Optimisation de machine outil

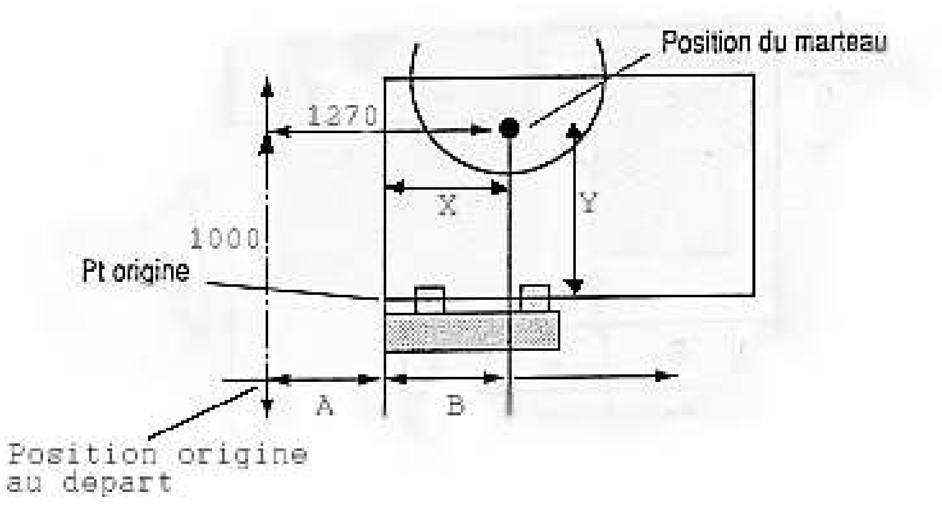
Laurent Viennot

`http://gyroweb.inria.fr/~viennot/enseignement/projets/poinconneuse/index.html`

Une poinçonneuse numérique permet de percer des trous dans une plaque de tôle automatiquement : une fois la tôle placée dans la machine, celle-ci exécute un programme qui lui indique comment manipuler la tôle et où la percer. Le but du sujet est d'optimiser le programme pour que la pièce soit effectuée le plus rapidement possible. En effet, l'ordre dans lequel les trous sont fait ne change rien à la pièce finale obtenue. Par contre, si on minimise les déplacements de la machine, on pourra diminuer le temps de production.

Ce sujet est tiré du fonctionnement du modèle ARIES 2.4.5 de la société AMADA. La première partie du sujet concerne l'optimisation de programmes composés d'un sous-ensemble restreint d'instructions du langage reconnu par la machine. La deuxième partie consiste à émuler la machine pour visualiser la pièce fabriquée par un programme ainsi que l'ordre d'exécution du programme. En extension, on pourra prendre en compte un ensemble plus grand d'instructions du langage.

1 Description de la machine



Une tourelle contient vingt outils de poinçonnage (avec des formes différentes). Une presse très puissante (le marteau) peut donner plusieurs centaines de coups à la minute (avec le même outil). Un changement d'outil prend environ trois secondes.

La tôle est tenu par deux pinces sur une table glissante. La machine peut déplacer la tôle en x ou en y pour poinçonner à l'endroit désiré. En vitesse maximale, la tôle peut être déplacée à 40m/min.

Pour produire une pièce, on cale une tôle dans une position précise de la machine. Cette position de départ permet de caler le repère absolu (qu'on nommera *repère de la tôle*) dans lequel le point $x = 0, y = 0$ est le coin inférieur gauche de la tôle. (x, y) désignera typiquement la position du marteau dans ce repère. Les tôles ont une taille standard de 2000mmx1000mm. On lance ensuite la machine sur un programme. Voici un exemple :

```
G92X1270.Y1000.;
G93X20.Y880.;
G90X75.Y50.T6;
Y205.;
X75.;
Y50.;
G50;
```

- Il y a une instruction par ligne et chaque ligne se termine par `;`.
- Les `G...` indiquent des numéros d'instruction.
- Les `X...` et `Y...` indiquent des positions, les nombres qui suivent sont lus en millimètres.
- `G92X1270.Y1000.;` indique la position de l'outil dans la position de départ. Les chiffres ne dépendent que de la machine, ils seront toujours les mêmes.
- `G93X20.Y880.;` indique de changer de repère. Les coordonnées seront maintenant exprimées dans le repère d'origine $x = 20, y = 880$. Les coordonnées de la nouvelle origine sont données dans le repère absolu de la tôle.
- `G90X75.Y50.T6;` est une instruction de poinçonnage : elle indique de frapper un coup en position $x = 75, y = 50$ avec l'outil numéro 6 ($t = 6$, soit tourelle en position 6). L'outil est centré sur la position donnée avant de frapper.
- `Y205.;` ne possède pas de code instruction. Cela indique de poinçonner en modifiant les paramètres donnés par rapport à la position d'avant. Il s'agit donc d'une abréviation de `G90X75.Y205.T6;`. Les trois avant-dernières lignes indiquent donc d'effectuer trois trous suppémentaires avec le même outil.
- `G50;` indique la fin du programme, la machine revient dans la position de départ.

On supposera pour commencer qu'un programme commence toujours par une instruction `G92` puis est constitué d'une suite d'instructions `G90` ou éventuellement `G93` et se termine par `G50`. La forme abrégée des `G90` peut-être utilisée.

2 Optimisation de la course de la machine

Le temps pris par la machine pour poinçonner un trou dépend principalement de la distance à parcourir jusqu'à la position où poinçonner, sauf quand il faut changer d'outil auquel cas le temps de changement d'outil sera prépondérant. (Voir le paragraphe 4 pour plus de précisions.)

Si on garde le même outil tout le long du programme, on reconnaît le problème du voyageur de commerce : il s'agit de faire un tour qui passe par toutes les positions où il faut faire un trou en minimisant le trajet parcouru. Ce problème est réputé difficile (on ne connaît pas d'algorithme polynomial pour le résoudre.)

On proposera tout d'abord une solution exhaustive pour les programmes de petite taille. Pour les programmes plus longs, on proposera des heurisitiques.

L'inventivité et la qualité des heuristiques proposées seront grandement appréciées dans l'évaluation du projet.

3 Émulation de la machine

On se propose de plus d'émuler l'exécution d'un programme par la machine, cela pour deux raisons. La première est de comparer les heuristiques proposées entre elles, et avec l'optimal pour de petits exemples.

La seconde est de visualiser graphiquement les poinçonnages effectués par la machine pour visualiser la pièce produite. On visualisera de plus le trajet suivi par la machine ainsi que le temps d'exécution de la pièce.

4 Paramètres de la machine

Le constructeur annonçant les performances suivantes :

- une vitesse maximale en x ou en y de 40m/min (il y a un moteur pour les déplacements en x et un autre pour les déplacements en y),
- une cadence de poinçonnage de 180coups/min pour un déplacement maximal de 25mm entre deux coups,
- une vitesse de rotation de la tourelle de 20tours/min,

on supposera que les mouvements de la machine sont régis par les lois suivantes :

- le temps nécessaire pour déplacer la tôle de (dx, dy) vaut $\max(t_x, t_y)$ où t_x est le temps du déplacement dx en x et t_y est le temps du déplacement dy en y,
- le mouvement selon x ou y s'effectue avec une accélération croissant linéairement avec le temps $a = \alpha t$ (avec $\alpha = 14\ m/s^3$) jusqu'à atteindre la vitesse maximale $v_m = 40/60\ m/s$ (pour $t < t_a$, le déplacement effectué après un temps t est de $x_1(t) = \alpha t^3/6$ où t_a est définit par $\alpha t_a^2/2 = v_m$; pour $t > t_a$, il est de $x_2(t) = x_1(t_a) + v_m(t - t_a)$),
- le temps pour frapper un coup est de $t_c = 0.02\ s$,
- le temps pour changer d'un outil en position i sur la tourelle à un outil en position $j \neq i$ est $t_o + \min(|i-j|, |j-i|)/n_o * v_o$ avec $t_o = 2.5\ s$ (temps pour débloquer l'outil précédent et pour bloquer le suivant), $n_o = 20$ (nombre d'outils) et $v_o = 20/60$ tours$/s$ ($\min(|i-j|, |j-i|)/n_o$ est la fraction de tour à parcourir en tournant la tourelle dans le sens le plus approprié),
- tous les mouvements de la machine peuvent s'effectuer simultanément.

La machine possède les outils suivants (avec position dans la tourelle) :

`T1`	rectangle horizontal 6mm x 80mm
`T12`	rectangle vertical 80mm x 6mm
`T17`	rectangle horizontal 6mm x 20mm
`T7`	carré 5mm x 5mm
`T8`	carré 10mm x 10mm
`T10`	carré 20mm x 20mm
`T2`	disque de diamètre 3.5mm
`T3`	disque de diamètre 4.5mm
`T4`	disque de diamètre 5.5mm
`T5`	disque de diamètre 6.5mm
`T6`	disque de diamètre 10mm
`T9`	disque de diamètre 22.3mm

5 Page web de suivi et d'exemples

On trouvera sur `http://gyroweb.inria.fr/~viennot/enseignement/projets/poinconneuse/index.html` des exemples de programmes à optimiser ainsi qu'un suivi du sujet (informations complémentaires éventuelles). Il sera donc important de consulter régulièrement cette page web.

6 Extension : plus d'instructions

En extension, on étendra le sous-ensemble du langage de la machine qu'on sait optimiser. Voici quelques constructions que vous pouvez rajouter.

6.1 Mémorisation de bloc

Il est possible de mémoriser un bloc d'instructions par la syntaxe :

```
U1;
...
V1;
```

Un bloc d'instructions entre `Ui` et `Vi` est mémorisé et peut-être rappelé par `Wi`. Pour $i < 60$, le bloc est exécuté en même temps qu'il est mémorisé. Voici un exemple qui reproduit la même figure de deux trous à trois endroits :

```
U1;
X34.08Y191.9T06;
X102.5;
V1;
G93X0.Y749.;
W1;
G93X0.Y497.;
W1;
```

On tentera de tirer parti de cette construction en optimisant au mieux tout ou partie d'un bloc mémorisé.

6.2 Reprise de pinces

La course de la machine n'est que de 1270mm en x et 1000mm en y ne lui permettant d'atteindre que les points (x, y) avec $0 \leq x \leq 1270$ et $0 \leq y \leq 1000$ dans le repère de la tôle. Pour atteindre toute la surface d'une tôle, il faut faire une reprise de pinces.

En effet, la tôle est tenue par deux pinces le long de l'axe des x. La machine peut bloquer la tôle par un autre système (deux vérins proches du marteau de poinçonnage peuvent venir appuyer sur la tôle), ouvrir les pinces, effectuer un déplacement en x et refermer les pinces avant de débloquer la tôle.

Une instruction `G27X...;` permet d'effectuer cette opération, le déplacement relatif en x des pinces est donné en paramètre. Il faut parfois effectuer un déplacement sans poinçonnage pour s'assurer que le déplacement relatif sera bien possible. L'instruction `G70X...Y...T...;` indique de faire un déplacement pour amener l'outil donné en position donné (comme un `G90` sans frapper). (L'indication d'outil peut-être omise.)

Dans la figure du paragraphe 1, A désigne la course utilisée avant repositionnement et B la course disponible pour le repositionnement.

Une séquence typique de reprise de pinces sera :

```
G92X1270.Y1000.;
G90X0.Y1000.T2;
G70X900.Y1000.;
G27X900.;
G90X2000.Y1000.;
G70X900.Y1000.;
G27X-900.Y1000.;
G50;
```

Ce programme effectue un trou en $(O, 1000)$ et un autre en $(2000, 1000)$. En positionnant l'outil en $x = 900$, on sûr de pour pouvoir déplacer les pinces de 900 car cela correspond au mouvement qui amènerait l'outil en $x = 0$. Pour revenir, une méthode simple consiste à replacer la tôle dans la même position qu'après la première reprise de pinces et à effectuer la reprise d'un déplacement inverse.

On supposera que le temps de reprise de pinces est égal à $t_r = 2\ s$ (le temps pour bloquer et débloquer la tôle) plus le temps du déplacement des pinces qui a déjà été spécifié au paragraphe 4 (il s'agit d'un déplacement en x).

On peut opitimiser à deux niveaux : d'une part effectuer des `G70` offrant le déplacement minimal pour avoir la course suffisante. D'autre part, il faut séparer les trous à faire en deux groupes : ceux qui sont faits avant la reprise de pinces et ceux qui sont faits après. La course de la machine étant supérieure à la moitié de la dimension en x de la tôle, il y a lieu de séparer au mieux.