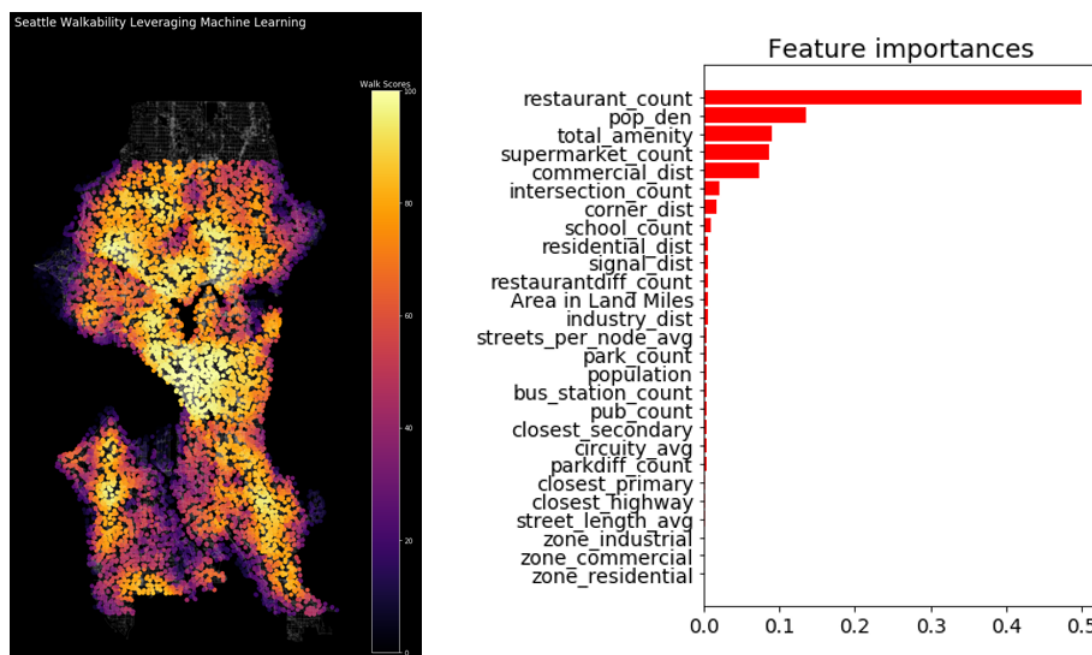
**Fig 3.4** – Variables reprises dans Walkscore

Les cadres en rose, dans la figure ci-dessus, indiquent les trois critères repris parmi les variables proposées dans la recherche théorique. Les critères relatifs à la configuration urbaine sont parfois désignés sous le sigle de P.F.M(Pedestrian Friendliness Metrics)[Pivo et Fisher, 2010].

Quant aux équipements, l'algorithme de calcul opère par une classification de ceux-ci, basés sur les enquêtes analogues de A.V Moudon. L'équation globale sera énoncée par la suite, mais les travaux en Machine Learning de l'ingénieur Perry Johnson, avec une rétro-ingénierie de l'algorithme, démontre que ce dernier, accorde bien une grande importance aux équipements :

Afin de déterminer l'équation globale de l'algorithme, nous allons aborder cela en deux par-

ties : Le calcul de la variable "équipements" et celui des variables relatives à la configuration urbaine.



**Fig 3.5** – Le travail de Johnson : Rétro-ingénierie et essai de prédiction du comportement de l'algorithme walkscore pour tenter de comprendre l'importance des variables qui le constitue

### 3.2.2.1 | La classification et les coefficients d'importance des équipements

Les équipements ne peuvent constituer un même noyau attracteur pour tous. Certains espaces seront jugés ainsi plus « attracteurs » et donc générateurs d'une plus grande marchabilité. Quant à quels seraient ces équipements et comment les classer, là encore l'algorithme s'appuie sur l'observation et le questionnaire réalisés par Moudon, pour classer l'importance des équipements ce qui conduit à la concrétisation de 9 catégories ou classes comme suit :

```

amenity_weights = {
    "grocery": [3],
    "restaurants": [.75, .45, .25, .25, .225, .225, .225, .225, .2, .2],
    "shopping": [.5, .45, .4, .35, .3],
    "coffee": [1.25, .75],
    "banks": [1],
    "parks": [1],
    "schools": [1],
    "books": [1],
    "entertainment": [1],
}

```

**Fig 3.6** – Partie de l'algorithme qui classe les équipements avec une "note"

Cette classification donnera lieu au concept des « Amenity Weights » et plus tard, dans l'outil paramétrique, sera nommée les « Amenity demand profile ». En d'autres termes, l'algorithme crée des classes spécifiques et attribue à chaque classe un ou plusieurs coefficients d'importance(C.I), selon leur fréquentation.

Ces coefficients d'importances de 0,2, pour le plus bas jusqu'à 3 le plus élevé, ne sont pas normalisés, c'est-à-dire qu'il y a une nécessité de retransposer cet intervalle [0,2 – 3] vers un intervalle de 0 à 100, puisque justement comme nous l'avions expliqué dans l'introduction de ce chapitre, le score, lui-même, se positionne sur cet intervalle.



Nous noterons, que selon le contexte d'étude, il existerait plusieurs sous-classes d'un même équipement. Par exemple, dans l'étude américaine de Moudon, les restaurants pourraient se décliner en plusieurs espaces restauratifs, d'où l'affectation de 10 facteurs(sous-classes) d'importances dans la catégorie "restaurants". Afin de transposer ces chiffres vers l'intervalle normalisé [0, 100], le calcul procédera comme suit :

$$CI(\text{normalisé de l'équipement}) = CI(\text{Equipement}) \times 100 / K \quad (1)$$

où :

$CI(\text{normalisé})$  = Coefficient d'importance/Amenity Weight dans un intervalle [0-100]

$CI(\text{équipement})$  = Coefficient d'importance/AW dans un intervalle [0,2-3]

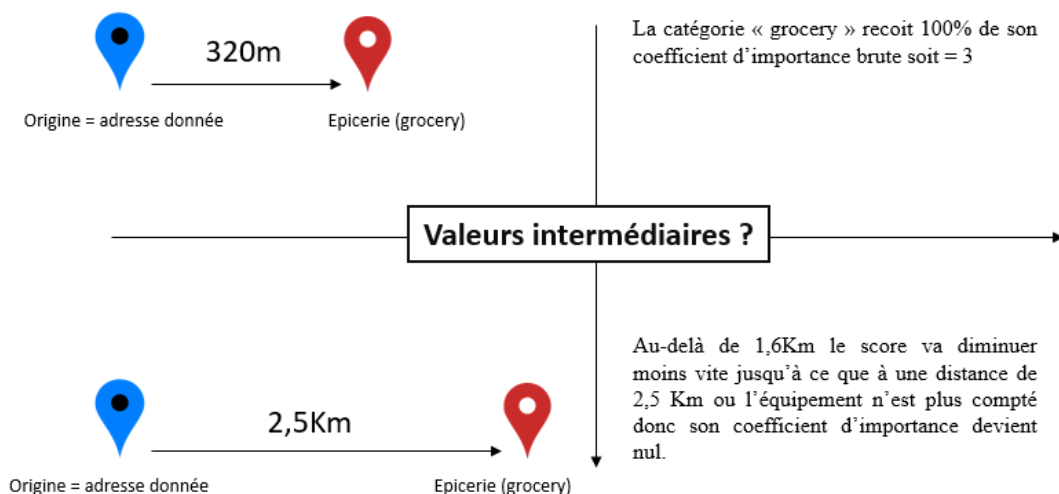
$K$  = constante = 15 qui est la somme totale des CI dans l'algorithme

A titre d'exemple,voici comment l'algorithme normalise le facteur d'importance de l'épicerie(Grocery) :

### Exemple

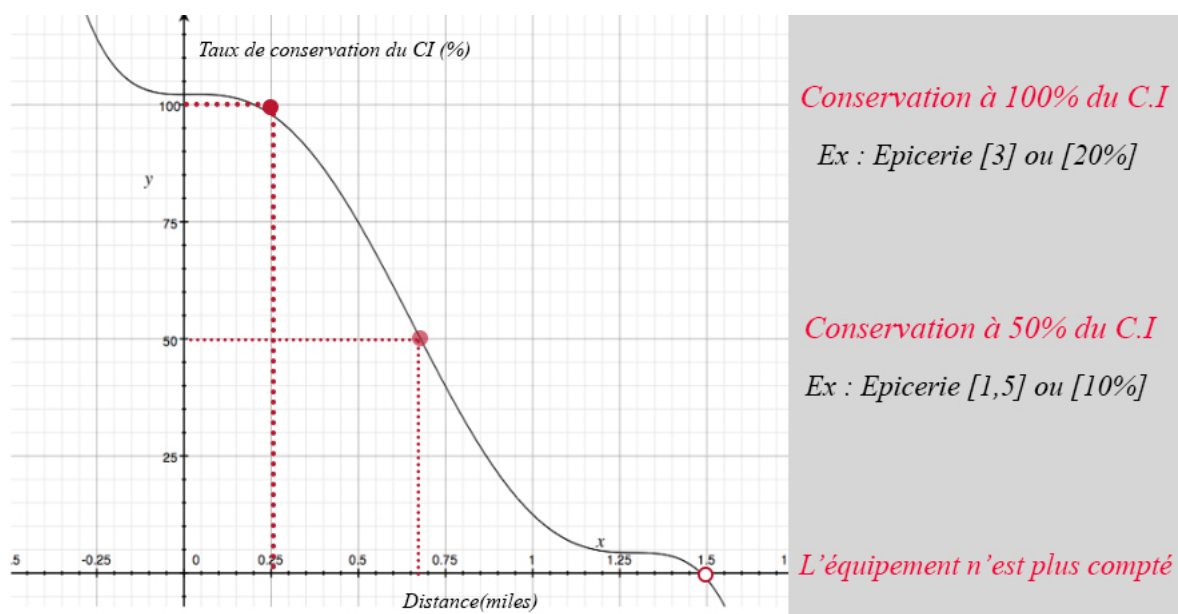
$$CI(\text{normalisé-épicerie}) = 3 \times 100/15 = 20$$

Le C.I normalisé 20 est donc le coefficient d'importance transposé à l'intervalle [0,100].C'est d'ailleurs la valeur maximale, que pourraient s'attribuer tous les équipements. Cependant, les facteurs décrits ne sont pas automatiquement attribués,du simple fait de la présence de l'équipement. En effet, la distance depuis l'origine vers cet équipement est elle aussi à prendre en compte et peut réduire le C.I comme suit :



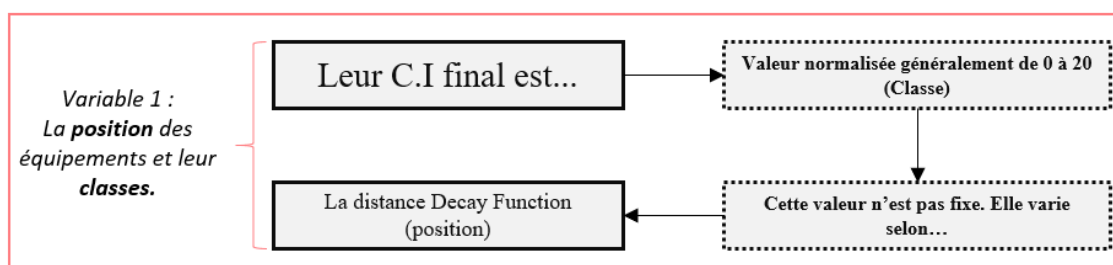
A une distance minimale référence de 320 mètres, l'équipement en question recevra son coefficient d'importance complet(100% donc 3 normalisé 20), ensuite l'algorithme impose une limite maximale de la distance qui est de l'ordre de 2.5-2.6 kilomètres. C'est une distance au-delà de laquelle le coefficient d'importance de l'équipement est nul et ce, même s'il est représenté dans la limite géographique de calcul.

Entre ces deux limites distancielles, subsiste alors la question de savoir comment traiter les valeurs des C.I intermédiaires(pour le cas de l'épicerie, entre 3 et 0). C'est là qu'intervient l'usage de la «décroissance ou atténuation avec la distance». Il s'agit d'un terme pouvant désigner l'impact de la distance sur les interactions sociales ou le comportement humain. Cette désignation est traduite par une courbe mathématique sous le nom de la «distance decay function» comme suit :



**Fig 3.7** – Courbe de la distance decay function pour l'obtention de valeurs intermédiaires des C.I

L'axe des ordonnées montre des pourcentages de 0 à 100. Il ne s'agit pas des valeurs normalisées mais plutôt de montrer le pourcentage de conservation de la valeur initiale du C.I, par rapport à une distance(axe des abscisses). Entre ces deux valeurs limites relevées(320m/0,25Miles et 2.5m/1.5Miles), la projection de la courbe de décroissance permet d'avoir les valeurs médianes automatiquement.



**Fig 3.8** – schéma de la construction de la variable 1 selon la méthode walkscore

### 3.2.2.2 | La pénalisation du P.F.M : L'influence de la forme urbaine dans le calcul

L'intersection et la longueur de l'îlot sont soulignés comme paramètres déterminants dans l'équation de la marchabilité [Ewing,2003, Moudon et al,2006 et Berrigan et al,2010]. Ces deux variables ne sont pas introduites exactement comme dans l'équation de Frank, dans la recherche théorique sur les indicateurs. Elles font office d'une "sanction" à la première variable, à savoir, le coefficient/facteur d'importance calculé des équipements, après normalisation.

$$\text{Ind. Marchabilité/Score (Walkscore)} = \text{Score équipements} + \% \text{PFM} \quad (2)$$

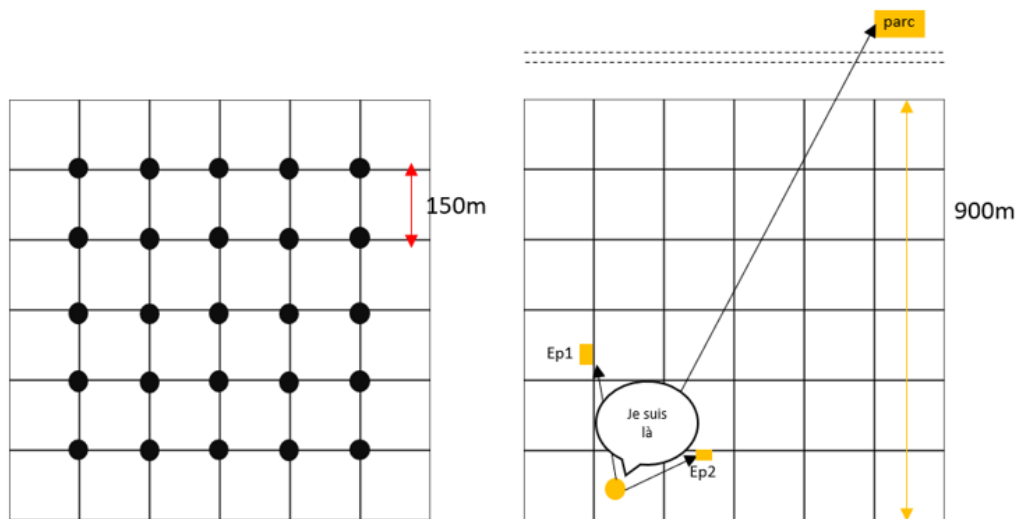
La soustraction d'un pourcentage P.F.M est opérée par le fait que l'algorithme traite la forme urbaine comme une «récompense» ou une «pénalisation» et ce, selon certaines limites. Ces dernières ainsi que leur pourcentages correspondants, sont décrites dans le tableau ci-dessous.

Le P.F.M : Pénalisation de l'environnement bâti (Intersection-longueur de l'îlot)			
Nombre d'intersection	Pénalisation en %	Longueur Ilot	Pénalisation en %
< 200 (nœuds)	Pas de pénalisation	< 120m	Pas de pénalisation
150-200	1%	120-150m	1%
120-150	2%	150-165m	2%
90-120	3%	165-180m	3%
60-90	4%	180-195m	4%
> 60	5%	> 200m	5%

**Tableau 3.2** – Pourcentage des sanctions PFM(variable 2) attribués à la variable 1 selon la forme urbaine

Ci-dessous, un exemple simple pour illustrer le score final de la marchabilité. Dans ce plan

local schématique, nous avons un nombre d'intersections(les nœuds) inférieur à 60 donc la pénalisation va être maximale soit de l'ordre de 5% pour ce paramètre-là. Tandis que pour le second paramètre du PFM qui est la longueur du bloc, celle-ci est de 150 mètres donc la pénalisation est de 2% pour ce paramètre-ci. La forme urbaine aboutit donc à un total de 7% de pénalisation, soit 7% à déduire du score des équipements.



**Fig 3.9** – exemple simple d'application de l'algorithme

Nous avons 2 épiceries à moins de 320 mètres de l'origine et un parc situé à l'extrémité de ville à une distance de 1,7Km. Par projection sur la courbe de la fonction de décroissance(Fig 3.7) de cette distance-là, le CI brut du parc [1] est réduit de 12% et devient équivalent à 0,12 puis normalisé selon la formule (1) vers :

Exemple
$CI(\text{normalisé du parc}) = 0,12 \times 100 / 15 = 0,8\%$
$CI(\text{normalisé de l'épicerie}) = 3 \times 100 / 15 = 20\%$

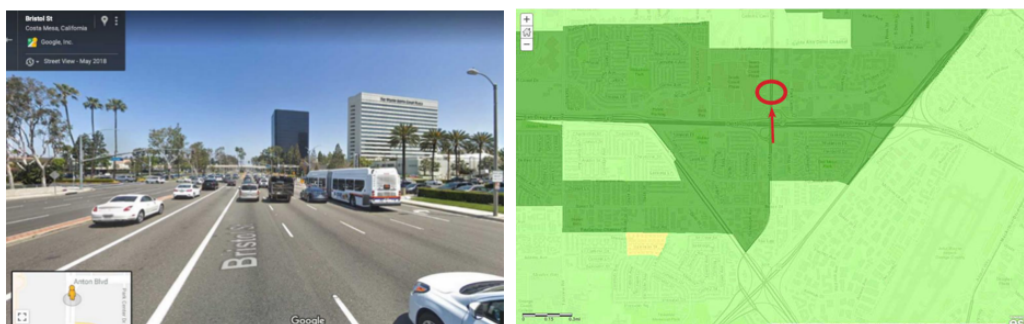
Étant donné la présence de 2 épiceries, dans cet exemple, le score global des équipements équivaut à 48%. Il est, ensuite, pénalisé par les 7% du PFM. Ceci aboutira au résultat suivant :

Exemple
Score de marchabilité(Walkscore) = $48\% + (48 \times (-7\%)) = 44,46\%$
Classification : zone dépendante à l'automobile

### 3.3 | Limites de l'algorithme

La principale limite de l'algorithme, ainsi que de tous les outils qui s'en inspirent, est le manque de classification afférentes aux voiries et cela a constitué la critique des recherches sur cet outil [Cubukcu et al, 2014] . Tout comme nous avons pu constater, la classification des équipements, nous permettait d'avoir une certaine hiérarchie et un système de notation adéquat, même si, éventuellement, il demeure relatif uniquement à certains contextes et cultures. (L'importance d'un équipement E n'est pas la même pour tous les pays ou même pour des régions d'un même pays).Le fait de procéder à un calcul, basé uniquement sur la représentation graphique (densité des intersections/longueur des îlots), sans modèle renseigné pourrait induire des erreurs. Nous citerons l'exemple de l'autoroute de Saint Bistol, en

Californie. Celle-ci est traitée, par les algorithmes d'évaluation de la marchabilité, comme étant marchable or la réalité du terrain, présentée dans figure ci-dessous gauche, dénote le contraire. Un système de « Tag » ou une certaine sémantisation du modèle graphique, qui serait pris en compte dans le calcul, aurait permis d'évincer cette erreur.



**Fig 3.10** – Erreurs de calcul dues à la déficience des critères pris en compte : à gauche : l'autoroute de St Bistol et à droite : l'algorithme présentant l'autoroute comme marchable.

## 3.4 | Résumé et conclusion partielle

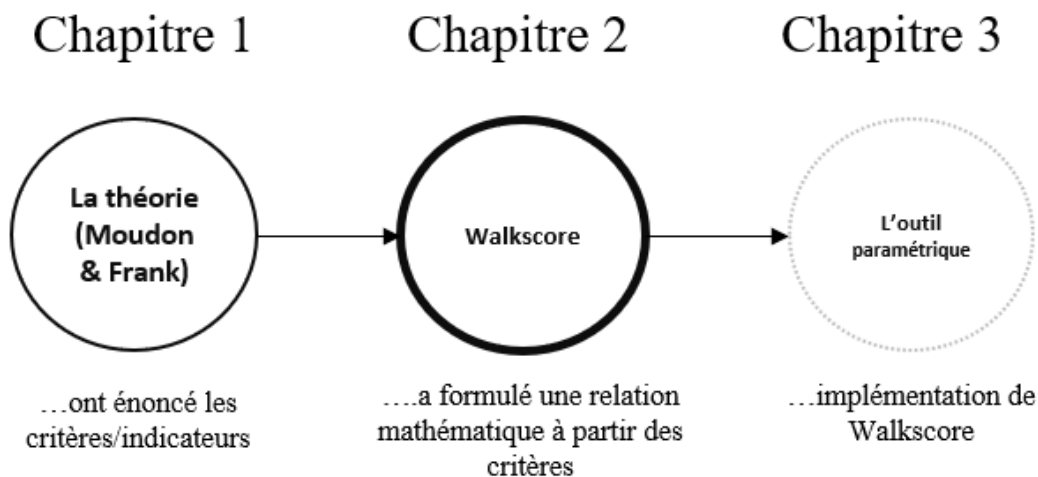
### Pour résumer...

L'algorithme de Walkscore donne un score de marchabilité **gradé de 0 à 100, à partir d'un point donné**. Pour effectuer ce calcul, l'algorithme met en jeu **les 3 critères** présentés dans le chapitre précédent à savoir : La **diversité/nombre d'équipements** ainsi que l'influence de l'environnement bâti(**PFM**), qui elle, se décline en 2 paramètres : Le **nombre d'intersection du réseau viaire(nœuds)** et la **longueur de l'îlot**. Ces 3 critères sont liés par la relation suivante :

$$\text{Score WalkScore(marchabilité)} = \text{Score des équipements} + \% \text{ PFM}$$

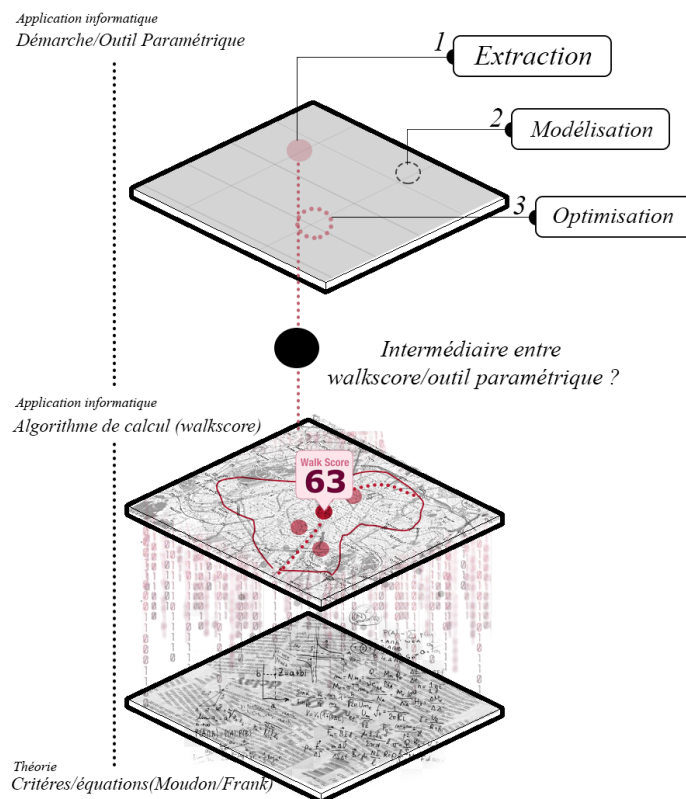
- Le score des équipements est une « note » affectée selon leur importance/contexte et leur proximité à un point donné et ce, jusqu'à 2,3Km (Distance Decay Function)
- Le % du PFM est une « sanction » à la « note » des équipements : l'algorithme retire des points en % , selon le nombre d'intersection/longueur de l'îlot du contexte étudié.

Cette relation mathématique sera implémentée par la suite dans le module de calcul « Urbano ».Ce dernier sera utilisé, comme « plugin », dans l'outil paramétrique et introduit dans le chapitre suivant.



# Modélisation et Optimisation de la marchabilité dans l'outil paramétrique

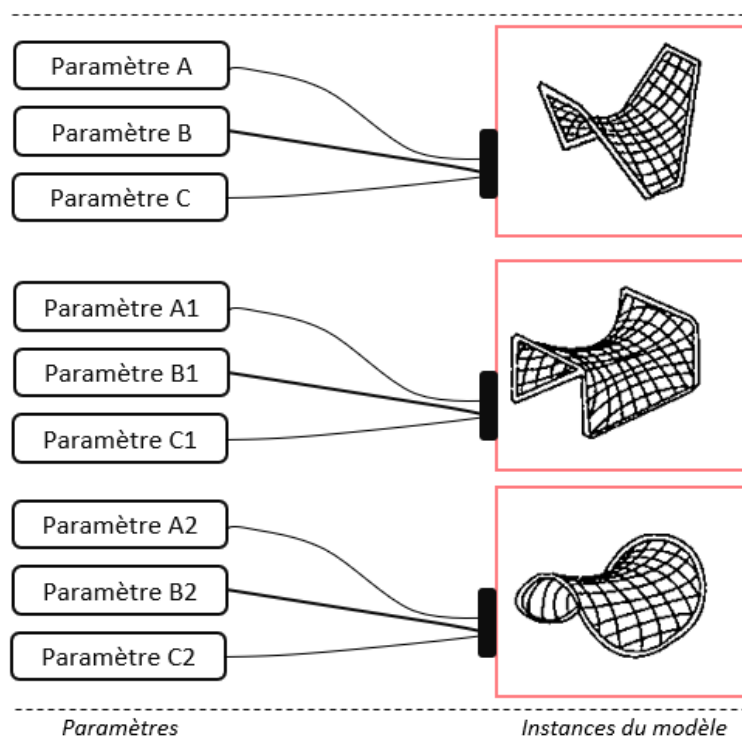
## 4.1 | La démarche paramétrique : Nouveau paradigme de la conception



**Fig 4.1** – Schéma de la démarche suivie dans ce chapitre

Outre la démarche empruntée par la space syntax qui traduit la ville par la théorie des graphes, l'avènement de la conception paramétrique permet une lecture de décomposition d'un système en éléments modifiables, les paramètres. Ceci aboutit à un changement de paradigme dans les travaux de simulation ou d'optimisations informatiques.

Dans cette méthodologie, il existe une mise en relation entre plusieurs variables d'un même modèle, et dont l'influence ou la modification, pourrait générer, dans un espace dit de recherche plusieurs variantes morphologiques, environnementales, matérielles etc. Ces dernières pourraient être qualifiées d'instances de modèle[Tedeschi, 2014].



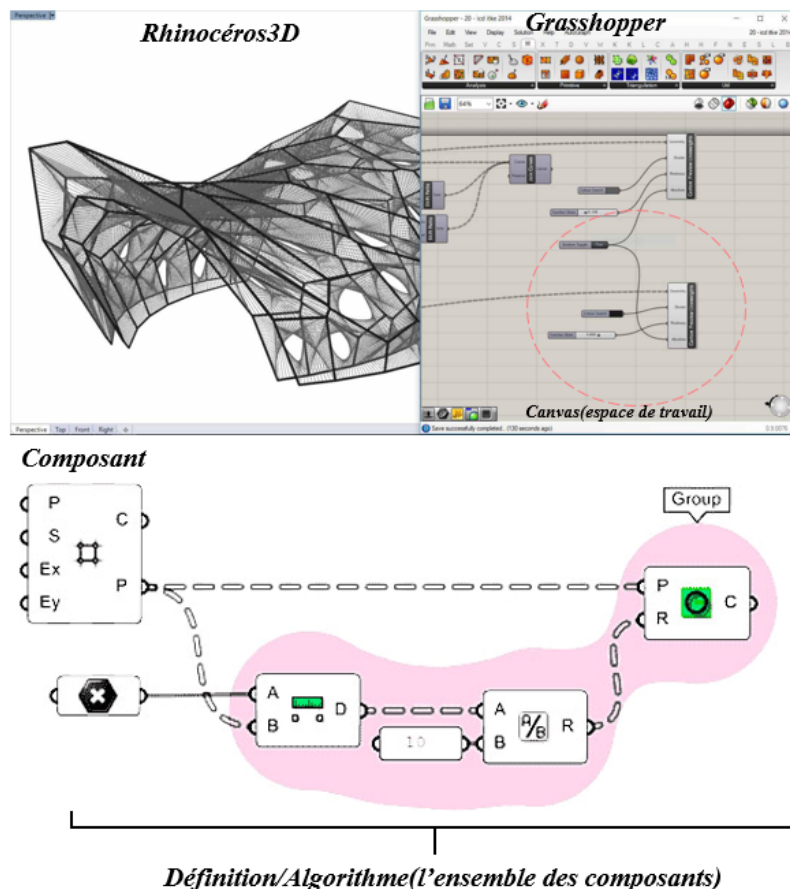
**Fig 4.2** – Exemple de décomposition paramétrique d'un même modèle et génération de variantes

#### 4.1.1 | L'outil paramétrique

Il ne s'agit pas, dans ce mémoire, d'aborder les modeleurs qui disposent d'une bibliothèque et de classes préconçues, tels que Autodesk Revit, et dont les opérations, bien que qualifiées de « paramétriques », restent soumises à un certain pré-réglage inaccessible à l'utilisateur. Les modeleurs dits libres seront le support de ce travail et dont nous citerons le plus accessible actuellement : Rhinocéros3D et son extension intégrée Grasshopper, développés respectivement par les architectes Robert McNeel et David Rutten. Ce dernier a, à travers Grasshopper,

crée un environnement de programmation visuelle permettant de s'approprier la démarche paramétrique et le design génératif.<sup>1</sup>

La programmation visuelle le rend encore plus accessibles aux architectes et notamment la représentation de l'espace de travail qui permet de garder un certain historique de l'opération algorithmique entière, dite *définition paramétrique*. Cette dernière est le résultat d'un ensembles de *composants*, qu'ils soient natifs ou personnalisés, effectuant une série d'opérations en boîte noire.



**Fig 4.3** – En haut : Interface de Grasshopper accouplée à celle de Rhino et en bas : illustration d'un ensembles de composants(définition)

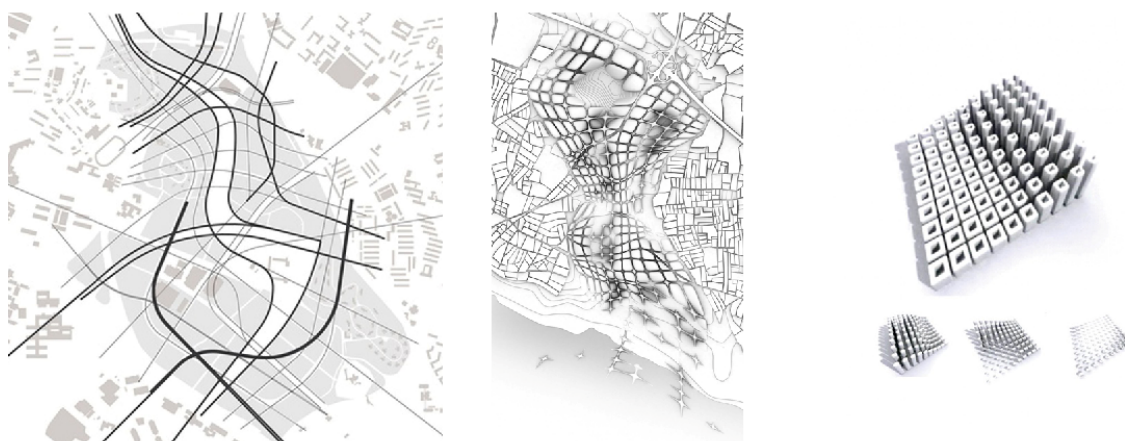
1. Méthode conceptuelle itérative faisant appel aux outils numériques pour générer de sorties et ce, à travers des paramètre définies par l'utilisateur.

## 4.2 | L'outil paramétrique pour la modélisation de l'urbain :

La notion de "parametric urbanism" est souvent attribuée aux travaux et aux écrits des architectes Patrik Schumacher et de Zaha Hadid[Leach, 2009]. Le processus de conception paramétrique, à cette échelle, implique de positionner l'urbanisme comme un système dynamique ou un métabolisme et nous pouvons constater ceci dans les travaux de Z.H Architects sur les plans de One North Masterplan et Thames Gateway .

L'implication de ceci est que la rigidité des modèles est "diluée", au profit de l'interaction de plusieurs paramètres, qui non seulement définissent les relations dans le projet urbain mais aussi sa "coopération" avec le contexte[Kolarevic,2005]. L'architecte Rivika Oxman développerait cette idée et soulignerait que la performance du projet serait elle même un paramètre dans cette approche.

Dans ce mémoire, l'atout que présente l'outil et l'approche paramétrique réside également dans le fait de pouvoir combiner les diverses approches et données qui sont afférentes à l'approche syntaxique et l'approche GIS afin de modéliser la marchabilité. Il s'agit d'une démarche qui prône la complémentarité des méthodes que Ratti avait abordé et que nous évoqué dans la section [2.2.2.4].






**Fig 4.4** – Travaux de Zaha Hadid Architects, à l'échelle urbaine, avec la démarche paramétrique : (a) : One North Masterplan (b) Kartal Pendik Masterplan et (c) Evolution du mod.urbain de Thames Gateway Masteplan

### 4.2.1 | L'extraction des données géographiques par l'outil paramétrique

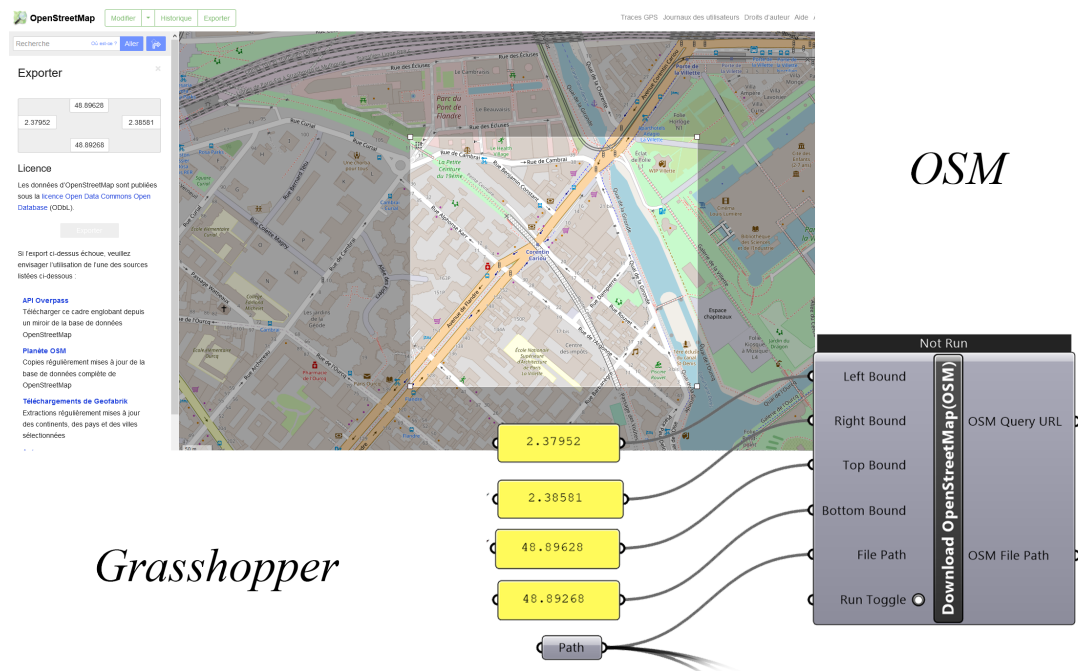
Comme schématisé dans la figure 4.1, cette partie sera abordée en trois étapes : extraction-génération du modèle de calcul et enfin optimisation de celui-ci. Avant Janvier 2020, l'extraction des données géographiques depuis des systèmes GIS, tel que Open Street Map, dans Grasshopper, s'effectuait à travers un plugin du nom de "Elk", mis au point par l'architecte Timothy Logan.

L'extraction était donc, auparavant, un processus séparé du modèle de calcul, qui lui consistait en l'introduction, par l'utilisateur, de la fonction "walkscore" que nous avons abordé dans le chapitre précédent.

	Extraction	Mod. du modèle de calcul	Optimisation
Etat de l'art	 Plugin Elk	<div>Score (Walkscore)=Score équipements + %PFM</div> Fonction objectif introduite par l'utilisateur	<div>Algorithme mono-objectif</div> Opt 2D/Minimiser le score par : -la réduction de la distance des équipements vis-à-vis des habitants
Méthode Mémoire	 Plugin Urbano	 Intégré comme composant de calcul	<div>Algorithmes mono-objectif et multi-objectif</div> Opt3D/Minimiser le score selon d'autres méthodes

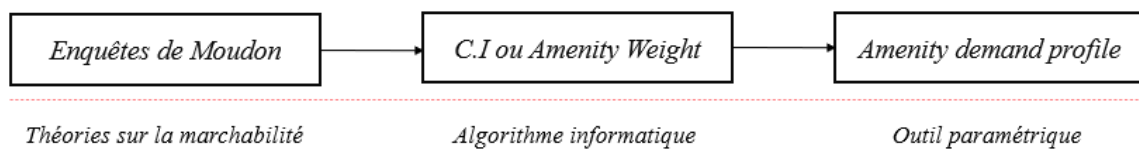
**Tableau 4.1** – Comparaison de la méthode actuelle avec celle empruntée par ce mémoire selon les différentes étapes du travail

Urbano, un plugin qui sera utilisé dans ce mémoire, a été crée en Janvier 2020, par le docteur Timur Dogan, à l'université de Cornell dans le département de génie civil. Entièrement dédié à la question de la mobilité urbaine, il réunit donc les deux fonctions d'extraction des données depuis le système d'Open Street Map ainsi celle de la génération d'un modèle de calcul, soit l'intégration de la fonction walkscore.

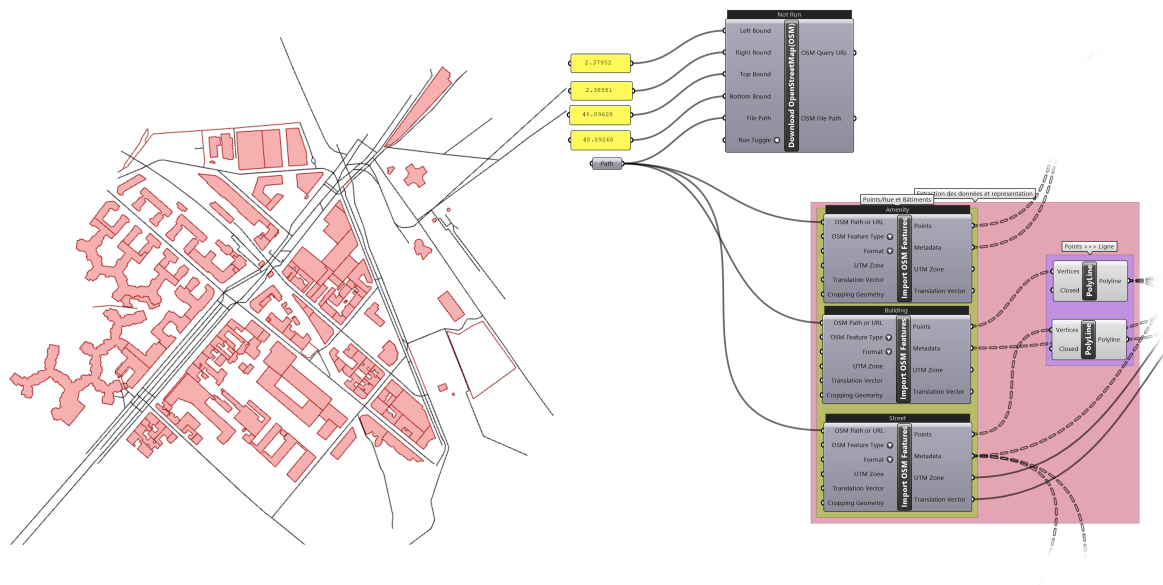


**Fig 4.5** – Extraction depuis OSM à travers le composant *Download OSM* d'Urbano qui permet d'intégrer des données GIS dans l'outil paramétrique

Comme pour son analogue, le plugin permet la récupération des données mais également leur classification par "Bâtiments résidentiels, Équipements et Rues". Cette classification, entre autres, permettra la construction du modèle Urbano. Ce dernier, pourra donc fournir, autres que les informations sur les rues, les *Amenity Demand Profile*, qui sont l'équivalent des Amenity Weight ou Coefficients/facteurs d'importance, basés sur Walkscore et évoqués également dans la section [3.2.2.1].



**Fig 4.6** – La notion d'ADP qui permet la construction du modèle Urbano et son développement à travers la recherche

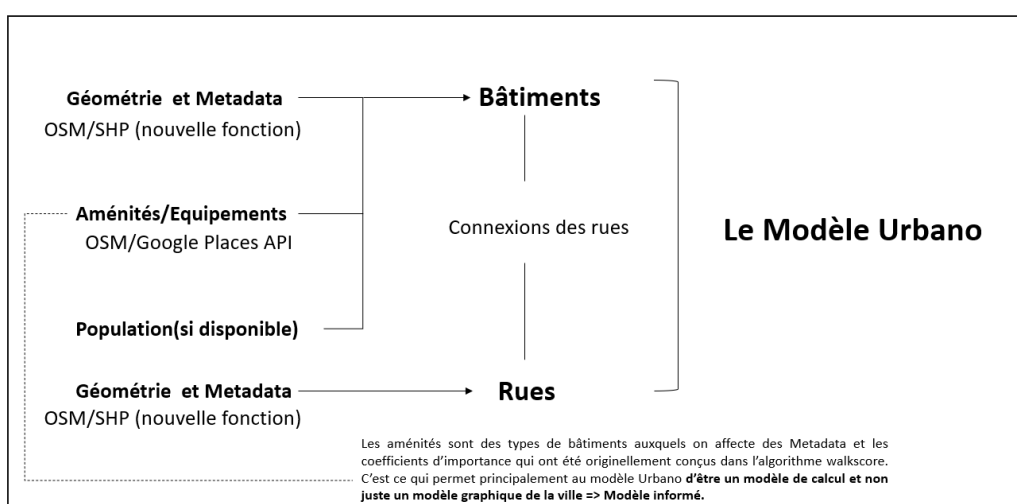


**Fig 4.7** – Extraction des données géographiques par les 3 composants principaux

#### 4.2.2 | La modélisation et simulation du modèle de calcul

Plus précisément, Le modèle géométrique devient un modèle informé Urbano par les deux types de composants suivants :

- Les composants MetaData (Sémantisation)
- Les composants d'affectation (Calcul numérique)



**Fig 4.8** – Composants participant à la construction du modèle de calcul Urbano

Le plugin attribue donc principalement des tag ou un système de classes, puisés depuis OSM, qui permet à l'utilisateur d'extraire des informations sur le graphique mais également d'y rajouter sa "géométrie", en y attribuant à son tour des méta données, tant que celles-ci correspondent à la classification effectuées par la bande de données d'OSM.

Un exemple de géométrie à rajouter et qui sera utilisé dans les expériences de ce mémoire, est le point(Position X,Y des équipements). Comme le montre la figure suivante, la géométrie affectée à des méta données des équipements(une classe + un C.I) est introduite dans le modèle, par le biais du composant **Create Amenity**. Puis convertis en "bâtiments", les points ajoutés sont "acheminés" dans le composant **Build Urbano Model**.

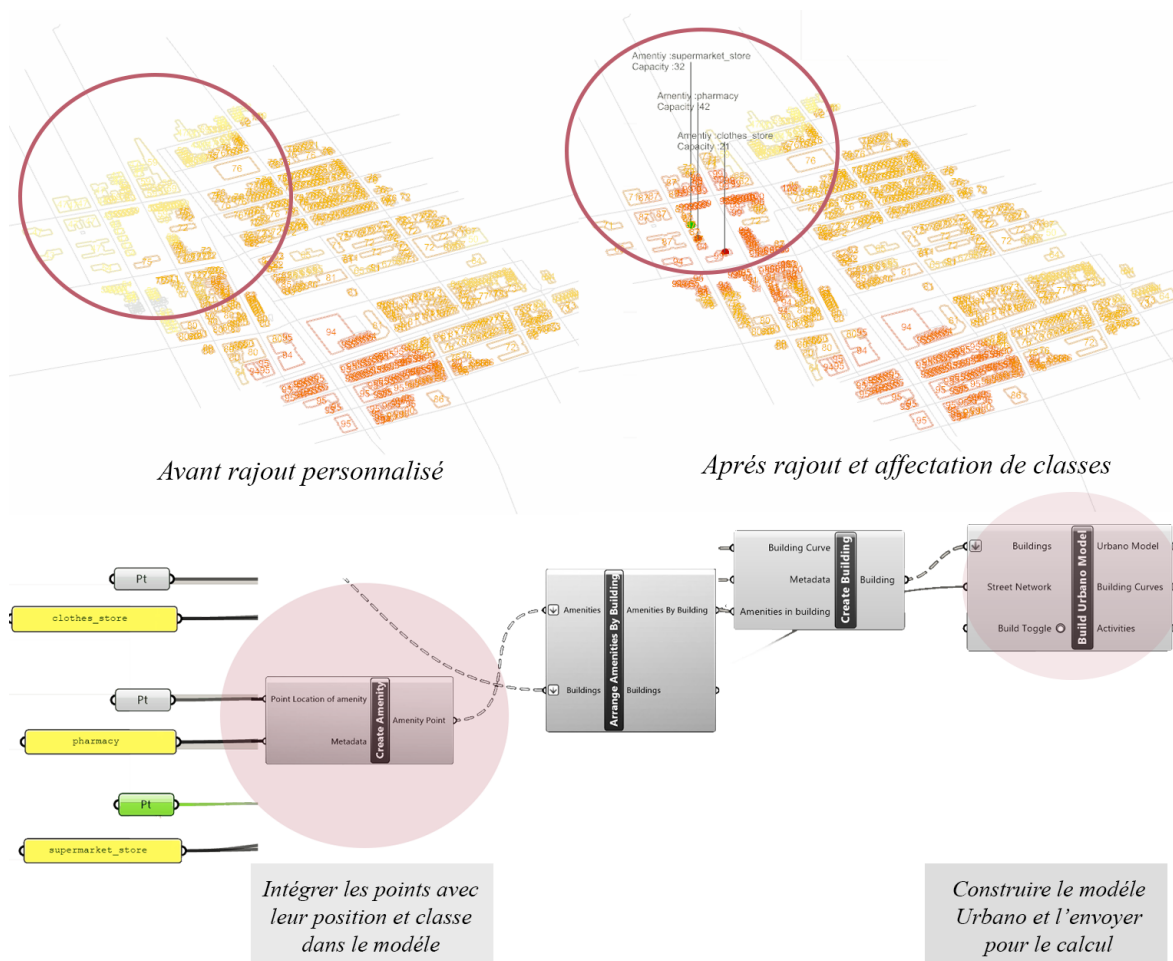
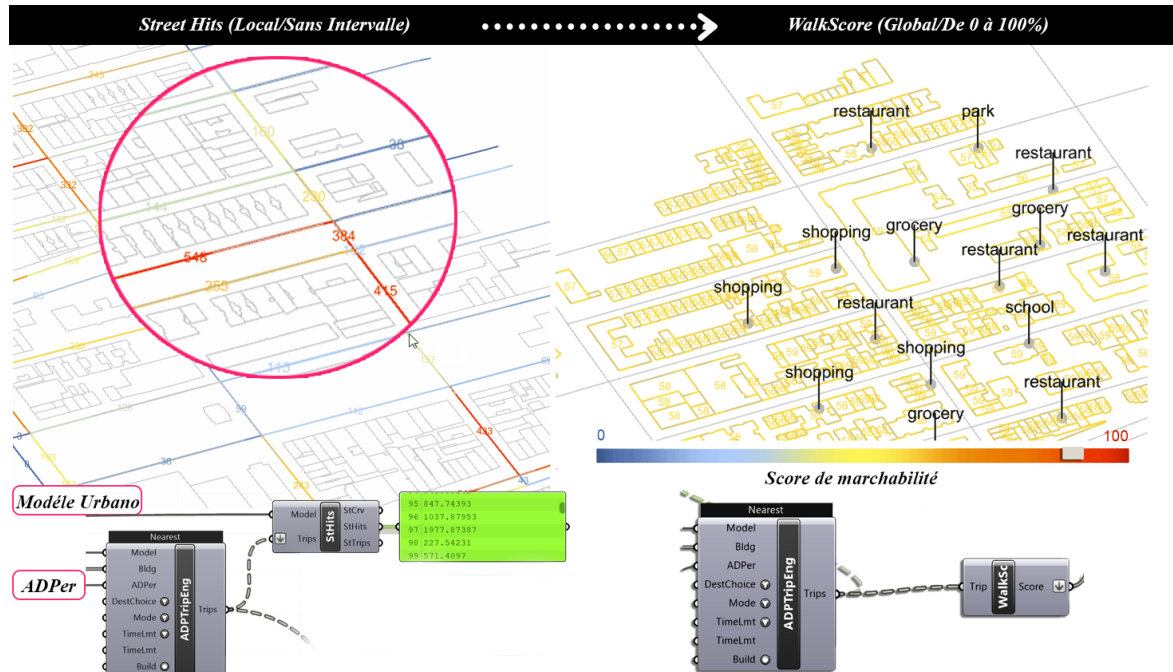


Fig 4.9 – Intégration de géométrie personnalisée, avec classes, dans le modèle de calcul

Les méta données(sémantisation) introduites nécessitent les composants de calcul pour avoir une certaine valeur numérique(affectation). Pour ajouter des coefficients/facteurs

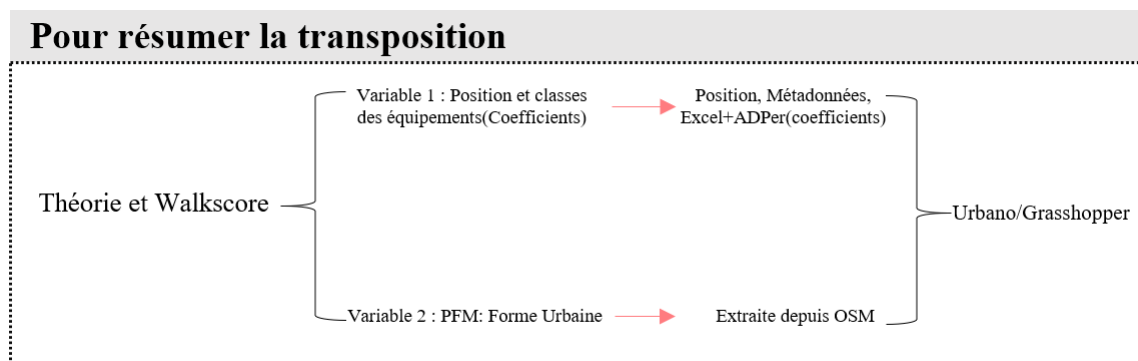


normalisée à travers le composant *StreetHits* et une mesure globale en pourcentage à travers le composant *Walkscore*.



**Fig 4.11** – Les deux mesures de calcul la marchabilité sous Urbano : A gauche StreetHits(local) et à droite Walkscore(global)

Dans les deux cas, les mesures obtenues sont à convertir en moyenne, lors des expériences, et ce, à travers le composant natif de Grasshopper : **Average**.



---

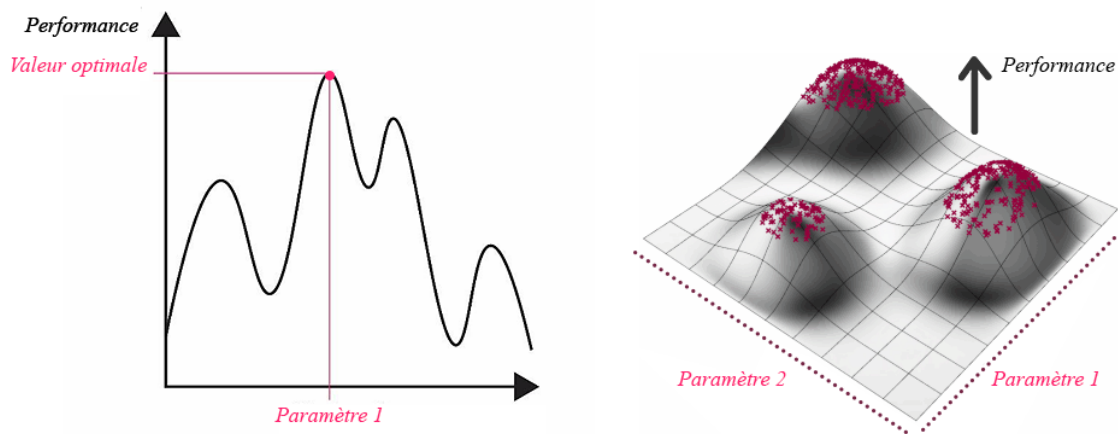
### 4.2.3 | L'optimisation dans l'outil paramétrique par les algorithmes bio-inspirés

#### 4.2.3.1 | L'optimisation face à la complexité de la conception urbaine :

L'optimisation est la modélisation d'un problème dans lequel il s'agit de trouver la solution appelée « optimum », en prenant en compte des paramètres et des contraintes à satisfaire.

Devant la complexité des problèmes de conception urbaine ou architecturale, la relation entre ces divers paramètres n'est pas toujours explicite et il s'agit là de procéder par une optimisation dite en boîte noire [Dissaux,2016]. Le type de problème ainsi que le nombre de dimensions des paramètres ou de la fonction objectif sont des critères déterminants dans le choix de la méthode d'optimisation ainsi que de l'algorithme afférent.

Celui-ci doit procéder à la recherche d'un paysage de solutions, et dans ce paysage abstrait, l'optimum serait le pic de la fonction objectif formulée par l'utilisateur. La notion de paysage a été dérivée de la simplification d'un problème avec des paramètres dans un espace à 3 dimensions[Rutten, 2013]. Pour un couple de paramètres, correspondrait, donc, une valeur sur l'axe Z qui est la valeur optimale pour ce couple là.<sup>2</sup>



**Fig 4.12** – Recherche de l'optimum et représentation : A gauche : Cas d'un paramètre et à droite : Cas de deux paramètres dans un paysage de solution

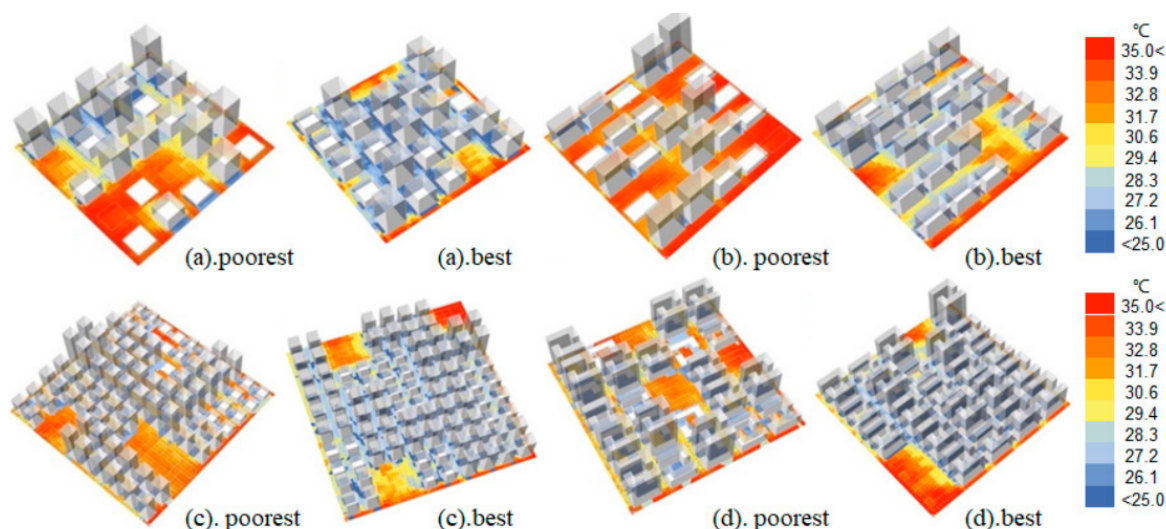
Comme énoncée dans l'introduction, l'optimisation est devenue accessible aux architectes à travers l'introduction des outils paramétriques. La littérature scientifique dénote l'attrait des

---

2. Rutten.D dans *Galapagos : On the Logic and Limitations of Generic Solvers*,2013

concepteurs de l'espace à intégrer ce processus d'optimisation, dans leur démarche conceptuelle et même comme un moyen de générer des esquisses [Khabazi, 2012 et Wortmann, 2018].

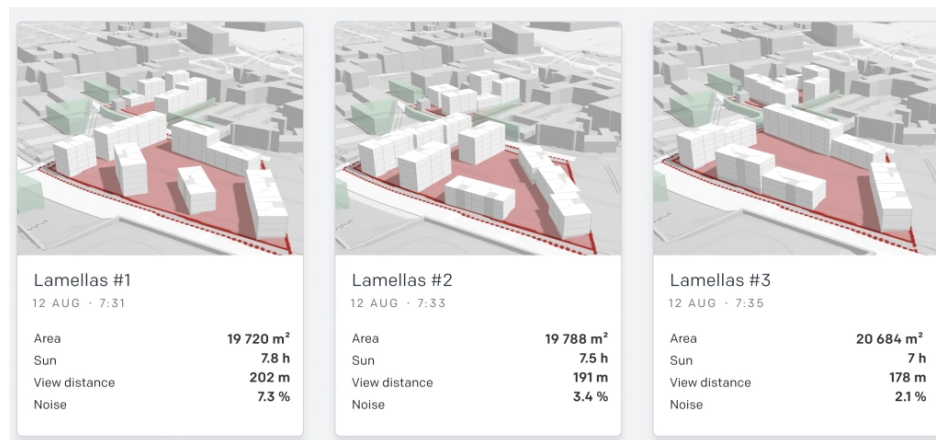
Ce processus exploratoire concerne surtout des recherches formelles désignées par « form finding » ou encore la recherche de certaines morphologies urbaines [Wang et al, 2019]. Dans ces contextes là, l'optimisation se rapporte souvent à des simulations du CFD : le Computational Fluid Dynamics. Cela implique, par exemple, d'adapter l'enveloppe ou l'orientation du bâtiment face à certaines contraintes climatiques.



**Fig 4.13** – Recherche d'une morphologie urbaine face à des données environnementales extraites de l'étude de Wang et al, 2019

Bien évidemment, les outils paramétriques ne sont pas les seuls à nous permettre d'effectuer des optimisations sur le contexte urbain. D'autres outils plus spécialisés et isolés permettent d'effectuer cette recherche de l'optimum. Nous citerons l'entreprise Space Maker<sup>3</sup> qui développe des outils, basés sur l'intelligence artificielle, afin de permettre aux concepteurs de l'espace de générer des solutions urbaines et d'expérimenter plusieurs variantes.

3. [www.spacemakerai.com/](http://www.spacemakerai.com/)



**Fig 4.14** – Recherche de la disposition optimale de bâtiments dans un îlot urbain par rapport à des critères tels que l’apport solaire et la vue -Travaux de spacemaker

Cependant, l’optimisation d’un comportement à l’échelle urbaine, telle que la marchabilité, est encore une démarche difficile et peu abordée. D’une part, les outils qui permettent la formulation du comportement sont encore peu développés. D’autres part, le comportement, même s’il peut être simulé, reste relativement sujet à un biais de confirmation. Néanmoins, nous nous attachons à construire une démarche « objective » de la simulation de la marchabilité, dans ce mémoire, comme une première approche dans l’outil paramétrique.

Face au panel de problèmes existants, les algorithmes d’optimisations, eux-même, peuvent converger vers diverses solutions[Wortmann,2018].C’est ainsi que nous aborderons la question du choix de l’algorithme.

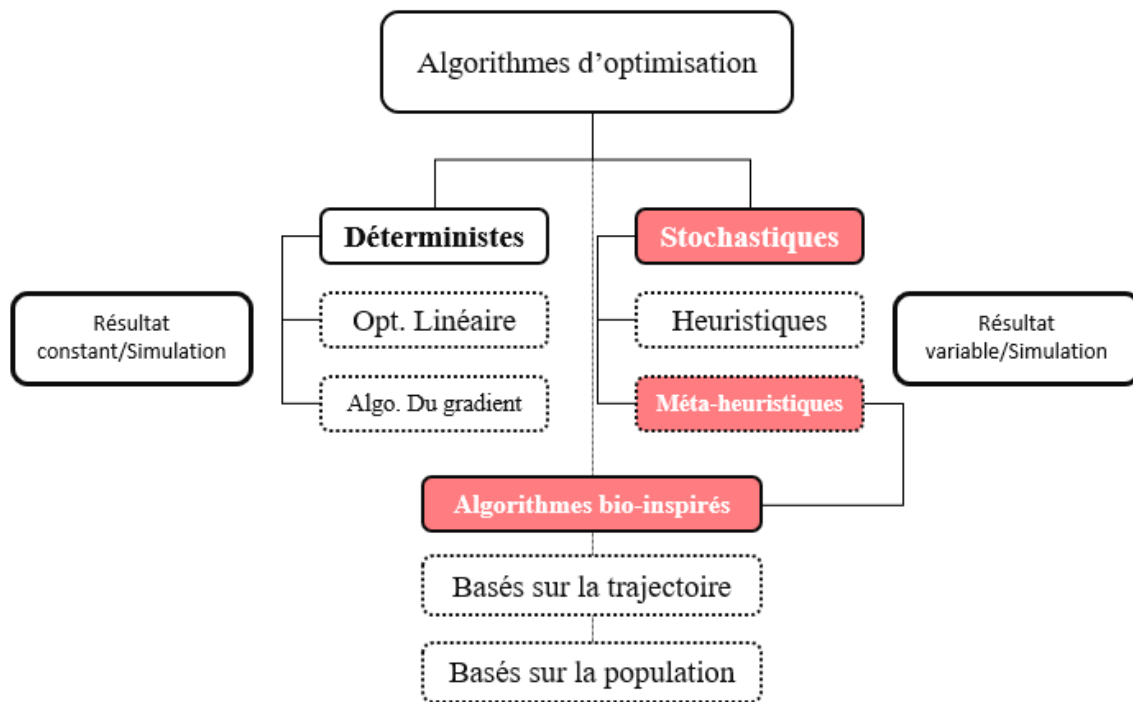
#### 4.2.3.2 | Le choix des algorithmes d’optimisation

Parmi les maintes classifications des algorithmes, il est possible de les répartir selon leur mode opératoire,notamment que dans ce travail, leur façon d’opérer sera à la fois un avantage et une limite :

-Une première famille est celle des algorithmes déterministes qui impliquent toujours d’avoir le même résultat quel que soit le nombre de fois où nous réeffectuons la simulation.

-La seconde famille est celle des algorithmes stochastiques,dont les métaheuristiques bio-inspirés sont une sous classe très riche.Du fait de leur mode opératoire, ils proposent un résultat différent ou légèrement différent ,à chaque simulation effectuée.De ce fait, il n’existe

pas un résultat prédéterminé ni aucune garantie d'atteindre un optimum.



**Fig 4.15** – Classification des algorithmes selon leur mode opératoire pour la recherche de l'optimum : dans le cadre colorés, les algorithmes qui seront utilisés dans ce mémoire

Il serait certes plus judicieux d'utiliser des algorithmes déterministes pour avoir des résultats bien concis et définitifs, néanmoins, le choix d'utiliser des algorithmes bio-inspirés qui sont stochastiques, donc avec un résultat «hasardeux », est justifié par le fait que, principalement, ce type d'algorithme permet de se soustraire d'une défaillance connue relative aux algorithmes déterministes : Celle de l'optimum local.

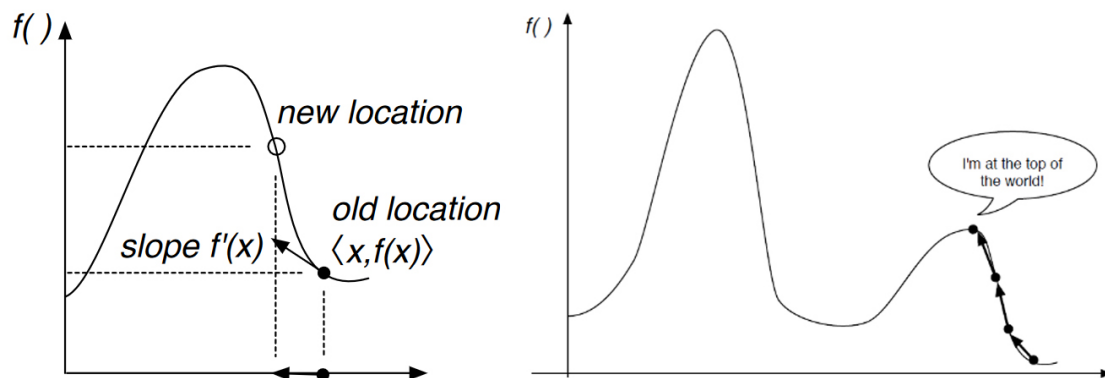
En effet, la recherche dans la plupart des algorithmes déterministes s'effectue sur la base du « voisinage » et cela implique de manquer ce que David Goldberg nommait «évènement principal» dans le paysage de solutions, c'est à dire le pic global du paysage.Par conséquent, la recherche est assujettie à une réduction dans l'exploration des solutions. Nous parlerons plutôt d'exploitation, bien plus que d'exploration, du paysage pour les algorithmes déterministes car pour trouver l'optimum, ils déploient un grand nombre de points de recherche et ce serait

alors des méthodes à privilégier dans des paysages de recherche dits « restreints ».[Goldberg, 1989 et Luke, 2013].

Nous noterons néanmoins, qu'il est possible également d'utiliser des méthodes hybrides d'optimisation, en combinant les avantages des algorithmes déterministes et des algorithmes stochastiques. Cependant dans l'outil paramétrique, il n'existe pas, actuellement, d'implémentation de telles méthodes.

### A - Les algorithmes déterministes : l'exploitation du paysage :

L'algorithme de la descente de Gradient fait appel à une approche mathématique traditionnelle pour trouver le maximum d'une fonction. En termes simples, il identifie à chaque fois, le prochain point, le plus haut, de la courbe pour y effectuer son « saut ». Cet algorithme souffre d'une défaillance relative à la fois à son temps de convergence prématuré vers la solution mais aussi au problème décrit ci-dessus qui est celui de l'optimum local[Luke,2018] et comme l'illustre les deux figures suivantes :



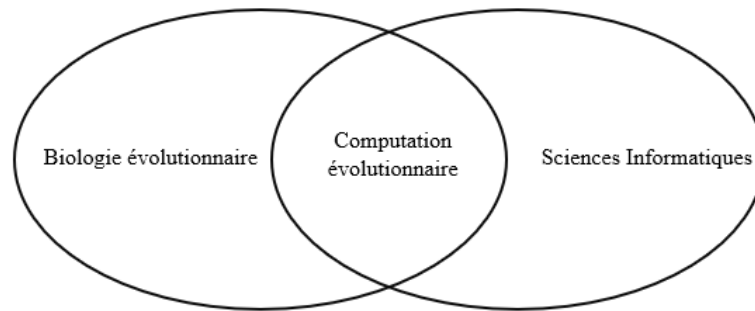
**Fig 4.16** – Mode opératoire de l'algorithme déterministe évoqué ci-dessus :A gauche : "les sauts" pour trouver l'optimum et à droite : le problème de l'optimum local

### B - Les algorithmes Stochastiques : Les métaheuristiques dans l'exploration du paysage :

Nous allons nous intéresser aux métaheuristiques qui relèvent de l'intelligence artificielle. Il s'agit d'algorithmes utilisés principalement lorsque la dérivée de la fonction objectif est inconnue et qu'il serait impossible de représenter l'optimum. [Luke,2018]

Il serait, dans ce cas possible d'essayer toutes les solutions éventuelles mais cela impliquerait une recherche de brute force et du fait du coût, en termes de temps et de ressources, que

cela impliquerait, les métaheuristiques, même si elles n'offrent pas de garantie d'optimalité ou s'adonne à une recherche «aveugle»<sup>4</sup>, sont pourtant dénotés comme la meilleure alternative aux algorithmes déterministes .Parmi cette classe, nous retrouvons les algorithmes bio-inspirés, sujet d'intérêt de ce mémoire.



**Fig 4.17** – Emergence des algorithmes bio-inspirés dans l'histoire : Transposition depuis la biologie dans les sciences informatiques

Un algorithme bio-inspiré ou évolutionnaire est un programme opérant selon des règles ou des comportements puisés dans les fondements de la biologie, telle que la sélection naturelle ou encore les modes d'auto-organisation dans les systèmes de vie. Émergent aux alentours de 1960 pour se développer jusqu'aux années 90, la computation évolutionnaire a donné lieu à ces algorithmes de recherche globale, dont nous citerons, principalement, les algorithmes génétiques (A.G) et les algorithmes d'optimisation par essaim particuliers (P.S.O).

**B – 1 Les algorithmes génétiques – (A . G) :** Tout d'abord, il convient de préciser la terminologie utilisée pour ces algorithmes spécifiquement et pour les algorithmes bio-inspiré d'une façon générale. L'emprunt du domaine du vivant, pour ces algorithmes, a conditionné une certaine nomination, que nous présenterons sommairement dans le tableau suivant :

4. Terme utilisé par Goldberg.D dans *enetic Algorithms in Search, Optimization, and Machine Learning*, 1989.Cela n'implique pas une recherche sans logique, au contraire, mais sous-entend que cette logique explore le paysage sans prédestination

Terminologie de référence	Abstraction/Transposition informatique
<b>Population / Ensembles d'individus</b>	Ensembles de solutions données par l'algorithme
<b>Génotype</b>	Structure numérique (résultat numérique)
<b>Phénotype</b>	Représentation de la solution
<b>Fitness fonction</b>	Fonction objectif (à minimiser ou maximiser)
<b>Fitness landscape</b>	Paysage de solutions à explorer

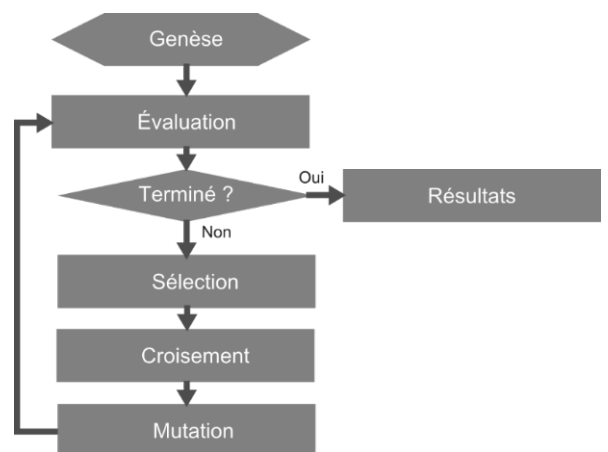
**Tableau 4.2** – Tableau de la transposition de terminologie entre la biologie et l'optimisation informatique réalisée par les A.G

Les algorithmes « génétiques », inventés par le professeur John Henry Holland en 1970, représentent la classe la plus importante de ces programmes. L'algorithme consiste en une suite d'opérations qui reposent sur le darwinisme et qui implique la survie du plus fort. En outre, l'optimum est atteint par un mode opératoire semblable à l'évolution biologique : Un ensemble d'individus(solutions potentielles) évolueront, à travers le paysage de solutions, jusqu'à la survie de l'optimum.

En effet, ces algorithmes commencent par la génération d'une population initiale, puis ils procèdent à l'évolution et la variation de ces solutions potentielles, dans un cycle itératif, à travers des opérateurs informatiques de « mutations » et de « croisement ».

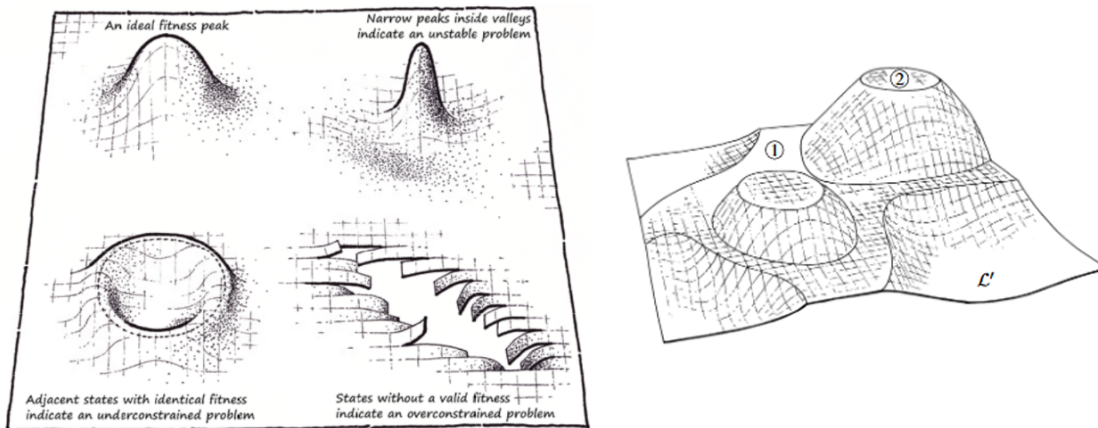
Cela implique que les solutions seront copiées puis interchangeées, entre elles, afin de créer des solutions nouvelles hybrides qui tireront à chaque fois le meilleur de la génération précédente. Nous pourrions d'ailleurs employer le terme de « parents » et « d'enfants »(offspring) pour représenter ce cycle de croisement et mutation dans les algorithmes génétiques[Goldberg, 1989].

Bien que ces algorithmes soient très sollicités dans les optimisations des problèmes architecturaux ou urbains [Chichoka, 2014 et Wormtann,2018], la garantie de l'optimalité est aussi sujette à la manière de formuler le problème. En effet, même s'il pourrait exis-



**Fig 4.18** – Cycle de l'optimisation par un AG

ter diverses façons d'exprimer les relations entre les paramètres, la fonction Fitness et les contraintes, il devrait y avoir un équilibre entre ces éléments. A cet effet, David Rutten démontre certaines typologies, notamment dans le cas de problèmes formulés avec beaucoup de contraintes ou de conditions ou au contraire avec une très grande liberté.



**Fig 4.19** – Les diverses topologies du paysage de solutions en fonction de la formulation du problèmes

### ***B-2 Les algorithmes de l'intelligence distribuée : L'algorithme d'optimisation par essaims particulaires (Particle swarm optimisation/P.S.O) :***

Rendu public, en 1989, par les ingénieurs Gerardo Beni et Jin Wang dans leur article dédié<sup>5</sup>, l'intelligence distribuée en informatique (swarm intelligence) repose sur la présence des « agents » qui sont les animaux et qui interagissent à la fois entre eux mais aussi avec leur environnement, qui dans le contexte de l'optimisation, serait le paysage de solutions à explorer. Cette interaction est régie par un certain nombre de règles qui implique une forme d'auto-organisation et qui diffère selon le système animalier dont l'algorithme s'inspire.

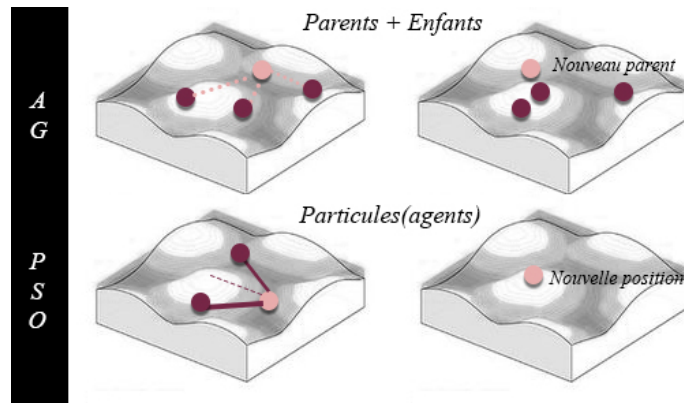


**Fig 4.20** – Essaim d'oiseaux : Un système biologique coordonné comme source d'inspiration du P.S.O

5. Beni et Wang dans *Swarm Intelligence in Cellular Robotic Systems*, 1989

Dans le cas de ce mémoire, nous allons uniquement aborder le système des essaims d'oiseaux, dont l'observation sur le comportement, a permis à l'ingénieur électrique R. Eberhart et au sociologue J.Kennedy de donner lieu aux algorithmes d'optimisation par essaims particuliers.

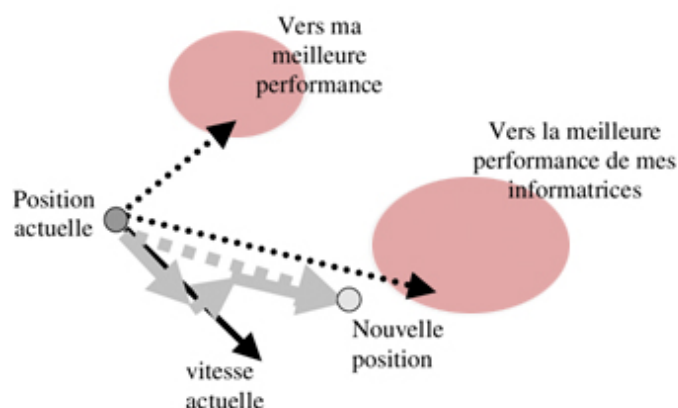
Bien que l'algorithme PSO est également dit « basé sur populations », comme l'algorithme génétique, ils ne procède pas à une computation itérative d'un ensemble de « parents-enfants », car les opérateurs de croisement et de mutation n'existent pas dans ce programme. Il s'agit, pour cet algorithme, plutôt d'une mise-à-jour itérative du mouvement de particules.



**Fig 4.21** – Comparaison entre les deux façons d'explorer le paysage des solutions par l'AG(dessus) et le PSO(bas)

Les opérateurs relatifs aux AG peuvent, dans ce cas, être assimilés aux trois règles qui régissent le comportement de la particule à savoir : l'alignement, la cohésion et la séparation, inspiré du mouvement des essaims d'oiseaux. Les créateurs du PSO se sont basés sur le modèle de simulation informatique d'un essaim qui a été développé par le physicien W.C.Reynolds[Kennedy,199 ], et ont intégré ce mode opératoire comme le processus de recherche de l'algorithme P.S.O.

L'exploration, quant à elle, est partitionnée en étape et est régie par la collaboration des diverses particules ou agents entre eux. Dans un paysage de solutions donné, chaque agent va mouvoir vers une nouvelle position jugée localement, pour lui, comme étant la meilleure et transmettre cette information aux autres agents, dans leur exploration. Suite à cela, les agents se déplaceraient alors selon une nouvelle position commune par la sommation des vecteurs de déplacement de chacun.



**Fig 4.22** – Mode opératoire du PSO : communication des agents entre eux pour déterminer la nouvelle position optimale

#### 4.2.3.3 | Choix de la méthode d'optimisation :

Du même que le choix de l'algorithme est important, la méthode l'est également et demeure en étroite relation aux algorithmes. Par méthode, nous mettons en lumière la différence entre l'optimisation mono-objectif et l'optimisation multi-objectif.

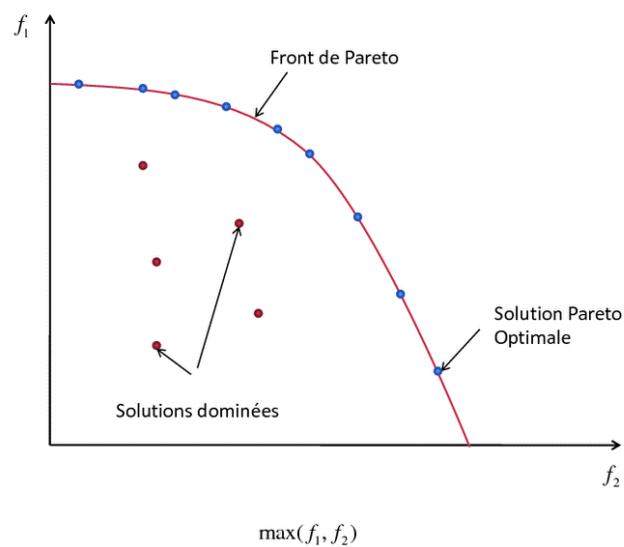
L'optimisation mono-objectif est facilement explicitée par le fait que l'algorithme n'aura qu'une seule fonction objectif ou fitness à traiter et par conséquent, la recherche de l'optimum et le processus d'optimisation en lui-même pourrait être relativement aisé. La meilleure solution au problème est clairement définie par le fait qu'elle est la seule à pouvoir remplir le coût optimal, en terme qualitatif ou quantitatif, et ce, à la fin de toutes les itérations.

Dans ce travail, l'optimisation mono-objectif est utilisée pour nous permettre de traiter des critères, un à un, d'une façon isolée et de se prononcer sur leur importance quant à leur influence sur le score de marchabilité. L'optimisation multi-objectif, quant à elle, demeure une voie d'exploration à privilégier en vue de la nature de certains problèmes complexes notamment pour le cas de la conception architecturale et urbaine qui impliquent de prendre en compte plusieurs critères et qui sont parfois, entre eux-mêmes, contradictoires. [Rittel et al, 1973].

**A - L'optimisation multi-objectif :** Dans son processus de réponse à deux ou plusieurs fonctions objectives, les algorithmes qui opèrent par le multi-objectif, ne donneront pas réellement « une » bonne solution, comme dans l'optimisation mono-objectif, mais aboutiront plutôt à une série de bonnes solutions.

Du fait que certains critères peuvent être conflictuels, comme évoqué précédemment, cette façon d’opérer est donc une conséquence logique de la nature complexe du problème à traiter et que tant qu’il y a possibilité de conflits, il y a nécessairement, pour l’algorithme, une exploration, de l’espace de décision ou recherche, dans le compromis.

Cette forme de « négociation » dans le processus de l’optimisation informatique est illustrée par le concept du front de Pareto. Concept relatif aux sciences économiques, son inventeur, Vilfredo Pareto illustre son approche par une frontière ou une limite fictive qui esquisse la « trajectoire » de plusieurs solutions optimales, désignées comme étant des « alternatives », en fonction de l’objectif qu’elles servent à remplir chacune. Ces solutions dominent tout le reste et il n’existe pas d’amélioration possible, dans le multi-objectif, pour un paramètre X, sans inévitablement détériorer le paramètre Y.

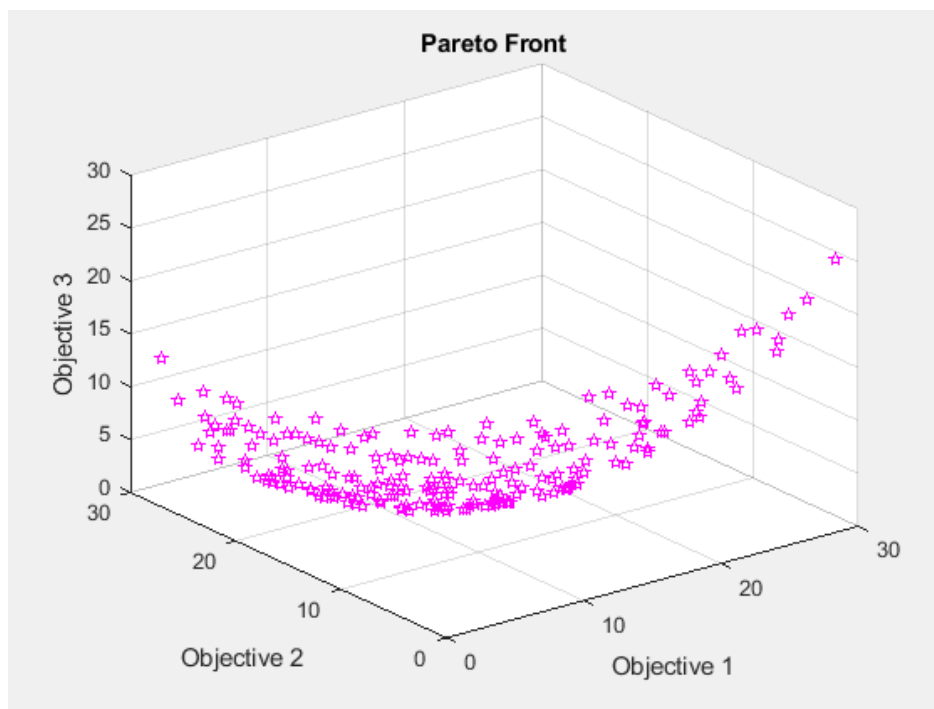


**Fig 4.23** – schéma d’une optimisation multi-objectif avec le cas de 2 objectifs à satisfaire : Ces deux objectifs sont matérialisés par les fonctions fitness  $f_1$  et  $f_2$  et les solutions optimales en front Pareto

Ci-dessus le schéma permettant d’expliciter ce principe du compromis. Pour deux objectifs donnés, représentés par les deux fonctions fitness  $f(1)$  et par  $f(2)$ , nous avons un ensemble de solutions « idéales » qui constituent le front Pareto mais dont aucune n’est réellement la meilleure, puisque une alternative qui tend à être optimale pour un objectif  $f(2)$  est considérée comme « non viable » pour l’objectif  $f(1)$

La figure ci-dessous sert à démontrer le passage à la troisième dimension qui s’opère par

l'introduction d'une troisième fonction objectif à maximiser ou minimiser. Dans les problèmes de conception architecturale ou urbaine, le multi-objectif peut par exemple se matérialiser par la nécessité de réduire le coût des éléments de construction, tout en recherchant, par exemple, la section maximale d'un élément permettant de recouvrir une façade. En soi, le conflit est concrétisé rien que par le fait de devoir maximiser un élément et minimiser un autre et ce, dans un seul processus.



**Fig 4.24** – Cas d'une optimisation multi-objectif avec 3 objectifs = limite de la représentation ?

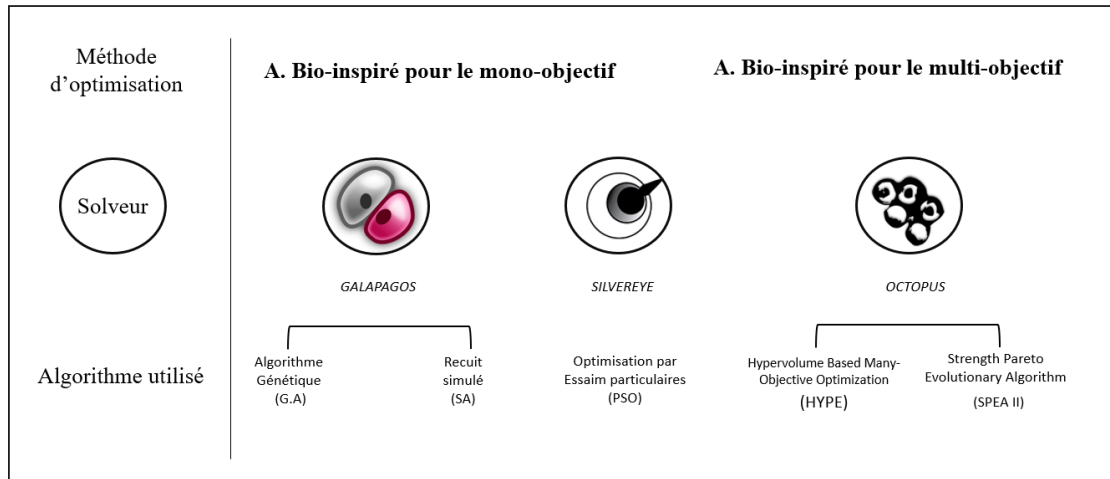
Les algorithmes évolutionnaires ont également des déclinaisons dans la méthode multi-objectif (Evolutionary Multi-objectif Optimization) qui opèrent selon le principe du front de Pareto et qui sont aussi associés à des principes biologiques. Nous citerons ici, le Strength Pareto Evolutionary Algorithm II (SPEA II) qui a été mis au point par le chercheur en S.I Eckart Zitzler dans l'ETH Zurich en 1998 et l'algorithme évolutionnaire HypE<sup>6</sup>, développé également par Zitzler, en compagnie du chercheur J.Bader dans la même école suisse.

#### 4.2.3.4 | Intégration des algorithmes d'optimisation bio-inspirés dans l'outil paramétrique :

Dans cette partie, nous allons aborder les algorithmes bio-inspirés, implémentés dans Grasshopper à travers une interface ou un programme permettant d'exécuter l'algorithme. Ce

6. Hypervolume-Based Many-Objective Optimization, algorithme multi-objectif qui sera utilisé dans ce mémoire.

programme est dit « solveur ». Ci-dessous un schéma illustrant les 3 solveurs qui seront utilisés dans les expériences avec les algorithmes qu'ils exécutent et que nous avons abordés précédemment.



**Fig 4.25** – Classification des algorithmes utilisés dans ce mémoire avec leur solveurs respectifs, selon la méthode : Mono ou Multi-objectif

Galapagos est un solveur qui a été intégré avec les outils natifs de Grasshopper par l'architecte et informaticien David Rutten. Ce dernier y a ajouté un second algorithme, autre que l'algorithme bio-inspiré dont nous parlons (l'algorithme génétique) : L'algorithme du recuit simulé qui est plutôt inspiré par un processus thermodynamique du refroidissement du métal et dont nous ne ferons pas usage.

Dans le Canvas de Grasshopper, Galapagos se présente sous la forme d'un composant « rose » à deux entrées : L'une pour les génomes soit les paramètres à modifier et la seconde pour la Fitness, soit pour la fonction objectif ou la valeur à minimiser ou maximiser.

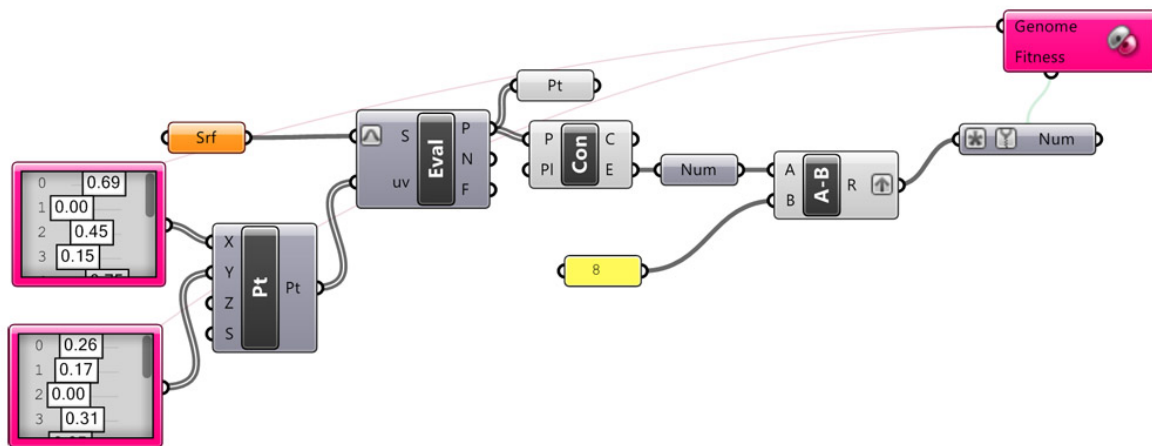


Fig 4.26 – Définition avec le solveur Galapagos à la fin et la notion des gene pools

- La première entrée prendrait donc des composants de type **sliders** : Des « curseurs » numériques qui permettent de se déplacer sur un intervalle spécifié par l'utilisateur : Exemple : des sliders d'angle se déplaceraient de 0 à 360 degrés.

Nous noterons qu'un **slider** = valeur mais qu'il est possible d'avoir un **slider** comprenant plusieurs valeurs, portant le nom de **gene pools**. L'exemple ci-dessus illustre deux gene pools correspondant à des valeurs de X et de Y pour positionner un ensemble de points, en même temps et ce, sans avoir à créer plusieurs **sliders** séparés pour chaque point.

- La seconde entrée prendrait un composant, appelé **number** et qui serait lui-même lié au résultat d'une Fonction Fitness (FF).

Galapagos, mettant en application l'Algorithme génétique, essaiera donc toutes les combinaisons des paramètres, en mouvant les sliders jusqu'à ce que le résultat de la Fitness Function soit minimal ou maximal et ce, selon la formulation du problème.

Dans la page ci-contre, deux figures, associées, présentant l'interface de Galapagos :

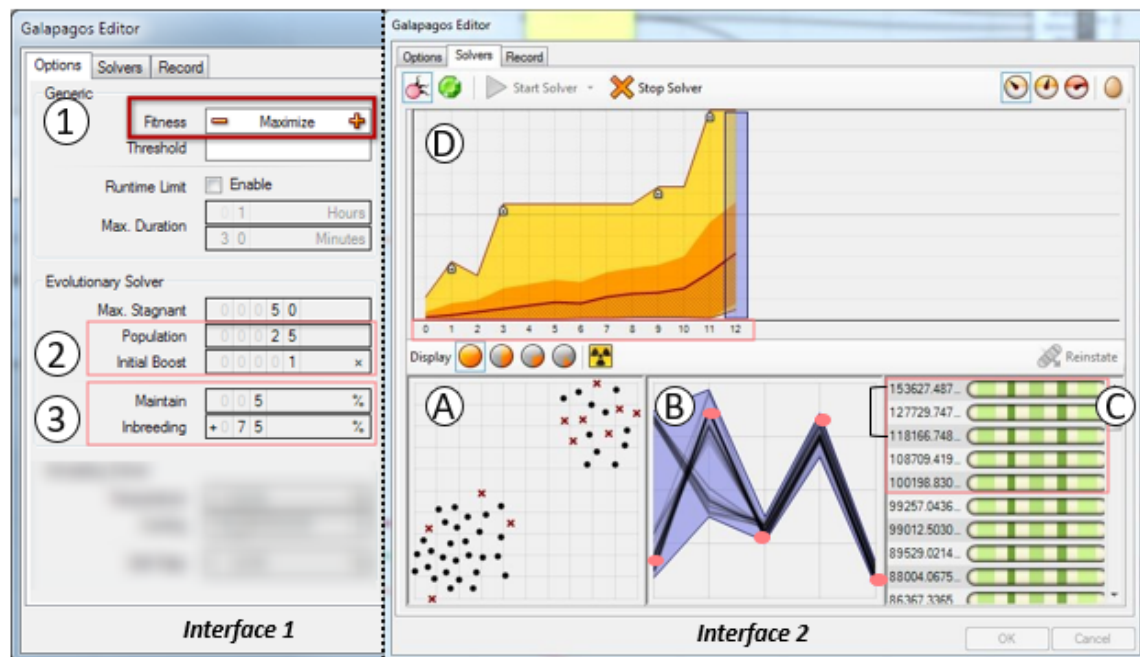


Fig 4.27 – Exemple d’une définition avec le solveur Galapagos à la fin et la notion des gene pools

La figure à gauche présente l’interface de départ comme suit :

- 1- Choisir de minimiser ou de maximiser la fonction
- 2- Définir la taille de la population et de la population initiale (Initial Boost x Population)
- 3- Pourcentages des opérateurs « sélection », « croisement » et « mutation » que nous avons abordés dans B-1

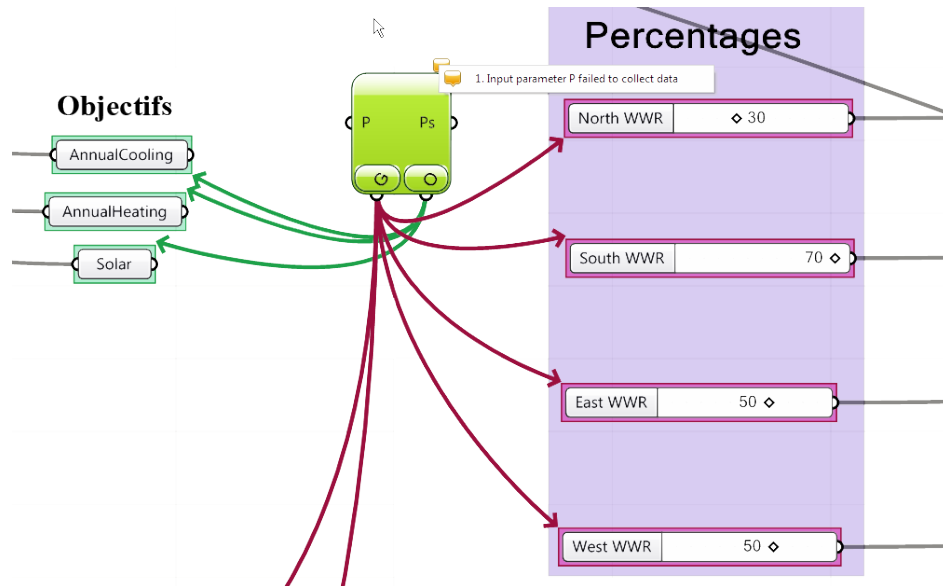
A – les génomes dans le paysage de solutions (vue en 2D) avec les croix représentant les génomes qui n’ont pas évolué.

B- La courbe représente la combinaison des paramètres : Pour chaque pic, nous avons une valeur correspondante dans le *slider* pour un paramètre donné. Par exemple, ici, nous pouvons dire que nous avons 5 paramètres donc 5 *sliders* (Une combinaison de 5 valeurs à un instant t).

C- La Fitness : représente les valeurs trouvées et correspondent au nombre indiqué de la population. Par exemple, nous pouvons supposer que l’algorithme va maintenir les 5 premières solutions, puis les croiser et les hybrider et ce, selon les pourcentages indiqués dans l’interface 1 à gauche.

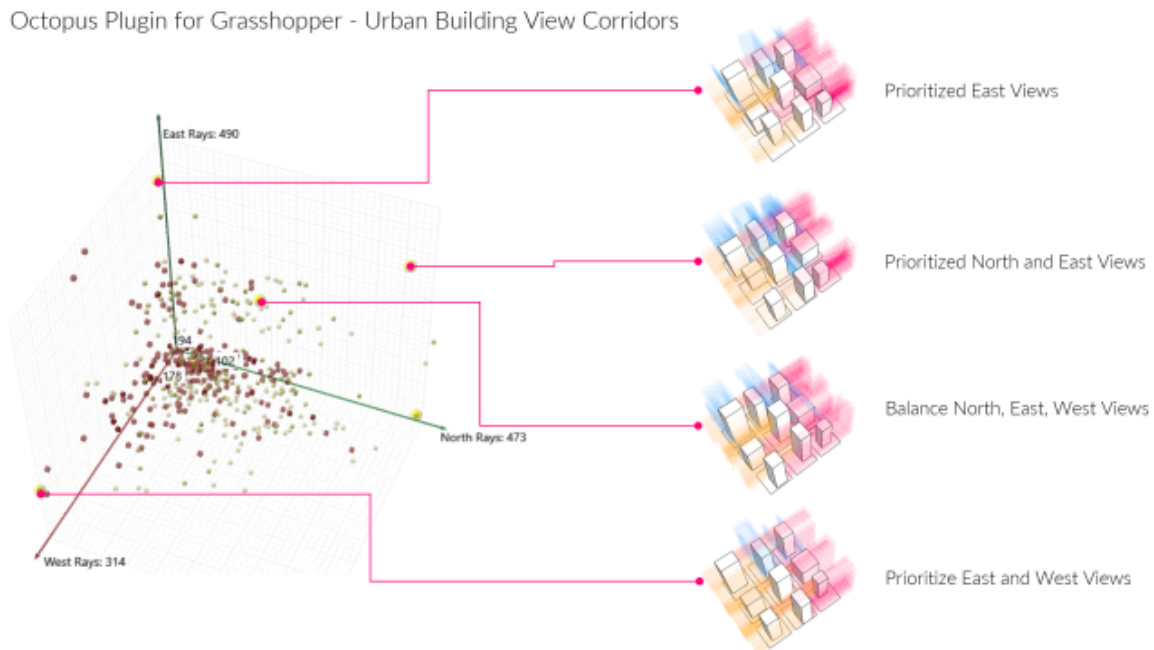
D- L’évolution convergente des solutions pendant le nombre d’itérations (dans le cadre).

L'interface de Silvereye et Octopus reprend les mêmes concepts, mise à part la représentation de la recherche du multi-objectif dans Octopus, qui est illustrée avec les deux figures suivantes :



**Fig 4.28** – Le composant du solveur octopus dans le canvas de grasshopper avec plusieurs objectifs(en vert) et les paramètres variables(en rose)

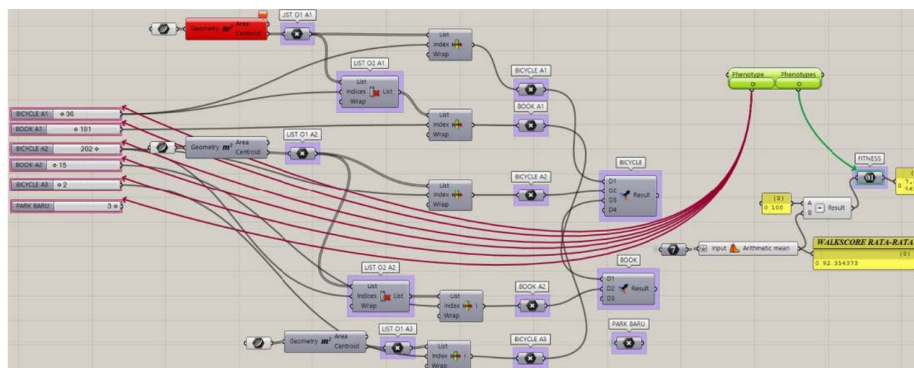
Octopus Plugin for Grasshopper - Urban Building View Corridors



**Fig 4.29** – Interface d'Octopus : Représentation de l'espace de recherche pour le cas de 3 objectifs

#### 4.2.3.5 | Optimisation de la marchabilité avec les algorithmes bio-inspirés

Dans leurs travaux du CAADRIA <sup>7</sup>, en 2019, Le professeur, en design computationnel, Aswin Indraprastha réintroduit la fonction objectif de Walkscore mais établit un autre type de relation. ils soustrait 100 du score qu'il calcule et qui est modifiable, et obtient donc un écart type qu'il essaie de minimiser par l'usage de l'algorithme d'Octopus mais mis à usage uniquement pour le seul objectif décrit ci-dessus. Ainsi, l'algorithme recherche la position des équipements à ajouter de façon à ce que le score soit maximal.



**Fig 4.30** – Algorithme sur grasshopper du travail pré-cité : recherche de la position des équipements

Par l'optimisation de l'emplacement d'un ensemble d'équipements, le score de marchabilité augmente alors de 40% pour les deux zones. Cependant étant donné qu'il reprend le PFM pour le score final, donc une analyse strictement en 2D et sans prise en compte des voies d'accès à l'entrée ou de la topographie, il met en lumière cette limite, dans son algorithme, et des possibles scores erronés que cela aurait pu induire.

Un professeur au MIT, Christoph Reinhart propose, dans le contexte du SimBuild 2012 <sup>8</sup>, des études qui introduisent la topographie. Il tente, à travers un terrain fictif qu'il modélise, d'optimiser une génération de forme urbaine selon la topographie. Sur cette base, il calcule le score de marchabilité, après avoir affecté les CI aux équipements. Néanmoins, il n'y a pas de comparaison entre la limite et ce qu'il propose. Son travail était surtout de construire une démarche paramétrique de la forme urbaine, qui s'adapterait à la dimension physique du terrain.

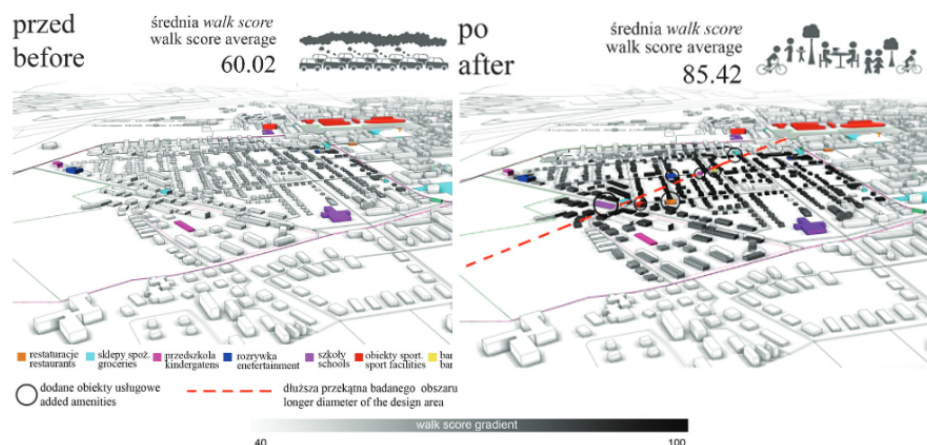
7. Intelligent and Informed, Proceedings of the 24th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2019, Volume 2, 161-170

8. Conférence Building Performance Analysis



**Fig 4.31** – Introduction de la topographie comme un paramètre dans la démarche paramétrique urbaine : usage uniquement pour la morphologie urbaine

D'autres travaux similaires dont ceux de l'ingénieur Judyta Chichoka, créatrice du solveur Silveye(solveur de l'algorithme P.S.O) présente la même démarche qui est "la minimisation de la distance sur une analyse 2D", en exposant la méthode Walkscore comme référence des critères pour le calcul final. A cette fin, seule la minimisation de la distance vis-à-vis de certains points de la ville est prise en compte, sans autre critère autre relatif au contexte.



**Fig 4.32** – Résultat du travail de J.Chichoka en minimisant la distance des équipements vis-à-vis de certains points seulement avec la méthode walkscore

Les travaux représentent, globalement, la même démarche sur l'outil paramétrique et bien que celui-ci soit apte à l'introduction des avantages de plusieurs méthodes externes, comme l'approche syntaxique et l'approche GIS, les travaux sur la marchabilité, encore récentes sur l'outil paramétrique, n'en tiennent pas compte pour faire évoluer le résultat.

# Partie expérimentale





## Expériences et Analyse

### 5.1 | Méthodologie du protocole expérimental

Avant de lancer les simulations, cette partie vise à expliciter comment est mis en place le processus expérimental. Ci-dessous un schéma des trois parties de l'explication que nous allons aborder :

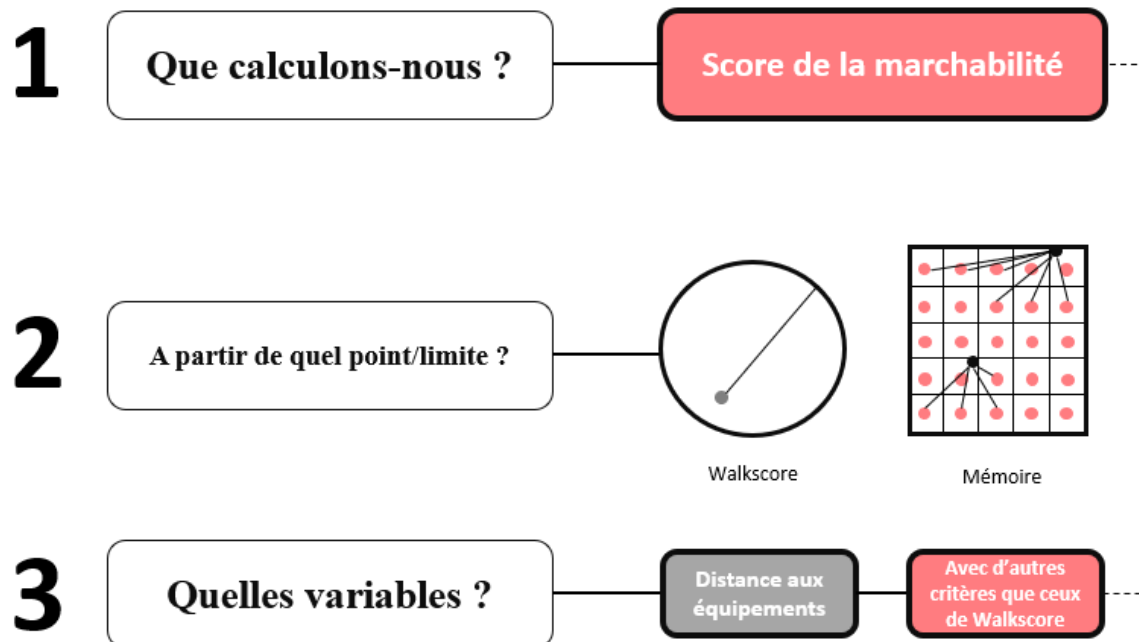


Fig 5.1 – schéma de l'explication de la partie à aborder

### 5.1.1 | Que calculons-nous ? Le score de marchabilité

Nous allons calculer le score de marchabilité à travers le module de calcul d'Urbano (composant *walkscore* ou *StreetHits*), qui reprend, de manière générale, les règles de l'algorithme WalkScore pour les introduire dans l'outil paramétrique.

Etant donné que nous cherchons à montrer l'influence de d'autres critères sur ce score, nous procéderons par comparaison. La base (condition initiale) sera de reprendre le calcul par la méthode actuelle ou les éléments avant optimisation/critères et les éléments de comparaison seront nos expériences.

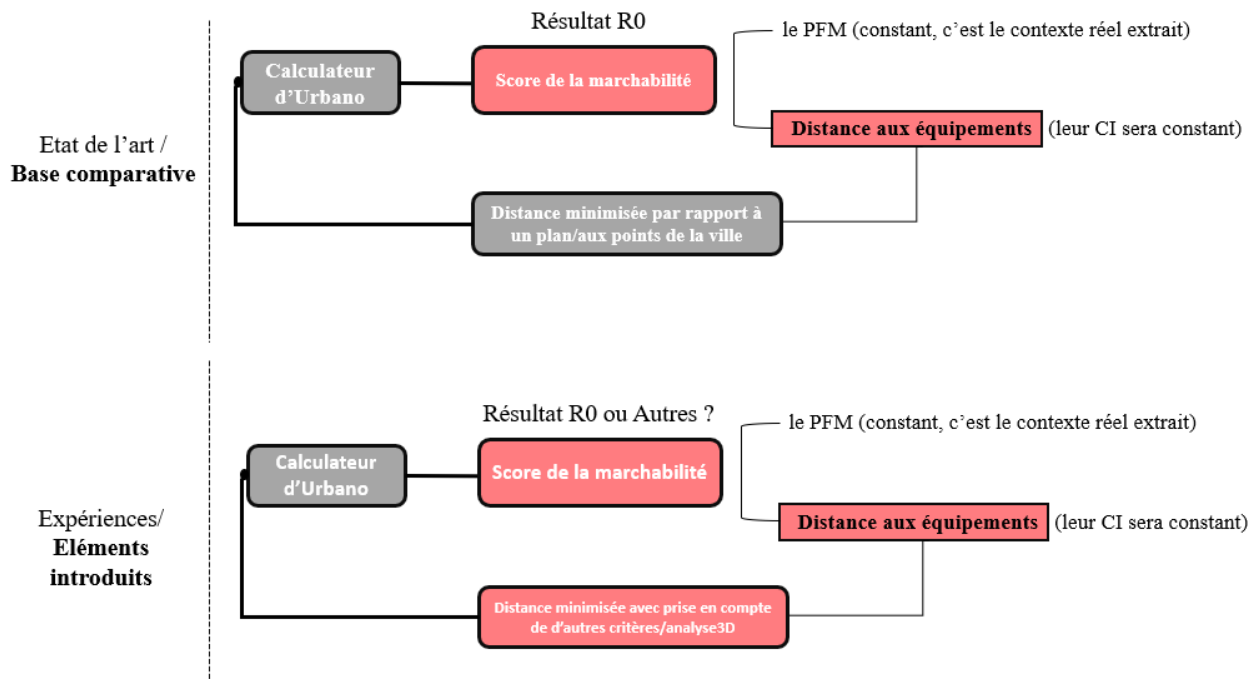


Fig 5.2 – Schéma de la base comparative : méthode actuelle et expériences

La méthode actuelle consiste à minimiser la distance des habitants par rapport à leur équipements et construit donc une relation prévisible et automatique telle que « Distance minimale = score maximal ». Elle prend en compte uniquement la méthode « Walkscore ».

Notre méthode consistera à apporter d'autres critères dont l'analyse avec une prise en compte de la topographie afin de démonter, à la fois, l'influence de ces critères sur le score mais aussi que la relation évoquée ci-dessus n'est pas toujours vraie (Par exemple, la distance pourrait augmenter et le score également). Cela sous-entend que des dispositions ou des configurations, trouvées par l'algorithme, pourraient paraître « contre intuitives » ou désavantageuses mais offriraient un meilleur optimum pour la marchabilité.

Pour cela, nous allons proposer l'ajout de nouveaux équipements, sur un contexte réel extrait, mais leur coefficients d'importance sera gardé constant voir négligeable (tous les équipements auront la même valeur d'une épicerie soit un  $ADP = 3$ ). Ce choix est dû au fait que nous cherchons à optimiser la distance par rapport à ces équipements, et que le type d'équipements, en lui-même, ne présente aucun apport à cette étude. En soi, le coefficient d'importance est une valeur extrêmement subjective en fonction du contexte mais l'affectation d'une valeur et d'une sémantisation du point est nécessaire pour la génération du modèle Urbano. (voir [4.2.2])

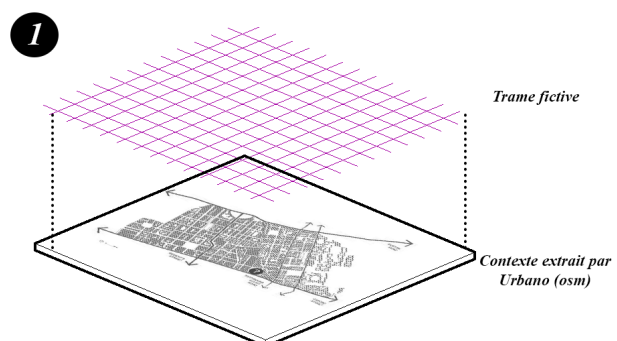
### 5.1.2 | Origine et limite physiques du calcul

Tandis que Walkscore procède à partir d'un seul point d'origine (l'adresse entrée par l'utilisateur) et calcule des isochrones, nous proposons de réinterpréter cette méthode en multipliant les points d'origines. **Walkscore procède par point car il informe l'utilisateur sur une donnée existante déjà. Nous procédons par de "multiples points" car nous sommes dans le cas de l'optimisation de tout un contexte et non d'un renseignement.**

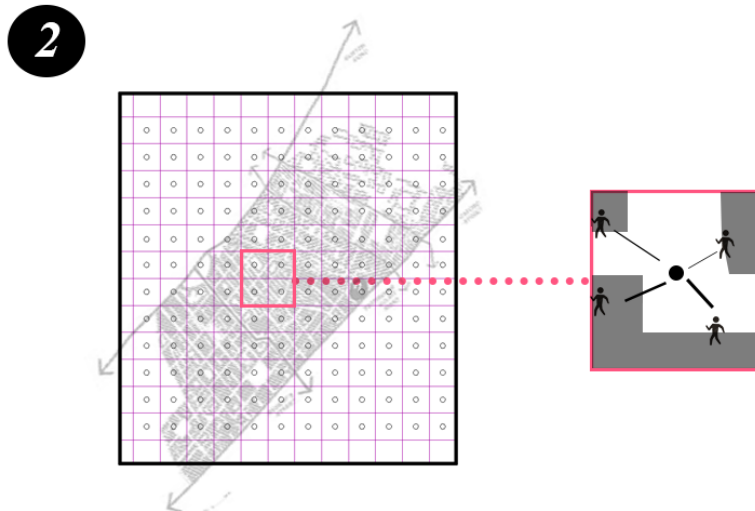
Nous détaillons, d'abord, notre méthode par des schémas puis sa correspondance algorithmique dans Grasshopper. Les étapes sont comme suit :

1 - La création d'une trame ou grille à projeter sur l'espace de la ville. Cette trame, une fois projetée, servira à englober **chaque morceau de la ville dans un carreau.**

2 - Dans chaque « morceau » de la ville, les gens peuvent se trouver dans des endroits différents donc afin de créer **une approximation** de leur position, nous allons chercher le **centre de chaque carreau**

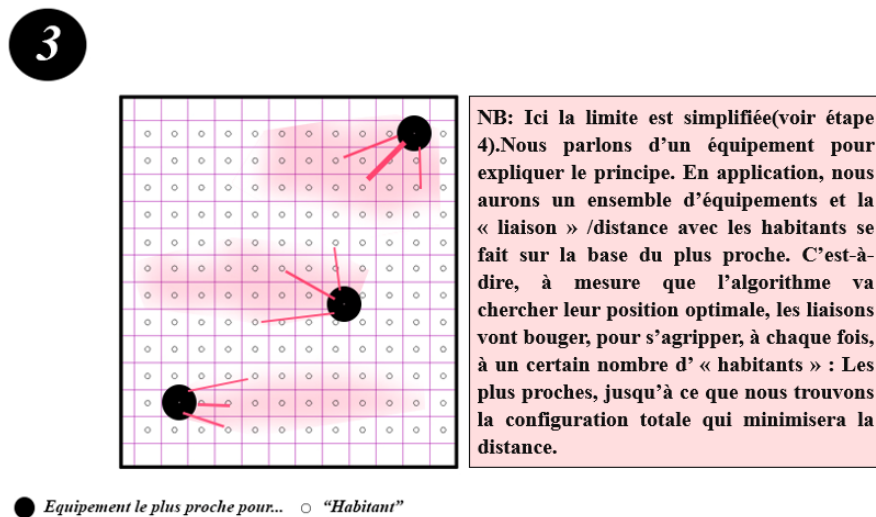


**Fig 5.3** – Projection de la trame fictive sur le contexte extrait



**Fig 5.4** – Moyenner la position de chaque habitant en trouvant le milieu de chaque carreau

3 - L'équipement (ou les équipements) est lui aussi représenté par des points. Nous traçons alors une ligne depuis tous les points de la ville vers cet équipement et nous obtenons alors plusieurs distances (égales aux nombres de points). La moyenne de ces distance est ce que l'algorithme va chercher à minimiser, en altérant dans sa recherche, la position de l'équipement jusqu'à il trouve son emplacement adéquat de façon à ce que la distance moyenne soit minimale.

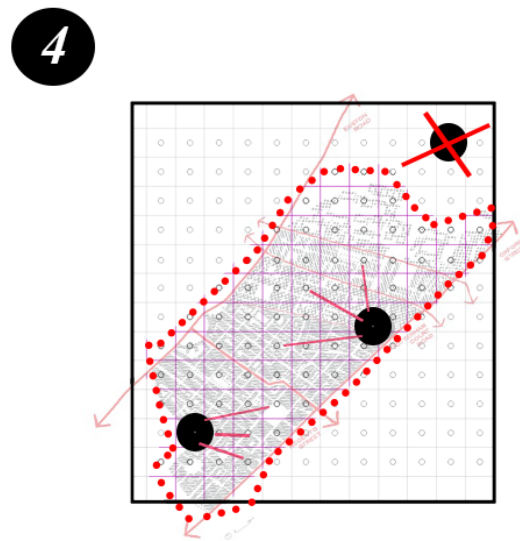


**Fig 5.5** – Disposer aléatoirement des équipements et lier chacun avec un certain nombre d'habitants les plus proches

Il est important de souligner, que jusqu'à là, nous ne nous prenons pas encore en compte le réseau viaire. Les liaisons tracées avec les équipements sont "diagonales" et servent uniquement à avoir une première distance. L'intégration du réseau viaire ainsi que de l'entrée sera détaillé dans la prochaine section.

4-Enfin, une dernière étape est la délimitation de l'aire de recherche. En effet, lorsque nous passons au cas de plusieurs équipements, il arrive que, dans sa recherche de plusieurs emplacements optimaux, l'un des équipements se retrouve assez loin et n'a plus d'importance. Pour cela, nous posons dans Grasshopper une forme de condition à l'algorithme qui est la suivante :

5-Puisque nous commençons à prendre le réseau de la ville en compte, cela implique de passer des liaisons diagonales aux chemins les plus courts, en empruntant la voie. Pour cela, nous ferons appel à l'algorithme A\*(A search). Nous allons également introduire le système de classification de OSM pour la détection de l'entrée. Par cela, nous entendons affiner encore plus la position approximative (le centre de chaque carreau) que nous avons établi dans l'étape 2.



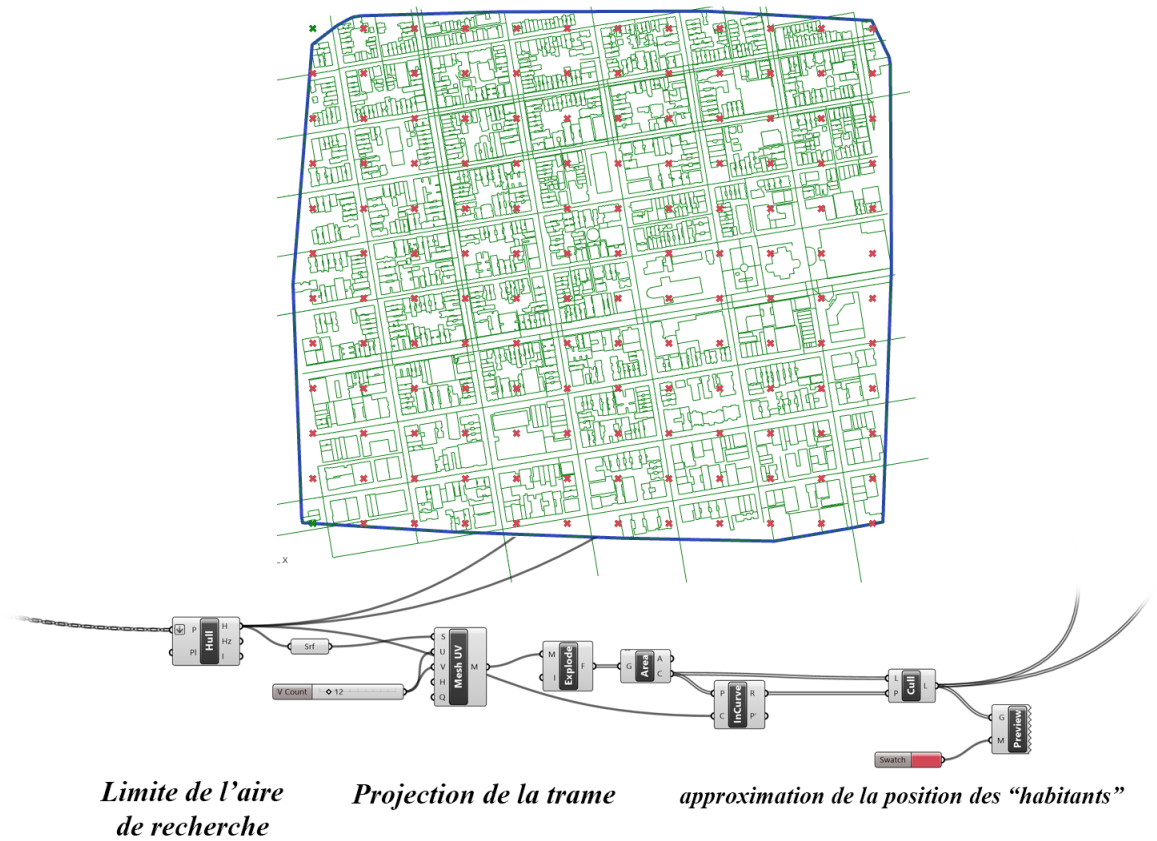
**Fig 5.6** – Délimitation de l'aire de recherche et condition posée : omission de tout point à l'extérieur de cette aire

#### Condition posée à l'algorithme

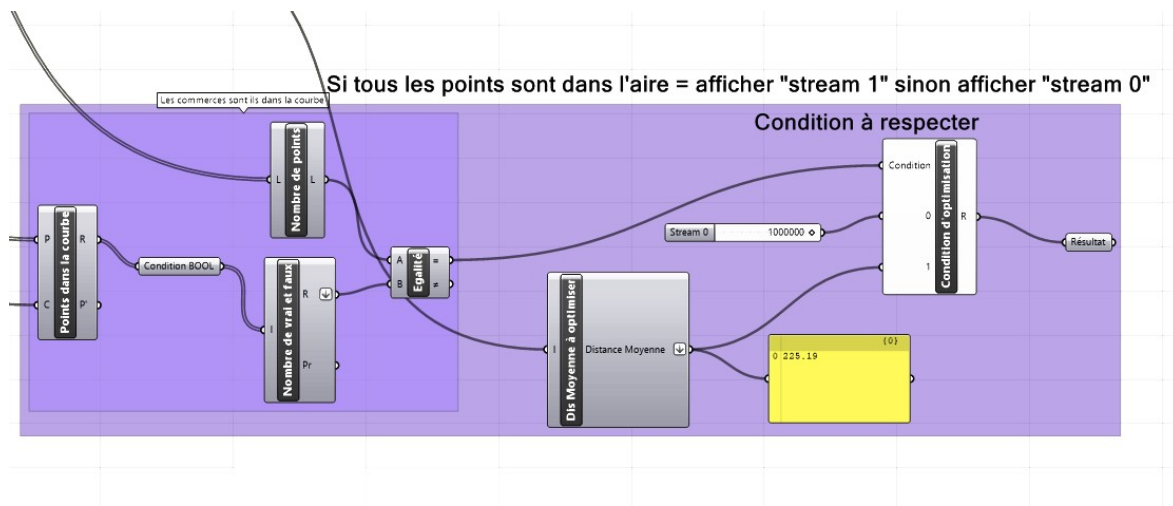
**Si** tous les équipements sont dans l'aire de la recherche => **Alors** afficher le résultat de l'optimisation (c'est la moyenne des longueurs minimisée)

**Sinon**

**Afficher une erreur** : par exemple une valeur 10000000. Et **continuer de chercher** jusqu'à ce que tous les points soient dans l'aire **et** que les itérations requises soient finies.

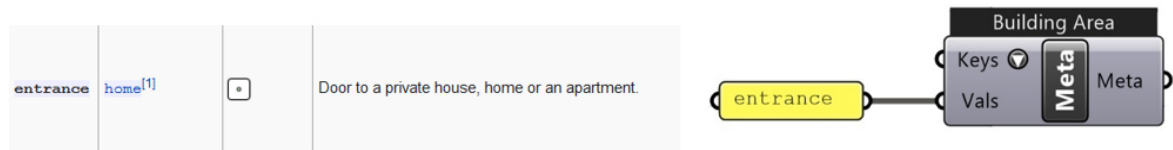


**Fig 5.7** – Délimitation de l'aire de recherche finale et pose de la condition : un point hors de l'aire de recherche n'est pas compté

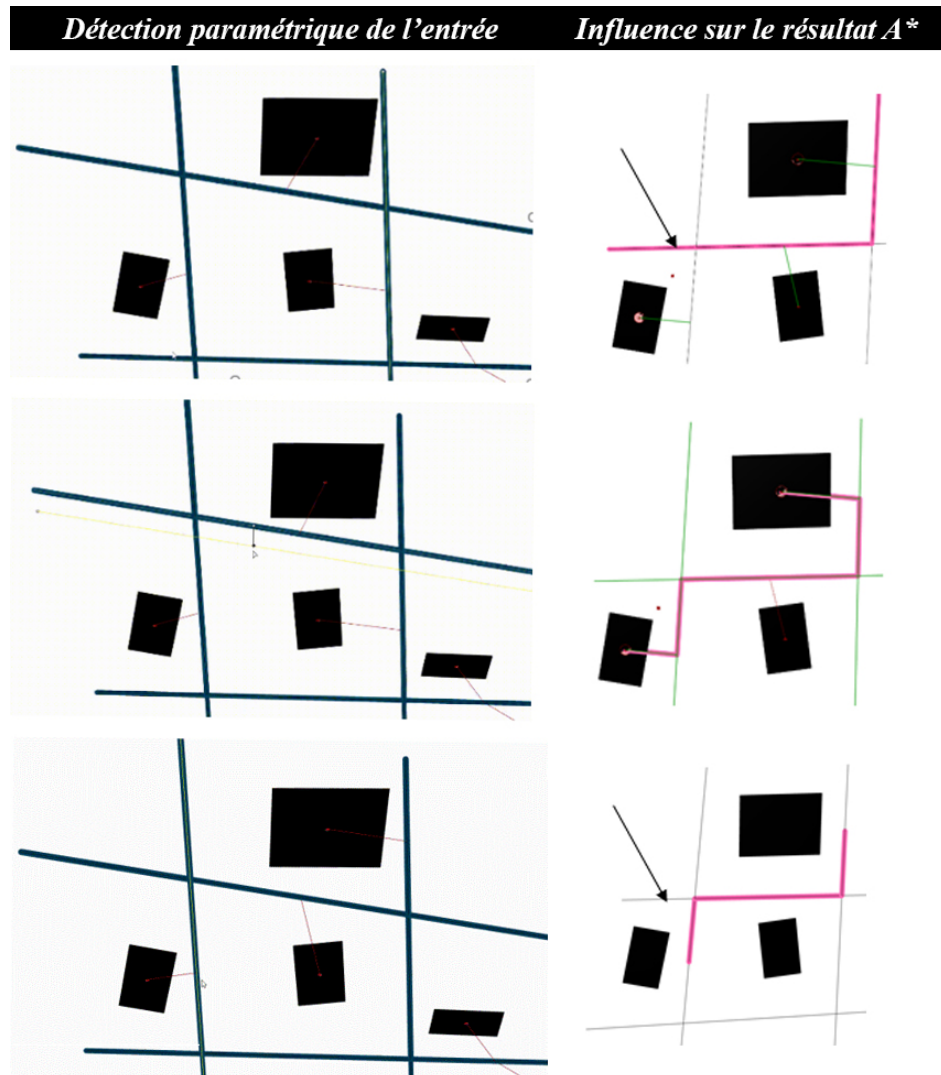


**Fig 5.8** – Dans Grasshopper, la condition est posée par le composant *pick'n'choose* qui permet de choisir d'afficher un résultat ou une erreur





**Fig 5.10** – Détection paramétrique de l'entrée(sans recours au tag) par le composant *curve closest point* dans Grasshopper et l'influence de l'entrée sur le changement du parcours marchable



**Fig 5.11** – Détection paramétrique de l'entrée sur Grasshopper

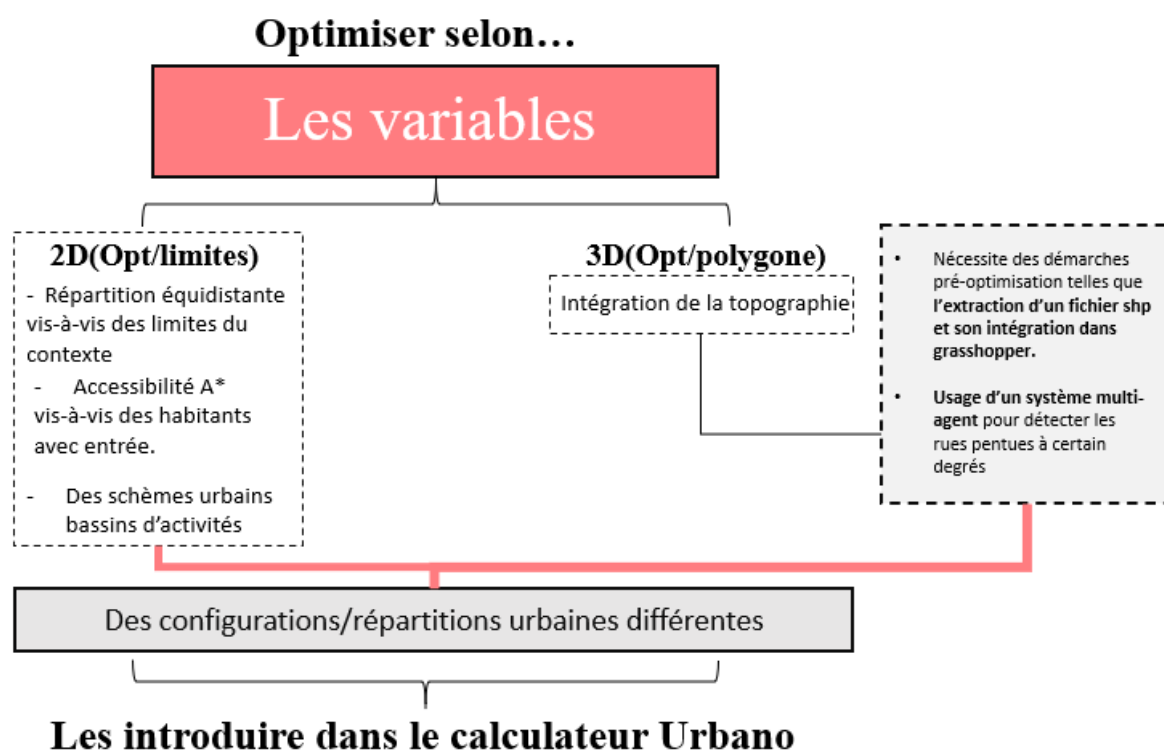
Si cette donnée n'est pas disponible, la démonstration ci-dessus, construite sur Grasshopper permet de récupérer une approximation de l'entrée, par détection automatique. Il s'agit de détecter les rues les plus proches(en divisant la courbe par des points avec le composant *divide curve*) par rapport à un bâtiment(en obtenant son centre) et de tracer une ligne

supposée être l'entrée par rapport à la proxémie paramétrique du réseau viaire. Néanmoins, cette méthode illustrée, dans ce mémoire, reste un moyen pour expliquer l'importance de l'entrée uniquement et n'est pas toujours valable dans la conception réelle.

La démonstration dans la page précédente prouve que, dans les travaux actuels, l'omission de l'entrée (pour les cas des bâtiments avec une classe "résidentiels") fait que l'algorithme choisira le chemin le plus court mais que celui-ci n'est pas, nécessairement, celui qui sera sollicité dans le parcours. Dans tous les cas, l'intégration de ce tag, se fait d'une manière implicite pour les expériences d'optimisation, afin d'affiner le résultat.

### 5.1.3 | Définition des variables à introduire

Nous pouvons répartir les variables à étudier en 2 classes et ce, selon leur unité d'analyse (2D segmentale et ponctuelle) ou (3D-polygonale).



En présentant, dans la section 5.1.2, notre méthode générale de l'interprétation de la ville, nous avons illustré, globalement, comment se ferait les variables 2D, plus tard : répartition équidistante et accessibilité A\* avec l'introduction de l'entrée. Donc, nous allons, dans cette

partie, expliciter la variable « bassins d'activités » soit une minimisation des distances des équipements vis-à-vis d'eux même.

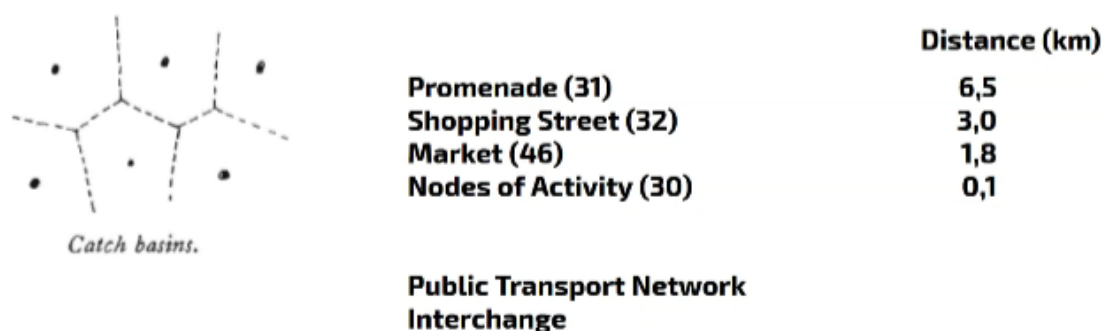
Puis, nous illustrerons la démarche pour intégrer la variable topographie dans Grasshopper pour l'optimisation selon ce critère. L'intégration de la troisième dimension, dans la simulation de la marchabilité, est actuellement omise, d'où la proposition, dans ce mémoire, d'une première approche qui fait appel à d'autres étapes extérieures à l'expérience d'optimisation elle-même.

### 5.1.3.1 | Les « bassins d'activités » d'Alexander : Minimisation de la distance dans le réseau des équipements selon une distance reference

:

Dans l'ouvrage référence « A language Pattern », C. Alexander énonce certains systèmes et « distances idéales » ou « références » qui contribuerait à favoriser la marche dans un lieu donné. Il propose la schématisation des bassins d'activités, qui pourraient par exemple être

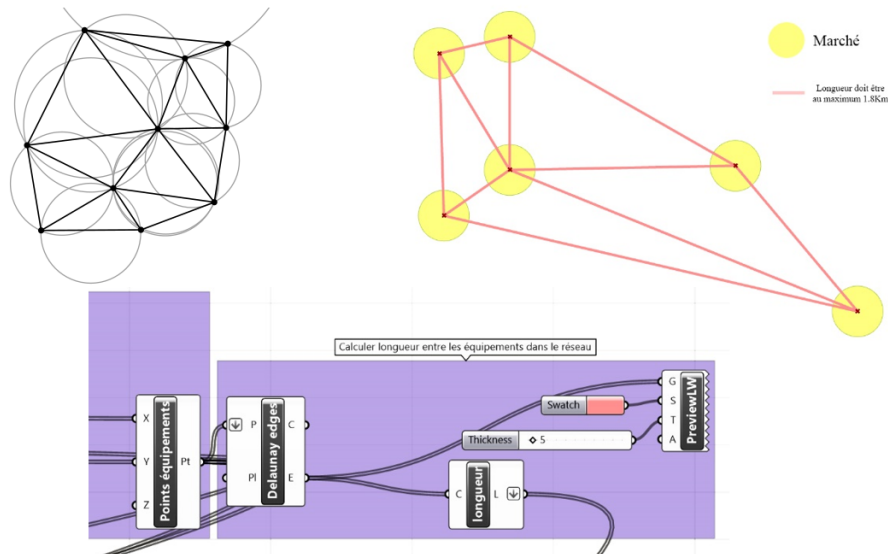
dans le caché de marchés, égale à 1. 8Km. En outre, chaque marché doit être situé à une distance maximale de 1.8Km de son voisin dans le réseau.



**Fig 5.12** – Extrait du livre de A language Pattern de C. Alexander montrant les schémas urbains des bassins d'activités avec une distance "idéale" dans le réseau de 1.8Km

Le choix de minimiser la distance en se basant sur cette configuration est justifiée par le fait qu'elle se distingue de la « base comparative », du fait que l'équipement sont placés par rapport à eux-mêmes dans le réseau et non par rapport aux habitants. Sur l'outil Grasshopper, nous proposons, par exemple, la représentation de cette configuration, établie par Alexander, à travers la figure du « triangle de Delaunay »<sup>1</sup>.

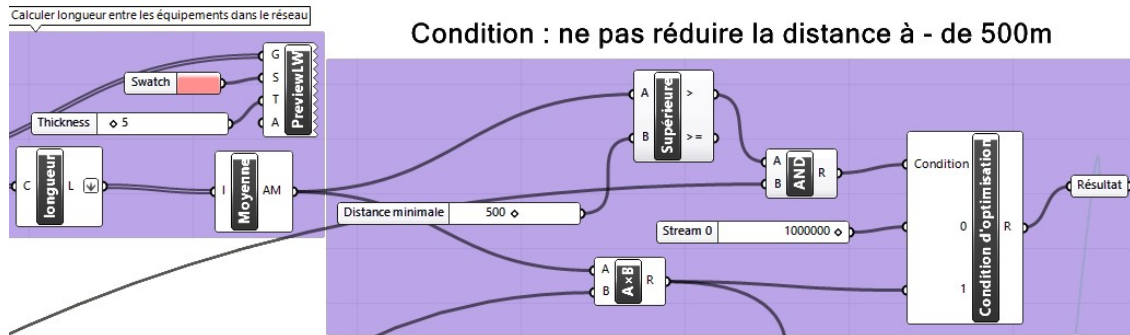
1. Inventée par Boris Delauney, il s'agit d'une méthode de triangulation qui permet de relier un set de vertex, tel qu'aucun de ces vertex ne se trouve dans le cercle circonscrit des triangles créés.



**Fig 5.13** – Interprétation du schéma d’Alexander par la figure de Delaunay dans l’outil paramétrique

La distance moyenne entre les points, dans le réseau, serait propice à la marche si elle était au maximum 1.8Km. Cependant, si nous introduisons ce but « réduire la distance entre les points à moins de 1.8Km » comme fonction objectif dans l’algorithme, celui-ci pourrait aboutir à une répartition superposée de tous les points, c’est-à-dire à réduire complètement cette distance à 0, ce qui n’est pas logique.

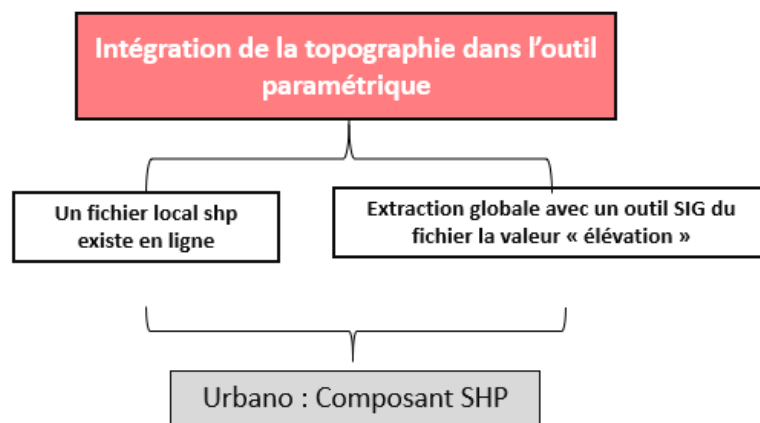
A cet effet, nous reformulons une condition, avec une méthode semblable à celle de la délimitation de l’aire de recherche : Introduire comme distance minimale une valeur de 500m, correspondant à 5minutes de marche à raison de 5-6Km/h, et ce, afin de garder justement un certain cheminement entre les équipements. Pour cela, nous utilisons le même composant et méthode que dans la figure 5.8 dans la délimitation de l’aire de recherche, qui est d’imposer à l’algorithme de ne pas prendre en compte des valeurs en dessous de cette limite autorisée.



**Fig 5.14** – Partie de la définition montrant la condition posée avec le critère des bassins d'activités d'Alexander

### 5.1.3.2 | Passage vers des simulations 3D : Intégration de la topographie dans la marchabilité :

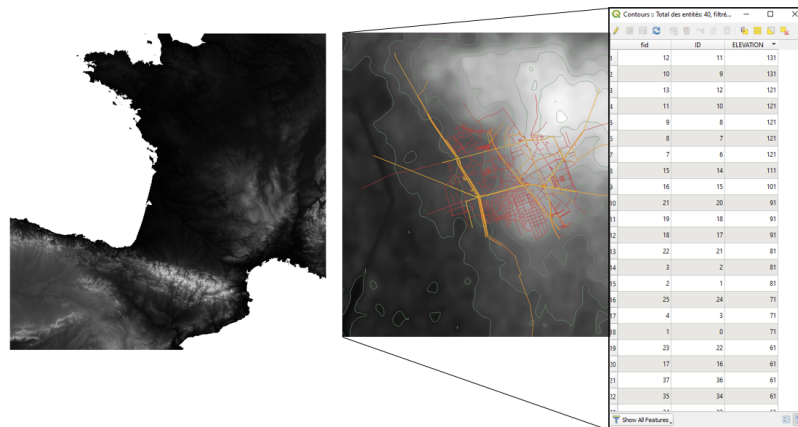
Pour intégrer la topographie, nous aurons recours à un fichier shp et selon les disponibilités des données, ce sera comme suit :



Lorsque celui-ci n'existe pas, le fichier shp peut-être récupéré à travers un logiciel SIG/GIS tel que Qgis(approche GIS/SIG). Pour cela, nous avons recours à un fichier global « DEM » ou un fichier TIF, tel que l'a cité Carlo Ratti, dans sa critique à la space syntax, que nous avons évoqué à la section [2.2.2.4] : De ce fichier global, nous extrayons la zone d'étude

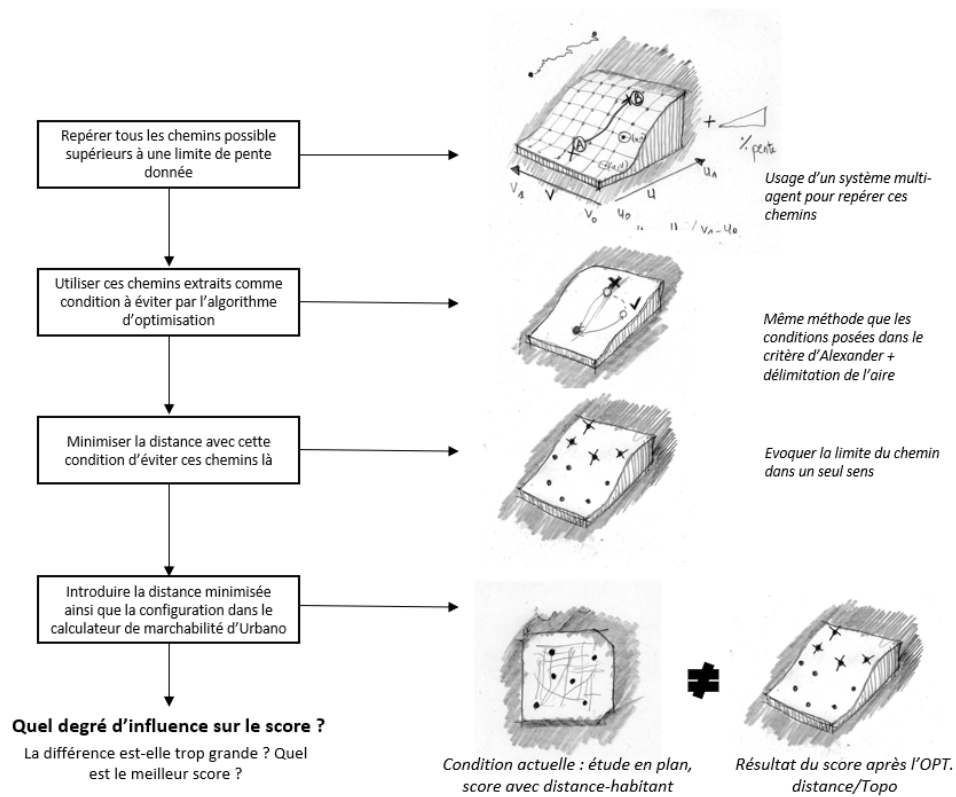
choisie, par une délimitation vectorielle puis nous importons sous format « shp » avec les valeurs d'élévation du DEM local et nous intégrons ceci dans l'outil paramétrique à travers le composant import *shp*. L'intérêt d'une telle approche est que la topographie est directement

intégrée dans la construction du modèle de calcul Urbano et non pas seulement, une autre entité géométrique.



**Fig 5.15** – Exemple Extraction et importation des valeurs d'élévation d'un DEM de la zone voisine à la rue Ménilmontant à Paris

Une fois, la topographie intégrée dans le module voici la méthode suivie dans ce mémoire



Dans cette partie, nous allons uniquement aborder la première étape qui est l'usage des systèmes multi-agent<sup>2</sup> pour repérer certains chemins, selon une pente limite, que nous choisirons. Les autres étapes relèveront de la partie « optimisations et résultats » qui suit cette section [Mise au point du processus expérimental] d'explication du processus expérimental.

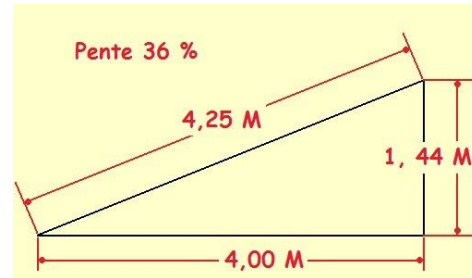


Fig 5.16 – règle de la pente

Le système multi-agent (SMA) qui traverserait les chemins possibles d'accès, est conditionné à une traversée basée sur la règle géométrique du calcul de pente formulée ainsi :

Pente (%) : (Hauteur / Distance horizontale) x 100

Ci-dessous, un exemple de la partie de cette règle introduite pour le parcours de l'agent. Nous avons d'abord appliqué ceci sur un seul point de départ (l'habitant) et un seul point d'arrivée (l'équipement). L'usage d'*Anemone*<sup>3</sup> et cette condition a permis d'obtenir plusieurs parcours "naturels". C'est-à-dire, sans prise en compte du réseau.

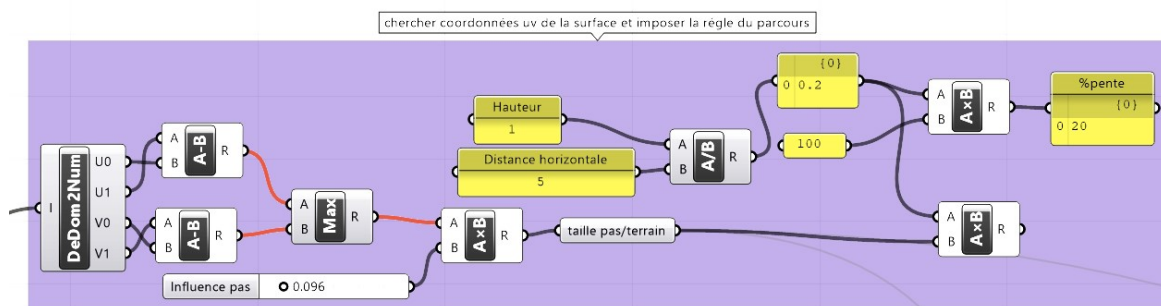
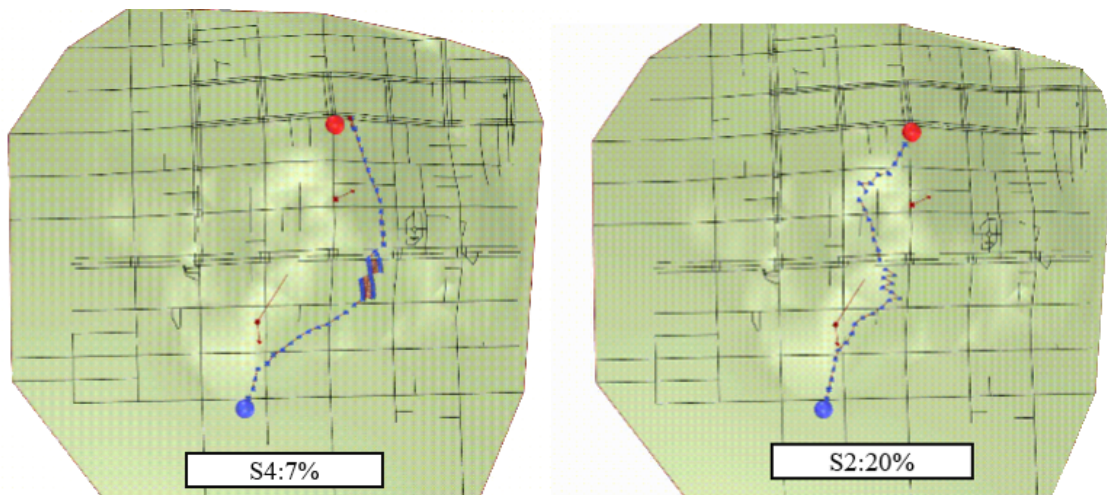


Fig 5.17 – Partie de l'algorithme utilisant la règle du déplacement du SMA

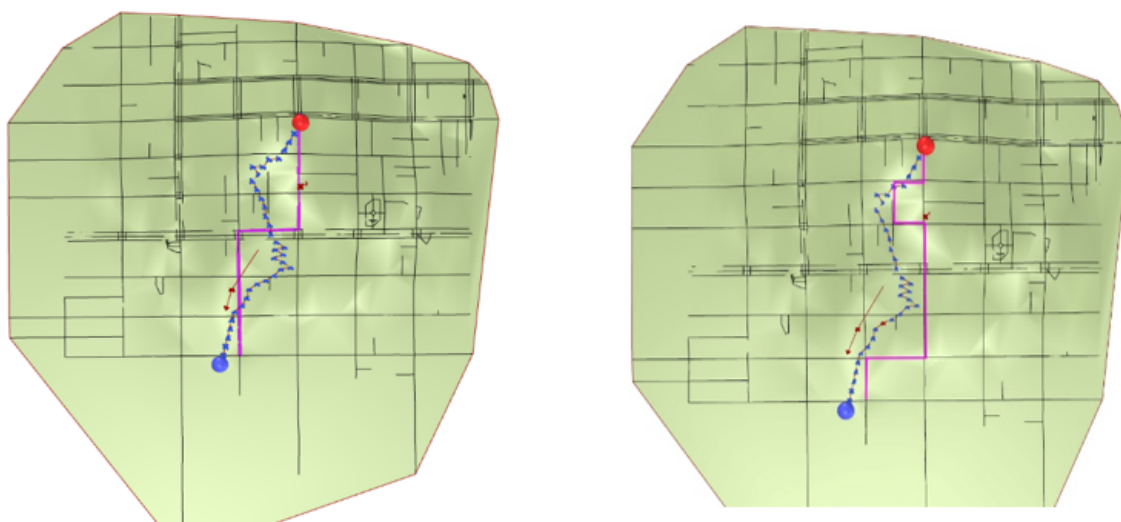
2. Système de simulation où les acteurs dits « agents » opèrent dans un environnement donné selon certaines règles ou relations imposées par l'utilisateur.

3. Plugin dans Grasshopper permettant de créer des boucles, normalement impossibles dans les fonctions natives de Grasshopper. Il est utilisé dans cette étude pour servir le comportement de l'agent.



**Fig 5.18** – Partie de la règle de la pente à suivre par le SMA pour l'extraction des chemins à une pente de 20% et exemples des résultats SMA avec deux pentes

Ces résultats sont d'abord présentés avec le SMA uniquement puis avec la prise en compte des réseaux, en utilisant le plugin *ShortestWalk*<sup>4</sup>, qui met en application l'algorithme A\*. Pour cela, le parcours du SMA par pente est transformé en une courbe et dernière est ensuite interpolée par le composant *interpolate*<sup>5</sup>, natif dans Grasshopper, et enfin transposée au réseau viaire d'un contexte par *ShortestWalk*.

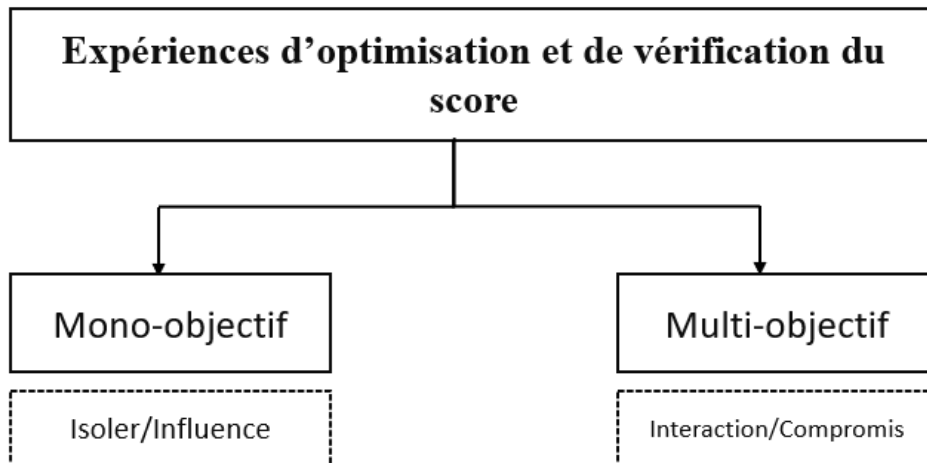


**Fig 5.19** – Transposition des parcours "naturels" du SMA vers le réseau viaire à travers le plugin ShortestWalk(A\*)

4. le plugin opère par la notion de graphes comme la S.S, pour trouver le chemin le plus court.

5. Ce composant permet, à travers un set de points, de simplifier une courbe existante. En l'occurrence ici, il s'agit du chemin tracé par le SMA qui est simplifié.

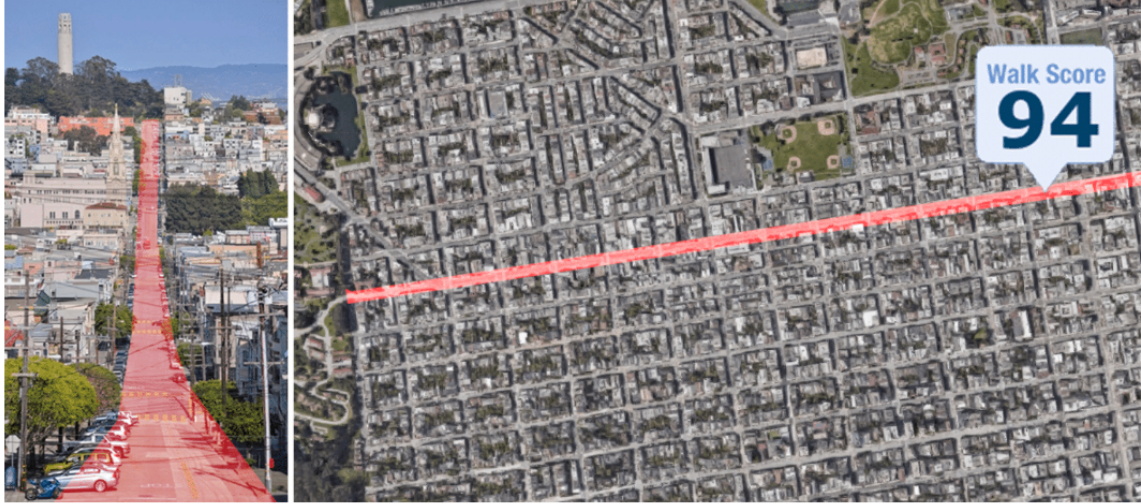
## 5.2 | Application du protocole : Les expériences d'optimisation



Nous avons deux types d'expériences :

- Les expériences mono-objectif servant à montrer l'influence de d'autres critères, ainsi qu'à démontrer que la relation de la distance à minimiser n'est pas toujours égale à un score de marchabilité meilleur (Hypothèse principale 1).
- Les expériences multi-objectif qui, soit font interagir deux mono-objectif, soit se basent sur des scénarios autres, permettent de montrer la nécessité de la méthode multi-objectif même pour des problèmes simples. (Hypothèse principale 2).

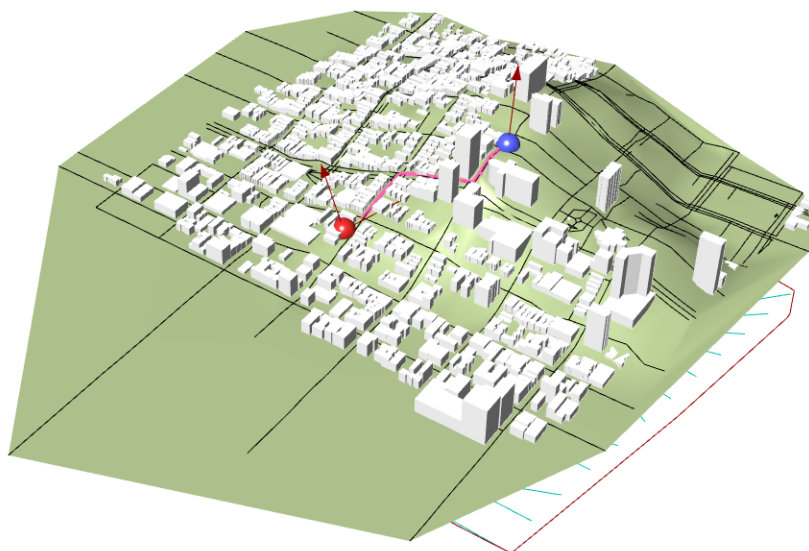
### 5.2.1 | Choix du contexte de l'étude



**Fig 5.20** – Le contexte de l'étude avec la rue principale et le score de marchabilité de l'aire de recherche dans l'expérience

Le contexte d'étude se situe à San Francisco(USA), plus précisément, dans le voisinage de Filbert Street. Ce choix est justifié par 3 arguments :

- Il permet de dépasser l'idée que l'algorithme est erroné uniquement car il n'a été conçu que pour un contexte américain principalement. En choisissant un contexte de là où il émane et en validant l'hypothèse, nous pourrions montrer que c'est la formulation du problème ou la relation elle-même qui est défailante et non pas les contextes.
- La rue principale à une pente de 40%, donc assez prononcée, pour que l'introduction de la topographie, comme nouveau critère, puisse nous permettre de constater son influence.
- L'accessibilité et la disponibilité des données telles que les tag « entrance » ainsi que le fichier shp qui va nous permettre de récupérer la topographie pour l'expérience SMA et l'intégration dans le module de calcul Urbano.



**Fig 5.21** – Le contexte extrait avec la topographie : Usage du fichier shp disponible en ligne

#### Le contexte et le score de marchabilité

Selon Walkscore, le contexte présente déjà un très bon score de marchabilité de 94%. En soi, le rajout des points (équipements) avec une affectation par défaut (épicerie) va augmenter ce score. Néanmoins, cette **augmentation** devra être différente, chaque fois, selon la position optimisée de ces équipements, par rapport à chaque critère. Justement si **elle** ne l'est pas, c'est que l'hypothèse principale est réfutée, car le score serait « indifférent » à la modification des critères.

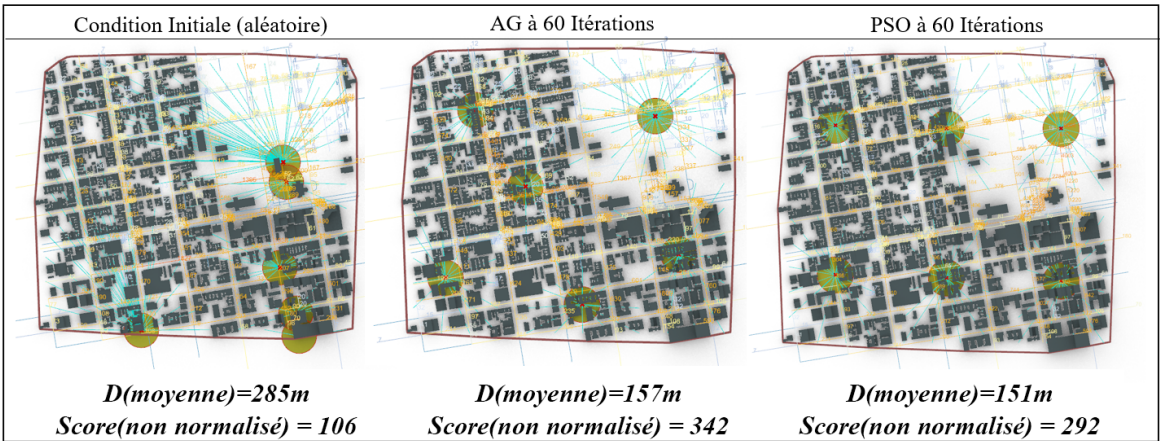
### 5.2.2 | Optimisations mono-objectif selon les critères :

#### 5.2.2.1 | Répartition Equidistante 2D : Limites : Equipements-Aire de recherche

L'expérience sert à replacer les points dans une position telle qu'ils soient répartis également dans la ville. Dans ce cas le résultat est assez prévisible (lorsque nous avons un nombre paire des points) mais nous allons noter un changement du à l'efficacité de l'outil qui va permettre de replacer la question de ce "type" de distance.

L'algorithme d'optimisation par essaim particulaires (PSO) a été reconnu comme étant plus efficace et plus rapide dans la recherche de l'optimum (ici la distance moyenne de la répartition équidistante). Cette faculté explique pourquoi, à 60 Itérations, il parvient à une distance plus minime que l'algorithme génétique. Néanmoins, il est à noter que même si il a

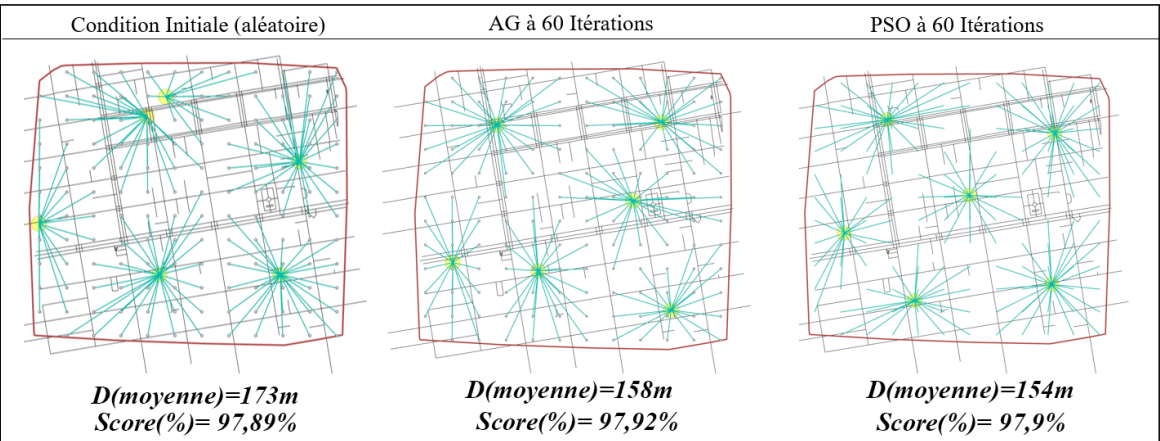
atteint l'optimum, il n'a pas donné un meilleur score de marchabilité (après avoir intégré la configuration obtenue dans le calculateur d'Urbano).



**Fig 5.22** – Résultat optimisation RE1

L'algorithme Génétique, en l'occurrence, n'ayant pas trouvé la « position » ou la distance optimale dans le temps donné, aboutit à une configuration autre qui elle induit un meilleur score et ceci s'expliquerait peut-être par la « distance decay function » qui tient compte du fait d'un certain « dégradé » selon la position des équipements eux-mêmes.

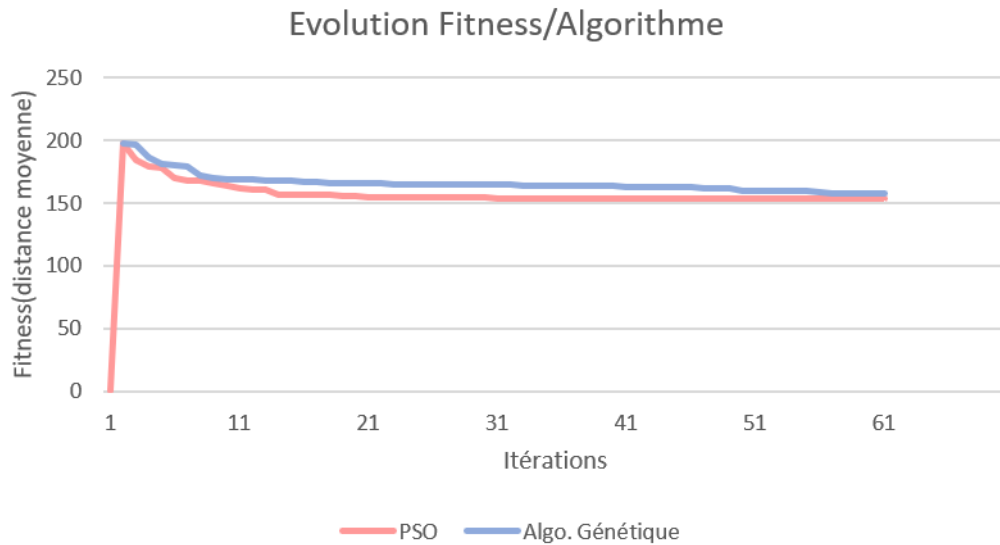
Par conséquent, la manière de procéder qui est « minimiser la distance aux équipements » dans le cas de la répartition équidistante, n'est pas toujours synonyme d'un meilleur score, et il n'y a donc pas moyen de prédire cette relation, autre que par la simulation informatique.



**Fig 5.23** – Résultat optimisation RE2 : Modification du nombre de points et nouvelles simulations

La nature des algorithmes, qui induit des résultats différents à chaque simulation, nous a

poussé à reprendre l'expérience, avec un nombre de points différents et une condition initiale autre, ainsi que des score normalisés (en%). Nous obtenons la même logique, qui est celle que pour un même nombre d'itérations donné (ici, choisi comme étant 60), l'efficacité du PSO permet d'avoir un meilleur optimum mais cette configuration, plus optimale, n'aboutit pas à un meilleur score de marchabilité.



**Fig 5.24** – Évolution de la fitness(distance minimisée) pour les deux algorithmes-cas RE

**Pour résumer l'analyse de l'expérience RE1 et RE2 :**

Pour l'instant, les hypothèses secondaires du mono-objectif sont, pour cette expérience du moins, validées : c'est-à-dire que :

- La relation « distance-marchabilité », telle que formulée actuellement, est fausse et n'est prédictible, donc il faudrait recourir à des simulations pour savoir ce qui augmenterait le score.
- Des résultats graphiques et numériques, qui n'ont pas atteint l'optimum ou ne semblent pas idéaux(distance moindre, mauvaise répartition), ont donné un meilleur score de marchabilité.
- Vu que nous avons formulé le problème, tel qu'il se fait actuellement, et nous avons obtenu des résultats inverses, cela fait que l'efficacité des algorithmes a été inversée.  
(Le PSO est reconnu plus performant que l'AG. Cependant, le score obtenu après Intégration de leur optimum, dans le calculateur, a que l'AG, par son optimum a donné un meilleur résultat.

Nous ne pouvons pas encore parler des hypothèses principales car elles impliquent d'essayer les autres critères également.

### 5.2.2.2 | Accessibilité A\* 2D : Limites : Équipements-Habitants-Réseaux

A la différence de l'expérience précédente, l'accessibilité implique de prendre en compte le réseau viaire et donc la mesure de la distance de celui-ci à travers A\*. Nous avons dans la section [5.1.2](Tag et automatisation de l'entrée),pour cela, proposé de corriger le repérage de l'entrée et le contexte donné émane d'une base de données ou ce renseignement existe(usage du tag).

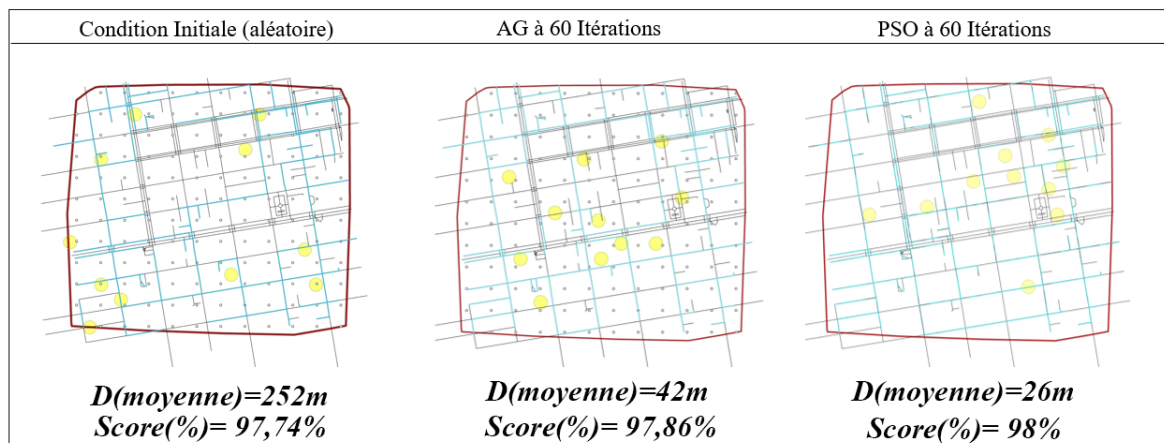


Fig 5.25 – Résultat d'optimisation A\*

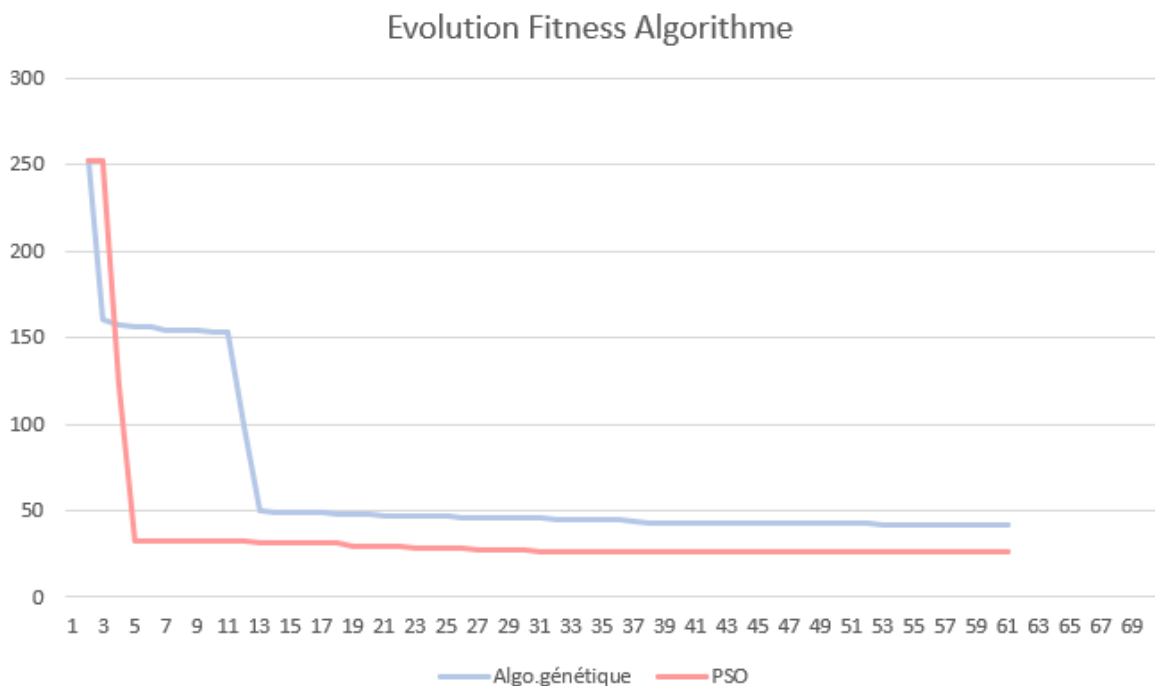
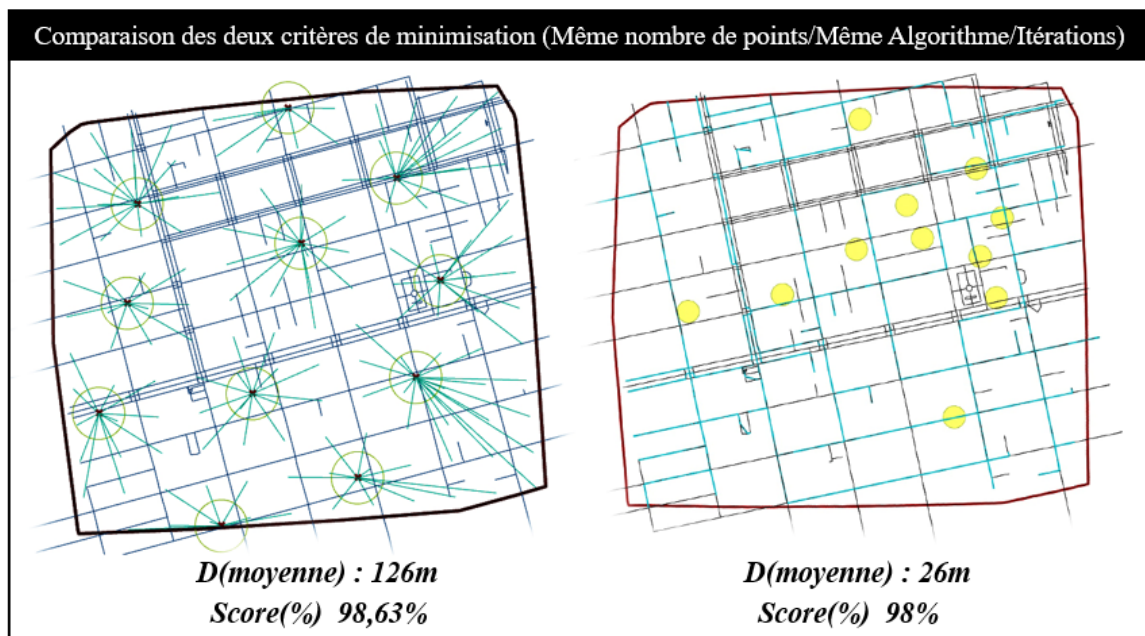


Fig 5.26 – Évolution de la valeur fitness pour les deux algorithmes - cas A\*

Ici, l'inverse de la première expérience est démontré. La distance moyenne obtenue par le PSO permet un meilleur score de marchabilité, et cette contradiction avec la première expérience renforce l'hypothèse que la simulation informatique est nécessaire et que la prédiction est impossible. Ajouté à cela, il serait judicieux de préciser le type de minimisation de distance, dans le premier cas, il s'agissait de la moyenne des liens avec l'équipement et dans le second (**milieu des carreaux fictifs, cas actuel**), c'est la moyenne des chemins les plus court ( $A^*$ ) pouvant être parcourus depuis l'entrée (**limite directe avec la rue-entrée**).



**Fig 5.27** – Comparaison  $A^*$  et RE : Même nombre de points/Algorithme mais différence du critère

Afin de vérifier l'influence des deux critères cités jusqu'à présent sur la marchabilité, nous prenons les optimums obtenus par le PSO et nous relevons les scores de marchabilité afférents à chacun. Pour la R.E, la distance est supérieure mais le score est meilleur, même si la différence dans l'optimum reste mineure (chiffres normalisés). Tant que cette différence existe, elle laisse supposer qu'il existerait bien une influence dans la différence des critères.

#### Pour résumer l'analyse de l'expérience accessibilité A\* et mise en relation avec la R.E

1) Les hypothèses secondaires du mono-objectif sont encore validées.

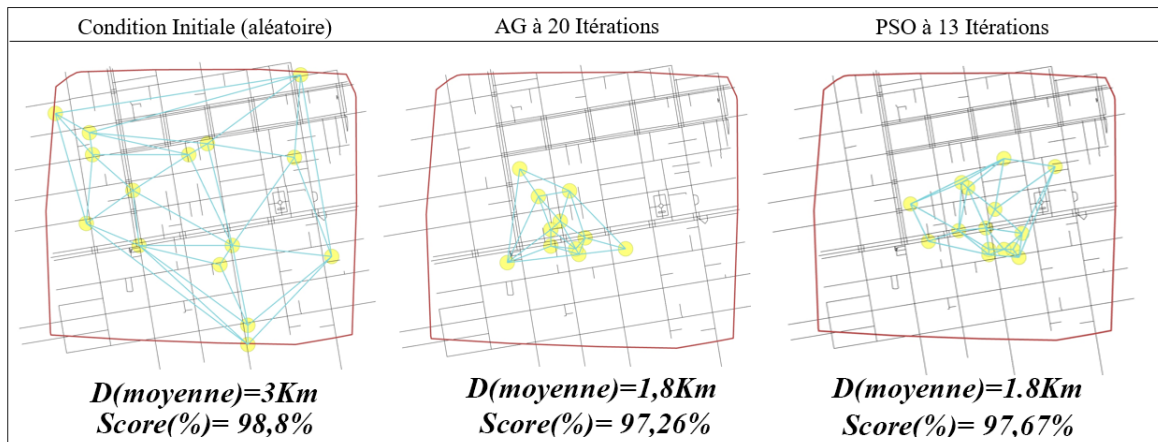
- La relation « distance-marchabilité » peut parfois aller dans le sens de l'état d'art ou contredire les travaux actuels, c'est-à-dire que nous n'avons aucune garantie ou connaissance au préalable si augmenter ou minimiser la distance, va induire un meilleur score ou pas.
- Un « meilleur » optimum peut survenir à travers configuration peu avantageuse (dans laquelle une partie du contexte n'est pas desservie. voir expérience PSO).

2) L'hypothèse principale **peut être validée ou pas**, nous ne pouvons pas l'affirmer, car même si la comparaison de deux optimums (même nombre de points, même algorithme, même nombre d'itérations et **critères différents**) a permis de déceler une légère différence, celle-ci peut être due aussi principalement à deux autres paramètres :

- Le mode opératoire des méta-heuristiques (en soi, pour un même critère, nous pouvons obtenir des résultats différents, alors nous nous pouvons pas trancher définitivement pour deux critères ou plus.)
- La Distance Decay Function (intégrée dans le module de calcul, afférente à Walkscore) : Les résultats peuvent varier (notamment pour A\*) car c'est la distance équipements-habitants (multiple points d'origine-réseaux) qui entre en jeu.

#### 5.2.2.3 | Schéma d'Alexander\* 2D : Limites : Équipements-Équipements

Le schéma d'Alexander est utilisé dans ce mémoire car il permet de réinterpréter le type de minimisation, toujours dans la continuité des deux premiers critères, et donc de se situer par rapport à une certaine limite ou relation. Dans ce cas précis, nous avons doublé les proportions du plan et la distance moyenne à atteindre est 1.8Km, tel qu'indiqué par l'ouvrage de C.Alexander.



**Fig 5.28** – Résultat optimisation 2D équipements-équipements basé sur le schéma d'Alexander

Outre l'efficacité du PSO en terme de rapidité à converger vers l'optimum, c'est aussi la condition initiale qui nous intéresse, car celle-ci, non optimisée selon le schéma d'Alexander, traduit un meilleur score, sans doute en raison de « la distance decay function » implémentée depuis walkscore, dans Urbano. Cela sert à démontrer que même dans un tel contexte, le schéma d'Alexander, tel que formulé dans ce problème d'optimisation, tend à faire émerger une répartition trop localisée et que la distance référence de 1,8Km(ou aux alentours) n'est pas susceptible, dans cette expérience, de favoriser la marche d'un lieu. Bien évidemment, la limite de cette expérience sera abordée dans le chapitre suivant.

#### Pour résumer l'analyse de l'expérience du schéma d'Alexander

1) Les hypothèses secondaires du mono-objectif sont encore validées.

- Dans le cas de ce type de minimisation, la relation telle que distance minimale = score maximal(travaux actuels) n'est pas toujours vraie. Dans cette expérience, la condition initiale aléatoire, pré-optimisation (3Km-98,8%), donne un meilleur score et dépassant les deux expériences d'optimisation.

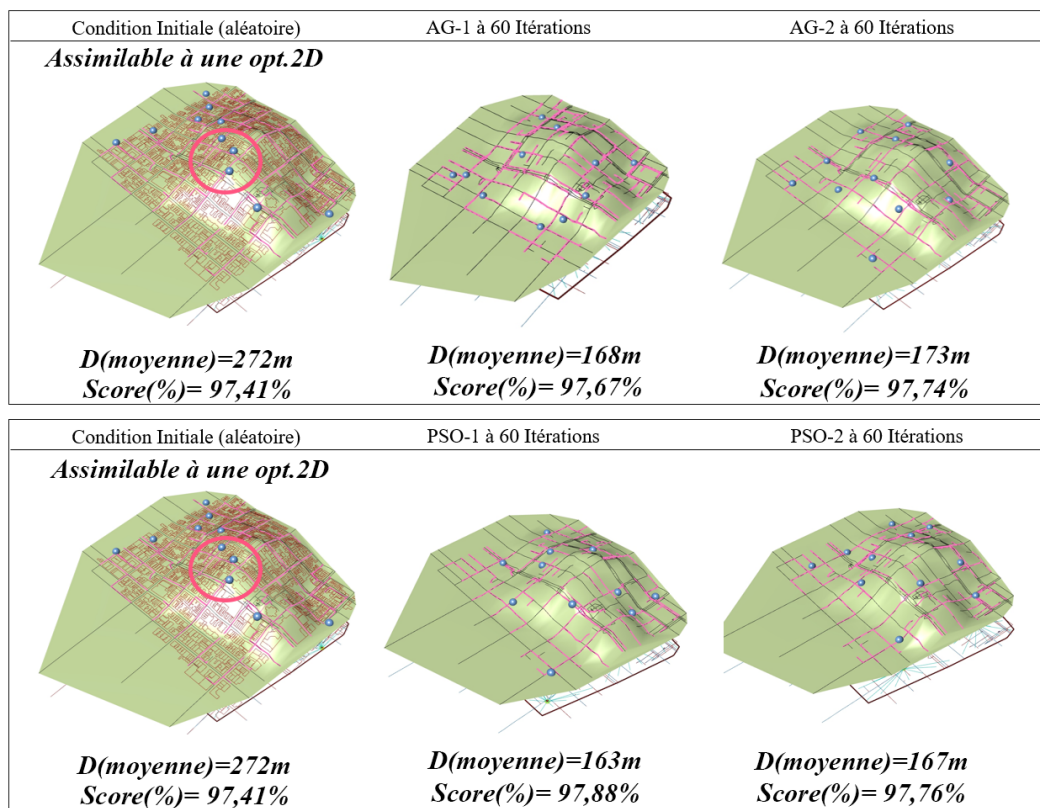
Pour une même distance référence obtenue par les deux algorithmes, nous remarquons un score différent. La variable serait donc la « Distance Decay Function »(DDF), **c'est-à-dire si nous avons le même nombre de point et le même distance moyenne entre ces points, c'est, ici, leur répartition spatiale qui impose cette différence de score.**

Ce constat sur l'influence de la DDF pourrait donc renforcer la limite que nous avons traitée dans l'expérience précédente qui est : Ce serait la DDF qui impliquerait la différence et non réellement la différence du critère, de ce fait, **la méthode que nous avons choisie ne nous permet pas de vérifier la première hypothèse principale totalement.**

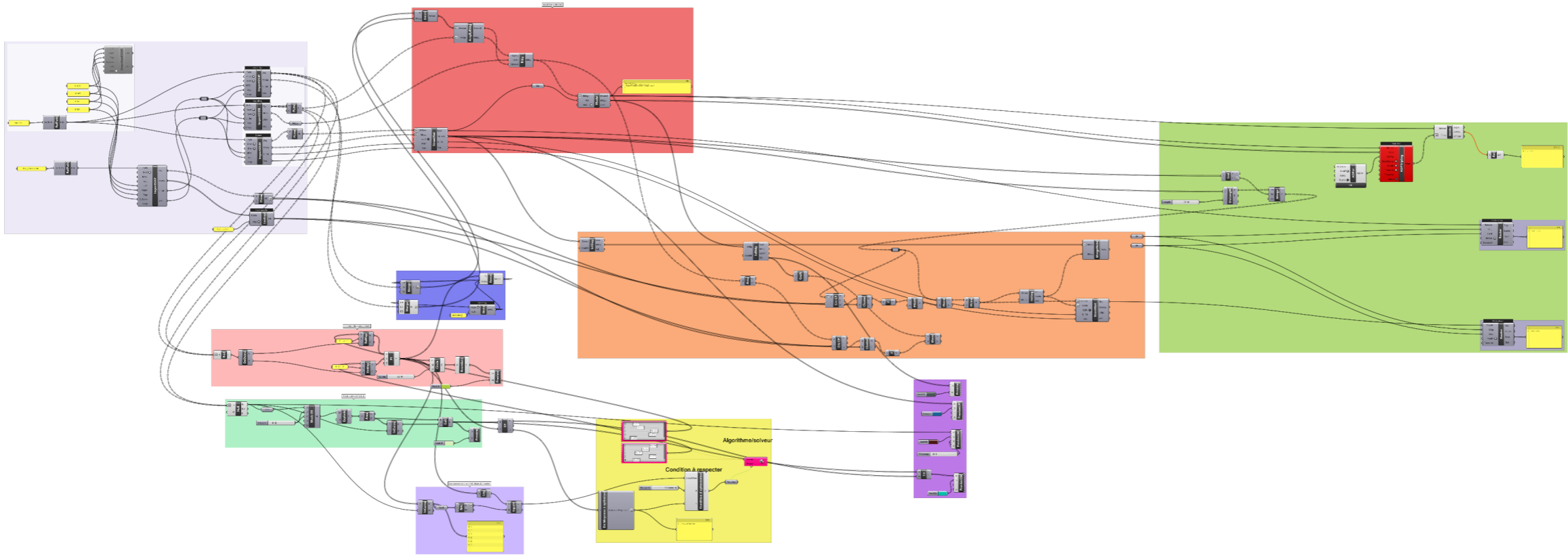
#### 5.2.2.4 | Optimisation(3D) A\* en intégrant la topographie :

Après avoir repéré les chemins à +20% de pente qui ont été posés comme condition à éviter(extraction par un SMA pré-optimisation), l'optimisation par l'intégration de la topographie nous permet de définir un réel apport à ce que l'état d'art présente actuellement et qui a été dénoté comme critique aux algorithmes de calcul : Se limiter à des analyses ou simulations/optimisations en plan.

Afin d'obtenir des résultats assez certains, nous avons refait l'expérience plusieurs fois sur Galapagos (Algorithme Génétique) et sur SilverEye (P.S.O). **La condition initiale**, quant à elle, a fait l'objet d'une recherche de coordonnées(x,y) visée. Bien qu'aléatoire, nous avons modifié le gene Pool de façon à ce que certains points puissent se retrouver dans la « pente à éviter » pour pouvoir comparer ce score assimilable à une répartition en 2D(ce qui se fait actuellement) aux scores optimisés par intégration de la topographie(Simulations AG et PSO/recherche dans le mémoire).



**Fig 5.29** – Résultat optimisation A\* avec l'intégration de la topographie : Au préalable avec l'expérience SMA et extraction du fichier shp



Extraction des données depuis OSM	Construction du modèle de calcul Urbano	Intégration de la topographie dans le modèle de calcul	Répresentation des équipements et des réseaux
Position des points cherchés reamppé	Condition “les équipements sont ils dans la courbe”	Fonction objectif GALAPAGOS et Condition = 1	Résultat WALKSCORE en 2D et avec la Topographie

Les optimisations effectuées permettent de dénoter que l'intégration de la topographie et l'évitement des chemins pentus présentent un meilleur score, donc, inversement la pente pourrait influencer à faire décroître la marchabilité. Ce résultat présente, cependant, une certaine incohérence du fait que nous pouvons imaginer que le chemin est étudié dans un seul sens.

Pour résumer l'analyse de l'expérience de la topographie
<p>1) L'hypothèse principale est partiellement validée car :</p> <ul style="list-style-type: none"> <li>Comparée à la condition initiale (qui est assimilable aux travaux actuels, sans prise en compte de la topographie) le résultat est différent et il est meilleur. Cela veut dire que l'algorithme construit sur grasshopper, permet de voir et de <b>constater avec un résultat numérique, comment en évitant ces chemins pentus, la marchabilité augmenterait</b> et que <b>ce critère influe le résultat, alors qu'il n'est pas inclut dans la relation mathématique actuelle.</b></li> </ul> <p>2) Les hypothèses secondaires sont validées par cette expérience aussi :</p> <ul style="list-style-type: none"> <li>Entre AG-1 et AG-2, pour un même nombre d'itérations et un même nombre de points et le même critère, un optimum qui a rempli moins bien l'objectif (AG-2, distance supérieure à AG-1) qu'un autre (AG1, distance plus minimisée) a donné un meilleur résultat de marchabilité. Donc, à nouveau, il n'y a <b>pas de relation automatique et prévisible</b> pour optimiser la marchabilité.</li> <li>Les expériences d'optimisations, dans ce mémoire, ont pour but de <b>vérifier l'influence et des différences de score</b> et non pas réellement d'optimiser ou de proposer une méthode de conception propice à la marche. De toute évidence, nous n'opérons pas ainsi à « implanter » des équipements selon la topographie et le <b>résultat graphique/configuration urbaine est « contre-intuitif »</b>, avec une mauvaise desserte du contexte. Néanmoins, ces configurations ont abouti à un meilleur score de marchabilité que la base initiale.</li> </ul> <p>Cette expérience a nécessité de faire appel à des méthodes SIG externes (à grasshopper) ainsi qu'à un travail avec un SMA avant de pouvoir procéder à l'optimisation elle-même.</p>

### Synthèse des expériences mono-objectif :

- Les hypothèses secondaires sont validées.
- L'expérience principale est impossible à vérifier complètement à cause de la nature des outils utilisés et de la Distance Decay Function.
- Pour pouvoir la valider, d'une manière sure, il aurait fallu avoir recours à un algorithme déterministe, comme point de référence. Se rapporter au chapitre suivant pour cela.

### 5.2.3 | Optimisations multi-objectif

L'optimisation multi-objectif vise à créer des scénarios de conception urbaines susceptibles de favoriser la marchabilité ; en mettant en relation certains critères abordés précédemment ou en invoquant d'autres.

Il est à rappeler que vu que nous parlons de multi-objectif, l'existence du Front de Pareto, abordé dans la section [4.2.3.3], ne permet pas l'obtention d'une mais de plusieurs « couples/-triplés de solutions » et c'est ce que le solveur d'Octopus(Algorithme évolutionnaire HypE) va nous présenter.

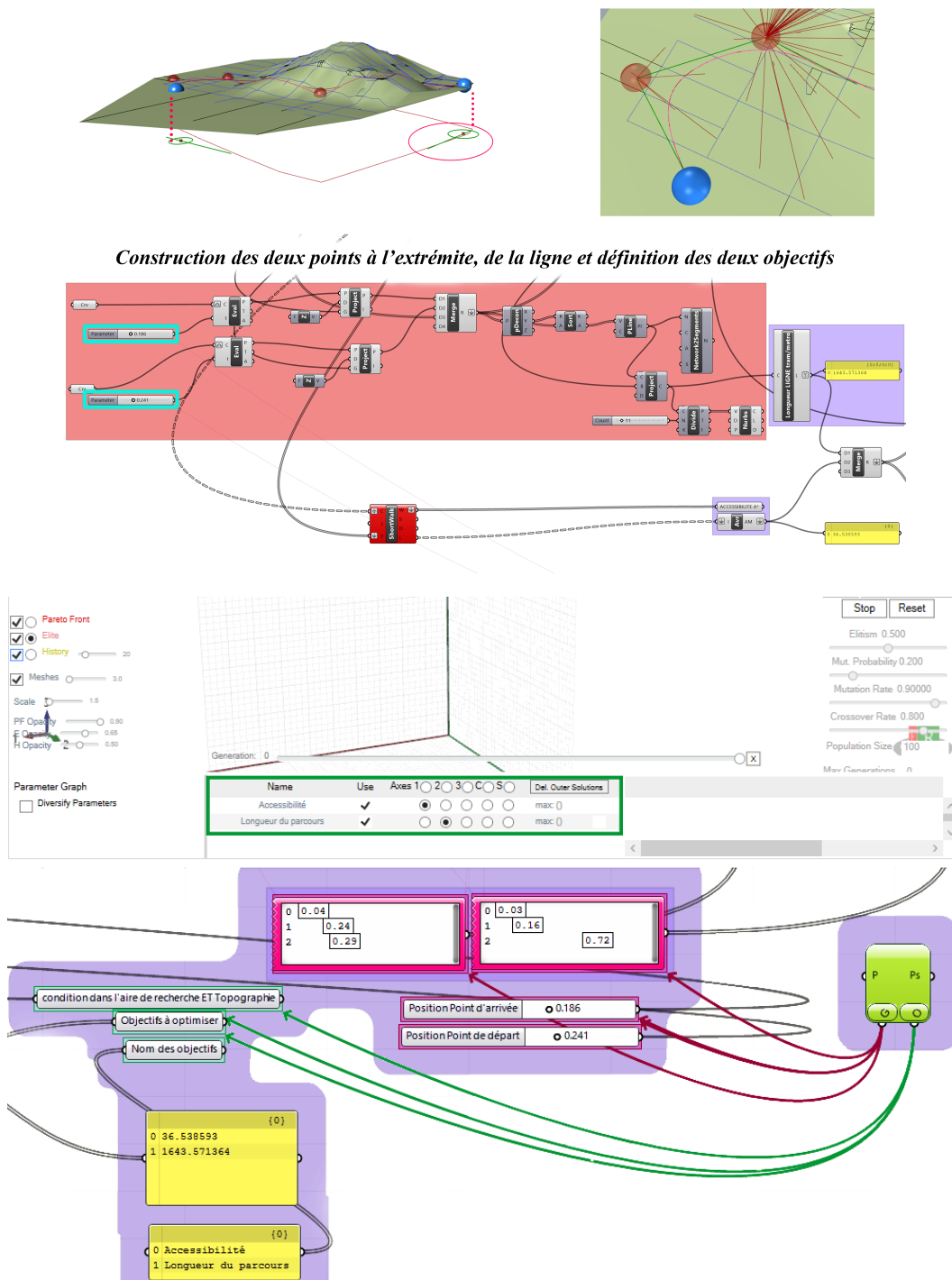
#### 5.2.3.1 | Minimiser la longueur d'une ligne de tram et maximiser l'accessibilité $A^*$ aux équipements passant par cette ligne

Nous nous sommes intéressés dans ce scénario au principe du T.O.D abordé dans la partie théorique[2.2.1] et qui est afférent à la notion de la marchabilité. L'idée étant que bien évidemment, l'existence d'une gare va permettre la densification d'un lieu et que cela va donc générer plus d'équipements et de fréquentations.

Pour cela, les deux stations sont d'abord représentées sur le plan sur une ligne qui constitue la limite de leur position et ce afin d'essayer de les garder en tant que point d'arrivée et de fin et non pas de les avoir disposés au milieu du contexte.

Cette démarche est réalisée par le composant *evaluate curve* qui permettra de mouvoir un point sur les deux lignes. *Evaluate curve* prend en entrée les deux *sliders* numériques et qui ,eux même, plus tard, feront partie des variables à introduire, entre autres, dans le solveur multi-objectif : Octopus.

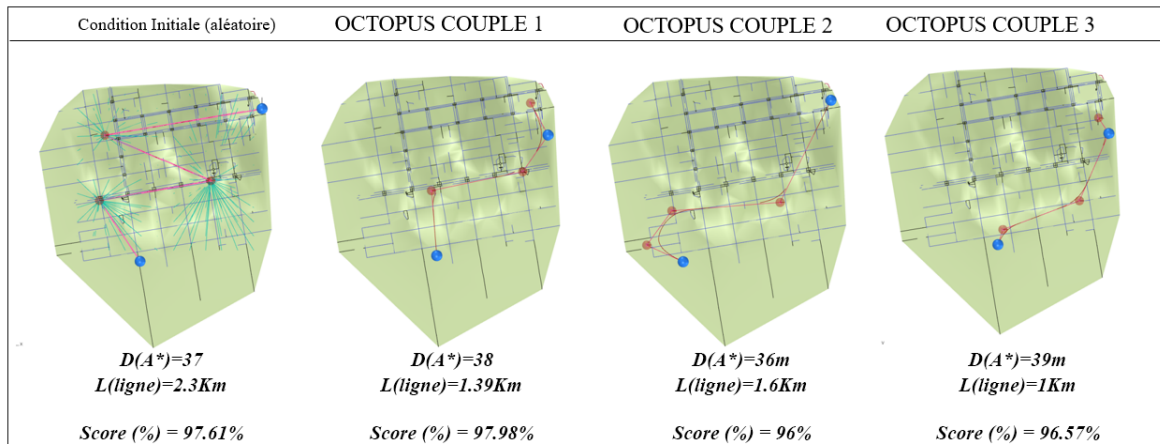
Les équipements, traversant cette ligne, sont disposés de la même façon avec un *gene pools* et l'ensemble est ensuite lié par une *polyline* qui, pour plus de flexibilité, est convertie en une *nurbs curve* à travers un composant du même nom.



**Fig 5.30** – Définition algorithmique du scénario : construction des variables et objectifs et introduction dans le solveur d'Octopus

auteur

Les deux objectifs ainsi que les 4 variables sont ensuite introduites dans le solveur d'Octopus.



**Fig 5.31** – Résultat optimisation multi-objectif avec la condition topographie également

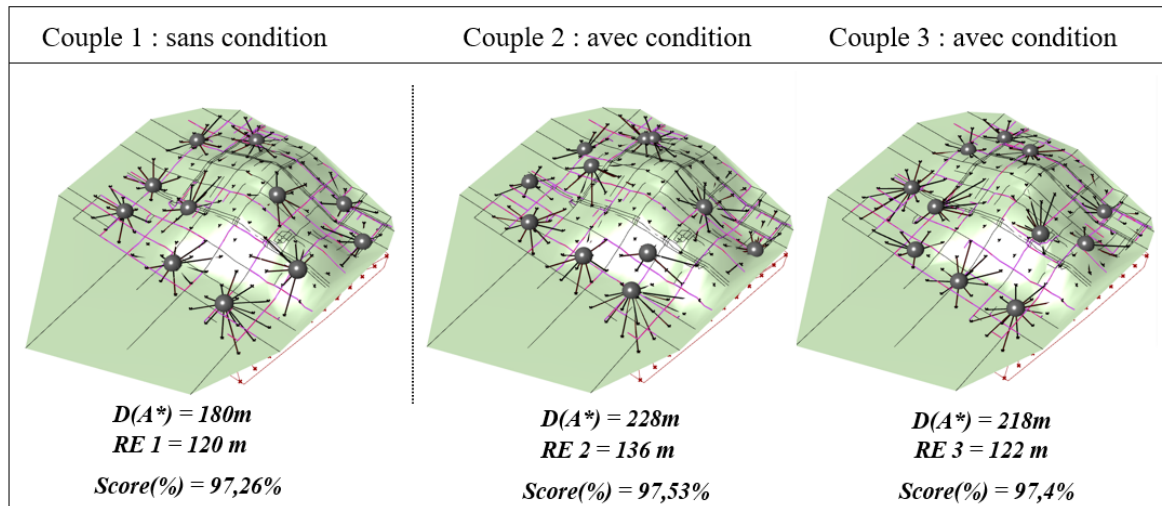
Les résultats présentent à la fois la condition initiale avec une longueur de ligne de 2.3Km et une Accessibilité  $A^*$  de 37.38, ainsi que 3 couples de solutions choisis dans le front Pareto donné par le logiciel d'Octopus dites aussi « instances ».

Le meilleur couple pour le score de marchabilité est le premier avec un score de 97.98%, comparé à une chute de score pour les deux autres couples et ce même si les objectifs ont l'air d'avoir été plus « satisfaits » pour ceux-là (couple 3 et couple 4). Dans tous les cas, nous noterons la notion du compromis propre au multi-objectif.

Pour résumer l'analyse de l'expérience 1 du multi-objectif
<p>1) L'hypothèse secondaire du multi-objectif est validée et indirectement celle du mono-objectif aussi :</p> <ul style="list-style-type: none"> <li>• La notion du compromis dans le multi-objectif pourrait avoir donné une configuration peu avantageuse avec le couple de solutions (3) mais ce dernier réalise un meilleur score de marchabilité que le couple (2).</li> <li>• Cela valide indirectement les expériences précédentes du mono-objectif, pour l'hypothèse qui stipule qu'une distance moindre n'est pas toujours égale à une meilleure marchabilité. En effet, le couple(2) présente une distance <math>A^*</math> inférieure à tous les autres mais aussi le score de marchabilité le plus bas.</li> </ul> <p>2) L'hypothèse principale du multi-objectif, comme pour le mono-objectif, ne peut être validée totalement car la différence du score peut-être due au simple fait d'avoir la Distance Decay Function et non à l'ajout de différents critères/scénarios.</p>



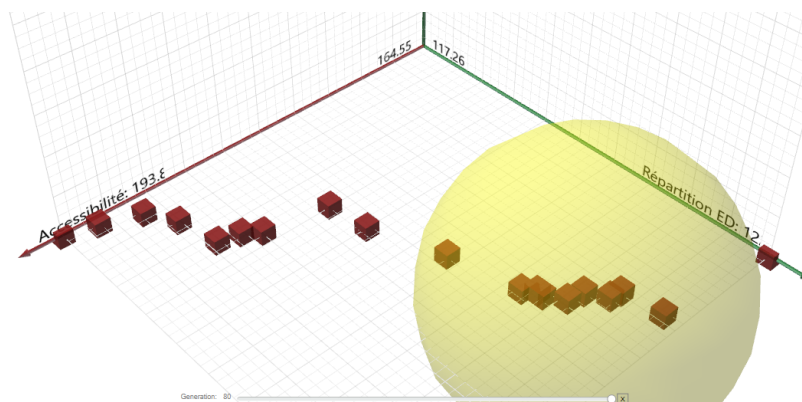
rapport à la première simulation(couple1) avec des distances supérieures mais un score de marchabilité meilleur. Le fait d'obtenir des résultats inverses signifie que la minimisation de distance ne définit pas, nécessairement, une bonne marchabilité.



**Fig 5.33** – Des instances de solutions capturées dans 2 types de simulations différentes : Le couple 1 : un résultat sans la condition et le couple 2 et 3 pendant une seconde simulation avec condition de la topographie

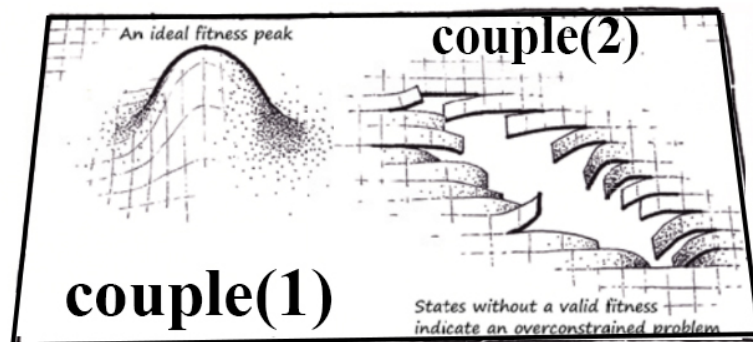
auteur

Le couple (3), quant à lui, semble moins satisfaire la condition de la topographie, même si celle-ci était imposée. Si nous nous rapportons à la satisfaction des objectifs, il aurait fallu éliminer donc la condition du terrain (Couple 1) et si nous souhaitons optimiser la marchabilité, c'est le couple 2 qui est à choisir.

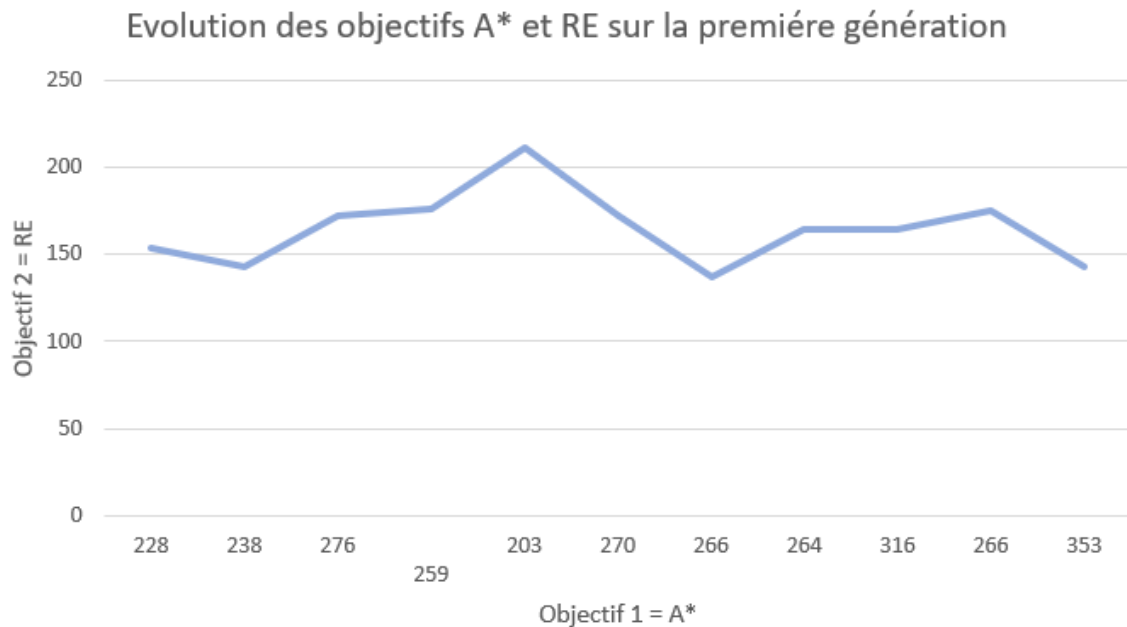


**Fig 5.34** – Le front pareto dans l'espace de recherche de la simulation sans condition dans laquelle nous avons choisi une instance, parmi les solutions du front (couple 1)

La formulation de ce problème, bien que simple mais avec la contrainte des objectifs contradictoires peut avoir donné lieu à un paysage de recherche(abstrait), dont la Fonction Fitness et la contrainte, ensembles, n'aboutissent pas au pic du paysage et donc pas d'optimums relativement.Nous avons évoqué les divers types de paysage de recherche selon la formulation du problème dans la section [4.2.3.2].



**Fig 5.35** – En relation avec les paysages abstraits présentés par Rutten, la formulation du problème avec des contraintes ou des objectifs contradictoires pourrait induire des paysages sans "optimum idéal



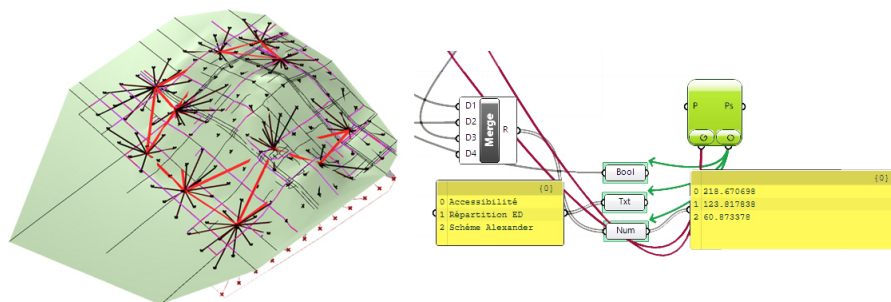
**Fig 5.36** – Évolution des couples de solution dans une génération dans Octopus

### Pour résumer l'analyse de l'expérience 2 du multi-objectif

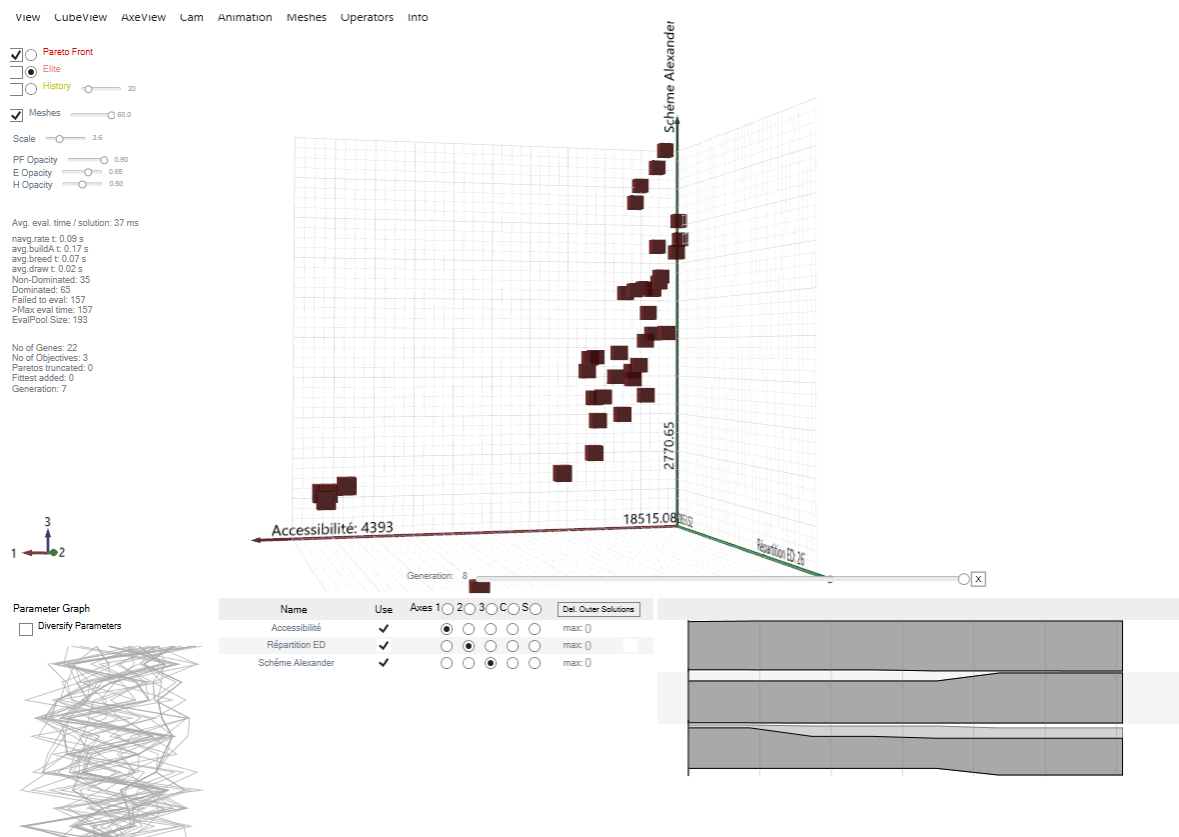
- 1) L'hypothèse secondaire est **définitivement validée** pour le mono(indirectement) et le multi-objectif(directement).
  - La minimisation des distances(quelque soit la limite/type) n'implique pas nécessairement une augmentation du score.
  - Des configurations moins satisfaisantes donnent un meilleur score.
- 2) L'hypothèse principale du multi-objectif peut-être **partiellement validée** :
  - Les deux couples avec la condition topographie(éviter les chemins repérés par le SMA), lorsque comparés avec la version mono-objectif(A\* en intégrant la topographie), présentent des résultats de marchabilité moins bons(nous avons le même nombre de points, c'est uniquement la méthode (mono et multi-objectif qui varie).Donc la différence des critères et leur interaction, dans le compromis, montre que même pour un problème simple comme celui-ci, le multi-objectif pourrait être nécessaire.
  - L'hypothèse principale, comme pour le mono-objectif, **n'est jamais totalement vérifiable** en raison de l'usage d'algorithmes méta-heuristiques.

#### 5.2.3.3 | 3 objectifs : RE, Accessibilité A\* et schéma d'Alexander :

Dans cette dernière recherche de l'optimum, nous ajoutons un dernier objectif qui est le schéma d'Alexander avec la distance de référence de 1.8Km à atteindre.Pour cette raison, nous avons, à nouveau, revu les proportions du plan afin qu'ils puissent, pour les 3 objectifs, avoir des valeurs concordantes. Nous laissons Octopus tourner sur 80 itérations,cette fois-ci, étant donné l'ajout d'un troisième objectif.

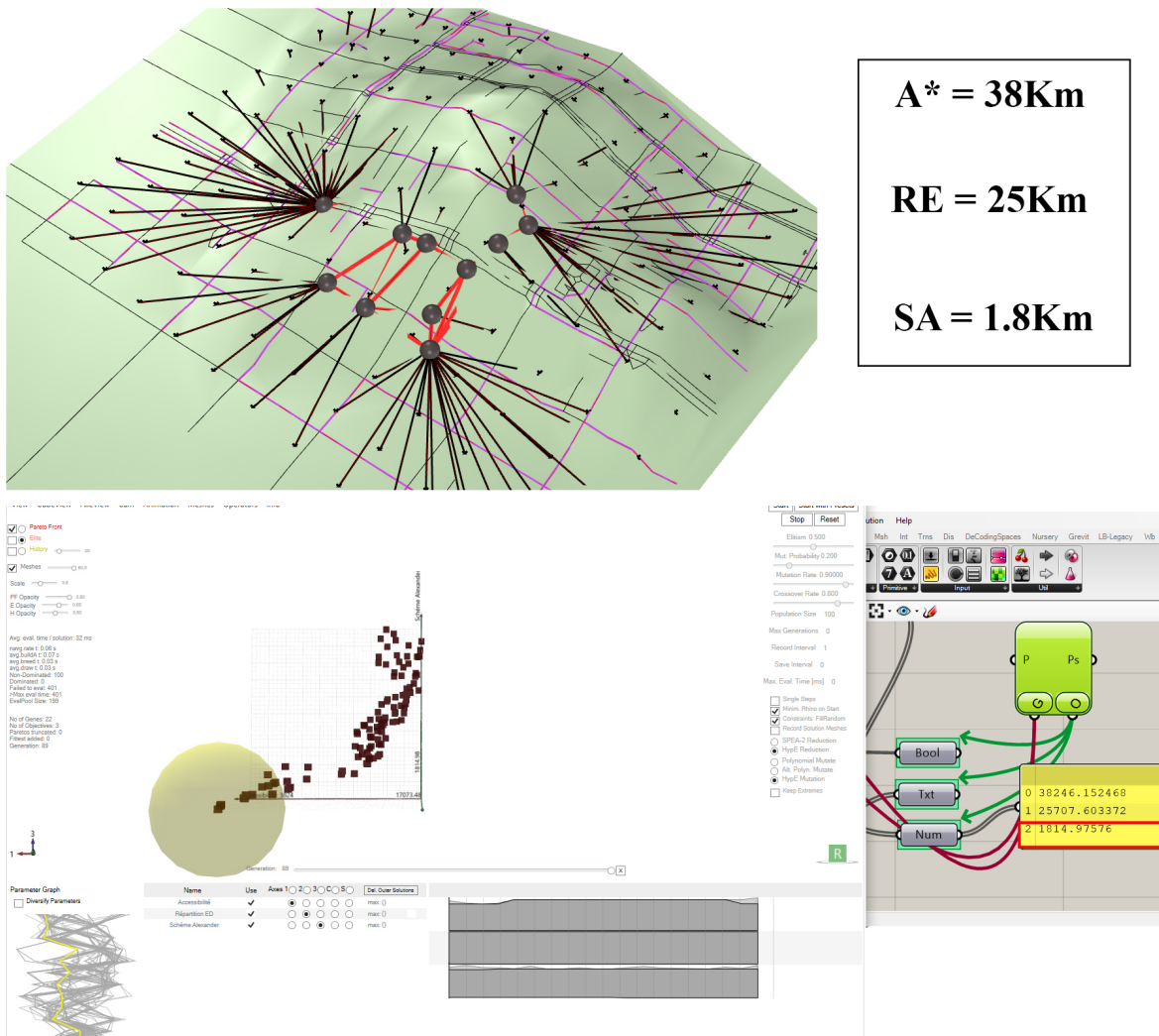


**Fig 5.37** – Mise en place des 3 objectifs dans le canvas de Grasshopper avec Octopus



**Fig 5.38** – Introduction d'un troisième objectif : Le schéma d'Alexander et exploration sur un troisième axe dans le paysage de recherche d'Octopus

Les 3 objectifs présentent des contradictions en terme de limites/types de minimisation, c'est à dire par rapport à quoi/qui minimisons-nous. Pour cela, même pour un problème simple comme celui-ci, nous pensons que remplir, à tout prix, un coût optimal telle que la distance référence du schéma théorique, va détériorer complètement les deux autres objectifs. Cependant, les expériences précédentes nous ont montré que, vu que le problème est mal formulé actuellement, de "mauvais" optimums peuvent aboutir à un meilleur score.



**Fig 5.39** – En haut : la configuration obtenue et en bas : Instance de solution qui a satisfait l'objectif d'Alexander et les valeurs correspondantes pour A\* et la RE dans Grasshopper et Octopus

La topographie est, du fait de la contradiction des 3 objectifs en soi, omise. L'ajout de la condition topographie = 1 fait que le processus de la convergence vers la solution n'avait pas démarré. Pour cette raison, nous avons aboli la condition du terrain.

---

### Pour résumer l'analyse de l'expérience 3 du multi-objectif

- Même pour une formulation aussi simple que « minimiser la distance par rapport à ... » le multi-objectif permet de déceler l'incompatibilité de certains critères les uns aux autres. **Ceux-ci sont choisis volontairement afin d'avoir une certaine contradiction, pour montrer que l'optimisation multi-objectif, est nécessaire dans même pour des problèmes simples**, où nous avons recours parfois au mono-objectif séparé.
- Parfois, le multi-objectif invalide les résultats obtenus au cours du mono-objectif ou encore ,comme pour l'ajout de la condition topographique, la recherche de la solution ne démarche même pas et l'algorithme reste bloqué car le problème est trop contraint ou que les objectifs ne sont pas bien formulés à l'égard de l'intégration de la topographie.

### Synthèse des expériences multi-objectifs :

- Les hypothèses secondaires sont définitivement validées.
- L'hypothèse principale, pour ces expériences, est partiellement validée : le recours aux algorithmes multi-objectif est parfois nécessaire même pour des problèmes aussi simples.

## 5.3 | Conclusion partielle

Les expériences réalisées, tant pour le mono-objectif que pour le multi-objectif, nous ont permises de constater le changement de résultat dans le score de marchabilité et ce, avec l'intégration d'autres critères. Le principal critère introduit dans ce mémoire, comme première approche des simulations informatiques sur la marchabilité, prend en compte la troisième dimension : la topographie du terrain. C'est ce critère-là qui a principalement motivé notre choix du contexte d'étude. La topographie nous a permis de constater de possibles défaillances dans la façon de formuler le problème, parfois avec des contraintes tellement chargées, que l'algorithme d'optimisation n'entame même pas la recherche de l'optimum.

L'optimisation de la distance, selon ces autres critères, a abouti par des scores contradictoires et a permis la validation des hypothèses secondaires. En effet, nous avons montré que les travaux ou postulat actuel, telle qu'une distance minimale [Cichoka, 2015] par rapport à une ou des origines données, est égale à une meilleure marchabilité, n'est pas nécessairement toujours applicable. Dans ce cas, cela sous-entend qu'il faudrait repenser à la manière de reformuler le problème d'optimisation et donc la fonction objectif à introduire dans l'algorithme.

Cependant, ces expériences restent très critiquables car, paradoxalement, alors que nous

avons essayé d'affiner le résultat actuel, le contexte ainsi que les critères eux même sont étudiés d'une manière isolée (Se référer à la section [6.1.2]). La méthode suivie est également défaillante car elle ne nous a pas permis de valider l'hypothèse principale du mono-objectif, qui était de savoir si le résultat est différent à cause de l'introduction de nouveaux critères ou à cause de l'usage de métaheuristiques. Pour cela, nous proposons une méthode de correction dans la section [6.1.1]

<b>Expériences du mono-objectif : Vérifier la prédictibilité de la relation et de l'influence des critères sur le score</b>		
<b>Critère</b>	<b>Algorithme bio-inspiré</b>	<b>Algorithme Génétique</b>
		<b>Optimisation par Essaim particulière</b>
Répartition Equidistante 2D (Equipements-aire)	6 points - Condition initiale (avant optimisation) D = 285m et Score = 106	
	D = 157 m et Score = 342	D = 151 et Score = 292
	11 points – Condition initiale (avant optimisation) D= 173 et Score = 97,89%	
	D = 158 m et Score = 97,92%	D = 126 m et Score = 97,9%
Accessibilité A* 2D (Equipements-multiple points)	11 points - Condition initiale (avant optimisation) D = 252m et Score = 97,74%	
	D = 42m et Score = 97,86%	D= 26m et Score = 98%
	D = 31 m et Score = 97,89%	D = 37m et Score = 97,91%
Exemple : Comparaison RE(PSO) et A*(PSO) – même nombre de points	<b>D = 154m et Score = 98,63% contre D = 26m et Score = 98%</b>	
	<b>D = 154m et Score = 98,63% contre D = 37m et Score = 97,91%</b>	
Schème d'Alexander (Equipements-Equipements)	Condition Initiale (avant optimisation) D = 3Km et Score =98,8%	
	D= 1.8Km et Score = 97,26%	D = 1.8Km et Score = 97,67%
Accessibilité A* 3D en intégrant la topographie du terrain (préparation SMA+shp)	Condition Initiale (assimilée volontairement à une analyse 2D et les points sont disposés sur la pente)	
	D=272 m et Score = 97,41%	
	D = 168m et Score = 97,67%	D = 163m et Score = 97,88%
	D = 173m et Score = 97,74%	D = 167m et Score = 97,76%

**Tableau 5.1** – Tableau synthétique avec toutes les valeurs obtenues par critères et par algorithme d'optimisation

## Limites de l'étude et perspectives

### 6.1 | Limites et regard critique sur les expériences réalisées :

#### 6.1.1 | Limites relatives à la nature des outils utilisés :

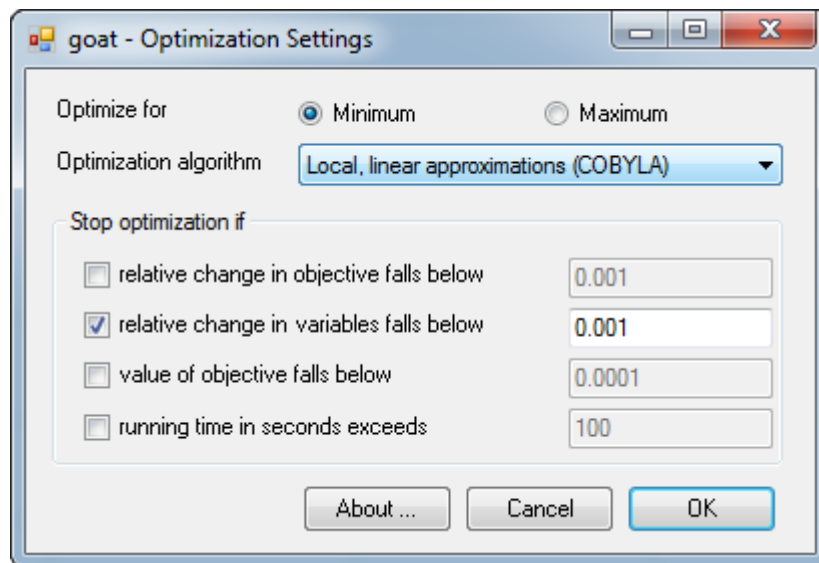
Nous avons évoqué dans la section[4.2.3.2] que les métaheuristiques, dont les algorithmes bio-inspirés, permettent d'outrepasser le problème de l'optimum local. Ils sont d'autant plus accessibles aux architectes, à travers les solveurs implémentés dans Grasshopper et Rhino. Néanmoins, ils ne permettent pas d'avoir le même résultat à chaque simulation ce qui accentue le biais de confirmation des hypothèses.



**Fig 6.1** – Logo GOAT

C'est aussi ce qui justifiait l'usage de deux algorithmes différents sur plusieurs simulations pour valider la recherche mais nous ne saurons jamais si une éventuelle simulation pourrait produire un résultat contradictoire à ce que nous avons avancé. C'est-à-dire, est-ce que la différence, même minime, obtenue est due à l'influence des critères ou bien à l'indéterminisme des algorithmes utilisés, qui fournissent un résultat différent à chaque simulation.

Pour cela, il existe sur grasshopper un solveur au nom de « GOAT » et qui use d'un algorithme COBYLA (programmation linéaire) pour effectuer des optimisations déterministes et donc donnant un même résultat pendant toutes les simulations et pouvoir trancher.



**Fig 6.2** – Interface du solveur GOAT avec l’algorithme déterministe de prog.linéaire COBYLA

Le problème de l’optimum local serait peut-être dépassé même avec cette recherche locale dans le cas de problèmes à moindre complexité et taille comme celui-ci. Nous aurions donc dû prendre le solveur GOAT, comme une optimisation de référence. Cette dernière nous aurait permis de comparer les résultats obtenus par les algorithmes non-déterministes, pour pouvoir se prononcer définitivement sur l’influence des résultats ou non, sur le score final.

### 6.1.2 | Limite relative à la taille du problème formulé et au processus conceptuel :

En raison des simulations de calcul parfois lentes dans « Urbano », la zone d’étude est volontairement restreinte, même si elle nous a permis de faire des constats ou de valider ou de réfuter certaines hypothèses, cette restriction engendre la même défaillance que Ratti avait évoqué pour la space syntax et que nous avons abordé dans la section [2.2.2.4]. Ce problème, qui ressemble à celui de l’optimum local algorithmique, est notamment accentué dans l’expérience d’Alexander. Dans cette dernière, nous avons réduit le calcul à un seul réseau d’équipement pour tester le schème urbain qu’il a présenté dans son livre

Néanmoins, sur un plan concret, la ville est faite de clusters de réseaux d’équipements et donc le fait que nous ayons isolé un seul et déduit que celui restreignait, au contraire la marchabilité, contrairement à ce qu’impliquait la théorie « A pattern language » n’est pas totalement valable. Mais en règle générale, la réflexion de départ impliquait que si nous introduisons un

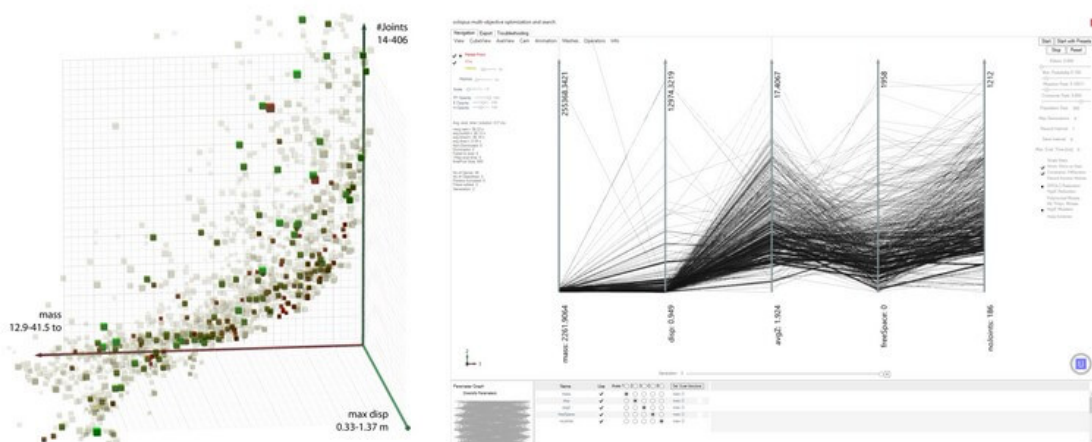
élément et que celui-ci provoquait une différence dans le score, alors nous aurions conclu que la relation actuelle est fausse.

## 6.2 | Les alternatives : vers d'autres modes de représentation :

### 6.2.1 | Représentation multi-variée d'un paysage de recherche abstrait multi-dimensionnel pour l'aide à la décision de l'architecte

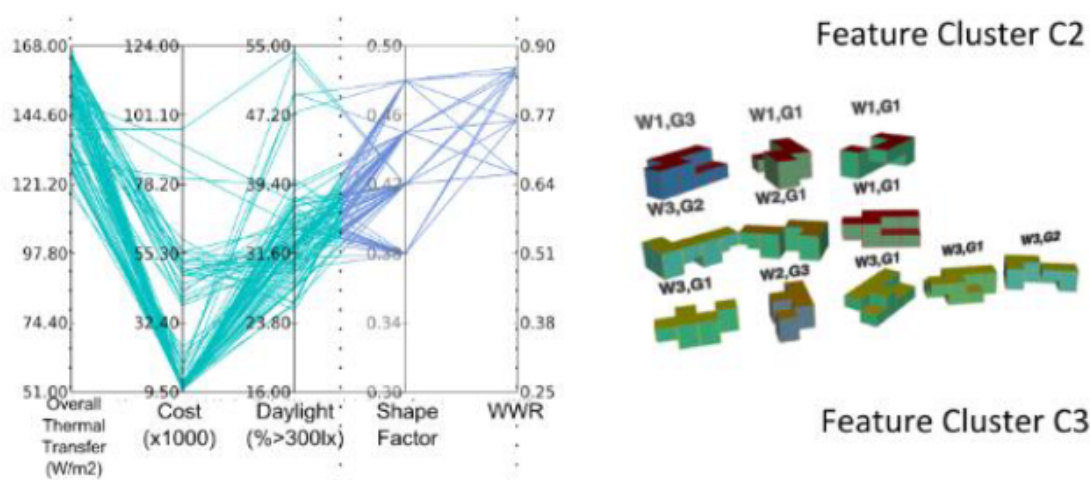
Nous avons évoqué le concept de Fitness landscape relatif à l'optimisation, comme étant un paysage abstrait permettant de représenter, à la fois, les paramètres et leur performance pour un objectif donné.

Lorsque nous cherchons à représenter uniquement les paramètres, nous parlons alors de Design space. Dans le fitness landscape, l'axe des Z est celui qui permet de représenter la performance pour un objectif donné des paramètres, mais qu'en est-il de la représentation de la performance de plusieurs objectifs et de plusieurs paramètres, soit un modèle multi-dimensionnel, dans des espaces/surfaces 2D ou 3D ? Il s'agit d'explorer la représentation multi-variée du fitness landscape.



**Fig 6.3** – Interface du solveur Octopus : A gauche l'espace de solutions et à droite à 3D et "les coordonnées parallèles", dans le même solveur, comme un moyen d'étendre les autres dimensions avec une limite : l'illisibilité de la représentation

Ce type de représentation existe déjà et une méthode particulière est également implémentée, à la fois, dans le solveur utilisé dans ce mémoire (Octopus) ainsi que l'analogue de Grasshopper, à savoir l'outil de Dynamo affilié à Autodesk Revit. La méthode est celle « des coordonnées parallèles », où chaque dimension du problème représente un axe. Chen et al. utilisent cette méthode ainsi que le « K-mean clustering » pour identifier des relations entre 5000 variantes d'un modèle et leur performance (objectif rempli) mais leur méthode reste partiellement réussie.



**Fig 6.4** – représentation du design space par la méthode des coordonnées parallèles et usage du « K-means clustering » pour identifier les relations entre paramètres et objectif - travaux de chen et al (2015)

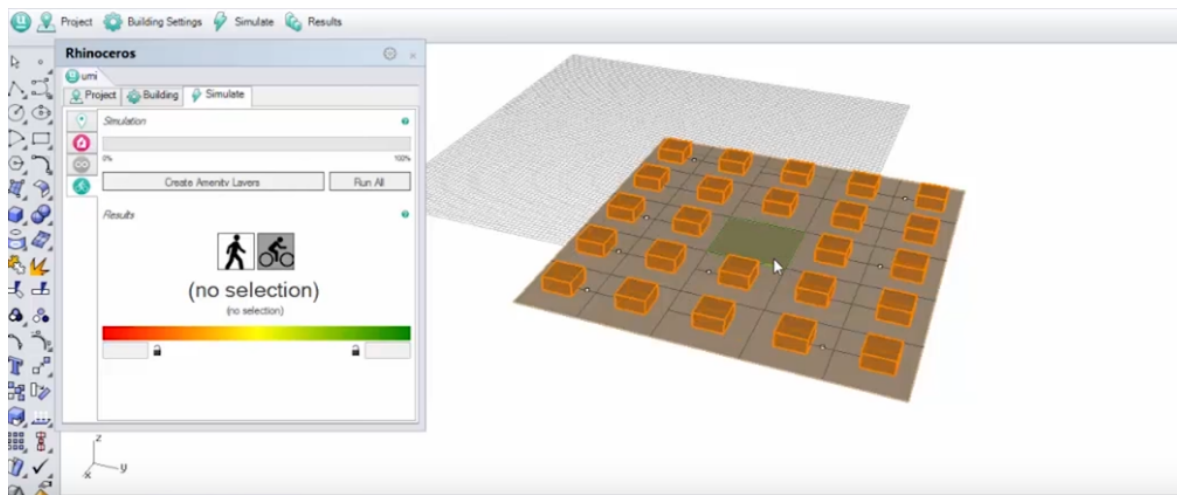
D'autres méthodes similaires telles que « star coordinates » permettent également de visualiser les paramètres des variantes. Cependant, le problème se situe toujours dans le cas de la visualisation de la performance (fitness landscape) de ces paramètres pour le multi-objectif (n-dimensions ou n est supérieur à 3) et qui **soit assez lisible** ou encore de la possibilité d'explorer par soi-même des hybrides ou des variantes du design non données par le processus de l'optimisation. Enfin, un autre atout serait l'exploration des résultats données par deux ou plusieurs algorithmes différents, dans des générations différentes.

Retrouver des hybrides, non explorés par les algorithmes d'optimisation, pourrait se faire éventuellement en utilisant du calcul basé sur des modèles de substitution (surrogate models) [Wortmann, 2018]. Quant à la représentation d'un fitness landscape à n-dimensions, qui soit affilié à l'outil paramétrique de l'architecte et assez lisible, cela constituerait une voie d'exploration pour d'éventuels sujets futurs, même si cela, serait très coûteux en terme computationnel.

N'ayant pas exploré ce mode de représentation dans ce travail, nous pensons, néanmoins, qu'il serait plus judicieux pour l'architecte de visualiser le fitness landscape à n-dimensions à travers des «cartes», qui permettent non seulement de mettre en relation les différents paramètres mais aussi les divers objectifs qu'ils concordent à satisfaire, et ce, afin que le concepteur de l'espace puisse développer une meilleure compréhension vis-à-vis du problème formulé et des réponses obtenues et donc mieux orienter son choix de la solution optimale.

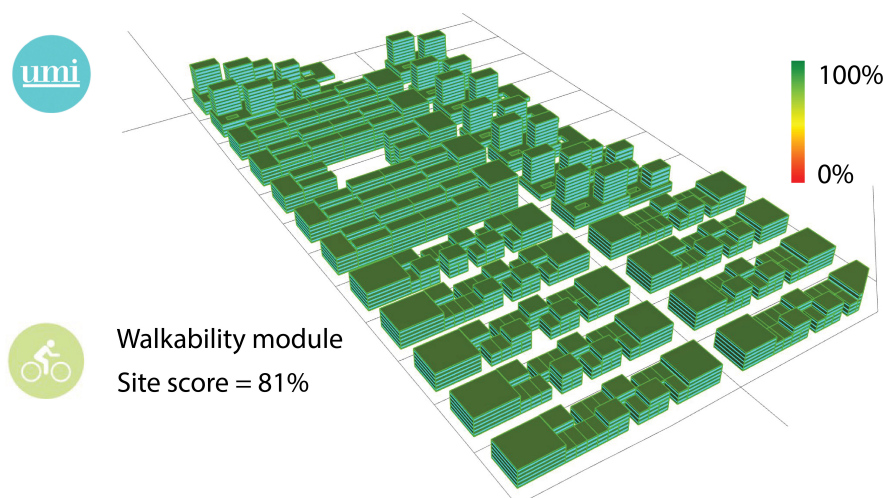
### 6.2.2 | Une autre méthode de représenter la marchabilité pour une nouvelle formule de calcul du score :

Le comportement de la mobilité est encore peu abordée dans l'outil paramétrique et ce, même s'il existe d'autres tentatives de mesure de la marchabilité, comme l'U.M. I :l'Urban Modeling Interface, développé par le MIT Sustainable Design Lab et qui lui aussi s'appuie sur le calcul de la marchabilité selon la méthodologie de walkscore. Contrairement à Urbano, l'UMI est acheminé à Rhinocéros et nécessite, par conséquent, une étape supplémentaire pour « paramétriser » le modèle sur grasshopper.



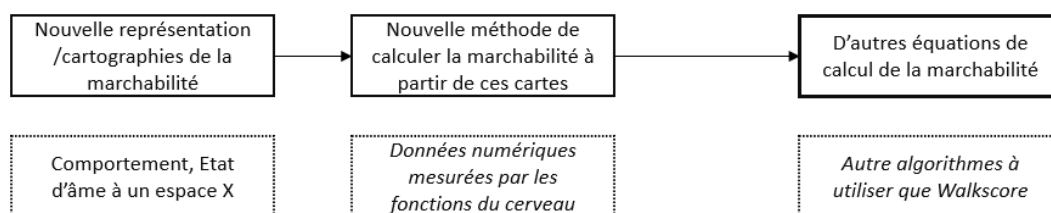
**Fig 6.5** – L'UMI intégrée à Rhino : une autre interface de mesure de la marchabilité dans les outils conceptuels de l'architecte

Néanmoins, tous ces outils construisent leur fonctionnement sur l'origine de Walkscore qui reste relativement critiquable comme algorithme de calcul, tandis que la modélisation d'un comportement telle que la marchabilité et sa mesure nécessiterait peut-être de se pencher sur d'autres disciplines qui étudient la relation entre l'humain et l'environnement bâti urbain.



**Fig 6.6** – Exemple d’application, sur un quartier fictif, du module de calcul de marchabilité, dans Rhino, par l’UMI

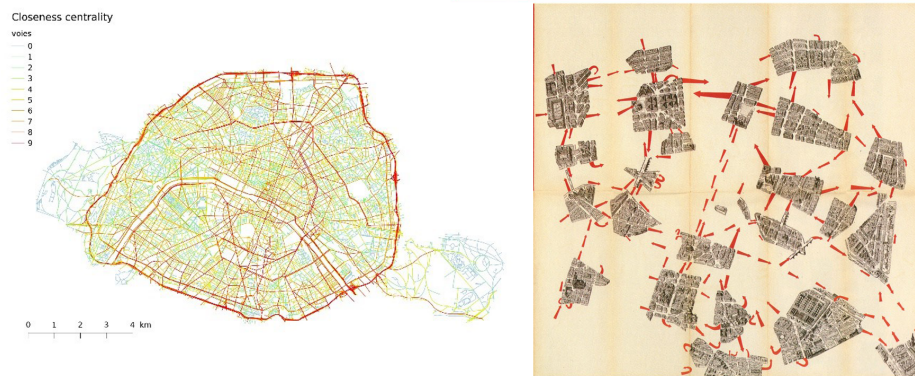
Si la marchabilité d’un espace pouvait être représenté autrement que par le biais d’un réseau viaire et de points (en l’occurrence les équipements et le nombre d’intersection qui vont constituer l’équation finale de calcul), alors l’équation de calcul, elle-même, pourrait être changée et réadaptée en fonction de ce nouveau mode de représentation, qui essaierait d’enrichir la vision de la ville purement fonctionnaliste et donc d’un calcul « absolu ». Cela pourrait permettre de compléter les variables objectives par les variables subjectives et de voir s’il y a résultats numériques assez proches ou concordants pour valider l’un et l’autre.



Nous avons notamment puisé cette alternative dans les cartes psychogéographiques de Guy Debord et de sa théorie de la dérive urbaine<sup>1</sup>. Cette représentation d’un parcours urbain illustre les liens entre la fonction de la ville (par exemple les quartiers) et l’état d’âme d’une

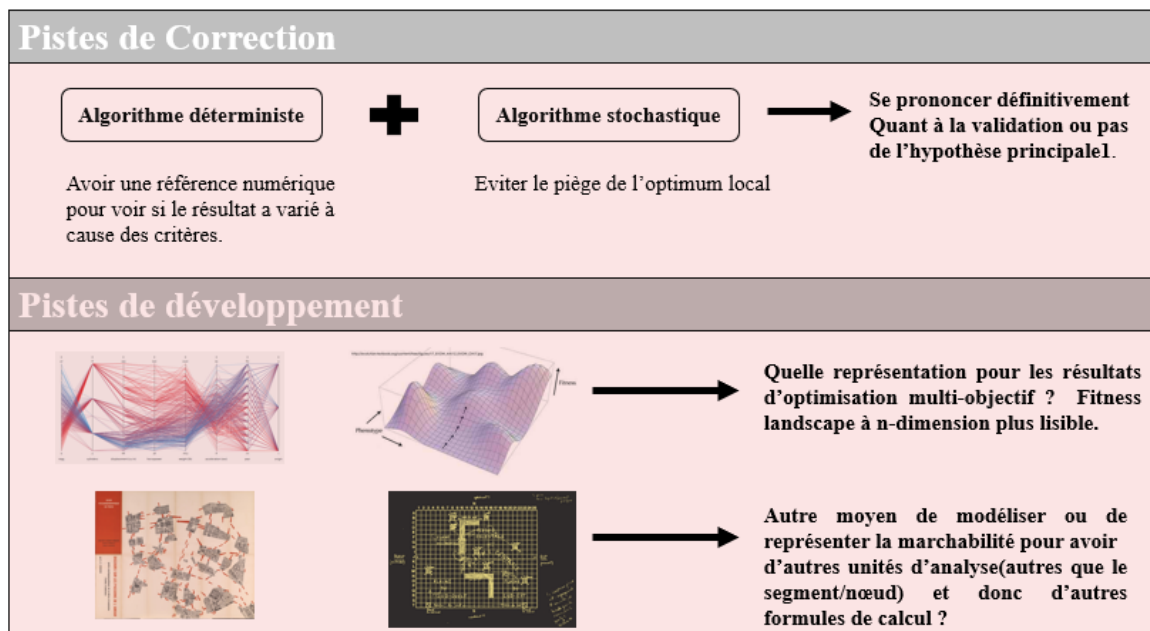
1. Découlant du mouvement situationniste dans l’urbain, la théorie et les cartes évoqués relatent une autre narration du trajet urbain, sous de nouveaux aspects

personne traversant ses divers fragments, rendant compte de plusieurs unités d'ambiance. Si l'état d'âme est difficilement mesurable, c'est actuellement l'objet de recherche dans la discipline de la neuroscience. Appliquée à l'architecture ou l'urbanisme, le comportement de marchabilité pourrait être expliqué, par d'autres variables ou choix, que ceux de l'équipement ou de l'urbanisme de voies. Il ne s'agit pas de mettre en compétition deux méthodes mais plutôt de créer une complémentarité pour avoir d'autres résultats.



**Fig 6.7** – La représentation space syntax à gauche et la représentation des cartes psychogéographiques à droite : Varier la représentation pour de nouveaux métriques ?

## 6.3 | Résumé et conclusion partielle :





## Conclusion générale

Dans ce travail de mémoire, nous avons traité de la question de l'optimisation, sur le plan global, et de la marchabilité, comme cas spécifique. Nous nous sommes intéressés à la marchabilité comme cas d'application, car elle nous permettait justement d'expérimenter la modélisation paramétrique, non d'un bâtiment mais d'un comportement. Ajouté à cela, l'attribution d'une équation relative ou mesure à la mobilité urbaine est intéressant à explorer, notamment que l'équation met en jeu des critères de l'environnement bâti. Conçue, parfois, comme un aspect non interprétable en terme numérique, la translation depuis les théories aux algorithmes puis finalement à l'outil paramétrique nous a permis d'approcher la mesure de la marchabilité.

Pour concrétiser cela dans l'outil paramétrique, nous avons proposé de combiner l'usage d'un module de calcul (Urbano), accouplé aux algorithmes bio-inspirés, afin de déconstruire l'équation, de tester de nouveaux critères et de vérifier finalement leur apport dans la mesure de la marchabilité. L'un des critères omis dans le calcul, actuellement, est la topographie. Par l'usage d'outils extérieurs, relatif au domaine du GIS, nous avons proposé une méthode d'introduire cette dimension, dans le module du calcul afin de vérifier ce qu'elle impliquait, comme modification, sur le score.

L'optimisation, quant à elle, relève d'un champ d'étude plus élargi et demeure toujours sujette à développement. Dans ce mémoire, la marchabilité nourrit également, à son tour, la question de l'optimisation car elle permet de soulever des questions quant au rapport « résultat numérique-configuration graphique » qui ressort du processus d'optimisation.

En effet, l'optimum ne correspondait pas toujours à la configuration spatiale la plus judicieuse. Cela tend à accentuer l'importance du recours à l'optimisation dans la phase d'exploration

du design, puisque les résultats performants ne seraient pas toujours les plus intuitifs.

L'usage de 3 algorithmes évolutionnaires d'optimisation, l'algorithme génétique, le P.S.O et l'algorithme HypE, nous a également poussé à traiter en prospective, la question du choix de l'algorithme. En effet, alors que paradoxalement, ces algorithmes deviennent de plus en plus ancrés dans les outils des architectes, la compréhension et le choix de tel ou tel programme d'optimisation reste relativement peu traitée.

L'automatisation du processus, dans la plate-forme paramétrique, rompt paradoxalement le lien que nous souhaitions créer[Dissaux, 2018, Wortmann,2020] puisque la compréhension et le choix judicieux d'un algorithme donné pour un problème ou des problèmes correspondants pourrait servir à améliorer le processus, qui ne se limiterait plus à la phase de recherche d'un optimum dans le fitness landscape, mais également à toute la réflexion construite autour de l'optimisation en elle-même. Cette dernière, au-delà de ce qu'elle impliquerait comme une "technique" générative, est également une méta-narrative sur les aspects du projet lui-même ainsi que son contexte[Kolarevic, 2015]. Dans cette continuité et continuum, l'algorithme, lui-même, ferait partie de ce "récit" et donc la réflexion autour de son choix aussi.

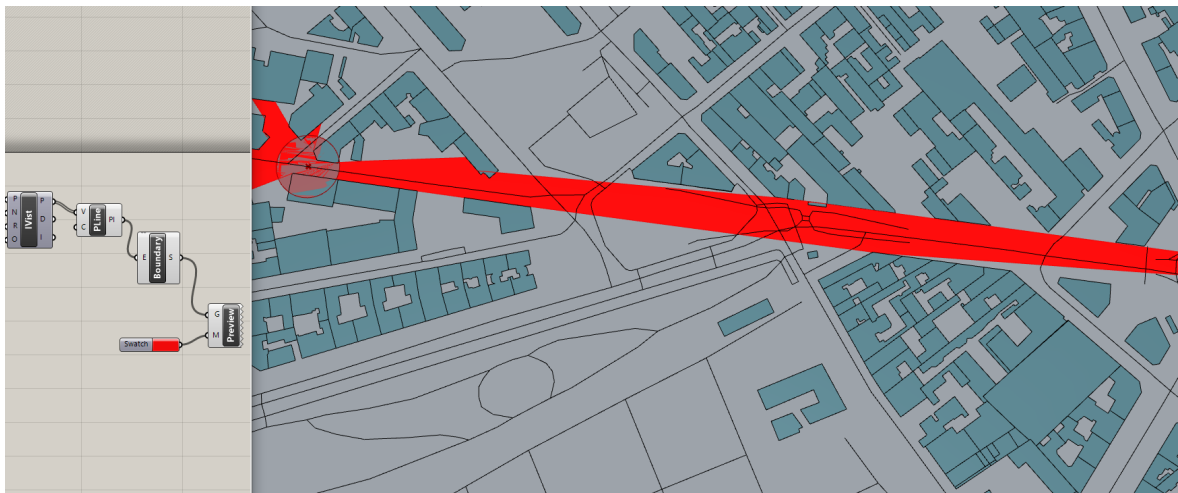
## Configuration et simulation

Configuration de la machine sur laquelle s'effectuait la simulation	
Processeur :	Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz 2.60 GHz
Mémoire installée (RAM) :	64,0 Go
Type du système :	Système d'exploitation 64 bits, processeur x64
Versions des outils principaux utilisés	
<ul style="list-style-type: none"><li>• Rhino 3D - version 6 avec Galapagos inclus.</li><li>• Silvereye 1.1.0 : <a href="https://www.food4rhino.com/app/silvereye-pso-based-solver">https://www.food4rhino.com/app/silvereye-pso-based-solver</a></li><li>• Octopus 0.4(instable) : <a href="https://www.food4rhino.com/app/octopus">https://www.food4rhino.com/app/octopus</a></li><li>• Urbano 1.0 puis 1.1(en dev.): <a href="https://www.food4rhino.com/app/urbano">https://www.food4rhino.com/app/urbano</a></li></ul>	
Temps de calcul du score de marchabilité	
Par critère, le temps de calcul du score avec la version d'Urbano 1.1 est de 2h à 4h approximativement (le temps maximal, approximativement 4h, est observé avec l'intégration de la topographie dans le composant shp puis dans le modèle Urbano)	

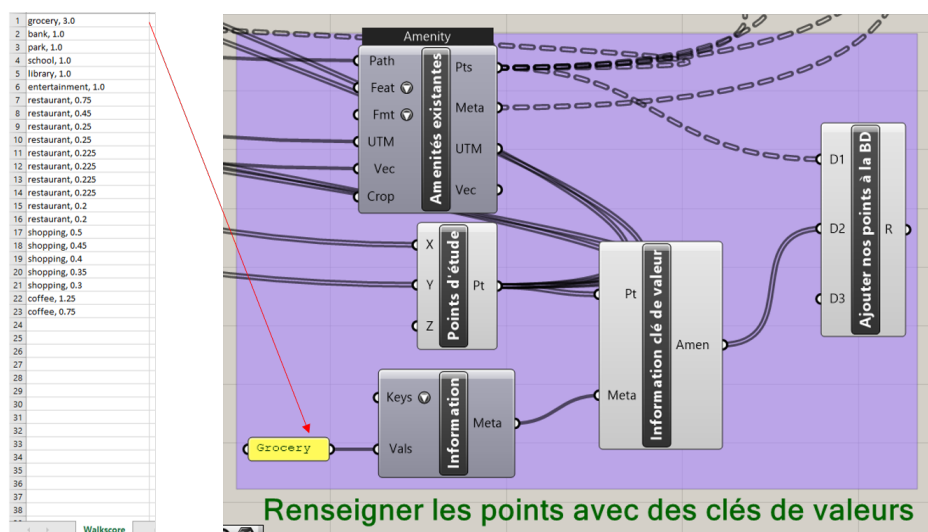


## Données Brutes

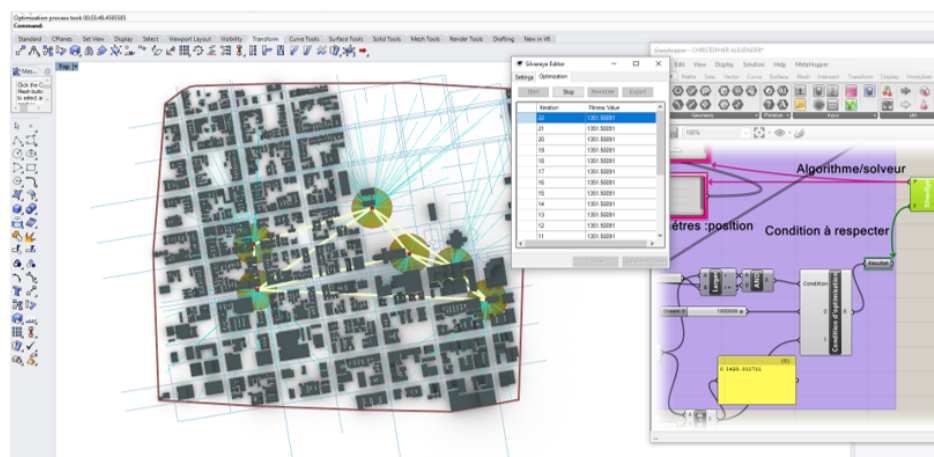
En complément à l'analyse de la connectivité réalisée sur DepthMapX, réalisée dans la section [2.2.2.4] l'usage de l'Isovist sur Grasshopper permet de déceler, à nouveau, les limites de la S.S 2D pour le métrique de l'aire visible. L'Isovist 3D n'est pas incorporé dans les outils natifs développés par l'UCL mais existe dans des outils externes.



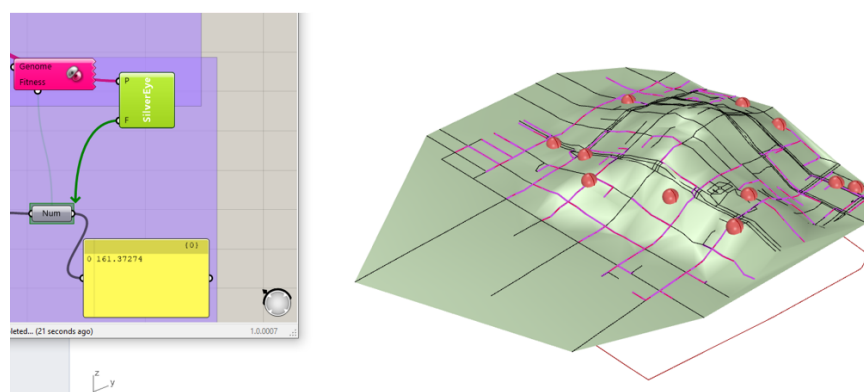
**Fig B.1** – L'analyse 2D de la S.S sur Grasshopper pour l'Isovist : pas de prise en compte de la limite visuelle sur le terrain



**Fig B.2** – Affectation de classes(types et C.I) aux points étudiés dans l’expérience depuis le fichier excel.



**Fig B.3** – Exemple de recherche par le PSO pour la configuration du schéma d’Alexander



**Fig B.4** – Exemple résultat pour le PSO avec l’intégration de la topographie

## Liste des figures

1.1	schéma illustrant les hypothèses 1 . . . . .	6
1.2	schéma illustrant les hypothèses 2 . . . . .	7
2.1	Collage,espace marchable . . . . .	11
2.2	abstraction,réseau avec villes satellites . . . . .	12
2.3	Collage,marchabilité en lien étroit avec le T.O.D . . . . .	12
2.4	plan pré et post-optimisation informatique du travail cité . . . . .	13
2.5	Compartif entre l'approche conventionnelle d'analyse et l'approche syntaxique . .	14
2.6	Passage de la représentation conventionnelle à la représentation S.S de gauche à droite (a),(b),(c) et(d) tels qu'expliqué ci-dessous . . . . .	15
2.7	Certains métriques de la Syntaxe Spatiale. . . . .	15
2.8	Les types de distances dont les deux premières calculables par la S.S et la troisième, une distance autre . . . . .	16
2.9	Schéma de la théorie évoquée-relation du mouvement par rapport à la forme et configuration urbaine . . . . .	17
2.10	Schéma de l'expérience de pensée proposée par Bill Hillier dans l'énoncé de la théorie étudiée . . . . .	18
2.11	Simulation sur l'outil paramétrique de la prédictibilité du mouvement par rapport à la distribution fonctionnelle dans le temps . . . . .	19
2.12	schéma de la carte Axiale de la S.S et manque d'informations par rapport à la troisième dimension . . . . .	20
2.13	schéma, paradoxe de la centralité . . . . .	21
2.14	Paradoxe de la centralité et mesure de l'intégration globale par Ratti : D'abord dans deux système urbain séparés, puis mis en communication - sur l'outil Axman PPC . . . . .	21

2.15	Rue Gasnier Guy avec l'analyse de la connectivité sur DepthMapX : La rue est désignée comme bien connectée mais cela ne suffit pas pour impliquer sa marchabilité	22
2.16	schéma complémentarité des méthodes : approche SIG et approche syntaxique, quel outil ? . . . . .	23
2.17	schéma d'un exemple de la notion des coefficients d'importance dans les variables de marchabilité . . . . .	27
2.18	schéma du rapport : nature de la donnée-échelle spatiale d'évaluation . . . . .	28
3.1	Walkscore : de la théorie à l'algorithme . . . . .	33
3.2	Limites du calcul et application sur l'Avenue de Flandre, à Paris. . . . .	35
3.3	Les Heat Maps de Walkscore : Un autre moyen de représenter le score . . . . .	35
3.4	Variables reprises dans Walkscore . . . . .	36
3.5	Le travail de Johnson : Rétro-ingénierie et essai de prédiction du comportement de l'algorithme walkscore pour tenter de comprendre l'importance des variables qui le constitue . . . . .	37
3.6	Partie de l'algorithme qui classe les équipements avec une "note" . . . . .	38
3.7	Courbe de la distance decay function pour l'obtention de valeurs intermédiaires des C.I . . . . .	40
3.8	schéma de la construction de la variable 1 selon la méthode walkscore . . . . .	41
3.9	exemple simple d'application de l'algorithme . . . . .	42
3.10	Erreurs de calcul dues à la déficience des critères pris en compte : à gauche : l'autoroute de St Bistol et à droite : l'algorithme présentant l'autoroute comme marchable. . . . .	43
4.1	Schéma de la démarche suivie dans ce chapitre . . . . .	45
4.2	Exemple de décomposition paramétrique d'un même modèle et génération de variantes . . . . .	46
4.3	En haut : Interface de Grasshopper accouplée à celle de Rhino et en bas : illustration d'un ensemble de composants(définition) . . . . .	47
4.4	Travaux de Zaha Hadid Architects, à l'échelle urbaine, avec la démarche paramétrique : (a) : One North Masterplan (b) Kartal Pendik Masterplan et (c) Evolution du mod.urbain de Thames Gateway Masterplan . . . . .	48
4.5	Extraction depuis OSM à travers le composant <b>Download OSM</b> d'Urbano qui permet d'intégrer des données GIS dans l'outil paramétrique . . . . .	50
4.6	La notion d'ADP qui permet la construction du modèle Urbano et son développement à travers la recherche . . . . .	50
4.7	Extraction des données géographiques par les 3 composants principaux . . . . .	51

---

4.8	Composants participant à la construction du modèle de calcul Urbano . . . . .	51
4.9	Intégration de géométrie personnalisée, avec classes, dans le modèle de calcul . .	52
4.10	Affectation des coefficients d'importances aux classes d'équipements depuis un fichier Excel jusqu'au composant ADPer . . . . .	53
4.11	Les deux mesures de calcul la marchabilité sous Urbano : A gauche Stree- tHits(local) et à droite Walkscore(global) . . . . .	54
4.12	Recherche de l'optimum et représentation : A gauche : Cas d'un paramètre et à droite : Cas de deux paramètres dans un paysage de solution . . . . .	55
4.13	Recherche d'une morphologie urbaine face à des données environnementales ex- traites de l'étude de Wang et al,2019 . . . . .	56
4.14	Recherche de la disposition optimale de bâtiments dans un îlot urbain par rapport à des critères tels que l'apport solaire et la vue -Travaux de spacemaker . . . . .	57
4.15	Classification des algorithmes selon leur mode opératoire pour la recherche de l'optimum : dans le cadre colorés, les algorithmes qui seront utilisés dans ce mémoire	58
4.16	Mode opératoire de l'algorithme déterministe évoqué ci-dessus :A gauche : "les sauts" pour trouver l'optimum et à droite : le problème de l'optimum local . . . .	59
4.17	Emergence des algorithmes bio-inspirés dans l'histoire : Transposition depuis la biologie dans les sciences informatiques . . . . .	60
4.18	Cycle de l'optimisation par un AG . . . . .	61
4.19	Les diverses topologies du paysage de solutions en fonction de la formulation du problèmes . . . . .	62
4.20	Essaim d'oiseaux : Un système biologique coordonné comme source d'inspiration du P.S.O . . . . .	62
4.21	Comparaison entre les deux façons d'explorer le paysage des solutions par l'AG(dessus) et le PSO(bas) . . . . .	63
4.22	Mode opératoire du PSO : communication des agents entre eux pour déterminer la nouvelle position optimale . . . . .	64
4.23	schéma d'une optimisation multi-objectif avec le cas de 2 objectifs à satisfaire : Ces deux objectifs sont matérialisée par les fonctions fitness f1 et f2 et les solutions optimales en front pareto . . . . .	65
4.24	Cas d'une optimisation multi-objectif avec 3 objectifs = limite de la représentation ?	66
4.25	Classification des algorithmes utilisés dans ce mémoire avec leur solveurs respec- tifs, selon la méthode : Mono ou Multi-objectif . . . . .	67
4.26	Définition avec le solveur Galapagos à la fin et la notion des gene pools . . . . .	68
4.27	Exemple d'une définition avec le solveur Galapagos à la fin et la notion des gene pools . . . . .	69

4.28	Le composant du solveur octopus dans le canvas de grasshopper avec plusieurs objectifs(en vert) et les paramètres variables(en rose) . . . . .	70
4.29	Interface d'Octopus : Représentation de l'espace de recherche pour le cas de 3 objectifs . . . . .	70
4.30	Algorithme sur grasshopper du travail pré-cité : recherche de la position des équipements . . . . .	71
4.31	Introudction de la topographie comme un parametre dans la démarche paramétrique urbaine : usage uniquement pour la morphologie urbaine . . . . .	72
4.32	Résultat du travail de J.Chichoka en minimisant la distance des équipements vis-à-vis de certains points seulement avec la méthode walkscore . . . . .	72
5.1	schéma de l'explication de la partie à aborder . . . . .	75
5.2	Schéma de la base comparative : méthode actuelle et expériences . . . . .	76
5.3	Projection de la trame fictive sur le contexte extrait . . . . .	77
5.4	Moyenner la position de chaque habitant en trouvant le milieu de chaque carreau . . . . .	78
5.5	Disposer aléatoirement des équipements et lier chacun avec un certain nombre d'habitants les plus proches . . . . .	78
5.6	Délimitation de l'aire de recherche et condition posée : omission de tout point à l'extérieur de cette aire . . . . .	79
5.7	Délimitation de l'aire de recherche finale et pose de la condition : un point hors de l'aire de recherche n'est pas compté . . . . .	80
5.8	Dans Grasshopper, la condition est posée par le composant <i>pick'n'choose</i> qui permet de choisir d'afficher un résultat ou une erreur . . . . .	80
5.9	La position des points(x,y) est déterminée par le <i>gene pools</i> qui constitue l'entrée <i>genome</i> du solveur Galapagos : il s'agit donc des variables . . . . .	81
5.10	Détection paramétrique de l'entrée(sans recours au tag) par le composant <i>curve closest point</i> dans Grasshopper et l'influence de l'entrée sur le changement du parcours marchable . . . . .	82
5.11	Détection paramétrique de l'entrée sur Grasshopper . . . . .	82
5.12	Extrait du livre de A language Pattern de C.Alexander montrant les schémas urbains des bassins d'activités avec une distance "idéale" dans le réseau de 1.8Km . . . . .	84
5.13	Interprétation du schéma d'Alexander par la figure de Delaunay dans l'outil paramétrique . . . . .	85
5.14	Partie de la définition montrant la condition posée avec le critère des bassins d'activités d'Alexander . . . . .	86
5.15	Exemple Extraction et important des valeurs d'élévation d'un DEM de la zone voisine à la rue Ménilmontant à Paris . . . . .	87

---

5.16	règle de la pente . . . . .	88
5.17	Partie de l'algorithme utilisant la règle du déplacement du SMA . . . . .	88
5.18	Partie de la règle de la pente à suivre par le SMA pour l'extraction des chemins à une pente de 20% et exemples des résultats SMA avec deux pentes . . . . .	89
5.19	Transposition des parcours "naturels" du SMA vers le réseau viaire à travers le plugin ShortestWalk(A*) . . . . .	89
5.20	Le contexte de l'étude avec la rue principale et le score de marchabilité de l'aire de recherche dans l'expérience . . . . .	91
5.21	Le contexte extrait avec la topographie : Usage du fichier shp disponible en ligne	92
5.22	Résultat optimisation RE1 . . . . .	93
5.23	Résultat optimisation RE2 : Modification du nombre de points et nouvelles simulations . . . . .	93
5.24	Évolution de la fitness(distance minimisée) pour les deux algorithmes-cas RE . .	94
5.25	Résultat d'optimisation A* . . . . .	95
5.26	Évolution de la valeur fitness pour les deux algorithmes - cas A* . . . . .	95
5.27	Comparaison A* et RE : Même nombre de points/Algorithme mais différence du critère . . . . .	96
5.28	Résultat optimisation 2D équipements-équipements basé sur le schéma d'Alexander	98
5.29	Résultat optimisation A* avec l'intégration de la topographie : Au préalable avec l'expérience SMA et extraction du fichier shp . . . . .	99
5.30	Définition algorithmique du scénario : construction des variables et objectifs et introduction dans le solveur d'Octopus . . . . .	103
5.31	Résultat optimisation multi-objectif avec la condition topographie également . .	104
5.32	Interface du solveur Octopus en cours de recherche : les box rouges constituent le début d'un front pareto, les deux axes représentent chacun les deux objectifs avec les valeurs max et min . . . . .	105
5.33	Des instances de solutions capturées dans 2 types de simulations différentes : Le couple 1 : un résultat sans la condition et le couple 2 et 3 pendant une seconde simulation avec condition de la topographie . . . . .	106
5.34	Le front pareto dans l'espace de recherche de la simulation sans condition dans laquelle nous avons choisi une instance, parmi les solutions du front (couple 1) .	106
5.35	En relation avec les paysages abstraits présentés par Rutten, la formulation du problème avec des contraintes ou des objectifs contradictoires pourrait induire des paysages sans "optimum idéal . . . . .	107
5.36	Évolution des couples de solution dans une génération dans Octopus . . . . .	107
5.37	Mise en place des 3 objectifs dans le canvas de Grasshopper avec Octopus . . . .	108

5.38	Introduction d'un troisième objectif : Le schéma d'Alexander et exploration sur un troisième axe dans le paysage de recherche d'Octopus . . . . .	109
5.39	En haut : la configuration obtenue et en bas : Instance de solution qui a satisfait l'objectif d'Alexander et les valeurs correspondantes pour A* et la RE dans Grasshopper et Octopus . . . . .	110
6.1	Logo GOAT . . . . .	113
6.2	Interface du solveur GOAT avec l'algorithme déterministe de prog.linéaire COBYLA	114
2.55	emAuteur	
6.3	Interface du solveur Octopus : A gauche l'espace de solutions et à droite à 3D et "les coordonnées parallèles", dans le même solveur, comme un moyen d'étendre les autres dimensions avec une limite : l'illisibilité de la représentation . . . . .	115
6.4	représentation du design space par la méthode des coordonnées parallèles et usage du «K-means clustering » pour identifier les relations entre paramètres et objectif -travaux de chen et al(2015) . . . . .	116
6.5	L'UMI intégrée à Rhino : une autre interface de mesure de la marchabilité dans les outils conceptuels de l'architecte . . . . .	117
6.6	Exemple d'application, sur un quartier fictif, du module de calcul de marchabilité, dans Rhino, par l'UMI . . . . .	118
6.7	La représentation space syntax à gauche et la représentation des cartes psychogéographiques à droite : Varier la représentation pour de nouvelles métriques? . . . . .	119
B.1	L'analyse 2D de la S.S sur Grasshopper pour l'Isovist : pas de prise en compte de la limite visuelle sur le terrain . . . . .	125
B.2	Affectation de classes(types et C.I) aux points étudiés dans l'expérience depuis le fichier excel. . . . .	126
B.3	Exemple de recherche par le PSO pour la configuration du schéma d'Alexander . . . . .	126
B.4	Exemple résultat pour le PSO avec l'intégration de la topographie . . . . .	126

---

## Liste des tableaux

2.1	Tableau des métriques de la S.S illustrés ci dessus et leur définition, telle qu'établie par le labo de l'UCL-spacesyntax . . . . .	16
2.2	Tableau de classification des types de variables et la méthode afférente . . . . .	25
2.3	Tableau de certains indicateurs avec les unités d'analyse et méthodes afférentes .	26
2.4	Tableau de comparaison des échelles et des méthodes d'évaluation de trois chercheurs dont L.D Frank . . . . .	29
2.5	Tableau de transposition des concepts de la SS et des 3-D de la marchabilité développés par l'ingénieur Kara Kockelman et repris par Kooshari pour le modèle de substitution. . . . .	30
3.1	Tableau des intervalles de score de l'algorithme et de la description correspondante.	34
3.2	Pourcentage des sanctions PFM(variable 2) attriubés à la variable 1 selon la forme urbaine . . . . .	41
4.1	Comparaison de la méthode actuelle avec celle empruntée par ce mémoire selon les différentes étapes du travail . . . . .	49
4.2	Tableau de la transposition de terminologie entre la biologie et l'optimisation informatique réalisée par les A.G . . . . .	61
5.1	Tableau synthétique avec toutes les valeurs obtenues par critères et par algorithme d'optimisation . . . . .	112



---

## Bibliographie

### Sur l’outil paramétrique/design algorithmique en général :

#### Ouvrages :

Burry, M. (2013). *Scripting Cultures : Architectural Design and Programming*. John Wiley Sons.

Frazer, J. (1995). *An Evolutionary Architecture*. Architectural Association Publications.

Menges, A., Ahlquist, S. (2011). *Computational Design Thinking : Computation Design Thinking*. John Wiley Sons.

Tedeschi, A. (2014). *AAD Algorithms-Aided Design* (14e édition). Le Penseur.

Terzidis, K. (2006). *Algorithmic Architecture*. Routledge.

#### Cours universitaires :

Guéna, F. (2019) Cours des techniques de représentation TR07 du M1-architecture : Initiation à la modélisation paramétrique, à l’ENSA Paris La Villette.

Guéna, F. (2020) Cours transversal intra-domaine CTID08 du M1-architecture : Systèmes numériques de la conception digitale, à l’ENSA Paris La Villette.

### Sur les algorithmes d’optimisation spécifiquement :

**Ouvrages :**

Bozorg-Haddad, O., Solgi, M., Loáiciga, H. A. (2017). *Meta-heuristic and Evolutionary Algorithms for Engineering Optimization* (1st edition). Wiley.

Buontempo, F. (2019). *Genetic Algorithms and Machine Learning for Programmers : Create AI Models and Evolve Solutions*. The Pragmatic Programmers.

Deb, K. (2009). *Multi-Objective Optimization Using Evolutionary Algorithms* (première édition). Wiley. Eberhart, R. C., Shi, Y., Kennedy, J. (2001). *Swarm Intelligence*. Morgan Kaufmann Publishers In.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning* (1er édition). Addison Wesley.

Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems : An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Bradford Books.

Luke, S. (2013). *Essentials of Metaheuristics*.

Poli, R., Langdon, W. B., McPhee, N. F. (2008). *A Field Guide to Genetic Programming*.

**Articles :**

Ashour, Y., Kolarevic, B. (2015). *Optimizing creatively in Multi-Objective Optimization*. Simulation Series, 47, 128 135.

Bielik, M., Koenig, R., Fuchkina, E., Schneider, S., Abdulmawla, A. (2019, juillet 11). *Evolving configurational properties : Simulating multiplier effects between land use and movement patterns*.

Buntrock, D. (2006). *Architecture in the Digital Age : Design and Manufacturing and Performative Architecture : Beyond Instrumentality* - Edited by Branko Kolarevic, Branko Kolarevic and Ali Malkawi. Journal of Architectural Education, 60(1), 67 68. [https://doi.org/10.1111/j.1531-314X.2006.00068\\_1.x](https://doi.org/10.1111/j.1531-314X.2006.00068_1.x)

Chaszar, A., Von Buelow, P., Turrin, M. (2016, mai 15). *Multivariate Interactive Visualization*

---

*of Data in Generative Design.*

Chen, K., Janssen, P., Schlueter, A. (2015, septembre 16). *Analysing Populations of Design Variants Using Clustering and Archetypal Analysis.*

Cichocka, J. (2015). *Generative design optimization in urban planning : Walkability-optimized city concept.* Architectus, 119 128. <https://doi.org/10.5277/arc150110>

Cichocka, J., Browne, W., Rodriguez, E. (2015). *Evolutionary optimization processes as design tools : Implementation of a revolutionary swarm approach.* Proceedings of 31th International PLEA Conference Architecture in (R) Evolution.

Cichocka, J. M. (2016). *Particle Swarm Optimization for Architectural Design.* Code of Space, Vienna.

Cichocka, Judyta M., Migalska, A., Browne, W. N., Rodriguez, E. (2017). SILVEREYE—*The implementation of particle swarm optimization algorithm in a design optimization tool.* International Conference on Computer-Aided Architectural Design Futures, 151 169.

Cichocka, Judyta Maria, Browne, W. N., Rodriguez, E. (2017). Optimization in the architectural practice—An International Survey.

Indraprastha, A. (2019). *Informed Walkable City Model*/paper/Informed-Walkable-City-Model-Developing-A-Model-for-Indraprastha/78401ef97dc58550ab9e03c9fa99d9a26096f338

JOHNVICTOR, A. C., Durgamahanthi, V., Venkata, R., Jethi, N. (2020). *Critical review of bio-inspired optimization techniques.* Wiley Interdisciplinary Reviews : Computational Statistics. <https://doi.org/10.1002/wics.1528>

Kabošová, L., Katunský, D., Kmet, S. (2020). *Wind-Based Parametric Design in the Changing Climate.* Applied Sciences, 10, 1 18. <https://doi.org/10.3390/app10238603>

Kolarevic, B. (2004). *Back to the Future : Performative Architecture.* International Journal of Architectural Computing vol. 2 - no. 1, 2. <https://doi.org/10.1260/1478077041220205>

Oxman, R. (2008). *Performance-Based Design : Current Practices and Research Issues.* International Journal of Architectural Computing vol. 6 - no. 1, pp. 1-17, 6. <https://doi.org/10.1260/147807708784640090>

Rakha, T., Reinhart, C. (2012). *GENERATIVE URBAN MODELING : A DESIGN WORK FLOW FOR WALKABILITY-OPTIMIZED CITIES.* Undefined. /paper/GENERATIVE-URBAN-MODELING

Razzaghmanesh, D., Aksoy, M. (2020, novembre 16). *The Effect of Parametric Design on Designing Performative Facades*.

Rios, L., Sahinidis, N. (2009). *Derivative-free optimization : A review of algorithms and comparison of software implementations*. *Journal of Global Optimization*, 56.

<https://doi.org/10.1007/s10898-012-9951-y> Rutten, D. (2014). Navigating multi-dimensional landscape in a foggy weather as an analogy for generic problem solving. Proceedings of the 16th international conference on geometry and graphics, Innsbruck, p. 14. Innsbruck : ISGG

Rutten, D. (2013). *Galapagos : On the Logic and Limitations of Generic Solvers*. *Architectural Design*, 83. <https://doi.org/10.1002/ad.1568>

Wetter, M., Wright, J. (2004). *A comparison of deterministic and probabilistic optimization algorithms for nonsmooth simulation-based optimization*. *Building and Environment*, 39, 989 999. <https://doi.org/10.1016/j.buildenv.2004.01.022>

Wortmann, T. (2017). *Surveying design spaces with performance maps : A multivariate visualization method for parametric design and architectural design optimization*. *International Journal of Architectural Computing*, 15(1), 38 53.

Wortmann, T., Costa, A., Nannicini, G., Schroepfer, T. (2015). *Advantages of surrogate models for architectural design optimization*. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing : AI EDAM*, 29(4), 471.

Wortmann, T., Fischer, T. (2020, août 1). *Does architectural design optimization require multiple objectives ? A critical analysis*.

Wortmann, T., Nannicini, G. (2016). *Black-box optimization for architectural design : An overview and quantitative comparison of metaheuristic, direct search, and model-based optimization methods*. *Living Systems and Micro-Utopias*, 177 186.

Wortmann, T., Natanian, J. (2020). *Multi-Objective Optimization for Zero-Energy Urban Design in China : A Benchmark*. *Proc. SimAUD2020*, 203 210.

Wortmann, T., Schroepfer, T. (2019). *From Optimization to Performance-Informed Design*. *Proceedings of the Symposium on Simulation for Architecture Urban Design*.

Zani, A., Andaloro, M., Deblasio, L., Ruttico, P., Mainini, A. (2017). *Computational Design and Parametric Optimization Approach with Genetic Algorithms of an Innovative Concrete Shading Device System*. *Procedia Engineering*, 180, 1473 1483. <https://doi.org/10.1016/j.proeng.2017.04.310>

---

### Videos et Sites :

- Tutoriel Galapagos de ParametricHouse/Rh.Grasshopper :

[https://www.youtube.com/watch?v=g3CPe1bPhEw&ab\\_channel=RhinoGrasshopper](https://www.youtube.com/watch?v=g3CPe1bPhEw&ab_channel=RhinoGrasshopper)

- View Optimization Using Galapagos For Grasshopper(payant)

Les 3 premiers cours sur les algorithmes évolutionnaires et Galapagos.

<https://thinkparametric.com/courses/view-optimization-using-galapagos-for-grasshopper>

- Optimization with Grasshopper(présentation de Koenig sur Galapagos et Octopus)

[https://www.youtube.com/watch?v=IlsJTFVtB\\_Y](https://www.youtube.com/watch?v=IlsJTFVtB_Y)

- Tutoriel : Analyse de la connectivité avec DepthMapX

<https://www.youtube.com/watch?v=QXH-CXWQQGM>

<https://ieatbugsforbreakfast.wordpress.com/author/davidrutton/> (Consulté en Novembre 2019)

<http://web.mit.edu/sustainabledesignlab/projects/umi/index.html> (Consulté en Novembre 2019)

<http://core.thorntontomasetti.com/projects/> (Consulté en Décembre 2019)

<https://www.spacesyntax.online/software-and-manuals/depthmap/> (Consulté en Janvier 2020)

<https://www.spacemakerai.com/> (Consulté en Avril 2020)

### Mémoires :

Dissaux, T. (2018). Mémoire de fin d'études : "*Optimisation dans la conception architecturale*". A l'université de Liège.

## Sur la marchabilité/Influence de l'environnement bâti sur la marchabilité :

### Ouvrages :

Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., Angel, S. (1977). *A Pattern Language : Towns, Buildings, Construction*. Oxford University Press.

Hillier, B. (1996). *Space is the Machine : A Configurational Theory of Architecture*. Cambridge University Press.

### Articles :

Buck, C., Tkaczick, T., Pitsiladis, Y., Bourdehaudhuij, I., Reisch, L., Ahrens, W., Pigeot, I. (2014). *Objective Measures of the Built Environment and Physical Activity in Children : From Walkability to Moveability*. Journal of urban health : bulletin of the New York Academy of Medicine, 92. <https://doi.org/10.1007/s11524-014-9915-2>

Cambra, P., Moura, F., Gonçalves, A. (2017, juillet 3). *On The Correlation of pedestrian flows to urban environment measures : A Space Syntax and Walkability Analysis comparison case*.

Cerin, E., Saelens, B. E., Sallis, J. F., Frank, L. D. (2006). *Neighborhood Environment Walkability Scale : Validity and development of a short form*. Medicine and science in sports and exercise, 38(9), 1682.

Duncan, D., Aldstadt, J., Whalen, J., Melly, S., Gortmaker, S. (2011). *Validation of Walk Score® for Estimating Neighborhood Walkability : An Analysis of Four US Metropolitan Areas*. International journal of environmental research and public health, 8, 4160-4179. <https://doi.org/10.3390/ijerph8114160>

Frank, L. D., Engelke, P. O. (2001). *The built environment and human activity patterns :*

---

*Exploring the impacts of urban form on public health.* Journal of planning literature, 16(2), 202-218.

Frank, L. D., Pivo, G. (1994). *Impacts of mixed use and density on utilization of three modes of travel : Single-occupant vehicle, transit, and walking.* Transportation research record, 1466, 44-52.

Frank, L. D., Sallis, J. F., Saelens, B. E., Leary, L., Cain, K., Conway, T. L., Hess, P. M. (2010). *The development of a walkability index : Application to the Neighborhood Quality of Life Study.* British journal of sports medicine, 44(13), 924-933.

Frank, L. D., Schmid, T. L., Sallis, J. F., Chapman, J., Saelens, B. E. (2005). *Linking objectively measured physical activity with objectively measured urban form : Findings from SMARTRAQ.* American journal of preventive medicine, 28(2), 117-125.

Hasan, M. M. (2020). *Application of Artificial Intelligence and Geographic Information System for Developing Automated Walkability Score.* <https://doi.org/10.13140/RG.2.2.18030.02886>

Hess, P. M., Vernez Moudon, A., Catherine Snyder, M., Stanilov, K. (1999). *Site design and pedestrian travel.* Transportation research record, 1674(1), 9-19.

Hillier, B., Penn, A., Hanson, J., Grajewski, T., Xu, J. (1993). *Natural Movement : Or, Configuration and Attraction in Urban Pedestrian Movement.* Environment and Planning B : Planning and Design, 20, 29-66. <https://doi.org/10.1068/b200029>

Koohsari, J., Owen, N., Cerin, E., Giles-Corti, B., Sugiyama, T. (2016). *Walkability and walking for transport : Characterizing the built environment using Space Syntax.* International Journal of Behavioral Nutrition and Physical Activity, 13. <https://doi.org/10.1186/s12966-016-0448-9>

Koohsari, J., Oka, K., Owen, N., Sugiyama, T. (2019). *Natural Movement : A Space Syn-*

*tax Theory Linking Urban Form and Function with Walking for Transport*. Health Place. <https://doi.org/10.1016/j.healthplace.2019.01.002>

Lee, C., Moudon, A. V. (2006). *The 3Ds+ R : Quantifying land use and urban form correlates of walking*. Transportation Research Part D : Transport and Environment, 11(3), 204 215.

Leslie, E., Coffee, N., Frank, L., Owen, N., Bauman, A., Hugo, G. (2007). *Walkability of local communities : Using geographic information systems to objectively assess relevant environmental attributes*. Health place, 13(1), 111 122.

Maghelal, P., Capp, C. J. (2011). *‘Walkability : A Review of Existing Pedestrian Indices’*. URISA Journal, 23, 5 19.

Moudon, A. V., Hess, P. M., Snyder, M. C., Stanilov, K. (1997). *Effects of site design on pedestrian travel in mixed-use, medium-density environments*. Transportation Research Record, 1578(1), 48 55.

Moudon, A. V., Lee, C., Cheadle, A. D., Garvin, C., Johnson, D., Schmid, T. L., Weathers, R. D., Lin, L. (2006). *Operational definitions of walkable neighborhood : Theoretical and empirical insights*. Journal of physical activity and health, 3(s1), S99 S117.

Pikora, T., Giles-Corti, B., Bull, F., Jamrozik, K., Donovan, R. (2003). *Developing a framework for assesment of environmental determinants of walking and cycling*. Social science medicine (1982), 56, 1693 1703. [https://doi.org/10.1016/S0277-9536\(02\)00163-6](https://doi.org/10.1016/S0277-9536(02)00163-6)

Shoorcheh, M., Varesi, H., Seyed Ahmadiyani, J., Litman, T. (2016). *Urban Growth Structure and Travel Behavior in Tehran City*. Modern Applied Science, 10, 32. <https://doi.org/10.5539/mas.v10n8p32>

*Travel and the Built Environment : A Synthesis*—Reid Ewing, Robert Cervero, 2001. (s. d.). Consulté 30 Novembre 2019, à l’adresse <https://journals.sagepub.com/doi/10.3141/1780-10>

---

**Sites :**

<https://www.walkscore.com/>(Consulté en Novembre 2019)

<https://towardsdatascience.com/engineer-walkscore-4dc1730b976c> (Consulté en Mai 2020)

**Manuel de l'algorithme Walkscore :**

<http://pubs.cedeus.cl/omeka/files/original/b6fa690993d59007784a7a26804d42be.pdf>