Minimisation des chutes de la matière première : un pas vers l'éco-conception

Youness FARHANE
LP2D EST,
USMBA
FES-Maroc
farhane.youness@yahoo.fr

Driss AMEGOUZ
LP2D, EST
USMBA
FES-Maroc
amegouz@yahoo.fr

Résumé —L'éco-conception joue un rôle important en favorisant le développement durable grâce à la création de richesse à l'échelle de la planète. Elle peut également ouvrir la voie à l'amélioration de connaissances, de technologies, de produits et de services qui favorisent la réalisation des objectifs économiques, environnementaux et sociaux qui assurent un développement durable. C'est dans cette perspective qu'on a mené des recherches pour contribuer à la minimisation des chutes lors de la fabrication : problèmes de découpe.

Cet article s'inscrit désormais dans une démarche d'éco-conception, en effet plusieurs applications peuvent être trouvées pour chacun des problèmes de découpe (1D, 2D et 3D), nous allons s'intéresser à l'optimisation de le découpe de matières premières en 2D (tôles, tissus, papier,...) ce qui va permettre de limiter la consommation de l'énergie de découpe et l'utilisation de la matière première, et de minimiser les chutes générées par la production de produits manufacturés. Tout ça aura sans doute un impact positif sur l'environnement, ce qui va constituer le premier pas vers l'intégration de l'éco-conception.

Dans cet article nous allons proposer une approche méta-heuristique pour la résolution de problèmes de géométrie complexe. Avant de présenter cette approche, les justifications qui nous ont conduits à ce choix sont énoncées. La méthode méta-heuristique se base sur des constats qu'on fait sur des polygones déjà existants dans une base de données bien définie.

Mots clés — ***Eco-conception, optimisation, algorithme, découpe, environnement*.**

I. INTRODUCTION

Les problèmes de découpe, ont fait l'objet de nombreuses études et recherches et ont généré une grande quantité de solutions et résultats. Les raisons à cela sont multiples. Tout d'abord, il existe de nombreuses variantes à ces problèmes, qui peuvent générer des modélisations et des méthodes de résolution déférentes. Ensuite, les enjeux industriels derrière toutes ces problématiques sont importants, entraînant une recherche perpétuelle sur ces sujets connus depuis de nombreuses années. Enfin, Les enjeux environnementaux ont été les moteurs initiaux de cette partie de nos recherches.

Les problèmes de placement ou de positionnement, regroupent une large gamme de variantes, à savoir les problèmes de découpe et de conditionnement et les problèmes d'agencement. Les traductions anglophones de ces termes sont respectivement Cutting & Packing problems (C&P) et Layout problems [1].

Il faut noté que le terme « placement » qu'on a et que nous allons utiliser pour toute la suite, sera utilisé pour indiquer la répartition d'un ensemble de géométrie sur une autre géométrie (contrainte), ce « placement » sera utilisé comme une trajectoire (modèle CAO) qui sera transmis ensuite (via un programme) à une MOCN pour la découpe ce qui va permettre d'optimiser la découpe de formes afin de minimiser les chutes de la matière première (tôle, papier, tissu,…).

Tout d'abord, les déférents éléments nécessaires à la présentation du problème de placement doivent être bien définis. Ceux-ci permettront par la suite de simplifier le problème de placement.

Avant de définir mathématiquement les problèmes de placement en détail, les déférents éléments les constituant doivent être définis, à savoir composants, attributs, variables et objectifs et contraintes [2].

Généralement, un composant est défini comme un élément à positionner caractérisé par une liste d'attributs. Dans le cadre de notre travail, les composants sont le plus souvent des formes non rectangulaires.

Les attributs expriment les propriétés des composants (formes primitives), ils sont utilisés pour évaluer les solutions de placement. Dans notre présente recherche et c'est d'ailleurs le cas de la plus part des problèmes rencontrés, ces formes primitives sont des données qui ne changent pas lors de la résolution. Les composants sont entièrement définis par leur géométrie et leurs attributs. La géométrie d'un composant peut être simplement une longueur pour un problème 1D, ou bien un modèle CAO complet pour les problèmes 2D ou 3D.

Un contenant est un élément défini avec plusieurs attributs, dans lequel l'ensemble des composants doit être positionné. Le contenant peut être de dimensions fixes ou variables [46]. Dans ce dernier cas, l'objectif du problème consiste à minimiser les dimensions du contenant, comme dans un problème de découpe. Enfin, le problème peut présenter plusieurs contenants. Cette situation peut soit signifier qu'il faut identifier le contenant où placer chaque composant, soit qu'il faut trouver le nombre minimum de contenants pouvant accueillir les composants [2].

Xème Conférence Internationale : Conception et Production Intégrées, CPI 2015, 2-4 Décembre 2015, Tanger - Maroc.

Xth International Conference on Integrated Design and Production, CPI 2015, December 2-4, 2015, Tangier - Morocco.

Les inconnues du problème sont les variables de positionnement (orientation, nombre, …), ainsi que certains attributs des composants. Comme on le verra par la suite, ces variables de positionnement peuvent être de déférents types et dépendent de la modélisation choisie.

Chaque problème de découpe présente au moins des contraintes de non-chevauchement entre les composants et des contraintes d'appartenance au contenant. Ces contraintes définissent les contraintes du problème de découpe. Des contraintes additionnelles de positionnement peuvent être ajoutées (par exemple des contraintes de couleur).

Les objectifs d'un problème de découpe sont ses critères d'évaluation. Ils sont exprimés comme des quantités numériques utilisées pour classer les déférentes solutions proposées. Les objectifs sont des fonctions mathématiques explicites ou des résultats numériques définis par le concepteur. C'est le cas lorsque les critères ne peuvent être modélisés explicitement par des fonctions mathématiques. Par exemple, les critères esthétiques qui font intervenir le point de vue subjectif du concepteur ou d'un expert, peuvent rarement être modélisés par une fonction mathématique. Dans la suite de ce travail, l'évaluation de chaque objectif est supposée renvoyer une valeur numérique (aires des chutes, aire du contenant).

II. FORMULATION DES PROBLEMES DE PLACEMENT

A. Définition des problèmes de placement :

Les définitions des déférents éléments d'un problème de placement permettent de définir ce dernier : Étant donné un ensemble de composants (formes primitives) ainsi qu'un ensemble de contenants (surfaces de base : carreaux de base, voir l'étude de cas), un problème de découpe consiste à trouver l'ensemble des variables de positionnement des composants afin de minimiser l'espace entre elles (minimisation des chutes), tout en respectant des contraintes (non-chevauchement et d'appartenance).

La découpe ou la solution de positionnement peut être vu comme une affectation de l'ensemble des variables de positionnement ainsi que des attributs. Une solution admissible est une affectation de toutes les variables, telle que l'ensemble des contraintes de découpe soit satisfait. Une solution réalisable sera dite optimale si aucune autre solution n'est meilleure.

B. Vocabulaire des problèmes de placement :

Lors du traitement d'un problème de placement ou de positionnement, plusieurs mots ou expressions sont utilisés pour désigner la même chose ou bien un même mot peut désigner déférentes choses. Le mot découpe a été utilisé pour définir une catégorie de problèmes, pour laquelle une série de composants faiblement hétérogène doit être affectée à un ensemble fini de contenants [3]. Dans ce travail ainsi que dans de nombreux travaux, le mot découpe fait référence à une solution pour minimiser les chutes.

Les déférents éléments d'un problème de placement ont été utilisés dans déférents contextes. Par exemple, dans les problèmes de bin packing, les termes objets, items ou encore pièces sont utilisés pour désigner les composants [2]. Dans les problèmes d'intégration à grande échelle, le terme module désigne un composant rectangulaire [4].

C. Caractéristiques des problèmes de placement :

Cette section présente les déférents aspects de modélisation des problèmes de placement.

1) Techniques de placement :

Une fois que les problèmes de découpe ont été définis, les techniques de placement doivent être mise en place pour positionner l'ensemble des composants.

Deux techniques de placement ont été élaborées : les méthodes de placement légal et relaxé.

Les méthodes de placement relaxé [48] autorisent le non-respect des contraintes de placement lors de la construction de la solution. Les méthodes légales garantissent le non-chevauchement des composants, en utilisant des techniques de placement pour générer les solutions. Le terme relaxé fait référence aux contraintes de placement qui sont relaxées lors de la résolution des problèmes. Si une telle méthode est utilisée, les contraintes de placement peuvent ne pas être satisfaites à la fin de l'optimisation.

Le choix de la méthode de placement n'est pas dicté par la géométrie des composants, mais plutôt par le type d'objectifs et de contraintes du problème.

2) Représentation des variables :

Les variables de positionnement sont les inconnues du problème de placement. Ces variables peuvent être de déférents types : translations, rotations voire même des permutations. Il existe plusieurs méthodes pour représenter la position et l'orientation des composants dans un problème de placement. Ce choix dépend de plusieurs critères, comme la géométrie des composants, des objectifs et contraintes du problème. Ce choix détermine la topologie de l'espace de recherche, et conditionne les algorithmes d'optimisation qui pourront être utilisés par la suite [2].

3) Les objectifs des problèmes de placement :

Les objectifs et les contraintes sont les critères utilisés pour qualifier les solutions. L'expression mathématique des objectifs et contraintes est cruciale. Les objectifs peuvent être de trois types [4] :

- Objectifs individuels : concerne un seul composant
- Objectifs globaux : s'appliquant sur l'ensemble des composants)
- Objectifs d'interaction (un objectif d'interaction fait intervenir une relation entre composants).

Pour tous les problèmes de découpe, l'objectif ou les objectifs sont globaux et s'exprimeront sur l'ensemble des composants.

4) Les contraintes des problèmes de placement

Pour tous les problèmes d'optimisation, les contraintes permettent d'identifier les solutions réalisables et irréalisables. Pour les problèmes de placement, trois types de contraintes peuvent être identifiées :

- Celles qui consistent à bloquer certains degrés de liberté des composants ;
- Celles qui s'expriment comme une combinaison de plusieurs paramètres ;
- Celles qui sont une combinaison d'objectifs.

Naturellement, les premières sont les plus simples à satisfaire. De plus, ces restrictions réduisent le nombre de degrés de libertés et par conséquent l'espace de recherche. Les secondes peuvent être vérifiées avant d'évaluer les objectifs. Enfin, les contraintes du troisième type nécessitent d'évaluer les objectifs pour savoir si les contraintes sont satisfaites.

Les contraintes de non-chevauchement entre composants ainsi que les contraintes d'appartenance des composants au contenant définissent les contraintes communes à tous les problèmes de déplacement.

5) Évaluation des contraintes de non-chevauchement et d'appartenance

Dans les problèmes de placement 2D, les calculs de chevauchement et d'appartenance peuvent être très complexes et représenter la majorité du temps de calcul. Pour un problème avec m composants, un nombre de « m! » tests sont nécessaires pour détecter le chevauchement et l'appartenance [6], ce qui va nous donner une idée sur le temps nécessaire pour traiter un problème de placement.

L'idéal consiste à utiliser une méthode de découpe avec des contraintes de non-chevauchement qui sont automatiquement satisfaites. Elle doit tout de même s'assurer que tous les composants ont pu être placés. Dans le cas où une telle méthode ne peut être utilisée, il faut s'assurer que chaque composant est bien à l'intérieur du contenant et qu'aucune paire de composants ne se chevauche. Ces tests peuvent être effectués de déferentes manières en fonction de la représentation géométrique choisie.

III. PROBLEMES DE DECOUPE

Les problèmes de découpe trouvent de nombreuses applications dans l'industrie, et présentent des intérêts économiques et écologiques certains, vu qu'ils aident à optimiser l'utilisation de la matière première et la consommation d'énergie. Les problèmes de découpe et de conditionnement (Cutting and Packing problems), forment une famille de problèmes à part entière. Dans la littérature française, l'expression découpe et empaquetage est aussi employée. Ils sont abréviés par la suite C&P.

Ce sont des problèmes dont les formulations verbale et mathématique sont simples, mais pour lesquels la combinatoire rend ces problèmes difficiles à résoudre. Les premiers travaux remontent à plus de quarante ans. Depuis, plusieurs classifications de ces problèmes ont été entreprises [5].

A. Définition des problèmes de découpe :

Les problèmes de découpe sont des problèmes d'optimisation, pour lesquels les composants sont uniquement géométriquement reliés entre eux. En d'autres termes, il n'y a aucune interaction explicite entre les composants. Tous les composants présentent les mêmes types d'attributs. Objectifs et contraintes peuvent toujours être modélisés comme des fonctions mathématiques et sont définis globalement, c'est-à-dire qu'ils impliquent l'ensemble des attributs des composants.

L'idée derrière chaque problème est la compacité : soit le nombre de composants à placer dans le contenant doit être maximisé, soit le nombre de contenants ou les dimensions de ceux-ci doivent être minimisées. Par conséquent, chaque schéma de résolution essaie de tirer parti de cette caractéristique récurrente. C'est la différence majeure avec les problèmes d'agencement, pour lesquels les objectifs n'expriment pas forcément un objectif de compacité.

B. Typologies des problèmes de découpe :

Le premier paramètre sert à identifier la dimension du problème. Le second renseigne sur le type d'affectation, à savoir si tous les composants doivent être placés à l'intérieur du ou des contenants ou si seulement une sélection de ces composants doit être positionnée. Dans le premier cas, le problème est une minimisation effectuée sur les contenants (taille ou nombre). Dans le second cas, il s'agit d'une maximisation effectuée sur les composants à positionner. Concrètement, il s'agira de maximiser l'espace d'utilisation ou une fonction profit en sélectionnant un sous-ensemble des composants. Le troisième paramètre fournit des indications sur le type de contenant. Enfin le quatrième paramètre contient des informations sur la nature des composants, à savoir s'ils sont identiques, faiblement différents ou fortement différents.

IV. METHODES PROPOSEES POUR LA RESOLUTION DES PROBLEMES DE DECOUPE :

Les méthodes proposées concernera l'activité de production des pièces de Zellige traditionnel et vue la complexité de leurs formes on peut faire une projction de ces méthodes sur n'importe quel type de découpe (cuir, verre, bois,…).

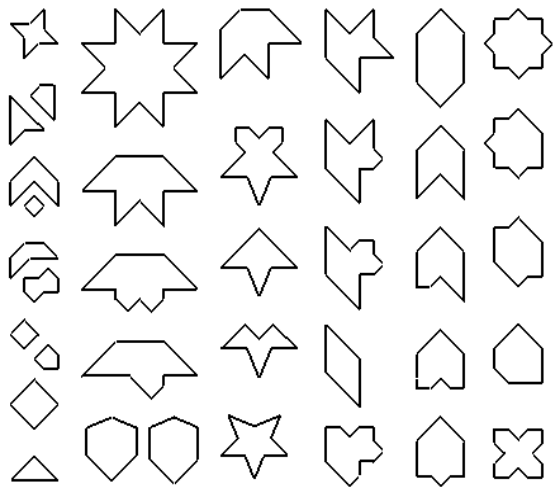


Figure 1 : Formes concernées par la solution proposée.

La méthode proposée est méta-heurestique concerne les formes polygonales. Elle est présentée de façon modulaire. Tout d'abord, le principe général de la méthode est énoncé sous forme d'étapes et de méthodologie et à la fin on propose l'algorithme correspondant.

C. *Méthode méta-heuristique :*

Les problèmes de découpe présentent une grande combinatoire avec de surfaces réduite ou minimales, avec des géométries en général très complexe, même avec des formes variable.

L'espace de recherche est très grand, voire infini lorsque le problème utilise des formes variables. Pour explorer efficacement l'espace de notre recherche, on va supposer que les formes sont prédéfinies (on parle des formes primitives) et un algorithme d'optimisation globale sera proposé à la fin de cette partie. Cet algorithme sera stochastique pour permettre une exploration avec des temps de calcul raisonnable. Il pilotera les variables de positionnement de l'ensemble des composants avec leurs attributs et aura pour objectif de proposer des solutions efficaces. Les variables de position des composants sont choisies continues, les variables d'orientation des composants seront soit discrètes soit continues en fonction des degrés de liberté des composants.

La difficulté majeure associée à la méthode de placement ordinaire réside dans le respect des contraintes de positionnement. Pour que les solutions proposées par l'algorithme d'optimisation globale soient réalisables, un algorithme de séparation est utilisé. Il a pour objectif de déplacer localement les composants afin que ceux-ci respectent au mieux les contraintes de positionnement.

L'utilisation d'algorithmes d'optimisation globale est quelque chose de classique dans la résolution des problèmes de positionnement de forme. De nombreuses méthodes utilisant des algorithmes évolutionnaires ont été proposées [7]. En revanche, le couplage d'un algorithme évolutionnaire et d'une méthode de séparation est quelque chose de nouveau.

Pour résumer, l'algorithme d'optimisation globale est chargé d'explorer efficacement l'espace de conception et l'algorithme de séparation a pour mission de rendre admissibles autant que possible les solutions proposées par l'optimiseur global. Les différents éléments nécessaires à la résolution des problèmes de placement sont présentés dans les paragraphes suivants.

1) *Représentation géométrique et détection de collisions*

En 2D, les composants peuvent être représentés de différentes manières. Le choix d'une représentation géométrique a pour objectif de faciliter les calculs géométriques notamment la détection de collisions. Cette dernière représente l'un des points durs de la résolution des problèmes de placement relaxé. La rapidité des algorithmes de détection de collision est cruciale pour la résolution des problèmes de placement. De leur efficacité dépendent les performances des méthodes de résolution proposées.

Les algorithmes de détection de collisions peuvent être classés selon trois niveaux : le premier consiste uniquement à détecter la collision, le second à quantifier la collision, enfin le troisième niveau consiste à utiliser les informations de collision pour séparer les composants se chevauchant. Plusieurs méthodes ont été mises en place pour quantifier les éventuelles collisions entre composants : la plus évidente, consiste à calculer les aires d'intersection entre chaque paire de composants.

Toutefois, cette méthode est trop coûteuse pour pouvoir être mise en place dans un cadre générique avec déférentes géométries (notamment des polygones). Pour s'affranchir de cette difficulté, des alternatives ont été développées, comme par exemple avec l'utilisation des profondeurs de pénétration que nous présentons ci-après.

Notre objectif est de choisir une représentation géométrique adaptée à chaque problème, dont on pourra tirer partie pour la réalisation de l'algorithme d'optimisation qui évite tout type de collision. Pour notre étude en 2D, ce choix va dépendre des degrés de liberté des composants, notamment de la libre rotation ou non des composants.

2) *Principe de l'algorithme d'optimisation :*

Cet algorithme a pour mission de rendre admissible autant que possible les solutions proposées par l'optimiseur de non-collision. À partir d'une solution non admissible, l'algorithme cherche à minimiser une fonction caractérisant le non-respect des contraintes de placement en déplaçant localement l'ensemble des composants. L'objectif est de déplacer au minimum les composants de manière à perturber le moins possible la solution initiale proposée. La méthode est globale dans le sens où tous les composants sont déplacés en même temps pour minimiser le non-respect des contraintes de placement.

Au niveau analytique, la fonction à minimiser s'écrit comme la somme de deux fonctions : la première traduit le non-respect des contraintes de non-chevauchement entre chaque paire de composants, la seconde exprime le non-respect des contraintes d'appartenance pour chaque composant. Si le problème de placement comporte m composants, alors la fonction à minimiser par l'algorithme d'optimisation s'écrit mathématiquement :

$$F(v) = w_{pen} \sum_{i=1}^{m-1} \sum_{j=j+1}^{m} f_{ij}(v) \quad + \quad w_{app} \sum_{i=1}^{m} g_i(v)$$

où v est le vecteur position des m composants, il contient généralement un vecteur de translation **x** et un vecteur d'orientation **o**. Les coefficients ω_{pen} et ω_{app} sont des paramètres positifs permettant de régler de l'influence des deux fonctions. Les abréviations pen et pro correspondent respectivement aux termes pénétration et appartenance, faisant référence aux contraintes de non-chevauchement et d'appartenance. Si les collisions sont à éviter à 100 % on prend : $w_{pen} = w_{pro} = 1$.

Le terme $f_{ij}(v)$ traduit numériquement le non-respect de la contrainte entre les composants C_i et C_j . Si ces deux composants ne se chevauchent pas, alors cette valeur est nulle. Le terme gi caractérise l'appartenance du composant C_i à l'intérieur du contenant. Si le composant C_i est situé intégralement dans le contenant, alors le terme $g_i(v)$ est nul. L'expression des fonctions f_{ij} et g_i dépend de la géométrie des composants, ainsi que de leurs degrés de libertés caractérise la possibilité de son orientation.

Les fonctions $f_{ij}(v)$ et $g_i(v)$, sont exprimées en fonction de la notion de la profondeur de pénétration, qui est une distance entre un point et un segment, cette notion nous allons la présenter d'une manière explicite dans la section précedente.

3) *Respect des contraintes pour éviter les collisions :*

L'organigramme proposé est présenté sur la figure ci-dessous, est très proche d'un algorithme génétique générationnel classique. La population initiale peut être générée de manière aléatoire, ou le concepteur peut fournir un ensemble de solutions de son choix ou encore des solutions obtenues avec des heuristiques de positionnement.

Avant l'évaluation de chaque solution, les contraintes de positionnement sont vérifiées. Si ces dernières ne sont pas satisfaites, alors l'algorithme modifie la solution de manière à la rendre admissible. La solution sera admissible dès lors qu'on n'a pas de collision ni entre composants elles-mêmes ni entre les composants et le contenant.

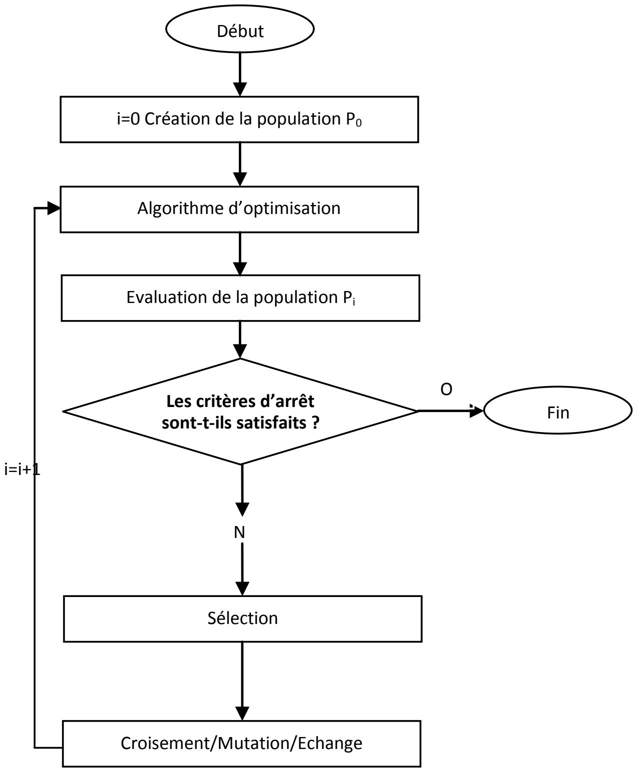


Figure 2 : Organigramme d'optimisation

Cette contrainte sera par la suite prise en compte comme une contrainte de l'algorithme génétique : si on n'a pas de collision, alors les contraintes de placement sont satisfaites, dans le cas contraire, la solution modifiée n'est toujours pas admissible. Ces opérations (choix de composant, orientation : translation, rotation) sont répétées jusqu'à l'évaluation complète de la population. Si l'un des critères d'arrêt est satisfait (pas de collision, il reste plus de composants, il reste plus de place dans le contenant), l'algorithme s'arrête. Sinon, les opérations de sélection par tournoi, de croisement, mutation et échange sont exécutées en vue de produire une nouvelle génération. Le critère d'arrêt peut être l'un de ces derniers critères cités, ou tout autre critère défini par l'utilisateur.

Dans de nombreux problèmes 2D, les orientations des composants sont discrètes, c'est-à-dire choisies parmi un ensemble de valeurs définies par l'utilisateur. Dans la majorité des cas, ce sont des orientations orthogonales (0◦, 90◦, 180◦, 270◦). Ces orientations sont généralement dictées par la géométrie des composants. Toutefois, rien n'empêche, dans la version de l'algorithme présentée ici, de prendre en compte d'autres orientations (tous les angles allant de 0° à 360°). La modélisation des problèmes prenant en compte la libre rotation des composants fait l'objet de la section suivante. Pour l'étude de cas (formes primitives des pièces de Zellige) nous allons prendre l'angle minimal de la forme primitive [9].

Les orientations des composants sont discrètes. Les variables du composant C_i sont contenues dans le vecteur $v_i = (c_i, \theta_i)=(x_i, y_i, \theta_i)$, où c_i sont les coordonnées du point de référence du polygone et θi est à choisir parmi l'ensemble d'orientations discrètes autorisées par le concepteur. Les composants sont pilotés à l'aide d'un point de référence choisi comme leur centre de gravité. La position du polygone P_i est calculée à l'aide d'une composition d'une rotation et d'une translation.

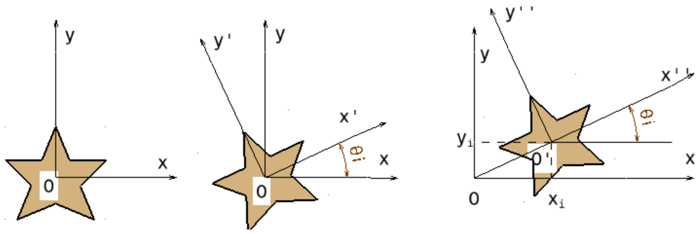


Figure 3 : Orientations possibles (rotation/translation)

Cette position se calcule de la manière suivante :

$$\begin{pmatrix} P^n_{ij,x} \\ P^n_{ij,y} \end{pmatrix} = c_{ij}(v_i) = \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \begin{pmatrix} P_{ij,x} \\ P_{ij,y} \end{pmatrix}$$

Où $P_{ij,x}$ et $P_{ij,y}$ représentent les coordonnées initiales en abscisses et ordonnées du point j du polygone Pi. Les nouvelles coordonnées de ce point sont notées $P_{n\ ij,x}$ et $P_{nij,y}$. La figure 46 illustre les calculs réalisés pour positionner chaque composant i. On applique tout d'abord la rotation θi autour de son centre de gravité, puis le composant est translaté du vecteur x_i.

La figure 47 présente un exemple de polygones, P_1 (bleu) et P_2 (marron) et un contenant P_0 (vert).

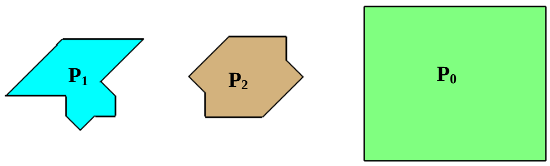


Figure 4 : Polygones P_0, P_1 et P_2

Les polygones P_1 et P_2 initialement ont respectivement une aire de surface S_1 et S_2 bien connues. Le polygone de chevauchement de P1 et P2 en couleur rouge (Figure 5) permet de savoir si P1 et P2 se chevauchent en analysant la différence d'aire de surfaces initiales S_1 et S_2 des polygones P_1 et P2. Si l'aire du polygone de chevauchement de P1 et P2 est non nulle, alors P1 et P2 se chevauchent. Si l'aire du polygone de chevauchement de P1 et P2 est nulle, alors P1 et P2 soit sont en contact ou ne sont pas en collision.

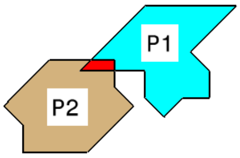


Figure 5 : Polygone de chevauchement (en rouge)

Le même raisonnement peut être appliqué pour tester l'appartenance d'un polygone à un autre polygone (on prend le cas d'un rectangle) : le polygone P2 est contenu dans le rectangle P0, si l'aire de la surface du polygone d'appartenance en contour rouge (Figure 6) est nulle.

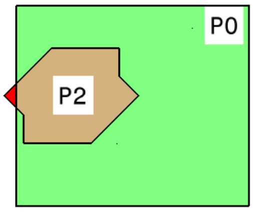


Figure 6 : Polygone d'appartenance (en rouge)

Même si le calcul de l'aire des polygones de chevauchement et d'appartenance est coûteux et doit être effectué pour chaque orientation des polygones, l'utilisation de cette méthode présente deux avantages :

- les collisions peuvent être détectées par un simple test de différence d'aire de surfaces;
- d'autre part, le test de collision permet de récupérer des informations (profondeurs de pénétration) qui seront utiles à l'algorithme de séparation pour évaluer numériquement le chevauchement de deux composants.

Le calcul de l'air de ces polygones est basé sur l'utilisation des sommes de Minkowski [8]. Dans le cas où les polygones sont convexes, leur somme de Minkowski est un polygone convexe, dont la complexité de calcul est linéaire avec le nombre de sommets. Dans le cas où les polygones sont quelconques, une décomposition préliminaire en un assemblage de polygones convexes est nécessaire. L'ensemble des polygones de chevauchement et des polygones d'appartenance doit être calculé par un prétraitement initial.

Avec une orientation discrète des composants, la première partie de l'algorithme d'optimisation (partielle) se base sur ces polygones de chevauchement et appartenance pour détecter la collision entre polygones et établir les directions qui vont permettre d'éviter les collisions entre composants, La solution consistera à translater les composants pour obtenir une solution optimale.

Lorsque l'orientation des composants n'est plus discrète mais continue, la détection de collisions basée sur l'utilisation des polygones de chevauchement et appartenance n'est plus performante. En effet, il faut calculer ces polygones pour chaque nouvelle orientation.

Même si ce calcul des aires de surfaces des polygones de chevauchement et des polygones d'appartenance est immédiat, l'optimisation partielle agie uniquement sur les composants, donc il faut passer à la deuxième partie de l'algorithme d'optimisation qui va nous permettre d'optimiser le placement des composants dans le contenant. Car l'algorithme d'optimisation partielle a peu de chance de proposer par lui-même la bonne combinaison d'orientations pour satisfaire les contraintes de placement.

La première idée qui vient à l'esprit, consiste à générer des solutions réalisables à partir de différentes optimisations et à utiliser ces solutions réalisables comme population initiale. Il faut pour cela exécuter plusieurs fois l'algorithme d'optimisation partielle, puis choisir les premières solutions réalisables identifiées, et relancer la méthode avec une population initiale différente autant de fois que nécessaire pour obtenir une population de solutions réalisables. Cette technique peut être coûteuse pour les problèmes où une solution réalisable est difficile à identifier.

La seconde idée, que nous avons retenue, consiste à utiliser une méthode qui se base toujours sur la notion des aires de polygones. Nous avons donc développé une méthode de placement optimal, qui assure la satisfaction des contraintes de non-chevauchement et d'appartenance tout en assurant l'optimisation du placement des composants. La méthode est une heuristique adaptée aux composants et contenants de géométrie complexe. Cette heuristique assure un placement séquentiel des composants qui garantit le respect des contraintes de placement (pas de collision). Des permutations aléatoires des composants à l'intérieur du contenant vont nous permettre de construire un polygone défini par les sommets des composants et les sommets du contenant. Pour chaque composant à placer, l'aire de surface du polygone construit est calculée.

Le composant concerné va être déplacé (translation et/ou rotation), pour avoir la plus petite aire de surface du polygone construit. Le procédé est répété jusqu'à ce que l'ensemble des composants soit positionné, ou bien qu'il n'y ait plus de place dans le contenant. Pour varier les solutions proposées, les directions de dépôt de l'heuristique peuvent être interverties : par exemple, les composants peuvent être placés en priorité en haut à droite.

La figure 50, montre le placement des composants (sans collision) et avec la plus petite aire de surface en vert Sr du polygone restant construit par les sommets des composants et du contenant plusieurs méthodes permettent la construction de cette polygone, nous allons choisir la méthode qui se base sur l'algorithme de Jarvis's mach [8].

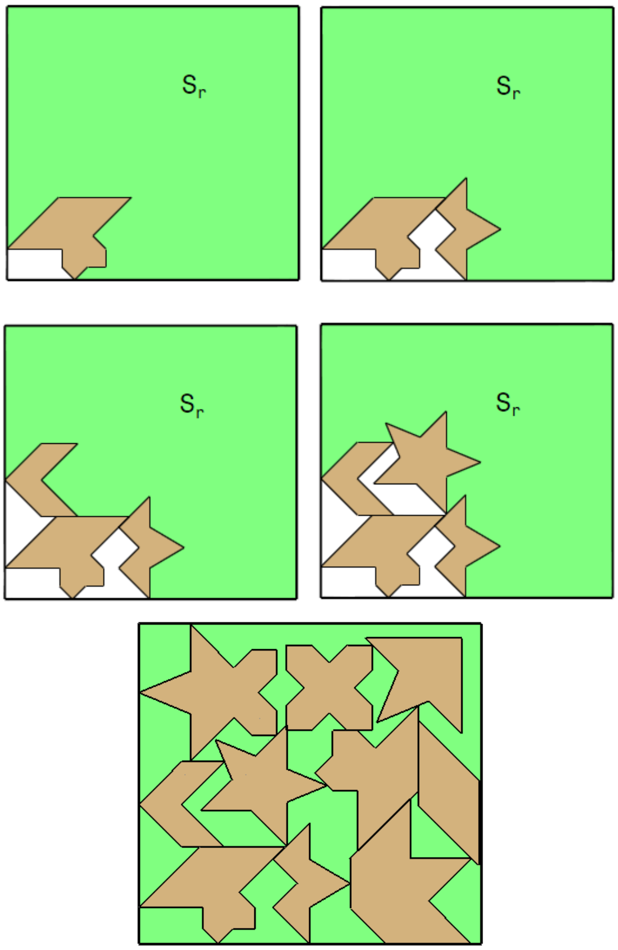


Figure 7 : positionnement des formes dans un contenant suivant notre heuristique.

Enfin, nous présentons l'algorithme d'optimisation qui travaille sur la permutation gérant l'ordre d'introduction des composants peut être utilisé pour les situations où l'initialisation est difficile. Avec notre méthode, tous les composants seront en contact tout en maximisant les composants placés. Elle sera donc bien adaptée aux problèmes à forte compacité, pour lesquels les composants des solutions optimales seront en contact (le cas le plus optimal, qui correspond à la trajectoire de l'outil de coupe). Cependant, les contraintes de placement spécifiques, comme l'alignement de composants, ne sont pas traitées. La figure 7 illustre le fonctionnement de notre heuristique sur une instance de problèmes de découpe de formes irrégulières. Les composants sont ici placés d'abord le plus à gauche, puis le plus en bas.

4) ***Algorithme d'optimisation :***

L'algorithme d'optimisation est utilisé pour faire respecter les contraintes de placement du problème (contraintes de non-chevauchement entre les composants et contraintes d'appartenance des composants au contenant). L'idée générale consiste à regrouper l'ensemble des contraintes de placement dans un seul algorithme.

Ce qui nous intéresse c'est la trajectoire de l'outil de coupe. Le pilotage des géométries se fait à l'aide de leur centre. Les dimensions des contenants sont définies à l'aide d'un vecteur de taille d=2, où chaque composante se reporte à la dimension associée. Un polygone (ou une forme primitive) i est entièrement définie par deux vecteurs : un vecteur de dimension et un vecteur de position.

Il faut chercher à annuler l'aire d'intersection des polygones i et j, et à évaluer l'aire de la partie du rectangle i « intersection » j située à l'intérieur du contenant. Les figures 5 et 6 présentent l'intersection de deux polygones et d'un polygone et un contenant rectangulaire.

Au niveau analytique, l'ensemble des contraintes présentées précédemment nous permet de construire plusieurs fonctions qualifiant le non-respect des contraintes. La fonction objectif à minimiser sera toujours une combinaison de deux fonctions : une traduisant les contraintes de non-chevauchement et une autre les contraintes d'appartenance :

- F1 : Minimisation de la distance entre composants,
- F2 : Minimisation de la distance entre composants et contenant,

L'algorithme d'optimisation présenté dans cette section est adapté aux composants représentés sous forme de polygones à orientations discrètes. C'est une idée inspirée des travaux faits dans le cadre de la résolution d'un problème de découpe de composants de géométrie irrégulière [4].

Lorsque les composants ne sont plus de géométrie simple, il devient difficile et coûteux de calculer l'aire de chevauchement entre chaque paire de composants pour évaluer le non-respect des contraintes de placement. Dans la section précédente, nous avons vu que l'aire des surfaces peut quantifier le chevauchement de deux composants (à côté de la sommes de Minkowski), d'autres moyens existent, notamment la profondeur de pénétration δ. L'algorithme présenté ici base sa fonction de minimisation sur les profondeurs de pénétration, et utilise pour cela les polygones de chevauchement et d'appartenance. Les contraintes de non-chevauchement et d'appartenance sont évaluées dans une seule et même fonction F (x, o), où x et o représentent respectivement la matrice de position des composants et le vecteur d'orientation des composants. L'orientation θi ∈ o de chaque composant est choisie parmi un ensemble discret d'orientations défini par l'utilisateur.

Les contraintes qu'il faut vérifier, sont le respect de la contrainte de non-chevauchement entre les composants i et j, ainsi que le respect de la contrainte d'appartenance entre le composant i et le contenant. Si ces deux contraintes sont satisfaites, alors la découpe est optimale. Pour présenter analytiquement et simplifier le problème, seules les translations des différents composants sont autorisées pour minimiser la fonction objectif. La prise en compte d'orientations discrètes, qui ne sont pas limitées aux orientations orthogonales, se fait au niveau de l'optimisation globale. Les intersections entre composants sont évaluées en utilisant les profondeurs de pénétration entre composants, pour lesquelles les gradients analytiques peuvent être facilement évalués. La profondeur de pénétration δ (Pi, Pj) entre deux polygones Pi et Pj [2] est définie par :

$$\delta\left(P_i, P_j\right) = \max\{\|z\| \mid int\left(P_i\right) \cap \left(P_j \oplus z\right) = \emptyset, \quad z \in R^2\}$$

où int(P_i) désigne l'intérieur du polygone P_i, et ($P_j \oplus z$) représente le polygone P_j translaté du vecteur z. $\delta(P_i, P_j)$ correspond à la distance de translation minimale pour assurer la séparation des polygones P_i et P_j . L'évaluation de la profondeur de pénétration δ (P_i, P_j) se fait à l'aide du polygone de chevauchement de P_i et P_j.

Pour chaque point d'un polygone, le point de la frontière le plus proche est calculé. Les vecteurs formés entre les points pk et leurs points les plus proches fournissent des directions pour éviter les collisions, et les normes de ces vecteurs correspondent aux profondeurs de pénétration δ. Pour les points situés sur l'axe médian du polygone, il n'y a pas unicité de la direction de séparation, on choisit aléatoirement la direction de séparation des polygones parmi les différentes possibilités.

Pour chaque point pk (les sommets d'un ou plusieurs polygones), les distances entre les points sommets et le polygone sont calculées. La plus grande valeur des distances entre chaque point pk et son point le proche sur le deuxième polygone correspond à la profondeur de pénétration entre deux polygones (figure 8).

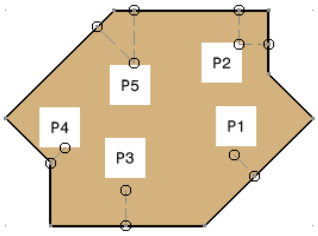


Figure 8 : Calcul des profondeurs de pénétration : contrainte de non-chevauchement.

Les différents points pk représentent les sommets du polygone situés à l'extérieur du contenant (sommets du polygone d'appartenance), pour certains points situés sur l'axe médian, plusieurs points du polygone peuvent donner la même profondeur de pénétration : le point p4 a trois possibilités pour réintégrer le contenant. Un vecteur de translation minimale pour que le polygone « i » appartienne au contenant doit être choisi dès le départ (figure 9).

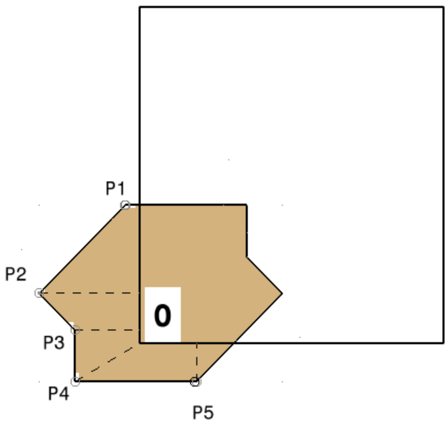


Figure 9 : Calcul des profondeurs de pénétration : contrainte d'appartenance.

Ci-dessous (figure 10) l'algorithme de l'heuristique proposée :

Entrée : Contenant P_0, Nombre de composants m, matrice position $x \in R^{m\times 2}$

Sortie : tableau T des points pi des sommets des formes concernées (polygones)

INITIALISATIONS :

$i \leftarrow 0 \quad j \leftarrow 0$

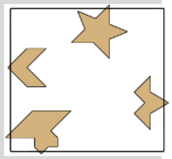
Début

pour $i \leftarrow 1$ **à** m **faire**

« respect de la contrainte d'appartenance »

si $\delta\left(P_i, P_0\right) \neq 0$ **alors**

translater le point p du polygone i de la valeur $z = \delta$



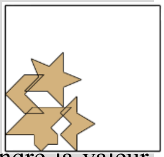
pour $j \leftarrow i$ **à** m **faire**

« respect de la contrainte de non-chevauchement »

si $\delta\left(P_i, P_j\right) \neq 0$ **alors**

translater le point p du polygone j de la valeur $z = \delta$

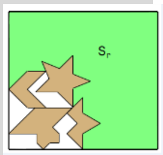
construire Sr et calculer son aire de surface



« optimisation du placement »

orienter le polygone j jusqu'à atteindre la valeur maximale de Sr

fin



fin

$i \leftarrow i + 1$

Fin

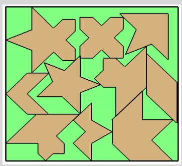


Figure 10 : Algorithme d'optimisation

V. CONCLUSION

La minimisation des chutes proposée, repose principalement sur les algorithmes d'optimisation des positionnements des géométries (problèmes de découpe), en général la plupart des recherches traitent des méthodes sur mesures pour résoudre les différents types de problèmes particuliers rencontrés dans l'industrie. Toutefois, la majorité des méthodes proposées ne permet pas d'intégrer des géométries particulières (convexe, complémentaire, …). Nous avons proposons deux méthodes sous forme d'algorithmes, la première heuristique pour les géomètrie rectangulaire, et la deuxième méta-heuristique pour traiter les géomètries de formes complexes. Le développement de telles méthodes va permettre de traiter la plupart des problèmes en deux dimensions qui va permettre une optimisation des chutes de découpe.

Nous avons proposé une méthode de positionnement pour sa souplesse d'adaptation aux différents problèmes rencontrés (il suffit de construire ses rectangles minimaux). Les contraintes d'optimisation sont directement des contraintes de non chevauchement lors du positionnement des composants. Les raisons qui ont motivé le choix d'une méthode de positionnement c'est parce qu'elle permet l'intégration des géométries dans la modélisation du problème, garantissant ainsi leur respect. Le positionnement de composants, la mise à distance entre composants ou encore le positionnement spécifique des composants peut donner naissance à de nouvelles géométries facilement gérées. Pour la méthode méta-heuristique, dédiée pour les formes complexes (de formes primitives définies), elle permet au concepteur d'interagir facilement avec les solutions et de proposer des solutions qui lui semblent intéressantes.

References

- [1] HARLD D. : A typology of cutting and packing problems. European Journal of Operational Research, 44(2): 145–159, janvier 1990.
- [2] : BURKE K., KENDALL G. et WHITWELL G. : A new placement heuristic for the orthogonal stock-cutting problem. Operations Research, 52(4): 655–671, 2004.
- [3] : David E. Goldberg : Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Professional, janvier 1989.
- [4] : PIERRE M., JOHN R. : Bi-Objective optimization of components packing using a genetic algorithm. Dans NASA/AIAA/ISSMO Multidisciplinary Design and Optimization Conference, pages 352–362, Seattle, Washington, septembre 1996.
- [5] : JACQUENOT G. Méthode générique pour l'optimisation d'agencement géométrique et fonctionnel. Ecole Centrale de Nantes-France, 2010.
- [6] : COEURJOLLY D. Algorithmique pour l'analyse et la modélisation en géométrie discrète. Université Claude Bérnard, Lyon 1-France, 2007.
- [7] : WATTEBLED P. Résolution heuristique d'un problème de conception de modules automatiques de rangement , Université de Lille 2010.
- [8] : LABURTHE F. Contraintes et Algorithmes en Optimisation Combinatoire, Université Paris VII, 1998.
- [9] : CASTERA J. M. « ARABESQUES, art décoratif au Maroc ». ACR Edition 96.