

Gestion des entrées

1) Formulaire de saisie

Maquette - Optimiseur de coupe

Veillez remplir les informations suivantes :

Barres avant la découpe :

Dimension	Unité	Quantité
0,00	m	1

Tronçons désirés :

Dimension	Unité	Quantité
0,00	m	1
0,00	m	1

Epaisseur de lame

Epaisseur	Unité
0,00	mm

Résultat :

Veillez entrer des données ...

Maquette - Optimiseur de coupe

Veillez remplir les informations suivantes :

Barres avant la découpe :

Dimension	Unité	Quantité
0,00	m	1

Tronçons désirés :

Dimension	Unité	Quantité
0,00	m	1
0,00	m	1

Epaisseur de lame

Epaisseur	Unité
0,00	mm

Résultat :

Veillez entrer des données ...

Les formulaires de saisie se situent dans la fenêtre principale.
A l'heure actuelle aucune de ces deux fenêtres n'a été privilégiée.
Elles sont néanmoins strictement similaires quant à la gestion des saisies.

2) Détail des champs de saisie

a) Les champs « dimension »

On ne doit pouvoir saisir dans ces champs que des nombres flottants strictement positifs. La plus grande unité de saisie est le mètre, mais le programme travaille sur des nombres qui seront dans tous les cas convertis en mm. Pour éviter les débordements de capacité du type « float » (qui a dans le pire des cas un maximum égal à $3.4e+38$), on pourrait par précaution inhiber la saisie d'un nombre supérieur à $3.4e+38 / 1000 = 3.4e+35$. De plus, la fonction `float ListPaires::calculeRendement (float longueurBarres)` est susceptible de multiplier une dimension par 100.

$$\rightarrow 0 < \text{dimension} \leq 3.4e+33$$

Ce qui correspond à peu près à 100000 milliards d'années lumière... Est-il vraiment utile de borner cette saisie vers les valeurs hautes ?

$$\rightarrow 0 < \text{dimension}$$

b) Les champs « quantité »

On ne doit pouvoir saisir dans ces champs que des nombres entiers strictement positifs. Pour éviter les débordements de type « int » (qui a dans le pire des cas un maximum égal à 2147483647), il faudrait par précaution inhiber la saisie d'un nombre supérieur à 2147483647, ainsi que la somme des saisies à cette même valeur.

→ $0 < \text{quantité} \leq 2147483647$

Ce qui correspond à peu près à deux milliards, il semble là encore inutile de borner cette saisie vers les valeurs hautes.

→ $0 < \text{quantité}$

c) Le champ « épaisseur de lame »

On ne doit pouvoir saisir dans ces champs que des nombres flottants positifs. Par contre une valeur égale à 0 est acceptable car la découpe peut-être sectionnante. La plus grande unité de saisie est l'inche, mais le programme travaille sur des nombres qui seront dans tous les cas convertis en mm. Là encore il n'est pour l'instant pas nécessaire de borner les valeurs hautes des saisies.

→ $0 \leq \text{épaisseur de lame}$

d) Les champs « unité »

Pour les champs quantité, ils ne peuvent prendre que les valeurs suivantes : mm, cm, m, inches, feet, yards. Ceci est imposé par une liste déroulante. Pas de précaution particulière à prendre.

3) Types des champs de saisie

On pourrait utiliser exclusivement des listes déroulantes (très user friendly) pour les champs « quantité », « dimension » et « épaisseur de lame ». Mais la liste déroulante plafonne à une quantité ou une dimension max, on ne peut pas dérouler une liste infinie, mais une liste plausible, or qui sommes-nous pour définir ce qui est plausible et ce qui ne l'est pas ? Selon quels critères ? Il existe autant de situations plausibles que d'utilisateurs.

L'autre solution reviendrait à utiliser un champ de saisie. Mais on perd en ergonomie.

L'idéal serait une solution hybride, une liste déroulante dans un champ éditable, avec validation de la saisie, on n'aurait plus alors de plafond, et on pourrait choisir dans une liste déroulante pour les cas courants

Concernant les unités une liste déroulante convient parfaitement.

4) Interactions entre les saisies

Certaines saisies peuvent se révéler incompatibles entre elles, entraînant l'impossibilité pour le programme de fournir des résultats, voire son plantage...

a) Interactions entre les « dimensions »

Une inadéquation entre les saisies de dimensions concernant les barres avant découpe et les tronçons désirés vient tout de suite à l'esprit : lorsque les tronçons demandés sont d'une taille supérieure aux barres avant découpe. Il devient impossible dès lors de fournir ces tronçons. Il conviendrait donc qu'une fonction se charge de vérifier grossièrement la faisabilité des découpes demandées avant le lancement du moteur d'optimisation, par vérification de la quantité de barres disponibles permettant d'accéder à la demande relative aux tronçons de plus grande taille. Ceci ne garantit pas encore la faisabilité (car au final c'est l'optimisation elle seule qui pourra prouver la faisabilité).

b) Interactions entre la longueur totale des barres et la longueur totale des tronçons

Une autre interaction problématique évidente : le manque de matière première. Encore une fois, une fonction peut grossièrement vérifier la faisabilité de la requête en comparant les longueurs cumulées des barres en entrée et les longueurs cumulées des tronçons en sortie. Là encore seule la non-faisabilité est démontrée.

c) Interactions entre les « dimensions » et l'« épaisseur de la lame »

Voilà qui peut sembler ridicule au possible, mais que se passe-t-il lorsque l'épaisseur de lame est plus épaisse que la plus petite des barres saisies en entrée ? Il s'agirait là de la détection d'une erreur de saisie très grossière. Faut-il la faire ? A priori oui, car l'utilisateur a pu tout simplement rentrer une valeur trop petite pour une barre en entrée, mal choisir une unité, ...

5) Détection des saisies aberrantes

Que faire pour les saisies qui semblent douteuses ? On peut :

- Tout autoriser : quelle que soit la saisie, les calculs d'optimisation sont lancés
- Tout interdire : en cas de saisie aberrante, l'utilisateur est appelé à modifier cette saisie
- Prévenir : une fenêtre apparaît, avertissant que la valeur saisie semble aberrante et proposant au choix de continuer ou de recommencer la saisie (longueur trop courte ou trop longue, quantité trop élevée, épaisseur de lame trop élevée), et avertissant le cas échéant d'un impact possible sur les temps de traitement (en cas de grosse quantité).

Le mieux serait probablement de prévenir et de laisser le choix à l'utilisateur .

La vraie difficulté est alors de distinguer ce qui est aberrant de ce qui ne l'est pas.

En fait un petit fichier de configuration indépendant pourrait-être la solution, ce fichier contiendrait :

- Les seuils min et max pour les longueurs des barres
- Même chose pour les longueurs des tronçons
- Le seuil max pour les quantités de barres
- Même chose pour les quantités des tronçons
- Un seuil max pour l'épaisseur de lame

Ces valeurs pourraient être définies par défaut et modifiées par l'utilisateur à l'aide d'un formulaire.

6) Synthèse

Au final deux types d'erreur peuvent conduire au plantage du programme :

- Le non respect des valeurs possibles pour les saisies unitaires
 - Quantités : entiers strictement positifs
 - Dimensions : flottants strictement positifs
 - Epaisseur de lame : flottant positifs
- Une surcharge mémoire due à la saisie de quantités trop importantes

Les autres « erreurs », en admettant que l'on laisse se faire l'optimisation des découpes malgré leur détection, ne conduiraient pas au plantage du programme, mais à une réponse incomplète, où certains tronçons ne pourraient pas être découpés (parce que leur taille est trop élevée où parce que la quantité de matière première est insuffisante). Elles pourraient être traitées de la même manière que les saisies aberrantes, et laisser le choix à l'utilisateur de continuer ou non.

Dans ce scénario, une fois vérifiée l'absence d'erreur incapacitante, une fonction pourrait détecter les autres erreurs possibles, les afficher à l'écran et laisser à l'utilisateur le choix de continuer ou de recommencer la saisie. Cette fonction aurait donc à charge de vérifier :

- Qu'il y ait à priori suffisamment de matière première par rapport aux tronçons demandés :

➔ **longueur totale des barres \geq longueur totale des tronçons**

- Que les tronçons de taille maximale soient plus petits que les barres de plus grande taille, en tenant compte de la quantité des barres et des tronçons

➔ **longueur max des barres \geq longueur max des tronçons**

➔ **quantité des barres les plus longues \geq quantité des tronçons les plus longs**

- Qu'il n'y ait pas de saisie aberrante :

➔ **seuil dimension min $<$ dimension barre $<$ seuil dimension max**

➔ **seuil dimension min $<$ dimension tronçon $<$ seuil dimension max**

➔ **quantité barres $<$ seuil quantité max**

➔ **quantité tronçons $<$ seuil quantité max**

➔ **épaisseur de lame $<$ seuil épaisseur de lame max**

7) Fichier de configuration

Il doit contenir les paramètres suivants :

- Seuil dimension min : 10 mm pourrait être une valeur min plausible (parmi d'autres)
- Seuil dimension max : 2500 mm pourrait être une valeur max plausible (parmi d'autres)
- Seuil quantité max : 100 pourrait être une valeur max plausible (parmi d'autres)
- Seuil épaisseur de lame max : le standard de l'épaisseur de lame est 3 mm, donc 4 mm est un bon candidat.

Lors des modifications de ces valeurs, les vérifications suivantes devront être faites :

- Seuil de dimension min, seuil de dimension max et seuil épaisseur de lame max doivent être des flottants strictement positifs :

- ➔ **0 < seuil dimension min**
- ➔ **0 < seuil dimension maximale**
- ➔ **0 < seuil épaisseur de lame max**

- Seuil de dimension max doit être supérieur à seuil de dimension min :

- ➔ **seuil dimension min < seuil dimension max**

- Seuil de quantité max doit être un entier strictement positif :

- ➔ **0 < seuil quantité max**