



Optimisation sous contrainte d'un générateur thermoélectrique pour la récupération de chaleur par différents algorithmes heuristiques

Mémoire

Mathieu Allyson-Cyr

Maîtrise en génie mécanique - avec mémoire
Maître ès sciences (M. Sc.)

Québec, Canada

Optimisation sous contrainte d'un générateur thermoélectrique pour la récupération de chaleur par différents algorithmes heuristiques

Mémoire

Mathieu Allyson-Cyr

Sous la direction de :

Louis Gosselin, directeur de recherche

Résumé

La présente étude porte sur le développement et l'optimisation d'un modèle de générateur thermoélectrique placé sur la surface d'une source de chaleur. La particularité de ce modèle est que la source de chaleur est sujette à un flux de chaleur et à une température de surface fixes. L'objectif principal est de développer un modèle de générateur thermoélectrique d'intérêt dans ce contexte particulier qui pourra s'adapter à différentes sources de chaleur et qui pourra inclure différents systèmes de refroidissement. Le modèle a été créé intégralement à l'aide du logiciel Matlab. Un algorithme génétique multiobjectif est ensuite utilisé comme outil d'optimisation afin de maximiser les performances tout en minimisant les coûts du générateur thermoélectrique. Les objectifs d'optimisation proposés sont donc de maximiser la puissance électrique et de minimiser le nombre de modules. Lorsqu'un collecteur thermique est inclus au système, il est aussi nécessaire de minimiser la puissance de pompage et l'aire totale d'échange du collecteur. Une première étude considère uniquement la puissance comme objectif d'optimisation afin d'observer l'impact des contraintes de température et de flux de chaleur de la source sur les designs optimaux. Des cas multiobjectifs seront ensuite étudiés avec les différents objectifs énoncés. Finalement, les performances de différents algorithmes d'optimisation heuristiques seront comparées entre eux en utilisant le modèle thermoélectrique développé comme banc d'essai. Les forces et faiblesses de chaque algorithme seront analysées selon divers critères de performance, lorsqu'appliqués à un cas d'optimisation complexe.

Abstract

This study presents a model of a thermoelectric generator placed directly on the surface of a heat source. One unique feature of this model is that the heat source is subject to fixed heat flux and surface temperature that the system must respect. The main objective is to develop this model in this particular context with the possibility to be adapted to any heat source and the option to add a cooling system. The model has been developed entirely on the software Matlab. Then, a genetic algorithm is used to perform an optimisation in order to find the design with the maximal power output and minimal number of thermoelectric modules. With the cooling system included, the total surface of exchange and pumping power is also considered. A preliminary analysis is conducted to analyse the impact of the heat flux and surface temperature constraint on such system. Thereafter, a multi-objective optimisation is performed to find the optimal design considering multiple optimisation objectives. Finally, different heuristic algorithms are compared for solving the thermoelectric model proposed. The performance is discussed using different performance criteria to show the pros and cons of each heuristic algorithm when solving a complex optimisation design problem.

Table des matières

| | |
|--|------|
| Résumé | iii |
| Abstract | iv |
| Table des matières | v |
| Liste des tableaux | vii |
| Liste des figures | viii |
| Nomenclature | x |
| Remerciement | xii |
| Avant-propos | xiii |
| Chapitre 1 : Introduction..... | 1 |
| Problématique | 1 |
| Objectifs | 2 |
| Méthodologie | 3 |
| Chapitre 2 : Thermoélectricité et générateur thermoélectrique | 6 |
| 2.1 Phénomènes thermoélectriques | 6 |
| 2.2 Modélisation d'un module thermoélectrique | 8 |
| 2.3 Générateur thermoélectrique..... | 10 |
| Chapitre 3 : Algorithmes d'optimisation heuristiques | 14 |
| 3.1. Initialisation d'un algorithme heuristique..... | 14 |
| 3.2. Algorithme génétique (GA) | 15 |
| 3.3. Colonie de fourmis avec variables continues (ACO _R)..... | 20 |
| 3.4. Essaim particulaire (PSO) | 22 |
| 3.5. Évolution différentielle (DE) | 25 |
| 3.6. Algorithme «Enseignement-apprentissage» (TLBO) | 28 |
| Chapitre 4 : Article #1 | 31 |
| Résumé | 32 |
| Abstract | 33 |
| 4.1. Introduction..... | 34 |
| 4.2. Thermoelectric 1D model | 36 |
| 4.3. Formulation of optimization problem for 2D TEG model..... | 43 |
| 4.4. Optimized TEG design for maximal global efficiency | 47 |
| 4.5. Formulation of optimization problem for 2D TEG with heat collector..... | 53 |
| 4.6. Optimal TEG solutions with heat collector | 59 |
| 4.7. Conclusion..... | 63 |
| Chapitre 5 : Article #2 | 65 |

| | |
|---|----|
| Résumé | 66 |
| Abstract | 67 |
| 5.1. Introduction..... | 68 |
| 5.2. Heuristic algorithms | 69 |
| 5.3. Test case: Thermoelectric generator design optimization | 75 |
| 5.4. Analysis of heuristic algorithm parameters effect on performance | 78 |
| 5.5. Impact of heuristics on sets of non-dominated solutions | 89 |
| 5.6. Conclusion..... | 94 |
| Chapitre 6 : Conclusion..... | 96 |
| Bibliographie | 99 |

Liste des tableaux

| | |
|--|----|
| Table 4.1: Characteristics and properties of the model | 38 |
| Table 4.2: Temperature dependant properties of selected materials | 46 |
| Table 4.3: Main parameters of the single-objective GA..... | 47 |
| Table 5.1: Discrete and continuous decision variables of the TEG model (with 12×12 grid), based on Ref. [13] | 77 |
| Table 5.2: List and range of algorithm parameters..... | 79 |
| Table 5.3: Best parameters for nearly optimal net power output..... | 87 |
| Table 5.4: Normalized mean metric value C_μ , normalized entropy H_n and percentage of points that belongs to the approximate Pareto front P^* | 94 |

Liste des figures

| | |
|--|----|
| Figure 2.1 : Schéma d'un thermocouple. | 6 |
| Figure 2.2 : Représentation schématique de l'effet Peltier et Thomson dans un thermocouple. | 7 |
| Figure 2.3 : Module thermoélectrique. | 9 |
| Figure 2.4 : Générateur thermoélectrique..... | 10 |
| Figure 2.5 : Schéma du modèle thermoélectrique de base en circuit fermé. | 12 |
| Figure 3.1 : Génération d'une population de 100 individus par une méthode d'échantillonnage aléatoire, LHS et FSS..... | 15 |
| Figure 3.2 : Structure de l'algorithme génétique..... | 17 |
| Figure 3.3 : Méthode de sélection par jeu de roulette..... | 18 |
| Figure 3.4 : Méthode de sélection par tournoi..... | 18 |
| Figure 3.5 : Croisement par dispersion..... | 19 |
| Figure 3.6 : Structure de l'algorithme ACO _R | 22 |
| Figure 3.7 : Exemple du déplacement d'une particule $x_{i,j}$ vers la nouvelle position $x_{i,j+1}$ pour une fonction objective à deux dimensions. | 24 |
| Figure 3.8 : Structure de l'algorithme PSO..... | 25 |
| Figure 3.10 : Exemple de la mutation pour une fonction objective à deux dimensions..... | 27 |
| Figure 3.11 : Exemple du croisement pour une fonction objective à 5 dimensions. | 28 |
| Figure 3.12 : Structure de l'algorithme TLBO..... | 29 |
| Figure 4.1 : Schematic representation of a) single thermoelectric module and b) TEG disposition with respect to the heat source. | 37 |
| Figure 4.2 : Thermoelectric performance with no imposed constraint on the hot side heat flux: a) Electric power output, b) thermoelectric efficiency and c) Hot side heat flux as a function of the number of modules and electric current..... | 41 |
| Figure 4.3 : a) Maximal electric power obtainable and b) corresponding optimal combinaison of number of modules and current for a constrained value of the hot side heat flux..... | 42 |
| Figure 4.4 : a) Maximized power output and b) required number of modules for fixed hot side heat flux and temperature as a function of the cold side convection coefficient and far field temperature..... | 43 |
| Figure 4.5 : 2D representation of the TEG model with electric connection layout example..... | 44 |
| Figure 4.6 : Total set of predefined TEG electric connection grid layout generated for 12 × 12 mesh..... | 45 |
| Figure 4.7 : Hot-side temperature and heat flux distributions for: (a) the 1st test case, and (b) the 2nd test case..... | 48 |
| Figure 4.8 : Optimization results with a 5 × 5 mesh (left) and a 12 × 12 mesh (right) for the first test case a) TEG global efficiency, b) density of modules, c) material distribution..... | 50 |
| Figure 4.9 : Optimization results with a 5 × 5 mesh (left) and a 12 × 12 mesh (right) for the second test case for a) TEG global efficiency, b) density of modules, c) material distribution..... | 51 |
| Figure 4.10 : Pareto front of the TEG solutions (12 × 12 mesh) with minimized number of modules and maximized power output for a) 1 st test case and b) 2 nd test case..... | 53 |

| | |
|---|----|
| Figure 4.11: 2D representation of combined TEG and heat collector model. | 54 |
| Figure 4.12: 2D representation of the coolant flow patterns in the heat collector with 12×12 mesh for grid #6 (Fig. 4.6) set in a) horizontal pattern and b) vertical pattern. | 55 |
| Figure 4.13: Heat collector duct and fin geometry. | 56 |
| Figure 4.14: Single-objective GA optimization results of 2D TEG model including the cooling system with a 12×12 mesh with heat transfer distribution #2: a) TEG efficiency, b) net efficiency, c) material distribution. | 60 |
| Figure 4.15: Pareto front of the multi-objective GA optimization results in plane view of objectives A_{tot} and n_{tot} with a) net power output color scaling, b) 3D view with net power output z-axis, c) optimal TEG grid layout scaling and d) optimal HE grid layout scaling. | 63 |
| Figure 5.1: Schematic view of the TEG system sandwiched between the heat source and a heat collector. ... | 76 |
| Figure 5.2: Constrained temperature and heat flux distributions over the heat source surface. | 77 |
| Figure 5.3: Impact of GA parameters on a) maximal net power, b) RSD and c) lte | 81 |
| Figure 5.4: Impact of PSO parameters on a) maximal net power, b) RSD and c) lte | 82 |
| Figure 5.5: Impact of $ACOR$ parameters on a) maximal net power, b) RSD and c) lte | 83 |
| Figure 5.6: Impact of DE parameters on a) maximal net power, b) RSD and c) lte | 84 |
| Figure 5.7: Effect of the algorithm parameters on a) RSD and b) the required number of iterations using GA, PSO, ACOR and DE for all the settings tested. | 86 |
| Figure 5.8: Convergence comparison of the net power optimization with algorithm parameters of Table 5.3. The lines are the mean value and the gray areas are the min-max range. | 87 |
| Figure 5.9: Summary of the heuristic algorithm performances for the problem tested. | 89 |
| Figure 5.10: Approximate Pareto front P^* after the non-dominated sorting obtained with the sum-weight method with all solutions from GA, PSO, ACOR, TLBO and DE combined. | 90 |
| Figure 5.11: Origin of the non-dominated solutions achieved by all algorithms. | 90 |

Nomenclature

| | |
|--------------|--|
| A | Aire, m ² |
| A_{pellet} | Aire transversal d'une patte thermoélectrique, m ² |
| C | Débit capacitif, W / K |
| C_p | Capacité calorifique, J / kg K |
| D_h | Diamètre hydraulique, m |
| f | Facteur de friction |
| h | Coefficient de convection, W / m ² K |
| I | Courant électrique, A |
| k | Conductivité thermique, W / m K |
| K | Conductance thermique, W / K |
| L | Longueur, m |
| \dot{m} | Débit massique, kg / s |
| n | Nombre de modules thermoélectriques |
| n'' | Densité de modules thermoélectriques, m ⁻² |
| P | Puissance électrique, W |
| P'' | Densité de puissance électrique, W / m ² |
| Pr | Nombre de Prandtl |
| q | Taux de transfert de chaleur, W |
| q'' | Flux de chaleur, W / m ² |
| R | Résistance électrique, Ω |
| R_{ext} | Résistance électrique externe du générateur thermoélectrique, Ω |
| Re | Nombre de Reynolds |
| S | Coefficient de Seebeck, V / K |
| St | Nombre de Stanton |
| T | Température, K |
| U | Coefficient de transfert global, W / m ² K |
| W_p | Puissance de pompage, W |
| ZT | Figure de mérite adimensionnelle |

Symboles grecs

| | |
|----------|---|
| β | Facteur de mutation (DE) |
| η | Efficacité |
| μ | Moyenne |
| Π | Coefficient de Peltier, V |
| ρ | Résistivité électrique, Ωm |
| ρ_v | Masse volumique, kg / m ³ |
| σ | Écart type |
| τ | Coefficient de Thomson, V / K |
| ζ | Taux d'évaporation de phéromone (ACO _R) |

Indices

| | |
|---------------|---|
| <i>C</i> | Relatif au côté froid du générateur thermoélectrique |
| <i>global</i> | Valeur globale |
| <i>H</i> | Relatif au côté chaud du générateur thermoélectrique |
| <i>HE</i> | Relatif à l'échangeur de chaleur |
| <i>JC</i> | Relatif à la jonction froide du générateur thermoélectrique |
| <i>JH</i> | Relatif à la jonction chaude du générateur thermoélectrique |
| <i>max</i> | Valeur maximale |
| <i>n</i> | Relatif au matériau thermoélectrique de type n |
| <i>net</i> | Valeur nette |
| <i>opt</i> | Valeur optimale |
| <i>p</i> | Relatif au matériau thermoélectrique de type p |

Acronymes

| | |
|------------------------|---|
| <i>ACO_R</i> | « Ant colony optimization (continuous domain) » |
| <i>DE</i> | « Differential evolution » |
| <i>FSS</i> | « Faure Sequence sampling » |
| <i>GA</i> | « Genetic algorithm » |
| <i>HE</i> | Échangeur de chaleur |
| <i>LHS</i> | « Latin Hypercube sampling » |
| <i>PSO</i> | « Particle swarm optimization » |
| <i>RSD</i> | Écart type relatif |
| <i>TE</i> | « Thermoelectric » |
| <i>TEC</i> | « Thermoelectric cooler » |
| <i>TEG</i> | « Thermoelectric generator » |
| <i>TLBO</i> | « Teaching-learning basic optimization » |

Remerciement

Tout d'abord, je tiens à remercier infiniment Louis Gosselin, qui a été mon directeur de recherche, pour son appui continuuel et sa grande disponibilité pendant toute ma maîtrise. Je le remercie grandement pour l'opportunité qu'il m'a offerte et l'intérêt qu'il m'a démontré dans la poursuite de mes études. Louis a été d'une aide précieuse en me transmettant ses conseils et ses méthodes de travail qui m'ont aidé tout au long du projet dans la réalisation du mémoire et des deux articles scientifiques. Cela est sans compter que ces nouvelles compétences me seront fort utiles dans ma vie professionnelle. Je le remercie pour son ouverture d'esprit aux idées que j'apportais et la liberté que j'avais à les développer. Je n'aurais pas espéré un meilleur environnement de travail et un meilleur directeur de recherche.

Je remercie également tous mes collègues du laboratoire de Transfert thermique et d'Énergétique pour leur soutien et leur accessibilité dans les moments difficiles et leurs conseils qui se sont avérés très utiles tout au long de mon parcours.

Finalement, je tiens à remercier ma mère, Carole Cyr, qui m'a soutenu moralement et qui m'a donné la force d'avancer. Son appui continuuel a fait naître en moi une confiance qui m'a permis de poursuivre ma maîtrise jusqu'au bout.

Avant-propos

J'ai été l'auteur principal des deux articles qui sont présentés dans ce mémoire. Le premier article a été soumis dans le journal « Applied Thermal Engineering » en mai 2018 et le second article a été soumis dans le journal « Engineering Optimization » en août 2018. J'ai effectué l'ensemble des calculs et des tests numériques nécessaires pour l'obtention des résultats qui sont présentés. Les articles ont été rédigés avec la collaboration de mon directeur de recherche Louis Gosselin qui a apporté sa contribution et son savoir.

Des changements mineurs ont été apportés aux deux articles (présentés aux chapitres 4 et 5) pour satisfaire les exigences de mise en forme de la FESP.

Chapitre 1 : Introduction

Problématique

L'utilisation responsable et efficace de l'énergie au Québec fait partie intégrante des stratégies gouvernementales pour répondre aux enjeux énergétiques actuels. Dans cette optique, le gouvernement du Québec a fixé des objectifs ambitieux pour augmenter la production d'énergie renouvelable, réduire les émissions de gaz à effet de serre, réduire la consommation de produits pétroliers et opter pour un usage plus efficace de l'énergie. Pour atteindre ces cibles, une grande importance est accordée à l'innovation énergétique. Conséquemment, de nouvelles technologies émergentes plus efficaces sont développées pour répondre à la demande en énergie actuelle dans divers secteurs d'activité (transport, commercial, industriel, résidentiel).

L'un des secteurs avec la plus grande consommation énergétique est le secteur industriel qui est responsable pour près de 37% de la consommation énergétique au Québec. Dans ce secteur, de nombreux procédés génèrent des pertes énergétiques sous forme de chaleur évacuées dans l'environnement. Lorsque la source de chaleur est pérenne et que la qualité de la source thermique est élevée, il devient intéressant de récupérer et de réutiliser cette énergie afin d'améliorer l'efficacité des nombreux procédés et de réduire la consommation énergétique.

Plusieurs technologies existent actuellement pour répondre à cette problématique. Par exemple, il est possible d'inclure un système de réseaux de chaleur composé de plusieurs conduites pour collecter la chaleur de ces différents procédés via un fluide caloporteur pour ensuite la réutiliser pour le chauffage des installations. Toutefois, une telle technologie demande beaucoup de ressources et il n'est pas toujours aisé de l'implémenter en pratique. Néanmoins, avec l'amélioration et le développement de nouvelles technologies, de nouvelles avenues potentielles pour la récupération énergétique s'offre à nous. La thermoélectricité est l'une de ces technologies qui est actuellement en plein essor. La thermoélectricité peut être utilisée pour convertir directement la chaleur en électricité et vice-versa via plusieurs phénomènes thermoélectriques. Cette technologie peut donc être utilisée comme générateur thermoélectrique (TEG) ou bien comme pompe à chaleur thermoélectrique. Bien que la thermoélectricité soit un phénomène connue depuis longtemps, son intérêt pour des applications énergétiques n'est que très récent. Cela est principalement dû à la faible efficacité de conversion des systèmes thermoélectriques actuels ce qui est souvent désavantageux économiquement. Néanmoins, le développement de nouveaux matériaux thermoélectriques plus performants et l'amélioration des designs thermoélectriques a généré un intérêt grandissant de ces systèmes pour des

applications industrielles. Malgré tout, les systèmes thermoélectriques offrent plusieurs avantages intéressants en pratique. En effet, un appareil thermoélectrique ne requiert aucune pièce mobile et aucun réfrigérant. De plus, un générateur thermoélectrique ne prend pas beaucoup d'espace. Par conséquent, un appareil thermoélectrique est réputé pour sa robustesse, sa durabilité et de ses dimensions permettant d'être installé facilement dans des endroits restreints.

De nos jours, les ingénieurs se retrouvent confrontés à des problèmes de plus en plus complexes dans de nombreux domaines. La conception de générateurs thermoélectriques fait partie de ce groupe de problèmes complexes. La majorité de ces problèmes consistent en l'optimisation d'un ou de plusieurs objectifs en fonction d'un ensemble de variables. Ces problèmes d'optimisation sont souvent non linéaires et peuvent faire intervenir des variables discrètes et continues. Dans certains cas, les problèmes sont non différentiables et il devient difficile d'utiliser certaines méthodes non linéaires connues. C'est avec l'arrivée de ce type de problèmes que les chercheurs se sont penchés sur le développement de méthodes alternatives de résolution. Une de ces solutions fait intervenir des méthodes de recherche directe et sans dérivée faisant appel à de nouveaux algorithmes communément appelés algorithmes heuristiques. Ces algorithmes utilisent des méthodes stochastiques et probabilistes pour obtenir une solution approximative, mais satisfaisante, par rapport à la solution optimale recherchée. Conséquemment, plusieurs optimisations du même problème peuvent donner des résultats différents. De plus, ces algorithmes sont indépendants du problème à résoudre et s'adaptent donc à un large éventail de problèmes complexes. La performance des algorithmes heuristiques dépend largement du type de problème ainsi que des paramètres d'optimisation unique à chaque algorithme. Selon la présence plus ou moins grande de minimums locaux, la non-linéarité du modèle, le nombre de variables et le type de variables (continue, discrète, mixte); il peut être avantageux d'utiliser un algorithme heuristique en particulier pour obtenir de bonnes solutions avec un temps de résolution plus faible. C'est pourquoi les algorithmes heuristiques feront également l'objet de ce mémoire.

Objectifs

L'étude qui suit portera sur deux éléments distincts. La première étude se concentrera sur la création d'un modèle de générateur thermoélectrique pour la récupération énergétique au niveau de surfaces chaudes dont le flux de chaleur et la température doivent être contrôlés adéquatement. Les différents objectifs qui ont été fixés au départ sont les suivants :

- Concevoir plusieurs modèles numériques pour le générateur thermoélectrique sur Matlab. Ces modèles devront inclure différents modes de refroidissement, soit par convection avec l'air ambiant, soit à l'aide d'un système de refroidissement. Ces modèles devront être généraux afin de s'appliquer à différents types de surface dont les contraintes de flux thermique et de température peuvent varier.

- Effectuer l'optimisation du système en considérant différentes connexions électriques et différents matériaux thermoélectriques. Le modèle devra être en mesure de considérer des variations de température et de flux de chaleur sur la surface de la source de chaleur.
- Effectuer l'optimisation des différents objectifs du modèle. Ces objectifs pourront être considérés seuls ou en groupe et devront refléter la maximisation de la performance thermoélectrique et la minimisation des coûts du système.

La seconde étude portera sur l'exploration et l'analyse de performance de différents algorithmes heuristiques. L'un des modèles développés dans la première étude servira de banc d'essai pour l'analyse des différents algorithmes heuristiques. Les objectifs qui ont été fixés initialement sont les suivants :

- Implémenter numériquement différents algorithmes heuristiques sur Matlab adaptés au modèle de générateur thermoélectrique de la première étude.
- Effectuer l'optimisation du système à l'aide des différents algorithmes heuristiques. Les différents objectifs à optimiser pourront être groupés ou être considérés indépendamment. Les différences au niveau de la vitesse de convergence, la précision des résultats et la fiabilité entre les algorithmes proposés seront analysées et commentées.
- Analyser l'impact des réglages propres à chaque algorithme sur différents critères de performance. Les résultats seront analysés et commentés.

Méthodologie

La méthodologie présentée dans cette section permettra d'atteindre les différents objectifs fixés précédemment. Le logiciel Matlab sera utilisé pour développer les différents modèles mathématiques pour le générateur thermoélectrique. Les modèles devront être en mesure de déterminer la puissance électrique générée par le système. Un premier modèle simplifié unidimensionnel sera d'abord développé avec une température et un flux de chaleur constant. Le modèle sera formé d'une surface chaude dont le générateur thermoélectrique sera directement appliqué dessus. La jonction froide du générateur thermoélectrique sera refroidie par l'air ambiant. Pour simplifier davantage ce modèle, les mêmes matériaux pour tous les modules thermoélectriques sont considérés dont les connexions électriques seront entièrement en série. Le modèle devra être en mesure de calculer la puissance électrique en sortie ainsi que l'efficacité thermoélectrique en fonction de plusieurs paramètres du modèle. Plusieurs variables de designs seront considérées dans ce premier modèle numérique : la température de la jonction chaude et froide du générateur, le flux de chaleur fixe de la source, le courant, le nombre de modules totaux, le ratio géométrique de l'aire transversale sur la longueur des modules ainsi que le coefficient de convection à la jonction froide. Puisqu'aucune étude n'a précédemment été portée spécifiquement sur le comportement d'un générateur thermoélectrique avec les

particularités détaillées précédemment, une analyse paramétrique sera adoptée pour bien comprendre l'influence des différentes variables de design sur la puissance électrique. Il sera donc possible d'analyser et de commenter les limitations de telles considérations sur les performances d'un générateur thermoélectrique.

Ensuite, afin de généraliser davantage le modèle précédent, la température et le flux de chaleur sur la surface seront variables. Un deuxième modèle sera développé pour s'adapter à plus de cas concrets dont le transfert thermique est généralement inégalement réparti sur la surface. Afin de considérer ce nouvel aspect, la surface chaude sera découpée en plusieurs volumes de contrôle auxquels seront appliqués des sous-générateurs thermoélectriques. Pour chaque volume de contrôle, la température et le flux de chaleur seront moyennés. Pour chaque sous-générateur thermoélectrique, les matériaux thermoélectriques utilisés seront les mêmes et les connexions entre les modules seront toutes en série. Néanmoins, il sera possible d'ajuster les connexions électriques entre les différents sous-générateurs thermoélectriques et de choisir différents matériaux pour chaque volume de contrôle. Ces nouvelles variables de design viennent s'ajouter à ce deuxième modèle plus complexe. Pour résoudre ce deuxième modèle, un algorithme génétique sera utilisé comme outil d'optimisation pour maximiser la puissance thermoélectrique. L'avantage de l'algorithme génétique est sa capacité à résoudre rapidement et efficacement des problèmes complexes qui peuvent comprendre plusieurs minimums locaux. Les designs optimaux pour différentes distributions de température et de flux de chaleur déterminées arbitrairement seront analysés et commentés.

Un troisième modèle sera également développé pour considérer l'intégration d'un système de refroidissement à la jonction froide du générateur thermoélectrique. Le système de refroidissement sera un collecteur thermique dont le fluide sera de l'air. Ce nouveau système comportera plusieurs variables de design additionnelles telles que la géométrie, le débit capacitif, la température d'entrée ainsi que le coefficient de transfert global. Avec l'ajout du collecteur thermique au modèle, il sera nécessaire de calculer la puissance de pompage reliée au système qui sera soustraite à la puissance électrique en sortie pour ne retenir que la puissance nette. Un algorithme génétique sera également utilisé pour optimiser la puissance nette du modèle. En considérant les mêmes distributions de température et de flux de chaleur que pour le deuxième modèle, il sera possible de comparer les designs optimaux du modèle sans et avec collecteur thermique et d'en tirer des conclusions sur les performances et les coûts d'un tel système.

Une analyse thermoéconomique du système sera finalement effectuée via une optimisation multiobjectif du modèle final comprenant le système de refroidissement additionnel. Jusqu'à présent, l'impact de l'amélioration d'un objectif sur d'autres objectifs de nature économique a été négligé. Conséquemment, deux objectifs additionnels sont donc considérés pour un total de trois objectifs avec la puissance nette. Ces objectifs additionnels sont le nombre de modules totaux et l'air total de l'échangeur. Ces objectifs serviront à

approximer les coûts du système sans avoir à déterminer directement les coûts de chaque composante du modèle. Une optimisation multiobjectif sera effectuée à l'aide d'un algorithme génétique et le front de Pareto résultant sera généré et porté graphiquement. Une analyse des objectifs suivra ensuite et des conclusions seront finalement portées sur le modèle. Ceci résume ce qui est présenté dans le premier article qui se retrouve au chapitre 4.

Une deuxième partie du mémoire portera sur l'analyse et la comparaison de différents algorithmes heuristiques comme l'algorithme génétique. Les algorithmes heuristiques sont des outils intéressants et de plus en plus utilisés pour leur facilité d'implémentation, leur capacité à résoudre rapidement des problèmes complexes et leur capacité à trouver un optimum sans avoir à parcourir tous les designs existants d'un modèle. Un total de 5 algorithmes heuristiques mono-objectif seront retenus pour cette analyse et seront implémentés numériquement sur Matlab. Le modèle thermoélectrique présenté au chapitre 4 servira de banc d'essai pour la comparaison des algorithmes. Ces algorithmes ont tous des fonctionnements différents et des réglages qui influencent les mécanismes de convergence. Ces réglages sont régis par des paramètres d'algorithme qui peuvent être choisis par l'utilisateur pour améliorer les performances pour un problème donné. Une première analyse portera donc sur l'analyse de l'impact des paramètres d'algorithme sur la puissance nette du modèle. Cette analyse paramétrique permettra de déterminer la robustesse, la vitesse de convergence ainsi que la précision de ces algorithmes. Cela permettra ultimement de sélectionner les réglages appropriés pour obtenir les meilleures performances pour l'optimisation du modèle thermoélectrique proposé.

Ensuite, une optimisation multiobjectif sera effectuée en choisissant les paramètres d'algorithmes optimaux. Puisque les algorithmes heuristiques seront mono-objectifs, une méthode d'optimisation par poids pondérés sera employée. Une approximation du front de Pareto optimal sera ensuite générée en effectuant un triage des solutions non dominées à partir de l'ensemble des algorithmes. Subséquemment, chaque algorithme sera comparé à ce front de Pareto approximatif selon des métriques de convergence et de diversité. Les résultats obtenus seront analysés, commentés et comparés avec des analyses similaires de la littérature. Ceci résume ce qui est présenté dans le second article au chapitre 5.

Le prochain chapitre constituera une introduction à la thermoélectricité et aux générateurs thermoélectriques. Le fonctionnement ainsi que les équations gouvernantes d'un tel système seront abordés. Le chapitre 3 présentera le fonctionnement général des différents algorithmes heuristiques présentés dans le second article au chapitre 5. Cela permettra au lecteur d'avoir une meilleure compréhension sur les mécanismes stochastiques et de recherche directe que ces algorithmes utilisent pour résoudre un grand éventail de problèmes.

Chapitre 2 : Thermoélectricité et générateur thermoélectrique

Ce chapitre permettra au lecteur d'avoir une meilleure compréhension du fonctionnement des phénomènes thermoélectriques et des appareils thermoélectriques servant à la génération d'électricité. Des notions de base sur la thermoélectricité seront présentées ainsi que les équations gouvernantes régissant le phénomène.

2.1 Phénomènes thermoélectriques

Afin de bien comprendre le fonctionnement d'un appareil thermoélectrique, il est primordial d'aborder les différents phénomènes thermoélectriques qui sont à la base de la conversion thermoélectrique. Les effets thermoélectriques consistent en l'effet Seebeck, l'effet Peltier et l'effet Thomson.

2.1.1. Effet Seebeck

L'effet Seebeck a été découvert par Thomas J. Seebeck en 1821. Il a été observé qu'une force électromotrice est produite lorsqu'un circuit formé de deux matériaux conducteurs dissimilaires est chauffé à l'une des jonctions (similaire à un thermocouple). Le schéma d'un thermocouple présenté à la Figure 2.1 permet de mieux comprendre le fonctionnement de l'effet Seebeck. Prenons deux matériaux (a et b) connectés thermiquement en parallèle et électriquement en série. Si on applique une différence de température aux jonctions ($T_{JH} - T_{JC}$) et que l'on installe un voltmètre au sein du circuit sur le conducteur b (inséré entre les points 1 et 2), un voltage sera mesuré.

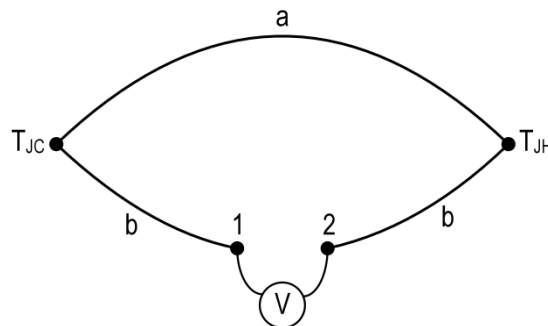


Figure 2.1 : Schéma d'un thermocouple.

Il a ensuite été observé qu'il existait une relation de proportionnalité entre la différence de température aux jonctions et le voltage calculée de la façon suivante:

$$V = S_{ab} (T_{JH} - T_{JC}) \quad (2.1)$$

où S_{ab} est le coefficient Seebeck défini par $S_a - S_b$ mesuré en V/K. L'effet Seebeck est principalement exploité dans les générateurs thermoélectriques pour la conversion de chaleur en électricité.

2.1.2. Effet Peltier

L'effet Peltier est l'effet inverse de l'effet Seebeck qui a été observé en 1834 par Jean-Charles Peltier. Une production de chauffage ou du refroidissement est observé à l'une des jonctions lorsqu'un courant est appliqué au thermocouple. Il est à noter que le chauffage ou le refroidissement dû à l'effet Peltier est réversible, mais celui-ci s'accompagne toujours de l'effet joule qui est irréversible. La Figure 2.2 représente schématiquement l'effet Peltier. Le taux de chaleur absorbé ou dégagé $q_{Peltier}$ aux jonctions est dicté selon la direction du courant I .

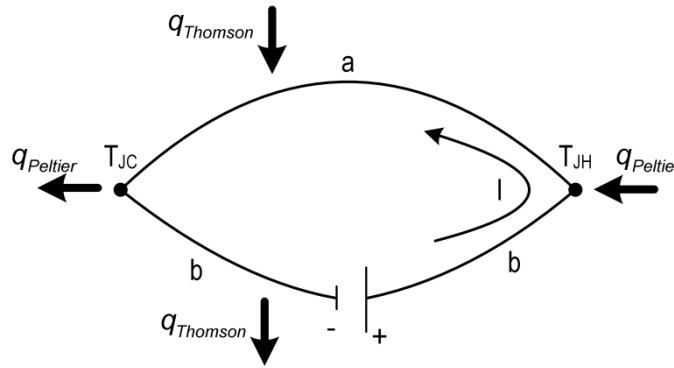


Figure 2.2 : Représentation schématique de l'effet Peltier et Thomson dans un thermocouple.

Une relation de proportionnalité est observable entre le courant appliqué et le taux de chaleur absorbé ou dégagé qui se calcule de la manière suivante :

$$q_{Peltier} = \Pi_{ab} I \quad (2.2)$$

où Π_{ab} est le coefficient Peltier défini par $\Pi_a - \Pi_b$ mesuré en V. L'effet Peltier est principalement présent dans les thermopompes thermoélectriques.

2.1.3. Effet Thomson

En 1854, William Thomson découvre un effet similaire à l'effet Peltier. Lorsqu'une différence de température existe dans un matériau conducteur avec la présence d'un courant, une absorption ou un dégagement de chaleur est observé selon la direction du courant. L'effet Thomson est également réversible et survient en plus

de l'effet Peltier tel que présenté à la Figure 2.2. Le taux de chaleur absorbé ou dégagé est calculé de la façon suivante :

$$q_{Thomson} = -\tau I \nabla T \quad (2.3)$$

où τ est le coefficient Thomson. En comparaison aux autres effets thermoélectriques, l'effet Thomson est relativement faible et son impact est souvent négligeable.

2.1.4. Interrelation entre les effets thermoélectriques

En plus de l'effet Thomson, William Thomson a également établi deux relations qui relient ensemble les trois effets thermoélectriques. Ces deux relations sont souvent utilisées dans les modèles thermoélectriques [1] :

$$\Pi = S T \quad (2.4)$$

$$\tau = T \left(\frac{\partial S}{\partial T} \right) \quad (2.5)$$

À partir de l'équation (2.4), il est possible d'exprimer le coefficient Peltier en fonction du coefficient Seebeck. De plus, d'après l'équation (2.5), si le coefficient Seebeck est indépendant de la température, alors le coefficient Thomson devient nul. Ces relations seront utiles dans le développement du modèle thermoélectrique présenté à la section 2.3.1.

2.2 Modélisation d'un module thermoélectrique

Un module thermoélectrique est la composante à la base d'un générateur thermoélectrique. Un module thermoélectrique est formé de deux pattes composées de deux matériaux semi-conducteurs dopés P et N. Les deux pattes sont agencées de manière à être connectées électriquement en série et thermiquement en parallèle telles que présentées à la Figure 2.3. Un matériau dopé P possède un excédent de trous (absence d'électron) alors qu'un matériau dopé N possède un excédent d'électrons. Des électrodes assurent la connexion électrique des deux pattes du module. Lorsqu'une différence de température est appliquée aux jonctions du module, un déplacement d'électron dans le matériau dopé N et un déplacement de trou dans le matériau dopé P est engendré par effet Seebeck. Lorsque le module est connecté à un circuit électrique fermé (relié à une résistance externe), un courant électrique est généré qui permet donc d'obtenir une puissance électrique.

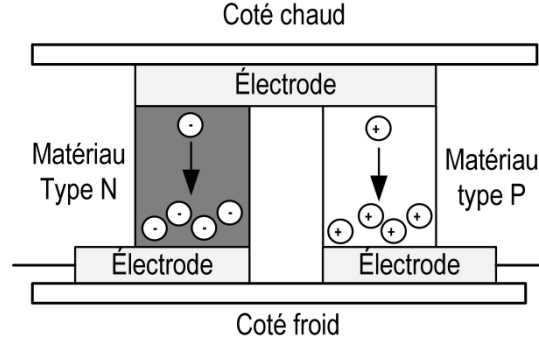


Figure 2.3 : Module thermoélectrique dans sa configuration de base.

La géométrie des pattes est un aspect important du design d'un module thermoélectrique puisqu'il influence grandement la génération thermoélectrique. La géométrie des pattes affecte la résistance électrique du module. Puisque les deux pattes sont disposées électriquement en série, il suffit d'additionner la résistance électrique de chacune [2] :

$$R = \left(\frac{\rho L}{A} \right)_P + \left(\frac{\rho L}{A} \right)_N \quad (2.6)$$

où L est la longueur de la patte et A est l'aire transversale. La conductance thermique est également influencée par la géométrie du module. Puisque les pattes sont disposées thermiquement en parallèle, la conductance totale se calcule de la façon suivante [2] :

$$K = \left(\frac{kA}{L} \right)_P + \left(\frac{kA}{L} \right)_N \quad (2.7)$$

Selon les équations (2.6) et (2.7), on observe que le ratio de l'aire sur la longueur (A/L) a un impact sur la conductivité thermique et la résistance électrique qui joue un rôle important dans la production thermoélectrique.

Le choix des matériaux est également un aspect à considérer puisqu'il influence directement le coefficient Seebeck qui détermine le voltage généré par une différence de température. Il est à noter que les propriétés thermoélectriques des matériaux pour la génération thermoélectrique sont dépendantes de la température pour la vaste majorité des semi-conducteurs utilisés en thermoélectricité (conductivité thermique, résistance électrique et coefficient Seebeck). Conséquemment, la capacité d'un matériau à convertir la chaleur en électricité n'est pas garantie pour toutes les plages de température. Les matériaux doivent donc être

sélectionnés selon le cas d'application. En thermoélectricité, la performance d'un module thermoélectrique est mesurée à partir de la figure de mérite adimensionnel [1] :

$$ZT = \frac{S^2}{RK} T \quad (2.8)$$

où T est la température absolue du module. Plus un matériau possède un coefficient Seebeck maximal, une conductivité électrique maximal et une conductivité thermique minimale, et plus celui-ci sera en mesure de convertir la chaleur en électricité. Il est à noter que la plupart des matériaux thermoélectrique actuels sont de l'ordre de $ZT \sim 1$ ce qui correspond à une faible efficacité de conversion approximative de 2-5%. Néanmoins, les avancées récentes permettront éventuellement d'obtenir des matériaux beaucoup plus performants. De nouveaux matériaux émergents avec des valeurs de $ZT \sim 2$ seront possibles avec les nanostructures [3].

2.3 Générateur thermoélectrique

Pour obtenir un générateur thermoélectrique, plusieurs modules sont combinés afin d'augmenter considérablement la force électromotrice générée. Il est possible de les assembler en série et même sur plusieurs étages. Le système doit ensuite être relié à une résistance externe qui permet de déterminer la puissance électrique en sortie. Un schéma 2D d'un générateur thermoélectrique composé de trois modules en série sur un étage est présenté à la Figure 2.4.

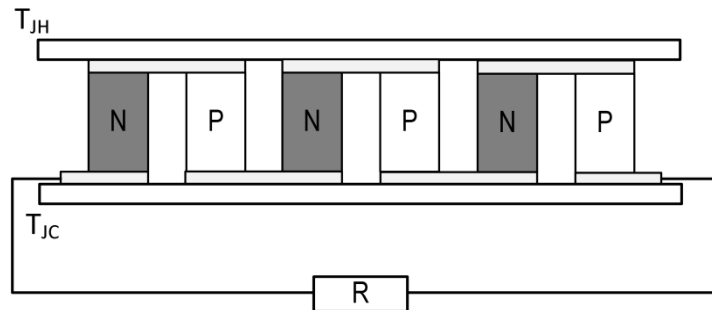


Figure 2.4 : Générateur thermoélectrique

Afin de déterminer la puissance thermoélectrique du système, il est nécessaire de développer un modèle mathématiquement. Ce modèle est basé sur des bilans d'énergie et de charge dont le développement mathématique est présenté à la section suivante.

2.3.1. Développement mathématique des équations thermoélectriques

Le modèle mathématique développé ici permettra au lecteur de comprendre la provenance des équations qui sont largement utilisées dans de nombreux modèles thermoélectriques. Ces équations sont d'ailleurs utilisées pour le développement du modèle thermoélectrique présenté au chapitre 4. Il est à noter que plusieurs hypothèses sont initialement posées afin de simplifier la démarche :

- Le générateur thermoélectrique est en régime stationnaire.
- Les résistances de contact thermique et électrique sont négligeables.
- Le coefficient Seebeck est indépendant de la température (effet Thomson négligeable).
- Le transfert de chaleur convectif et radiatif aux surfaces est négligeable.
- Aucun champ magnétique n'est considéré.

Pour un matériau isentropique, deux équations gouvernantes régissent la thermoélectricité soit l'équation de la continuité de la charge électrique et l'équation de diffusion de la chaleur :

$$\vec{\nabla} \cdot \vec{j} = 0 \quad (2.9)$$

$$\rho_v C_p \frac{\partial T}{\partial t} = -\vec{\nabla} \cdot \vec{q} + \dot{q} \quad (2.10)$$

où j est la densité de courant, ρ_v est la masse volumique et C_p est la capacité calorifique. En régime permanent, l'équation (2.10) devient :

$$-\vec{\nabla} \cdot \vec{q} + \dot{q} = 0 \quad (2.11)$$

De plus, le champ électrique généré E est affecté par la loi d'ohm et l'effet Seebeck qui s'exprime de la façon suivante :

$$\vec{E} = \vec{j} \rho + S \vec{\nabla} T \quad (2.12)$$

où ρ est la résistivité électrique. Une relation pour le flux de chaleur \vec{q} est obtenue à partir des travaux de Thomson et Onsager qui prouvent que les phénomènes thermiques et thermoélectriques sont indépendants. Le taux de chaleur s'exprime donc comme la combinaison de l'effet Peltier et de la conduction de chaleur dans le matériau :

$$\vec{q} = \Pi \vec{j} - k \vec{\nabla} T = S T \vec{j} - k \vec{\nabla} T \quad (2.13)$$

La génération de chaleur \dot{q} due à l'effet joule s'exprime de la façon suivante :

$$\dot{q} = \vec{E} \cdot \vec{j} = j^2 \rho + \vec{j} \cdot \vec{S} \nabla T . \quad (2.14)$$

En substituant les équations (2.13) et (2.14) dans l'équation (2.11), on obtient l'équation suivante :

$$\nabla \cdot (k A \nabla T) + \frac{I^2 \rho}{A} = 0 . \quad (2.15)$$

Prenons un module thermoélectrique que l'on connecte en circuit fermé à une résistance électrique externe tel que présenté sur la Figure 2.5. Chaque patte (dopé N et P) possède leurs propres propriétés géométriques (A/L) ainsi que leurs propres propriétés thermoélectriques (ρ , k et S).

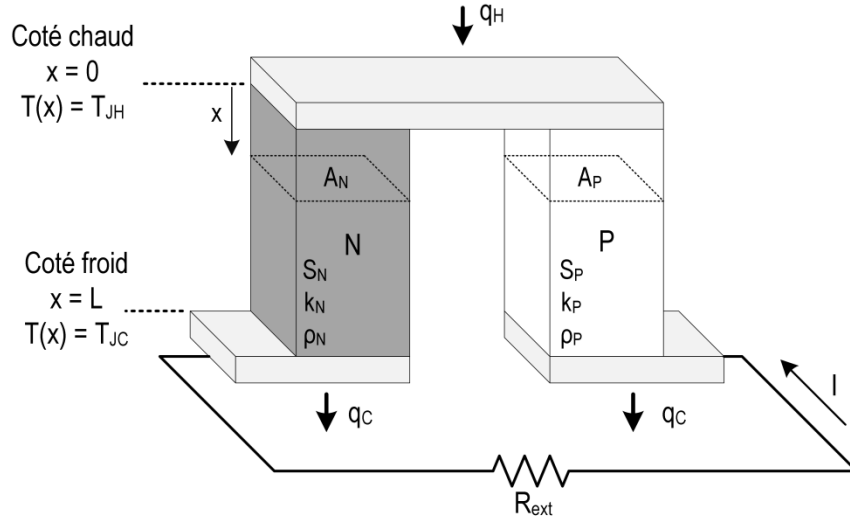


Figure 2.5 : Schéma du modèle thermoélectrique de base en circuit fermé.

Lorsqu'on pose les conditions limites $T(x=0) = T_{JH}$ et $T(x=L) = T_{JC}$, on peut résoudre l'équation (2.15) et trouver la solution pour le gradient de température :

$$\left. \frac{dT}{dx} \right|_{x=0} = \frac{I^2 \rho L}{2 A^2 k} - \frac{(T_{JH} - T_{JC})}{L} . \quad (2.16)$$

On peut ensuite déterminer le taux de chaleur entrant dans les modules à partir de la jonction chaude ($x=0$) en incluant les pattes de type P et N :

$$q_H = n \left[(S_P - S_N) T_{JH} I + \left(-kA \frac{dT}{dx} \Big|_{x=0} \right)_P + \left(-kA \frac{dT}{dx} \Big|_{x=0} \right)_N \right] \quad (2.17)$$

où n est le nombre de modules total connectés en série dans le générateur thermoélectrique. En substituant l'équation (2.16) dans l'équation (2.13) on obtient la relation suivante :

$$q_H = \left[(S_P - S_N) T_{JH} I - \frac{I^2}{2} \left(\frac{\rho_P L_P}{A_P} + \frac{\rho_N L_N}{A_N} \right) + \left(\frac{k_P A_P}{L_P} + \frac{k_N L A_N}{L_N} \right) (T_{JH} - T_{JC}) \right]. \quad (2.18)$$

Finalement, en substituant les équations (2.6) et (2.7) dans (2.18), on obtient l'équation finale [2][4] :

$$q_H = S T_{JH} I - \frac{R I^2}{2} + K (T_{JH} - T_{JC}) \quad (2.19)$$

où S est le coefficient Seebeck du module défini par $S = S_P - S_N$. De la même façon, en effectuant l'analyse du flux de chaleur sortant à la jonction froide ($x = L$), on obtient la relation suivante [2][4] :

$$q_C = S T_{JC} I + \frac{R I^2}{2} + K (T_{JH} - T_{JC}) \quad (2.20)$$

Les équations (2.19) et (2.20) forment les équations de base en thermoélectricité. Finalement, la puissance électrique résultante est calculée à partir de la différence entre le flux de chaleur entrant et sortant :

$$P = q_H - q_C = n \left[S I (T_{JH} - T_{JC}) - R I^2 \right]. \quad (2.21)$$

La puissance électrique peut également être exprimée en fonction de la résistance externe du système :

$$P = R_{ext} I^2 \quad (2.22)$$

Les équations (2.19) à (2.22) sont utilisées dans le développement du modèle thermoélectrique présenté à l'article #1 du chapitre 4.

Chapitre 3 : Algorithmes d'optimisation heuristiques

Cette section permettra au lecteur de comprendre le fonctionnement général d'un algorithme d'optimisation heuristique ainsi que les particularités des mécanismes de convergence de ceux qui ont été étudiés dans le cadre de ce mémoire.

3.1. Initialisation d'un algorithme heuristique

Les algorithmes présentés dans ce chapitre utilisent les mêmes mécanismes d'initialisation. En d'autres mots, la façon dont les algorithmes sont préparés pour solutionner un problème est la même. Une première étape consiste à créer une population. Cette population forme un ensemble de solutions initiales qui doit idéalement couvrir la majorité de l'espace de design pour de meilleures performances. Selon le problème d'optimisation, il est nécessaire de choisir une population plus ou moins grande pour bien couvrir l'espace de design. En contrepartie, le temps de calcul sera lié à la taille de la population. Dans un souci de simplicité et de généralisation entre chaque algorithme, tous les individus d'une population sont regroupés dans la variable x . Pour chaque itération j , une solution i est donc défini par $x_{i,j}$, et ce, pour l'ensemble des algorithmes présentés dans ce chapitre.

La population initiale peut être générée aléatoirement. Toutefois, lorsque le nombre de dimensions pour une fonction objectif est grand, l'échantillonnage aléatoire peut présenter certaines lacunes. Néanmoins, il est possible de générer la population de départ à l'aide d'autres méthodes d'échantillonnages. Par exemple, des méthodes pseudo-stochastiques tel que le « latin hypercube sampling » (LHS) ou de faibles écarts comme le « Faure sequence sampling » (FSS) permettent de mieux couvrir l'espace de design. Pour bien distinguer la différence entre les différentes méthodes d'échantillonnages, une population de 100 individus pour un objectif à deux dimensions $\{x_1, x_2\}$ est généré à partir d'un échantillonnage aléatoire, de LHS et de FSS. Les résultats sont présentés à la Figure 3.1. On y observe que l'échantillonnage aléatoire génère plusieurs groupements de points ce qui entraîne plusieurs zones de l'espace de design sans aucune solution de départ. Lorsque le nombre de dimensions devient important, cet effet est accentué. Conséquemment, il est possible que la vitesse de convergence ainsi que la précision d'un algorithme heuristique en soient diminuées si ces zones inexplorées sont nombreuses et si elles contiennent le minimum global recherché. En contrepartie, le LHS couvre mieux l'espace de design puisque les individus sont plus distancés dus à la méthode employée lors de l'échantillonnage. Le FSS, quant à lui, est une méthode idéale pour couvrir intégralement l'espace de

design puisque l'ensemble des individus est quasi équidistant. De façon générale, des méthodes d'échantillonnages comme le LHS et le FSS permettent d'atteindre de meilleures performances en augmentant la vitesse de convergence, la précision ainsi que la diversité des solutions obtenues [5]. Dans le cadre des articles présentés au chapitre 4 et au chapitre 5, l'ensemble des populations initiales est généré à partir d'un LHS avec une fonction développée à l'interne sur Matlab.

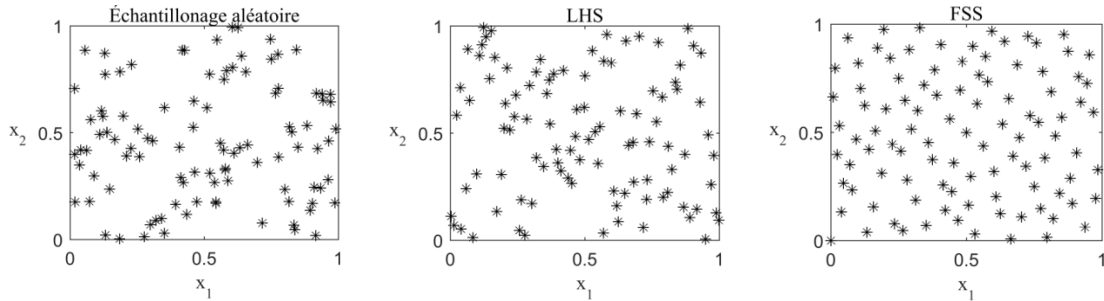


Figure 3.1 : Génération d'une population de 100 individus par une méthode d'échantillonnage aléatoire, LHS et FSS.

Les paramètres d'algorithme doivent également être réglés adéquatement pour déterminer la façon dont l'espace de design sera exploré par la méthode de recherche directe. Selon le choix de l'utilisateur, l'algorithme peut intensifier la recherche autour de solutions intéressantes ou plutôt opter pour une exploration de l'espace de design. De façon générale, une intensification de la recherche permet de converger autour d'un minimum. Toutefois, ce minimum peut être un minimum local si l'intensification est trop importante ou survient trop tôt dans l'optimisation. À l'inverse, une exploration élevée permet d'explorer des zones de l'espace de design qui peuvent contenir la solution recherchée. Cependant, une exploration excessive peut ralentir considérablement la convergence de l'algorithme. Généralement, le choix optimal est d'avoir initialement une exploration élevée et ensuite augmenter graduellement l'intensification à l'aide des réglages intrinsèques à chaque algorithme. Les sous-sections suivantes présenteront les différents algorithmes d'optimisation heuristiques explorés dans le chapitre 5 ainsi que leur fonctionnement. L'ensemble des algorithmes présentés ici a été implémenté sur Matlab à l'exclusion de l'algorithme génétique qui était déjà présent avec l'« Optimization toolbox » de Matlab [6].

3.2. Algorithme génétique (GA)

L'algorithme génétique est un algorithme de type évolutionnaire bien connu de la littérature scientifique basé sur le principe de l'évolution. L'algorithme simule l'évolution d'une population en générant des progénitures avec des traits potentiellement supérieurs à leurs parents. Des mécanismes naturels tels que la sélection, le croisement ainsi que la mutation sont repris par cet algorithme.

Pour résumer le fonctionnement général de l'algorithme, la Figure 3.2 présente la structure complète de l'algorithme génétique incluant chacun des mécanismes naturels utilisés lors d'une itération. Initialement, une population est créée et des parents sont sélectionnés selon leur valeur de la fonction objective. De nouvelles solutions appelées « enfants » sont ensuite créées par le croisement de ces parents. Finalement, une mutation est effectuée sur ces nouvelles solutions. Cette mutation modifie aléatoirement la valeur de certains gènes des enfants générées lors du croisement. Cela permet de conserver de la diversité dans les solutions et d'éviter une convergence prématurée de l'algorithme vers un minimum local. Finalement, l'algorithme reforme une nouvelle population composée des meilleurs individus de l'itération précédente et les enfants incluant les mutations. Pour conserver le nombre d'individus constant dans la population, les mauvaises solutions sont tronquées et retirées. Optionnellement, l'utilisateur peut déterminer un pourcentage de la population de l'itération précédente qui survivra à la prochaine génération. Ce groupe d'individus aussi appelé « élite » est déterminé par le ratio d'élitisme. Il est à noter que chacune des étapes comprend des réglages et des méthodes qui peuvent être sélectionnés individuellement par l'utilisateur. Ces aspects seront traités dans les prochaines sections.

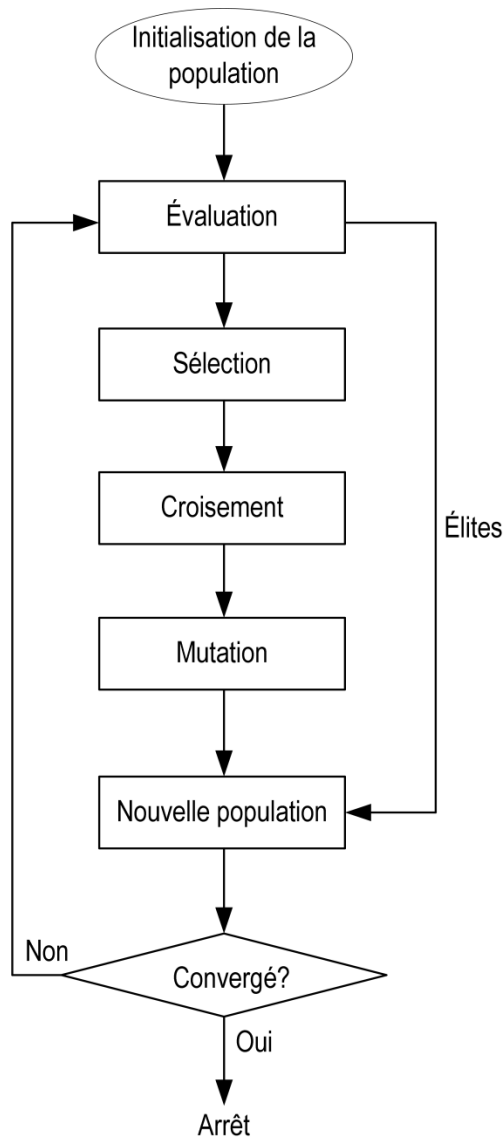


Figure 3.2 : Structure de l'algorithme génétique.

3.2.1. Sélection

La première étape est la sélection des parents. Le nombre de parents sélectionnés peut être modifié avec le ratio de croisement (CR) qui détermine le pourcentage de la population qui seront des parents. Ce ratio détermine également le nombre d'enfants qui seront générés lors du croisement. Différentes méthodes peuvent être employées pour la sélection des parents. Une de ces méthodes est la sélection par roulette qui simule un jeu de roulette sur laquelle chaque individu se retrouve. Un exemple est illustré sur la Figure 3.3 pour une population de 4 individus. Une probabilité est calculée pour chaque solution à partir d'une distribution gaussienne. Plus une solution est bonne et plus sa présence sur la roue est élevée (probabilité plus grande d'être choisi comme parent).

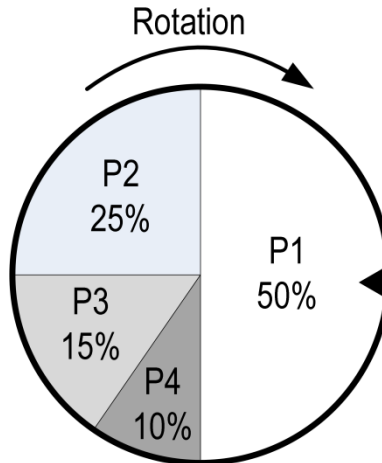


Figure 3.3 : Méthode de sélection par jeu de roulette.

Une autre méthode de sélection consiste à effectuer un « tournoi » entre plusieurs individus. Au départ, des individus sont sélectionnés aléatoirement dans la population pour former plusieurs tournois. Les meilleures solutions de chaque lot seront sélectionnées comme parents. Un exemple est présenté à la Figure 3.4 dans une population dont un tournoi est effectué entre 3 solutions d'une population de 4 individus. L'utilisateur peut agrandir le nombre d'individus par tournoi au besoin.

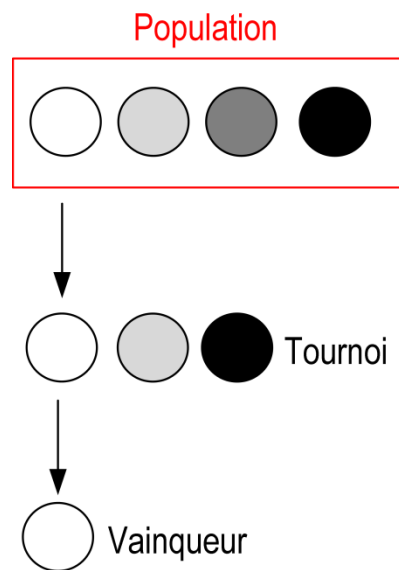


Figure 3.4 : Méthode de sélection par tournoi.

3.2.2. Croisement

Après avoir sélectionné les parents, la prochaine étape consiste à effectuer un croisement entre deux parents pour obtenir une nouvelle solution (enfant). Ce croisement peut se faire de différentes façons. L'une de ces

méthodes est le croisement par dispersion qui consiste à créer aléatoirement un vecteur binaire de même longueur que les vecteurs parents. Pour chaque variable, si le vecteur binaire à la valeur 0, alors l'enfant héritera de la valeur du parent #1. Sinon, l'enfant héritera de la valeur du parent #2. Un exemple de croisement est illustré à la Figure 3.5 pour une solution avec 7 variables.

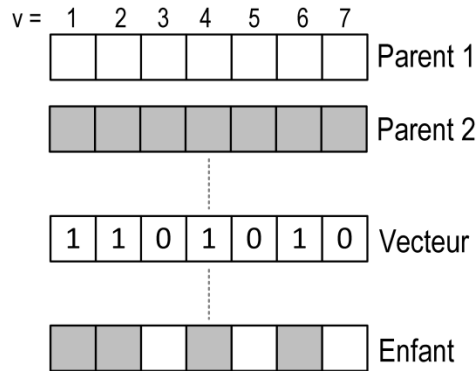


Figure 3.5 : Croisement par dispersion.

3.2.3. Mutation

La mutation est la dernière étape de l'algorithme. La mutation permet de modifier une solution de façon à générer plus de diversité. Le nombre de mutants est défini par le ratio de mutation qui détermine le pourcentage de la population qui subira une mutation. Un second paramètre est également requis pour déterminer le nombre de variables affectées par la mutation sur chaque solution. Similairement à la sélection et au croisement, plusieurs méthodes existent pour effectuer la mutation. L'une de ces méthodes consiste à modifier la valeur d'une variable v selon une proportion des limites de la variable de la façon suivante :

$$X_{v,new} = X_{v,old} + \alpha (X_{v,max} - X_{v,min}) * randn \quad (3.1)$$

où α détermine l'intensité de la mutation, $X_{v,max}$ et $X_{v,min}$ sont les limites maximales et minimales de la variable v ; et $randn$ est un nombre aléatoire entre -1 et 1.

La nouvelle population est ensuite générée à partir des enfants et des meilleurs individus de l'itération précédente incluant les mutations. Ces étapes sont répétées à chaque génération jusqu'à atteindre le critère d'arrêt fixé par l'utilisateur.

3.1.4. Algorithme génétique Multi-objectifs

L'algorithme génétique peut également être utilisé pour effectuer des optimisations multiobjectifs. Des variantes de l'algorithme génétique tel que NSGA-II est largement utilisée en pratique et permet de trouver un

ensemble de solutions en considérant les compromis entre plusieurs objectifs d'optimisation [7]. Les mécanismes de base de l'algorithme sont sensiblement les mêmes que ceux pour un seul objectif. Toutefois, puisqu'il existe plusieurs solutions optimales (compromis entre les différents objectifs), le classement des individus est différent. L'algorithme sélectionnera les solutions optimales basées sur la valeur de l'objectif ainsi que la distance physique entre les solutions (« crowding-distance ») afin de bien couvrir l'ensemble des valeurs possible des objectifs pour les différents designs retenus. Ce classement permet donc d'obtenir des points non dominés qui formeront un ensemble de solutions optimales également appelé le front de Pareto. Cet algorithme est utilisé pour produire certains résultats présentés à l'article #1 au chapitre 4.

3.3. Colonie de fourmis avec variables continues (ACO_R)

L'algorithme « colonie de fourmis » (ou « Ant colony optimization » de son nom anglais) fut introduit par Dorigo et ses collègues [8]. Cet algorithme basé sur le comportement des fourmis à la recherche de nourriture fut initialement développé pour la résolution de problèmes discrets. En d'autres mots, l'algorithme permet de trouver une solution optimale parmi un ensemble fini de solutions réalisables. Initialement, chaque fourmi explore aléatoirement l'espace de design. Lorsqu'une fourmi trouve une source de nourriture, celle-ci évalue la quantité et la qualité de la source et relâche une quantité de phéromone correspondante lors de son retour au nid. La trace de phéromone guide les autres fourmis vers cette source de nourriture. L'algorithme fut ensuite adapté pour la résolution de problèmes continus [9]. Cette version de l'algorithme est celle présentée dans cette section et celle utilisée dans l'article #2 au chapitre 5.

Initialement, un classement est attribué à chaque solution x_i selon leur valeur de la fonction objective. Un poids ω est ensuite attribué à chaque individu i de la population selon la fonction gaussienne suivante :

$$\omega_i = \frac{1}{q k \sqrt{2\pi}} e^{\frac{-(i-1)^2}{2q^2 k^2}} \quad (3.2)$$

où k est le nombre total d'individus et q est un facteur d'intensification qui affecte la répartition du poids sur les solutions. Lorsque la valeur de q est faible, les solutions aux premiers rangs obtiennent un poids plus élevé. À l'inverse, une valeur de q élevée génère un poids uniformisé entre les individus. La probabilité de sélectionner la solution i est ensuite calculée de la façon suivante :

$$p_i = \frac{\omega_i}{\sum_{i=1}^k \omega_i} \quad (3.3)$$

L'algorithme procède ensuite à la construction de nouvelles solutions (fourmis). Ces solutions sont construites variable par variable et le nombre total de nouvelles solutions construites peut être adapté par l'utilisateur au besoin. Idéalement, le nombre de nouvelles solutions ne doit pas dépasser la taille de la population. À chaque itération, l'algorithme calcule un écart-type pour chaque variable de chaque solution. Pour chacune des étapes de construction c (chaque variable), l'écart-type est calculé de la façon suivante :

$$\sigma_i^c = \xi \sum_{e=1}^k \frac{|X_e^c - X_i^c|}{k-1} \quad (3.4)$$

où ξ est un paramètre ($\xi > 0$) qui simule l'évaporation de phéromone dans l'algorithme de base. Une valeur élevée de ξ réduit la vitesse de convergence de l'algorithme et favorise une exploration plus globale de l'espace de design. À chaque étape de construction c , l'algorithme sélectionne un individu en fonction de sa probabilité p . Différentes méthodes existent pour sélectionner la solution telle que le jeu de roulette présenté à la section 3.2. La valeur attribuée à la variable c de la nouvelle solution se calcul à partir de l'écart type de la solution i sélectionnée par la méthode probabiliste de la façon suivante :

$$X_{new}^c = X_i^c + randn * \sigma_i^c \quad (3.5)$$

où $randn$ est un nombre aléatoire entre -1 et 1. Lorsque le nombre total de nouvelles solutions est atteint, celles-ci sont évaluées et ensuite ajoutées à la population. Un triage est ensuite effectué selon la valeur de leur fonction objective. Pour conserver le nombre d'individus constant, les mauvaises solutions sont retirées de la population pour la prochaine itération. La structure complète de l'algorithme est présentée à la Figure 3.6.

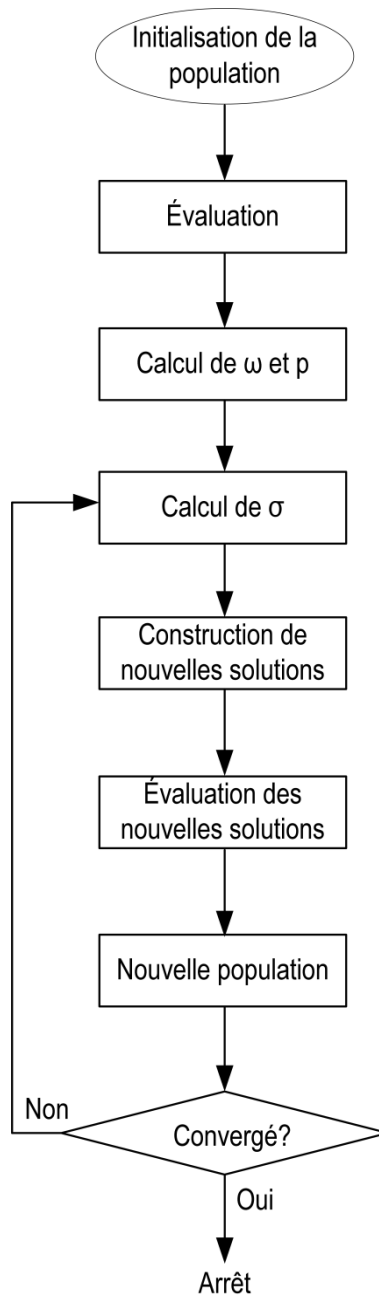


Figure 3.6 : Structure de l'algorithme ACO_R

3.4. Essaim particulaire (PSO)

L'algorithme essaim particulaire (ou « Particle swarm optimization » de son nom anglais) fut introduit pour la première fois par Kennedy et Eberhart [10]. Cet algorithme heuristique a été développé à la suite de l'étude du comportement social des troupes d'animaux tel que les oiseaux et les poissons. L'algorithme est de type « swarm », c'est-à-dire un ensemble de « particules » formant une population qui interagissent ensemble pour rechercher un optimum. Chaque « particule » est une solution potentielle du problème.

L'algorithme est composé initialement d'une population x . Pour chaque individu i à chaque itération j , l'algorithme modifie la solution $x_{i,j}$ à l'aide du concept de vitesse $v_{i,j+1}$ de la façon suivante :

$$x_{i,j+1} = x_{i,j} + v_{i,j+1} \quad (3.6)$$

où $x_{i,j+1}$ est la nouvelle position de l'individu i dans l'espace de design. La particularité de l'algorithme repose sur la façon dont le vecteur de vitesse est calculé. En effet, trois contributions sont utilisées afin de déterminer la vitesse de chaque individu à chaque itération. La première contribution vient de l'apprentissage personnel. À chaque itération, chaque individu conserve en mémoire sa meilleure position personnelle dénotée P_i . La deuxième contribution vient de l'interaction sociale entre les particules. La meilleure position globale de l'essaim dénotée G_j est connue par l'ensemble de la population. La dernière contribution provient de la vitesse de l'itération actuelle. Une proportion de cette vitesse est reprise pour le calcul de la vitesse de l'itération suivante. L'équation présentée ci-dessous décrit essentiellement comment la vitesse $v_{i,j+1}$ est calculée à partir de ces trois contributions :

$$v_{i,j+1} = w v_{i,j} + c_1 r_1 (P_{i,j} - x_{i,j}) + c_2 r_2 (G_j - x_{i,j}) \quad (3.7)$$

où r_1 et r_2 sont des nombres aléatoires entre 0 et 1. Les composantes de l'équation (3.7) sont communément appelés dans l'ordre : la composante inertielle, la composante d'apprentissage personnelle et la composante d'apprentissage global. Un exemple schématique du déplacement d'une particule est illustré sur la Figure 3.7 pour une fonction objectif à deux dimensions dont les trois composantes de la vitesse $v_{i,j+1}$ (en bleu) sont additionnées à la position actuelle $x_{i,j}$ pour obtenir la nouvelle position $x_{i,j+1}$.

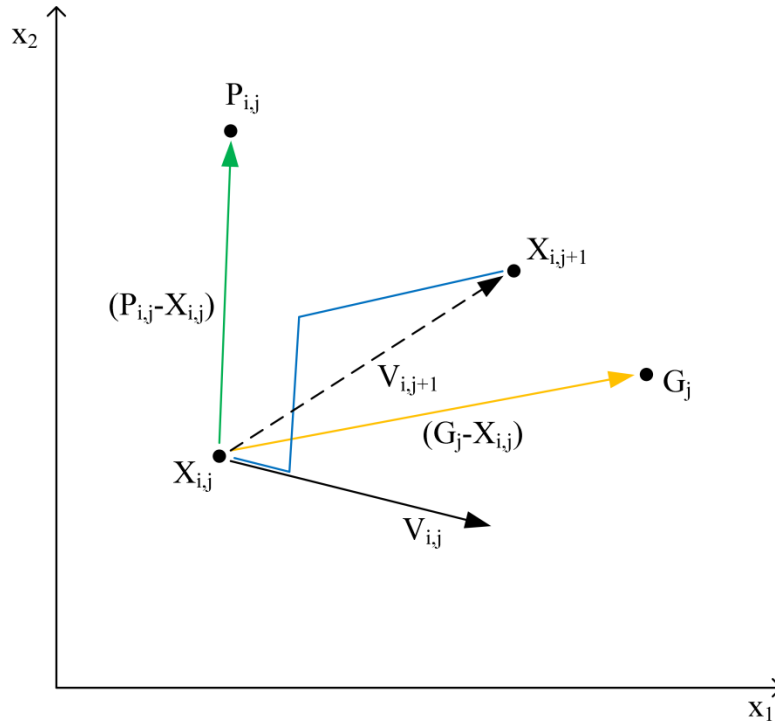


Figure 3.7 : Exemple du déplacement d'une particule $x_{i,j}$ vers la nouvelle position $x_{i,j+1}$ pour une fonction objective à deux dimensions.

Le facteur inertiel w permet de contrôler l'impact de la vitesse actuelle sur la nouvelle position des individus. Plus w est élevé et plus les particules auront une « inertie » importante ce qui réduira l'impact de la composante d'apprentissage personnelle et global sur la vitesse de déplacement de la particule favorisant ainsi une exploration de l'espace de design. Ce coefficient peut offrir un avantage pour la résolution de problème avec la présence de minimum locaux en permettant aux particules de s'échapper plus facilement de minimums locaux. À l'inverse, w plus faible favorise une recherche locale de l'espace de design. Afin d'intensifier la recherche de solutions autour du minimum global, il est fréquent d'ajouter un amortissement à w ce qui réduit w à chaque itération pour favoriser une recherche locale lorsque l'essaim se rapproche de la solution optimale. Le facteur d'apprentissage personnel c_1 contrôle le poids de la meilleure solution personnelle de la particule sur son changement de position. Le facteur social c_2 contrôle le poids de la meilleure solution global de l'essaim sur le changement de la position de la particule. Les facteurs w , c_1 et c_2 sont des paramètres qui peuvent être ajustés indépendamment selon le problème d'optimisation afin d'améliorer les performances de l'algorithme. La structure complète de l'algorithme est présentée à la Figure 3.8.

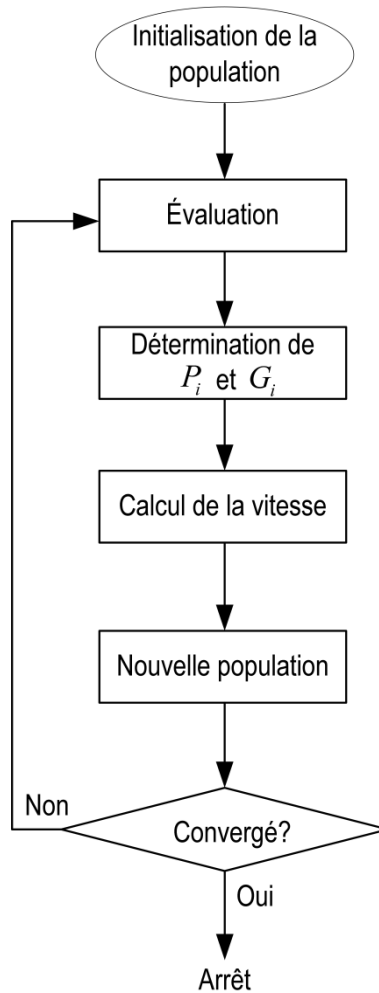


Figure 3.8 : Structure de l'algorithme PSO.

3.5. Évolution différentielle (DE)

L'évolution différentielle (ou «Differential evolution» de son nom anglais) est un algorithme utilisant lui aussi des méthodes stochastiques et de recherches directes. L'algorithme fut développé par Storn et Price en 1997 avec les objectifs suivants [11] :

- (1) Être en mesure de résoudre des fonctions multimodales, non linéaires et non différentiables.
- (2) Permettre l'utilisation de calcul en parallèle.
- (3) Un nombre minimal de paramètres d'optimisation avec des réglages simples et robustes.
- (4) Converger avec grande précision et fiabilité.

DE utilise des opérateurs similaires à l'algorithme génétique tels que la mutation, le croisement et la sélection. Toutefois, ceux-ci sont effectués dans un autre ordre dont les mécanismes sont différents. La structure complète de l'évolution différentielle est présentée à la Figure 3.9.

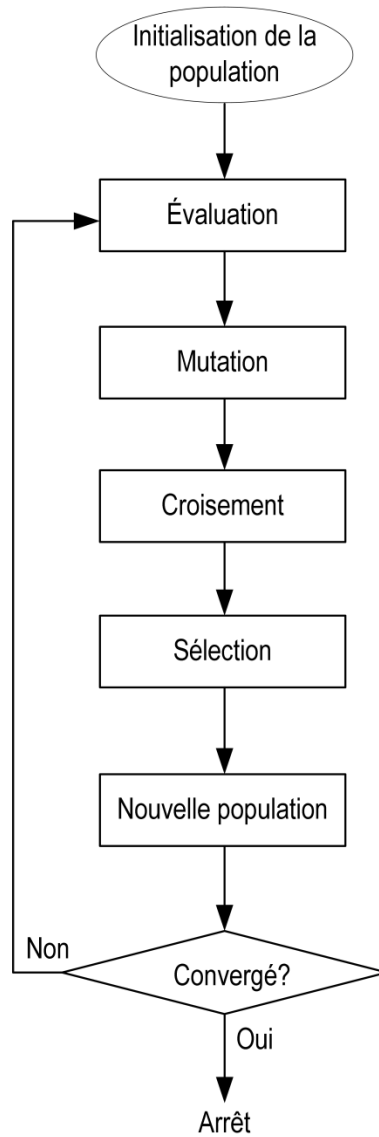


Figure 3.9 : Structure de l'algorithme DE.

3.5.1. Mutation

La mutation est la première phase de l'algorithme. Pour chaque individu i à chaque itération j , l'algorithme sélectionne aléatoirement trois vecteurs solutions $\{x_{r1,j}, x_{r2,j}, x_{r3,j}\}$ pour former un nouveau vecteur mutant $M_{i,j+1}$:

$$M_{i,j+1} = x_{r1,j} + \beta (x_{r2,j} - x_{r3,j}) \quad (3.8)$$

où β est un paramètre de l'algorithme ayant une valeur entre $]0,2]$ et qui modifie l'intensité de la différence entre les vecteurs solutions x_{r_2} et x_{r_3} sur le vecteur mutant. De façon générale, une valeur élevée de β augmente l'exploration de l'espace de design. La vitesse de convergence est considérablement réduite si ce paramètre devient trop important. Un exemple de la création d'un vecteur mutant $M_{i,j+1}$ est illustré à la Figure 3.10 pour un objectif à deux dimensions.

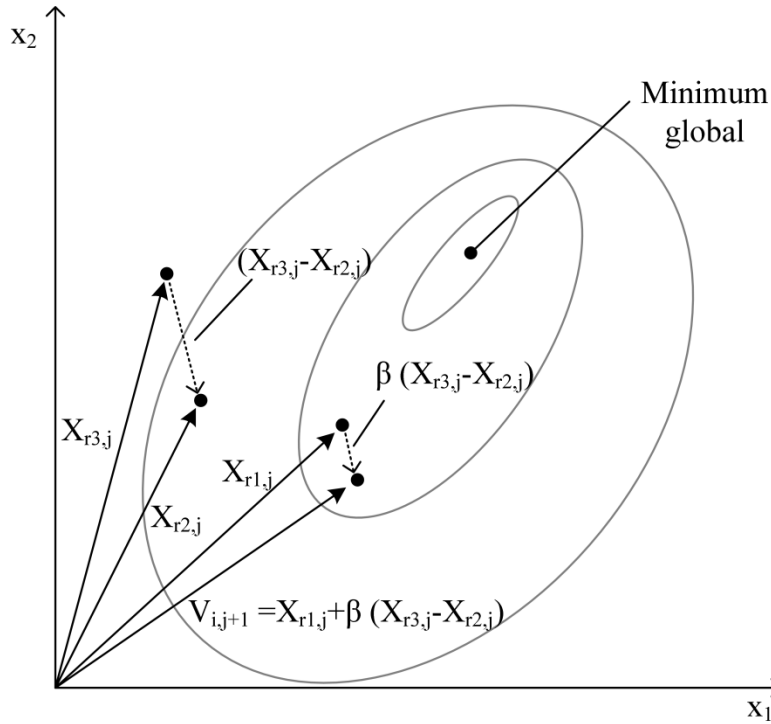


Figure 3.10 : Exemple de la mutation pour une fonction objective à deux dimensions.

3.5.2. Croisement et sélection

Ensuite, pour chaque individu $x_{i,j}$ de l'itération j , un croisement est effectué avec le vecteur mutant $M_{i,j+1}$ pour générer un vecteur de croisement $U_{i,j+1}$. Le paramètre d'algorithme p_{CR} est introduit pour déterminer le ratio de croisement qui peut varier entre 0 et 1. La Figure 3.11 illustre le mécanisme de croisement entre $x_{i,j}$ et $M_{i,j+1}$ avec un exemple de fonction objectif qui dépend de 5 variables. Un nombre aléatoire r entre 0 et 1 est généré pour chaque variable v de $x_{i,j}$. Si p_{CR} est plus élevé que ce nombre, alors $U_{i,j+1}$ prend la valeur de $M_{i,j+1}$. Dans le cas contraire, la valeur de $x_{i,j}$ sera retenue.

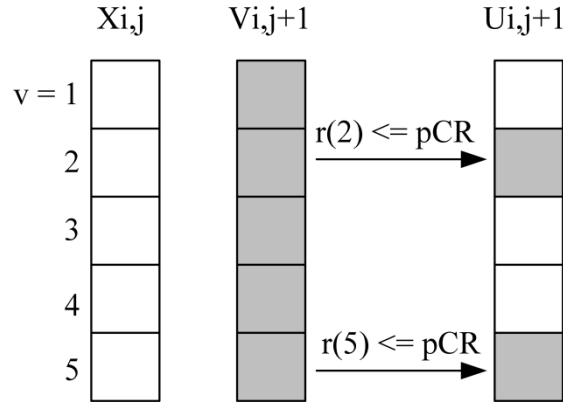


Figure 3.11 : Exemple du croisement pour une fonction objective à 5 dimensions.

La dernière étape est la sélection. Une comparaison est effectuée entre $x_{i,j}$ et $U_{i,j+1}$. Le vecteur ayant la meilleure solution sera retenu. Si le vecteur de croisement est retenu, celui-ci remplace le vecteur $x_{i,j}$ pour la prochaine itération.

3.6. Algorithme «Enseignement-apprentissage» (TLBO)

L'algorithme enseignement-apprentissage (ou « teaching-learning basic algorithm » de son nom anglais) est un algorithme stochastique qui simule l'évolution d'étudiant en apprentissage [12]. Le concept d'enseignement est utilisé pour améliorer le score des étudiants avec l'enseignement prodigué par le professeur. À chaque itération j , le meilleur individu devient le professeur. Chaque étudiant i est un membre de la population et une solution du problème $x_{i,j}$. L'algorithme fonctionne en deux phases : l'enseignement et l'apprentissage. Chacune des phases demandent d'évaluer toute la population (l'ensemble des solutions). Conséquemment, cet algorithme double le nombre d'évaluations de la fonction objective ce qui augmente le temps de calcul et l'utilisation de la mémoire informatique en comparaison avec d'autres algorithmes heuristiques. En contrepartie, aucun réglage spécifique n'est requis avec cet algorithme ce qui le rend facile d'utilisation. La structure complète de l'algorithme est présentée à la Figure 3.12.

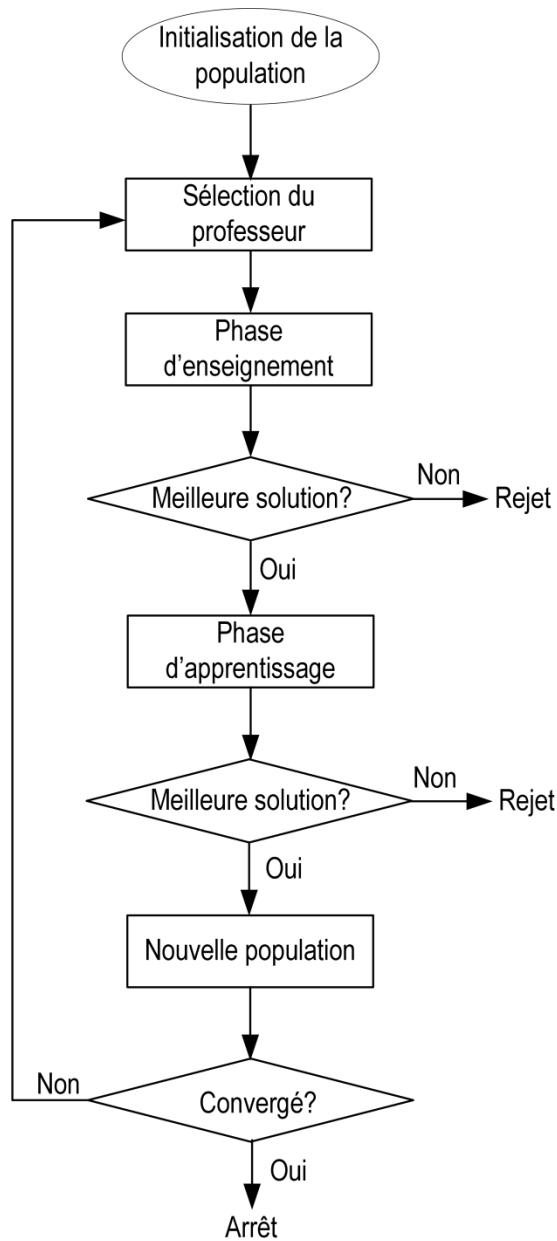


Figure 3.12 : Structure de l'algorithme TLBO.

3.6.1. Enseignement

Lors de la première phase, un professeur (la meilleure solution) est désigné pour prodiguer son savoir à l'ensemble des étudiants. À chaque itération j , le meilleur individu de la population est sélectionné pour devenir le professeur T_j dont la valeur de son objectif est définie par μ_T . La moyenne des étudiants est ensuite calculée pour chaque variable de design v pour former le vecteur μ_v . Le professeur T_i cherche à

augmenter le résultat de chaque étudiant par la différence entre sa valeur et la moyenne du groupe de la façon suivante :

$$Diff_Mean = r (\mu_T - T_F \mu_v) \quad (3.9)$$

où r est un nombre aléatoire entre 0 et 1 et T_F est un facteur heuristique qui peut prendre la valeur de 1 ou

2. Les nouvelles solutions sont ensuite calculées de la manière suivante :

$$X'_{i,j} = X_{i,j} + Diff_Mean \quad (3.10)$$

où $X'_{i,j}$ est une nouvelle solution i obtenue de la première phase à l'itération j . Si $X'_{i,j}$ est une solution moins bonne que $X_{i,j}$, alors $X_{i,j}$ remplace $X'_{i,j}$ lors de la phase suivante.

3.6.2. Apprentissage

Lors de la seconde phase, les étudiants interagissent ensembles pour améliorer leur savoir en utilisant une approche similaire à des groupes de discussion. Les solutions $X'_{i,j}$ seront comparées ensemble aléatoirement. Chaque solution $X'_{i,j}$ est comparée avec une solution aléatoire X'_p parmi la population ($X'_{i,j} \neq X'_p$) de la façon suivante :

$$X''_{i,j} = \begin{cases} X'_{i,j} + r (X'_{i,j} - X'_{p,j}), & \text{si } f(X'_{i,j}) < f(X'_{p,j}) \\ X'_{i,j} + r (X'_{p,j} - X'_{i,j}), & \text{si } f(X'_{p,j}) < f(X'_{i,j}) \end{cases} \quad (3.11)$$

où $X''_{i,j}$ est la nouvelle solution obtenue et r est un nombre aléatoire entre 0 et 1. Si $X''_{i,j}$ est une solution moins bonne que $X'_{i,j}$, alors $X''_{i,j}$ est rejetée et $X'_{i,j}$ sera la solution finale retenue dans la nouvelle population. Ces deux phases sont répétées jusqu'à atteindre le critère d'arrêt fixé par l'utilisateur.

Chapitre 4 : Article #1

Titre :

Optimization of Thermoelectric Generators to Recover Heat from a Hot Surface while Maintaining its Temperature and Heat Flux Distributions

Co-auteurs :

Mathieu Allyson-Cyr, Louis Gosselin

Journal :

Applied Thermal Engineering, submit in May 2018

Résumé

Un générateur thermoélectrique (TEG) est un appareil utilisé pour convertir directement la chaleur en électricité. Dans plusieurs secteurs d'activité, l'idée d'attacher des TEGs sur une surface chaude sans affecter thermiquement le système auquel il s'y rattache peut être intéressante. Dans cet article, la performance de ces appareils est optimisée lorsqu'ils sont appliqués sur une surface en maintenant la température et le flux de chaleur constants. La distribution des modules thermoélectriques, leurs géométries et leurs matériaux ainsi que les caractéristiques d'un collecteur thermique à la jonction froide sont optimisés pour maximiser la puissance électrique en sortie. Les résultats montrent que pour maintenir la jonction chaude à une température et un flux de chaleur constant, le courant électrique ainsi que le nombre de modules thermoélectriques ne peuvent pas varier indépendamment. Une optimisation multi-objectifs est également présentée dont la puissance nette est maximisée alors que le nombre de modules thermoélectriques et la surface totale d'échange du collecteur thermique sont minimisés. Les résultats montrent qu'il existe plusieurs designs qui génèrent une puissance légèrement plus faible, mais avec une surface d'échange et un nombre de modules réduits (solution moins coûteuse). Seulement quelques dispositions de connections électriques pour le TEG et de géométrie pour l'échangeur de chaleur ont été retenues lors de l'optimisation.

Abstract

Thermoelectric generators (TEG) can be used to convert waste heat to electricity. In many applications, the idea of attaching TEGs to an existing hot surface without thermally affecting the system to which it belongs is appealing. In this work, we optimize the performance of such TEGs, subjected to fixed surface heat flux and temperature distributions. The distribution of modules, their shapes and materials, and the features of the cold-side heat exchanger are optimized to maximize the power output. Results show that to keep fixed hot-side heat flux and temperature, the electric current and number of modules cannot vary independently. A multi-objective optimization problem is also introduced and solved where the net power output is maximized while minimizing the number of modules and the overall surface area. Results showed the existence of several solutions providing a slightly lower power output, but with less surface area and modules (i.e., cheaper solutions). Only a few of the possible layouts were selected for the TEG electric connections and for the heat exchanger pattern.

4.1. Introduction

Access to energy is a vital socio-economic lever for our societies, but it has a strong environmental footprint. In the industrial sector, studies reveal that nearly half of the energy input is actually wasted as residual heat in the United-States [13], and similar figures are found in other countries. Recovering this waste heat thus offers a huge opportunity to address energy related environmental issues and enhance the productivity of the industry. Among the different options, thermoelectric devices have attracted a lot of attention over the years because they create virtually no greenhouse gas emissions, they have no mobile parts, no working fluids and are highly durable.

Thermoelectric devices are composed of a pair of semi-conductor pellets doped N and P usually connected electrically in series and thermally in parallel. Many of them can be assembled together to form a thermoelectric generator (TEG) which has the ability to transform heat to electricity directly by using Seebeck effect [1].

However, their rather low energy conversion efficiency ($\sim 1\text{-}5\%$) – which is mainly due to the low figure-of-merit ($ZT \sim 1$) of actual thermoelectric materials – often makes them not economically profitable in industrial sectors. In other words, a large portion of the heat input is still lost as heat on the cold side of a TEG. In order to improve the efficiency, a thermoelectric generator can be sandwiched between the hot and cold sides of a heat exchanger (HE) for improved heat transfer through the device and to recover portions of the remaining heat at the same time [14]. Refs. [15] and [16] have studied and optimized thermoelectric heat exchanger for improved performance. The design can be optimized for lower cost and higher power output with smaller size by reducing the number of modules and the heat exchanger dimensions. Moreover, the electric configuration of thermoelectric heat exchanger seems to be less important on thermoelectric performances than thermal parameters. In addition, the reduction of thermal resistance between TEG and HE walls proved to be an important step in TEG improvement [17][18][19].

Other types of HE geometry such as helical thermoelectric heat exchanger presented by Ref. [20] yielded promising results. The irregular configuration increased the surface of exchange with almost identical energy conversion efficiency compared to traditional configuration, leading to higher power output that could be potentially useful for some applications. There is also a trade-off to be made between optimal heat exchanger and thermoelectric generator efficiency. Ref. [21] studied liquid-to-liquid thermoelectric heat exchangers and found that heat exchanger efficiency of 0.5 maximized thermoelectric power generation.

The capability of thermoelectric devices to be used for low power generation makes them useful as portable devices using body heat such as thermoelectric watch [22]. They can also be used in the industrial sector for low-temperature waste heat recovery technology.

Thermoelectric devices are also considered for high-temperature waste heat recovery usages in sectors such as automobile, aircraft, ship and light metal industry [23]. These sectors involve important amount of waste heat carried away by exhaust pipes that could be harvested using thermoelectric heat exchanger devices [24]. Thermoelectric heat exchanger can also be installed around chimneys to recover waste heat. However, the temperature drop along the chimney affect the thermoelectric generation drastically and must be considered for this type of application [25].

In automobile combustion engines, most of the energy is wasted as heat and evacuated by the engine coolant (30%) or the exhaust gases (40%) [26]. Thermoelectric devices can be installed around the exhaust pipe or on the radiator while the other side of the TEG is either cooled by air or by an extension of the cooling system [27]. It is reported that TEGs can only reach a maximal efficiency of 5% in optimal operating conditions. Nonetheless, TEGs can be easily installed on the exhaust pipe and could potentially lead to better fuel efficiency. Ref. [28] optimized the cooling unit geometry to improve the heat transfer while minimizing the pressure drop. This improvement of the cooling unit leads to higher thermoelectric generation. Ref. [29] proposed a segmented folded-shaped exhaust pipe for a more uniform temperature distribution and, thus, better heat transfer over the TEG. It is noted, however, that higher engine speed results in higher pressure drop as well, making the design not suitable for some cases. Another type of exhaust duct improvements has been proposed in Ref. [30] by adding aluminium packing within the duct directly. The design has been tested and higher thermoelectric generation can be achieved at lower pressure drop.

In the industrial sector, TEG could be used for cogeneration applications. Ref. [31] studied a cogeneration system using thermoelectric generation to recover exhaust heat from diesel engines and gas turbines in the manufacturing industry of Thailand. However, this approach is not economically profitable at the moment because of material limitations. Thermoelectric heat exchanger could help to recover waste heat in other systems and applications such as steam boilers, aluminium production, molten metal furnaces and glass industry [32][33].

One of the “dreams” or visions of many is to achieve a TEG system design that can be directly attached or glued to hot surfaces present in a variety of processes and systems. The necessity to maintain a certain temperature gradient through the TEG modules demands a proper cooling mechanism on the cold side of the TEG. Another important challenge of such systems is that very often the heat source is subject to specific

thermal constraints such as a specified cooling rate and/or specified temperature profiles. For example, this is the case for electrolysis cells in aluminium smelters where the cooling rate on the sidewalls is specified by the cell design in order to preserve the integrity of the solid ledge within the cell. This ledge protects the sidewall from early failure [34][35][36] and could melt if the heat transfer rate is increased or occupy too much space in the pot if the heat transfer rate is decreased. Similar thermal constraints are also everywhere (e.g., engines, etc.). In literature, such a consideration has not been taken explicitly into account in TEG design procedures. In this paper, the potential of thermoelectric generators applied to a hot surface is investigated for waste heat recovery under temperature and heat flux constraints. In other words, heat flux and temperature profiles for the heat source surface are imposed in the proposed TEG design approach. A 1D analytical model is first developed in Section 4.2 and then integrated into a 2D model used to optimize a TEG system covering a surface in Section 4.3. In Section 4.4, the design optimization method is tested with different test cases. A cold-side heat collector is introduced in the model in Section 4.5 and optimal designs are reported in Section 4.6.

4.2. Thermoelectric 1D model

The purpose of this model is to introduce the main TEG equations and introduce the impact of the thermal constraints on the TEG optimization. Consider a system with a hot surface (e.g., engine, electrolysis cell, etc.) dissipating heat to the environment which is at the temperature T_c . The surface has a temperature distribution T_{JH} and one would like to install a thermoelectric generator (TEG) on it. As introduced above, it is assumed that one would like to keep the same heat dissipation rate even once the TEG is attached to the surface. By adapting the number and distribution of the modules and the electric current, the power output of the TEG system can be maximized while maintaining the heat flux and surface temperature to specific values. In this section, the one-dimensional case is treated to introduce and illustrate the problem. This 1D model will be used in a 2D model in the next section.

4.2.1. Mathematical model

A schematic representation of a thermoelectric module is shown in Fig 4.1a. In the present model, both p- and n-type legs of TE modules are assumed to have the same dimensions. The area-to-length ratio of each TE pellet is fixed to 0.0001 m. T_{JH} and T_{JC} represent the temperature at the hot and cold junctions respectively. The heat flux entering and leaving the modules are expressed by q''_H and q''_C . The induced electric current is represented by the variable I while R_{ext} is the external electric resistance to which the TEG is connected. The ambient temperature on the cold side of the thermoelectric module is T_c . Multiple thermoelectric modules

are electrically connected together in series to form the TEG. The placement of the TEG over the surface of the heat source is presented in Fig. 4.1b.

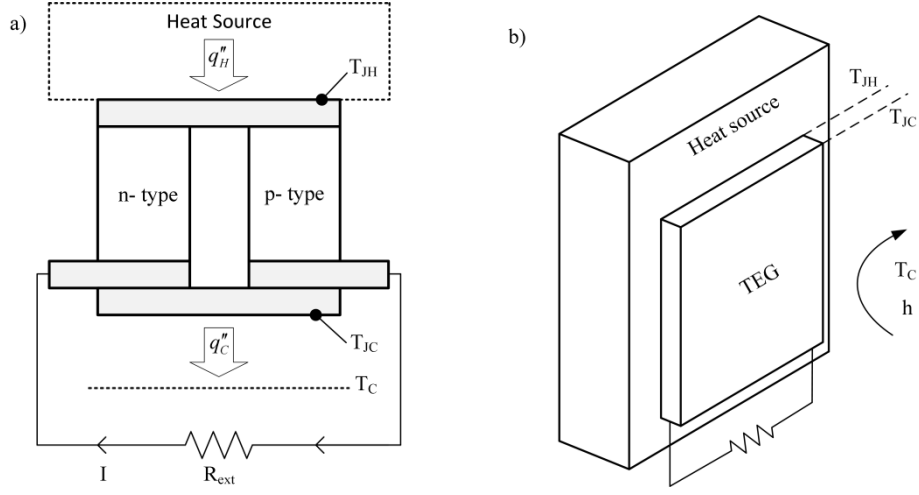


Figure 4.1: Schematic representation of a) single thermoelectric module and b) TEG disposition with respect to the heat source.

As will be shown below, the electric power can be optimized for specific values of q_H'' and T_{JH} . This can be achieved by adapting the number of modules and the electric current properly.

The steady-state governing equations describing the performance of a thermoelectric module consist in energy balances including conduction, as well as Seebeck and Joule effects. The heat flux entering and leaving the module are expressed as [37]:

$$q_H'' = \frac{n}{A} \left(S I T_{JH} - \frac{R I^2}{2} + K (T_{JH} - T_{JC}) \right) \quad (4.1)$$

$$q_C'' = \frac{n}{A} \left(S I T_{JC} + \frac{R I^2}{2} + K (T_{JH} - T_{JC}) \right) \quad (4.2)$$

where $K = k A_{TE} / L_{TE}$ and $R = \rho L_{TE} / A_{TE}$. Note that k and ρ are the thermal conductivity and the electrical resistivity respectively. Also, the ratio A_{TE} / L_{TE} is the surface-to-length ratio of the thermoelectric module. The electric power produced by the TEG is obtained by subtracting q_C'' from q_H'' :

$$P = A(q_H'' - q_C'') = n(SI(T_{JH} - T_{JC}) - R I^2) = R_{ext} I^2 \quad (4.3)$$

The boundary condition at the hot side is a prescribed temperature T_{JH} given by the source. At the cold side, an effective convection cooling coefficient is applied:

$$q_c'' = h(T_{JC} - T_c) \quad (4.4)$$

Eqs. (4.1), (4.2) and (4.4) can be solved to find the unknown parameters required to calculate the electric power output. The solution is given by:

$$T_{JC} = \frac{0.5(nRI^2 + 2hAT_c + 2nKT_{JH})}{(hA + nK - nSI)} \quad (4.5)$$

$$q_c'' = \frac{0.5hn(RI^2 + 2SIT_c + 2K(T_{JH} - T_c))}{(hA + nK - nSI)} \quad (4.6)$$

$$q_H'' = \frac{0.5n(nRSI^3 - I^2(2nT_{JH}S^2 + hAR + 2nRK) + hA(2T_{JH}SI + 2K(T_{JH} - T_c)))}{A(hA + nK - nSI)}. \quad (4.7)$$

Finally, the efficiency of the TEM is given by:

$$\eta = \frac{P}{Aq_H''}. \quad (4.8)$$

Table 4.1: Characteristics and properties of the model

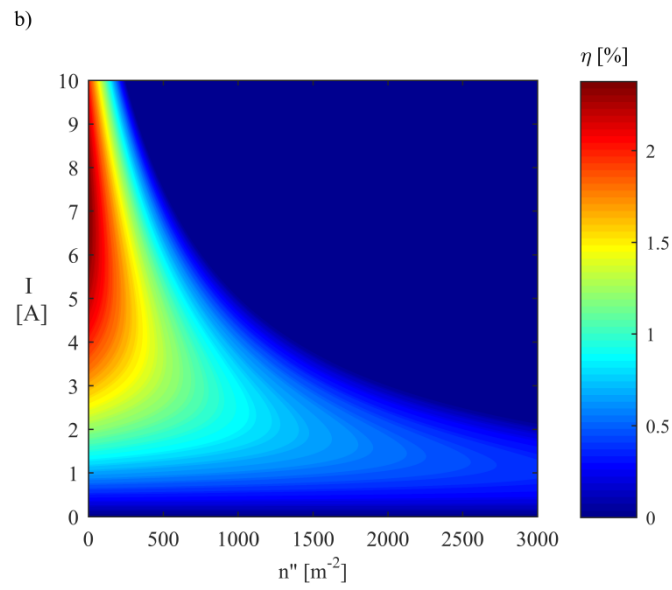
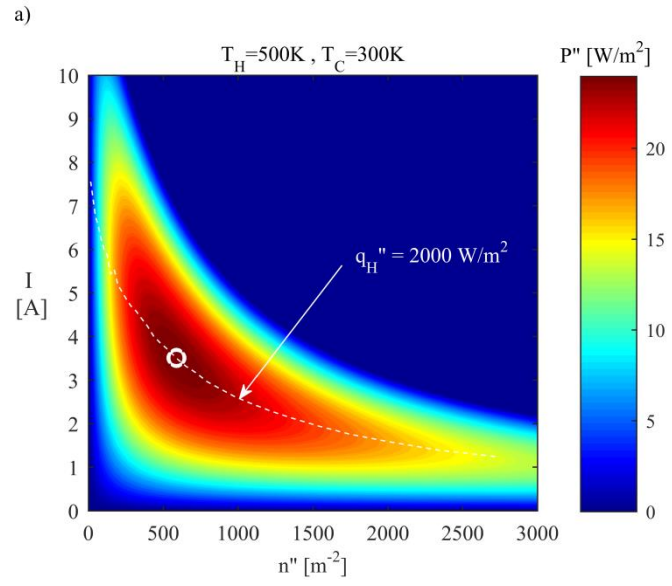
| | |
|--|--------------------------|
| Pellet dimensions (L×W×H) | 5×5×2.5 mm |
| Surface of exchange (A) | 1 m ² |
| Electric resistivity (ρ) [15] | 1.2×10 ⁻⁷ Ωm |
| Thermal conductivity (κ) [15] | 1.5 W/mK |
| Seebeck coefficient (s) [15] | 2.2×10 ⁻⁴ V/K |
| Cold side convection coefficient (h) | 20 W/m ² K |
| Ambient temperature (T _c) | 300 K |
| Hot Side Temperature (T _H) | 500 K |

4.2.2. Results

A parametric analysis was performed based on the 1D mathematical model described in the previous subsection. For the purpose of this analysis, the values that were used for h , A , T_c and the material properties are

shown in Table 4.1. Note that other values will be used later in this paper. For prescribed temperature T_{JH} and T_{JC} , the number of modules n and the current I were varied and the power output, hot side heat flux and efficiency were calculated and are reported in Figure 4.2. The power and heat transfer rate are expressed per unit of surface area. The zone in the upper section of the figures corresponds to situations in which it is not possible to generate any power. The power output in Figure 4.2a exhibits a maximum which is indicated by a small white circle. This result is coherent with literature [38]. A maximum power of 24.2 W/m^2 is achieved with $n = 588$ and $I = 3.5 \text{ A}$ with the current settings. In Figure 4.2b, the efficiency is shown as a function of n and I . There is also an optimum which is visible, but not at the same position as for the power output. This comes from the fact that in this first parametric sweep, no constraint on the heat flux was invoked. In fact, one can see in Figure 4.2c that the hot side heat flux varies significantly as a function of n and I , between 0 and $\sim 3000 \text{ W/m}^2$. This emphasizes the interest to design TEG systems to specifically preserve the heat balance of the hot surface on which it is attached; otherwise it could be detrimental to the performance of the overall system by cooling it too much or preventing heat from being dissipated properly.

Therefore, a constraint on the hot side heat flux was then invoked and added to the model. In other words, q''_H was not free to vary anymore, but was specified. This corresponds to “walking” along a line of constant q''_H in Figure 4.2c rather than covering the entire design space. As can be seen in Figure 4.2c, fixing the hot side heat flux imposes a relation between the two independent variables n and I , which can no longer be varied independently. For each given value of q''_H , the maximal work output was found by considering all possible combinations of n and I to achieve that specific heat flux. The result is shown in Figure 4.3a, and the corresponding optimal current and number of modules are reported in Figure 4.3b. One can see that the optimal number of modules increases with the imposed heat flux, while the optimal current is reduced. It is interesting to note that the maximal power output does not increase monotonically with the hot side heat flux. In fact, it starts to decrease at some point due to the combinations of n and I (i.e. large n , small I) required to respect the heat flux constraint which prevents a satisfactory performance in terms of power output. In other words, adding the additional constraint that the hot side heat flux (in addition to the hot side temperature) must be preserved when the TEG is attached to the surface tends to deteriorate the waste heat recovery. The optimal designs for fixed hot side heat fluxes are shown in Figure 4.2 by the dashed line.



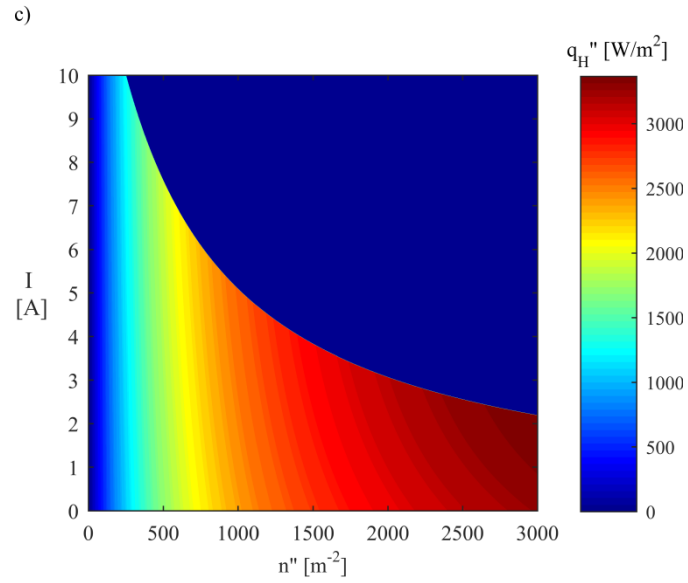
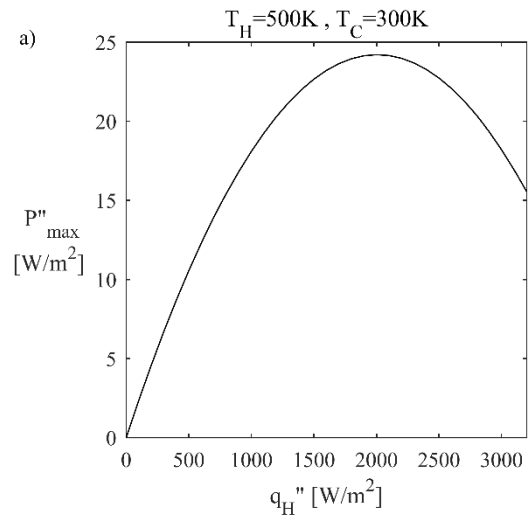


Figure 4.2: Thermoelectric performance with no imposed constraint on the hot side heat flux: a) Electric power output, b) thermoelectric efficiency and c) Hot side heat flux as a function of the number of modules and electric current.



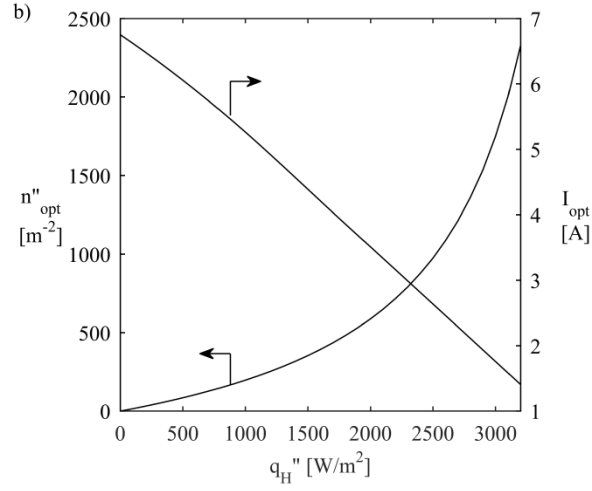


Figure 4.3: a) Maximal electric power obtainable and b) corresponding optimal combinaison of number of modules and current for a constrained value of the hot side heat flux.

Finally, the impact of the boundary condition on the cold side of the TEG was studied. For a given hot side heat flux value (2000 W/m^2 in the present scenario), the convection coefficient h and cold sink temperature T_c were varied. In each case, the optimal combination of current and number of modules (i.e., combination maximizing the power output while respecting the heat flux constraint) was found. One can observe the maximized power output in Figure 4.4a as a function of h and T_c . Essentially, as the cooling becomes more efficient (i.e., large h and/or small T_c), it is possible to produce more power. Note that the optimal number of modules tends to decrease when the cooling is better, while the optimal current increases.

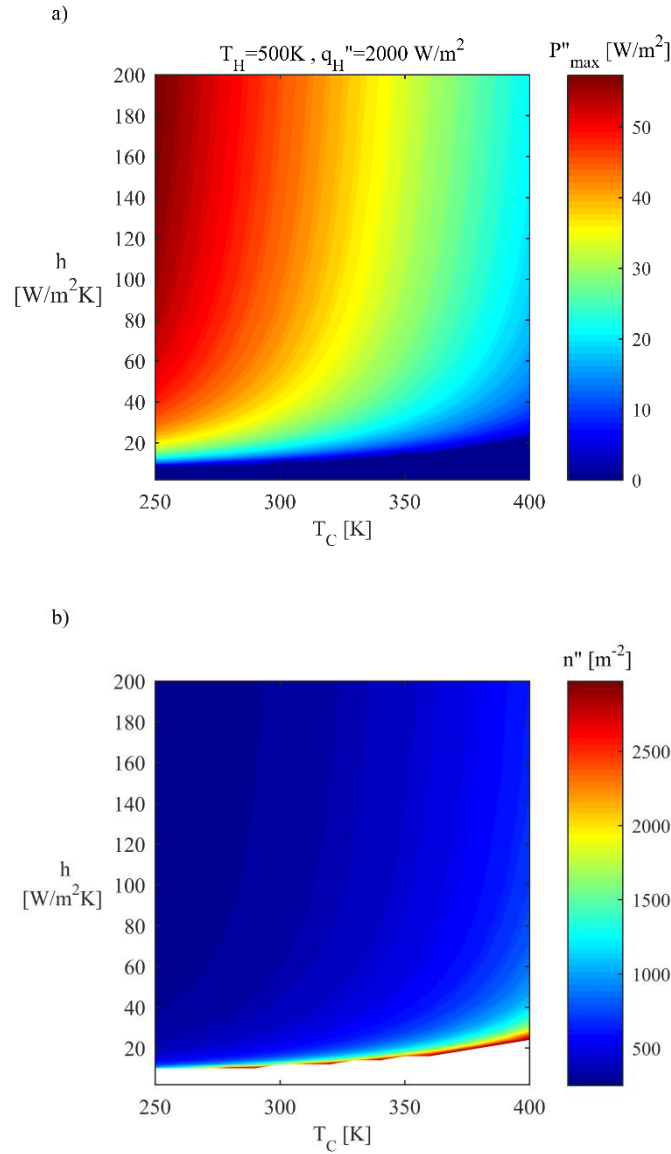


Figure 4.4: a) Maximized power output and b) required number of modules for fixed hot side heat flux and temperature as a function of the cold side convection coefficient and far field temperature.

4.3. Formulation of optimization problem for 2D TEG model

In this section, a more elaborate model is developed based on the 1D mathematical model of the previous section. The model, as shown in Figure 4.5, consists in two superimposed layers where the TEG is discretized into multiple control surfaces. The source has a hot side temperature distribution and a heat flux distribution rather than uniform values. However, the cooling parameters (i.e., T_c and h) remain uniform in the present section. This assumption will be released later. The 1D model of the previous section is applied to each

discretized control surface of the TEG. This is equivalent to having all modules within a small control surface connected in series. However, different electric connections between the different control volumes were modeled. A list of predefined possible connection patterns has been elaborated. For the sake of illustration, a series of patterns is illustrated schematically in Figure 4.6 for a 12×12 mesh. Each color represents a parallel segment in the TEG connection layout while all cells of the same color are connected in series. As shown in Figure 4.6, grid #1 is composed of 12 parallel segments forming 1 vertical line of cells per segment; grid #2 is composed of 6 parallel segments with 2 vertical lines of cells per segment and so on until every cell are connected in series in grid #12. Grids #13 to #24 are similarly generated but every parallel segment is generated from horizontal line of cells. For this specific 12×12 discretization, a total of 24 grids were thus generated. For cells connected in series, the layout is not important since the current is the same in every module of this specific segment.

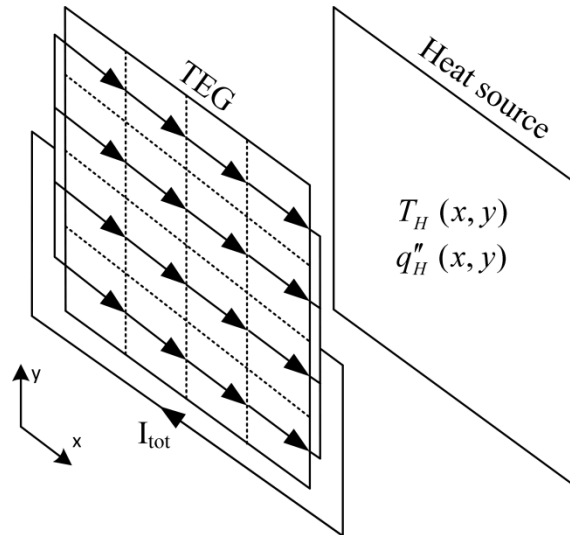


Figure 4.5: 2D representation of the TEG model with electric connection layout example.

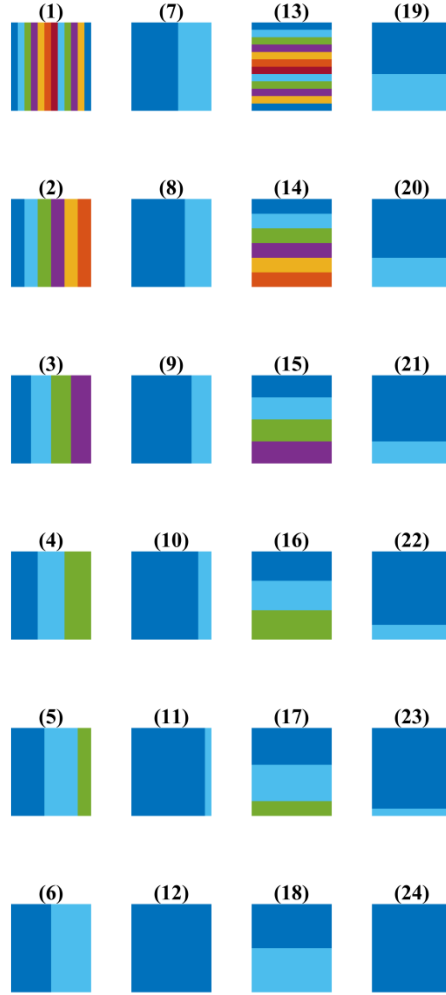


Figure 4.6: Total set of predefined TEG electric connection grid layout generated for 12×12 mesh.

For each cell of the mesh, a single p- and n-type thermoelectric material combination can be selected. In the present model, a list of three material combinations was considered and the properties (Seebeck coefficient s , electric resistivity ρ and thermal conduction k) were evaluated at the average temperature of the legs, i.e.

$(T_{JH} + T_{JC})/2$. The properties were evaluated from the following correlations:

$$\begin{aligned}
 S_{mat,i} &= a_{mat,i} T^3 + b_{mat,i} T^2 + c_{mat,i} T + d_{mat,i} \\
 \rho_{mat,i} &= a_{mat,i} T^3 + b_{mat,i} T^2 + c_{mat,i} T + d_{mat,i} \\
 k_{mat,i} &= a_{mat,i} T^3 + b_{mat,i} T^2 + c_{mat,i} T + d_{mat,i}
 \end{aligned} \tag{4.9}$$

where i is material group selected (between 3 possibilities) and mat is the material type (P or N). The coefficients a , b , c and d of the three selected material combinations are given in Table 4.2.

Table 4.2: Temperature dependant properties of selected materials

| # | Material | | a | b | c | d |
|---|-----------------|-------------|---------------------------|---------------------------|--------------------------|--------------------------|
| 1 | p-Type | S_{p1} | 8.3335×10^{-12} | -1.3273×10^{-8} | 6.3023×10^{-6} | -7.0396×10^{-4} |
| | Bi_2Te_3 | ρ_{p1} | -7.3559×10^{-13} | 6.1348×10^{-10} | -6.3483×10^{-8} | -1.7788×10^{-6} |
| | [39] | k_{p1} | 1.5888×10^{-8} | -3.3157×10^{-6} | -2.1177×10^{-3} | 1.5775 |
| | n-Type | S_{n1} | -3.9819×10^{-12} | 7.3415×10^{-9} | -3.8211×10^{-6} | 3.9458×10^{-4} |
| | Bi_2Te_3 | ρ_{n1} | -6.8342×10^{-13} | 6.6613×10^{-10} | -1.5518×10^{-7} | 1.8059×10^{-5} |
| | [39] | k_{n1} | 2.1889×10^{-8} | -4.5960×10^{-6} | -4.5128×10^{-3} | 2.4815 |
| 2 | p-Type | S_{p2} | -6.8014×10^{-12} | 9.8276×10^{-9} | -4.1376×10^{-6} | 6.0053×10^{-4} |
| | $PbTe$ | ρ_{p2} | -4.1627×10^{-13} | 6.4523×10^{-10} | -2.8324×10^{-7} | 4.7443×10^{-5} |
| | [40] | k_{p2} | 6.5993×10^{-9} | -8.6306×10^{-6} | 1.4631×10^{-3} | 2.0723 |
| | n-Type | S_{n2} | 1.5724×10^{-11} | -2.2240×10^{-8} | 9.4361×10^{-6} | -1.4055×10^{-3} |
| | $PbTe$ | ρ_{n2} | -5.6866×10^{-13} | 8.6821×10^{-10} | 3.5943×10^{-7} | 5.3577×10^{-5} |
| | [40] | k_{n2} | -5.8721×10^{-8} | 1.0764×10^{-4} | -6.1955×10^{-2} | 1.3526×10^1 |
| 3 | p-Type | S_{p3} | -1.1002×10^{-12} | 2.1874×10^{-9} | -1.3340×10^{-6} | 3.9654×10^{-4} |
| | $CeFe_4Sb_{12}$ | ρ_{p3} | 0 | -2.2697×10^{-12} | 7.3360×10^{-9} | 3.5028×10^{-6} |
| | [41] | k_{p3} | -1.4359×10^{-9} | 2.6877×10^{-6} | -1.5653×10^{-3} | 2.9901 |
| | n-Type | S_{n3} | -3.3485×10^{-13} | 1.1127×10^{-9} | -1.111×10^{-6} | 1.5057×10^{-4} |
| | $CoSb_3$ | ρ_{n3} | 1.10159×10^{-14} | -3.4487×10^{-11} | 3.6462×10^{-8} | -2.3784×10^{-6} |
| | [41] | k_{n3} | -5.2863×10^{-9} | 1.9584×10^{-5} | -1.9299×10^{-2} | 9.3857 |

These selected materials perform better at different temperatures because of their respective dimensionless figure of merit zT . The figure of merit for a single TE module combining both p- and n- type semiconductors is express as:

$$zT = \frac{(S_p - S_n)^2}{\left[(\rho_p k_p)^{1/2} + (\rho_n k_n)^{1/2} \right]^2} T \quad (4.10)$$

where T is the average temperature of the thermoelectric module, (S_p, ρ_p, k_p) are the properties for the p-type material and (S_n, ρ_n, k_n) are the properties for the n-type material. The maximal zT values of material combinations #1, #2 and #3 is achieved at mean temperatures around 300 K, 600 K and 900 K respectively.

The objective function in the optimisation process is the global efficiency of the TEG obtained from Eq. (4.9) by summing the total electric power output from each cell of the mesh. The first design variable is the total current circulating in the TEG I_{tot} . Also, the electric connection layout is optimized by selecting the best pattern from the predefined list of possible layouts. Other design variables represent the material selection for

each cell of the mesh, i.e. one design variable per control surface. Finally, the last variables are the surface-to-length ratio of the legs for each thermoelectric material. In the end, the total number of variables is thus equal to $M+5$, where M is the number of cells in the mesh used to discretize the hot surface.

Since the problem includes both discrete and continuous variables, and because the objective function is not derivable with respect to some of the variables, it was decided to use a genetic algorithm in order to perform the optimization. The optimization toolbox 7.4 of Matlab has been used [6]. The main parameters of the GA are presented in Table 4.3. GAs use probabilistic methods. Consequently, two runs with identical parameters and settings can lead to different results. To ensure the repeatability of the results, optimization runs were performed 5 times. Only the best results among the 5 runs were kept and are presented in this paper.

Table 4.3: Main parameters of the single-objective GA

| |
|---|
| Individuals per population = 30 x number of variables |
| Crossover rate = 0.8 |
| Mutation rate = Default properties |
| Elitism = 5% of the population |
| Number of generations = 500 |
| Objective tolerance = 1×10^{-9} |

4.4. Optimized TEG design for maximal global efficiency

The optimization procedure described above was applied to two test case problems with a square surface of 25 m² and with the temperature and heat flux distributions shown in Figure 4.7a and Figure 4.7b. Both distributions from Figure 4.7 have maximal and minimal temperatures of 1000 K and 450 K respectively while the maximal and minimal heat fluxes are 15 kW/m² and 2 kW/m².

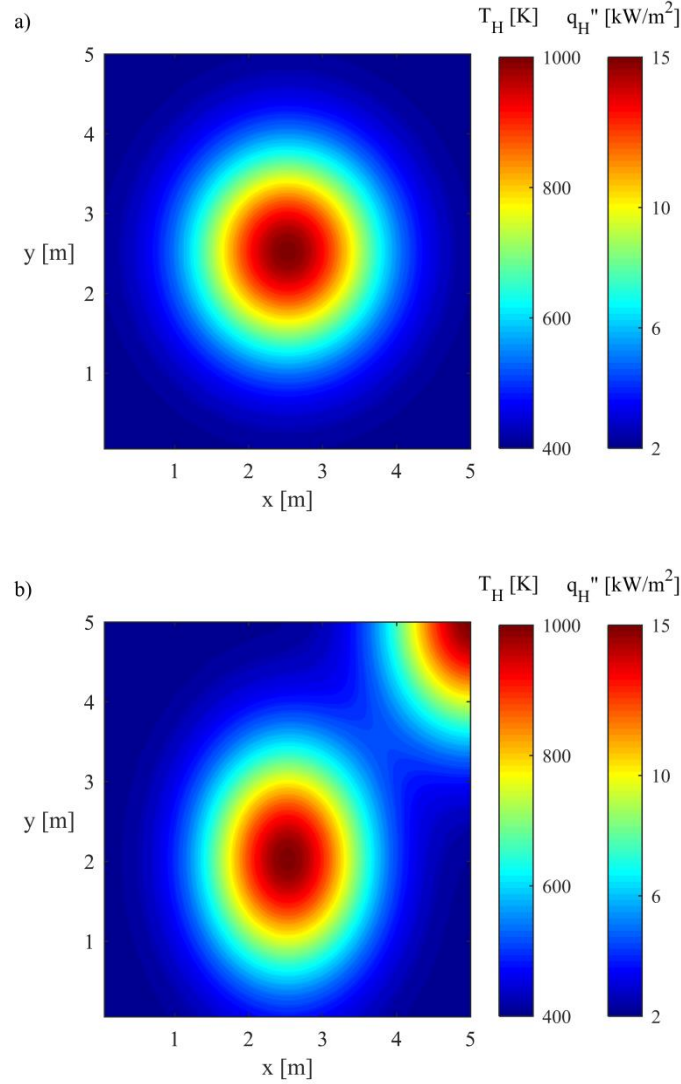


Figure 4.7: Hot-side temperature and heat flux distributions for: (a) the 1st test case, and (b) the 2nd test case.

The optimization results are reported for two grids (i.e., a coarser grid of 5×5 and a finer grid of 12×12) in Figures 4.8 and 4.9 for first (symmetric) and second (asymmetric) temperature/heat transfer distributions respectively. It is important to note that the number of variables increases with the mesh refinement. The number of variables goes from 30 with the 5×5 mesh to 149 with the 12×12 mesh. From Figure 8, the maximized global efficiency with the 5×5 mesh is 4.7% with a total current of 4.77 A, a total number of modules of 24,685 and surface-to-length ratio for module with material #1, #2 and #3 of 9.9 mm, 4.8 mm and 1.7 mm respectively. The distribution of the modules on the surface in Figures 4.8c and 4.9c is presented per unit of surface area (n'') for an easier comparison between the finer and coarser grids which have different cell dimensions. The maximized global efficiency for the equivalent problem, but with the 12×12 mesh, is

4.6% with a total current of 9.41 A, a total number of modules of 25,668 and surface-to-length ratio for module with material #1, #2 and #3 of 9.9 mm, 5.1 mm and 1.8 mm respectively. Similarly in Figure 4.9, the global efficiency with the coarse mesh is 4.7% with a total current of 5.04 A, a total number of modules of 26,610 and surface-to-length ratio for module with material #1, 2 and 3 of 10.0 mm, 5.6 mm and 1.9 mm respectively. The global efficiency with the finer mesh is 4.8% with a total current of 3.03 A, a total number of modules of 43,780 and surface-to-length ratio for module with material #1, 2 and 3 of 10.0 mm, 3.3 mm and 1.2 mm respectively. It is therefore visible that for these test cases, refining the grid has only a limited impact on the global efficiency of the TEG. In other words, the “granularity” of the optimized solution does not have to be very small. In addition, the surface-to-length ratio for each material is essentially the same for the two meshes. However, in Figure 4.9b, the number of modules for the finer mesh is a lot higher than for the coarser mesh. This result highlights that design with similar performances might require more or less thermoelectric modules.

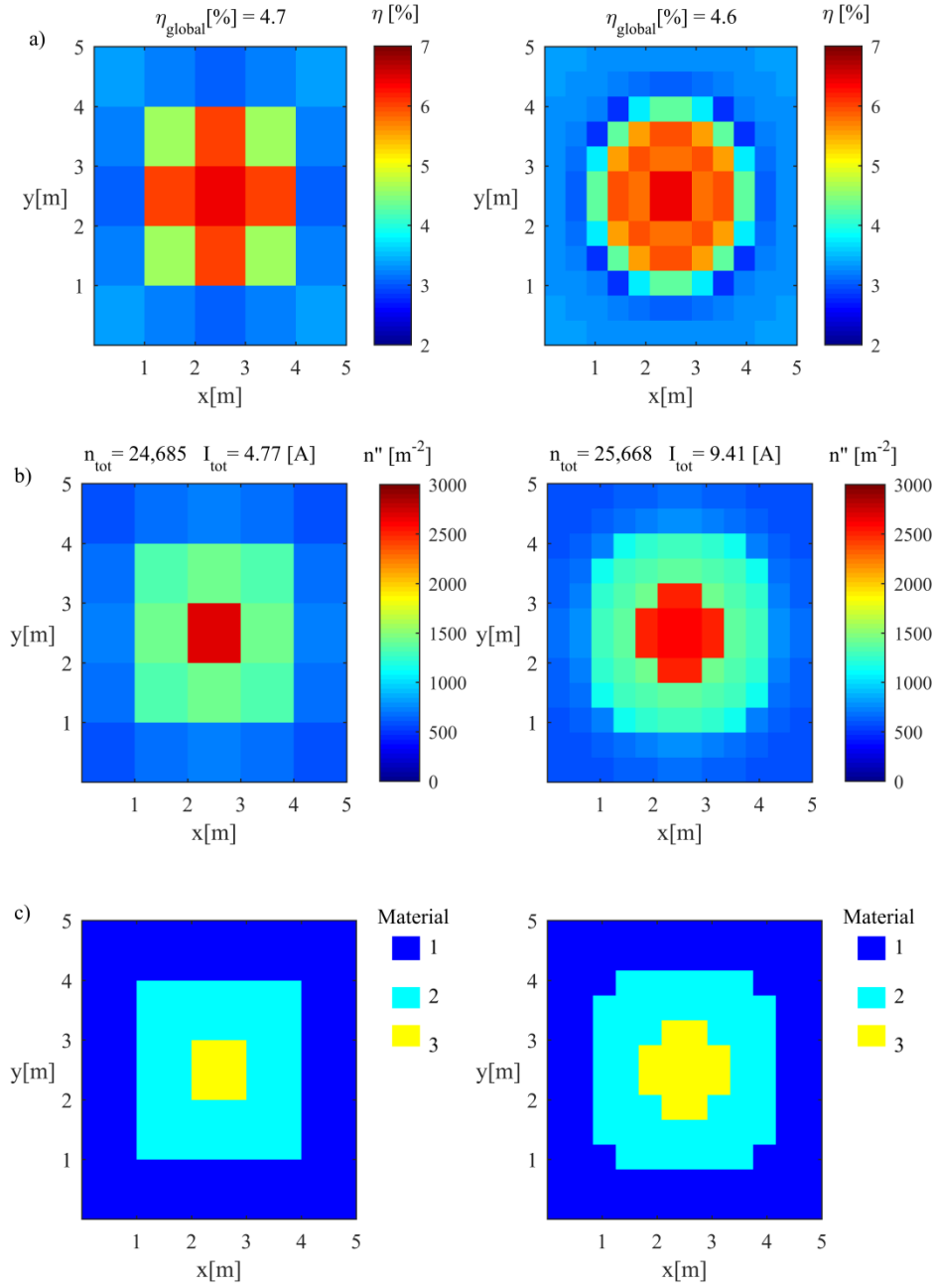


Figure 4.8: Optimization results with a 5×5 mesh (left) and a 12×12 mesh (right) for the first test case
a) TEG global efficiency, b) density of modules, c) material distribution.

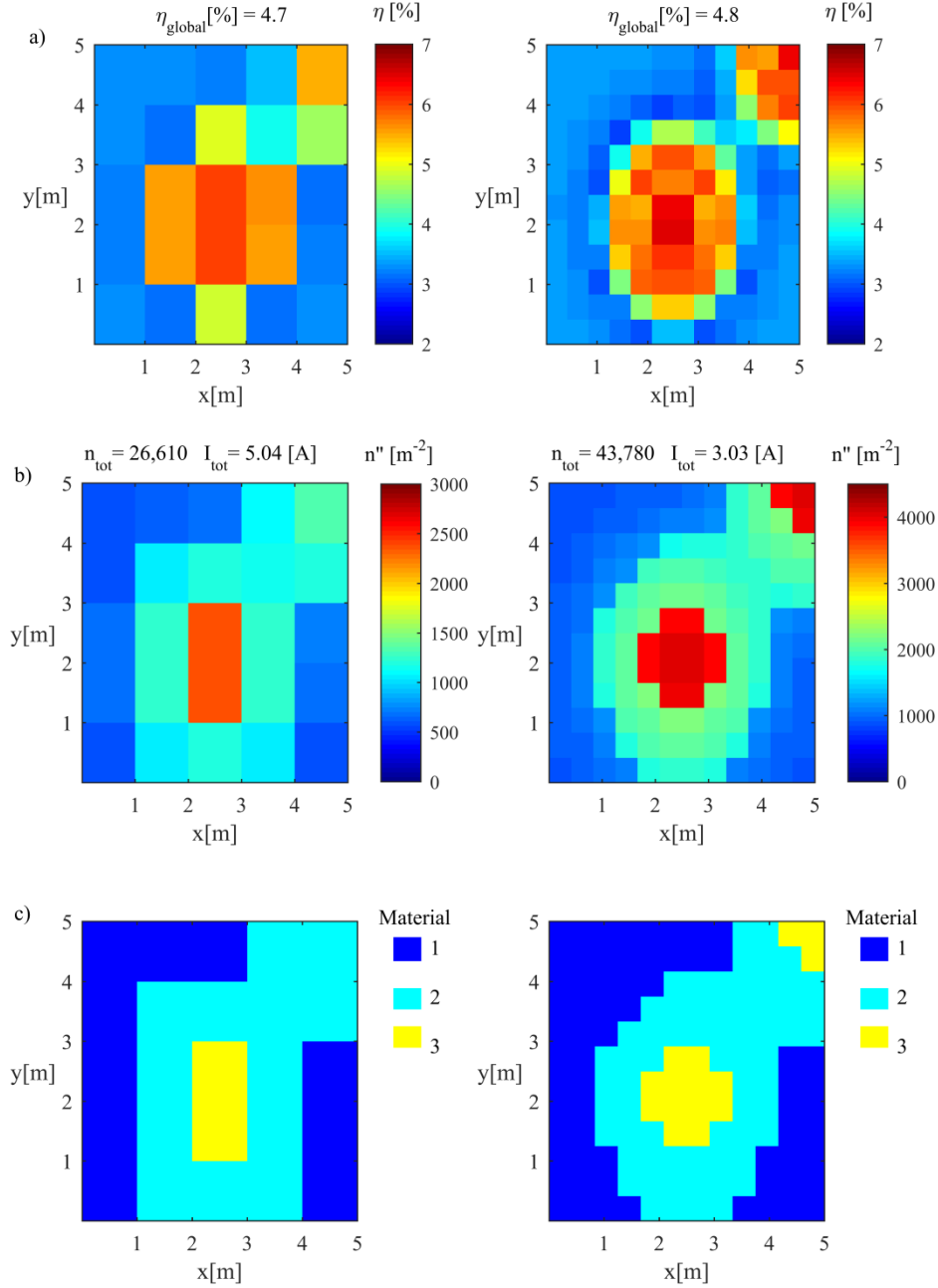


Figure 4.9: Optimization results with a 5×5 mesh (left) and a 12×12 mesh (right) for the second test case for a) TEG global efficiency, b) density of modules, c) material distribution.

Therefore, a slightly different optimization approach was also investigated. In fact, since the purchase cost of the TEG is related to the total number of modules, a multi-objective optimization solver (NSGA-II) was used to simultaneously minimize the number of modules and maximize the power produced. The algorithm was stopped after 500 generations and a population of 30 times the number of variables was used. As before, the genetic algorithm was run several times, and a non-dominated sorting was performed with the final solutions

obtained for all the runs. The same test cases as above were used with a 12×12 mesh. The resulting Pareto front is reported in Figure 4.10 for both heat transfer distributions. The color in Figure 4.10 represents different electric connection grid layouts from Figure 4.6 selected for each solution on the Pareto front while the white square is the maximal thermoelectric power output solution from Figure 4.8 and Figure 4.9 when only one objective was considered. In Figure 4.10, the designs with the highest thermoelectric power output are also the ones with the highest number of thermoelectric modules for both heat transfer distributions. From Figure 4.10b, it can be seen that the number of modules of the maximal power output design is a lot higher than for the designs obtained from multi-objective optimization. From Figure 4.10, one can see that different thermoelectric grids could lead to lower number of modules but also to a lower thermoelectric power output. However, for higher thermoelectric power output, the choice of the thermoelectric grid can reduce the number of modules considerably while the power output is less affected. Therefore, it might be beneficial from an economic point of view to choose a design with a slightly lower power output to reduce the required number of thermoelectric modules. Note that some of the thermoelectric grid numbers are selected at different power values. In these cases, the main design difference is the chosen aspect ratio and total current.

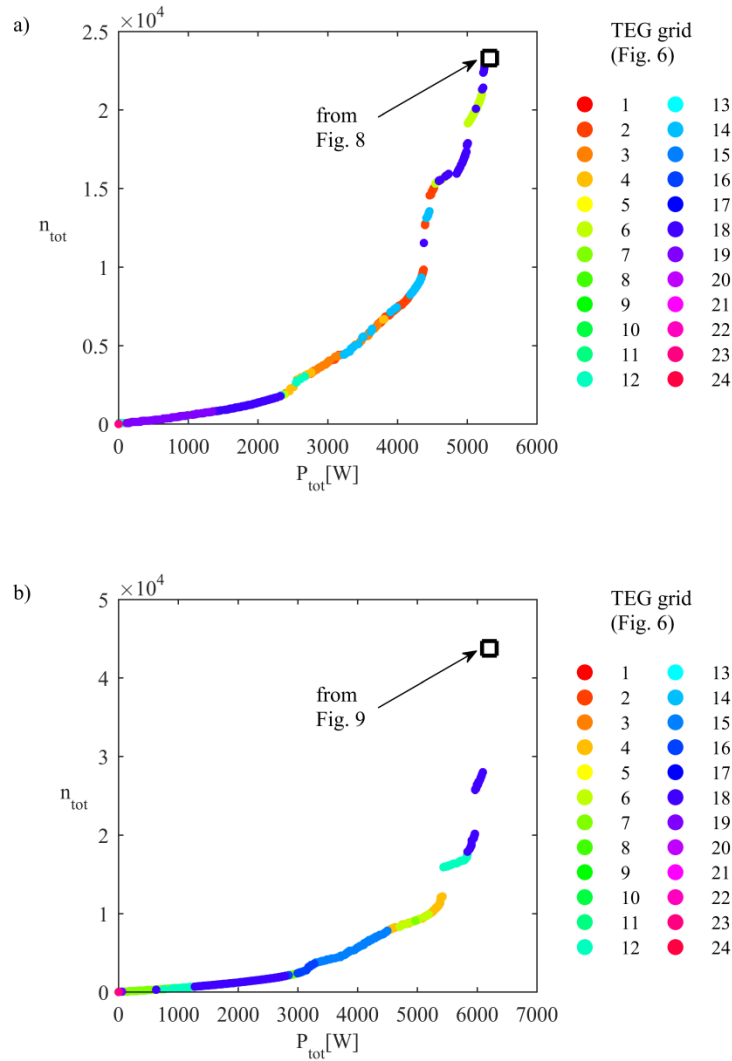


Figure 4.10: Pareto front of the TEG solutions (12 × 12 mesh) with minimized number of modules and maximized power output for a) 1st test case and b) 2nd test case.

4.5. Formulation of optimization problem for 2D TEG with heat collector

4.5.1. Heat transfer calculation

The cooling related parameters were uniform in the model of the previous section. In practice, as explained before, it might be necessary to use some cooling pad or loop to maintain an effective temperature difference between TEG junctions. In such cases, the thermoelectric generator is sandwiched between the source and a heat collector (cold side). As a result, new parameters and design variables emerge and need to be added to

the model such as the type of coolant, heat exchanger geometry, inlet coolant temperature, heat capacity rate \dot{C} and overall heat transfer coefficient U of the cooling system. The model, as shown in Figure 4.11, consists now in three superimposed layers where the TEG is sandwiched between the source and the heat collector. Both the TEG and the heat collector are discretized into multiple control surfaces.

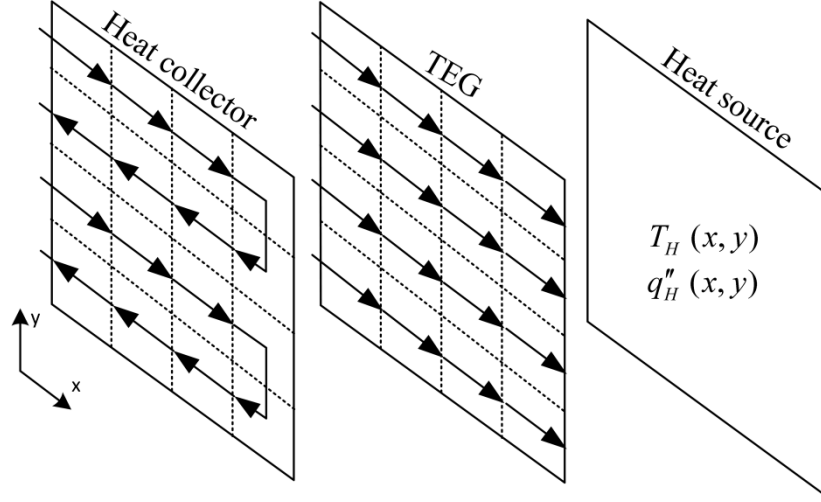


Figure 4.11: 2D representation of combined TEG and heat collector model.

Similarly to the electric connection grids introduced before, different flow grids connecting the different cells of the mesh were modeled for the heat collector. In this work, the same grid layouts as for the electric connections (see Figure 4.6) were used for the coolant flow layouts. However, in this case, the connection layout between cells in series is important because the path chosen affects the evolution of the coolant temperature. Therefore, for each heat collector grid of Figure 4.6, the flow path can be arranged into multiple possible ways. In this paper, the flow layouts for a given arrangement of Figure 4.6 are limited to two possibilities and an example is shown in Figure 4.12 with grid #6 (from Figure 4.6). For each parallel segment, the flow from cell to cell can either go from left to right (horizontal pattern) as shown in Figure 4.12a or from top to bottom (vertical pattern) as shown in Figure 4.12b. As a result, a total of 48 coolant flow layouts are possible for 12×12 mesh (2 for each grid).

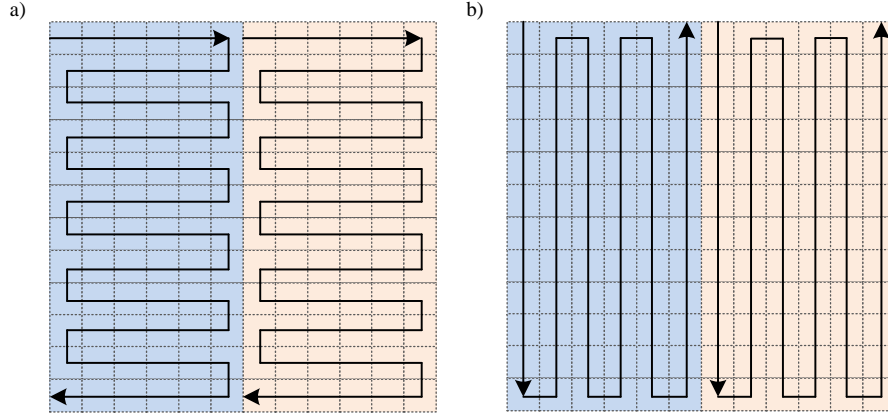


Figure 4.12: 2D representation of the coolant flow patterns in the heat collector with 12×12 mesh for grid #6 (Figure 4.6) set in a) horizontal pattern and b) vertical pattern.

With the addition of the heat collector to the model, the cold side temperature in each cell of the mesh is now unknown. Consequently, another equation must be added to the model and the cold side temperature T_c now increases as the coolant flows along the possible heat collector flow paths. An energy balance for a cell of the mesh within the heat collector yields:

$$q_c = C (T_c^{out} - T_c^{in}) \quad (4.11)$$

where T_c^{in} and T_c^{out} are the inlet and outlet fluid temperature for a given cell. In order to solve the problem, the temperature is calculated by following the flow path from cell to cell in order to obtain T_c^{out} which becomes T_c^{in} for the next cell and so on. The entrance temperature of the heat collector is set at 300 K in the present test case.

In addition, Eq. (4.4) must be replaced to represent the heat transfer between the TEG cold side and the heat collector. This new equation is given by:

$$q_c = U A_T \Delta T_{lm} \quad (4.12)$$

where A_T is the total cold side surface of exchange in the control volume and ΔT_{lm} is the log mean temperature difference:

$$\Delta T_{lm} = \frac{(T_{JC} - T_c^{in}) - (T_{JC} - T_c^{out})}{\ln \left(\frac{(T_{JC} - T_c^{in})}{(T_{JC} - T_c^{out})} \right)} \quad (4.13)$$

Hence, Eqs. (4.1), (4.2), (4.12) and (4.13) form the new system to solve. The heat collector flow grid sets the order in which each cell of the mesh is solved to calculate the distribution of the cold side temperature T_c .

4.5.2. Pumping power calculation

This section presents the model to evaluate the power required to generate the flow in the heat collector on the cold side of the TEG. In the test case, the fluid selected for the collector is standard air. The geometry of the heat collector in a given cell of the mesh must be adapted in such a way that it allows to extract the exact amount of heat that has to be removed to maintain a proper energy balance. For the sake of simplicity, conduction within the heat exchanger wall, contact resistance and surface fouling were neglected. The overall heat transfer coefficient in the heat collector can thus be expressed as:

$$\frac{1}{UA_T} = \frac{1}{\eta_o \bar{h} A_T} = \frac{1}{\left(1 - \frac{NA_f}{A_T} (1 - \eta_f) \right) \bar{h} A_T} \quad (4.14)$$

where η_o is the overall fin system efficiency, η_f , the fin efficiency, A_f , the fin surface exposed to the fluid, A_T , the total surface of exchange and \bar{h} , the average convection coefficient in the heat collector. For simplicity, it was assumed that the extended surface heat collector geometry had rectangular fins of uniform cross-sectional area and an adiabatic tip as presented in Figure 4.13. Each of the N fins is equally distant from each other. The fin width H covers the entire duct, thus dividing the flow into $(N + 1)$ sub-channels.

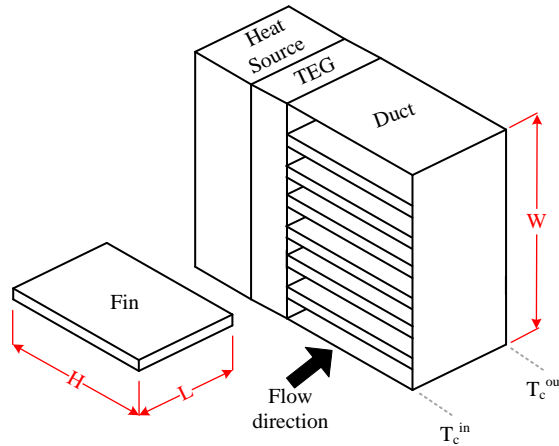


Figure 4.13: Heat collector duct and fin geometry.

The friction coefficient (f) was assumed constant in the present work, based on the assumptions of a turbulent flow and rough surface. In the test case, a value of $f = 0.05$ was used. For known friction parameters, an approximation of the convection coefficient \bar{h} can be obtained using the modified Reynolds analogy. This analogy expresses the relation between friction and heat transfer. For turbulent flow, this analogy states that:

$$\frac{f}{8} = St Pr^{2/3} = \frac{\bar{h}}{\rho_v V C_p} Pr^{2/3} = \frac{\bar{h} A_D}{C} Pr^{2/3} \quad (4.15)$$

where St is the Stanton number, Pr is the Prandtl number, C is the capacity rate and A_D is the total cross-sectional area of the duct [37]. The cross-sectional area A_D can be approximated by the product $H \times W$ when the fins are thin. Thus, after replacing \bar{h} in Eq.(4.14), one finds:

$$U = \left[1 - \frac{2NHL_c}{2NHL_c + WL_c} (1 - \eta_f) \right] \left(\frac{fC}{8HW Pr^{2/3}} \right) \quad (4.16)$$

where L_c is the length for a single cell of the duct. For prescribed N and U values, H can be calculated from Eqs. (4.15) and (4.16):

$$H = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad (4.17)$$

where $a = \frac{16NUL_c W Pr^{2/3}}{fC}$, $b = \frac{8UL_c W^2 Pr^{2/3}}{fC} - 2NL_c \eta_f$ and $c = -L_c W$.

The pressure drop for each duct is given by the combination of wall friction and minor losses:

$$\Delta P = f \frac{L}{D_h} \frac{\rho_v V^2}{2} + h_L \quad (4.18)$$

where D_h is the hydraulic diameter of a single sub-channel, L is the duct length, ρ_v is the fluid density, V is the fluid velocity and h_L accounts for the minor losses. One should note that ΔP is the same for each sub-channel because they are in parallel in the duct. Minor losses are considered with a uniform factor κ_L of 0.5 in cases where the flow changes direction between consecutive cells. The minor losses for each cell are expressed as:

$$h_L = m K_L \frac{\rho_v V^2}{2} \quad (4.19)$$

where m is the number of flow direction changes from by each sub-channel within each cell. For simplicity, each sub-channel of the same duct has identical length and the fluid density ρ_v is evaluated for each duct at the mean temperature of the total length. Therefore, the fluid velocity of each sub-channel of a duct is the same.

In order to calculate ΔP , it is necessary to find the fluid velocity in each parallel duct. The flow velocity of each duct is calculated from the mass conservation which is express as:

$$\dot{m}_{tot} = \rho_v A_D V \quad (4.20)$$

where $\dot{m}_{tot} = C / c_p$. Then, the total pumping power of the heat collector is calculated using the pressure drop from Eq. (19) for each duct composing the heat collector. The pumping power of each duct is given as:

$$W_{duct} = \left(\dot{m}_{tot} \frac{\Delta P}{\rho_v} \right)_i \quad (4.21)$$

The total pumping power W_p is obtained by summing over each individual duct of the heat collector. In the end, the net efficiency is obtained by subtracting from the power produced the one required to move the fluid:

$$\eta_{net} = \frac{P - W_p}{q_H} \quad (4.22)$$

4.5.3. Model constraint

The modified Reynolds analogy is used for the estimation of the convection coefficient assuming a fully developed turbulent flow. There is a minimal value of the Reynolds number for turbulent flow where the friction factor is only dependant on the relative roughness of the duct [42]. Here, the friction coefficient is assumed fixed for the entire heat collector ($f = 0.05$). Thus, the Reynolds number must be higher than 40,000 [42]. During the optimization, this constraint is verified and a penalty is added to the each objective function when it is not respected in such a way that these solutions are rejected.

4.6. Optimal TEG solutions with heat collector

The addition of the heat collector to the model increases the initial cost of the whole system. As a result, one could be interested in minimizing the cost of the TEG as well as the cost of the heat collector. Thus, three objectives were considered in the optimisation process for this model. The first objective is the net efficiency η_{net} obtained from the net power output divided by the inlet hot side heat transfer rate. The net power P_{net} is the difference between the thermoelectric power P and the pumping power w_p in the cooling system detailed in the previous subsections. The second and third objectives consist in minimizing the number of modules n and the surface of the cooling system A_c . Both of these objectives are related to the cost of the system, since more modules and a larger heat transfer surface area will increase the purchase cost of the system. In addition to the previously introduced design variables, the overall heat transfer coefficient U , heat capacity rate C , and flow pattern of the cooling system are additional variables. Furthermore, the number of fins N present in each cell of the heat collector grid is also a design variable. Therefore, the total number of design variables is now equal to $M+9$, where M is the number of cells in the mesh.

First, a preliminary optimisation was performed with only one objective, namely the net power output (i.e. $P_{net} = P - w_p$) for the sake of comparison with the previous results without a cooling system. Then, the global optimisation regrouping all three objectives was performed. The same multi-objective genetic algorithm as above (NSGA-II) was used [6]. The main parameters of the GA were those from Table 4.3 except that the population has been extended to 70 times the number of variables. The number of generations is also increased to 2000. Each optimization run was performed 5 times to ensure repeatability.

4.6.1. Single-objective GA optimisation results

The optimization procedure described above was applied to a test case problem which consists in a square surface of 25 m² with the temperature and heat flux distributions shown in Figure 4.7b. First, the single-objective optimisation for the net power output was performed with a 12 × 12 mesh and the results are reported in Figure 4.14. The thermoelectric global efficiency is 9% which is almost 96% higher than the design without the heat collector. The addition of the cooling loop to the system lowers the cold side junction temperature (T_{JC}) of the TEG providing a higher temperature difference through the module. Hence, the thermoelectric power generation is improved for the same inlet heat flux. However, because of the pumping power requirement, the net power output is reduced by 3% as shown in Figure 4.14a.

The overall lower thermoelectric leg temperature eliminates the recourse to the material combination #3 in the optimal design as shown in Figure 4.14c. From Figure 4.14b, one observes that the number of thermoelectric modules is reduced by 3 times compared to the design of Figure 4.9 for a total of 14,672. Therefore, the addition of the heat collector to the model increased the power output and lowers the number of modules.

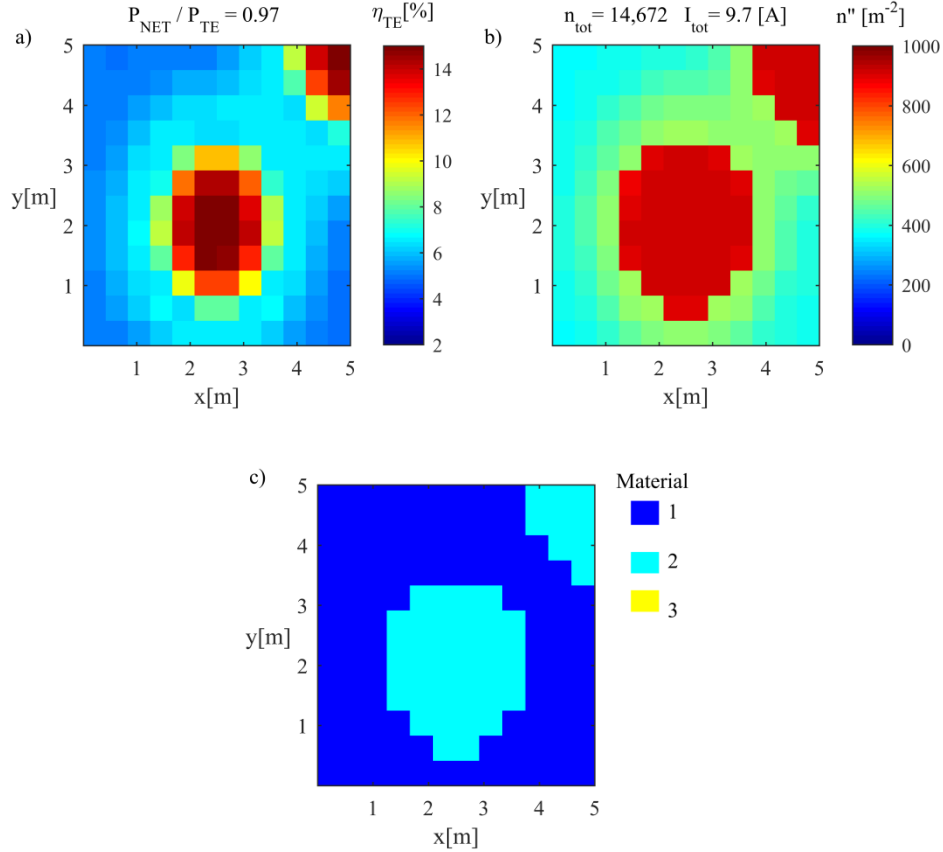


Figure 4.14: Single-objective GA optimization results of 2D TEG model including the cooling system with a 12×12 mesh with heat transfer distribution #2: a) TEG efficiency, b) net efficiency, c) material distribution.

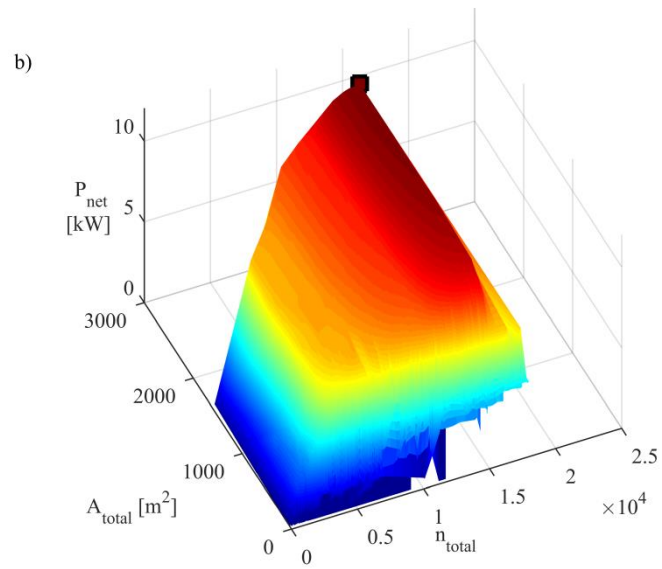
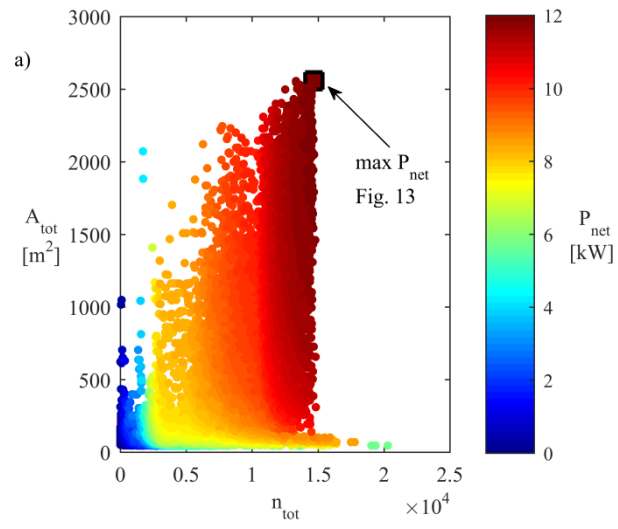
4.6.2. Multi-objective GA optimisation results

The three-objective genetic algorithm optimization is done five times with the same test case as in Section 4.6.1 for the 12×12 mesh. A non-dominated sorting is applied to all solutions to keep the non-dominated solutions (Pareto front). The result is reported in Figure 4.15. The square in Figure 15a is the maximal net power output solution from Figure 4.14 (single objective optimization). A 3D view is also presented in Figure 4.15 for better visibility of the 3D Pareto front.

From Figure 4.15a, it can be observed that the maximal net power design from Figure 4.14 requires more thermoelectric modules as well as a larger surface of exchange than the other solutions on the Pareto front. Nonetheless, for a similar power output, other solutions have been obtained with a lower surface of exchange. Therefore, the multi-objective optimization highlights other possible designs that could be more economically profitable than the design from Figure 4.14.

Figures 4.15c and 4.15d present the thermoelectric and heat collector grid layout from Figure 4.6 selected by the algorithm for each solution on the Pareto front respectively (illustrated by the different colors). In Figure 4.15d, the pattern retained by the algorithm for the flow layout is mostly the vertical pattern for every grid. Solutions with horizontal patterns end up in the bottom left corner of the Pareto front with the lowest net power production, surface of exchange and number of modules. Therefore, these designs were not reported in Figure 4.15. In Figure 4.15b, it can be observed that different groups of solutions with the same TEG grid are clustered in the same areas of the Pareto front. From the 24 possible TEG grid layouts, only few are present among the optimal solutions. TEG grid #12 (from Figure 4.6) is the best design choice in the highest power output cluster ($P_{net} = 8$ to 12 kW). Similarly, TEG grids #15 and #16 are selected for designs with lower net power output ($P_{net} = 4$ to 8 kW). In Figure 4.15d, solutions with similar HE grid layouts are also clustered in similar areas of the Pareto front. However, the flow grid #7 is selected for the higher power output cluster while flow grid #9 and #18 are selected for lower net power output. The high power output cluster in Figure 4.15c and Figure 4.15d covers an important range of surface of exchange and number of modules. The main variables that create this wide set of solutions on the Pareto front are the heat capacity rate C and the global heat transfer coefficient U . This result suggests that the heat collector parameters play an important role in the TEG performance.

In the high power output cluster, one can observe from Figure 4.15 that the heat collector surface of exchange can be reduced with minimal impact on the net power output. For instance, in comparison with the maximal power output solution of Figure 4.13, similar solutions on the Pareto front with roughly the same net power output exist with 40-50% less surface of exchange. As for the total number of modules, solutions with lower number of modules reduced the net power output significantly.



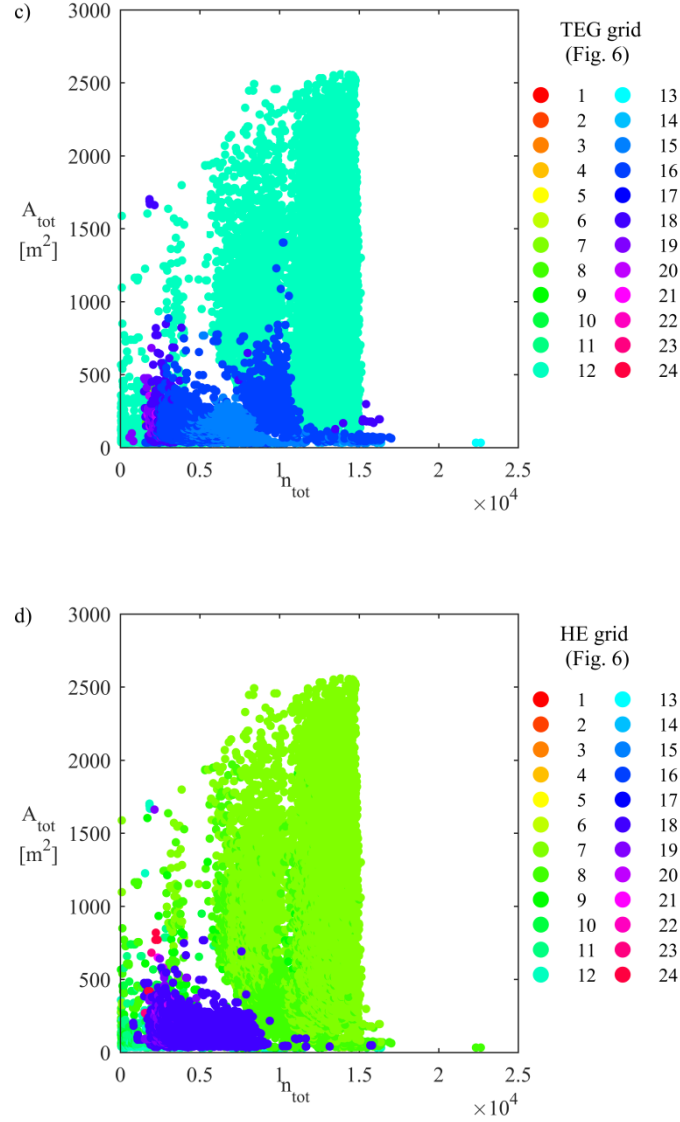


Figure 4.15: Pareto front of the multi-objective GA optimization results in plane view of objectives A_{tot} and n_{tot} with a) net power output color scaling, b) 3D view with net power output z-axis, c) optimal TEG grid layout scaling and d) optimal HE grid layout scaling.

4.7. Conclusion

In this study, we elaborate a method to design a TEG system to be applied to a hot surface without affecting thermally the system to which the hot surface belongs. Genetic algorithms are coupled to a numerical TEG model in order to maximize performance while keeping constant the source heat flux and temperature distributions. Single objective and multi-objective optimization approaches were used, revealing the best TEG

configurations in different test cases. The model proposed here can be adapted to any heat source with a specified heat flux and temperature distributions.

Different conclusions can be drawn. First, it has been observed that only specific combinations of total current and number of modules allow maintaining simultaneously the heat flux and temperature on the hot surface. Depending on the source heat flux and temperature distributions, the total current and number of modules combination might not yield the maximal power output possibly obtainable from the TEG.

The approach was then used within a 2D model with temperature and heat flux distributions rather than constant values. Multiple materials with temperature dependant properties were considered as well as different electric configurations for the TEG. The result shows that multiple designs are possible for optimized performances. Yet, providing finer granularity to the TEG solution results in slightly better overall thermoelectric designs. Furthermore, trade-off between optimal power generation and equipment cost is observed by adapting the thermoelectric connection grid, total current, aspect ratio and material selection.

Next, a heat collector was added to the cold side of the TEG. A multi-objective GA was used to optimize the TEG performance. It was shown that higher thermoelectric powers generally require more thermoelectric modules and a larger heat collector. Hence, trade-off between performances and cost could be investigated in future work. The importance of the TEG connection and heat collector flow layout is observed from the optimization. A larger set of possible TE and HE grids could be added in future work. Further investigation on the improvement of electric power output versus the addition of a cooling system should be performed in order to show its economic benefits.

Improvements to the model and design procedure could be studied in the future. For instance, with high temperature difference at the TEG junctions, it can become important to take into account the material properties variation within thermoelectric legs so as not to overestimate power generation. Other modifications can be made to the model by considering different TEG architectures, e.g., stratified thermoelectric modules. Adding more materials and HE fluids into the list from which the optimization can select from can also prove beneficial.

Chapitre 5 : Article #2

Titre :

Comparison between Five Stochastic Global Search Algorithms for Optimizing Thermoelectric Generator Designs

Co-auteurs :

Mathieu Allyson-Cyr, Louis Gosselin

Journal :

Engineering Optimization, submit in August 2018

Résumé

Les algorithmes stochastiques par recherche globale font partie des méthodes d'optimisation n'ayant pas recours aux dérivés qui permettent de résoudre une grande panoplie de problèmes rapidement. Ces algorithmes heuristiques permettent de trouver une solution approximative satisfaisante lorsque la solution exacte est difficile à obtenir via d'autres méthodes. Dans cet article, cinq algorithmes heuristiques sont comparés pour la résolution d'un problème non-linéaire, multi-objectif et à variables mixtes. Une méthode de résolution par poids pondérés est introduite pour la résolution du problème. Les algorithmes considérés dans l'article sont : Colonie de fourmis (ACO), algorithme génétique (GA), essaim particulaire (PSO), évolution différentielle (DE) et enseignement-apprentissage (TLBO). Une étape préliminaire est d'abord présentée pour la détermination des meilleurs paramètres d'algorithme. Ensuite, les algorithmes sont comparés sur un cas d'optimisation du design d'un appareil thermoélectrique basé sur un modèle disponible de la littérature. Les résultats montrent que les paramètres intrinsèques de chaque algorithme ont plusieurs effets non-négligeables sur la performance. Néanmoins, avec les paramètres optimaux, l'optimisation du problème démontre que la performance de chaque algorithme est comparable. Globalement, l'évolution différentielle offre les meilleures performances en générant un front de Pareto plus diversifié avec un nombre de solution non-dominés plus grand.

Abstract

Stochastic global search algorithms are a class of derivative-free optimization method that can solve a large variety of problems quickly. These heuristics find satisfying approximate solutions when exact solutions are difficult or impossible to obtain otherwise. In this work, the best settings of five continuous single-objective heuristics are determined for solving a mixed-integer non-linear multi-objective optimization problem. The algorithms treated in the paper are ant colony optimization, genetic algorithm, particle swarm optimization, differential evolution and teaching-learning basic algorithm. The optimization problem consists in optimizing the design of a thermoelectric device, based on a model available in literature. Results showed that the inner settings can have different effects on the algorithm performance criteria depending on the algorithm itself. A formulation based on the weighted sum method is introduced for solving the multi-objective optimization problem with optimal settings. Results showed that all the heuristic algorithms have comparable performances. Differential evolution was able to generate more diversified solutions across the near-optimal Pareto front with the highest number of non-dominated solutions in comparison with the other algorithms.

5.1. Introduction

Modern engineering design problems are complex problems that can become quite large. They typically involve discrete and continuous variables and non-differentiable and non-linear functions. As a result, these optimization problems are often difficult to solve, in particular with “traditional” optimization techniques. Stochastic global search algorithms are a class of derivative-free optimization methods that are an appealing option for solving these problems [43][44]. This family of algorithms uses stochastic and direct-search methods to find good approximate solutions to complex problems with little to no prior knowledge of the optimization problem. These algorithms are often referred to as heuristics and can be classed into evolutionary algorithms (EAs) such as genetic algorithm (GA) and differential evolution (DE) or swarm intelligence (SI) like particle swarm optimization (PSO). Due to their inherent stochastic features, these algorithms can lead to different solutions to the same problem each time they are used, and thus the optimality is not guaranteed. Nonetheless, these algorithms can converge faster than other methods and can find nearly optimal solutions efficiently when used correctly [45].

The convergence mechanism of heuristic algorithms can be adapted using two main forces in search for global minima: diversification and intensification. Diversification is a term used to describe the exploration of the overall design space, whereas the intensification characterizes the local search of a specific area of the design space. Both forces are contradictory and complementary, and must be balanced to reach nearly optimal solutions efficiently [46]. For the vast majority of existing heuristics, algorithm settings must be tuned prior to the optimization [47][48][49]. These parameters directly influence the convergence mechanisms and are often problem dependant. Setting the algorithm parameters for complex nonlinear mixed-variable problems becomes a crucial part of the optimization process for most of the heuristic algorithms. Various methods exist to determine optimal settings. Most of these methods are considered as an offline tuning which consists in determining the best algorithm parameters before actually using the algorithm for solving a specific optimization problem [50]. Alternatives also transform the offline tuning problems into a continuous optimization problem to be used with continuous optimization techniques [51].

With appropriate settings, heuristics can solve various types of problems. The algorithms presented in this paper are developed for solving continuous single-objective problems. Nonetheless, these algorithms can be used for solving multi-objective optimization problems. According to the literature, one of the most common methods is the weighted summation approach which has been intensively studied [52]. This approach is a general scalarization method combining all the objectives function into a single function where the objectives are summed with weighted parameters. Under some limitations, this approach can be used with single-objective heuristics for solving mixed-integer non-linear multi-objective optimization problems. However, the

weighted sum method has some drawbacks that can lead to inefficient Pareto front representation [53]. Few performance comparisons have been found in the literature between different stochastic global search algorithms with this method.

When confronted to a complex optimization problem, the selection of a "fitting" heuristic algorithm in terms of "optimal" quality in solutions and "acceptable" computation time is not straightforward. In addition, the settings suggested for various heuristics in theory or abstract models are not representative of real-life models [54]. Further study must be conducted for selecting an appropriate algorithm for complex real-life optimization problem with multiple local minima or high number of variables with the use of particular optimization method (like weighted summation approach).

In this paper, five different population-based stochastic heuristic algorithms are compared for optimizing the design of a thermoelectric device, which is a mixed-integer nonlinear constrained problem (Ref. [55]). First, the impact of the algorithm internal parameter settings on the convergence speed, repeatability and capacity to reach optimal solutions is investigated. Then, a multi-objective optimization problem with a weighted sum method is solved using the best setting identified in the previous section. Modified convergence and diversity metrics are used to compare the performance of the population-based heuristics.

5.2. Heuristic algorithms

This section presents the five selected population-based heuristic algorithms to be compared. The problem solved in this study is described below in Section 3 and consists in optimizing a thermoelectric generator design. This reveals to be a multi-objective nonlinear optimization problem with both continuous and discrete variables. The main features of each algorithm are summarized in the sub-sections below. References are provided for readers interested to learn more about the details of each heuristic. The algorithms have been implemented in *Matlab*.

It should be noted that the algorithm presented here are created for continuous optimization problems. Since the test case is a mixed-variable optimization problem, the discrete variables must undergo a specific treatment. Since all the discrete variables of the thermoelectric problem are categorical variables, there is no intrinsic ordering. Therefore, the integer variables are considered as continuous variables and then rounded to their nearest valid indices before evaluating the objective function.

5.2.1. Genetic Algorithm (GA)

Genetic algorithms are one of the most popular EAs that use the concept of natural selection to create offspring. A first version of the algorithm has been introduced in Ref. [56]. The bio-inspired concepts of selection, crossover and mutation are used to produce better offspring for the next generation (i.e. better solutions).

According to the literature, GAs have been highly studied and applied to many fields such as heat transfer problems [57], HVAC systems [58], flow-shop scheduling [48] and thermoelectricity [16]. The algorithm has also been adapted to tackle mixed-discrete optimization problems [59][60] and multi-objective optimization [7].

The selection is the first step and it consists in selecting promising parents for the generation of offspring. Several methods can be used to select the individuals in the population. In this paper, three selection methods are retained: uniform section, roulette selection and tournament selection. The first selects a parent with a normal distribution based on the value of its solution and the number of parents. The roulette selection method simulates a roulette wheel in which every individual is present with the area proportional to its solution value. The tournament method chooses a group of 4 individuals randomly in the population and then selects the best ones to become a parent.

A crossover is then performed between two parents to generate new offspring. The crossover ratio (CR) is used to determine the number of individuals in the population that will become the parents (and the number of offspring). The method used to make a crossover child can also be adapted. In this paper, three crossover methods are retained: scattered, single point and double points. Each of these methods provides different ways that determine which parent will provide the child with which of its variable values. The scattered method creates a random binary vector equal to the number of variables and then gives one of the parent variable to the child depending on the value of the binary vector. The single point method selects randomly a value between 1 and the maximal number of variables. The child receives all the variable values of the first parent before this point and the rest comes from the second parent. The double point method is similar but two points are randomly selected. The first parent gives the variable values between the two points to the child and the rest is passed from the second parent

The last step is the mutation. The mutation replaces randomly some variable values in some individuals of the population. With the GA used here [6], the number of individuals experiencing mutations is fixed to the remaining fraction of individuals that has not been subject to the crossover ($1 - CR$). An adaptable feasible method is used here for the mutation in order to satisfy variable bounds. The pseudocode of the algorithm can be summarized:

| | |
|----------------|--|
| <u>Step 1.</u> | State variable bounds, algorithm parameters and termination criterion. |
| <u>Step 2.</u> | Initialize the population, the crossover fraction CR and the selection and crossover methods. |
| <u>Step 3.</u> | Select individuals to become parents. |
| <u>Step 4.</u> | Perform the crossover on the parents to generate Pop*CR children. |
| <u>Step 5.</u> | Perform the mutation on Pop*(1–CR) individuals. |
| <u>Step 6.</u> | Re-form a population combining mutants, children and initial individuals and then eliminate the worst individuals. |
| <u>Step 7.</u> | Repeat steps 3 to 6 until the stopping criterion is met. |

5.2.2. Particle Swarm Optimization (PSO)

PSO is a bio-inspired heuristic mimicking the behavior of bird flocks or fish schooling [10]. The algorithm has been developed for solving continuous nonlinear problems. Nonetheless, the algorithm can be adapted for solving mixed problems by converting discrete variables into continuous values as explained in Ref. [61]. PSOs have been successfully used in many real-world applications like heat exchangers [62][63], thermoelectricity [64][65], inverse heat transfer problems [66], mechanical designs [67] and geothermal power plants [68]. Improvements the algorithm have been developed for multi-objective optimization [69], easier parameter tuning [70], better search efficiency [71] or for handling local minima more efficiently [72].

The algorithm explores the design space, searching for the optimal solution by changing the trajectories of each “particle” of the swarm. Each particle is a member of the population and a solution vector of the optimization problem. Each individual particle moves in the multidimensional space toward the optimal solution by modifying its position using the information of its best personal score (P_{Best}) and the best score of the entire swarm (G_{Best}).

The velocity of each particle is modified using three elements that can be adapted with different parameters. The first element is the velocity of the previous iteration. The parameter $w_{inertia}$ is the inertia weight affected to the velocity to control the exploration factor of each particle and minimize the risk of the algorithm to get trapped into a local minimum. The second and third elements use the information of P_{Best} and G_{Best} to change the direction of the particle. The personal learning coefficient c_1 and the global learning coefficient c_2 are used to control the intensification of the search. In addition, w can be damped at each iteration with the parameter w_{damp} to gradually reduce the inertia weight $w_{inertia}$ and increase the intensification of the

search when the swarm is close to the global minima. The last parameter is the velocity limitation. Without control over the minimal and maximal value of the velocity, the swarm could explode with very high velocity, moving the particles outside the limits of the design space. The pseudocode of the algorithm can be summarized as:

- Step 1. State variables bounds, velocity limits, algorithm parameters and termination criterion.
- Step 2. Initialize the population and the algorithm parameters.
- Step 3. Evaluate the objective function of each particle. P_{Best} and G_{Best} can be changed if they are better than their actual values.
- Step 4. Modify each particle position. $w_{inertia}$ can be damped at this step if a damping parameter is defined.
- Step 5. Repeat steps 3 and 4 until the stop criterion is met.

5.2.3. Ant Colony Optimization for continuous variable (ACO_R)

ACO is an algorithm inspired from the behavior of ants foraging. The algorithm was initially developed for solving discrete problems [8][73]. One particularity of the algorithm is how it recreates the movement of ants from the nest to the foraging area and vice-versa following the shortest path. The ants are solutions that follow one of the paths that are potential solutions in the solution space. When an ant follows a specific path, it has the solution component from that particular path. When an ant moves between the food source and the nest, it releases a pheromone trail that other ants can smell to change their path. The pheromone trail is used to probabilistically sample the search space (paths). The shorter the path, the stronger the concentration of the pheromone trail, increasing the probability that ants move to that specific path. This bio-inspired mechanism allows ants to interact with other nest mates to move through shorter path in the design space (minimal cost). An evaporation parameter is also included to the pheromone trail in order to control the premature convergence to a path that might not be the global minimum. At each iteration, the concentration of the pheromone trail is lowered to reduce the attraction of the other ants and increase the exploration of the design space by other ants.

ACOs have been widely used in many fields of application. To name a few, ACOs has been successfully applied to heat transfer problems [74][75], thermoelectricity applications [76] and water distribution systems [49]. ACO has also been combined with other heuristic algorithms such as PSO for solving inverse heat transfer problems [77] for higher effectiveness and overall robustness. Variations of the algorithm exist to solve other kinds of problems. ACO_R is a variant for continuous domain [9] and ACO_{MV} is a variant for mixed-variable

problems [78]. For the mixed-variable problem of the present test case, ACO_R was used with a treatment of the discrete variables similar to that in ACO_{MV} where discrete variables are treated as categorical variables, similarly to Ref. [79].

The main idea behind ACO is the pheromone trail used for incremental construction of solutions. A finite set of possible paths exist and a probability is given from best to worst to each of them depending on the pheromone value. However, with continuous variables, the concept of pheromone is different. Instead of a discrete probability distribution, a weight is given to each solution in the population rated from best to worst with a probability density function (PDF). The weight determines the attractiveness of a solution during the construction process and a probability for an ant to choose this solution is set based on the weight value. The algorithm parameter q is the intensification factor used in evaluating the weight. A small value of q increases the weight for the best-ranked solutions while a high value of q makes the weight more uniform among the ranked-solutions. Then, the probability to select the solution is calculated for each solution based on the weight and their rank.

In the first step, k ants (potential solutions) are randomly generated. In a second step, the algorithm starts the construction of new solutions variable by variable by generating Gaussian random variables. The number of new solutions created can be adapted. In this paper, the sample used to construct new solutions is fixed to half the population. For each variable of each solution, a standard deviation is calculated including the deviation-distance ratio ζ that works in a way similar to the pheromone evaporation rate in ACO. ζ affects the long term memory by reducing the search for already explored points. Therefore, the convergence rate is often lower with a higher value of ζ . Variable by variable, the algorithm selects a solution with a roulette wheel selection scaled with the probability. Then, a Gaussian random variable is generated from the standard deviation and the value of the variable from the selected solution. The pseudocode can be summarized by these steps:

- | | |
|----------------|--|
| <u>Step 1.</u> | State variable bounds, algorithm parameters and termination criterion. |
| <u>Step 2.</u> | Initialize the population. |
| <u>Step 3.</u> | Calculate the Gaussian functions w and the probability associate to each solution rank. |
| <u>Step 4.</u> | Construct solutions (new solutions). For each individual variable, select the Gaussian kernel function with the probability to select each Gaussian function w . Then, generate random Gaussian variable |
| <u>Step 5.</u> | Update the population by eliminating the n_{sample} worst solutions. |
| <u>Step 6.</u> | Repeat steps 3 to 5 until the stop criterion is met. |

5.2.4. Teaching-Learning Basic Optimization (TLBO)

TLBO is inspired by the apprenticeship of a teacher with learners by exchanging information to reach global minima [12][80][81]. Similarly to other heuristic algorithms, TLBO is a population-based algorithm where each individual of the population is a learner and the teacher is the best individual in the population. However, one particularity of this algorithm is that it has no internal parameters to fix, which makes it easy to adapt to any optimization problems. However, since the algorithm is developed to solve non-constrained nonlinear continuous problems, its capability to solve mixed-integer non-linear problems is not clear.

TLBOs have been used for solving many kind of optimization problems such as heat exchangers optimization [82], thermoelectric cooler [83] and heat pipes [84]. The algorithm can be used to solve multi-objective optimization problems [85].

The algorithm has two distinct phases. The first phase is the teaching phase where the best learner (i.e. the selected teacher) tries to provide knowledge to the other learners to improve the overall mean score of the class. In the second phase, the learners interact with each other to improve their best score. The best learner among the two teaches his knowledge to the other one. This is done by adding to the worst learner a fraction of the difference in the solution vector between the two learners. This fraction is randomly generated for each interaction between two learners [12]. The intensity of this exchange could be increased but here, the ratio of exchange is chosen randomly for each variable of the solution. It should be noted that for both phases, the objective function is evaluated for the entire population. Therefore, the algorithm required twice the number of objective function evaluation as the algorithm presented in this paper. Finally, the best solution (learner) is updated to become the teacher for the next iteration. The pseudocode can be summarized as:

- | | |
|----------------|---|
| <u>Step 1.</u> | State variable bounds, algorithm parameters and termination criterion. |
| <u>Step 2.</u> | Initialize the population. |
| <u>Step 3.</u> | Calculate the population mean and identifying the best solution to become the teacher. |
| <u>Step 4.</u> | Teaching phase: modifying solutions with the best solution (i.e. the teacher). Update the best solution if one of the learners is better. |
| <u>Step 5.</u> | Learning phase: a learner is compared to another one randomly where the better solution is used to modify the other learner. Update the best solution if one of the learners is better. |
| <u>Step 6.</u> | Repeat steps 3 to 5 until the stop criterion is met. |

5.2.5. Differential Evolution (DE)

DE is an algorithm with an approach similar to that of GAs (i.e. mutation, crossover, selection). The algorithm has been introduced by Ref. [11] to solve nonlinear continuous problems.

According to the literature review, DEs have been less studied than other heuristics. Nonetheless, authors have used DE in some applications such as optimal shell-and-tube heat exchangers [86] and other various engineering problems [87]. Variations of DEs also exist for solving multi-objective optimization problems [88].

At each generation, the algorithm selects randomly a target vector and two other solution vectors from the population pool and performs a mutation using a scaling factor β . This factor affects how different the mutant vector will be from the target vector. Afterwards, a crossover is done between the target vector and the mutant vector to obtain the trial vector with a crossover probability p_{CR} . The crossover is performed on each variable of the vector. Higher p_{CR} increases the probability that the trial vector will have variable values from the mutant vector. The best solution between the trial vector and the target vector is the solution retained. The pseudocode can be summarized by:

- | | |
|----------------|---|
| <u>Step 1.</u> | State variable bounds, algorithm parameters and termination criterion. |
| <u>Step 2.</u> | Initialize the population. |
| <u>Step 3.</u> | Perform the mutation with three randomly selected vectors. |
| <u>Step 4.</u> | Perform the crossover between the mutant vector and the trial vector. |
| <u>Step 5.</u> | Select the best solution between the trial vector and the target vector. Retain the best one in the population for the next generation. |
| <u>Step 6.</u> | Repeat Steps 3 to 5 until the stop criterion is met. |

5.3. Test case: Thermoelectric generator design optimization

As mentioned above, the five selected algorithms have been analysed on a series of continuous problems. However, few comparison exercises were found, and in particular, the capability of these algorithms to solve non-linear mixed variable problems is not well documented. The optimization problem described in the present section has been used for that purpose. It consists in optimizing the design of a thermoelectric generator system developed for heat recovery applications, see Ref. [55].

Thermoelectric generators (TEGs) are devices capable of converting directly heat into electricity by using the Seebeck effect. Featuring no mobile parts or working fluids, highly durable TEGs are attracting more

and more attention for heat recovery applications. However, the high cost and low efficiency of actual thermoelectric materials is the main limitation for commercial use. The thermoelectric generator model used in this paper is based on the one developed in Ref. [55]. The heat source has constrained temperature and heat flux distributions over its surface. In order to achieve a significant temperature difference through the thermoelectric modules, a cooling system is installed on the cold side of the TEG. Figure 5.1 shows a schematic view of the system.

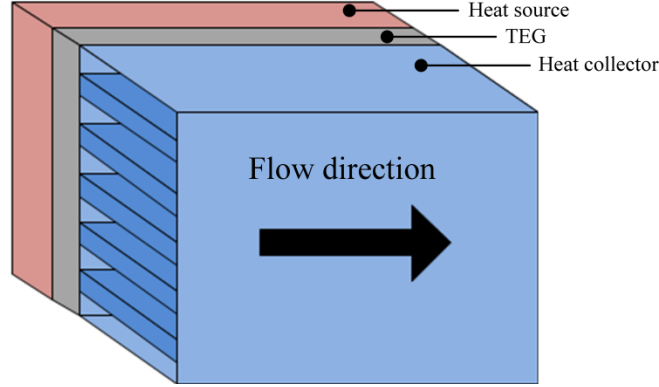


Figure 5.1: Schematic view of the TEG system sandwiched between the heat source and a heat collector.

5.3.1. Objective functions

The objective of this design optimization is to maximize the electric power output while minimizing the cost of system. The cost of the TEG system is related to the number of thermoelectric modules and the total surface of exchange of the cooling system. Thus, a total of three objectives should be simultaneously optimized. A weighted sum method is used here since most of the algorithms presented were primarily developed for single-objective problem. Therefore, the objective function is:

$$F = s_1 (-P_{net}) + s_2 (n_{total}) + s_3 (A_{total}) \quad (5.1)$$

where s_i are the weight ratio of each objective i , P_{net} is the net power obtained from the difference between the electric power output and the pumping power for the cooling system, n_{total} is the total number of modules of the TEG and A_{total} is the total surface of exchange of the heat exchanger (cooling system). In order to obtain the equivalent of a Pareto front corresponding to all three objectives, multiple combinations of weight ratios are tested with each algorithm. Every combination of weight ratio varying from [0:0.2:1] is tested for a total of 216 solutions for each run. Afterwards, a non-dominated sorting is performed on the solution set to eliminate the dominated solutions. It should be noted that the weighted summation approach is unable to generate solution of non-convex portion of a Pareto optimal front [53].

5.3.2. Model description

In this paper, the fixed hot-side heat flux and temperature distribution shown in Figure 5.2 were used to test the algorithms. The surface is divided into a 12×12 grid for a total of 144 cells of equal dimensions. The list of 165 variables (both discrete and continuous) of the model is presented in Table 5.1.

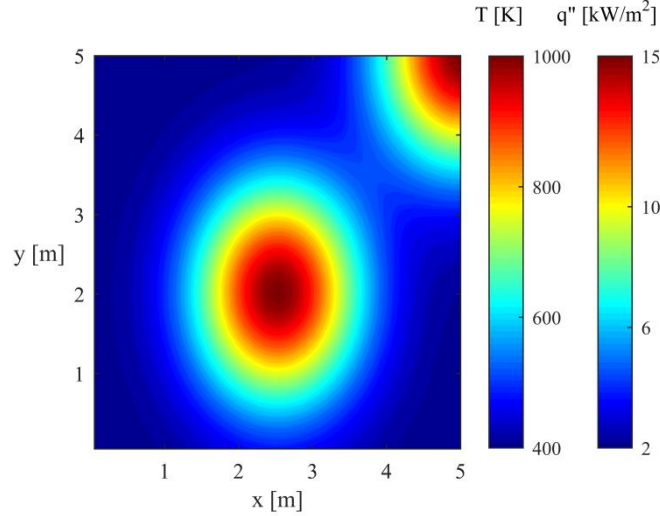


Figure 5.2: Constrained temperature and heat flux distributions over the heat source surface.

Table 5.1: Discrete and continuous decision variables of the TEG model (with 12×12 grid), based on Ref. [55]

| Variables | Number of variables | Bound |
|--|---------------------|-----------------|
| Thermoelectric material selection | 144 | {1...3} |
| Thermoelectric electric connection grid layout | 1 | {1...24} |
| Heat collector flow grid layout | 1 | {1...24} |
| Heat collector duct geometry | 1 | {1,2} |
| Number of fins (for each duct) | 12 | {0...100} |
| Total electric current [A] | 1 | [0.1...100] |
| Surface-to-length ratio of modules [m] | 3 | [0.0001...0.01] |
| Global heat transfer coefficient U [W/m²K] | 1 | [1...100] |
| Capacity rate of the collector C [W/K] | 1 | [100...25,000] |

5.4. Analysis of heuristic algorithm parameters effect on performance

The “best” algorithm settings can be different from a problem to another and can affect the comparison exercise proposed here. Different techniques have been proposed to fine-tune the parameters of evolutionary algorithms (e.g., [89] and [90]), but for the type of problems addressed here, only scarce information or rules of thumb were found in literature for selecting the optimal algorithm parameters. Therefore, it was decided to investigate the effect of these parameters on the performance of the algorithms. A specific case was used for this purpose, with weight ratio values $\{s_1, s_2, s_3\}$ of $\{1, 0, 0\}$. In other words, this corresponds to the problem of maximizing the net power output only. Note that a preliminary analysis (not shown here) has verified that a change in the weights did not affect significantly the main findings of this section (i.e., which algorithm parameters work better). The stop criterion is when the number of iterations reaches 100. The number of individuals in the population is set to 2000. The initial population is different for each run and is generated using a Latin hypercube sampling (LHS) rather than creating a uniformly distributed random population. It was demonstrated that the initial population generated from LHS is more uniform across the design space which leads to faster convergence speed and higher diversity of optimal solutions [5]. Table 5.2 presents the list of parameters tested and their values. One should note that every possible parameter combination of Table 5.1 has been tested. To verify the repeatability, a total of 5 runs are done with each algorithm for each parameter combination.

Table 5.2: List and range of algorithm parameters

| Algorithm | Parameters | Combination tested |
|-----------|------------------|--|
| GA | CR | [0 : 0.05 : 1] |
| | Crossover method | Scattered - single point - double points |
| | Selection method | uniform - roulette wheel - Tournament |
| PSO | $w_{inertia}$ | 1 |
| | w_{damp} | [0.7 ; 0.99 ; 1] |
| | c_1 | [0 : 0.2 : 2] |
| | c_2 | [0 : 0.2 : 2] |
| ACO | q | [0.001 : 0.001 : 0.02] |
| | ζ | [0.1 : 0.1 : 1] |
| DE | β | [0.01 : 0.01 : 0.1] |
| | pCR | [0.1 : 0.1 : 1] |
| TLBO | - | - |

5.4.1. Impact of algorithm parameters on convergence rate, repeatability and optimal solutions

The combination of algorithm parameters can either increase the exploration of the search space or increase the intensification of the search around local or global minima. For a given problem, a trade-off must be made between different algorithm parameters to reach a global minimum in less iterations. However, the effect of those algorithm parameters on the performance (i.e. convergence rate, repeatability and achievement of global minima) is not always clear. For example, some algorithms can reach optimal performance with a wide range of algorithm parameters combinations making them easier to use, such as PSO [68]. Therefore, one might be interested in determining the effect of those algorithm parameters on the algorithm performance. Therefore, this section features the algorithm parameter analysis of GA, PSO, ACO_R and DE. One should remember that TLBO does not have internal parameters to choose.

In this section, the algorithm parameters from Table 2 are tested and compared for each algorithm. For every possible combination of algorithm parameters, the performance of an algorithm was assessed by calculating:

- (i) The best solution obtained among the 5 optimization runs;

(ii) The coefficient of variation, also known as the relative standard deviation (RSD), which is the ratio of the standard deviation over the mean for the 5 optimization runs. A low RSD value indicates a better repeatability;

(iii) The number of iterations required to reach 98% of the maximal objective function (Ite). The average Ite among the five runs is calculated.

These metrics are reported in Figures 5.3 to 5.6. In Figure 5.3, one can see the impact of the crossover ratio CR and the selection method of GA on its performance. The crossover method has been fixed with the scattered function. In Figure 5.3a, a crossover ratio of 0 (i.e., mutation ratio of 1) provides poor solutions with either of the selection methods. In addition, high mutation reduces considerably the convergence speed and the reliability. It can be seen from Figures 5.3b and 5.3c that the number of iterations required and the RSD tend to be higher with CR below 0.6. With CR over 0.6, each selection method is able to reach satisfying solutions with less iterations and with a better reliability. Nonetheless, for a better reliability, the tournament selection method should be used since the RSD is the lowest as presented in Fig 5.3b (~0.015). Therefore, proper parameters for the test case could be CR between 0.6 to 1.0 while using the tournament selection method and the scattered crossover method.

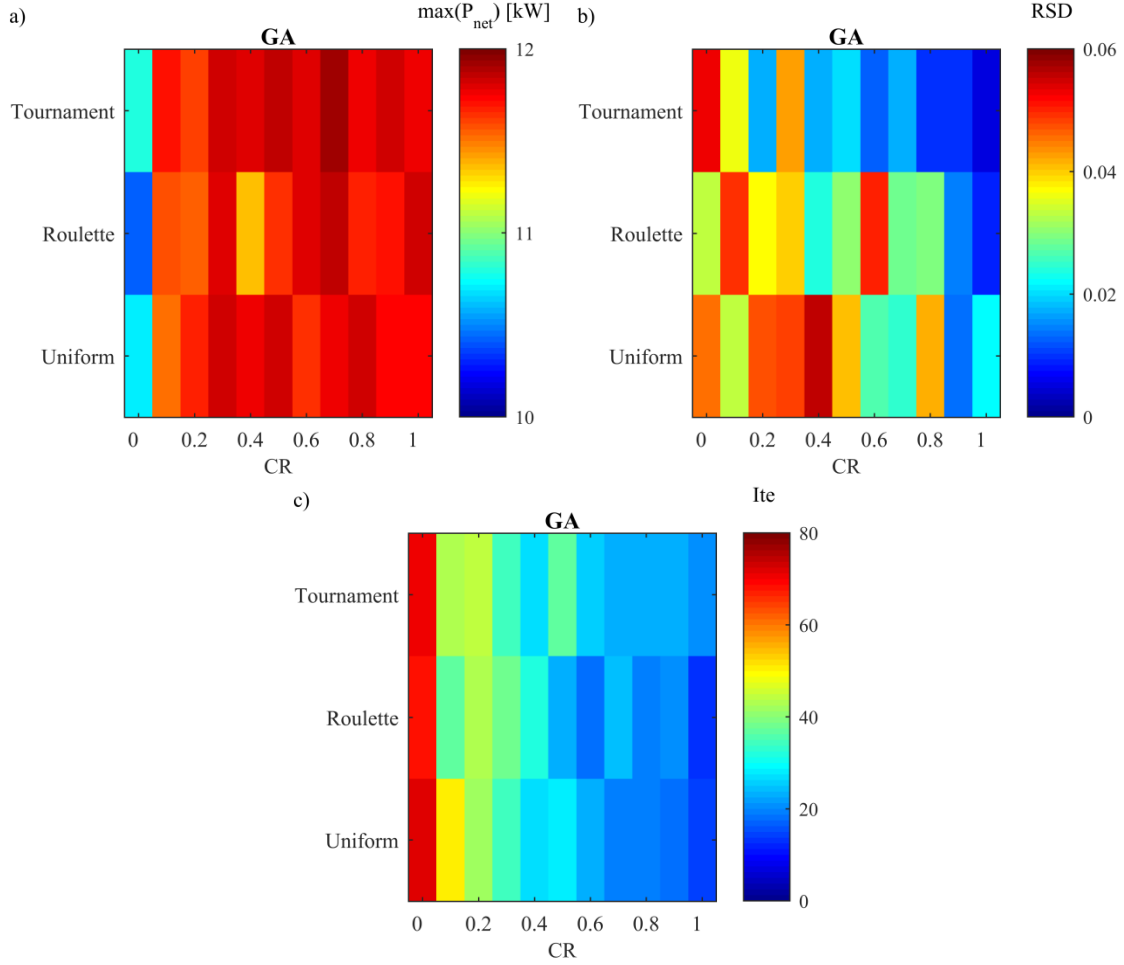


Figure 5.3: Impact of GA parameters on a) maximal net power, b) RSD and c) Ite .

In Figure 5.4, the effect of the personal and global learning coefficient (c_1 and c_2) of PSO on the performance is presented. The inertia weight $w_{inertia}$ and inertia damping w_{damp} have been fixed to 1 and 0.99 respectively based on Ref. [47]. One should note that, reducing w_{damp} resulted in premature convergence of the algorithm while removing the damping resulted in too much exploration leading to much lower convergence speed. In Figure 5.4a, the results show that the value given to c_1 and c_2 has no significant impact on the solution obtained. Increasing the intensification of the search (i.e. by increasing c_1 and c_2) seems to have no effect on the performance. However, having $c_2 = 0$ is not an appropriate choice because the algorithm does not converge as shown in Figure 5.4a. Furthermore, the impact of c_1 is negligible for this test case. From Figure 5.4b, PSO seems to have an overall good reliability since the RSD is relatively low (~ 0.01). As for the convergence speed, the results from Figure 5.4c show that higher values of c_1 and c_2 increase the number of iterations required to reach the final solution. With a higher weight given

to personal and global learning, some individuals in the population are more easily attracted toward local minima. Hence, the local minima obstruct the swarm in the search of the global minima, resulting in lower convergence speed. Therefore, with $w_{inertia}=1$ and $w_{damp}=0.99$, appropriate c_1 and c_2 values could be between 0.4 and 1.4 for this test case.

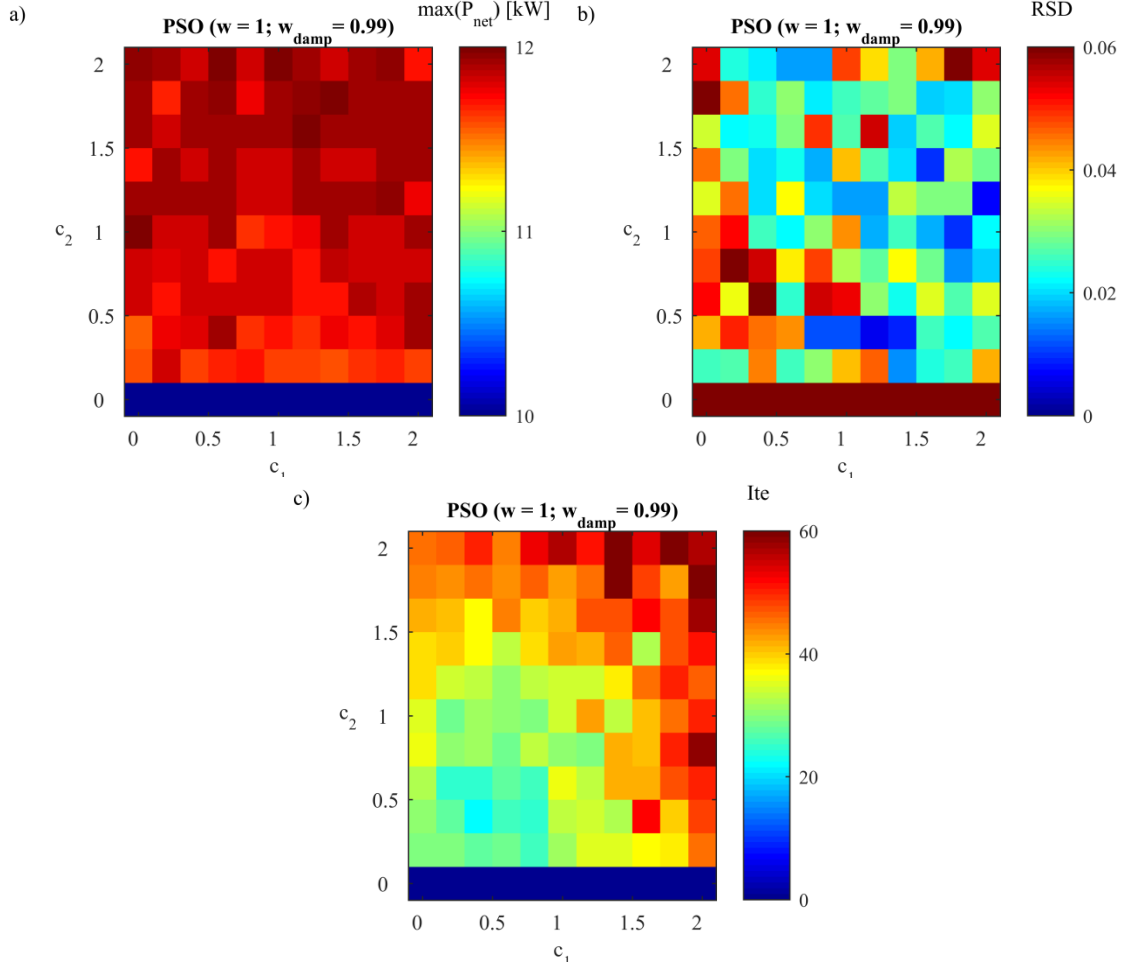


Figure 5.4: Impact of PSO parameters on a) maximal net power, b) RSD and c) Itc.

In Figure 5.5, the effect of the intensification factor q and the deviation-distance ratio ζ of ACO_R on the performance is presented. In Figures 5.5a and 5.5c, the results show that a high value of ζ reduces the capacity of the algorithm to reach good solutions as well as the convergence speed. Consequently, lower ζ values (between 0.3 and 0.6) appear to be the best choice. In Figure 5.5a, with ζ between 0.3 and 0.6, q can cover a wide range without affecting the capacity of the algorithm to reach good solutions. However, q has a direct impact on the convergence speed of the algorithm. In Figure 5.5c, it can be seen that lower values of q (i.e. increasing the weight for high-ranked solutions) reduce the total number of iterations required for the

algorithm to converge. Thus, good parameters combination could be $\zeta \sim 0.4$ and q below 0.01 for fast convergence and good capacity to reach good satisfying solution. In Figure 5.5b, RSD is globally higher with every parameter combinations and no tendency can be observed, which suggests that reliability provided by ACO_R is generally low for this test case.

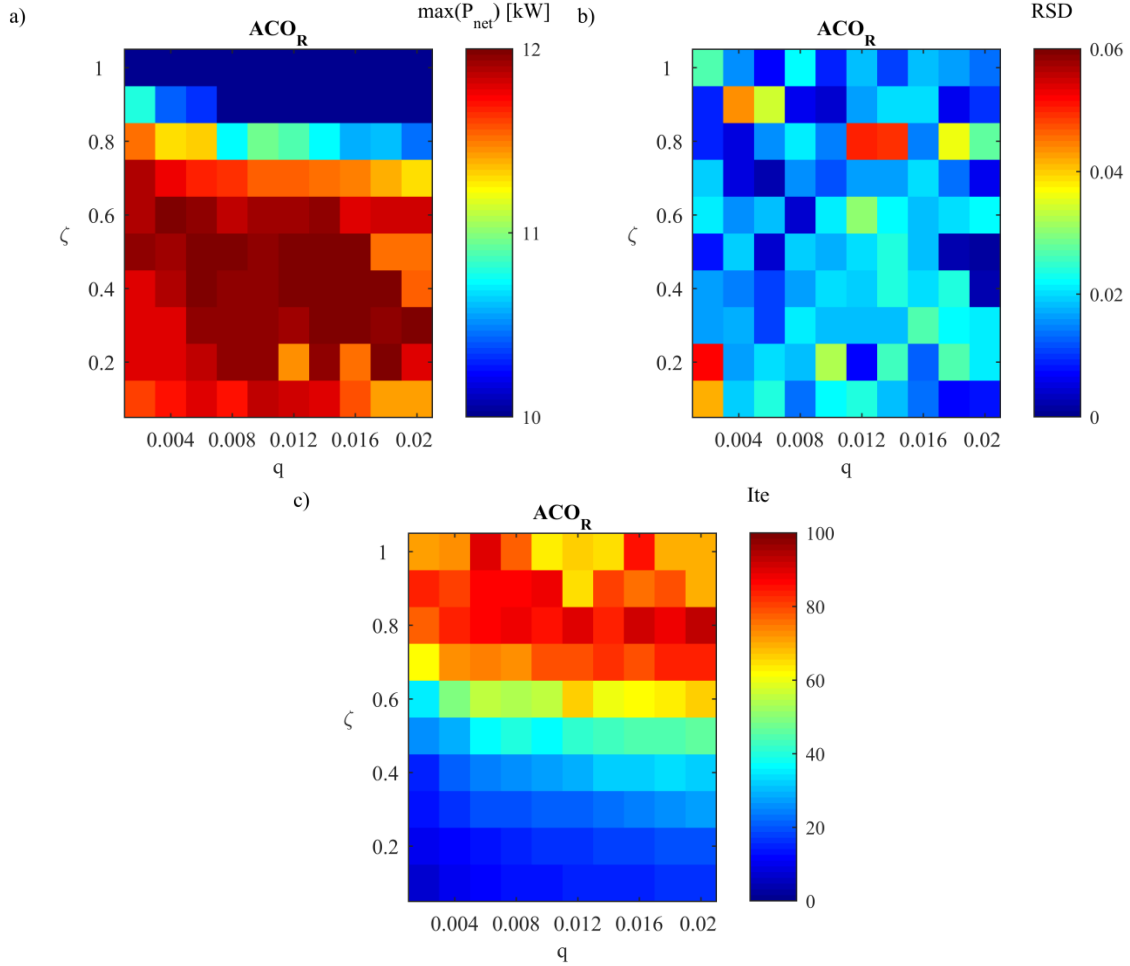


Figure 5.5: Impact of ACO_R parameters on a) maximal net power, b) RSD and c) Ite .

In Figure 5.6, the effect of the crossover probability p_{CR} and scaling factor β of DE on the performance are presented. Figure 5.6a shows that higher p_{CR} leads to better solutions. In addition, having lower β also improves the solutions. Nonetheless, the effect of β is less present with high p_{CR} . One should note that a p_{CR} value of 1 makes all trial vectors mutant which prevents the algorithm to converge correctly. In Figure 5.6b, the RSD is relatively low and constant which reveals that the algorithm has a good reliability overall. The convergence speed in Figure 5.6c shows that high p_{CR} and low β reduces the number of

iterations required to converge. In the end, proper parameters for DE for this test case are approximatively $pCR \sim 0.8$ and $\beta \sim 0.01$.

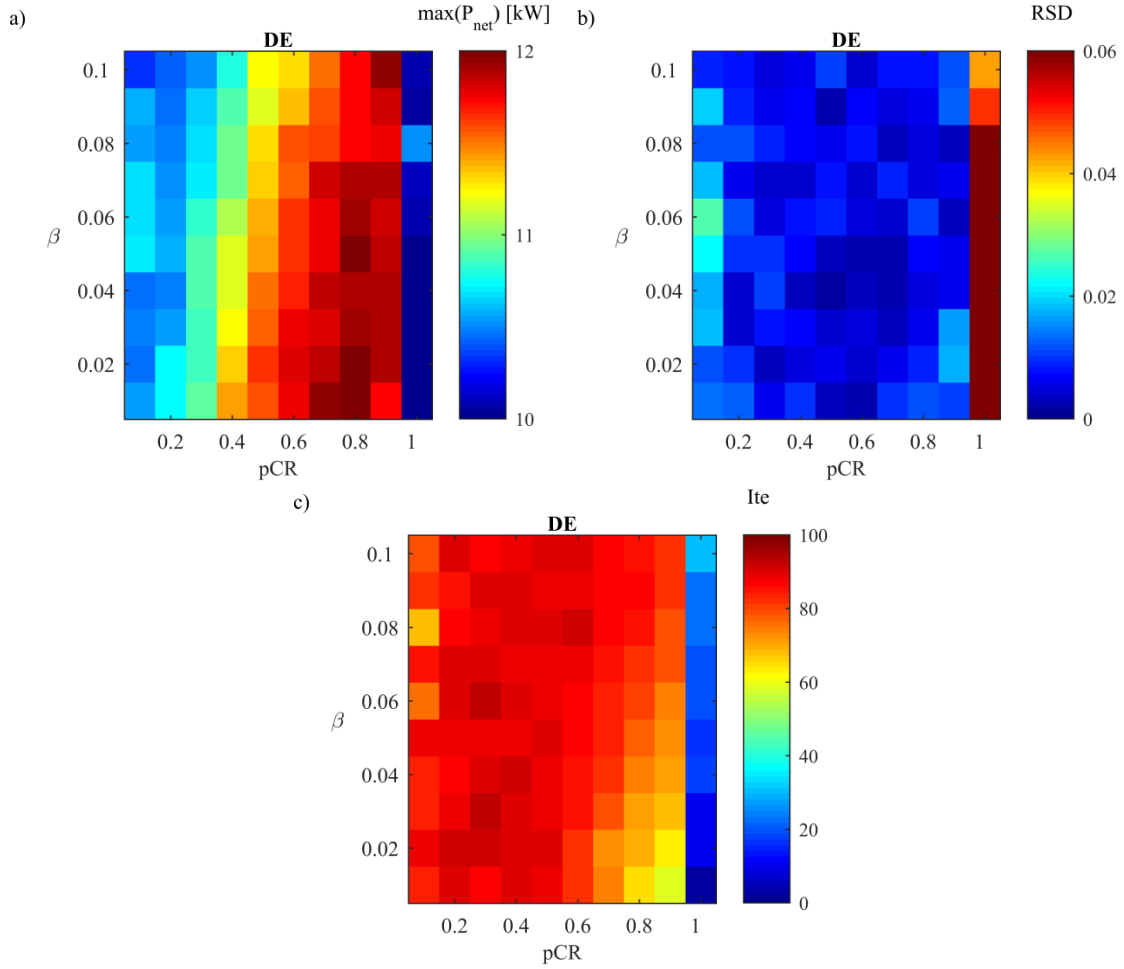


Figure 5.6: Impact of DE parameters on a) maximal net power, b) RSD and c) Ite .

To better compare the different algorithms, the resulting net power output versus the RSD is reported in Figure 5.7 for every combination of algorithm parameters tested for GA, PSO, ACO_R and DE. The power output versus the required number of iterations (average over the 5 runs with the same settings) is also shown in this figure. Each point in Figure 5.7 represents a single combination of parameters.

In Figure 5.7a, it can be seen that DE (in blue) has to lowest RSD values and setting different algorithm parameters mostly affect the best solution obtained. On the other hand, PSO (in green) has higher RSD values as well as higher P_{net} globally with most of the algorithm parameters tested. Nonetheless, the maximal net power achieved does not change significantly with the PSO settings. ACO_R (in red) appears as a “compromise” between PSO and DE. In Figure 5.7a, it can be seen that ACO_R has most of its points (in red)

clustered between DE (in blue) and PSO (in green) for the RSD value. In addition, most of the cluster of points is located close to higher power output values. Therefore, results from Figure 5.7a suggest that ACO_R has both higher repeatability and better capacity to reach good solutions with multiple algorithm parameter combinations. In contrast, GA (in black) has a more spread out cluster and most of its points reach lower net power output as well as higher RSD .

In Figure 5.7b, the net power output versus Ite is shown for each combination of algorithm parameters. It can be seen that DE (in blue) has the slowest convergence rate with most of its parameter combinations. In general, both GA and PSO required around 25 to 50 iterations. Similar to their RSD , changing the algorithm parameters for these algorithms has a significant impact on the convergence speed but still remains faster than DE. As for ACO_R , Figure 5.7b reveals that the choice of algorithm parameters has an important impact on the convergence speed. In fact, the required numbers of iterations for ACO_R vary from 5 to 95. In addition, it can be seen that the points with the lowest or the highest numbers of iterations required lead to less net power output. Therefore, the convergence rate and level of optimality are highly affected by the ACO_R setting.

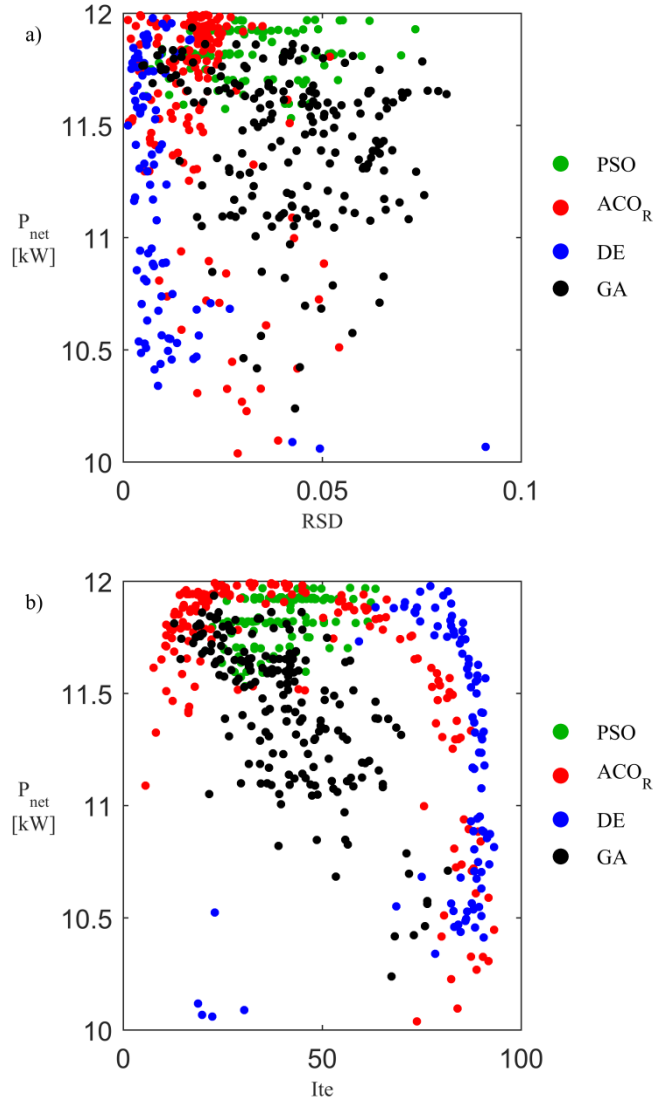


Figure 5.7: Effect of the algorithm parameters on a) RSD and b) the required number of iterations using GA, PSO, ACOR and DE for all the settings tested.

5.4.2. Algorithm performance comparison with proper algorithm parameters

Based on the previous figures, a set of algorithm parameters has been chosen for each algorithm. These choices offered a satisfying trade-off between the capability to achieve nearly optimal solutions, reliability and convergence rate. The selected algorithm parameters, the maximal objective function value and the RSD of each algorithm are reported in Table 5.3. The convergence of GA, PSO, ACO_R, DE and TLBO with the algorithm parameters from Table 5.3 is reported in Figure 5.8. The solid lines are the average solution of all five runs at each iteration while the gray areas are the range between the best and worst solution among the five runs. The size of the gray area is directly related to the RSD .

Table 5.3: Best parameters for nearly optimal net power output

| Alg. | Best parameters | | $\max(P_{net})$ [W] | RSD |
|------|-----------------|------------|------------------------|--------|
| GA | % Crossover | 0.70 | 11,936 | 0.0173 |
| | Crossover fct. | Tournament | | |
| | Selection fct. | Scattered | | |
| PSO | $w_{inertia}$ | 1 | 11,928 | 0.0217 |
| | w_{damp} | 0.99 | | |
| | c_1 | 0.6 | | |
| | c_2 | 1 | | |
| ACOR | q | 0.006 | 11,977 | 0.0109 |
| | ζ | 0.4 | | |
| DE | β | 0.02 | 12,004 | 0.0091 |
| | pCR | 0.8 | | |
| TLBO | - | - | 11,817 | 0.0083 |

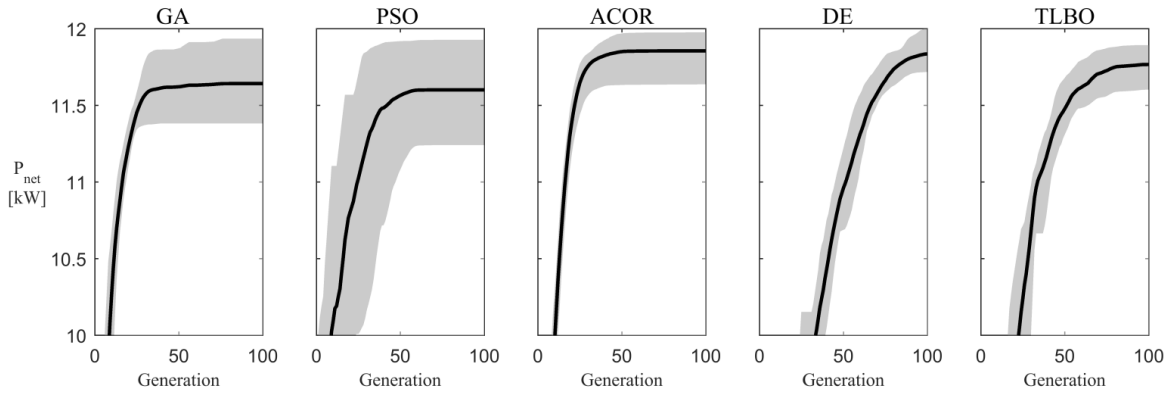


Figure 5.8: Convergence comparison of the net power optimization with algorithm parameters of Table 5.3. The lines are the mean value and the gray areas are the min-max range.

It can be seen in Figure 5.8 that the convergence of both DE and TLBO is slower. Furthermore, DE even seems not to have fully converged even after 100 iterations. Nonetheless, DE has a smaller “gray area” revealing its low RSD value. Moreover, DE was able to achieve the best solution of all (i.e. a net power of 12,004 W). Even if TLBO required no algorithm parameters to set, it has to perform twice the number of objective function evaluations making it more time-consuming. Globally, it can be concluded that DE is highly

reliable and reaches good solutions, but converges slowly. On the contrary, TLBO has slow convergence and lower capability to reach nearly optimal solutions.

The low rate of convergence of DE and TLBO can be compensated with GA, PSO and ACO_R. However, the gray area of PSO in Figure 5.8 is important, meaning that different runs might be required and that it is likely that the solution of a run will not be the global minima. Regardless, a wide range of coefficient c_1 and c_2 can be selected to reach similar solutions for easier tuning as shown previously in Figure 5.7. For similar mean P_{net} reach as PSO, GA is another interesting option requiring less number of iterations and having a better repeatability as presented in Figure 5.8. Overall, the algorithm that reaches the highest solution is DE, the algorithm that has the fastest convergence speed is ACO_R, the algorithm that as the best reliability is DE and TLBO. Other studies on DE compared to other population-based heuristics support this conclusion [47][91][92]. The main reason for the superiority of DE is because it does not use a random variation operator like in GAs and ACO_R. The random variation rather occurs from the selection and recombination of individuals in the population where only better individuals are replaced in the population.

The conclusions on the performance of all five algorithms are summed up into Figure 5.8. The “robustness to parameter variation” is an added criterion that evaluates how sensitive an algorithm is to changes of its internal setting. This qualitative criterion is assessed from the objective function RSD from all the algorithm parameter combinations. Figure 5.9 expresses that PSO is the only algorithm where changing the algorithm parameters has little to no impact on the solution reach.

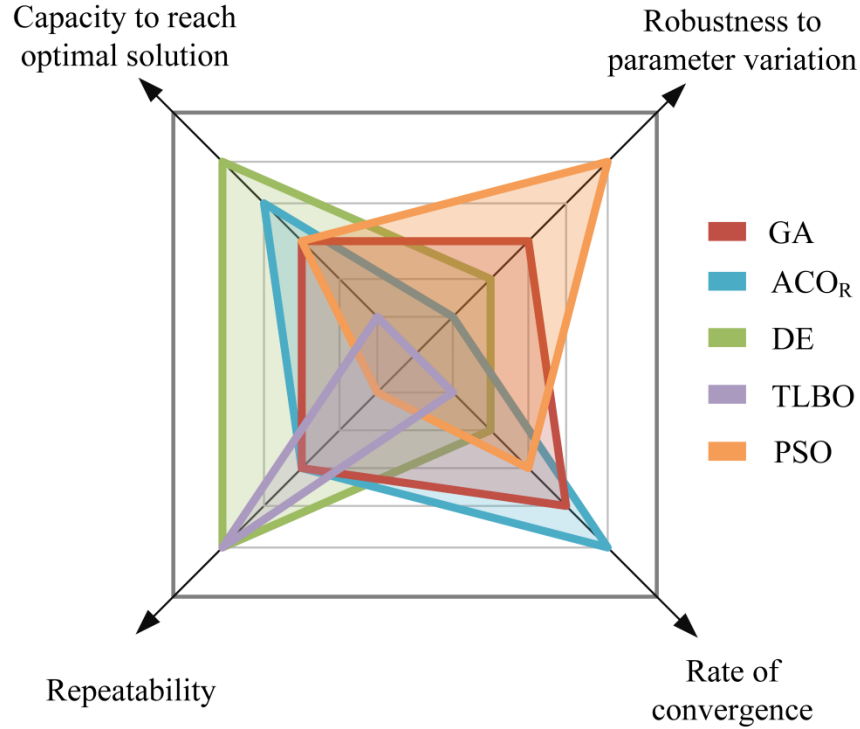


Figure 5.9: Summary of the heuristic algorithm performances for the problem tested.

5.5. Impact of heuristics on sets of non-dominated solutions

The multi-objective optimization introduced above is solved with the five heuristics compared here using the algorithm parameters found in Table 3. As mentioned before, the weighted sum method is used to build a single objective function combining the original three objectives, see Eq. (5.1). Optimizations are done for every combination of weights s_i between $[0:0.2:1]$ for a total of 216 optimization problems for each algorithm. The same number of individuals and maximal number of iterations as above is used in this section. Once again, five different runs are done for each algorithm for a total of 5400 optimization runs. This section compares the sets of solutions that are obtained as a function of the optimization algorithms.

5.5.1. Creation of Pareto front

With the total of 5400 solutions obtained, it is possible to create the equivalent of a Pareto front. A non-dominated sorting is performed on the 5400 points to keep only the non-dominated points (1077 points in this case) which were then reported in Figure 5.10. This represents a good estimation of the Pareto front for this problem. In fact, Figure 5.10 was compared to the multi-objective optimization solutions for the same thermoelectric model reported in Ref. [55]. Both sets of non-dominated points were found to have nearly identical hyperspace surfaces over the solution space. This demonstrates the validity of the weighted sum method used for this test case.

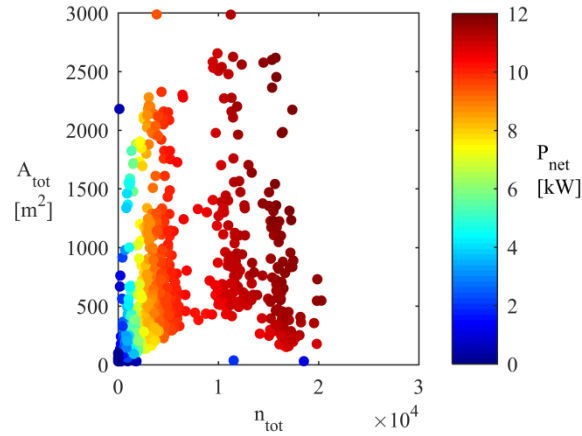


Figure 5.10: Approximate Pareto front P^* after the non-dominated sorting obtained with the sum-weight method with all solutions from GA, PSO, ACOR, TLBO and DE combined.

The Pareto front P^* from Figure 5.10 is obtained by selecting non-dominated points from the solutions provided by all the algorithms. It is interesting to find out which points from P^* originate from which algorithm along with their location over the Pareto front. In Figure 5.11, the solutions are represented on the Pareto front as a function of the algorithm from which they originate. In addition, the percentage of points on P^* provided by each algorithm is reported in Table 5.4. These percentages were calculated to determine which algorithms were able to find the best solutions with the weighted summation approach. It was found that all heuristics were able to provide non-dominated solutions, with DE providing the most (32%) and PSO, the least (10%).

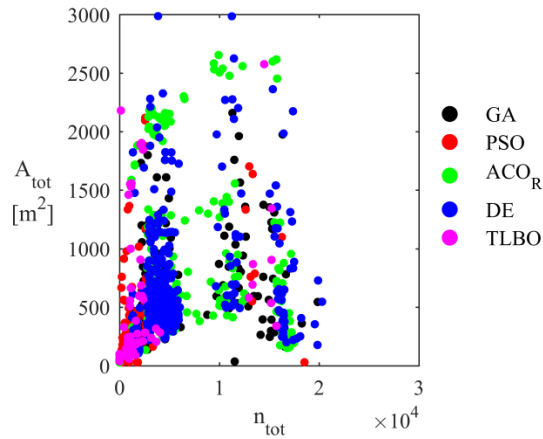


Figure 5.11: Origin of the non-dominated solutions achieved by all algorithms.

5.5.2. Metric for convergence

To evaluate the performance of each algorithm, an approach similar to the metric for convergence of Ref. [93] is used. The convergence metric typically evaluates the convergence at each generation of a given set of

points to a reference set which is the allegedly “true” Pareto front (Figure 5.10 or P^* in this case). With the weighted sum method, each optimization (i.e. each point on the Pareto front) is independent and, thus, a convergence analysis does not provide any useful information on the convergence speed. Instead, only the final front of each algorithm was compared to the true Pareto front.

The set of points from Figure 5.10 forms a reference set P^* . For each point i of each run of each algorithm, the smallest normalized Euclidian distance d_i from P^* is calculated with:

$$d_i = \min_{j=1}^{|P^*|} \sqrt{\sum_{k=1}^M \left(\frac{f_k(i) - f_k(j)}{f_k^{\max} - f_k^{\min}} \right)^2} \quad (5.2)$$

where $f_k(i)$ are the function values of the objective k of point i from the algorithm studied, $f_k(j)$ are the function values of the objective k of point j from P^* , f_k^{\min} and f_k^{\max} are the minimal and maximal function values of objective k from the set P^* . The averaged distance is the retained value for each of the five optimizations for each algorithm:

$$C = \frac{\sum_{i=1}^{|F|} d_i}{|F|} \quad (5.3)$$

where F is the total number of points on the front for the chosen algorithm. The lower the value of C , the closer the algorithm the Pareto front of Figure 5.10. Since 5 runs are performed for each algorithm, a total of 25 metric values C are calculated. Then, to keep the metric values between 0 and 1, C are normalized by the maximal value obtained from the worst algorithm (C_n). Therefore, the worst algorithm has a score of 1 while the others are below.

5.5.3. Metric of diversity

The weighted sum method does not guarantee the diversity in solutions over the Pareto front. Even with a proper method for generating weigh ratios, the distribution of solutions over P^* might vary from one algorithm to another. This can be seen from Figure 5.11 where TLBO points (in pink) are all clustered in one area and DE points (in blue) are more evenly distributed across the Pareto front. Another metric can be used to evaluate the diversity of the non-dominated solutions originating from each algorithm on the front of Figure 5.10. In other words, this metric estimates how well each algorithm can generate good solutions all along P^* with a

weighted sum method. The metric of diversity is based on a similar approach of Ref. [94]. The concept of Shannon's entropy (or information theory) is applied to measure how a set of points is spread across a feasible region. A set of solutions with higher entropy means that the solutions have a better coverage of the solution space.

First, the three dimensional feasible domain denoted by F''' is normalized and then subdivided into an $a_1 \times a_2 \times a_3$ grid of cells. The subdivision is determined so that the decision-maker is indifferent to solutions within the same cell. The feasible domain is delimited by the extreme solutions from the approximate Pareto front from Figure 10. For a given cell, an influence function Ω to each solution is defined. The influence function is a decreasing function of the distance between the specific cell and that solution. In this paper, Ω is defined as a normal distribution by the Gaussian function given as follow:

$$\Omega = \frac{1}{\sigma \sqrt{2\pi}} e^{-0.5 \left(\frac{r}{\sigma} \right)^2} \quad (5.4)$$

where N is the total number of solutions and $\Omega_s(r_{s \rightarrow c})$ is the influence function evaluated from the Euclidian distance between the solution s and the cell c .

Afterwards, the density functions for each cells is turned into a normalized density function:

$$D_{ijk}(c) = \sum_{s=1}^N \Omega_s(r_{s \rightarrow c}) \quad (5.5)$$

where N is the total number of solutions and $\Omega_s(r_{s \rightarrow c})$ is the influence function evaluated from the Euclidian distance between the solution s and the cell c .

Afterwards, the density functions for each cells is turned into a normalized density function:

$$\rho_{ijk} = \frac{D_{ijk}}{\sum_{k_1=1}^{a_1} \sum_{k_2=1}^{a_2} \sum_{k_3=1}^{a_3} D_{k_1 k_2 k_3}} \quad (5.6)$$

At last, the entropy H is defined as follow:

$$H = - \sum_{k_1=1}^{a_1} \sum_{k_2=1}^{a_2} \sum_{k_3=1}^{a_3} \rho_{k_1 k_2 k_3} \ln(\rho_{k_1 k_2 k_3}) \quad (5.7)$$

where $\rho_{k_1 k_2 k_3} \ln(\rho_{k_1 k_2 k_3})$ is assumed zero when $\rho_{k_1 k_2 k_3} = 0$. Then, to keep the metric values between 0 and 1, H is normalized by the maximal value obtained from the best algorithm (H_n). One should note that, any solution within a single cell is considered the same. Therefore the division of the feasible area is performed in order to keep the number of solutions within a cell to a minimal while reducing the computational time. Here, the solution space is subdivided into $20 \times 20 \times 20$ cells. A value of σ of 0.01 is selected in this test case.

5.5.3. Assessment of heuristics based on convergence and diversity metrics

One can see from Figure 5.11 that all the algorithms are present on the Pareto front P^* in the sense that all algorithms were able to generate non-dominated solutions. However, the number and the distribution of the non-dominated points of each algorithm on P^* is different. The metrics presented are quantitative tools that provide useful information on the performance of each heuristic algorithm analysed in this paper. For each algorithm, the normalized mean values C_μ calculated from C_n and the normalized entropy H_n are reported in Table 5.4. From Table 5.4, the value of the relative mean convergence metric C_μ of each algorithm is roughly similar (values over 0.9) which shows that the front achieved with each heuristic has a similar distance from the allegedly "true" Pareto front. Nonetheless, TLBO stands out from the other algorithms with C_μ of 0.84, which indicates that it is closer to the Pareto front of Figure 5.10. However, from Figure 5.11, it is visible that the best solutions provided by TLBO are clustered in the lower left corner of the Pareto front which is when w_i values are near zero, whereas all the other algorithms cover more of the Pareto front. Therefore, from the metric C_μ alone, even if some algorithms like TLBO could appear to be a better choice to reach optimal solutions, the algorithm might not provide the best solutions over the entire solution domain. In contrast, it can be seen from Table 5.4 that other algorithms, such as DE and ACO_R, have a higher value of C_μ , but are more spread out over the solution domain with their higher value of H_n .

The normalized entropy value H_n also supports this statement. In fact, the normalized metric value H_n of TLBO (0.77) is the lowest while DE and ACO_R have the highest values (1.00 and 0.98 respectively). In addition, both DE and ACO_R have generated more points on the set P^* with 32.7% and 21.3% respectively.

Table 5.4: Normalized mean metric value C_μ , normalized entropy H_n and percentage of points that belongs to the approximate Pareto front P^*

| Algorithm | C_μ | H_n | % <i>pts</i> |
|------------------|---------|-------|--------------|
| GA | 0.97 | 0.96 | 17.9 |
| PSO | 0.93 | 0.84 | 10.4 |
| ACO _R | 0.97 | 0.98 | 21.3 |
| DE | 0.99 | 1.00 | 32.7 |
| TLBO | 0.85 | 0.77 | 17.7 |

In sum, with the results shown in Table 4, it can be concluded that the five algorithms compared in this work were “functional” and did not yield completely different levels of performance for the test case problem. Therefore, the choice of algorithm is based on the decision-maker needs. That being said, DE outperforms the other algorithms in terms of reliability, capacity to reach optimality and diversity of solutions using the weighted sum method for the multi-objective optimization. However, it has been demonstrated that DE has a slow convergence speed and relatively high setting sensibility on reaching near-optimal solution in comparison to the other algorithms. ACO_R and GA have faster convergence speed, good capacity to reach optimality and good diversity. However, the algorithm parameters of ACO_R are difficult to tune. For easier tuning and adjustments, other algorithm like PSO and TLBO are appealing options. PSO has a fast convergence speed but poor reliability, while TLBO has slow convergence speed but high reliability. However, both algorithms have lower quality and diversity of near-optimal solutions when used with a weighted sum method.

5.6. Conclusion

The performances of five stochastic global search algorithms with weighted sum approach for solving multi-objective optimization problem are compared. A nonlinear mixed-variable constrained optimization test case is considered which consisted in optimizing the design of a thermoelectric device. This comparative analysis is conducted to highlight the forces and weaknesses of each algorithm and help the decision-maker in selecting the best algorithm for their specific optimization problem. The algorithms compared in this paper are: GA, PSO, ACO_R, TLBO and DE.

In a first step, the parameters of each algorithm influencing the intensification and diversification mechanisms have been analyzed to assess their impact on the capacity to reach optimal solutions, the convergence speed and the reliability. TLBO has been removed from this first analysis since the algorithm does not have any specific parameter to fine-tune. From this analysis, it can be concluded that the choice of proper parameters

for a specific problem can be influential, although some algorithms are more sensitive to the values of these settings than others. For ACO_R and DE, the solution reached is highly influenced by the choice of parameter values, whereas little effect has been observed on GA and PSO.

In a second step, the weighted sum method is used to solve the multi-objective optimization test case with proper settings. An approximation of the "true" Pareto front is obtained by performing a non-dominated sorting on the set of solutions obtained from every weight ratio combination with all five algorithms. Then, each algorithm is compared using convergence and diversity metrics. In general, all the algorithms are relatively close to the true Pareto front and yield comparable performances. Nonetheless, it has been observed that DE produced more solutions on the Pareto front as well as a highest level of diversity, followed by ACO_R.

The results of this paper provide a useful insight on the selection of heuristic algorithms and their specific settings for similar complex design optimization problems. Future work could investigate other heuristics and perform this comparison with other types of problems.

Chapitre 6 : Conclusion

L'un des objectifs principaux de ce mémoire était la modélisation d'un générateur thermoélectrique incluant différentes méthodes de refroidissement. Ce modèle se base sur les équations fondamentales de la thermoélectricité présentées au chapitre 2. De plus, ce modèle inclut une contrainte importante au niveau de la source de chaleur en considérant un flux de chaleur et une température de surface que le générateur thermoélectrique doit maintenir constante. Par cette contrainte, on s'assure que le générateur ne refroidit pas la source de chaleur ce qui peut avoir un attrait pour certains cas d'application. De plus, le modèle est général et peut donc être adapté à plusieurs cas d'application. Le design proposé a la particularité de subdiviser le générateur en plusieurs volumes de contrôle dont le choix des matériaux et les connexions électriques peuvent être adaptés. De plus, ce modèle comprend plusieurs variables de designs mixtes dont le courant électrique, les connexions électriques, la géométrie du collecteur thermique, les matériaux thermoélectriques, la géométrie des modules, le débit capacitif et le coefficient de transfert global du collecteur thermique. Une optimisation thermoéconomique du modèle est ensuite proposée pour maximiser la conversion thermoélectrique tout en minimisant les coûts liés au système.

Un algorithme génétique a été proposé pour effectuer l'optimisation du design. Cet outil d'optimisation permet d'obtenir des designs optimaux en ne calculant qu'une fraction de l'ensemble de l'espace de design. Ceci s'avère être avantageux lorsque le problème comprend de nombreuses variables de design (et donc un grand espace de design). L'algorithme utilise des méthodes stochastiques. Conséquemment, plusieurs optimisations ont dû être lancées pour s'assurer de la validité des résultats.

Le chapitre 4 a présenté le modèle thermoélectrique développé ainsi que les optimisations effectuées. L'un des objectifs était d'explorer l'impact d'une contrainte de température et de flux de chaleur à la surface chaude sur les performances d'un générateur thermoélectrique. Il a été démontré qu'il n'est pas possible de sélectionner le courant électrique ainsi que le nombre de modules séparément. De plus, il existe seulement une combinaison du nombre de modules et de courants permettant d'atteindre les performances optimales. Toutefois, la présence de cette contrainte empêche le générateur d'atteindre les performances maximales atteignables. Néanmoins, il a été démontré par une optimisation multiobjectif qu'il est possible de réduire considérablement le nombre de modules requis ainsi que le format du collecteur thermique avec une réduction négligeable de la puissance thermoélectrique en sortie. Il est donc possible de réduire le format du système de refroidissement et les coûts du système tout en conservant la conversion d'énergie à son optimal.

Le second volet du mémoire portait sur l'exploration d'algorithmes heuristiques pour l'optimisation de problèmes complexes. L'un des objectifs du projet était d'implémenter sur Matlab différents algorithmes

heuristiques et de les comparer sur un problème d'optimisation complexe en utilisant le modèle thermoélectrique développé au chapitre 4. Afin de bien comprendre le fonctionnement de chaque algorithme heuristique présenté, le chapitre 3 a d'abord détaillé le fonctionnement général et les particularités de chacun d'entre eux.

Le chapitre 5 s'est concentré sur la comparaison de ces différents algorithmes heuristiques pour l'optimisation du modèle thermoélectrique. Tout d'abord, une analyse paramétrique est proposée pour explorer l'impact des différents réglages intrinsèques à chaque algorithme sur les différents critères de performance proposés. Cela a permis, en outre, de déterminer les paramètres d'algorithme optimaux pour le problème proposé et d'observer que le choix des paramètres joue un rôle important sur les mécanismes de convergence et la fiabilité de chaque algorithme. Une optimisation multiobjectif a ensuite été proposée pour résoudre le problème selon les différents objectifs d'optimisation déterminés au chapitre 4. Puisque les algorithmes heuristiques proposés sont mono-objectif, une méthode d'optimisation par poids pondérés est proposée pour effectuer l'optimisation multiobjectif. Un triage des points non dominés a ensuite été effectué afin d'obtenir l'équivalent d'un front de Pareto. Les résultats ont démontré que tous les algorithmes ont des performances acceptables. Néanmoins, certains de ces algorithmes démontrent une fiabilité supérieure ainsi que la capacité d'obtenir de meilleurs designs.

À la lumière des résultats obtenus dans ce mémoire, l'optimisation numérique d'un système thermoélectrique permet donc d'améliorer le design résultant avant son implémentation à des problèmes concrets. Considérant la faible efficacité d'un générateur thermoélectrique avec les technologies actuelles, un attrait accru est accordé à ces optimisations afin de rendre viables ces systèmes dans un éventuel futur pour des applications commerciales et industrielles. Ce modèle pourrait s'avérer être une première étape pour l'intégration de la thermoélectricité à des enveloppes intelligentes pour la récupération énergétique. Par exemple, il serait possible d'installer des générateurs thermoélectriques dans les parois d'un équipement ou l'enveloppe d'un bâtiment afin de récupérer l'énergie thermique tout en contrôlant la température interne. Également, l'utilisation de divers outils d'optimisation telle que l'implémentation d'algorithmes heuristiques s'avère être une approche intéressante et efficace pour résoudre ces problèmes complexes. Les résultats de la comparaison des algorithmes proposés permettront de choisir plus facilement un algorithme heuristique approprié pour résoudre un problème similaire plus rapidement et avec de meilleures solutions.

Dans le cadre de travaux futurs, il serait intéressant d'améliorer le modèle thermoélectrique. Par exemple, l'ajout de résistances de contact, de transfert convectif et radiatif en périphérie et la variation des propriétés thermoélectriques avec le changement de température à l'intérieur d'un module pourraient être des améliorations intéressantes pour augmenter la précision du modèle. Il serait également intéressant d'appliquer

le modèle à un cas concret afin de réellement apprécier les performances d'un tel système et d'en faire la comparaison avec des cas similaires de la littérature. Il serait aussi intéressant d'analyser une plus grande variété d'algorithmes sur plusieurs problèmes d'optimisation différents.

Bibliographie

- [1] D Rowe, Éd., *Thermoelectrics Handbook*. CRC Press, 2005.
- [2] C. Wu, « Analysis of waste-heat thermoelectric power generators », *Applied Thermal Engineering*, vol. 16, n° 1, p. 63-69, janv. 1996.
- [3] L.-D. Zhao, V. P. Dravid, et M. G. Kanatzidis, « The panoscopic approach to high performance thermoelectrics », *Energy Environ. Sci.*, vol. 7, n° 1, p. 251-268, déc. 2013.
- [4] *Introduction to thermoelectricity*. New York, NY: Springer Berlin Heidelberg, 2016.
- [5] C. Preechakul et S. Kheawhom, « Modified genetic algorithm with sampling techniques for chemical engineering optimization », *Journal of Industrial and Engineering Chemistry*, vol. 15, n° 1, p. 110-118, janv. 2009.
- [6] « MATLAB and OptimizationToolbox Release 2016b, The MathWorks, Inc., Natick, Massachusetts, United States. »
- [7] K. Deb, A. Pratap, S. Agarwal, et T. Meyarivan, « A fast and elitist multiobjective genetic algorithm: NSGA-II », *IEEE Transactions on Evolutionary Computation*, vol. 6, n° 2, p. 182-197, avr. 2002.
- [8] M. DORIGO, « Optimization, Learning and Natural Algorithms », *Ph.D. Thesis, Politecnico di Milano, Italy*, 1992.
- [9] K. Socha et M. Dorigo, « Ant colony optimization for continuous domains », *European Journal of Operational Research*, vol. 185, n° 3, p. 1155-1173, mars 2008.
- [10] J. Kennedy et R. Eberhart, « Particle swarm optimization », in *IEEE International Conference on Neural Networks, 1995. Proceedings, 1995*, vol. 4, p. 1942-1948 vol.4.
- [11] R. Storn et K. Price, « Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces », *Journal of Global Optimization*, vol. 11, n° 4, p. 341-359, déc. 1997.
- [12] R. V. Rao, V. J. Savsani, et D. P. Vakharia, « Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems », *Computer-Aided Design*, vol. 43, n° 3, p. 303-315, mars 2011.
- [13] T. Berntsson et A. Asblad, « IETS Industrial Energy-Related Technologies and Systems - Annex XV: Industrial Excess heat Recovery », 05-mai-2015. [En ligne]. Disponible sur: <http://www.iea-industry.org/ongoing-annexes/annex-15/19-annex-15.html>. [Consulté le: 13-janv-2017].
- [14] S. B. Riffat et X. Ma, « Thermoelectrics: a review of present and potential applications », *Applied Thermal Engineering*, vol. 23, n° 8, p. 913-935, juin 2003.
- [15] S. Bélanger et L. Gosselin, « Thermoelectric generator sandwiched in a crossflow heat exchanger with optimal connectivity between modules », *Energy Conversion and Management*, vol. 52, n° 8-9, p. 2911-2918, août 2011.
- [16] S. Bélanger et L. Gosselin, « Multi-objective genetic algorithm optimization of thermoelectric heat exchanger for waste heat recovery », *Int. J. Energy Res.*, vol. 36, n° 5, p. 632-642, avr. 2012.
- [17] P. Aranguren, D. Astrain, A. Rodríguez, et A. Martínez, « Net thermoelectric power generation improvement through heat transfer optimization », *Applied Thermal Engineering*, vol. 120, n° Supplement C, p. 496-505, juin 2017.
- [18] D. Astrain, J. G. Vián, A. Martínez, et A. Rodríguez, « Study of the influence of heat exchangers' thermal resistances on a thermoelectric generation system », *Energy*, vol. 35, n° 2, p. 602-610, févr. 2010.
- [19] A. Sarhadi, R. Bjørk, N. Lindeburg, P. Viereck, et N. Pryds, « A thermoelectric power generating heat exchanger: Part II – Numerical modeling and optimization », *Energy Conversion and Management*, vol. 119, p. 481-487, juill. 2016.
- [20] X. Meng et R. O. Suzuki, « Helical configuration for thermoelectric generation », *Applied Thermal Engineering*, vol. 99, p. 352-357, avr. 2016.

- [21] É. V. Sempels et F. J. Lesage, « Optimal Thermal Conditions for Maximum Power Generation When Operating Thermoelectric Liquid-to-Liquid Generators », *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 7, n° 6, p. 872-881, juin 2017.
- [22] V. Leonov et R. J. M. Vullers, « Wearable Thermoelectric Generators for Body-Powered Devices », *Journal of Elec Materi*, vol. 38, n° 7, p. 1491-1498, juill. 2009.
- [23] D. Champier, « Thermoelectric generators: A review of applications », *Energy Conversion and Management*, vol. 140, p. 167-181, mai 2017.
- [24] M. Nesarajah et G. Frey, « Optimized Design of Thermoelectric Energy Harvesting Systems for Waste Heat Recovery from Exhaust Pipes », *Applied Sciences*, vol. 7, n° 6, p. 634, juin 2017.
- [25] P. Aranguren, M. Araiz, D. Astrain, et A. Martinez, « Thermoelectric generators for waste heat harvesting: A computational and experimental approach », *Energy Conversion and Management*, vol. 148, p. 680-691, sept. 2017.
- [26] C. Yu et K. T. Chau, « Thermoelectric automotive waste heat energy recovery using maximum power point tracking », *Energy Conversion and Management*, vol. 50, n° 6, p. 1506-1512, juin 2009.
- [27] L. Kütt et M. Lehtonen, « Automotive waste heat harvesting for electricity generation using thermoelectric systems #x2014; An overview », in *2015 IEEE 5th International Conference on Power Engineering, Energy and Electrical Drives (POWERENG)*, 2015, p. 55-62.
- [28] J. W. Qiang, C. G. Yu, Y. D. Deng, C. Q. Su, Y. P. Wang, et X. H. Yuan, « Multi-Objective Optimization Design for Cooling Unit of Automotive Exhaust-Based Thermoelectric Generators », *Journal of Elec Materi*, vol. 45, n° 3, p. 1679-1688, mars 2016.
- [29] C. Q. Su, C. Huang, Y. D. Deng, Y. P. Wang, P. Q. Chu, et S. J. Zheng, « Simulation and Optimization of the Heat Exchanger for Automotive Exhaust-Based Thermoelectric Generators », *Journal of Elec Materi*, vol. 45, n° 3, p. 1464-1472, oct. 2015.
- [30] E. A. Ibrahim, J. P. Szybist, et J. E. Parks, « Enhancement of automotive exhaust heat recovery by thermoelectric devices », *Proceedings of the IMechE*, vol. 224, n° 8, p. 1097-1111, août 2010.
- [31] J. H. Panya Yodovard Joseph Khedari, « The Potential of Waste Heat Thermoelectric Power Generation From Diesel Cycle and Gas Turbine Cogeneration Plants », *Energy Sources*, vol. 23, n° 3, p. 213-224, avr. 2001.
- [32] T. Hendricks et W. T. Choate, « Engineering scoping study of thermoelectric generator systems for industrial waste heat recovery », Pacific Northwest National Lab.(PNNL), Richland, WA (United States), 2006.
- [33] W. T. Choate, T. J. Hendricks, et R. Majumdar, « Waste Heat Recovery Using Thermoelectric Devices in the Light Metals Industry », Pacific Northwest National Laboratory (PNNL), Richland, WA (US), PNNL-SA-53260, mai 2007.
- [34] M. Fleer, O. Lorentsen, W. Harvey, H. Palsson, et G. Saevarsdottir, « Heat recovery from the exhaust gas of aluminum reduction cells. », Reykjavik University, School of Science and Engineering/REYST, 2010.
- [35] R. Zhao, « Analysis, Simulation and Optimization of Ventilation of Aluminum Smelting Cells and Potrooms for Waste Heat Recovery », Université Laval, 2015.
- [36] S. Namboothiri, P. Lavoie, D. Cotton, et M. P. Taylor, « Controlled cooling of aluminium smelting cell sidewalls using heat exchangers supplied with air », *Light Metals*, p. 317-322, 2009.
- [37] F. P. Incropera, D. D. Dewitt, T. L. Bergman, et A. S. Lavine, *Fundamentals of Heat and Mass Transfer*, 6th edition. John Wiley & Sons, Inc., 2007.
- [38] L. Chen, F. Meng, et F. Sun, « Thermodynamic analyses and optimization for thermoelectric devices: The state of the arts », *Sci. China Technol. Sci.*, vol. 59, n° 3, p. 442-455, janv. 2016.
- [39] X. Hu, H. Takazawa, K. Nagase, M. Ohta, et A. Yamamoto, « Three-Dimensional Finite-Element Simulation for a Thermoelectric Generator Module », *Journal of Elec Materi*, vol. 44, n° 10, p. 3637-3645, oct. 2015.

- [40] « Thermoelectric Generator Power Bulk Materials BiTe, PbTe - Tecteg Power Generator.com ». [En ligne]. Disponible sur: <http://tecteg.com/thermoelectric-generator-power-bulk-materials-bite-pbte/>. [Consulté le: 26-sept-2017].
- [41] G. Jeffrey Snyder, « Thermoelectric Power Generation », in *Thermoelectrics Handbook*, 0 vol., CRC Press, 2005, p. 9-1-9-26.
- [42] B. R. Munson, D. F. Young, et T. H. Okiishi, *Fundamentals of fluid mechanics*, 2nd ed. New York: Wiley, 1994.
- [43] L. T. Biegler et I. E. Grossmann, « Retrospective on optimization », *Computers & Chemical Engineering*, vol. 28, n° 8, p. 1169-1192, juill. 2004.
- [44] L. Rios et N. Sahinidis, « Derivative-free optimization: a review of algorithms and comparison of software implementations », *Journal of Global Optimization*, vol. 56, n° 3, p. 1247-1293, juill. 2013.
- [45] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, et A. Mahajan, « Mixed-integer nonlinear optimization*† », *Acta Numerica*, vol. 22, p. 1-131, mai 2013.
- [46] C. Blum et A. Roli, « Metaheuristics in combinatorial optimization: Overview and conceptual comparison », *ACM computing surveys (CSUR)*, vol. 35, n° 3, p. 268-308, 2003.
- [47] S. Paterlini et T. Krink, « Differential evolution and particle swarm optimisation in partitional clustering », *Computational Statistics & Data Analysis*, vol. 50, n° 5, p. 1220-1247, mars 2006.
- [48] C. Oguz et M. F. Ercan, « A Genetic Algorithm for Hybrid Flow-shop Scheduling with Multiprocessor Tasks », *Journal of Scheduling*, vol. 8, n° 4, p. 323-351, 2005.
- [49] A. C. Zecchin, A. R. Simpson, H. R. Maier, et J. B. Nixon, « Parametric study for an ant algorithm applied to water distribution system optimization », *IEEE Transactions on Evolutionary Computation*, vol. 9, n° 2, p. 175-191, avr. 2005.
- [50] M. Birattari, « Tuning Metaheuristics: A Machine Learning Perspective », *Tuning Metaheuristics*, p. 226.
- [51] Z. Yuan, M. A. Montes de Oca, M. Birattari, et T. Stützle, « Continuous optimization algorithms for tuning real and integer parameters of swarm intelligence algorithms », *Swarm Intelligence*, vol. 6, n° 1, p. 49-75, mars 2012.
- [52] R. T. Marler et J. S. Arora, « Survey of multi-objective optimization methods for engineering », *Struct Multidisc Optim*, vol. 26, n° 6, p. 369-395, avr. 2004.
- [53] I. Das et J. E. Dennis, « A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems », *Structural Optimization*, vol. 14, n° 1, p. 63-69, août 1997.
- [54] *Metaheuristics*. New York, NY: Springer Berlin Heidelberg, 2016.
- [55] M. Allyson-Cyr et L. Gosselin, « Optimization of thermoelectric generators to recover heat from a hot surface while maintaining its temperature and heat flux distribution », *Applied Thermal Engineering*, submitted 2018.
- [56] J. H. Holland, *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence*, 1st MIT Press ed. Cambridge, Mass: MIT Press, 1992.
- [57] L. Gosselin, M. Tye-Gingras, et F. Mathieu-Potvin, « Review of utilization of genetic algorithms in heat transfer problems », *International Journal of Heat and Mass Transfer*, vol. 52, n° 9-10, p. 2169-2188, avr. 2009.
- [58] W. Huang et H. N. Lam, « Using genetic algorithms to optimize controller parameters for HVAC systems », *Energy and Buildings*, vol. 26, n° 3, p. 277-282, janv. 1997.
- [59] K. Deep, K. P. Singh, M. L. Kansal, et C. Mohan, « A real coded genetic algorithm for solving integer and mixed integer optimization problems », *Applied Mathematics and Computation*, vol. 212, n° 2, p. 505-518, juin 2009.
- [60] S. S. Rao et Y. Xiong, « A Hybrid Genetic Algorithm for Mixed-Discrete Design Optimization », *J. Mech. Des*, vol. 127, n° 6, p. 1100-1112, oct. 2004.
- [61] C. Guo, J. Hu, B. Ye, et Y. Cao, « Swarm intelligence for mixed-variable design optimization », *J. Zhejiang Univ. Sci. A*, vol. 5, n° 7, p. 851-860, juill. 2004.

- [62] A. P. Silva, M. A. S. S. Ravagnani, E. C. Biscaia, et J. A. Caballero, « Optimal heat exchanger network synthesis using particle swarm optimization », *Optim Eng*, vol. 11, n° 3, p. 459-470, sept. 2010.
- [63] M. Yousefi, M. Yousefi, R. P. Martins Ferreira, et A. N. Darus, « A swarm intelligent approach for multi-objective optimization of compact heat exchangers », *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, vol. 231, n° 2, p. 164-171, avr. 2017.
- [64] D. R. O. G. L. A. L. de Almeida, et O. A. C. Vilcanqui, « Parameter identification of thermoelectric modules using particle swarm optimization », in *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, 2015, p. 812-817.
- [65] A. Ibrahim, S. Rahnamayan, M. Vargas Martin, et B. Yilbas, « Multi-objective thermal analysis of a thermoelectric device: Influence of geometric features on device characteristics », *Energy*, vol. 77, p. 305-317, déc. 2014.
- [66] Udayraj, K. Mulani, P. Talukdar, A. Das, et R. Alagirusamy, « Performance analysis and feasibility study of ant colony optimization, particle swarm optimization and cuckoo search algorithms for inverse heat transfer problems », *International Journal of Heat and Mass Transfer*, vol. 89, n° Supplement C, p. 359-378, oct. 2015.
- [67] S. He, E. Prempan, et Q. H. Wu, « An improved particle swarm optimizer for mechanical design optimization problems », *Engineering Optimization*, vol. 36, n° 5, p. 585-605, oct. 2004.
- [68] J. Clarke, L. McLay, et J. T. McLeskey, « Comparison of genetic algorithm to particle swarm for constrained simulation-based optimization of a geothermal power plant », *Advanced Engineering Informatics*, vol. 28, n° 1, p. 81-90, janv. 2014.
- [69] A. J. Nebro, J. J. Durillo, J. Garcia-Nieto, C. A. C. Coello, F. Luna, et E. Alba, « SMPSO: A new PSO-based metaheuristic for multi-objective optimization », in *2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making(MCDM)*, 2009, p. 66-73.
- [70] M. Clerc et J. Kennedy, « The particle swarm - explosion, stability, and convergence in a multidimensional complex space », *IEEE Transactions on Evolutionary Computation*, vol. 6, n° 1, p. 58-73, févr. 2002.
- [71] Z. H. Zhan, J. Zhang, Y. Li, et H. S. H. Chung, « Adaptive Particle Swarm Optimization », *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, n° 6, p. 1362-1381, déc. 2009.
- [72] R. Eberhart et J. Kennedy, « A new optimizer using particle swarm theory », in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995. MHS '95*, 1995, p. 39-43.
- [73] M. Dorigo, G. Di Caro, et L. M. Gambardella, « Ant Algorithms for Discrete Optimization », *Artificial Life*, vol. 5, n° 2, p. 137-172, Spring 1999.
- [74] E. Hetmaniok, D. Słota, et A. Zielonka, « Determination of the Heat Transfer Coefficient by Using the Ant Colony Optimization Algorithm », in *Parallel Processing and Applied Mathematics*, 2011, p. 470-479.
- [75] B. Zhang, H. Qi, Y.-T. Ren, S.-C. Sun, et L.-M. Ruan, « Application of homogenous continuous Ant Colony Optimization algorithm to inverse problem of one-dimensional coupled radiation and conduction heat transfer », *International Journal of Heat and Mass Transfer*, vol. 66, n° Supplement C, p. 507-516, nov. 2013.
- [76] I. C. Silva, F. R. do Nascimento, E. J. de Oliveira, A. L. M. Marcato, L. W. de Oliveira, et J. A. Passos Filho, « Programming of thermoelectric generation systems based on a heuristic composition of ant colonies », *International Journal of Electrical Power & Energy Systems*, vol. 44, n° 1, p. 134-145, janv. 2013.
- [77] B. Zhang, H. Qi, S.-C. Sun, L.-M. Ruan, et H.-P. Tan, « A novel hybrid ant colony optimization and particle swarm optimization algorithm for inverse problems of coupled radiative and conductive heat transfer », *Thermal Science*, vol. 20, n° 2, p. 461-472, 2016.

- [78] T. Liao, K. Socha, M. A. M. de Oca, T. Stützle, et M. Dorigo, « Ant Colony Optimization for Mixed-Variable Optimization Problems », *IEEE Transactions on Evolutionary Computation*, vol. 18, n° 4, p. 503-518, août 2014.
- [79] A. E. L. Rivas et L. A. G. Pareja, « Coordination of directional overcurrent relays that uses an ant colony optimization algorithm for mixed-variable optimization problems », 2017, p. 1-6.
- [80] R. V. Rao, V. J. Savsani, et J. Balic, « Teaching-learning-based optimization algorithm for unconstrained and constrained real-parameter optimization problems », *Engineering Optimization*, vol. 44, n° 12, p. 1447-1462, déc. 2012.
- [81] R. V. Rao, V. J. Savsani, et D. P. Vakharia, « Teaching-Learning-Based Optimization: An optimization method for continuous non-linear large scale problems », *Information Sciences*, vol. 183, n° 1, p. 1-15, janv. 2012.
- [82] R. V. Rao et V. Patel, « Multi-objective optimization of heat exchangers using a modified teaching-learning-based optimization algorithm », *Applied Mathematical Modelling*, vol. 37, n° 3, p. 1147-1162, févr. 2013.
- [83] R. Venkata Rao et V. Patel, « Multi-objective optimization of two stage thermoelectric cooler using a modified teaching-learning-based optimization algorithm », *Engineering Applications of Artificial Intelligence*, vol. 26, n° 1, p. 430-445, janv. 2013.
- [84] R. V. Rao et K. C. More, « Optimal design of the heat pipe using TLBO (teaching-learning-based optimization) algorithm », *Energy*, vol. 80, p. 535-544, févr. 2015.
- [85] F. Zou, L. Wang, X. Hei, D. Chen, et B. Wang, « Multi-objective optimization using teaching-learning-based optimization algorithm », *Engineering Applications of Artificial Intelligence*, vol. 26, n° 4, p. 1291-1300, avr. 2013.
- [86] B. V. Babu et S. A. Munawar, « Differential evolution strategies for optimal design of shell-and-tube heat exchangers », *Chemical Engineering Science*, vol. 62, n° 14, p. 3720-3739, juill. 2007.
- [87] F. Neri et V. Tirronen, « Recent advances in differential evolution: a survey and experimental analysis », *Artificial Intelligence Review*, vol. 33, n° 1/2, p. 61-106, févr. 2010.
- [88] S. Kukkonen et J. Lampinen, « GDE3: the third evolution step of generalized differential evolution », in *2005 IEEE Congress on Evolutionary Computation*, 2005, vol. 1, p. 443-450 Vol.1.
- [89] M. López-Ibáñez, J. Dubois-Lacoste, L. Pérez Cáceres, M. Birattari, et T. Stützle, « The irace package: Iterated racing for automatic algorithm configuration », *Operations Research Perspectives*, vol. 3, p. 43-58, 2016.
- [90] F. Hutter, H. H. Hoos, et K. Leyton-Brown, « Sequential Model-Based Optimization for General Algorithm Configuration », in *Learning and Intelligent Optimization*, vol. 6683, C. A. C. Coello, Éd. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, p. 507-523.
- [91] J. Vesterstrom et R. Thomsen, « A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems », in *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, 2004, vol. 2, p. 1980-1987 Vol.2.
- [92] R. K. Ursem et P. Vadstrup, « Parameter identification of induction motors using stochastic optimization algorithms », *Applied Soft Computing*, vol. 4, n° 1, p. 49-64, févr. 2004.
- [93] K. Deb et S. Jain, « Running performance Metrics for evolutionary multi-objective optimization », 2002.
- [94] A. Farhang-Mehr et S. Azarm, « Diversity assessment of Pareto optimal solution sets: an entropy approach », in *Proceedings of the 2002 Congress on Evolutionary Computation*, 2002. CEC '02, 2002, vol. 1, p. 723-728.