



HAL
open science

Réalisation d'un logiciel pour l'optimisation énergétique du bâtiment

Louis Orta

► **To cite this version:**

| Louis Orta. Réalisation d'un logiciel pour l'optimisation énergétique du bâtiment. Sciences de
| l'ingénieur [physics]. 2016. dumas-01841734

HAL Id: dumas-01841734

<https://dumas.ccsd.cnrs.fr/dumas-01841734>

Submitted on 17 Jul 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

TRAVAIL DE FIN D'ETUDE

Organisme d'accueil :

University of Colorado Boulder

Sujet :

Réalisation d'un logiciel pour l'optimisation énergétique du bâtiment

Auteur : Louis ORTA

Promotion 61

VA : Bâtiment

Année 2015-2016

Jury :

Président de Jury : Dr. Mohamed El Mankibi

Tuteur entreprise : Dr. John Zhai

Expert : M. Florian Simon

Date de soutenance : 16/09/2016

NOTICE ANALYTIQUE

	NOM	PRENOM	
AUTEUR	ORTA	Louis	
TITRE DU TFE	Réalisation d'un logiciel pour l'optimisation énergétique du bâtiment		
	ORGANISME D'AFFILIATION ET LOCALISATION	NOM PRENOM	
MAITRE DE TFE	University of Colorado Boulder, Boulder	Zhai John	
COLLATION	Nbre de pages du rapport 49	Nbre d'annexes (3)	Nbre de réf. biblio. 12
MOTS CLES	Optimisation, énergie, bâtiment, algorithme génétique, EnergyPlus		
TERMES GEOGRAPHIQUES	Etats-Unis, Colorado, Boulder, University of Colorado Boulder		
RESUME	<p>Dans un contexte d'attention accrue quant à la consommation et au prix de l'énergie, les nouveaux concepteurs de bâtiment recherchent avant tout l'efficacité énergétique et économique, en tentant d'optimiser au maximum leurs choix. Ce projet a pour but de répondre à ce besoin d'optimisation grâce à l'outil informatique. Le logiciel, créé de toute pièce pendant le travail de fin d'étude, couple un algorithme génétique à l'outil de simulation du bâtiment EnergyPlus. Ce programme a pour mission de fournir la combinaison optimale des éléments de construction, dans une optique de réduction des coûts.</p>		
ABSTRACT	<p>In a context of growing interest about energy consumption and price, new building constructors are looking for energetic and economic efficiency, trying to optimize their choices. This project aims to answer this optimization need thanks to computer science. The software, entirely designed during this internship, couple a genetic algorithm (GA) to the simulation engine EnergyPlus. This software's goal is to find the optimal combination of the building elements to reduce the annual costs.</p>		

REMERCIEMENTS

Je voudrais adresser mes premiers remerciements au Dr Zhai qui m'a accueilli au sein de son laboratoire et de son équipe. Le docteur Zhai a su montrer une extrême confiance en moi et une patience à toute épreuve jusqu'à la remise du logiciel. Un grand merci également à son équipe qui a su m'accueillir chaleureusement et m'apporter son soutien moral.

Ce travail de fin d'étude ainsi que ce voyage n'auraient pas pu être possible sans l'aide du responsable de la V.A. bâtiment M. El Mankibi qui m'a proposé cette opportunité. Cette expérience m'a permis d'apprendre bien plus que les bases de la programmation et le langage JAVA, elle a aussi été une expérience au sein d'une équipe de recherche et une ouverture culturelle aux États-Unis.

Ce travail marquant également la fin de mes études, j'en profiterai pour remercier globalement l'ENTPE pour la précieuse formation reçue et l'expérience acquise au long de ces trois années. Mes remerciements à mes camarades et notamment Jordan, Mathieu et Pierre pour tous ces moments inoubliables.

Enfin le dernier mot sera pour mes parents qui m'ont toujours soutenu, quels qu'aient été mes choix, et grâce à qui je suis ici aujourd'hui.

SOMMAIRE

Table des matières

1 ETAT DE L'ART	9
1.1 Programmes de simulation énergétique du bâtiment	9
1.1.1 DOE2	9
1.1.2 BLAST	9
1.1.3 TRNSYS	10
1.1.4 ENERGYPLUS	10
1.1.5 Autres programmes	10
1.1.6 Bilan	11
1.2 Interfaces EnergyPlus	11
1.2.1 Interfaces de base	11
1.2.2 Logiciels	13
1.2.3 BEopt : un programme d'optimisation	14
1.3 Optimisation énergétique du bâtiment	15
2 PRESENTATION DU PROJET	16
2.1 EnergyPlus comme programme de simulation	16
2.1.1 Le choix	16
2.1.2 Les connaissances	16
2.2 Un programme en JAVA	17
2.2.1 Choix	17
2.2.2 Apprentissage	19
2.3 L'algorithme Génétique	19
2.3.1 Principe	19
3 HOW IT WORKS (ENGLISH)	22
3.1 Main Structure	22
3.2 Databases	22
3.3 Writing IDF	23

3.4	Running Simulation	23
3.5	Getting Individual skills	23
3.6	GA	24
3.7	Interface	24
3.7.1	Run a simulation	25
3.7.2	2-Manage the databases	27
4	RESULTATS	29
4.1	Logiciel	29
4.2	Interface	29
4.3	Simulations	29
4.4	Données utilisées	29
4.5	Résultats sur le bâtiment simulé	30
4.5.1	Comparaison au bâtiment original	30
4.5.2	Optimisation	30
4.5.3	Influence du nombre d'individus/génération	32
4.5.4	Temps de calcul	32
5	CONCLUSION	33
6	BIBLIOGRAPHIE	34
7	APPENDIX A: GET THE « PREPARED IDF FILE »	39
7.1	Why do I need it	39
7.2	What does the program actually write	39
7.3	How to prepare this file	39
7.3.1	Exterior walls	39
7.3.2	- Zone infiltration	41
7.3.3	- Windows ratio	42
7.3.4	-Fix the outputs	43
8	APPENDIX B: HOW TO RUN A SIMULATION	45
9	APPENDIX C: CALCULATIONS FOR WINDOWS RATIO	48

LISTE DES FIGURES

Figure 1 Interface EP-launch	12
Figure 2 Interface IDF Editor	13
Figure 3 méthode de recherche séquentielle.....	14
Figure 4 Exemple de structure objet.....	18
Figure 5 Représentation des gènes d'un individu.....	19
Figure 6 Schéma de croisement « cross-over ».....	20
Figure 7 Schéma de fonctionnement de l'algorithme génétique.....	21
Figure 8 global software logics.....	22
Figure 9 Principe of the genetic algorithm used	24
Figure 10 Main Interface window	25
Figure 11 Simulation Launcher window.....	26
Figure 12 Result file displayed at the end of the simulation	27
Figure 13 Database Interface	27
Figure 14 Dialog to register a new exterior wall.....	28
Figure 15 Energy cost for every generation.....	30
Figure 16 Total cost for each generation.....	31
Figure 17 total cost function of energy cost.....	31
Figure 18 influence du nombre d'individu/génération	32
Figure 19 Example: find the "exterior walls description"	40
Figure 20 Example: Zone infiltration after rewriting	41
Figure 21 Example: setting the windows' construction price	42
Figure 22 Example : fixing the outputs	44
Figure 23 Simulation Launcher.....	45
Figure 24 calculation for windows ratio	48

INTRODUCTION

La situation globale actuelle laisse présager d'importants problèmes liés à l'énergie dans un futur proche. Besoins énergétiques décuplés lié au grandissement et à la modernisation de la population mondiale, et raréfaction des sources d'énergie fossile. Pour contrer cette tendance, des gains en « efficacité énergétique » peuvent être une solution, ils diminuent la consommation énergétique et d'autre part les coûts qui lui sont liés.

Parallèlement, les progrès colossaux qu'ont connus les domaines de l'informatique et des technologies ces dernières décennies ont mis entre nos mains des outils de simulations puissants.

Dans le domaine du bâtiment, la rencontre de ces deux facteurs a donné naissance aux outils de simulation énergétique. Toujours plus puissants, toujours plus précis.

La quantité d'énergie dépensée pour le chauffage d'une maison, son éclairage, l'ensemble de ses systèmes représente une source potentielle d'économies très importante. Economie d'énergie mais aussi d'argent. Comment déterminer si l'investissement dans une rénovation sera rentable en termes d'économie ? Comment le rendre le plus rentable possible ? Nous voilà dans un problème d'optimisation.

Ce projet a pour vocation de créer un outil informatique d'**optimisation énergétique du bâtiment**. Ce logiciel aura pour mission de déterminer les bouquets d'options de construction les plus efficaces en termes d'économie. Pour cela, il couple un outil de simulation : EnergyPlus à un algorithme d'optimisation : l'algorithme génétique.

1 ETAT DE L'ART

Cette section a pour but de dresser un état de l'art en matière d'optimisation énergétique du bâtiment et de démontrer l'intérêt du projet réalisé pendant ce TFE.

1.1 Programmes de simulation énergétique du bâtiment

Lorsque l'on parle de simulation du bâtiment, de nombreux et divers programmes nous viennent à l'esprit. Ces cinquante dernières années, partout à travers le monde une multitude de logiciels ont été développés à des fins similaires, par différents laboratoires/entreprises sans forcément de préoccupation pour l'existant.

Nous nous préoccupons dans cette section uniquement des programmes liés : au calcul des zones de charge thermique (zone load), à l'enveloppe du bâtiment, aux gains solaires, à la ventilation, à la production d'énergie renouvelable, aux équipements électriques du bâtiment, aux systèmes et équipements dits « HVAC » (Chauffage, ventilation, climatisation) et à l'évaluation économique des performances. Ces programmes prennent également en compte pour la plupart des données climatiques en entrée.

1.1.1 DOE2

Doe2 est un programme capable de prévoir le besoin énergétique d'un bâtiment et les coûts liés. Développé vers la fin des années 60 aux Etats-Unis, il prend ses origines dans un programme du Ministère des Postes et Communications (Post Office). Il est sponsorisé par le Département américain de l'énergie (DOE). DOE se caractérise surtout par l'utilisation d'une approche utilisant un « facteur de pondération de pièce » (room weighting factor approach). Cette approche implique l'utilisation de facteurs représentant l'évolution de la charge thermique ou de la température de l'air pour un gain instantané de chaleur.

Encore utilisé largement en recherche aujourd'hui, DOE2 présente l'avantage d'être rapide et simple d'utilisation, il est cependant très difficile à faire évoluer. DOE2 a donné naissance au logiciel gratuit eQuest qui implémente une interface graphique à DOE2 et qui se veut simple d'utilisation.

1.1.2 BLAST

BLAST est lui développé au début des années 70, aux États-Unis, dans le cadre d'un programme mené par le « Bureau national pour la détermination des charges standard ». La préoccupation énergétique fait alors son apparition mondiale avec le premier choc pétrolier. BLAST est sponsorisé par le Département de la Défense des Etats-Unis. BLAST se distingue de DOE2 par l'utilisation d'une approche différente des transferts thermiques : « the thermal balance method » ou méthode de l'équilibre thermique. Cette méthode consiste à appliquer la première loi de la thermodynamique à chacune des surfaces entourant l'espace concerné, puis à résoudre le système d'équation. Le Département de la Défense cesse de supporter le développement de BLAST en 1995 pour cause de contraintes budgétaires.

1.1.3 TRNSYS

Développé par le Laboratoire d'énergie solaire de l'Université du Wisconsin depuis le début des années 80, TRNSYS est un programme de modélisation thermique du bâtiment. Il permet de simuler un bâtiment en régime dynamique et est surtout reconnu pour sa capacité à simuler les systèmes de chauffage, de ventilation et de conditionnement d'air. Il possède sa propre interface graphique et une bibliothèque de modèles très fournie. Constamment mis à jour, TRNSYS est actuellement considéré comme le concurrent majeur d'EnergyPlus. Il propose les avantages de posséder une interface graphique bien construite et d'être plus poussé en matière de modélisation de système HVAC. TRNSYS est toutefois payant.

1.1.4 ENERGYPLUS

Si Blast et Doe2 ont été largement utilisés pendant plusieurs décennies, le vieillissement des codes utilisés et l'évolution colossale en termes de capacité informatique, ont rendu difficile et couteux en temps la mise à jour de ces logiciels. En 1996, le Département américain de l'énergie décide de développer un nouvel outil de simulation énergétique du bâtiment : EnergyPlus. Combinant les ressources, les équipes et les points forts de chacun de BLAST et DOE2, EnergyPlus a été redéveloppé de zéro. Il est écrit dans le langage Fortran 90 et ne possède pas d'interface propre. Il utilise un fichier IDF (input data file) comme entrée et émet des fichiers de sortie (output files). EnergyPlus est open-source, il est donc utilisable et téléchargeable par tout le monde. Il est fourni avec des interfaces simplistes mais difficile d'accès permettant de créer un fichier d'entrée et de lancer les simulations.

1.1.5 Autres programmes

On compte des dizaines d'autres programmes de simulation, développés dans différents laboratoires partout dans le monde. Ces programmes sont souvent développés et utilisés nationalement. On citera par exemple :

- ESP-r : outil open-source, développé par le British Energy System Unit depuis les années 1970. Le programme est utilisé pour des travaux de recherche principalement.
- DeST : développé par l'université chinoise de Tsinghua depuis le début des années 1980
- Le logiciel CLIM2000, développé en France. Egalement open source, ce programme est un « outil de résolution numérique d'équations algèbro-différentielles », il permet de calculer « l'évolution temporelle du bâtiment et de ses systèmes de chauffage et de ventilation de façon couplée ».

1.1.6 Bilan

L'ensemble de ces programmes sont développés partout à travers le monde, principalement par différents pays soucieux d'avoir leurs propres logiciels. Parmi ce très grand choix, EnergyPlus et TRNSYS semblent proposer les versions les plus évoluées de calculateur à l'heure actuelle.

EnergyPlus se démarque par sa gratuité et sa capacité à accepter une multitude de modules. Ces modules peuvent être développés par ceux qui le souhaitent, afin de lui apporter de nouvelles capacités. EnergyPlus souffre cependant d'un manque d'interface utilisateur officielle et efficace. De nombreuses interfaces payantes ou non ont été développées en externe.

TRNSYS se distingue, lui, par son interface graphique qui se veut facile d'utilisation. Il est une référence en termes de simulation de systèmes de chauffage, de ventilation et de climatisation complexes. Il reste toutefois payant.

Comme le constatent Drury B. Crawley et ses collègues dans un de leurs articles visant à comparer 20 des principaux programmes de simulation du bâtiment :

« Même parmi les outils les plus matures, il n'existe aucun langage commun pour décrire ce que peuvent potentiellement faire ces outils. Il y a beaucoup d'ambiguïté, ce qui va continuer à demander un travail supplémentaire dans le futur ».

Si la diversité de ces programmes est une aubaine en cela qu'elle permet de nombreuses différentes simulations à comparer, elle pose également un problème en termes de comparaison des résultats et même de choix du programme. Un défi majeur actuel est donc de réussir à faire communiquer ces différents logiciels, de nombreux travaux de recherches ont d'ailleurs été publiés dans ce sens.

1.2 Interfaces EnergyPlus

1.2.1 Interfaces de base

Lorsqu'on télécharge le package EnergyPlus, de nombreux outils sont fournis. Certains peuvent servir d'interface. Nous présenterons ici les deux plus communs. L'ensemble de ces outils sont disponibles sur le site d'EnergyPlus, dans la rubrique Extras.

1.2.1.1 Ep-launch

Pour lancer une simulation EnergyPlus il n'existe pas une infinité de méthodes. Il est d'abord possible de lancer une simulation depuis l'exécuteur de commande Windows, cela nécessite une syntaxe parfaite et un minimum de connaissance en informatique. Il est possible de lancer EnergyPlus « manuellement » depuis un exécutable, il faut alors rassembler tous les fichiers nécessaires dans un même dossier et les renommer.

L'outil Ep-launch, fourni avec EnergyPlus, permet de sélectionner simplement le fichier IDF à simuler et le fichier EPW des données climatiques à utiliser. Il permet également d'avoir aisément accès aux sorties générées.

L'interface Ep-launch se présente comme ceci :

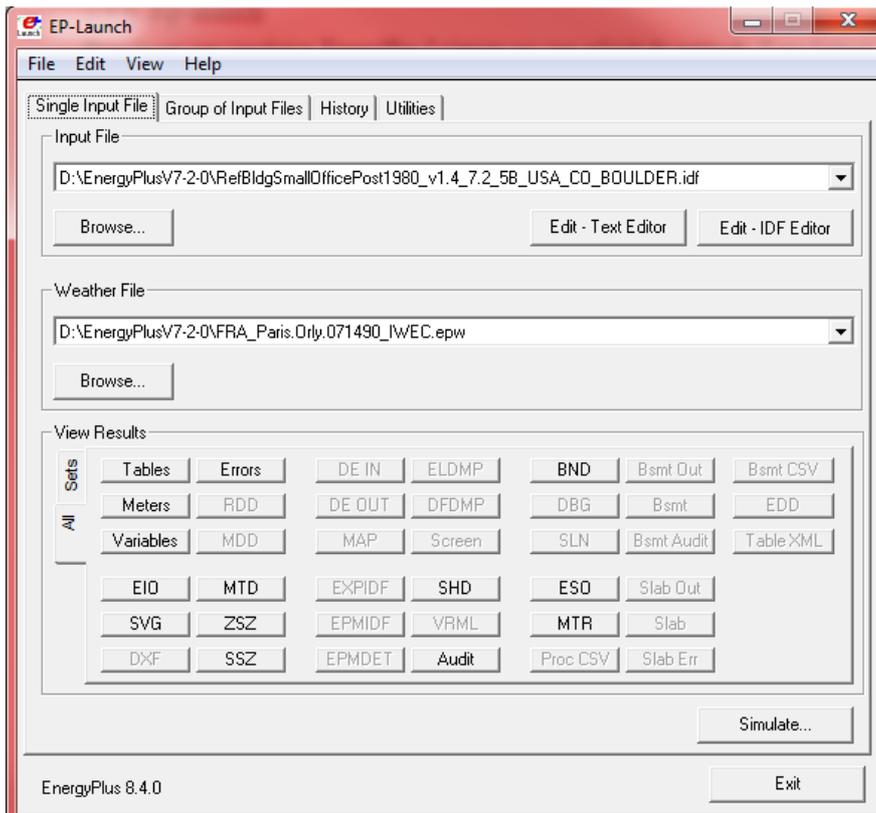


Figure 1 Interface EP-launch

1.2.1.2 IDF Editor

Avant de pouvoir lancer une simulation, il faut un fichier à simuler, IDF Editor propose d'écrire ce fichier. Pour pouvoir écrire un fichier IDF depuis IDF Editor, il faudra s'armer d'une immense patience. En effet, chaque caractéristique de chacun des matériaux utilisés, chaque surface, chaque point devra être renseigné. Il est donc quasiment impossible de créer un IDF à partir de cette interface. Elle permet cependant d'accéder facilement aux données d'un bâtiment et de les modifier si besoin.

La lecture et la modification sont simplifiées par cette interface qui facilite l'accès aux différents objets.

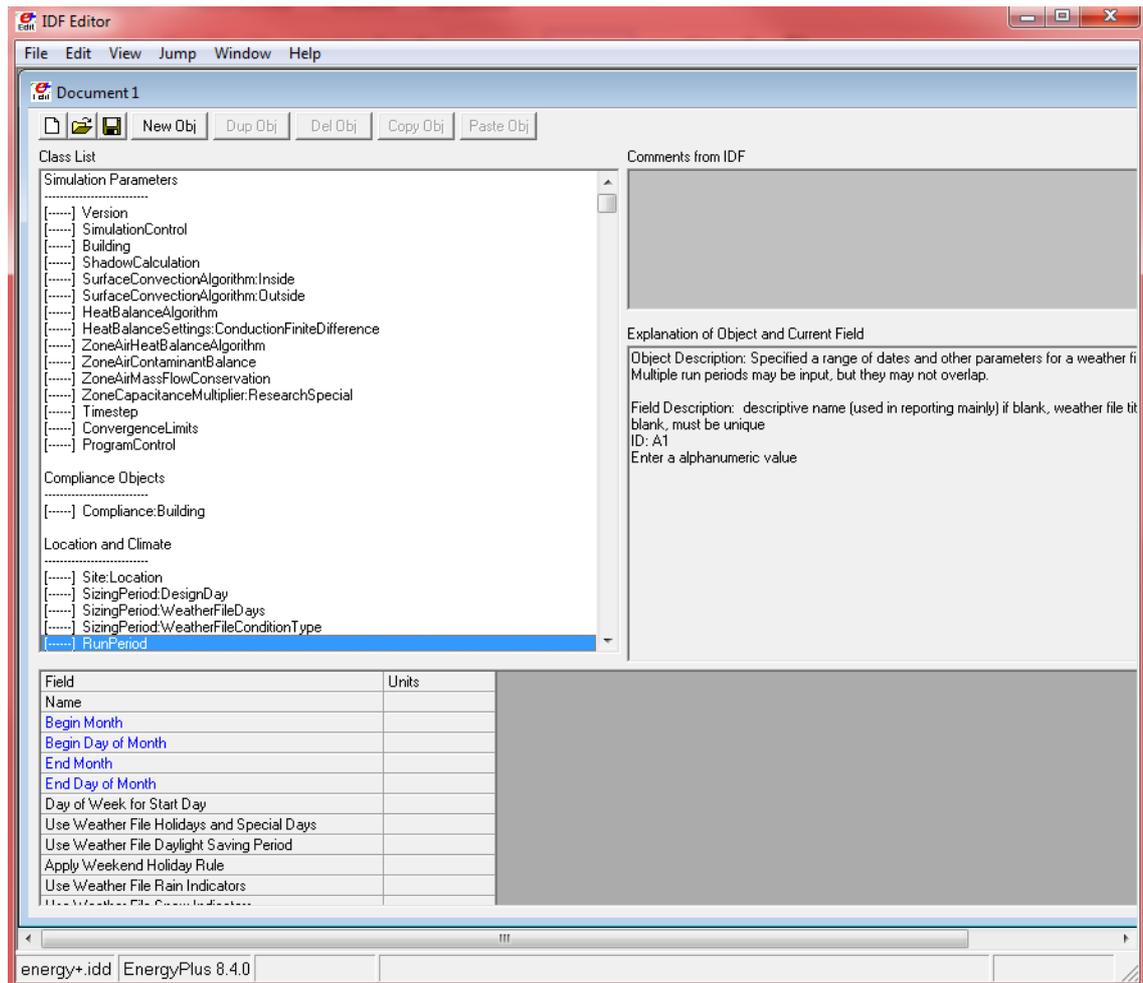


Figure 2 Interface IDF Editor

1.2.2 Logiciels

Les quelques logiciels de simulation présentés ici font partie d'une multitude d'autres logiciels qui ont pour point commun d'utiliser EnergyPlus comme calculateur.

1.2.2.1 DesignBuilder

Interface EnergyPlus par excellence, DesignBuilder sort en 2005. Il propose de simuler un bâtiment créé directement depuis son interface. Il propose une très grande variété d'options avancées : ventilation naturelle, contrôle de la lumière du jour, sol chauffant et bien d'autres. DesignBuilder est largement utilisé dans le secteur privé, conception et bureaux d'études par exemple.

1.2.2.2 EFFEN

EFFEN a vu le jour en 2007. C'est un logiciel qui propose la particularité d'étudier les impacts énergétiques des systèmes de fenêtre. Il est utilisé pour concevoir de grands bâtiments tertiaires ou résidentiels comportant d'importantes surfaces vitrées.

1.2.2.3 *Sefaira*

Plutôt destiné aux architectes, Sefaira permet d'optimiser les formes et surfaces des bâtiments à des fins d'économies énergétiques. Le logiciel a la particularité de calculer et donc de donner les informations rapidement, afin de multiplier les configurations testées.

1.2.2.4 *AECOSim Energy Simulator*

Sorti par Bentley Systems en mai 2013, la seconde version de ce logiciel utilise EnergyPlus7.2. Ce logiciel propose notamment une compatibilité avec un très grand nombre de formats et logiciels : AutoCad, Revit et MicroStation par exemple.

1.2.3 BEopt : un programme d'optimisation

Si l'on commence à penser optimisation, il est nécessaire d'évoquer BEopt. Ce logiciel a grandement inspiré notre projet.

BEopt a été développé par le Laboratoire National des Energies Renouvelables américain.

Conçu pour étudier les bâtiments résidentiels, BEopt permet de créer un bâtiment et de trouver des groupes de solution pour optimiser le prix du bâtiment et l'énergie consommée. Il est possible de rechercher différents niveaux d'économie d'énergie, en allant jusqu'au bâtiment à énergie positive. BEopt permet d'intégrer la production d'énergie renouvelable par l'intermédiaire notamment de photovoltaïque.

Il utilise un algorithme d'optimisation séquentiel : l'algorithme séquentiel part du bâtiment d'origine et fait varier un paramètre, la valeur de ce paramètre la plus efficace est alors retenue et il passe au suivant. La figure suivante illustre ce processus pour un problème de minimisation :

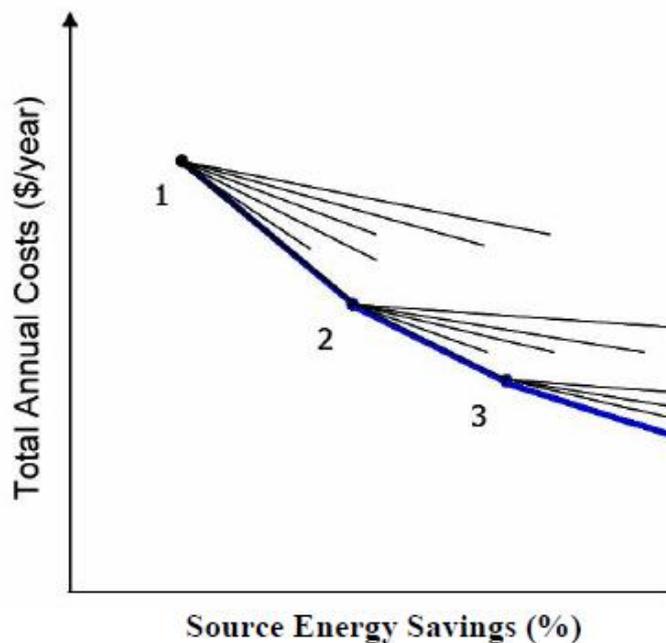


Figure 3 méthode de recherche séquentielle

Bien sûr, les interactions entre paramètres sont prises en compte à chaque stade. L'algorithme est plus complexe qu'il n'en a l'air sur cette figure

1.3 Optimisation énergétique du bâtiment

L'idée de ce projet a été influencée par un travail de recherche en particulier. Dans le cadre de son master de science à l'université du Colorado, Daniel TUHUS-DUBROW a comparé les performances de divers algorithmes pour un problème d'optimisation du bâtiment. Son travail a d'abord consisté en un couplage algorithme génétique-DOE2 puis à la comparaison avec d'autres programmes. Les deux autres types d'optimisations proposées sont l'optimisation séquentielle (avec BEopt) et l'optimisation par essaim particulaire. La technique séquentielle a été vite éliminée par le nombre trop important de simulations requises. Les résultats montrent que l'algorithme génétique est plus rapide parce qu'il requiert moins de simulations pour arriver au résultat optimal : en moyenne 20% de simulations ont pu être économisées par rapport à la technique à essais particulaire. Au final une différence maximale de 0,5% avec la solution optimale peut être notée.

De nombreux travaux de recherche ont déjà couplé l'algorithme génétique à différents outils de simulation et prouvé l'efficacité de cette association :

- Jonathan A. Wright, Heather A. Loosemore, Raziye. "Farmani, Optimization of building thermal design and control by multi-criterion genetic algorithm". In: Energy and Buildings, volume 34, pages 959-972 (October 2002).

Dans cet article sont étudiées les performances des algorithmes génétiques multicritères. Les chercheurs concluent de leurs résultats que l'algorithme génétique est parfaitement adapté aux problématiques d'optimisation du bâtiment.

- Mohamed Hamdy , Ala Hasan, Kai Siren. "A multi-stage optimization method for cost-optimal and nearly-zero-energy building solutions in line with the EPBD-recast 2010". In: Energy and Buildings, volume 56, Pages 189–203 (2013).

Ce travail de recherche s'intéresse à l'optimisation d'une maison en Finlande. Parcourant un espace de 3×10^9 possibles, le programme permet aux chercheurs d'identifier les points clés de la réduction des coûts liés à l'énergie et de minimiser ceux-ci de manière très efficace. La solution finale présente une réduction de la facture énergétique de 47% par rapport à la valeur standard du code de construction finnois.

- Ming-Der Yang^a, Min-Der Lin, Yu-Hao Lin, Kang-Ting Tsai. "Multiobjective optimization design of green building envelope material using a non-dominated sorting genetic algorithm". In : Applied thermal engineering (2016).

Dans cet article plus récent, un algorithme génétique est couplé à un modèle de simulation de l'enveloppe d'un bâtiment. Ces simulations étaient plus précisément faites dans le cas de l'étude d'un « bâtiment vert ». Les résultats montrent un gain économique allant jusqu'à 47,1% par rapport au design originel proposé par l'architecte.

Evidemment d'autres études sont à dénoter ces 15 dernières années. La plupart concluent à la nécessité de développer des systèmes d'optimisation. D'autre part si l'efficacité (rapidité) de l'algorithme génétique n'est plus à démontrer, la solution trouvée ne diffère souvent que très légèrement de l'optimum que l'on trouverait par une autre solution.

2 PRESENTATION DU PROJET

Au vu des nombreux logiciels déjà existants sur le marché, la volonté de créer un nouveau logiciel peut susciter une interrogation.

Si de nombreuses études et travaux de recherche ont couplé l'algorithme génétique à EnergyPlus et démontré les performances de ce couple. Aucun logiciel de ce genre n'a jamais été mis sur le marché.

La réelle nouveauté que veut apporter ce logiciel est l'utilisation d'une base de données en ligne bien spéciale.

L'idée est simple : fournir une base de données réellement représentative du marché à l'utilisateur. La base de données est alimentée par les industriels qui fournissent volontairement caractéristiques et prix de leurs matériaux ou systèmes. Si leur produit est bon, ou rentable, il sera sélectionné par le logiciel et sera suggéré à l'utilisateur. S'il ne l'est pas, il faudra se poser la question de son prix ou de sa qualité.

L'industriel y gagne de la visibilité pour ses meilleurs produits, l'utilisateur des données fiables et une vision élargie du marché.

Dans ce rapport, il ne sera cependant question que de la réalisation du programme, la gestion des bases de données et leur mise en ligne devra faire l'objet d'un développement futur.

2.1 EnergyPlus comme programme de simulation

2.1.1 Le choix

Le choix d'EnergyPlus comme programme de simulation s'explique par différentes raisons.

EnergyPlus est d'abord un programme développé par le département américain de l'énergie, il apparaît donc naturel de l'utiliser dans un projet au sein d'un laboratoire d'université américaine.

EnergyPlus est aussi un programme « open source », il est entièrement libre de droit et gratuit. L'utilisation de programmes libres de droit est un objectif du projet, de cette manière il est utilisable par tous et ne requiert aucun investissement.

L'absence d'interface efficace ne pose aucun problème étant donné que le projet consiste à en créer une.

Une grande base de données de fichier IDF de « bâtiment génériques » américains (bureaux, écoles, hôpitaux, etc.) partout sur le territoire est disponible gratuitement et permet donc déjà un grand lot de simulation.

On peut trouver ces fichiers sur : <http://energy.gov/eere/buildings/commercial-reference-buildings>.

Enfin EnergyPlus est simple à exécuter, il suffit d'avoir un exécutable EnergyPlus et un fichier IDF bien fait pour pouvoir lancer une simulation.

2.1.2 Les connaissances

Les connaissances relatives au fonctionnement d'EnergyPlus ont d'abord été apportées par le cours Utilisation Rationnelle de l'Energie (URE) dispensé à l'ENTPE. La présentation du logiciel DesignBuilder mais surtout l'explication du fonctionnement du programme de calcul (qui n'est autre qu'EnergyPlus), ont permis une approche facilitée du programme et de ses composants.

Le reste des connaissances a été obtenu majoritairement à travers la documentation officielle disponible sur le site EnergyPlus.net. De très nombreuses et précieuses informations sont disponibles dans l'onglet « Documentation ». Les fichiers sont très fournis et un temps d'adaptation est requis pour pouvoir y naviguer.

On peut y trouver notamment les deux documents :

- **Input Output Reference**, cataloguant toutes les informations traitées par EnergyPlus : entrées comme sorties. Y sont d'abord exposés et expliqués les fichiers utilisés par EnergyPlus. De même, chaque objet et ses caractéristiques y sont décrits. Sont également données les unités, les valeurs par défaut, les informations obligatoires et celles facultatives. Enfin, des exemples de fichier IDF sont fournis.
- **Interface Developer**, dédié comme son nom l'indique aux développeurs d'interface, ce guide a été d'une extrême utilité. Y sont expliqués notamment les coulisses d'EnergyPlus, les conventions d'écriture et les prérequis nécessaires pour lancer une simulation.

Enfin, une intarissable source de connaissance n'est autre qu'internet, beaucoup d'informations sont disponibles notamment sur certains forums.

2.2 Un programme en JAVA

2.2.1 Choix

Pour développer un programme, il existe de très nombreux langages de programmation, dont les plus connus sont : JAVA, C/C++, Python, BASIC et bien d'autres.

Dans le cadre de ce projet, le JAVA a été retenu pour diverses raisons :

- Le JAVA est un langage orienté objet : en JAVA, on définit des « classes » (la classe mur par exemple), ces classes possèdent des propriétés (épaisseur, propriétés thermiques, couleur). Il est ensuite possible de créer une infinité d'instances (ou objet) de cette classe (mur1, mur2, mur3...). Il est également possible de créer pour cette classe des fonctions, qui ne s'appliqueront qu'aux objets issus de cette classe.

Cette structure de langage se trouve être un très bon choix pour travailler avec EnergyPlus. En effet les fichiers d'entrée IDF (Input Data File) ne se trouvent être qu'une suite d'objets écrits avec leurs noms et leurs caractéristiques sous une certaine syntaxe. Ce type de langage est donc parfaitement adapté pour l'écriture de la multitude de ces fichiers IDF dont le logiciel a besoin.

L'image ci-dessous représente trois objets « BuildingSurface : Detailed », donc des surfaces présentes dans le bâtiment et leurs caractéristiques propres. Les commentaires situés après les « ! » sont à titre indicatif, et précisent la caractéristique décrite.

```

BuildingSurface:Detailed,
Perimeter_ZN_3_wall_west, !- Name
wall, !- Surface Type
int-walls, !- Construction Name
Perimeter_ZN_3, !- Zone Name
Surface, !- Outside Boundary Condition
Perimeter_ZN_4_wall_north, !- Outside Boundary Condition Object
NoSun, !- Sun Exposure
NoWind, !- Wind Exposure
AutoCalculate, !- View Factor to Ground
4, !- Number of Vertices
0.0000,18.4600,3.0500, !- X,Y,Z ==> Vertex 1 {m}
0.0000,18.4600,0.0000, !- X,Y,Z ==> Vertex 2 {m}
5.0000,13.4600,0.0000, !- X,Y,Z ==> Vertex 3 {m}
5.0000,13.4600,3.0500; !- X,Y,Z ==> Vertex 4 {m}

BuildingSurface:Detailed,
Perimeter_ZN_4_ceiling, !- Name
ceiling, !- Surface Type
Attic Non-res Floor, !- Construction Name
Perimeter_ZN_4, !- Zone Name
Surface, !- Outside Boundary Condition
Attic_floor_perimeter_west, !- Outside Boundary Condition Object
NoSun, !- Sun Exposure
NoWind, !- Wind Exposure
AutoCalculate, !- View Factor to Ground
4, !- Number of Vertices
0.0000,0.0000,3.0500, !- X,Y,Z ==> Vertex 1 {m}
5.0000,5.0000,3.0500, !- X,Y,Z ==> Vertex 2 {m}
5.0000,13.4600,3.0500, !- X,Y,Z ==> Vertex 3 {m}
0.0000,18.4600,3.0500; !- X,Y,Z ==> Vertex 4 {m}

BuildingSurface:Detailed,
Perimeter_ZN_4_floor, !- Name
Floor, !- Surface Type
ext-slab, !- Construction Name
Perimeter_ZN_4, !- Zone Name
Ground, !- Outside Boundary Condition
, !- Outside Boundary Condition Object
NoSun, !- Sun Exposure
NoWind, !- Wind Exposure
AutoCalculate, !- View Factor to Ground
4, !- Number of Vertices
5.0000,5.0000,0.0000, !- X,Y,Z ==> Vertex 1 {m}
0.0000,0.0000,0.0000, !- X,Y,Z ==> Vertex 2 {m}
0.0000,18.4600,0.0000, !- X,Y,Z ==> Vertex 3 {m}
5.0000,13.4600,0.0000; !- X,Y,Z ==> Vertex 4 {m}

```

Figure 4 Exemple de structure objet

- Le langage Java possède une interface de programmation applicative (API) très bien documentée. L'API est, pour un langage, la voie de communication avec d'autres programmes. Grâce à son API, JAVA peut par exemple aisément communiquer avec un tableur Excel ou une base de données SQL. La documentation Java, très fournie à ce sujet, permet de vite prendre en main ces interactions. Dans ce projet, l'API utilisée pour communiquer avec la base de données est très souvent sollicitée.
- Le langage JAVA est très « portable ». Aujourd'hui, la très grande majorité des utilisateurs possèdent la machine virtuelle JAVA (JVM) qui permet d'exécuter les programmes écrits en JAVA. Cet environnement est disponible gratuitement en téléchargement. Il n'y aura donc que très peu de chances d'avoir des soucis de compatibilité. Une fois le code jugé terminé, il est très simple d'exporter un exécutable et de rendre « portable » son application.
- L'environnement de développement libre « Eclipse IDE » est très aisé à prendre en main et agréable à manipuler. L'environnement de développement est le logiciel possédant de nombreuses fonctionnalités très utiles et qui permet de coder le programme.

2.2.2 Apprentissage

L'apprentissage du langage Java, mais plus globalement des bases de la programmation, s'est fait durant le TFE.

Ces connaissances ont pu être acquises grâce à une base de notions informatiques, obtenues d'abord personnellement mais aussi en première année du cursus Entpe.

La très grande partie des connaissances liées au langage JAVA ont été acquises cependant grâce à la plateforme en ligne Openclassrooms. Ce site internet propose des cours très bien faits sur beaucoup de notions informatiques. Le cours sur JAVA permet par exemple d'apprendre les bases de la programmation orientée objet, du langage et bien plus.

Ici encore l'échange de connaissances entre utilisateurs au travers des forums, notamment Stackoverflow.com (destiné aux programmeurs) a été d'une aide très précieuse.

Cet apprentissage s'est fait seul et a demandé beaucoup de temps, eu égard au point de départ. Ce fait explique que le but premier du TFE ait été de rendre un logiciel opérationnel et efficace. Malheureusement, très peu de temps a été consacré à la génération et l'étude de résultats.

2.3 L'algorithme Génétique

2.3.1 Principe

L'algorithme génétique est un algorithme visant à résoudre un problème d'optimisation aux nombreuses variables. Il fonctionne par itération et cherche à obtenir une solution optimale au problème.

L'algorithme génétique est dit évolutionniste, il s'inspire de la théorie de l'évolution pour parvenir à ses fins. Cette section tentera d'expliquer les principes généraux de ce type d'algorithme

2.3.1.1 Individu

Chaque solution potentielle au problème est incarnée par un individu, chaque individu possède un ensemble de gènes (ou génotype) correspondant à ses caractéristiques. Dans notre cas chaque individu (ou solution) possède un gène lié au type de mur extérieur, un gène lié à la surface vitrée orientée au sud etc.

2.3.1.2 Gènes

Il existe différentes façons de représenter les gènes dans un algorithme génétique, dans le cadre de ce programme on utilise un codage binaire, chaque gène est donc représenté par une série de 0 et de 1. La confusion est souvent faite entre gène et génotype.

Ci-dessous la représentation de l'identité génétique d'un individu, chaque gène est lié à l'une de ses caractéristiques.



Figure 5 Représentation des gènes d'un individu

Population

Chaque itération de l'algorithme génère une population composée d'individus. L'idée est donc de faire évoluer ces populations jusqu'à obtenir l'individu (ou solution) optimal. Le nombre d'individu par population est variable et laissée au choix du concepteur, il est à adapter en fonction du problème, du temps de calcul et de bien d'autres paramètres.

Evolution

La première population est générée aléatoirement. A chaque itération, il s'agit d'améliorer les performances des individus. Il faut donc dans un premier temps les évaluer, la valeur évaluée dépend évidemment de chaque problème. Dans le cas de notre projet, on cherchera à évaluer le coût annuel lié à l'énergie.

Une fois la population évaluée, on en sélectionne les meilleurs individus pour qu'ils se « reproduisent » et génèrent la population suivante. Ici encore le nombre d'individus à sélectionner et la façon de les sélectionner sont laissés à l'appréciation du développeur ; dans ce projet il a été décidé de sélectionner un nombre d'individus égal à la moitié de la population, et d'en générer autant pour créer une nouvelle population de même taille.

La génération des nouveaux individus se fait par croisements ou « cross-over » entre deux individus existants. L'algorithme coupe les génotypes des deux individus aléatoirement et recombine les différentes parties coupées. La figure suivante illustre un croisement.

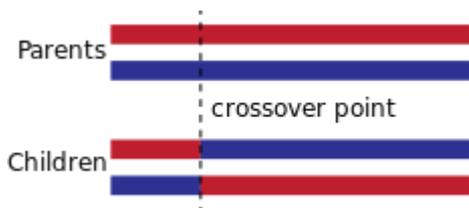


Figure 6 Schéma de croisement « cross-over »

Il existe également une probabilité pour chaque gène de muter : changement aléatoire d'un 0 en 1 ou inversement à chaque itération. La probabilité de mutation est ici encore laissée au choix du développeur.

L'itération peut s'arrêter de différentes façons : nombre de populations limité, atteinte d'une valeur précise ou approximée, convergence. Dans le cas de ce projet, un nombre d'itération est fixée au lancement du programme.

2.3.1.3 Schéma de fonctionnement

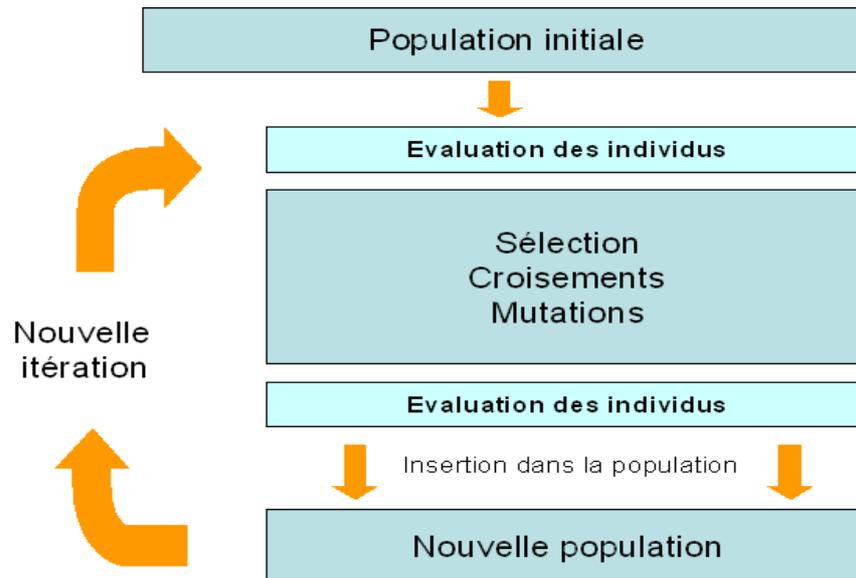


Figure 7 Schéma de fonctionnement de l'algorithme génétique

Limites

Comme montré précédemment, tous les paramètres de l'algorithme sont laissés à l'appréciation du programmeur, cela implique une grande variabilité dans la conception du programme. Des travaux de recherche tendent à montrer que l'optimisation de ces paramètres doit être effectuée individuellement pour chaque problème. Il n'existe donc pas de règle en matière de définition des paramètres du GA et pour l'instant seule l'expérience semble pouvoir en déterminer les meilleurs.

Certains travaux : Mühlenbein (1992), Bäck (1993) et Salomon (1996) démontrent toutefois qu'une probabilité de mutation faible doit être maintenue pour conserver une efficacité dans la convergence.

Dans une de ses publications John J. Grefenstette évoque la possibilité de permettre à l'algorithme de modifier lui-même ses paramètres au cours du processus, ce qui nécessiterait un nombre conséquent d'itération. Cette alternative pourrait donner naissance dans un futur proche à une nouvelle génération d'algorithme génétique.

3 HOW IT WORKS (ENGLISH)

This section aims to explain the program structure and the different functions that makes it work.

3.1 Main Structure

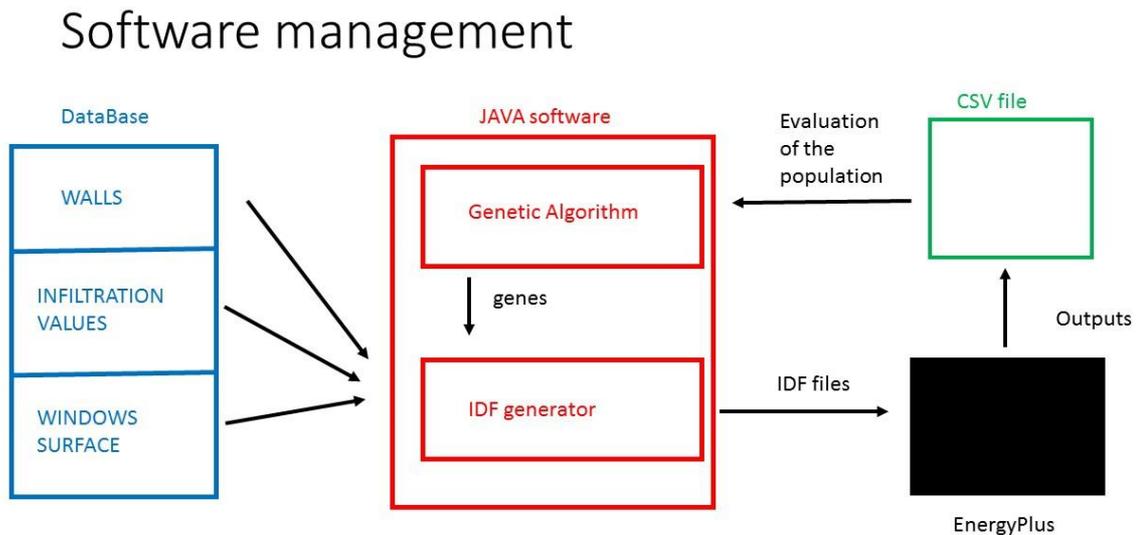


Figure 8 global software logics

The software is based on many interactions between the Java program and other components such as the database, EnergyPlus and EnergyPlus output files.

When the program starts, the genetic algorithm (GA) generates the first generation of individuals. Once created, the population needs to be evaluated. One by one the individuals' genes are sent to another part of the program that writes as many IDF file.

When written every IDF file is sent to EnergyPlus to be run. After a short time of calculation, output files are created. One of this outputs: the csv file, is used to get the values needed to evaluate the individual.

All the population is now evaluated, the program finds the best individuals of this population and generates a new population (the next generation) from that.

Programs stop when the number of generation asked is reached.

3.2 Databases

At the moment only exterior walls and materials' (composing the walls) databases are created.

The communication with databases is done by SQLite, a database engine. The SQLite JDBC Driver implemented in the program allows accessing and creating databases. All the data are now stored in the same file in the Project folder. It is the "test" DB file.

Every connection to this database file is done using the "**connection1**" class.

Every database management is done in the concerned element class: “**ExtWall**” for the exteriors walls, “**Material**” for the materials and “**AirGap**” for the materials of type airgap. (EnergyPlus actually makes differences between objects Material and Material: Airgap in the IDF). Each one of this classes owns:

- A function “add” to add an object to the database
- A function “inventory” to display in a text file the whole database
- A function “extract” that get the desired object information
- A function “syntax” that writes into a String the object’s information according to EnergyPlus syntax
- A function “modify” to allow modifying any characteristic of the objects
- A function “delete” to erase one of them

Class ExtWall also have a function “extWallPrice” that writes the object ComponentCost:LineItem object used for economics calculation.

3.3 Writing IDF

The IDF writing process works according to several steps. The concerned class is **IdfPrint** and its main function *PrintIDF*. It prints an IDF file from a binary number: the individual genes.

Observation:

First of all it is important to note that the software doesn’t write the whole IDF but only the variable elements. This means that currently it needs a “prepared IDF” file as an input. Writing an entire IDF is a really tough task (geometry, schedules, systems, etc.) that would need a further development. Details about this prepared file are given in the “Prepare the IDF document”.

First step of the writing process consists in copying the “prepared input file” to be able to write on it. This is done by the function *copyBaseIDF* that takes as an input the name of the file and look for it in the project directory.

Function then extracts all associated components and gets the String objects from the syntax functions described previously. It then opens a stream to this file and writes everything.

3.4 Running Simulation

To launch, the simulation program uses the fact that it is possible to run EnergyPlus from the Windows Command Prompt with the valid syntax and arguments. The function used is “Run”.

All information about this syntax are available or in the EnergyPlus Interface Developer documentation and an example in the program.

3.5 Getting Individual skills

Once EnergyPlus process is over, it generates output files according to the outputs asked in the IDF. The one this software uses is the CSV file;

Thanks to the function *getdatas* of the class **GetResult** the program can access the data in the CSV. It gets the desired line thanks to the function *getline*. Function is now programed to get data from the economic table but could get any output desired. It currently extracts energy costs for a year and conception costs for the variable items.

3.6 GA

Several classes are used to manage the GA: **GAIndividual** is the individual class, it has the genes as characteristics, it also has the functions to mutate and cross over individuals. The class **Population** has a group of Individuals and the corresponding skills as characteristics, it has the functions to get the best individuals. Finally, the class **Algorithm** manages evolutions of the populations and is the main command to start the entire process.

When running the process from the *runGa* function of the class **Algorithm**, a new object of the class **Population** is created and the GA individuals randomly created. This population is then evaluated to give a new one, this new one is evaluated etc.

Evolution mechanism (function *evolvePopulation* of class **Algorithm**) is the following:

- getting the best individuals with the *getBestIndividuals* function of the **Population** class. The idea is just to get all the population skills like explained previously and compare it. These individuals will be the first half of the next population.

- Two by two, the individuals selected are cut randomly somewhere in their genes and crossed, it's the cross-over. The *crossOver* function can be found in the **GAIndividual** class. The individuals created form the second half of the new population.

- All individuals of the new generation are submitted to a potential mutation and shuffled. *Mutate* function can be found in the **GAIndividual** class and the *shufflePopulation* function in the **Population** class.

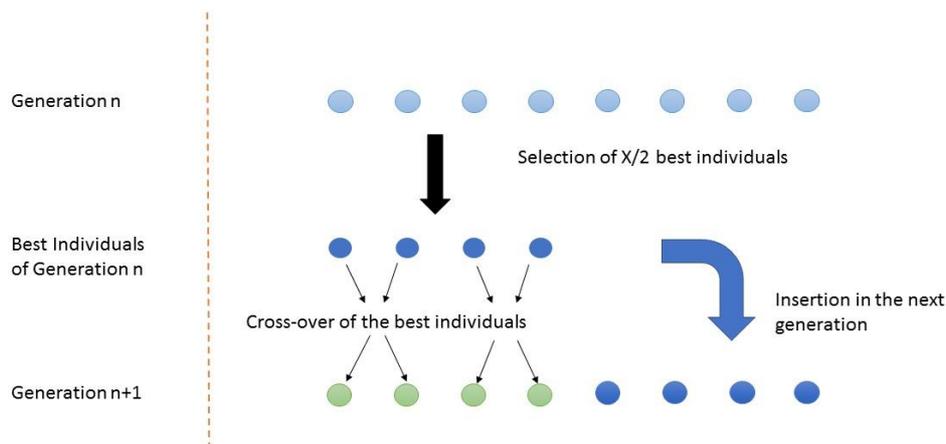


Figure 9 Principe of the genetic algorithm used

3.7 Interface

The user interface is split in 2 parts at the moment: Main interface and Database interface (**MainInterface** and **DataBaseInterface** classes).

The Main interface allows the user to join the Database interface or to start a simulation.



Figure 10 Main Interface window

3.7.1 Run a simulation

The simulation button opens a dialog where several parameters are asked as simulation inputs. Default parameters are the ones for the DOE *Commercial Reference Building Small Office, post-1980 construction* used for the first simulations.

Once the simulation started, the program stays on this window until it's over.

The image shows a 'Simulation Launcher' window with the following fields and values:

- idf name :** enter an idf name (add .idf) :
- epw name :** write an epw name (add .epw) :
- number of generations desired**
Generations Number :
- number of individuals per generation desired**
Individual Number :
- Exterior wall name :** enter the descriptions of exterior walls used in surface description :
- Exterior wall name :** name of the wall concerned by windows ratio :
- Windows' construction name :** enter the windows' construction name :
- Number of external zones :** enter the number of external zones:
- External zones description :** enter the external zones description:

At the bottom of the window are 'OK' and 'cancel' buttons.

Figure 11 Simulation Launcher window

When the entire simulation is done a txt file appears, it is the result file. It is written piece by piece all along the simulation (in classes **Algorithm**, **GetResult** and **IdfPrint**). It displays the result for every individual simulated during the process and the best individual. This file is written to be easily usable in an Excel like spreadsheet software.

```

result example - Bloc-notes
Fichier Edition Format Affichage ?
Exterior wall number, infiltration flow (/exterior surface),windows ratio, Energy costs ($/m²/year), Conception costs ($/m²/year)
*****
Generation0
*****
4,0.002266,0.1,7123.7,3059.382
4,0.002266,0.4,7231.0,3274.741
3,0.003399,0.4,8561.5,3325.5369999999994
1,5.665E-4,0.2,6228.21,2764.753
7,0.003399,0.3,8101.28,3565.14799999999997
3,0.003399,0.1,8518.15,3084.841
4,0.003399,0.2,7934.73,3386.7470000000003
3,0.001133,0.3,7031.38,2478.569
*****
Generation1
*****
3,0.001133,0.3,7031.38,2478.569
3,0.002266,0.1,7747.13,2829.26199999999997
1,0.002266,0.4,7214.85,3172.34899999999997
4,0.002266,0.1,7123.7,3059.382
1,5.665E-4,0.2,6228.21,2764.753
4,0.002266,0.4,7231.0,3274.741
4,5.665E-4,0.2,6265.38,2875.59
4,0.001133,0.3,6505.86,2691.797
*****
Generation2
*****
1,5.665E-4,0.2,6228.21,2764.753
1,0.001133,0.3,6484.0,2585.183
4,0.001133,0.3,6505.86,2691.797
4,5.665E-4,0.2,6265.38,2875.59
4,5.665E-4,0.3,6300.77,2947.376
3,0.001133,0.2,7018.77,2398.337
3,0.001133,0.3,7031.38,2478.569
4,5.665E-4,0.2,6265.38,2875.59
*****
Generation3
*****
4,5.665E-4,0.2,6265.38,2875.59
4,5.665E-4,0.2,6265.38,2875.59
1,0.001133,0.3,6484.0,2585.183
1,5.665E-4,0.2,6228.21,2764.753
4,5.665E-4,0.3,6300.77,2947.376
4,5.665E-4,0.2,6265.38,2875.59
1,5.665E-4,0.2,6228.21,2764.753
1,0.001133,0.2,6450.71,2509.174
*****
Last Generation
*****
1,0.001133,0.3,6484.0,2585.183
1,0.001133,0.2,6450.71,2509.174
1,5.665E-4,0.2,6228.21,2764.753
1,5.665E-4,0.3,6265.07,2840.762
1,0.001133,0.2,6450.71,2509.174
1,0.001133,0.2,6450.71,2509.174
1,5.665E-4,0.2,6228.21,2764.753
1,5.665E-4,0.2,6228.21,2764.753
*****
Best Individual
*****
1,0.001133,0.2,6450.71,2509.174

```

Figure 12 Result file displayed at the end of the simulation

3.7.2 2- Manage the databases

The Database management interface opens on one of 4 modules: exterior walls, materials, one for infiltration and windows ratio.

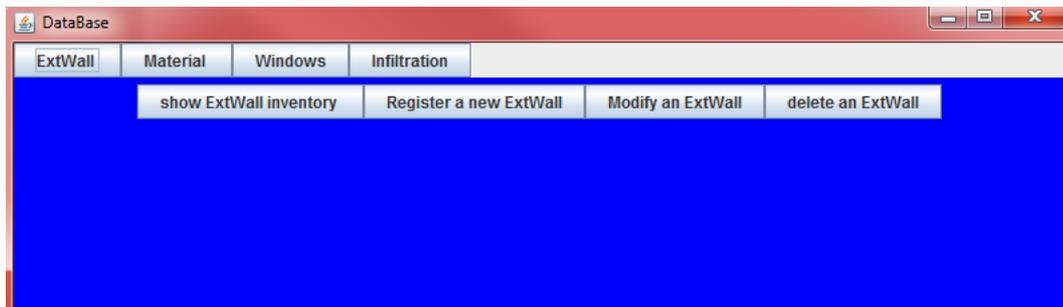


Figure 13 Database Interface

Exterior walls and materials module allow to display the database, register a new element, modify one or delete one. The display action opens a txt file, this file is an image of the database and it is not the real database. The 3 other buttons open on dialogs windows that asks for the parameters to create, change or delete an element.

The dialog box titled "ExtWall DataBase" contains the following fields and controls:

- ExtWall id :** enter an id :
- ExtWall name :** write a name :
- outsideLayer name :** write a name : **Outside layer type** choose a type :
- Layer2 name :** write a name : **Layer 2 type** choose a type :
- Layer3 name :** write a name : **Layer 3 type** choose a type :
- Layer4 name :** write a name : **Layer 4 type** choose a type :
- Layer5 name :** write a name : **Layer 5 type** choose a type :
- Price :** write a price : must be Double (example 15.2) :

Buttons: OK, Cancel

Figure 14 Dialog to register a new exterior wall

The windows ratio module allows the user to change the default values of windows ratio tested by the program. This change is operated only for the session and won't be active anymore if the application is closed before the simulation is run.

Infiltration management module is created but not effective yet. It should be working the same way than the windows ratio one.

4 RESULTATS

4.1 Logiciel

Premièrement une « version Bêta » du logiciel est disponible et effective. Cette version a été remise et approuvée par le Professeur Zhai de l'université du Colorado. Le logiciel se présente actuellement sous la forme d'un dossier contenant les fichiers nécessaires au fonctionnement du logiciel et un exécutable « run.jar ».

Pour être exécuté, deux conditions sont requises : enregistrer le dossier dans le répertoire C:\ de l'ordinateur et posséder l'environnement JAVA (disponible gratuitement).

4.2 Interface

Une interface complète est disponible et permet de lancer la simulation selon les paramètres souhaités, mais aussi de gérer entièrement les bases de données : ajout, suppression ou modification d'objet. Des images des différentes parties de l'interface sont disponibles en annexe ainsi que des guides d'utilisation.

4.3 Simulations

Pour s'assurer du bon fonctionnement du logiciel, plusieurs simulations ont été effectuées sur le même bâtiment avec différents paramètres.

Le bâtiment étudié est un bâtiment tertiaire issu de la base de données du département américain de l'énergie : DOE Commercial Reference Buildings. Cette base de données recense différents types de bâtiments tertiaires construits selon des standards américains. Des fichiers IDF sont disponibles pour chacun des bâtiments. Le bâtiment sélectionné s'intitule : « Small Office, post-1980 construction », il est originellement situé à Boulder. Il n'occupe qu'un étage et se décompose en 4 zones.

La simulation est réalisée avec un fichier de données climatiques (EPW) prises à Paris.

4.4 Données utilisées

Pour tester le logiciel, plusieurs données ont été utilisées : murs, matériaux, infiltrations et ratios de fenêtre.

Les différents murs et les matériaux qui les composent ont été extraits de la base de données « DOE Commercial Buildings », ces constructions et leurs caractéristiques sont donc des valeurs sûres. Cependant, ces fichiers ne comportent aucune donnée économique, exceptés les prix de l'énergie. Il a donc fallu donner un prix à ces différents éléments.

Les prix du m² de murs, de fenêtre ainsi que les différentes valeurs d'infiltration et ratios de fenêtres sont issus d'un projet réalisé par un étudiant de l'université de Brown. (*Daniel TUHUS-DEBROW, Assessing the suitability of genetic algorithms for application in building envelope optimization, Brown University, 2002*). Dans ce projet, l'étudiant compare l'efficacité de différents algorithmes d'optimisation, pour l'optimisation énergétique d'un bâtiment. De nombreuses données utilisées dans cette étude ont été également utilisées pour nos simulations. Voir exemple des bases de données en annexe.

Pour les simulations, ont été utilisées : 8 différentes compositions de murs, 4 valeurs pour le ratio de fenêtre du mur orienté sud, 4 valeurs différentes d'infiltration.

Dans les simulations, le prix des constructions est amorti sur 10 années. La valeur que l'on cherche à minimiser correspond donc à : Prix de l'énergie pour une année + Prix des éléments/10

4.5 Résultats sur le bâtiment simulé

4.5.1 Comparaison au bâtiment original

Il était compliqué de comparer ce bâtiment à l'original en termes de coût total, ne connaissant pas les coûts de construction associés à ce dernier. Une comparaison en termes d'énergie consommée a donc été réalisée et montre des résultats satisfaisants. La figure ci-dessous illustre les progrès effectués :

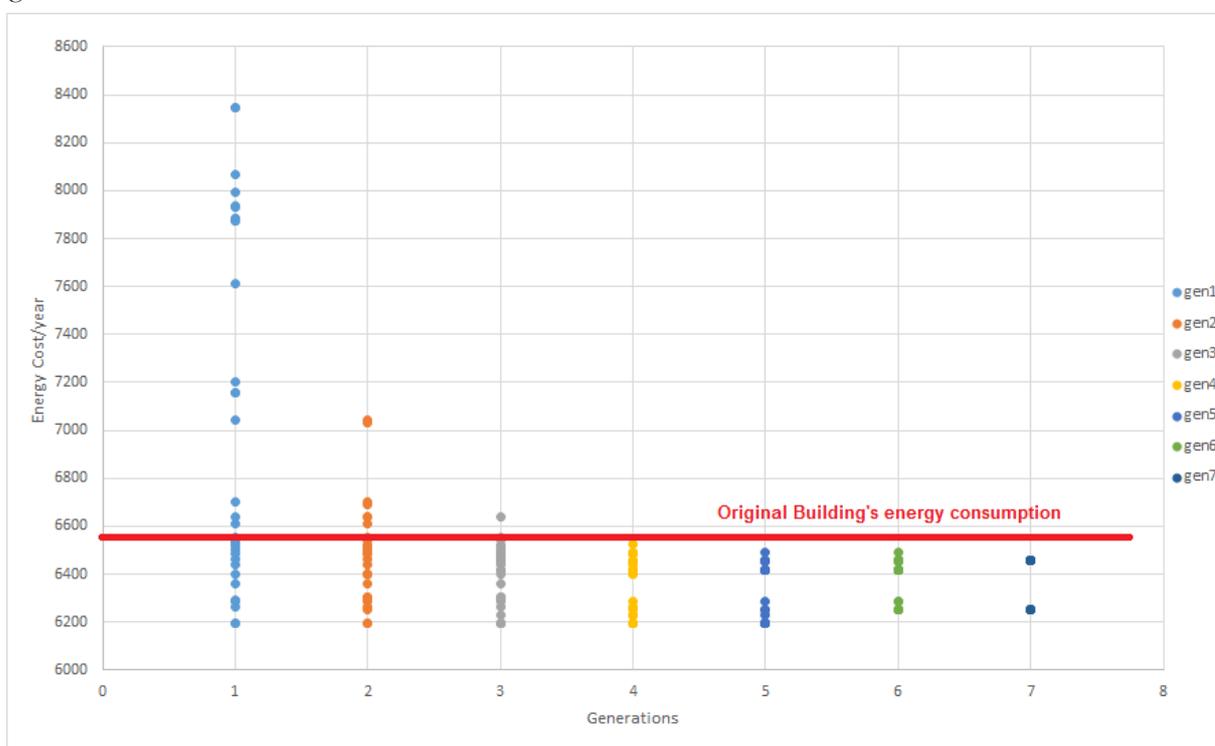


Figure 15 Energy cost for every generation

Il est nécessaire de noter que l'énergie consommée n'est pas la fonction minimisée dans le programme. Ceci explique que certains individus paraissent « meilleurs » dans les premières générations. Cependant les meilleurs individus, ceux de la dernière génération, proposent quand même une consommation d'énergie moindre.

4.5.2 Optimisation

Suite à plusieurs simulations, le résultat optimum obtenu est un coût total de 8662 \$ par an pour ce bâtiment de bureaux.

Le graphe suivant représente la répartition des individus au fur et à mesure des générations par rapport à la fonction à minimiser.

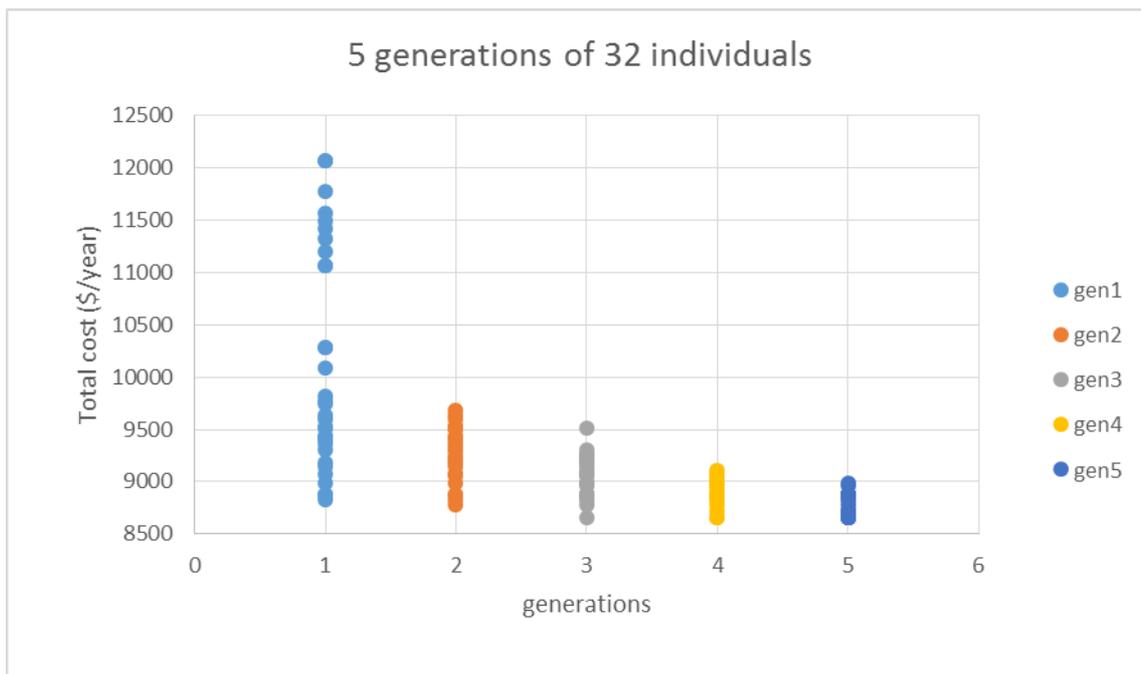


Figure 16 Total cost for each generation

On constate une claire convergence des solutions, au fur et à mesure des générations, vers un minimum de coût.

Le graphe suivant représente le coût total à l'année (fonction minimisée) en fonction du coût uniquement lié à l'énergie.

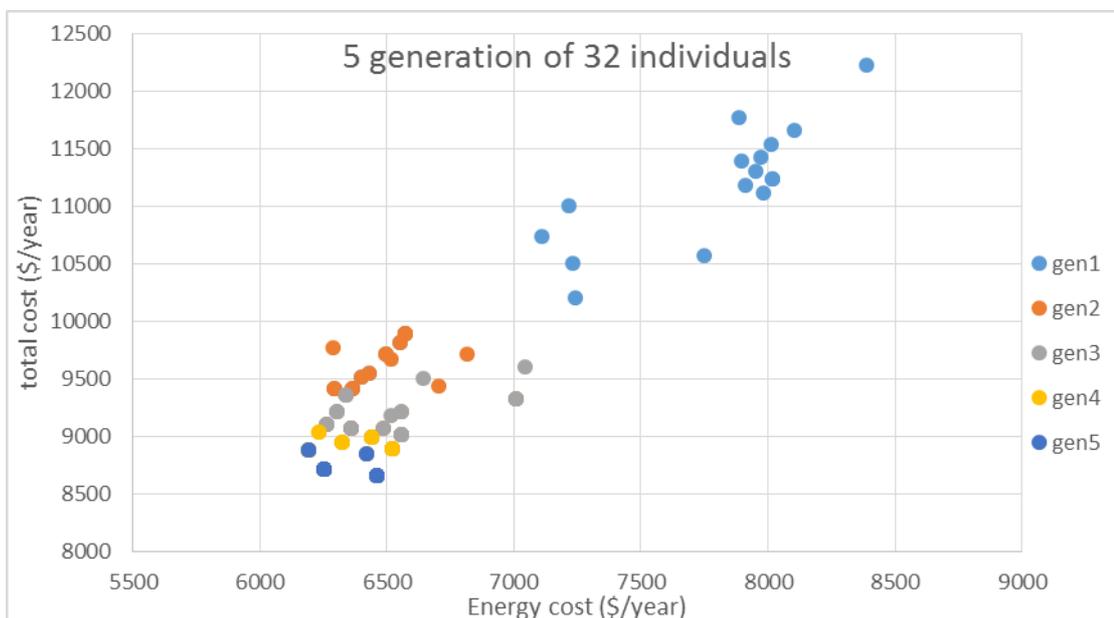


Figure 17 total cost function of energy cost

Une des volontés du projet est de proposer non pas une, mais différentes alternatives à la fin d'une simulation. Les deux derniers types de graphes proposés sont donc entièrement réalisables grâce aux données de sorties que fournit le logiciel. Le second graphe permet en effet de repérer la solution la moins chère mais également celles qui pourraient être un peu plus chères globalement et économiseraient plus d'énergie. Il est tout à fait envisageable dans une optique de conception plus « verte » de privilégier l'économie d'énergie à l'économie d'argent.

4.5.3 Influence du nombre d'individus/génération

Le choix du nombre de générations et d'individus qui les composent a un impact certain sur la durée d'optimisation mais aussi sur la qualité des résultats. Plusieurs simulations ont été effectuées en faisant varier ces paramètres et les résultats sont les suivants :

		Number of generations		
		5	7	9
individuals per generation	8	8959	8882	8896
	16	8662	8662	8662
	32	8662	8662	

Final solution value (\$/year)

		Number of generations		
		5	7	9
individuals per generation	8	29,7	27,4	34,03
	16	37,4	53,21	68,4
	32	100,5	104,6	

Optimization duration (minutes)

Figure 18 influence du nombre d'individu/génération

On peut noter ici que le nombre de paramètres possibles étant restreint, la simulation atteint l'optimum (rappelons que 8662\$/an est le coût le plus petit que l'on puisse obtenir) sans requérir beaucoup de générations et d'individus.

Des tests supplémentaires seront requis dans le futur, pour évaluer du mieux possible l'impact du nombre de générations/individus sur la simulation. Des nombres précis pourraient être fournis à l'utilisateur à titre indicatif pour effectuer une optimisation efficace en fonction du nombre de paramètres sélectionnés.

4.5.4 Temps de calcul

On constate dans le tableau précédent qu'il n'existe pas de corrélation entre le nombre d'individus simulés et la durée du processus. Chacune des simulations et l'ensemble de l'optimisation ont été chronométrés. On constate premièrement que les simulations EnergyPlus représentent 99% de la durée globale du processus. On remarque aussi que chacune des simulations requiert une durée différente pour être effectuée. Cette variabilité peut s'expliquer par le fait que d'autres tâches étaient effectuées simultanément sur l'ordinateur. On peut donc supposer que la durée varie également selon l'ordinateur utilisé.

La durée de calcul est une donnée essentielle pour l'utilisateur. Elle lui permet de savoir où en est sa simulation et éventuellement de choisir ses paramètres de simulation en fonction.

5 CONCLUSION

Le principal objectif de cette étude était de rendre une version bêta d'un logiciel d'optimisation du bâtiment. Pour cela une étude de l'existant a été menée. Des connaissances en termes d'EnergyPlus mais aussi en programmation Java ont été acquises via différentes sources et le programme créé petit à petit avec ses connaissances. Le programme rendu prend en charge un document IDF « préparé », il génère des combinaisons de solutions et les simule pour acquérir une solution optimale grâce à l'algorithme génétique. Les premières simulations effectuées sur un bâtiment de bureau montrent que le programme fonctionne, l'algorithme converge vers une configuration optimale et le système de base de données est opérationnel. Quelques documents expliquant le fonctionnement du logiciel et comment l'utiliser ont été fournis.

Des améliorations seront évidemment nécessaires pour rendre le logiciel entièrement opérationnel et efficace.

Cette dernière expérience de stage a avant tout été extrêmement instructive, les compétences acquises en Java pendant ce stage sont sûrement le plus grand bénéfice de ces quelques mois de TFE. Le TFE a également été une nouvelle expérience dans une équipe de recherche, dans un domaine dans lequel je voudrais travailler. Enfin ce stage a été une opportunité rare d'ouverture culturelle à l'international qui a une grande valeur sur le plan personnel et professionnel.

6 BIBLIOGRAPHIE

-Bruce D.Hunn, Robert D. Busch, P.Richard Rittelmann. Fundamentals of building Energy Dynamics. MIT Press. 1996

-Michael Wetter, Jonathan Wright. «A comparison of deterministic and probabilistic optimization algorithms for nonsmooth simulation-based optimization». In: Building and Environment, volume 39, pages 989-999 (2004).

-Drury B. Crawley , Linda K. Lawrie, Frederick C. Winkelmann, W.F. Buhl, Y.Joe Huang, Curtis O. Pedersen, Richard K. Strand, Richard J. Liesen, Daniel E. Fisher, Michael J. Witte, Jason Glazer. “EnergyPlus: creating a new-generation building energy simulation program”. In: Energy and buildings”, volume 33, pages 319-331(2001).

-Drury B. Crawley, Jon W. Hand, Michaël Kummert, Brent T. Griffith. «Contrasting the capabilities of building energy performance simulation programs ».In : Building and environment, volume 43, pages 661-673(2008).

-Willman AJ. “Development of an evaluation procedure for building energy design tools”. In: Proceedings of building energy simulation volume 85, pages 302–307.(1985).

-Jonathan A. Wright, Heather A. Loosemore, Raziye. “Farmani, Optimization of building thermal design and control by multi-criterion genetic algorithm”. In: Energy and Buildings, volume 34, pages 959-972 (October 2002).

-Mohamed Hamdy , Ala Hasan, Kai Siren. “A multi-stage optimization method for cost-optimal and nearly-zero-energy building solutions in line with the EPBD-recast 2010”. In: Energy and Buildings, volume 56, Pages 189–203 (2013).

- Ming-Der Yanga, Min-Der Lin, Yu-Hao Lin, Kang-Ting Tsai. "Multiobjective optimization design of green building envelope material using a non-dominated sorting genetic algorithm”. In: Applied thermal engineering (2016).

-U.S. Department of Energy, “Guide for Interface Developers”. In: EnergyPlus version 8.5 Documentation. March 2016

-U.S. Department of Energy, “Input Output Reference”. In: EnergyPlus version 8.5 Documentation. March 2016

-U.S. Department of Energy, “Engineering Reference”. In: EnergyPlus version 8.5 Documentation. March 2016

7 APPENDIX A: GET THE « PREPARED IDF FILE »

7.1 Why do I need it

This first version of the software doesn't allow you to create your own building, it actually is just an existing one and makes some of his property vary. To properly use this program, you would need to get an existing IDF file and delete some of its properties, or to write your own according to some rules.

Somewhere in the program each individual (or potential solution) needs to be evaluated, what means that this corresponding IDF needs to be created and run in EnergyPlus. The prepared IDF file will be a common base for every IDF that is run.

7.2 What does the program actually write

To this common base, the program adds characteristics according to the individual's genes.

In this first version it writes:

- Composition of the exteriors walls
- Materials characteristics,
- Windows ratio oriented to the south
- Zone infiltration values.
- The prices of every of this components

The process is pretty simple. For each individual the program copies the prepared IDF to another file and writes the desired characteristics at the end of this new file.

7.3 How to prepare this file

First of all: the IDF file must be written to be run in EnergyPlus 8.4 version. Any previous version IDF can be updated using *IDFVersionUpdater*, program distributed within the EnergyPlus package.

To make all these modifications, we recommend using IDF Editor Program also included in the package. All the examples will be done using this program.

If you are using an IDF file, the following adjustment must be done before running the software:

7.3.1 Exterior walls

The current "construction" object used for exterior walls must be deleted to allow the program to replace it. Note the name of that construction or find it in the "BuildingSurface: detailed" objects, it will be asked later as an input.

The screenshot shows a software window with a menu bar (File, Edit, View, Jump, Window, Help) and a toolbar with icons for New Obj, Dup Obj, Del Obj, Copy Obj, and Paste Obj. Below the toolbar is a 'Class List' on the left and a 'Comments from IDF' area on the right. The 'Class List' contains various object classes, with '[0043] BuildingSurface:Detailed' selected. The 'Comments from IDF' area contains an 'Explanation of Object and Current Field' section with the following text:

Object Description: Allows for detailed entry of building
 Field Description:
 ID: N20
 No default value available

Below the class list and comments is a table with three columns: Field, Units, and a value. The 'Surface Type' field is highlighted with a red box, showing the value 'Wall' and 'Mass Non-res Ext Wall'.

Field	Units	Obj24
Name		Perimeter_ZN_1_wall_south
Surface Type		Wall
Construction Name		Mass Non-res Ext Wall
Zone Name		Perimeter_ZN_1
Outside Boundary Condition		Outdoors
Outside Boundary Condition Object		
Sun Exposure		SunExposed
Wind Exposure		WindExposed
View Factor to Ground		autocalculate
Number of Vertices		4
Vertex 1 X-coordinate	m	0
Vertex 1 Y-coordinate	m	0
Vertex 1 Z-coordinate	m	3,05
Vertex 2 X-coordinate	m	0
Vertex 2 Y-coordinate	m	0
Vertex 2 Z-coordinate	m	0
Vertex 3 X-coordinate	m	27,69
Vertex 3 Y-coordinate	m	0
Vertex 3 Z-coordinate	m	0
Vertex 4 X-coordinate	m	27,69
Vertex 4 Y-coordinate	m	0
Vertex 4 Z-coordinate	m	3,05
Vertex 5 X-coordinate	m	
Vertex 5 Y-coordinate	m	
Vertex 5 Z-coordinate	m	
Vertex 6 X-coordinate	m	
Vertex 6 Y-coordinate	m	
Vertex 6 Z-coordinate	m	

Figure 19 Example: find the "exterior walls description"

Exterior Wall description: Prior to run, the program will ask for an "exterior walls description". In this file, every surface corresponding to an exterior wall uses the construction "Mass Non-res Ext Wall", that's the Exterior Wall description needed. The corresponding construction needs to be deleted in the original file.

7.3.2 - Zone infiltration

First “Zone infiltration objects” must also be deleted for external zones.

Then external zones must be renamed the same way, with a number going from 1 to x (for x zones). See example below. Program currently just works on flows from exterior areas, “central” zones infiltration doesn’t need to be erased.

The screenshot shows the IDF Editor interface. On the left, the Class List contains various objects, with 'Zone Infiltration Design Flow Rate' selected. The main area displays the object's properties in a table format. Below the table, there is a text box for 'Explanation of Object and Comment Field' containing a detailed description of the infiltration calculation formula and its application.

Field	Obj1	Obj2	Obj3	Obj4	Obj5	Obj6
Name	Core_ZN_Infiltration	Attic_Infiltration	ExWall	ExWall2	ExWall3	ExWall4
Zone or ZoneList Name	Core_ZN	Attic	Perimeter_ZN_1	Perimeter_ZN_2	Perimeter_ZN_3	Perimeter_ZN_4
Schedule Name	INFL_QUARTER_ON_SCH	ALWAYS_ON	INFL_QUARTER_ON_SCH	INFL_QUARTER_ON_SCH	INFL_QUARTER_ON_SCH	INFL_QUARTER_ON_SCH
Design Flow Rate Calculation Method	AirChanges/Hour	AirChanges/Hour	Flow/ExteriorArea	Flow/ExteriorArea	Flow/ExteriorArea	Flow/ExteriorArea
Design Flow Rate	0	0	0	0	0	0
Flow per Zone Floor Area	0	0	0	0	0	0
Flow per Exterior Surface Area	0.36	0.000665	0.000665	0.000665	0.000665	0.000665
Air Changes per Hour	1	1	0	0	0	0
Constant Term Coefficient	1	1	1	1	1	1
Temperature Term Coefficient	0	0	0	0	0	0
Velocity Term Coefficient	0	0	0	0	0	0
Velocity Squared Term Coefficient	0	0	0	0	0	0

Explanation of Object and Comment Field
 Object Description: Infiltration is specified as a design level which is modified by a Schedule fraction, temperature difference and wind speed.
 Infiltration=design * FSchedule * (A + B*(Zone-Tobj) + C*Windspd + D * Windspd^2)
 If you use a ZoneList in the Zone or ZoneList name field then this definition applies to all the zones in the ZoneList.
 Field Description: "B" in Equation ID: N6

Figure 20 Example: Zone infiltration after rewriting

Zone Infiltration: This picture shows what the program actually writes for the external zones (here Perimeter_ZN_X). To be able to do that the program requires the name of these zones, as written in the IDF, here it was “Perimeter_ZN_”. It then iterates from 1 to X (X being the number of external zones) to describe the different zones infiltration.

7.3.3 - Windows ratio

For the simulation, a wall will need to be chosen for the window’s ratio. Every window belonging to this wall must be deleted. The program will create a new one.

When launching, the program will also ask for the construction object you want to be used (type of window) for this window, and the name of the wall you want for testing the windows ratio.

If not included, the price of the window must be added as a “ComponentCost:LineItem”.

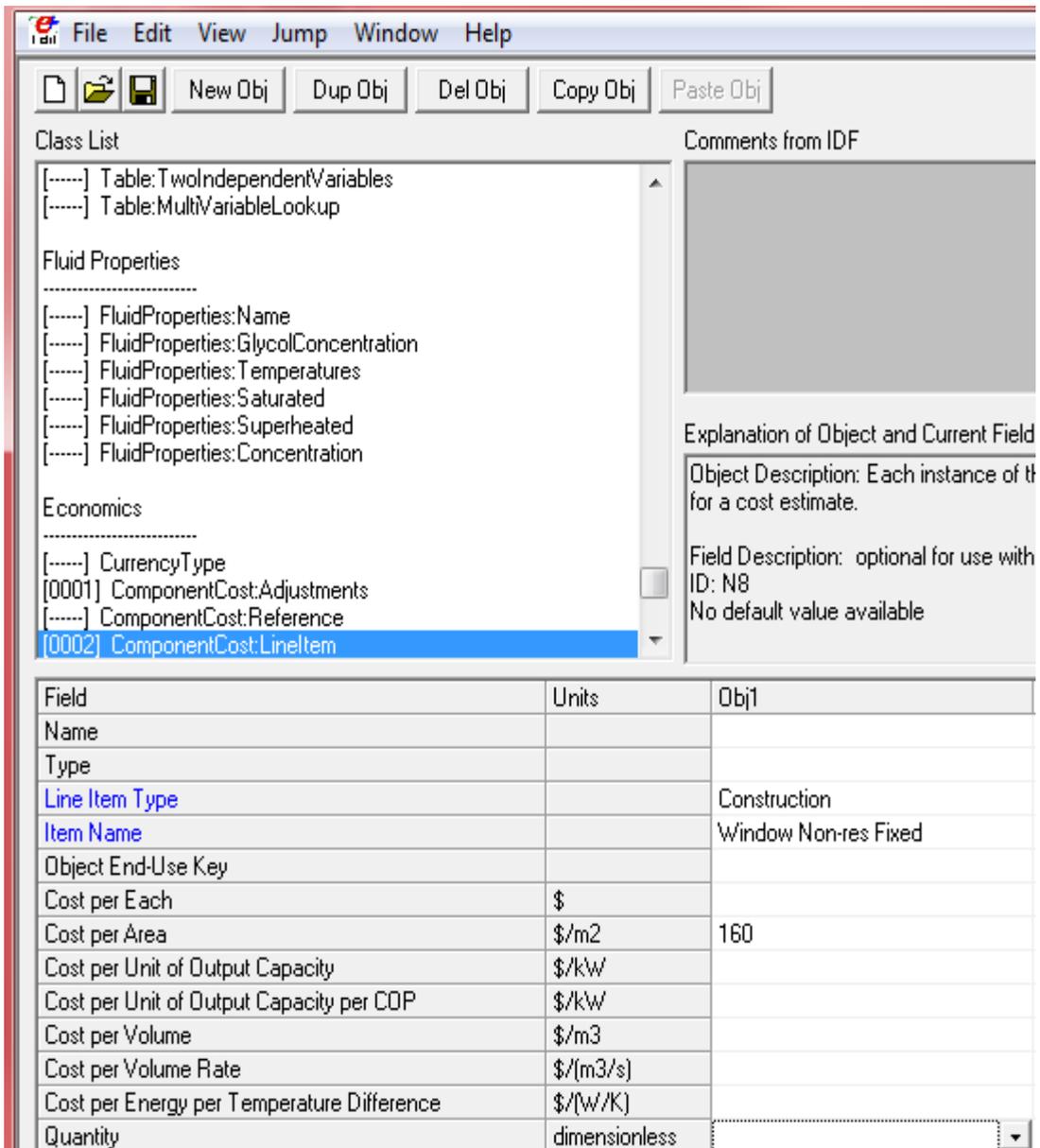


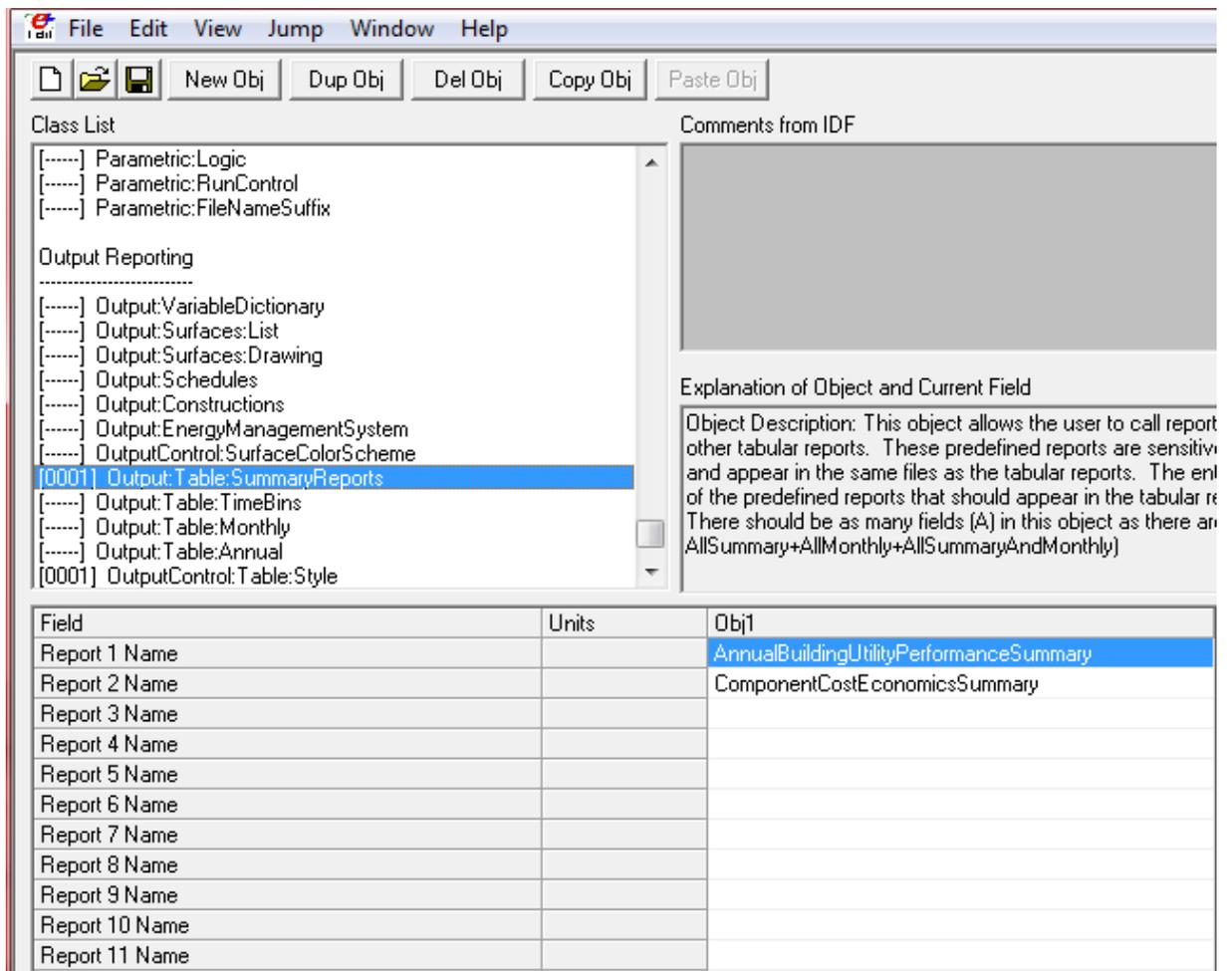
Figure 21 Example: setting the windows' construction price

Windows' price:

Here's an example of setting a window's price. It must be done in \$/m² for the reason that the window's ratio (and so window's surface) is different in each simulation. Construction object used here is "Window Non-res Fixed".

7.3.4 -Fix the outputs

Two outputs are needed to run the program. Outputs are chosen in the IDF. In the Output Reporting section: "AnnualBuildingUtilityPerformanceSummary" and "ComponentCostEconomicsSummary" must be selected in "the Output:Table:SummaryReports" object. Comma must be selected in the "OutputControl:Table:Style".



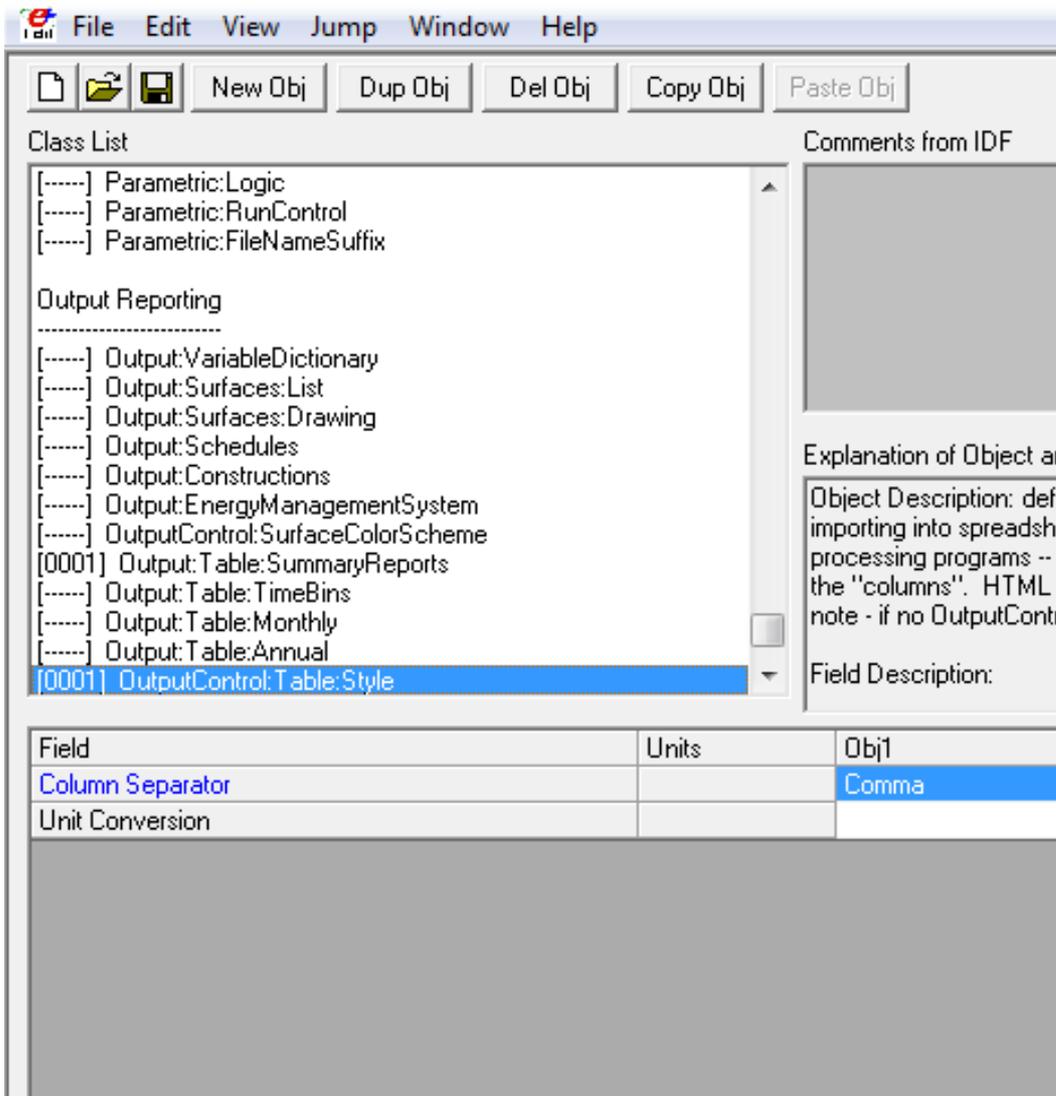


Figure 22 Example : fixing the outputs

Setting the outputs: These pictures show how to fix the needed outputs in IDF Editor. It's possible to ask for more outputs if you want more information on the final solution.

8 APPENDIX B: HOW TO RUN A SIMULATION

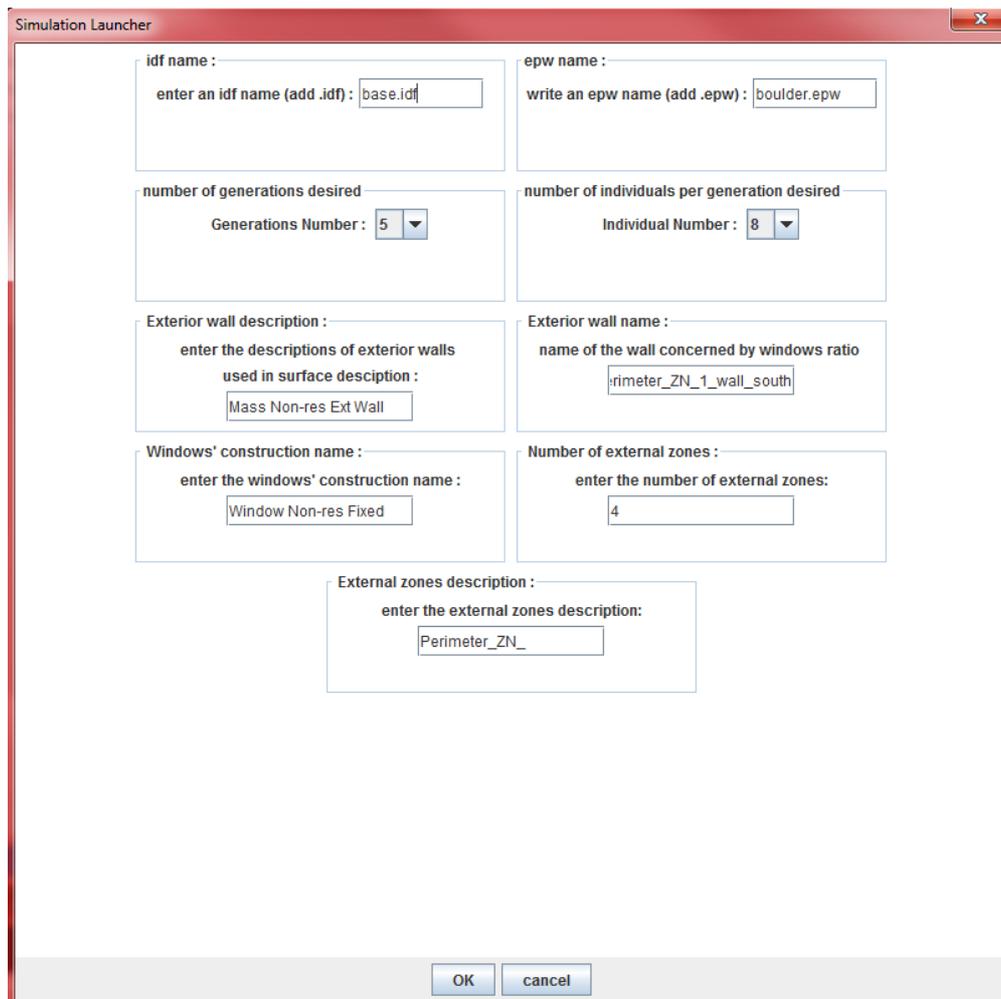
The simulation launcher dialog asks the user for many information. It would be impossible to understand how to run the simulation for someone that never used the program. This section aims to explain what is asked in the launcher and why.

8.1.1.1 Prepare your file

As long as the program just writes the variable part of the input data file (IDF), you will need to get an existing one and prepare it for the program. Everything about what needs to be done is explained in the *Get the "prepared IDF file"* document.

8.1.1.2 Simulation Launcher

When the IDF file is ready, it is time to open the Simulation Launcher dialog. This dialog window looks like this:



The Simulation Launcher dialog box contains the following fields and controls:

- idf name :** enter an idf name (add .idf) :
- epw name :** write an epw name (add .epw) :
- number of generations desired**
Generations Number :
- number of individuals per generation desired**
Individual Number :
- Exterior wall description :**
enter the descriptions of exterior walls
used in surface description :
- Exterior wall name :**
name of the wall concerned by windows ratio
- Windows' construction name :**
enter the windows' construction name :
- Number of external zones :**
enter the number of external zones:
- External zones description :**
enter the external zones description:

At the bottom of the dialog are **OK** and **cancel** buttons.

Figure 23 Simulation Launcher

Every field is really important and default parameter can help you understand what is needed.

8.1.1.3 IDF name

Here the user needs to write the name of the IDF he has prepared and wants to be run. It's that file that the program will copy and complete for every individual simulated.

This file needs to be **in the “Project” folder**. Default file is “base.idf”. When entering the name the “.idf” must be added.

8.1.1.4 EPW name

As for the IDF file, an EPW file needs to be selected. This file will be the weather file used in every simulation. The file needs to be **in the “simulation” folder of the “Project” folder**. Default file used is the “boulder.epw”. When entering the name the “.epw” must be added.

8.1.1.5 Generation number

Number of generation the genetic algorithm will go through. The higher this number is the longer and the more precise the simulation will be. This parameter must be chosen according to experience and to the number of parameters tested.

8.1.1.6 Individual Number

Number of individuals the genetic algorithm will go through every generation.

8.1.1.7 Exterior wall description

In the prepared IDF file, surfaces corresponding to the exterior walls are supposed to be all built of the same “construction”, the same way. What is asked here is the name of this construction object. The original one is deleted during the IDF preparation. Getting this, the program can rewrite the “construction” object selected according to the genes and give it the right name.

In the “base.idf” file every external surface corresponding to a wall uses the construction “Mass Non-res Ext Wall”.

8.1.1.8 Exterior wall name

Here is a completely different parameter because it concerns the windows ratio. When simulating, the program attributes the windows ratio selected to an only wall (a surface actually). In this field the user choose which wall.

In the default simulation, the south wall is selected. The corresponding surface is named “Perimeter_ZN_1_wall_south”

8.1.1.9 Window's construction name

The window's construction name is the name of the “construction” object used to describe the windows in the building; or at least the one that is going to be used for the new window.

In the default simulation, every window is made of the “Window Non-res Fixed” construction. It's what is used.

8.1.1.10 Number of external zones

Information about external zones is the more tricky to get.

In order to apply the infiltration rate to every zone in contact with the exterior, the program needs to know how many of it exists.

In the default case, 4 zones are linked to the exterior and are exposed to infiltration.

8.1.1.11 External zones description

When preparing the IDF file, the user makes sure to give every external zone the same name, and give them a number from 1 to X (X being the number of external zones). This allows the program to describe every single zone infiltration.

In the “base.idf” file, every external zone has the same name. There is four of it: Perimeter_ZN_1 , Perimeter_ZN_2, Perimeter_ZN_3 and Perimeter_ZN_4.

8.1.1.12 Simulation duration

According to the many simulations run, there is no rule to determine the time that a simulation takes. It might vary depending on the computer used and what is being done in parallel on this computer.

As an idea, simulations took in average 30 seconds per individual simulated. For 5 generations of 8 individuals, average time would be 20 minutes.

8.1.1.13 After clicking “ok“

No information on what is happening in the program during the simulation is available for the moment. The launcher window just freezes during the simulation. Once completed, a result text file must appear. Patience.

9 APPENDIX C: CALCULATIONS FOR WINDOWS RATIO

One of the characteristics that the program makes vary is the windows ratio of a selected wall. EnergyPlus doesn't include any notion of windows ratio so that we had to create it. This appendix explains how it is considered in the program.

When about to start, the program asks which wall is going to be considered and what are its dimensions. This is all for considering windows ratio.

From the wall's dimensions and the ratio asked, the program creates a unique window of fixed height ($h= 1,5$ m) and variable length (l). This information allows us to create a window object by calculating l but also X and Y (coordinates of the bottom left corner required in EnergyPlus) to make sure that the window is centered. Following figure illustrates the calculus.

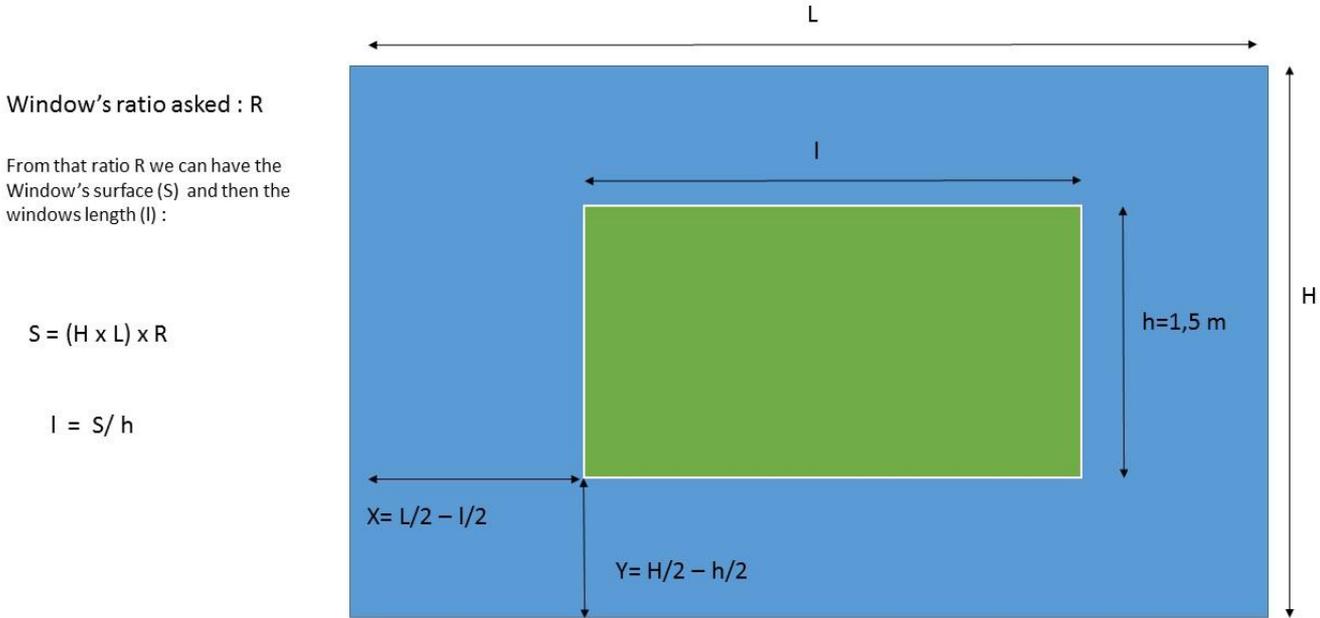


Figure 24 calculation for windows ratio

Limits:

This way of considering window's ratio is for sure limited. It requires a lot of information from the user and considers a unique window, which has a direct impact on shading or solar gains calculations for example.