



HAL
open science

Les systèmes cellulaires de production : l'agencement inter-cellules

Abdelghani Souilah

► **To cite this version:**

Abdelghani Souilah. Les systèmes cellulaires de production : l'agencement inter-cellules. Sciences de l'ingénieur [physics]. Université Paul Verlaine - Metz, 1994. Français. NNT : 1994METZ002S . tel-01776032

HAL Id: tel-01776032

<https://hal.univ-lorraine.fr/tel-01776032v1>

Submitted on 24 Apr 2018

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

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



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

THESE

présentée à

**L'U.F.R Mathématiques, Informatique, Mécanique et
Automatique
DE L'UNIVERSITE DE METZ**

pour l'obtention du titre de

DOCTEUR DE L'UNIVERSITE DE METZ
Option : Sciences de l'ingénieur

par
Abdelghani SOUILAH

<p>LES SYSTEMES CELLULAIRES DE PRODUCTION : L'AGENCEMENT INTER-CELLULES</p>
--

Soutenue le 07.01.1994, devant la commission d'examen composée de:

Mr. Pierre BAPTISTE	Professeur à l'université de Besançon	Rapporteur
Mme. Martine LUMBRERAS	Professeur à l'université de Metz	Examinatrice
MM. Bernard MUTEL	Professeur et directeur du LRPS ENSAIS (Strasbourg)	Rapporteur
Jean-Marie PROTH	Directeur de recherche à l'INRIA	Examineur
François VERNADAT	Chargé de recherche à l'INRIA	Examineur

Metz 1994

vb 79400 .

Dra

S/M3
94/2

AVANT-PROPOS

BIBLIOTHEQUE UNIVERSITAIRE - METZ	
N° inv.	19940058
Cote	S/M3 94/2
Loc	Mopasin

Le travail présenté dans ce mémoire a été effectué au sein de l'équipe du projet SAGEP de l'INRIA-Lorraine. Cette équipe est animée par Monsieur Jean-Marie PROTH, directeur de recherche, auquel je tiens à exprimer ma reconnaissance pour son accueil, sa disponibilité, ses conseils et son soutien tout au long de la préparation de cette thèse.

Que Mademoiselle Marie-Claude PORTMANN, Messieurs François VERNADAT, Chengbin CHU, Xiao-lan XIE, Thomas HAMANN, Said LAFTIT et Vanio SAVI soient remerciés pour leurs conseils et leurs suggestions qui ont constitué une aide précieuse lors de la réalisation de ce travail.

Je désire témoigner mes remerciements et ma reconnaissance aux membres du jury qui m'ont fait l'honneur de participer à la soutenance de cette thèse, et en particulier :

- à Madame Martine LUMBRERAS, Professeur à l'université de Metz, qui m'a fait l'honneur de présider ce jury,
- à Monsieur Pierre BAPTISTE, Professeur à l'université de Besançon,
- à Monsieur Bernard MUTEL, Professeur et directeur du LRPS/ENSAIS (Strasbourg).

Je désire également remercier tous les membres du projet SAGEP pour leurs témoignages d'amitié et leurs soutiens.

Contenu

INTRODUCTION

1) La fonction production	5
2) Les systèmes discrets de fabrication et leur agencement	5
2.1) Les systèmes à agencement orienté produits	6
2.2) Les systèmes à agencement orienté opérations	7
2.3) Les systèmes cellulaires de fabrication	7
3) Organisation de la thèse	8

PARTIE A

1) Etat-de-l'art	11
1.1) Domaines d'application et représentations géométriques associées	11
1.2) Formulations mathématiques et méthodes de résolution existantes	15
2) Problème d'agencement des systèmes cellulaires de fabrication	17
2.1) Introduction	17
2.2) Module de pré-traitement des données	19
2.2.1) Discontinuité des flux	20
2.2.2) Les produits multi-gammes	20
2.3) Module de conception des cellules	21
2.3.1) Position du problème	21
2.3.2) Méthode de résolution	23
2.3.3) Problème des machines multiples	25
2.4) Module d'agencement intra-cellulaire	26
2.4.1) Processus général de résolution	26
2.4.2) Position du problème d'affectation	29
2.4.3) Solution du problème d'affectation	31
3) Conclusion de la partie A	35

PARTIE B

1) Introduction	37
2) Agencement inter-cellulaire	38
2.1) Composantes du problème	38
2.2) Position du problème	39
2.2.1) Fonction objectif	39

2.2.2) Contraintes	40
2.3) Solution proposée	42
2.3.1) Représentation géométrique du système	42
2.3.2) Processus de résolution du problème	46
2.4) Calcul des plus courts chemins	57
2.4.1) Position du problème	58
2.4.2) Etat de l'art	59
2.4.3) Algorithmes proposés	59
2.4.3.1) Algorithme 1	60
2.4.3.2) Algorithme 2	62
2.4.3.3) Evaluation	69
2.5) Conclusion	73
3) Introduction à l'agencement dynamique des systèmes de fabrication ..	73
3.1) Introduction	73
3.2) Etat-de-l'art	75
3.3) Agencement dynamique mono-périodique	78
3.3.1) Position du problème	79
3.3.2) Résolution du problème	80
3.3.3) Conclusion	80
3.4) Agencement dynamique multi-périodique	81
3.4.1) Position du problème et méthode de résolution	81
3.4.2) Conclusion et remarques	84
3.5) Conclusion	85
4) Conclusion de la partie B	85

PARTIE C

1) Introduction	87
2) Fonctionnement du logiciel	88
2.1) Matrice des flux inter-cellulaires	89
2.2) Contraintes	90
2.3) Configuration initiale	90
2.4) Paramètres de contrôle du recuit simulé	96
2.5) Intervention de l'utilisateur	97
3) Architecture du logiciel	100
3.1) Arbre des appels	100
3.2) Description des sous-programmes	100
4) Résultats	107
4.1) Résultats intermédiaires	107

4.2) Résultat final	108
5) Exemples d'application	108
5.1) Exemple détaillé	108
5.1.1) Description des données	108
5.1.2) Agencement	113
5.2) Série d'exemples	115
6) Conclusion de la partie C	115
 CONCLUSION GENERALE	 119
 ANNEXES	
A. Méthode du recuit simulé	121
B. Méthode d'exploration par séparation et évaluation (Branch and Bound)	129
C. Délimitation des zones libres de l'atelier	133
D. Délimitation des entourages libres des cellules	140
E. Exemples numériques	142
E.1. Formation des cellules	142
E.2. Agencement intra-cellulaire	142
E.3. Agencement inter-cellulaire	144
 BIBLIOGRAPHIE	 157

INTRODUCTION

1. La fonction production
2. Les systèmes discrets de fabrication
et leur agencement
3. Organisation de la thèse

1. La fonction production

La production est une activité qui met en jeu de manière interactive des opérateurs, des moyens et de la matière, dans le but de faire évoluer cette dernière vers un état d'achèvement donné. Cette évolution se déroule durant une certaine période. L'accumulation des coûts opérationnels élémentaires engendre le coût de fabrication du produit.

La fonction production englobe principalement les activités suivantes :

1. La transformation de la matière par usinage, par déformation ou par assemblage.
2. La manutention (ou le transport) qui assure le déplacement de la matière sur ou entre les ressources.
3. La gestion de la qualité, qui permet d'assurer la conformité du produit fini à l'état d'achèvement demandé.
4. Le stockage, nécessaire dans toutes les étapes de la production, avant, pendant et après celles-ci.

Les opérations de transformation, avec celles liées au contrôle de qualité, ajoutent une valeur au produit. Les autres ne font qu'augmenter le coût de la production. Un des facteurs de diminution du coût de fonctionnement d'un système de fabrication est une bonne disposition des ressources dans l'espace du système (fonction agencement). C'est à cette activité que nous consacrons cette thèse.

2. Les systèmes discrets de fabrication et leur agencement

Dans ce qui suit, nous nous intéressons uniquement aux systèmes discrets de fabrication.

Les systèmes de fabrication ainsi que la manière dont ils sont agencés dépendent en premier lieu de la nature des produits et du type de fabrication associé. Ils se ramènent à deux types de base : les systèmes linéaires et les systèmes non linéaires.

Introduction

a. Les systèmes linéaires sont dédiés à la fabrication de masse d'un nombre relativement petit de types de produits ayant des gammes de fabrication semblables. Pour ce genre de systèmes, on utilise souvent un agencement orienté produits.

b. Les systèmes non linéaires sont dédiés à la fabrication de quantités d'importance moyenne relatives à une grande variété de types de produits ayant des gammes différenciées. Pour ce genre de systèmes, il est préférable d'utiliser un agencement orienté opérations.

Il est rare de se trouver exactement dans l'une de ces catégories. Les deux modes cohabitent. Ayant constaté cette cohabitation, des chercheurs ont eu l'idée d'exploiter les avantages de l'un et de l'autre de ces types de systèmes pour la mise au point d'un troisième : les systèmes cellulaires de fabrication.

Rappelons d'abord les deux systèmes de base.

2.1. Les systèmes à agencement orienté produits

Dans ce type de système l'agencement des ressources est fait en fonction des exigences du produit. C'est-à-dire que les ressources sont groupées de sorte que chaque groupe de ressources fabrique entièrement un ou plusieurs types de produits utilisant les mêmes ressources. Ceci est illustré par la figure I.1.

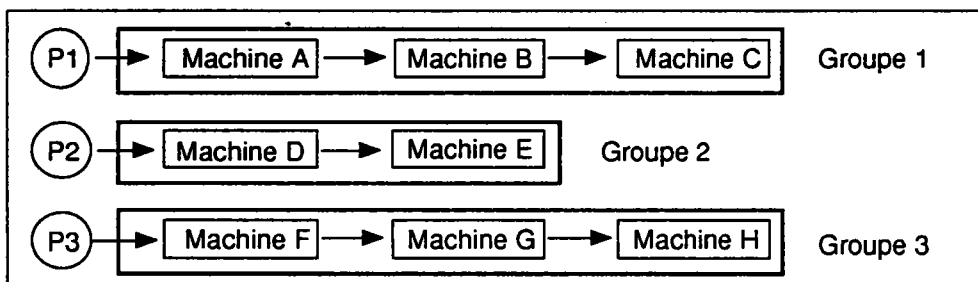


Figure I.1 : Système de fabrication orienté produits

Avantages

1. Contrôle aisé,
2. Temps de fabrication réduits,
3. Stockage réduit,
4. Possibilités d'utiliser une main d'œuvre non qualifiée

Inconvénients

1. Investissement important (duplication de machines),
2. La panne d'une machine bloque la production,
3. Le système est rigide.

2.2. Les systèmes à agencement orienté opérations

Dans ce type de systèmes, les machines sont groupées en fonction des opérations qu'elles réalisent. Ainsi, les machines de tournage sont réunies en section, de même que les machines de fraisage, de perçage etc... . Pour subir les différents types d'opérations de leur gamme de fabrication, les produits passent d'un groupe à un autre. Ceci est illustré par la figure I.2.

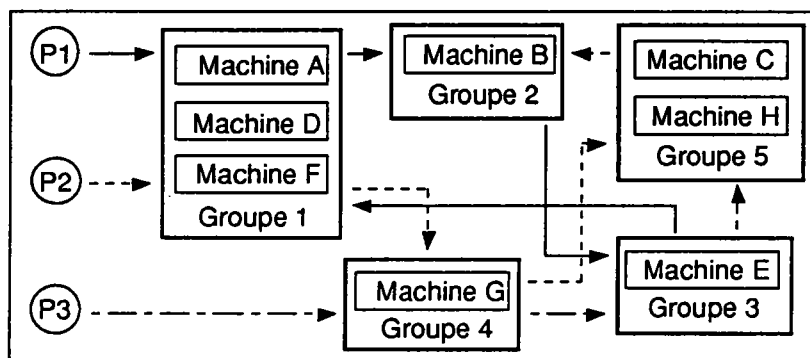


Figure I.2 : Système de fabrication orienté opérations

Avantages

1. Extrêmement flexible,
2. Les pannes des machines ne bloquent pas toute la production,

Inconvénients

1. Nécessité d'un équilibrage de la production (ordonnancement),
2. Nécessité des stocks tampon entre les machines, d'où un plus grand investissement en matières premières et en en-cours.

Le troisième type de systèmes de fabrication, né du mixage de ces deux types, a la rapidité des systèmes orientés produits et la flexibilité des systèmes orientés opérations.

2.3. Les systèmes cellulaires de fabrication

Le professeur Mitrofanov de l'université de Leningrad a introduit en 1946 le concept de "Technologie de groupe" (TG) afin d'établir le lien entre les formes géométriques des produits et les méthodes de fabrication utilisées pour les réaliser. Depuis cette époque, de grands efforts ont été déployés dans ce domaine.

Introduction

Une des applications les plus remarquables de ce concept a été proposée par le professeur J.L. Burbidge de l'institut de technologie de Cranfield ; il s'agit des Systèmes Cellulaires de Fabrication (SCF).

Ce type de systèmes est particulièrement dédié aux job-shops fabriquant des produits ayant des gammes de fabrication courtes, non linéaires, et pouvant être groupées en familles de gammes utilisant les mêmes ressources. L'objectif de tels systèmes est de grouper les ressources de l'atelier en cellules. Chaque cellule est dédiée à une famille de produits qui utilisent essentiellement les ressources de cette cellule. Ceci est illustré par la figure 1.3 (Dans cette figure nous représentons les flux principaux en traits gras et les flux secondaires en traits fins).

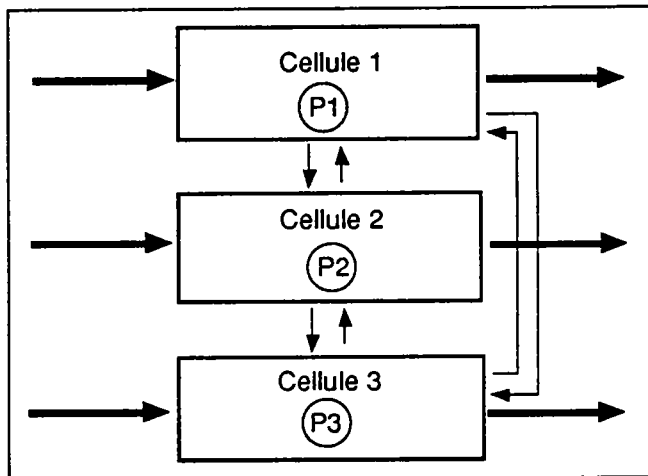


Figure 1.3 : Système cellulaire de fabrication

Les avantages des SCF sont multiples : En plus de la vitesse et de la flexibilité, ces systèmes conduisent à une réduction des temps de manutention et des en-cours, à l'amélioration de la productivité et à la simplification du contrôle et de la gestion (gestion hiérarchisée).

3. Organisation de la thèse

Nous organisons cette thèse en trois parties. La partie A est consacrée à la présentation du problème global d'agencement des systèmes cellulaires de fabrication. Elle est suivie par une présentation de la méthode de résolution de ce problème.

La partie B est dédiée au thème principal de cette thèse : le problème de l'agencement des cellules dans les ateliers de fabrication. Nous avons

également étudié, dans cette partie, le problème de l'agencement dynamique des systèmes de fabrication.

Enfin, dans la partie C, nous présentons le logiciel d'agencement des ressources dans un atelier de fabrication suivi de quelques applications.

L'annexe contient les différents algorithmes et techniques utilisés ainsi que les résultats des applications numériques.

PARTIE **A**

1. Etat-de-l'art de l'agencement des systèmes de fabrication

1.1. Domaines concernés et représentations géométriques associées

1.2. Formulations mathématiques et méthodes de résolution existantes

2. Agencement des systèmes cellulaires de fabrication

2.1. Introduction

2.2. Module de pré-traitement des données

2.3. Module de conception des cellules

2.4. Module d'agencement intra-cellulaire

3. Conclusion

1. ETAT-DE-L'ART

Dans cette partie, nous présentons les travaux effectués antérieurement dans le domaine de l'agencement des systèmes de production. Cet état-de-l'art est organisé de la manière suivante : nous commençons par le recensement des différents domaines où se pose ce problème. Puis, en se basant sur le schéma de la figure A1, nous introduisons les différentes représentations géométriques avec les possibilités d'utilisation de chacune d'elles dans les domaines recensés. Enfin, nous listons les approches de résolution proposées dans la littérature.

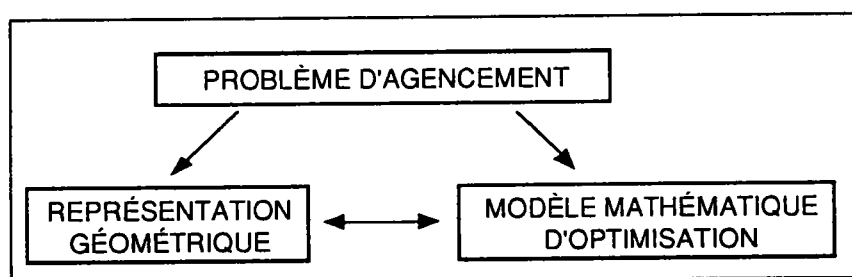


Figure A1 : Relation entre problème d'agencement, représentation géométrique et modèle mathématique d'optimisation

1.1. Domaines concernés et représentations géométriques

L'agencement est un problème très général. On le rencontre notamment dans les domaines de l'industrie manufacturière, de l'urbanisme, du textile ou en électronique (VLSI). Deux représentations géométriques de base ont été adoptées jusqu'à présent.

1.1.1. Représentation géométrique pour un agencement par partage de la surface utile

Dans la littérature, cette représentation a été très largement utilisée pour résoudre les problèmes connus sous le nom de "Facility Layout Design" (Hassan et coll. 1986). La figure A2 montre un exemple de partage d'une surface rectangulaire donnée. Elle s'adapte très bien aux problèmes d'agencement dans le domaine de l'urbanisme, et particulièrement pour l'agencement interne des

bâtiments d'habitation, de travail ou de services. Elle peut être utilisée également pour l'agencement des systèmes cellulaires de fabrication dans lesquels les cellules sont orientées opérations (groupes de machines réalisant les mêmes opérations. Voir Introduction §2.2). Cela est possible puisque le positionnement des machines dans la cellule, et par conséquent la forme de la cellule, n'a pas une grande influence sur le fonctionnement du système (voir figure A3). Seule la surface de la cellule importe. Ceci dit, une tentative d'agencement de cellules orientées opérations a été faite par Kar Yan Tam (1992) en utilisant cette représentation géométrique du système de fabrication. Une autre utilisation est l'agencement de cellules fonctionnelles CMOS en micro-électronique (VLSI) . La figure A4 donne un exemple d'un tel positionnement.

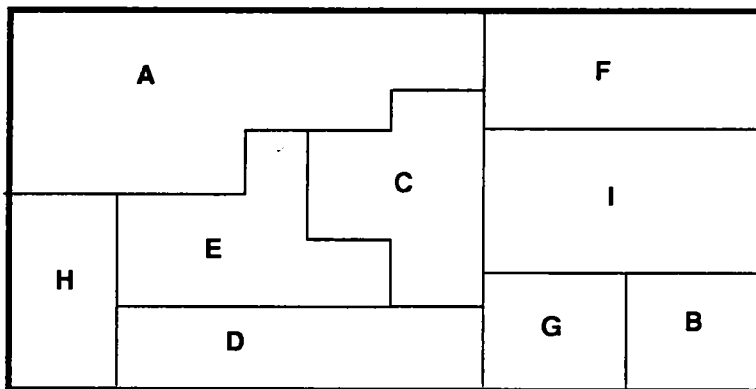


Figure A4 : Agencement par partage de la surface utile

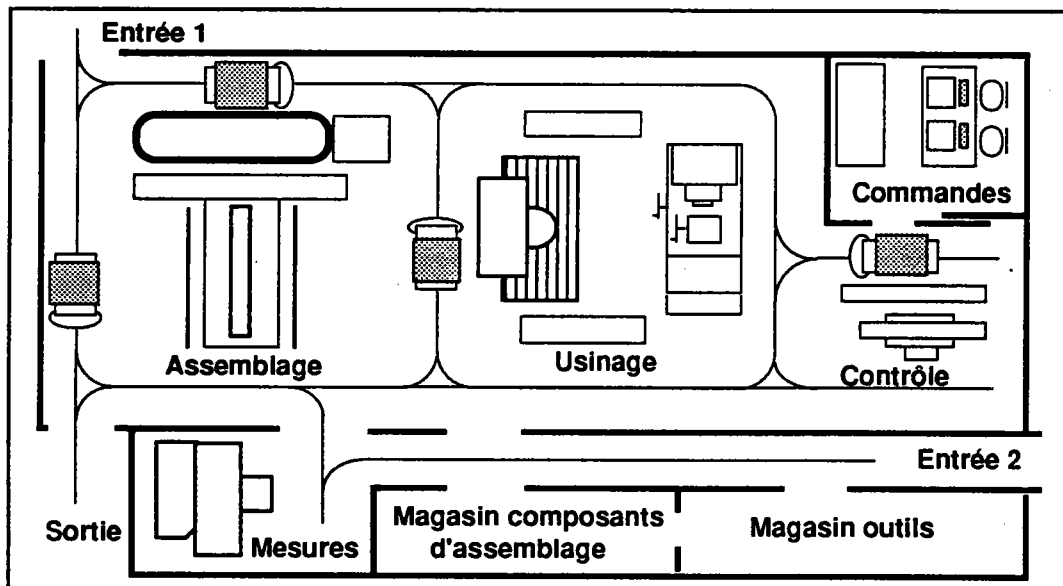


Figure A3 : Positionnement des ressources sur la surface d'un atelier

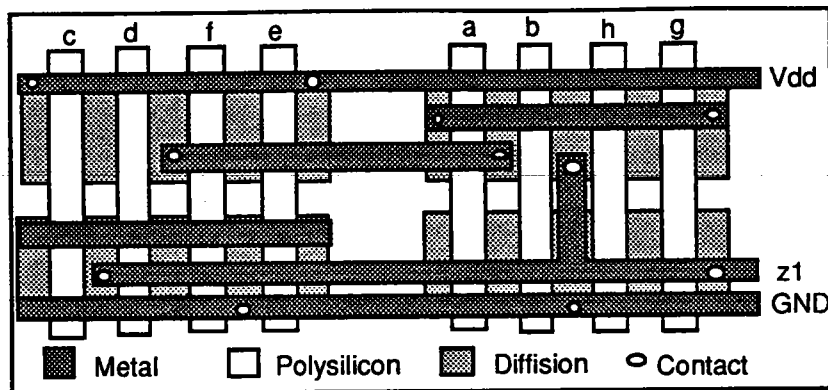


Figure A4 : Agencement des circuits CMOS

1.1.2. Représentation géométrique pour un agencement par positionnement des ressources de forme et de dimensions données

Cette représentation géométrique est utilisée principalement dans les systèmes de fabrication, où il s'agit d'installer d'une manière optimale des machines ou des cellules de forme et de dimensions données sur une surface donnée (voir figure A5 où un ensemble de machines $\{M_1, M_2, \dots, M_6\}$ sont affectées à 6 sites disposés suivant 2 lignes). Le problème est identique lorsqu'il s'agit de positionner les ateliers d'une usine sur la surface allouée (voir figure A6) ou de positionner plusieurs usines sur un territoire donné.

Elle est également utilisée en électronique et en micro-électronique pour l'agencement des macro-cellules d'un circuit intégré (Sechen, 1990). La figure A7 représente ce cas. Elle peut être utilisée pour l'agencement externe des bâtiments dans le domaine de l'urbanisme.

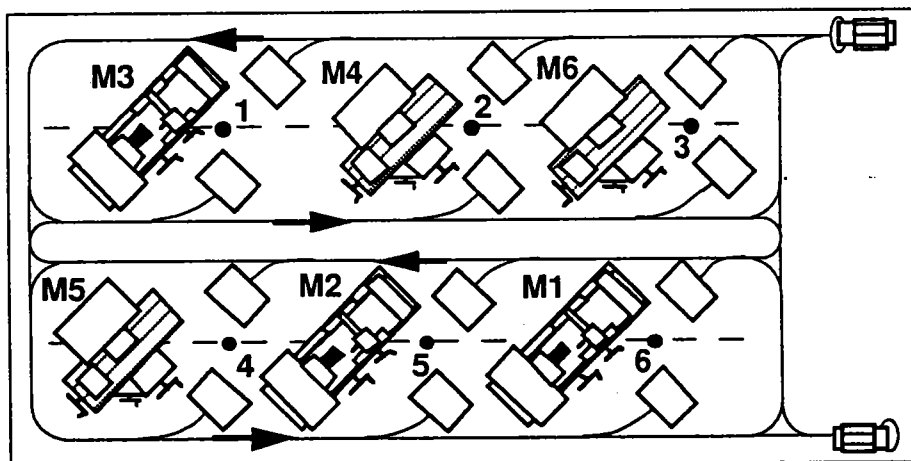


Figure A5 : Agencement d'un atelier de fabrication

Le positionnement des ressources se faisait jusqu'à présent par Affectation à des Sites Fixés à l'avance sur la surface utile (ASF) (voir figure A8). Cela exigeait des réajustements pour éviter les chevauchements entre les ressources.

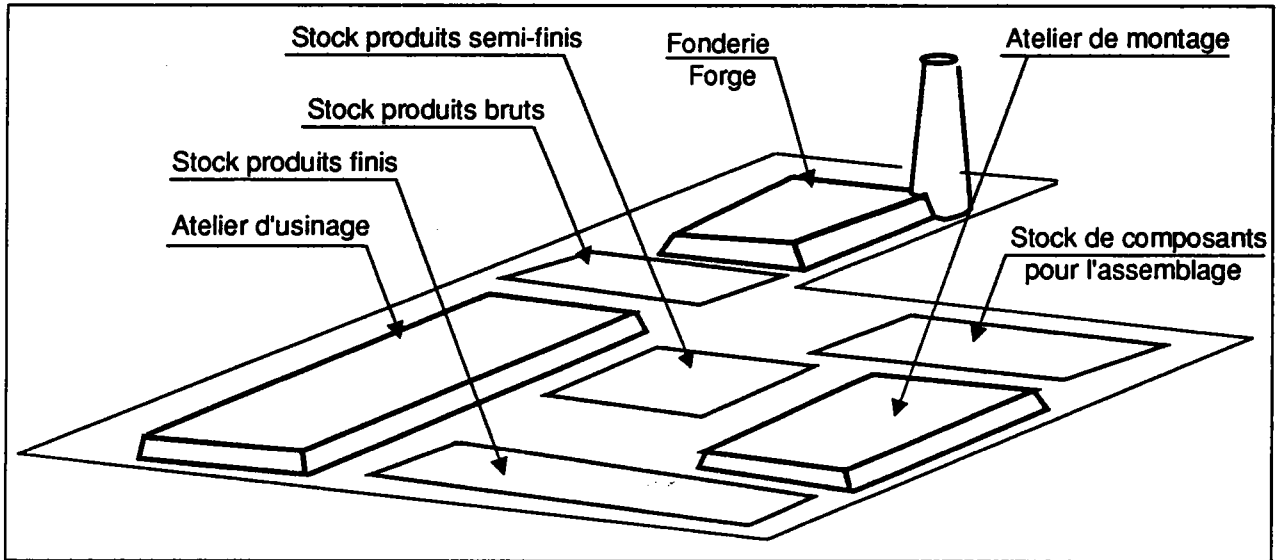


Figure A6 : Positionnement d'ateliers sur la surface de l'usine

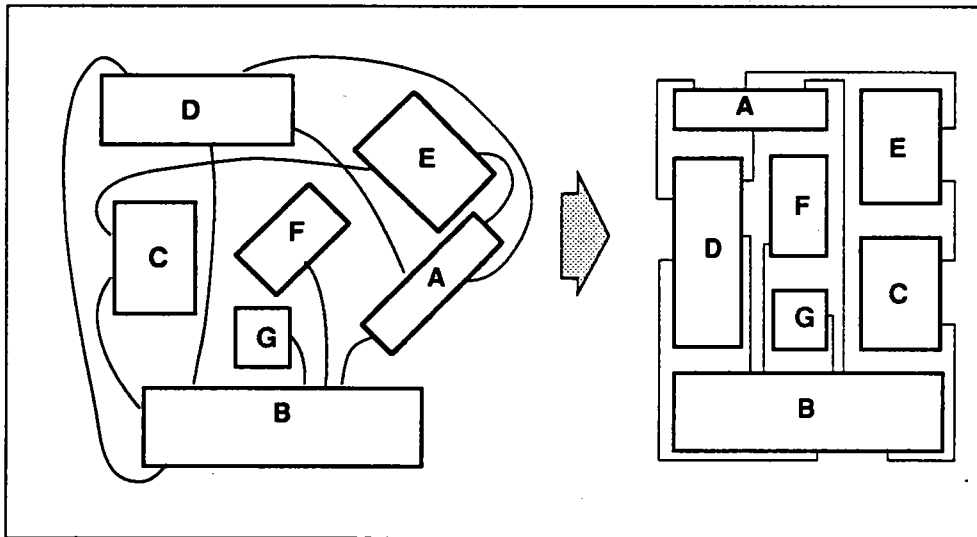


Figure A7 : Agencement de macro-cellules d'un circuit intégré

Nous proposons une représentation nouvelle de positionnement des ressources dans laquelle ces ressources, de formes rectangulaires et de dimensions données, sont Affectées à des Zones Libres (AZL) dont les limites sont définies par les positions des objets environnants (voir figure A9).

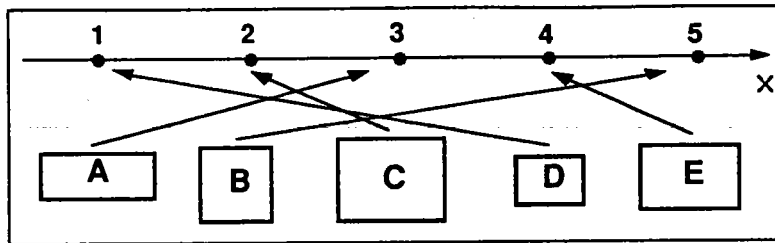


Figure A8 : Représentation géométrique pour un agencement par Affectation à des Sites Fixés (ASF)

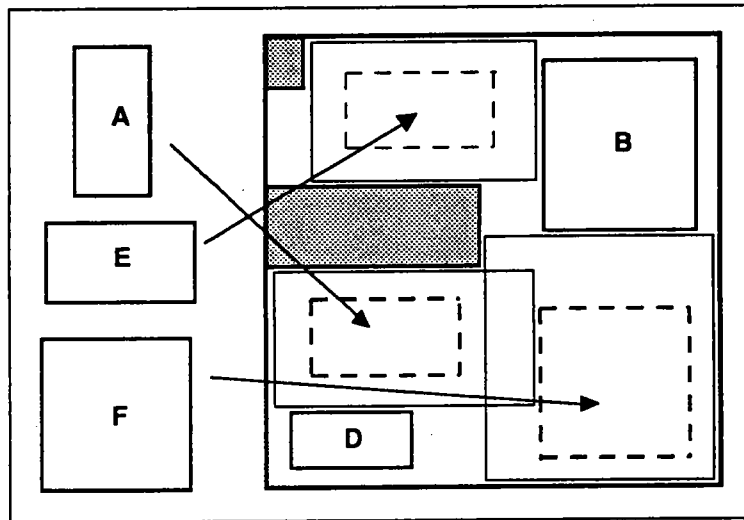


Figure A9 : Représentation géométrique pour un agencement par Affectation à des Zones Libres (AZL)

Les avantages de notre représentation géométrique par rapport à celle de l'agencement par partage sont d'une part la conservation des formes des ressources au cours de la résolution du problème, et d'autre part la possibilité de calculer les plus courts chemins entre les ressources, et par conséquent la possibilité d'utiliser des critères quantitatifs tenant compte de cette variable. Son avantage par rapport au modèle d'agencement par ASF est d'éviter de fixer a priori une configuration de base du système. Ceci est assez contraignant surtout lorsque les tailles des ressources sont très différentes (voir paragraphe 2.4.1).

1.2. Formulations mathématiques et méthodes de résolution existantes

Le problème d'agencement a initialement été formulé comme un problème d'affectation quadratique (Quadratic Assignment Problem 'QAP') par

Koopmans et Beckman (1959). C'est un problème NP-complet (Sahni et Gonzalez 1976) pour lequel la taille des systèmes à agencer est limitée (Finke et al. 1985). Pour cette raison, Kaufman et Broeckx (1978) et Balas et Mazzola (1984) ont suggéré de linéariser le critère. Cela n'ayant pas donné satisfaction, d'autres chercheurs ont développé des algorithmes optimaux et sous-optimaux.

Les premiers algorithmes ont été développés à partir des années 60. Certains de ces algorithmes sont basés sur la méthode dite d'exploration par séparation et évaluation (voir annexe B) (Gilmore, 1962 ; Lawer, 1963 ; Gavett et Plyter, 1966 ; Bazaraa et El-shafei, 1979 ; Kaku et Thompson, 1986), et d'autres sont basés sur la méthode dite du 'cutting plane' (Bazaraa et Sherali, 1980 ; Burkard et Bonninger, 1983). Les inconvénients de ces méthodes exactes résident dans le fait qu'elles exigent des capacités mémoire et des temps de traitement importants (ils augmentent avec le nombre d'objets à installer).

Dans la catégorie des algorithmes sous-optimaux, nous pouvons distinguer les algorithmes par construction, les algorithmes par amélioration et les algorithmes mixtes ou hybrides.

Dans les algorithmes par construction, les ressources sont installées l'une après l'autre, en utilisant certaines règles de priorité. Parmi les logiciels utilisant cette technique, il y a HC66 (Hillier et Connors, 1966), ALDEP (Seehof et Evens, 1967), CORELAP (Lee et Moore, 1967), PLANET (Deisenroth et Appel, 1972), INTALA (Dangelmaier, 1984), et FLAT (Heragu et Kusiak, 1986). Un cas particulier de ces algorithmes par construction est l'algorithme basé sur la théorie des graphes. Dans cet algorithme, la solution est construite en se basant sur un graphe traduisant la matrice des liens entre les ressources (Nozari et Ensore, 1981 ; Hassan et Hogg, 1987 ; Leung, 1992). Les algorithmes de cette catégorie donnent des performances temps de traitement/mémoire très intéressantes. Mais elles ne peuvent donner qu'une seule solution dont l'appréciation de la qualité reste à la charge de l'utilisateur.

Dans les algorithmes par amélioration, le processus commence toujours par une solution initiale pouvant être générée aléatoirement. On procède ensuite à des améliorations successives de cette solution afin d'arriver à une solution souvent sous-optimale, où plus aucune amélioration n'est à espérer. Certains logiciels ont été développés autour de cette approche, comme H63 (Hillier 1963), CRAFT (Buffa et coll., 1964), COFAD (Tompkins et Reed, 1976) et PVT (Warnecke et coll., 1981).

Les algorithmes hybrides, quant à eux, utilisent le principe des algorithmes par construction pour avoir une bonne solution initiale, puis celui

des algorithmes par amélioration pour affiner cette solution. Les applications les plus connues sont celles de El-shefei (1977), Bazaraa et Kirna (1983), ainsi que le logiciel FLAC (Scriabin et Vegin, 1985).

Il existe aussi d'autres méthodes dont l'objectif est de résoudre des problèmes à plusieurs critères. Certaines utilisent des techniques de recherche opérationnelle, et optimisent un critère qui est une combinaison linéaire des critères considérés. Malakooti dans CAFLAS (1987) et Urban (1987) ont appliqué cette technique. D'autres utilisent les techniques de l'intelligence artificielle. Les produits les plus connus qui ont appliqué ces techniques sont FADES (Fisher et Nof, 1984), IFLAPS (Tirumpatikumara et coll., 1985) et KBML (Kusiak, 1990).

Les logiciels que nous avons présenté dans cet état-de-l'art résolvent le problème d'agencement de base, c'est-à-dire la disposition des ressources ou des services sur une surface donnée. Nous proposons, dans ce qui suit, de résoudre un problème plus global qui est l'agencement des systèmes cellulaires de fabrication.

2. PROBLÈME D'AGENCEMENT DES SYSTÈMES CELLULAIRES DE FABRICATION

Nous nous intéressons au problème d'agencement des système de production fabricant un grand nombre de types de produits ayant des gammes très diversifiées. Dans ce type de système, les flux entre les ressources sont importants, ce qui conduit à des coûts de transport et de manutention des produits considérables si l'agencement n'est pas fait convenablement. Pour ce type de système de production, l'agencement cellulaire est le plus approprié. Nous proposons pour cela une nouvelle approche d'agencement cellulaire des systèmes de fabrication dans laquelle les cellules sont orientées produits (voir introduction §2.3).

2.1. Introduction

Notre approche de résolution de ce problème comporte quatre modules. Nous les résumons dans ce qui suit :

Module 1 : Dans ce premier module se fait le pré-traitement des données brutes concernant la production. Il transforme le problème multi-gammes initial

(dans lequel certains types de produits disposent de plusieurs gammes de fabrication) en un problème mono-gamme (dans lequel chaque type de produit dispose d'une seule gamme de fabrication). Puis, il déduit les routages de ces gammes (voir définitions de gamme et de routage ci-dessous). En d'autres termes, les principales activités de ce module sont : (i) l'allocation des produits aux ressources et (ii) la localisation et le dimensionnement des stocks dans le système.

Module 2 : C'est le module de conception des cellules. Dans ce module, on regroupe les ressources en cellules et les types de produits en familles. Chaque cellule est dédié à une famille de types de produits. L'objectif est de minimiser le trafic inter-cellules en tenant compte de certaines contraintes de cohabitation entre les ressources.

Module 3 : Le but de ce troisième module est de placer physiquement les ressources à l'intérieur des cellules. Cette opération est réalisée en trois étapes. Dans la première étape, un premier Système Expert (SE) sélectionne un Système de Manutention (SdM). La deuxième étape consiste à déduire la configuration de base de la cellule à partir du SdM sélectionné, et ceci à l'aide d'un second SE. La troisième étape est consacrée au problème d'affectation des ressources à des sites disposés sur la configuration de base choisie. L'objectif est de minimiser le coût de manutention à l'intérieur de la cellule. Ce module a fait l'objet d'une thèse au sein du projet SAGEP (Hamann, 1992).

Module 4 : Ce module constitue le thème principal de cette thèse. Il a pour objectif l'installation des cellules sur le sol de l'atelier, l'objectif étant de minimiser le coût de fonctionnement du système de production.

La figure A10 résume la stratégie ci-dessus. Le dernier module représenté sur le diagramme est le module d'affinement du résultat final. Il donne à l'utilisateur la possibilité de faire les modifications jugées nécessaires pour le respect de certains critères qualitatifs non pris en compte dans les modules précédents.

Dans la suite nous présentons avec plus de détails chacun des trois premiers modules, le quatrième faisant l'objet de la partie B de cette thèse.

Définitions

Avant de poursuivre, nous donnons quelques définitions utiles pour la suite.

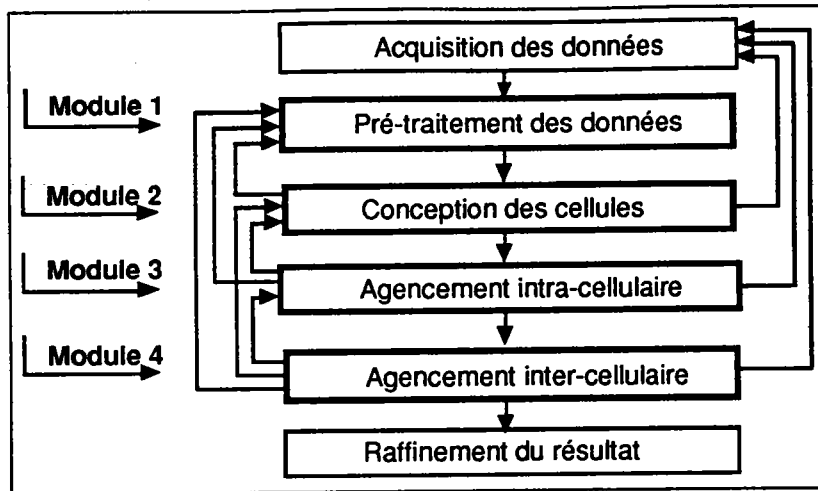


Figure A10 : La stratégie de conception des systèmes cellulaires de fabrication

- **Cellule** : ensemble de ressources groupées et destinées à réaliser, aussi complètement que possible, une famille de produits qui nécessitent des ressources similaires.

- **Système de manutention** : moyen de déplacement à faible portée des produits entre les machines ou sur une même machine (utilisé généralement dans les cellules ou sur les machines).

- **Système de transport** : moyen de déplacement à grande portée des produits (utilisé généralement entre les cellules).

- **Opération** : action réalisée sur un produit à l'aide d'une machine. Les principaux types d'opération sont : (i) les opérations de déformation de matière (moulage, forgeage, estampage, ...), (ii) les opérations d'enlèvement de matière (usinage, découpage, ...), (iii) les opérations d'apport de matière (assemblage, soudage, ...), et (vi) les opérations de changement des caractéristiques physiques de la matière (traitements thermiques).

- **Gamme de fabrication** : suite ordonnée d'opérations subites par un produit donné durant sa fabrication avec les temps opératoires correspondants.

- **Routage** : suite ordonnée des machines que visite le produit durant sa fabrication. Elle est extraite de la gamme de fabrication du produit et de la composition du parc machine de l'atelier (voir point suivant).

2.2. Module de pré-traitement des données

Le module de pré-traitement a pour objectif : (i) de répartir la production d'un type de produits donné sur l'ensemble des gammes disponibles, et (ii) de

déduire les routages relatifs à ces gammes ; les discontinuités des flux détermineront l'emplacement des stocks.

2.2.1. La discontinuité des flux

Le problème de la discontinuité des flux est généralement dû, pour un produit donné, soit à un changement de stratégie de fabrication, d'une production par lot à une production par flot ou inversement, ou à une forte différence entre les périodes de travail de ressources successives. Cela engendre un problème de mise en place de dimensionnement et de gestion du stock de produits entre les opérations.

Soit par exemple un produit P_i dont le routage est : $R_i = \langle M_{i,1}, M_{i,2}, M_{i,3}, M_{i,4}, M_{i,5} \rangle$. Nous supposons qu'il y a une discontinuité dans le flux de matière entre les machines $M_{i,3}$ et $M_{i,4}$. Cette discontinuité est due, par exemple, au fait que $(M_{i,1}, M_{i,2}, M_{i,3})$ travaillent 24h/24h et que $(M_{i,4}, M_{i,5})$ ne travaillent que 16h/24h. Comme parmi nos objectifs lors de la composition des cellules, il y a l'homogénéité du fonctionnement des machines dans la cellule, l'autonomie de la cellule, et la continuité du flux dans la cellule, on ne pourra jamais mettre $M_{i,3}$ et $M_{i,4}$ dans une même cellule sans avoir remédié à ce problème en insérant entre elles un stock tampon **S**. Le routage devient à ce moment là :

$$R_i = \langle M_{i,1}, M_{i,2}, M_{i,3}, \mathbf{S}, M_{i,4}, M_{i,5} \rangle.$$

2.2.2. Les produits multi-gammes

Pour un même produit, il peut y avoir plusieurs gammes de fabrication. Cela est possible du fait que certaines opérations peuvent être faites sur des machines de types différents, et/ou que l'ordre entre certaines opérations de la gamme peut être modifié, et/ou que différentes séquences d'opérations (effectuées sur des machines totalement différentes) sont possibles. Nous supposons donc que le produit dispose d'une gamme principale et d'une ou de plusieurs gammes secondaires.

Une répartition de la demande d'un produit donné sur ses différentes gammes devient indispensable lorsque les capacités des machines réalisant les opérations des gammes principales sont insuffisantes pour satisfaire la demande. Cela doit se faire tout en préservant l'équilibre des charges des machines. Cette répartition peut n'être qu'occasionnelle dans les cas suivants : (i) une demande supplémentaire survient en causant une augmentation de la charge de certaines machines au-delà de leur capacité ; (ii) une machine importante de la gamme principale tombe en panne. On procède alors

automatiquement à un changement de gamme ou a une répartition de la demande sur les autres gammes du produit.

2.3. Module de conception des cellules

Le but de ce module est de grouper les ressources en cellules de sorte que, durant leur fabrication, les produits passent la plus grande partie (sinon la totalité) de leur temps de fabrication dans une même cellule. Cela est fait en minimisant le trafic inter-cellules, sachant que le nombre de machines dans chaque cellule est limité. Ce problème de conception de cellule de fabrication, largement traité dans la littérature, est un problème NP-difficile. Plusieurs méthodes exactes et heuristiques ont été proposées pour le résoudre. Certaines sont basées sur les méthodes de classification. Un aperçu global sur ces méthodes est donné par C-H Chu (1989). D'autres sont basées sur l'agrégation progressive (McAuley, 1972). Il existe aussi des méthodes basées sur la théorie des graphes (Rajagopalan et Batra, 1975). Pour réduire encore plus le trafic inter-cellules, Seifoddin (1989) propose une méthode de duplication des machines goulot, basée sur une étude de rentabilité. La méthode que nous proposons procède par amélioration, elle est basée sur l'algorithme du Recuit Simulé (R.S). Cette approche a été étudiée par Harhalakis, Proth et Xie (Harhalakis et al. 1990).

2.3.1. Position du problème

Soit $M = \{M_1, M_2, \dots, M_q\}$ l'ensemble des machines, et $P = \{P_1, P_2, \dots, P_p\}$ l'ensemble des types de produits destinés à être fabriqués sur M .

Pour chaque type de produits P_k ($k = 1, 2, \dots, p$), on définit :

(i) une séquence unique de machines visitées par le produit lors de sa production. Cette séquence, dite routage, s'écrit : $P_k = \langle M_{k,1}, M_{k,2}, \dots, M_{k,s_k} \rangle$,

où : $M_{k,t} \in M$ pour $k = 1, 2, \dots, p$ and $t = 1, 2, \dots, s_k$;

s_k étant la longueur du routage du produit P_k .

(ii) r_k le ratio moyen de produits du type P_k à produire par unité de temps.

Pour chaque triplet $(P_k, M_i, M_j) \in P \times M \times M$, on note $v_{k,i,j}$ le nombre de fois où M_i suit M_j ou M_j suit M_i dans le routage P_k .

Pour chaque couple de machines $(M_i, M_j) \in M \times M$, on exprime le trafic entre ces machines par :

$$t_{i,j} = \sum_{k=1}^p r_k \times v_{k,i,j} \quad (1)$$

Soit $C = \{C_1, C_2, \dots, C_w\}$ une partition de l'ensemble M telle que :

$$\bigcup_{i=1}^w C_i = M \quad \text{et} \quad \bigcap_{i=1}^w C_i = \emptyset \quad (2)$$

$$\text{Soit } E_c(C) = \{(m,n) / m \in C_k \text{ et } n \in C_l; k, l = 1, 2, \dots, w \text{ et } k \neq l\} \quad (3)$$

l'ensemble des couples de machines (M_m, M_n) appartenant à deux cellules différentes C_k et C_l de C .

Le trafic inter-cellules total de la partition C est :

$$T_c(C) = \sum_{(m,n) \in E_c(C)} t_{m,n} \quad (4)$$

Soit S l'ensemble des partitions de M qui vérifient les contraintes suivantes :

$$(i) \forall C \in S \text{ et } \forall C_i \in C \ (i = 1, 2, \dots, w), \text{ card}(C_i) \leq N, \quad (5)$$

(N étant le nombre maximal de machines par cellule).

(ii) Certains couples de machines (M_i, M_j) doivent obligatoirement obéir à l'une des deux conditions suivantes :

$$\forall C \in S, (i, j) \in E_c(C) \quad (6)$$

$$\text{ou } \forall C \in S, (i, j) \notin E_c(C) \quad (7)$$

(c'est-à-dire que les machines i et j **doivent** être placées dans des cellules différentes (6) ou dans une même cellule (7)). Si cette contrainte n'est pas respectée, la partition associée ne peut être acceptée.

(iii) Pour d'autres couples de machines, le respect de (6) et (7) n'est que **souhaitable** à des degrés variables.

Le problème à résoudre consiste donc à trouver une partition $C^* \in S$ de l'ensemble des machines M telle que :

$$T_c(C^*) = \text{Min}_{C \in S} T_c(C) \quad (8)$$

Pour tenir compte des contraintes du type (iii), nous ajoutons dans l'expression du trafic reliant deux cellules d'une partition \mathbf{C} donnée un facteur de pénalisation. Ce facteur devient non nul si une contrainte de ce type n'est pas respectée. La nouvelle expression du trafic est :

$$T_c(\mathbf{C}) = \sum_{(m,n) \in E_c(\mathbf{C})} t_{m,n} + \left[\max_{(i,j) \in M^2} (t_{i,j}) \times \sum_{(m,n) \in M^2} (k_{m,n} \times \alpha_{m,n}) \right] \quad (9)$$

où :

$$k_{m,n} = \begin{cases} 1 & \text{si l'une des contraintes (iii) n'est pas respectée,} \\ 0 & \text{sinon.} \end{cases} \quad (10)$$

$\alpha_{m,n}$ est un poids représentant l'importance donnée à la contrainte entre M_m et M_n . Sa valeur est prise dans le tableau A1.

<i>Désignation de la contrainte</i>	<i>Poids</i>
Très importante	2,000
Importante	1,000
Moyennement importante	0,500
Peu importante	0,250
Facultative	0,125

Tableau A1 : Poids des contraintes de proximité

2.3.2. Méthode de résolution

Comme nous l'avons noté précédemment, plusieurs approches ont été proposées pour résoudre ce problème. Nous avons montré dans la présentation de la méthode du Recuit Simulé (voir annexe A) que seuls trois points de l'algorithme dépendent de l'application visée. Ces points sont : la construction de la partition initiale, le passage d'une partition à une autre, et le calcul de la valeur du critère associée à une partition donnée. Nous allons décrire, dans ce qui suit, chacun de ces trois points.

2.3.2.1. Construction d'une solution initiale

La configuration initiale est construite aléatoirement. Le nombre initial de cellules est égal à :

$$\begin{cases} \text{card}(M) / N & \text{si } \text{card}(M) \text{ est un multiple de } N \\ \lfloor \text{card}(M) / N \rfloor + 1 & \text{sinon} \end{cases} \quad (11)$$

où $\lfloor x \rfloor$ est l'entier immédiatement inférieur à x ($x \in \mathfrak{R}^+$).

Lors de la construction de la partition initiale, chaque machine M_i ($i = 1, 2, \dots, q$) est affectée à une cellule choisie aléatoirement en respectant les contraintes (i) et (ii). L'algorithme est le suivant :

1. Pour chaque sous-ensemble de machines à grouper impérativement :

Faire :

- Sélectionner aléatoirement une cellule ne contenant pas de machine incompatible avec une des machines du sous-ensemble.

Si la cellule sélectionnée ne peut pas recevoir toutes les machines du sous-ensemble,

alors, sélectionner une autre cellule,

sinon, effectuer le sous-ensemble à la cellule et aller en 2.

Tant que le nombre de cellules non sollicitées est non nul.

Si le sous-ensemble n'a pas été affecté à l'une des cellules existantes,

alors créer une nouvelle cellule et lui affecter le sous-ensemble.

2. Pour chaque machine restante et soumise à une contrainte d'éloignement :

Faire :

- Sélectionner aléatoirement une cellule ne contenant pas de machine incompatible avec la machine considérée.

Si la cellule sélectionnée ne peut plus recevoir de machines,

alors, sélectionner une autre cellule,

sinon, effectuer la machine à la cellule et aller en 3.

Tant que le nombre de cellules non sollicitées est non nul.

Si la cellule n'a pas été affectée à l'une des cellules existantes,

alors créer une nouvelle cellule et lui affecter la machine.

3. Affecter les machines restantes aléatoirement.

2.3.2.2. Transformations élémentaires

La transition d'une partition C de l'ensemble M vers une partition voisine C' est réalisée en utilisant une des transformations élémentaires ci-dessous. La transformation à effectuer est choisie aléatoirement (voir figure A11) :

- a • Transfert d'une machine vers une autre cellule.
- b • Permutation de deux machines appartenant à des cellules différentes.
- c • Transfert d'une machine de sa cellule vers une cellule nouvellement créée.

On procède comme suit :

- les transformations, les cellules et les machines sont choisies aléatoirement,
- si les partitions ne vérifient pas les contraintes (i) et (ii), elles sont refusées.

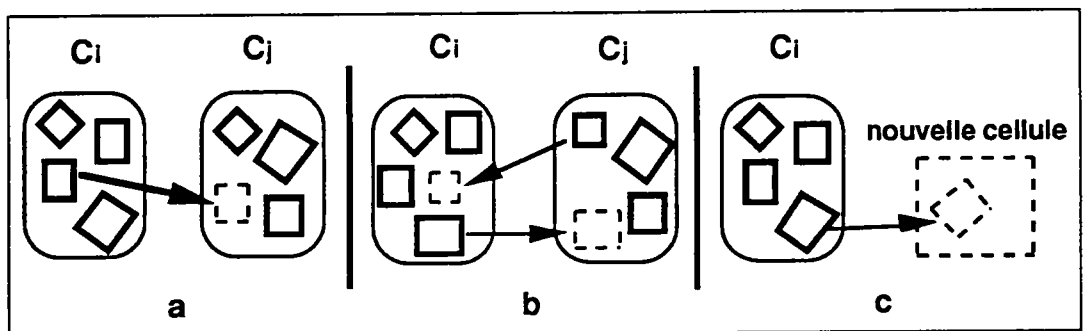


Figure A11 : Transformations élémentaires

2.3.2.3. Coût d'une partition

Enfin, le coût d'une partition donnée est calculé à l'aide de la formule (9). Lorsque le système ne viole aucune contrainte les coefficients $\alpha_{m,n}$ sont nuls.

Notons que l'un des avantages incontestables du RS, lorsqu'il est utilisé pour résoudre ce problème particulier, est qu'il permet très facilement d'intégrer de nouveaux types de contraintes. Les autres méthodes ne le permettent pas ; elles sont dédiées et rigides.

2.3.3. Problème des machines multiples

Dans l'exposé ci-dessus, nous avons considéré qu'il n'y avait qu'un seul exemplaire de chaque type de machine dans l'atelier, alors qu'il est très fréquent, dans l'industrie mécanique principalement, d'avoir dans un même atelier plusieurs machines qui réalisent les mêmes opérations.

Dans le cas où l'atelier est organisé en cellules orientées opérations (cellule de tournage, cellule de fraisage, etc...), l'affectation des produits aux

machines n'est pas pré-déterminée, elle se fait lors de la fabrication suivant des règles locales (priorités, temps de réglage, savoir faire de l'opérateur, ...). Par contre, dans une organisation en cellules orientées produits (où les machines sont souvent de types différents), l'affectation des produits aux machines doit être faite au préalable. On doit connaître le routage exact de chaque produit avant de lancer la production. Cette affectation des produits aux ressources doit se faire en tenant compte de la capacité des machines et des machines multiples afin d'éviter un accroissement du trafic inter-cellules auquel on ne pourra pas remédier par la suite. Xie (1993) propose une heuristique pour le problème de composition des cellules qui tient compte des machines multiples et dont l'efficacité a été mise en évidence sur des exemples industriels .

2.4. Module d'agencement intra-cellulaire

Dans l'introduction (paragraphe 2.1 de ce chapitre), nous avons présenté les trois étapes de la méthode de résolution du problème d'agencement intra-cellulaire : le choix du système de manutention, le choix de la configuration de la cellule et l'affectation des machines aux sites. Nous organisons ce paragraphe comme suit : nous donnons d'abord le processus de résolution du problème général, nous montrons ensuite ce que sont les résultats des deux premières étapes, puis nous posons le problème de la troisième étape et proposons une méthode pour le résoudre.

2.4.1. Processus général de résolution

La figure A12 illustre le processus global de conception d'une cellule de fabrication que nous décrivons ci-dessous.

Le rôle du premier SE est de sélectionner un SdM (système de manutention) en se basant sur les caractéristiques des produits (matière, poids, forme, dimensions, taille du lot, demande, ...), celles des ressources (type, capacité, dimensions, procédé de chargement et de déchargement des produits, vitesse de manutention, précision de manutention, ...) et celles de la cellule (surface maximale, forme, présence humaine, ...). Le choix se portera sur l'un des types de SdM suivants : les robots, les ponts roulants, les convoyeurs, les systèmes en carrousel et les véhicules filo-, opto- ou auto-guidés (voir figure A13).

Le deuxième SE est consacré au choix de la configuration de base de la cellule, c'est-à-dire de la manière dont sont disposés les sites auxquels seront

affectées les ressources. Ce choix dépend principalement du SdM choisi mais aussi des caractéristiques des produits, des machines et de la cellule. Les formes de base les plus communes sont : la ligne simple, la ligne circulaire, double ligne et multi-lignes. On peut aussi avoir des configurations de base quelconques, décrites par les coordonnées des sites par rapport à un repère orthonormé donné (voir figure A14). Ces deux systèmes experts ont été largement traité par T. Hamann dans sa thèse (Hamann, 1992), voir aussi (Hamann et Vernadat 1992), (Hamann et al., 1992) et (Kusiak, 1986).

Une fois le SdM et la configuration de base choisis, on entame la troisième étape de ce module qui est la résolution du problème d'affectation. Un raffinement de la solution finale est souvent nécessaire. Il est effectué par l'opérateur. Nous nous intéressons particulièrement à ce troisième problème que nous traitons en détails dans la suite.

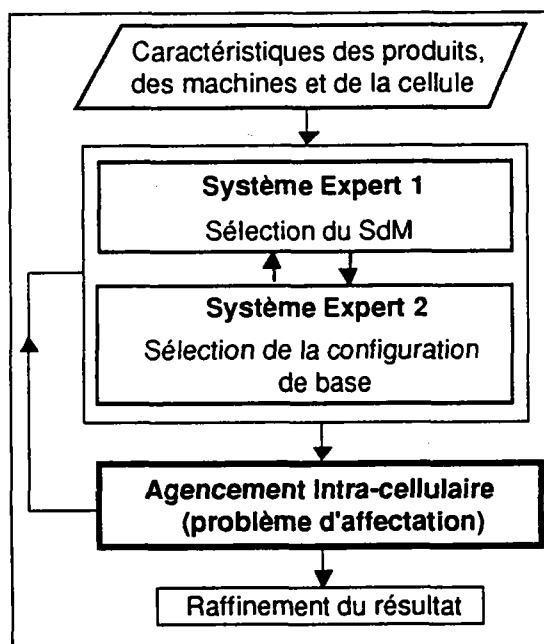


Figure A12 : Processus global de résolution du problème d'agencement intra-cellulaire

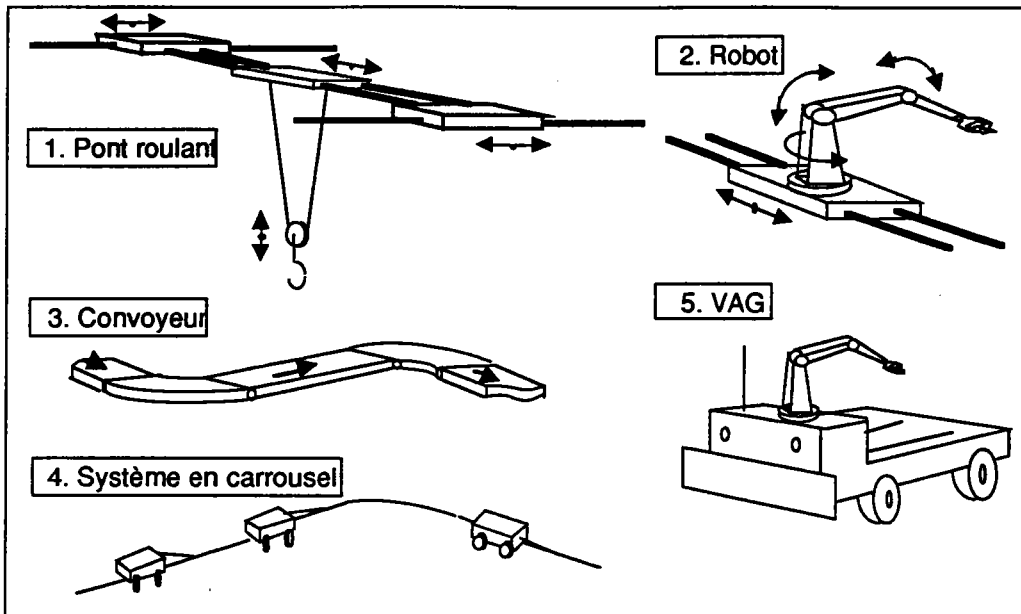


Figure A13 : Les SdM utilisables dans une cellule de fabrication

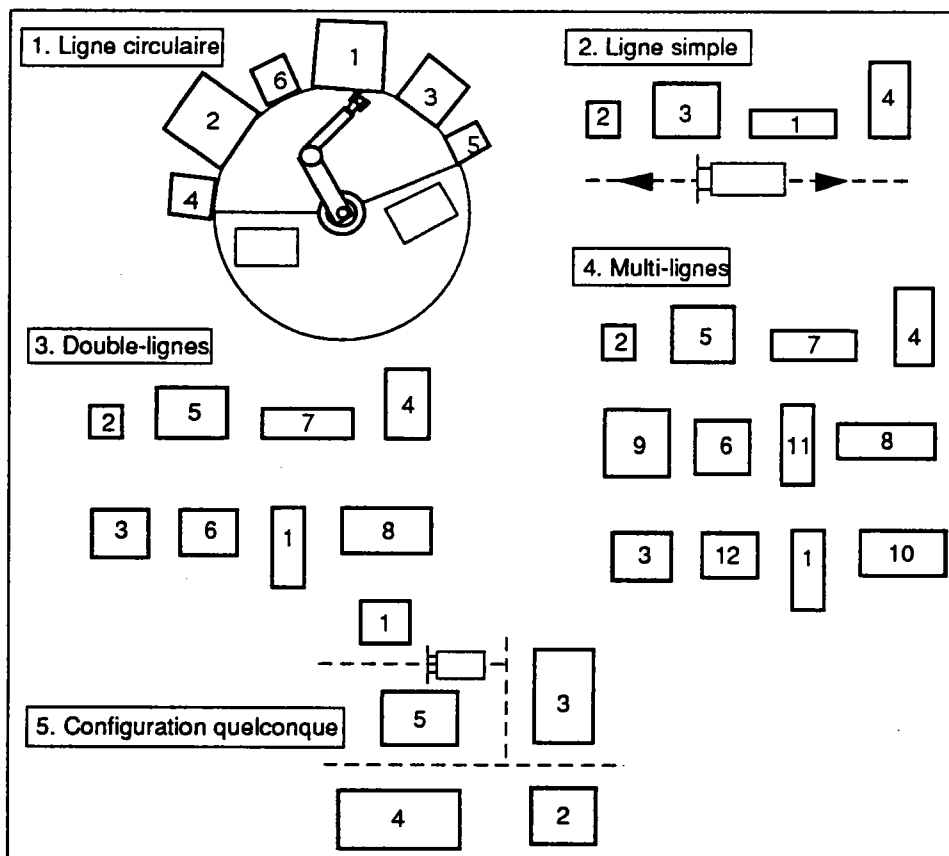


Figure A14 : Les différentes configurations de base des cellules

2.4.2. Position du problème d'affectation

Ce problème a été très souvent modélisé comme un problème d'affectation quadratique qui est NP-difficile. Nous proposons ci-dessous un modèle mathématique bien adapté à la méthode de résolution que nous utilisons ; le recuit simulé.

Soit C une cellule composée de m machines, $C = \{M_1, M_2, \dots, M_m\}$ ($C \subset M$). Cette cellule est dédiée à la fabrication d'une famille de produits P_f comportant n types de produit, $P_f = \{P_1, P_2, \dots, P_n\}$ ($P_f \subset P$). En plus des notations utilisées dans le module précédent, nous introduisons les suivantes :

- Pour chaque triplet $(P_k, M_i, M_j) \in P_f \times C \times C$, nous notons par $u_{k,i,j}$ la taille du lot des produits de type P_k transportés de la machine M_i vers la machine M_j .
- Le flux de matière partant de la machine M_i vers la machine M_j est défini par :

$$f_{i,j} = \sum_{k \in E_{i,j}} r_k \times \frac{v_{k,i,j}}{u_{k,i,j}} \quad (12)$$

où $E_{i,j} = \{k / u_{k,i,j} \neq 0\}$ (Note : $u_{k,i,j} = 0$ si et seulement si $v_{k,i,j} = 0$).

- Pour chaque couple de machines $(M_i, M_j) \in C \times C$, on définit :
 - (i) le coût de manutention par unité de distance, noté $c_{i,j}$ (nous supposons qu'il n'y a qu'un seul moyen de manutention par cellule, $c_{i,j}$ est identique pour tous les couples de machines) ;
 - (ii) la distance entre la sortie de la machine M_i et l'entrée de la machine M_j , notée $d_{i,j}$ (le calcul de cette distance dépend à la fois du SdM et de la configuration de base).

L'indice du site d'une machine M_i est noté $s(i)$.

Il s'agit, dans ce problème, de trouver l'affectation des machines aux sites qui minimise :

$$\sum_{i=1}^m \sum_{\substack{j=1 \\ j \neq i}}^m c_{i,j} \times f_{i,j} \times d_{i,j} \quad (13)$$

Cette fonction objectif (minimisation du coût de manutention par unité de temps) est soumise aux contraintes suivantes :

- (i) $\forall i, j \in \{1, 2, \dots, m\}; (i \neq j) \Rightarrow (s(i) \neq s(j))$, (c'est-à-dire qu'il ne peut y avoir plus d'une machine sur un même site).

(ii) $\forall i, j \in \{1, 2, \dots, m\}$; si $(i \neq j)$ et $(M_i$ est adjacente à $M_j)$ alors $\Delta_{i,j} \geq \Delta_{\min}$,

où : $\Delta_{i,j}$ est la distance entre les côtés adjacents des machines M_i et M_j ,

Δ_{\min} est la distance minimale admissible entre deux côtés adjacents.

(iii) Certaines machines sont définitivement pré-affectées à certains sites et certains sites sont définitivement interdits à certaines machines.

(iv) Pour certains couples de machines (M_i, M_j) , il est préférable ou même nécessaire de les éloigner (resp. rapprocher) en gardant entre leurs centres une distance supérieure ou égale à $\Delta_{\min_{i,j}}$ (resp. inférieure ou égale à $\Delta_{\max_{i,j}}$).

Les contraintes du type (iv) peuvent être dures (c'est-à-dire que si elles ne sont pas respectées, la solution est rejetée), ou bien molles (c'est-à-dire que si elles ne sont pas respectées, la valeur du critère est pénalisée en conséquence). La pénalisation de la valeur du critère d'une configuration qui ne respecte pas une contrainte molle du type (iv) consiste à modifier l'expression de la fonction objectif en lui additionnant un facteur de pénalisation. Ce facteur est le résultat de la multiplication des deux paramètres suivants :

- le premier, noté $\delta_{i,j}$, mesure l'importance de la violation de cette contrainte, sa valeur est calculée comme suit :

$$\delta_{i,j} = \begin{cases} \Delta_{\min_{i,j}} - d_{i,j} & \text{si } M_i \text{ et } M_j \text{ doivent être tenues éloignées et } d_{i,j} < \Delta_{\min_{i,j}}, \\ d_{i,j} - \Delta_{\max_{i,j}} & \text{si } M_i \text{ et } M_j \text{ doivent être tenues proches et } d_{i,j} > \Delta_{\max_{i,j}}, \\ 0 & \text{sinon.} \end{cases} \quad (14)$$

- le second, noté $\phi_{i,j}$, est le poids associé à la contrainte entre M_i et M_j , sa valeur est donnée par :

$$\phi_{i,j} = \begin{cases} \lambda_{i,j} \times \max_{k,l \in \{1,2, \dots, m\}} (f_{k,l}) & \text{si } \delta_{i,j} \neq 0, \\ 0 & \text{sinon.} \end{cases} \quad (15)$$

$\lambda_{i,j}$ est un coefficient qui augmente avec l'importance donnée à la contrainte et dont les valeurs sont prises dans le tableau A1.

L'expression de la fonction objectif est alors :

$$\sum_{i=1}^m \sum_{j=1}^m c_{i,j} \times (f_{i,j} \times d_{i,j} + \phi_{i,j} \times \delta_{i,j}) \quad (16)$$

2.4.3. Solution du problème d'affectation

Comme pour le problème précédent, nous proposons une nouvelle méthode basée sur l'algorithme du RS. Nous discutons, dans ce qui suit, les parties de l'algorithme du RS spécifiques à cette application, à savoir : la construction d'une solution initiale et le passage d'une solution à une solution voisine. Nous discutons aussi les problèmes de dimensionnement de la cellule et de localisation de ses entrées/sorties.

2.4.3.1. Quelques hypothèses

1. Dans le but de simplifier la présentation, nous considérons un exemple où la cellule est du type multi-lignes et où le SdM est un pont roulant. Ce cas est une généralisation des configurations du type ligne simple et double ligne. Nous simplifions les calculs, en supposant que l'entrée et la sortie de chaque machine sont au centre géométrique de celle-ci.

2. Les variables à calculer à chaque itération du RS sont : la distance entre les entrées/sorties des machines (voir expression (16)) et les distances entre les côtés adjacents de deux machines (voir contrainte (ii)). Les positions des machines dans la cellule sont données par les coordonnées de leur centres par rapport à un repère orthonormé (voir figure A15). Ces deux variables sont calculées comme suit :

• La distance entre une machine $M_i (x_i, y_i)$ et une machine $M_j (x_j, y_j)$ est Euclidienne, elle est donnée par :

$$d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (17)$$

• Si l_i (resp. w_i) est la dimension du côté parallèle à l'axe des abscisses (resp. à l'axe des ordonnées) d'une machine M_i , alors la distance entre deux côtés adjacents de deux machines M_i et M_j , si ces côtés sont parallèles à l'axe des ordonnées, est donnée par :

$$\Delta x_{i,j} = |x_i - x_j| - \frac{1}{2}(l_i + l_j), \quad (18)$$

et s'ils sont parallèles à l'axe des abscisses, elle est donnée par :

$$\Delta y_{i,j} = |y_i - y_j| - \frac{1}{2}(w_i + w_j) \quad (19)$$

2.4.3.2. Construction d'une solution initiale

Les lignes de la cellule sont séparées par une même distance Δ_{row} , et les sites d'une même ligne sont séparés par une même distance Δ_s .

$$\Delta_s = \max\left\{\Delta_{\min}, \max_{i,j}\{\Delta\min_{i,j}\}\right\} + \max_i\{l_i\} \quad (i, j = 1, 2, \dots, m) \quad (20)$$

$$\Delta_{row} = \max\left\{\Delta_{\min}, \max_{i,j}\{\Delta\min_{i,j}\}\right\} + \max_i\{w_i\} \quad (i, j = 1, 2, \dots, m) \quad (21)$$

Puisque, dans notre cas, le SdM est un pont roulant et les entrées/sorties des machines sont aux centres des machines, les deux distances sont égales à :

$$\Delta_s = \Delta_{row} = \max\left\{\Delta_{\min}, \max_{i,j}\{\Delta\min_{i,j}\}\right\} + \max_i\{\max(l_i, w_i)\} \quad (i, j = 1, 2, \dots, m) \quad (22)$$

La construction d'une configuration initiale se fait en deux étapes (voir figure A15).

Dans la première étape, les machines sont affectées aléatoirement aux sites en tenant compte des contraintes (i) et (iii). Dans la seconde étape, les machines sont rapprochées le plus possible les unes des autres en respectant les contraintes (ii) et (iv). Ces deux opérations sont suivies par le calcul des distances entre les machines. Cela permet de calculer la valeur du critère correspondant à cette solution initiale. Elle est donnée par l'expression (16).

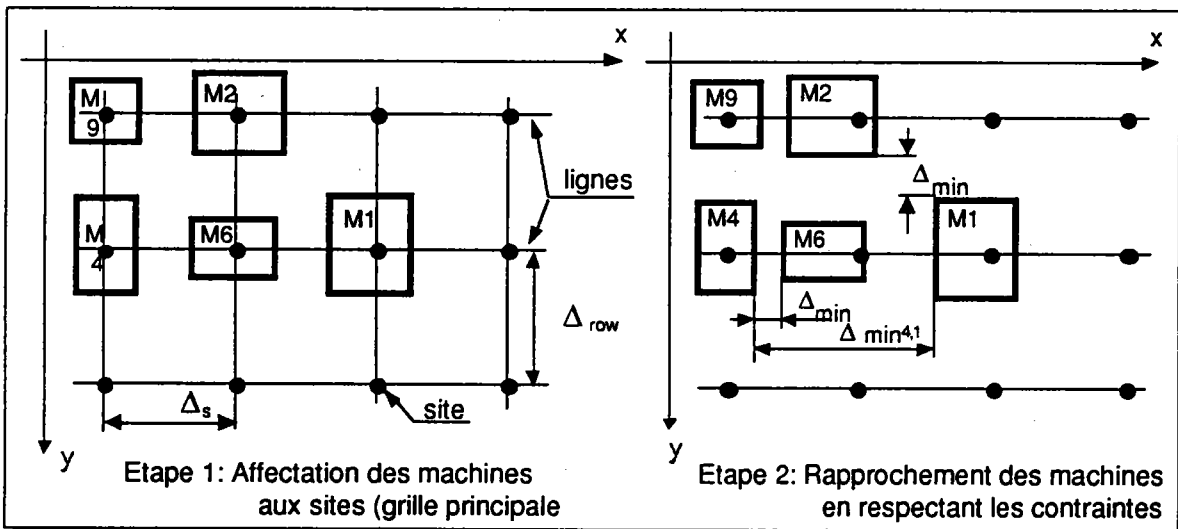


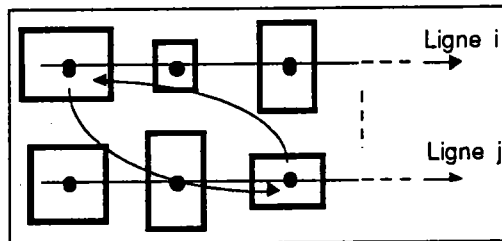
Figure A15 : Construction d'une configuration initiale

2.4.3.3. Transformations élémentaires

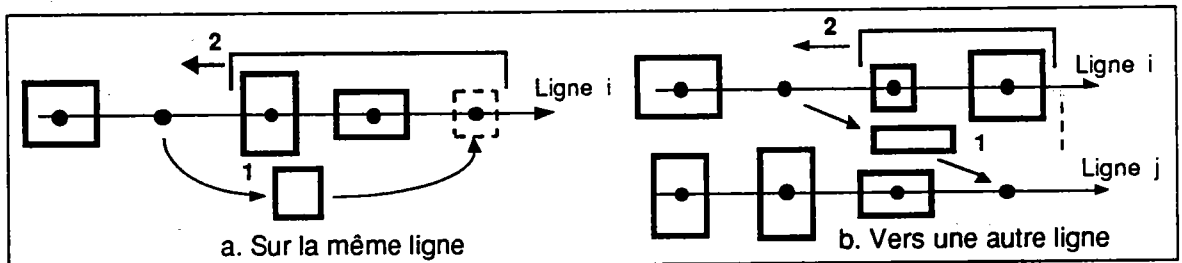
Deux opérations se succèdent dans toute transformation élémentaire. Dans la première, on modifie les séquences des machines sur les lignes de la grille principale (où les sites d'une même ligne sont séparés par la distance Δ_s , et les lignes adjacentes par la distance Δ_{row}) (voir figure A15, Etape 1), en

réalisant une des transformations élémentaires suivantes, choisie aléatoirement :

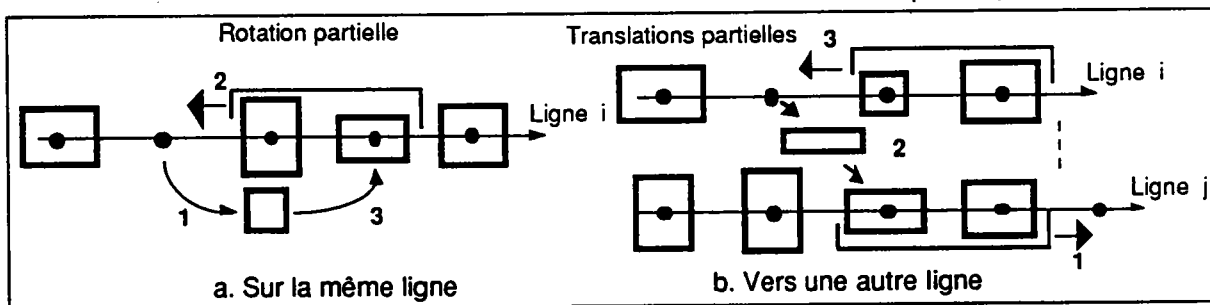
- 1 • Permutation de deux machines choisies aléatoirement (figure A16.1),
- 2 • Transfert d'une machine vers un site libre de la même ligne ou d'une autre ligne (la machine et la ligne sont choisis aléatoirement) (figure A16.2),
- 3 • Transfert d'une machine vers un site occupé (la machine et le site sont choisis aléatoirement) (figure A16.3) :
 - par permutation circulaire, si l'ancien et le nouveau site sont sur la même ligne,
 - par translation, si les deux sites sont sur des lignes différentes.



A16.1 : Permutation de deux machines



A16.2 : Transfert d'une machine vers un site libre



A16.3 : Transfert d'une machine vers un site occupé

Figure A16 : Transformations élémentaires

Dans la deuxième étape, les machines sont rapprochées les unes des autres en respectant les contraintes (ii) et (iv) (voir figure A15, étape 2). Les distances entre les machines sont calculées à l'aide de (17) et sont utilisées pour le calcul de la valeur du critère à l'aide de (16).

2.4.3.4. Dimensionnement de la cellule

Dans le module d'agencement inter-cellulaire (voir partie B), les cellules sont supposées être de forme rectangulaire. Les dimensions d'une cellule sont donc celles du plus petit rectangle pouvant la contenir, auquel on ajoute une bande de sécurité comme le montre la figure A17.

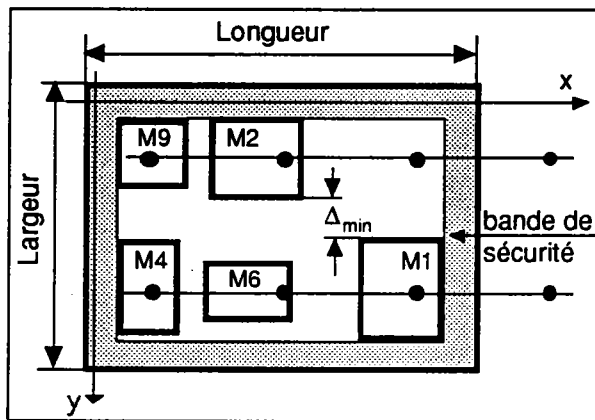


Figure A17 : Dimensions d'une cellule

2.4.3.5. Localisation des entrées/sorties

Nous proposons deux approches différentes pour la localisation des entrées/sorties de la cellule. Chacune d'elles est dédiée à un cas bien précis :

a. Première approche

Cette première approche est appliquée lorsque les postes d'entrée et de sortie de la cellule sont connus. Ils sont alors considérés comme des machines et sont intégrés aux routages des produits (en début du routage pour l'entrée et en fin de routage pour la sortie). On calcule alors la nouvelle matrice des flux inter-machines, et on résout ce problème comme le précédent en minimisant le critère formulé par (16) en tenant compte des contraintes du problème.

b. Deuxième approche

La seconde approche est utilisée lorsque les entrées/sorties doivent être affectées à deux des sites préalablement choisis. Une première étape consiste, dans ce cas, à résoudre le problème d'affectation des machines sans tenir

compte des entrées/sorties de la cellule. Puis, une deuxième étape consiste à affecter les entrées/sorties aux sites possibles dans le but de minimiser le critère suivant :

$$\sum_{k=1}^m \left[\sum_{l=1}^{Ne} (F_{E_i,k} \times d_{E_i,k}) + \sum_{j=1}^{Ns} (F_{k,S_j} \times d_{k,S_j}) \right] \quad (23)$$

où $F_{E_i,k}$ (resp. $d_{E_i,k}$) est le flux (resp. la distance) entre la i ème entrée de la cellule et celle de la machine k , et F_{k,S_j} (resp. d_{k,S_j}) est le flux (resp. la distance) entre la sortie de la machine k et la j ème sortie de la cellule.

3. CONCLUSION DE LA PARTIE A

Cette partie A a été consacrée à la présentation du problème d'agencement des systèmes cellulaires de production. Nous avons décrit la procédure générale de résolution de ce problème. Nous avons ensuite expliqué assez brièvement les deux premiers modules de la procédure : le pré-traitement des données, la conception des cellules. Nous avons détaillé le module d'agencement intra-cellulaire en insistant particulièrement sur le problème d'affectation des ressources aux sites disposés sur la configuration de base choisie pour la cellule. Ce problème est combinatoire, nous avons utilisé la méthode du recuit simulé pour le résoudre. Les résultats obtenus sont de bonne qualité comparés à ceux donnés par d'autres méthodes.

Les problèmes rencontrés dans les modules de conception des cellules et d'agencement intra-cellulaire sont des problèmes combinatoires. L'algorithme du recuit simulé a été utilisé pour les résoudre et a donné des résultats de bonne qualité. Des exemples numériques pour chacun de ces deux modules sont donnés dans l'annexe E.

Nous consacrons la partie B de la thèse au quatrième module qui est l'agencement des cellules dans l'atelier de fabrication. Ce module constitue le sujet principal de cette thèse.

PARTIE **B**

1. Introduction

2. Agencement inter-cellules

2.1. Composantes du problème

2.2. Position du problème

2.3. Solution proposée

2.4. Calcul des plus courts chemins

2.5. Conclusion

3. Introduction à l'agencement dynamique des systèmes de fabrication

3.1. Introduction

3.2. Etat-de-l'art

3.3. Agencement dynamique mono-périodique

3.4. Agencement dynamique multi-périodique

3.5. Conclusion

4. Conclusion

1. INTRODUCTION

Cette partie est consacrée essentiellement à l'agencement des cellules dans l'atelier de fabrication. Nous introduisons ensuite le problème de l'agencement dynamique des systèmes de fabrication.

Dans le problème de l'agencement inter-cellulaire, nous disposons de cellules construites par les modules précédents et de l'atelier de fabrication. Un bon agencement d'un système de fabrication doit atteindre les objectifs suivants :

- une manutention minimale,
- un trafic minimal,
- une accessibilité maximale (les produits doivent pouvoir atteindre toutes les cellules),
- et une sécurité maximale (la sécurité des hommes, des ressources et des produits).

La présence de plusieurs critères, enlève sa signification à la notion d'optimalité. Nous cherchons donc un critère unique qui conduit à un trafic et à une manutention minimaux, les autres critères étant introduits dans le modèle sous forme de contraintes.

Le problème tel que nous le modélisons est très facilement généralisable à des domaines autres que les systèmes de production (voir le paragraphe 1.1 de la partie A). On peut également étendre le problème en tenant compte de l'évolution dynamique du système de production en cherchant un agencement du système qui tienne compte des aléas et des fluctuations des paramètres de fonctionnement et des contraintes.

Nous présentons, dans le paragraphe suivant, le problème statique d'agencement des cellules dans un atelier de fabrication. Nous considérons, pour cela, des paramètres de fonctionnement constants relatifs à un scénario de demande donné imposé au système de fabrication. Ce scénario étant valable sur une période donnée de l'horizon de planification. Nous proposons, dans ce qui suit, un modèle mathématique et une représentation géométrique associée, suivi d'une méthode de résolution adaptée. Dans le chapitre 3 de cette partie,

nous généralisons le problème d'agencement aux systèmes dynamiques de production. Nous y montrons comment l'utilisation des solutions du problème d'agencement statique permet elle de résoudre le problème d'agencement dynamique.

2. AGENCEMENT INTER-CELLULAIRE

2.1. Composantes du problème

Deux aspects sont à prendre en considération lors de l'agencement d'un système de fabrication, à savoir la géométrie du système et les paramètres qui gouvernent son fonctionnement.

a. La géométrie du système comporte :

1. la géométrie de l'atelier proprement dit, c'est-à-dire les limites de sa surface, les obstacles physiques qui s'y trouvent (murs, sources d'énergie, zones interdites, ...), ainsi que les positions de ses entrées/sorties,

2. la géométrie des cellules (ou des ressources en général) à placer ou à replacer, c'est-à-dire leurs dimensions et les positions de leurs entrées/sorties. Les stocks inter-cellulaires, les zones de stationnement des transporteurs (VAG), les magasins d'outils et même une simple machine peuvent être considérés comme des cellules.

Notons que toutes les composantes de notre système ont une forme rectangulaire. La figure B1 résume les aspects de la géométrie, les détails seront introduits par la suite.

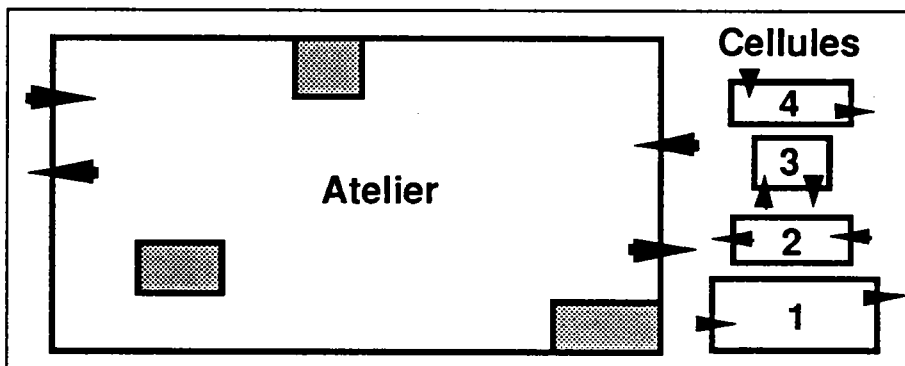


Figure B1 : L'atelier et les cellules à agencer

b. Le fonctionnement du système de production est caractérisé par les routages des produits, leur ratio moyen de production, les tailles des lots, et les capacités des ressources. Ces données ajoutées à la composition des cellules, nous permettent de calculer la matrice des flux circulant dans l'atelier.

Dans la suite, nous posons le problème et nous donnons les éléments de la méthode de résolution utilisée.

2.2. Position du problème

Le problème de l'agencement inter-cellulaire consiste à trouver la position d'un point donné de chaque cellule et l'orientation de celle-ci sur la surface libre de l'atelier. Cette opération est réalisée par affectant les cellules à des zones libres de l'atelier (voir figure A9, partie A). L'objectif est de minimiser la somme des produits (Flux x distance) entre les couples (sortie d'une cellule, entrée d'une autre cellule) en tenant compte des contraintes physiques.

Notons les deux hypothèses suivants :

- Une cellule de fabrication peut généralement avoir plusieurs entrées et plusieurs sorties, ceci peut parfaitement être pris en compte dans la formulation du problème. La seule chose qui change pour cela dans les données est la taille de la matrice des flux (nombre de points de départ et d'arrivée des produits) connaissant pour chaque produit son point d'entrée et son point de sortie dans la cellule. Afin de simplifier la présentation du problème et de sa méthode de résolution, nous supposons que les cellules disposent d'une seule entrée et d'une seule sortie, sans aucune risque de restriction.
- Nous supposons également que toute entrée (resp. sortie) de l'atelier est considérée comme une sortie (resp. entrée) de cellule.

2.2.1. Fonction objectif

Une formulation de ce critère est la suivante :

$$\text{Minimiser } \sum_{i=1}^{N_d} \sum_{\substack{j=1 \\ j \neq i}}^{N_a} T(i, j) \times D(i, j). \quad (24)$$

où :

N_d : nombre de sorties de cellule,

N_a : nombre d'entrées de cellule,

$T(i, j)$: coût du flux de matière par unité de distance entre la sortie de cellule i et l'entrée de cellule j ,

$$T(i, j) = \sum_{k=1}^p \left(c_k \times r_k \times \frac{V_{k,i,j}}{U_{k,i,j}} \right) \quad (25)$$

où :

c_k : coût de transport d'une pièce de type P_k par unité de distance,

r_k : ratio des produits du type P_k à produire,

$V_{k,i,j}$: nombre de fois où l'entrée de cellule j suit la sortie de cellule i dans le routage du produit P_k ,

$U_{k,i,j}$: taille du lot du produit du type P_k transporté entre la sortie de cellule i et l'entrée de cellule j .

$D(i, j)$: distance couverte par le transporteur entre la sortie de cellule i et l'entrée de cellule j . Cette distance dépend du type de transporteur utilisé.

2.2.2. Contraintes

Cette fonction objectif est soumise à deux types de contraintes ; les contraintes fortes qui ne peuvent être violées et les contraintes molles qui, si elles sont violées, engendrent une pénalisation de la valeur du critère.

2.2.2.1. Contraintes fortes

Elles assurent :

- l'accessibilité des entrées/sorties des cellules et par conséquent le non chevauchement entre les objets se trouvant dans l'atelier.
- le positionnement définitif de certaines cellules dans des endroits bien définis de l'atelier.

2.2.2.2. Contraintes molles

Ces contraintes tendent à préserver une distance minimale ou maximale entre certains couples de cellules, selon qu'il s'agisse de les éloigner ou de les rapprocher. Afin d'intégrer ces contraintes dans le problème, nous ajoutons dans la fonction objectif un facteur de pénalisation qui devient non nul chaque fois qu'un non respect d'au moins une de ces contraintes est décelé.

Ce facteur dépend d'une part du type de la contrainte (rapprochement ou éloignement) et de l'importance de sa violation (c'est-à-dire de combien la distance entre les deux cellules est supérieure au maximum fixé pour le rapprochement ou inférieure au minimum fixé pour l'éloignement). Ceci est représenté par la variable suivante :

$$DC(\pi_i, \pi_j) = \begin{cases} D_{\min_{i,j}} - D(\pi_i, \pi_j) & \text{si les cellules } i \text{ et } j \text{ doivent être} \\ & \text{éloignées et } D(\pi_i, \pi_j) < D_{\min_{i,j}}, \\ D(\pi_i, \pi_j) - D_{\max_{i,j}} & \text{si les cellules } i \text{ et } j \text{ doivent être} \\ & \text{rapprochées et } D(\pi_i, \pi_j) > D_{\max_{i,j}}, \\ 0 & \text{sinon.} \end{cases} \quad (26)$$

où : $D_{\min_{i,j}}$ (resp. $D_{\max_{i,j}}$) représente la distance minimale (resp. maximale) désirée entre les cellules C_i et C_j .

π_i et π_j appartiennent respectivement aux cellules C_i et C_j , tels que :

$$D(\pi_i, \pi_j) = \underset{\substack{A \in C_i \\ B \in C_j}}{\text{Min}} D(A, B) \quad (27)$$

Le facteur de pénalisation dépend, d'autre part, de l'importance de la contrainte que nous représentons par le terme suivant :

$$TC(i, j) = \begin{cases} \lambda_{i,j} \times \max_{\substack{k=1, 2, \dots, N_d \\ l=1, 2, \dots, N_s}} (T(k, l)) & \text{si } DC(\pi_i, \pi_j) \neq 0, \\ 0 & \text{sinon.} \end{cases} \quad (28)$$

où :

$\lambda_{i,j}$ est un poids qui sert à représenter l'importance donnée à la contrainte. Ses valeurs sont prises dans le tableau 1 (partie A de la thèse).

Ce facteur de pénalisation, noté $PF(i, j)$, est défini pour tout couple de cellules. $\lambda_{i,j}$ est nul lorsque les cellules C_i et C_j ne sont soumises à aucune contrainte de proximité.

La valeur de $PF(i, j)$ est donnée par :

$$PF(i, j) = TC(i, j) \times DC(\pi_i, \pi_j) \quad (29)$$

La nouvelle expression de la fonction objectif est alors :

$$\text{Minimiser } \sum_{i=1}^{Nd} \sum_{\substack{j=1 \\ j \neq i}}^{Na} [T(i, j) \times D(i, j) + PF(i, j)] \quad (30)$$

2.3. Solution proposée

La résolution du problème se fait par améliorations successives en passant par une séquence de modifications de l'agencement de l'atelier. La réalisation de ces modifications nécessite une préparation des données géométriques que nous expliquons dans le paragraphe suivant. Partant de cette représentation géométrique du système, nous montrons comment s'applique l'algorithme de résolution.

2.3.1. Représentation géométrique du système

Dans le but de discrétiser l'espace géométrique de l'atelier, nous recouvrons l'atelier (voir figure B5) par une grille (voir figure B6) dont la maille élémentaire est un carré de dimensions ($u \times u$). Cette unité de longueur (u) est choisie en fonction du niveau de précision désiré des solutions et du temps de calcul à investir. En effet, plus l'unité u est petite, plus les solutions sont précises, mais plus le temps de traitement est long. Il est donc nécessaire de trouver un compromis. Nous proposons, pour cela, l'alternative suivante qui suppose que le temps de traitement n'est pas un souci pour l'utilisateur :

En supposant que les dimensions des objets (y compris celles de l'atelier) sont des nombres rationnels que l'on peut écrire sous forme de fractions. Nous réduisons celles-ci au même dénominateur. u est alors le rapport entre le pgcd des numérateurs divisé par le dénominateur commun. Nous approchons ainsi tous les éléments du système (atelier, obstacles et cellules) par des rectangles de dimensions supérieures aux dimensions réelles de ces éléments et multiples de l'unité de longueur u .

2.3.1.1. Re-dimensionnement de l'atelier

L'atelier vide est initialement composé d'une forme externe, rectangulaire ou pas, et par des obstacles internes fixes. Ces obstacles sont généralement des

murs, des trous dans le sol, des poutres, des armoires électriques, etc... . Le re-dimensionnement de l'atelier vide se fait comme suit :

- inscrire l'atelier dans le plus petit rectangle pouvant le contenir dont les dimensions (n_c , n_l) sont des multiples de u ($n_c \cdot u \geq D_x$ et $n_l \cdot u \geq D_y$) (voir figure B2),
- tout obstacle se trouvant à l'intérieur de l'atelier est inscrit dans le plus petit rectangle pouvant le contenir ayant des dimensions multiples de u (voir figure B2),
- toutes les zones ne faisant pas partie de l'atelier et se trouvant totalement ou partiellement dans le rectangle de l'atelier sont transformées en obstacles rectangulaires dont les dimensions sont également des multiples de u .

Dans la figure B2 l'atelier réel est représenté en traits forts et le résultat du re-dimensionnement est représenté en traits fins.

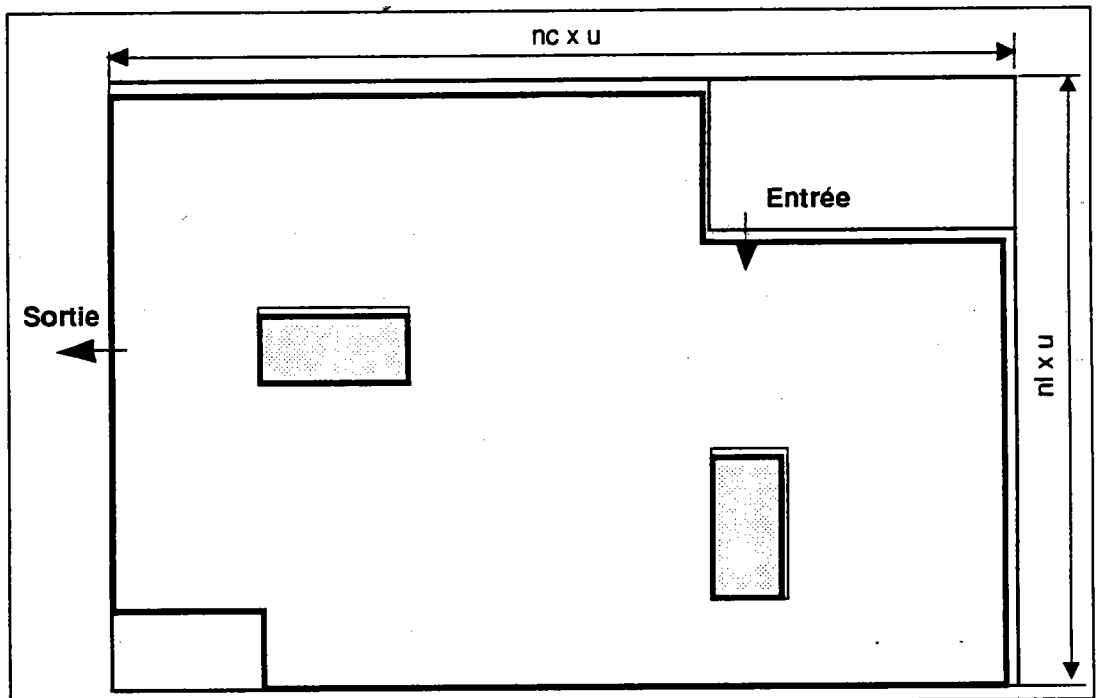


Figure B2 : Re-dimensionnement de l'atelier vide

2.3.1.2. Re-dimensionnement des cellules

Dans la plupart des cas, les cellules ont une forme quelconque non rectangulaire. Elles peuvent cependant, de la même manière que pour l'atelier, être approchées par des rectangles de dimensions supérieures à leurs dimensions réelles et multiples de u (voir figure B3).

Dans certains cas, on peut avoir à ajouter à la surface de la cellule une surface supplémentaire. Dans l'exemple de la figure B4, la cellule reçoit à l'entrée des pièces brutes longues qui doivent être introduites en longueur dans la machine de découpage. Il est donc nécessaire de prévoir une place supplémentaire permettant l'accès de ces pièces. Cet espace est utilisé aussi pour le stockage de ces produits bruts.

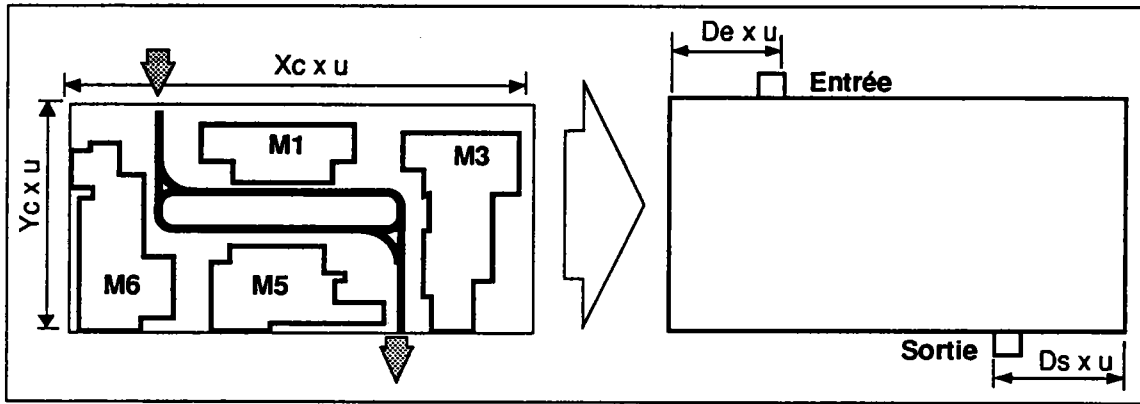


Figure B3 : Re-dimensionnement d'une cellule

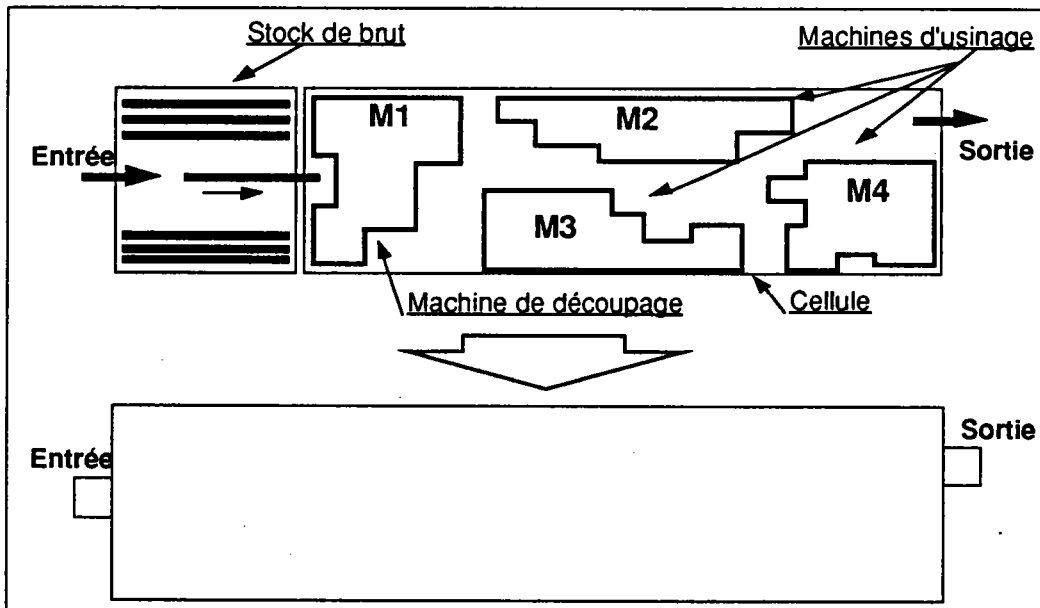


Figure B4 : Cas particulier de dimensionnement de cellule

2.3.1.3. Grille de l'atelier et matrice associée

Finalement, les configurations de l'atelier, telles que celle de la figure B5, sont remplacées par une grille rectangulaire où les obstacles (cellules et zones interdites) sont des rectangles dont les mailles sont grisées (voir figure B6).

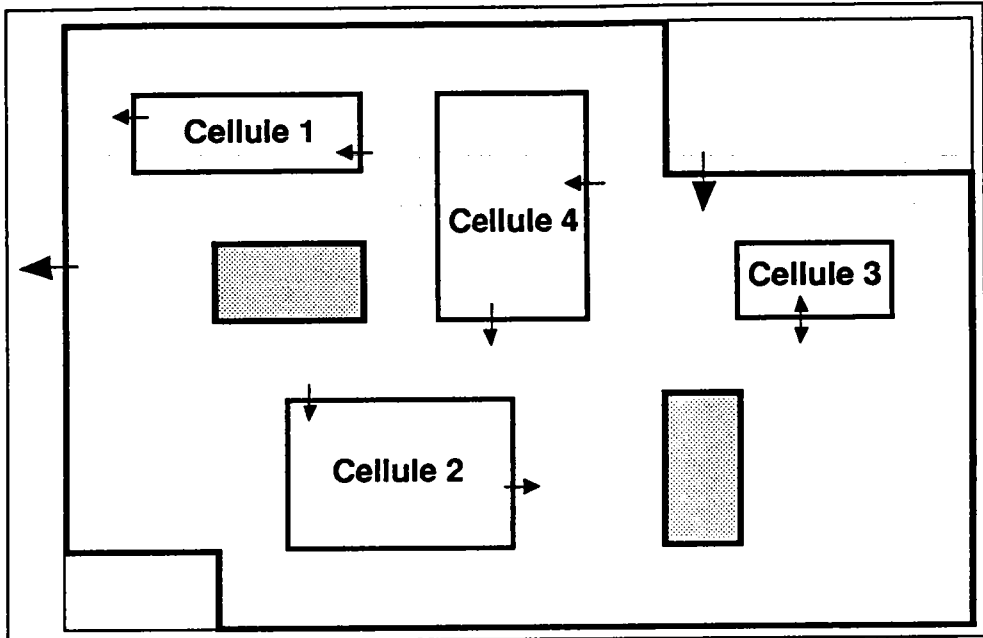


Figure B5 : Atelier de fabrication cellulaire

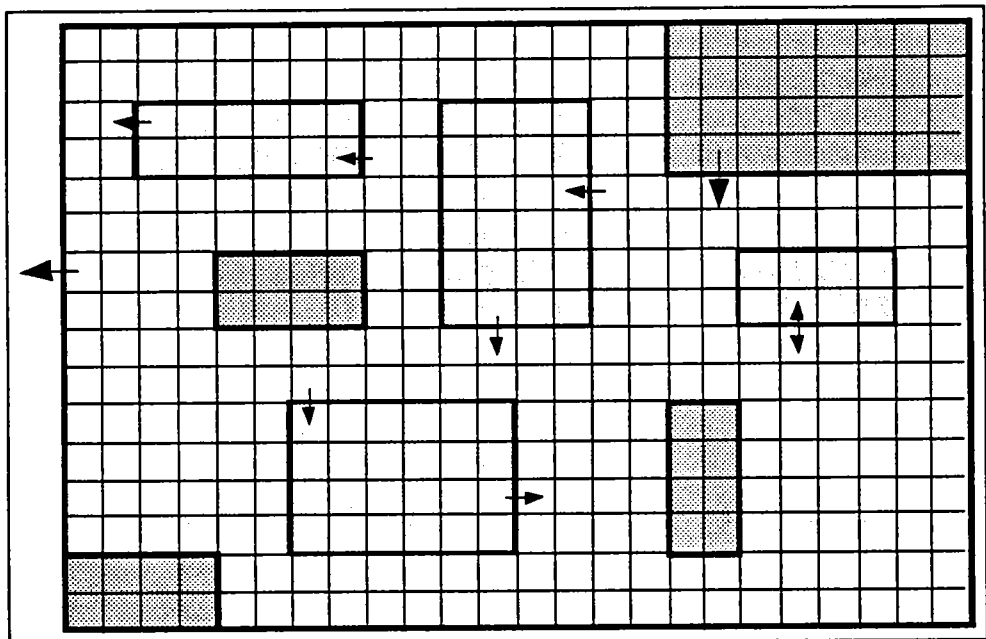


Figure B6 : Grille recouvrant l'atelier de fabrication

Cette grille est représentée dans le programme informatique par une matrice $(n_l \times n_c)$ notée [MATEL]. Chaque élément de [MATEL] représente une maille de la grille recouvrant l'atelier. Les mailles grisées dans la figure B6, soit celles qui correspondent aux obstacles, sont représentées dans la matrice (voir figure B7) par un (1) et les autres par un (0).

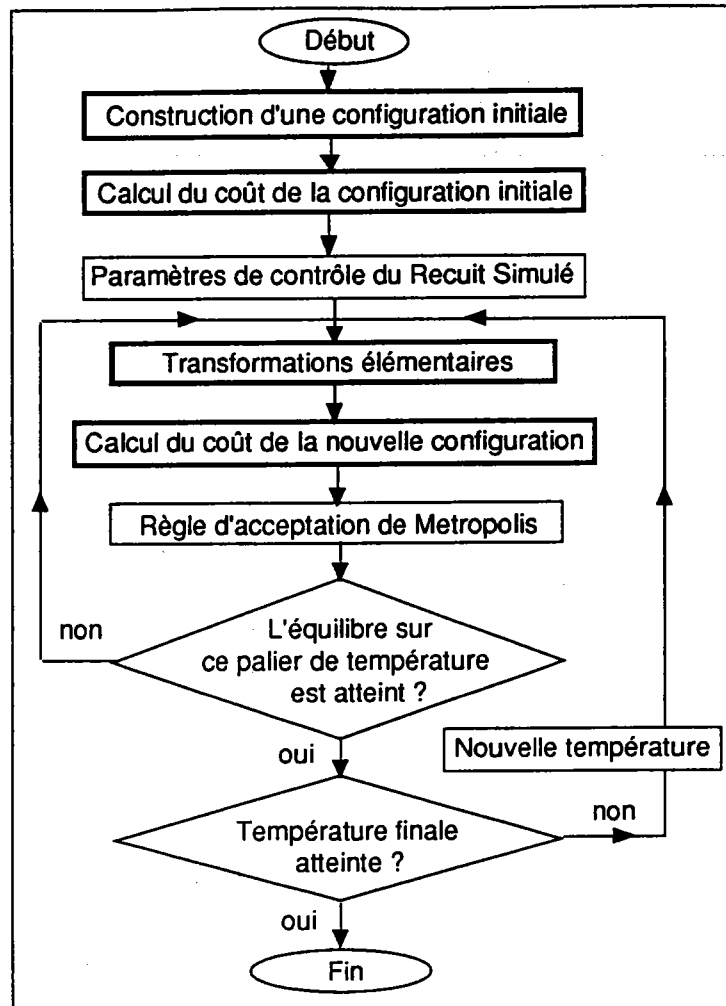


Figure B8 : Organigramme de l'algorithme du recuit simulé appliqué à l'agencement inter-cellulaire

2.3.2.1. Construction d'une configuration initiale

Sauf dans des cas particuliers, qui sont rares dans la réalité et que nous citerons par la suite, les résultats finaux donnés par l'algorithme du RS ont la particularité de ne pas être trop affectés par la qualité de la solution initiale si l'on effectue un nombre suffisant d'itérations. Il ne s'agit donc pas de construire une solution initiale de très bonne qualité mais d'en construire une qui soit admissible. Le problème de la construction d'une configuration initiale peut être difficile. Sa difficulté augmente lorsque la différence entre la surface libre de l'atelier et la somme des surfaces des cellules à installer diminue. Ce problème devient à ce moment là proche des problèmes de découpage dans le plan ou de remplissage bi-dimensionnel de conteneurs. Ces problèmes sont NP-complets (Baker et coll., 1980 ; Berkey et Wang, 1987). La difficulté augmente également

avec la surface et la dispersion des zones interdites et des cellules déjà installées dans l'atelier en cours de remplissage.

La construction de la solution initiale peut se faire de trois manières différentes. Elle peut être : a) totalement donnée par l'utilisateur, b) construite par le système avec la possibilité d'intervention de l'utilisateur, ou c) construite entièrement par le système.

Après l'acquisition des données et le re-dimensionnement des éléments du système, et avant toute entreprise de construction d'une solution initiale, il faut effectuer des tests d'admissibilité. Ces tests sont nécessaires mais non suffisants.

1. On vérifie d'abord si la surface libre de l'atelier est supérieure à la somme des surfaces des cellules. Il est clair que l'agencement sera d'autant plus aisé que le premier membre de l'inégalité (31) ci-dessous sera important. En d'autres termes, on vérifie si l'inégalité suivante est bien respectée :

$$\Delta S = S_{\text{atel.}} - \left(\sum_{z=1}^{N_z} S_{z_o.}(z) + \sum_{c=1}^{N_c} S_{c_e.}(c) \right) > 0 \quad (31)$$

où : $S_{\text{atel.}}$: surface de l'atelier vide (avec les zones interdites),
 $S_{z_o.}(z)$: surface de la zone interdite d'indice z ,
 $S_{c_e.}(z)$: surface de la cellule d'indice c ,
 N_c : nombre de cellules,
 N_z : nombre de zones interdites.

2. Nous vérifions ensuite que chacune des cellules a au moins une chance d'être placée dans l'atelier. Pour se faire nous vérifions s'il existe au moins une zone libre rectangulaire de l'atelier vide pouvant recevoir la cellule en question (nous définissons ces zones en annexe C).

Nous proposons une heuristique simple pour la construction d'une solution initiale :

a. Dans le cas où elle est totalement construite par le système, l'algorithme est le suivant :

-
1. Classer les cellules dans l'ordre décroissant de leur surface.
 2. Pour chaque cellule (prise dans l'ordre ci-dessus) faire :
 - Trouver toutes les **zones libres*** rectangulaires pouvant la contenir,
 - Installer la cellule dans la zone libre qui a la surface la plus proche de la sienne
- Fin faire.
-

b. Dans le cas où la possibilité d'intervenir est permise à l'utilisateur, le système lui demandera son avis après avoir choisi la zone libre pouvant contenir la cellule et avant l'installation de cette dernière. Au cas où l'utilisateur estime que la zone n'est pas convenable, il choisira une autre zone et installera la cellule lui même.

* **zones libres** : Nous avons évoqué dans l'algorithme ci-dessus le terme "zones libres" rectangulaires que nous avons pas défini. Sa définition et la méthode utilisée pour les trouver sont données dans l'annexe C.

Nous tenons enfin à préciser que dans les cas où ΔS est proche de 0, la configuration initiale peut être décisive dans la recherche de l'optimum. Par exemple, la figure B24 présente la configuration initiale d'un atelier où le flux entre les cellules A et B est bien supérieur aux autres flux. Dans ce cas, il est nécessaire de mettre les cellules A et B côte-à-côte pour pouvoir espérer trouver la configuration optimale. En d'autres termes, il faudrait permuter un ensemble de cellules (dont le nombre d'éléments est à déterminer) avec une cellule. Dans le cas présenté, il faut permuter {C, D} avec B ou {C, D} avec A. Le fait d'envisager de tels permutations augmenterait d'avantage la complexité du problème. Notre méthode réalise des permutations entre deux cellules uniquement.

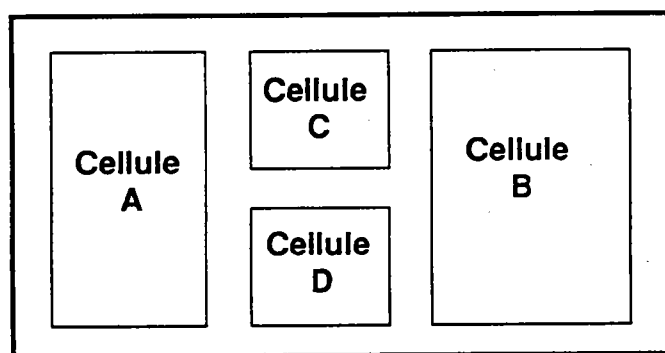


Figure B24 : Cas où l'optimum n'est jamais atteignable en utilisant la méthode proposée.

Bien que cette situation ne se rencontre que rarement dans la réalité, un moyen d'y remédier est de reprendre plusieurs fois le processus du recuit simulé, en partant à chaque fois d'une configuration initiale différente.

2.3.2.2. Passage d'une configuration à une configuration voisine

Dans le recuit physique, le changement d'état se fait d'une manière naturelle qualifiée d'aléatoire et de continue (les états intermédiaires entre l'état liquide et l'état solide sont innombrables et les valeurs de l'énergie libre entre deux états consécutifs sont très voisines).

Dans notre application, l'évolution du processus de recherche d'une solution optimale est discontinue et étant donné que le nombre d'états (ou configurations) est très grand, le nombre d'états que nous pourrions produire sera relativement petit. Néanmoins, l'aspect aléatoire du passage d'une configuration à une autre subsiste.

Nous disposons de quatre transformations élémentaires permettant de passer d'une configuration à une autre. L'une de ces transformations est choisie aléatoirement à chaque itération du processus du recuit simulé. Ces quatre types de transformations élémentaires sont : la permutation de deux cellules, le déplacement d'une cellule vers une zone libre de l'atelier (ou mutation), la translation d'une cellule dans son propre entourage et la rotation d'une cellule dans son entourage.

Avant de passer à l'explication détaillée de chacune de ces transformations et de la manière dont elle sont réalisées, signalons que pour faire en sorte que ce processus soit complètement aléatoire, la transformation est tirée aléatoirement et de manière équiprobable parmi les quatre transformations possibles. Pour chaque transformation, la ou les cellules sont également tirées aléatoirement et d'une manière équiprobable parmi toutes les cellules pouvant être déplacées (c'est à dire celles qui ne sont pas soumises à une contrainte d'affectation à une position fixée).

Chacune des quatre transformations se déroule en deux étapes : après le tirage aléatoire d'une ou de deux cellules, le système vérifie la possibilité de réalisation de la transformation en question et la réalise si possible. Nous expliquons dans la suite la manière dont chacune de ces transformations est effectuée.

A. Permutation de deux cellules

Après avoir choisi aléatoirement les deux cellules à permuter, le système réalise les deux opérations suivantes :

A.1. Test d'admissibilité

Dans cette première phase on vérifie si l'entourage de chacune des deux cellules peut contenir l'autre cellule. Pour cela, il faut commencer par la détermination de ces entourages. La figure B9 nous montre un exemple d'entourage de cellule. La manière dont ces entourages sont déterminés sera expliquée dans l'annexe C.

Soient C_i et C_j les cellules à permuter et EC_i et EC_j leurs entourages respectifs. Nous définissons les distances et les dimensions suivantes :

- DI est la distance minimale séparant deux objets quelconques.
- $dx(i)$ (resp. $dy(i)$) est la dimension de la cellule i suivant l'axe x (resp. y).
- $dex(i)$ (resp. $dey(i)$) est la dimension de l'entourage de la cellule (i) suivant l'axe x (resp. y).

La permutation est possible, si les quatre inégalités suivantes sont respectées :

1. $\max(dx(i), dy(i)) + 2 \cdot DI \leq \max(dex(j), dey(j))$,
2. $\min(dx(i), dy(i)) + 2 \cdot DI \leq \min(dex(j), dey(j))$,
3. $\max(dx(j), dy(j)) + 2 \cdot DI \leq \max(dex(i), dey(i))$,
4. $\min(dx(j), dy(j)) + 2 \cdot DI \leq \min(dex(i), dey(i))$.

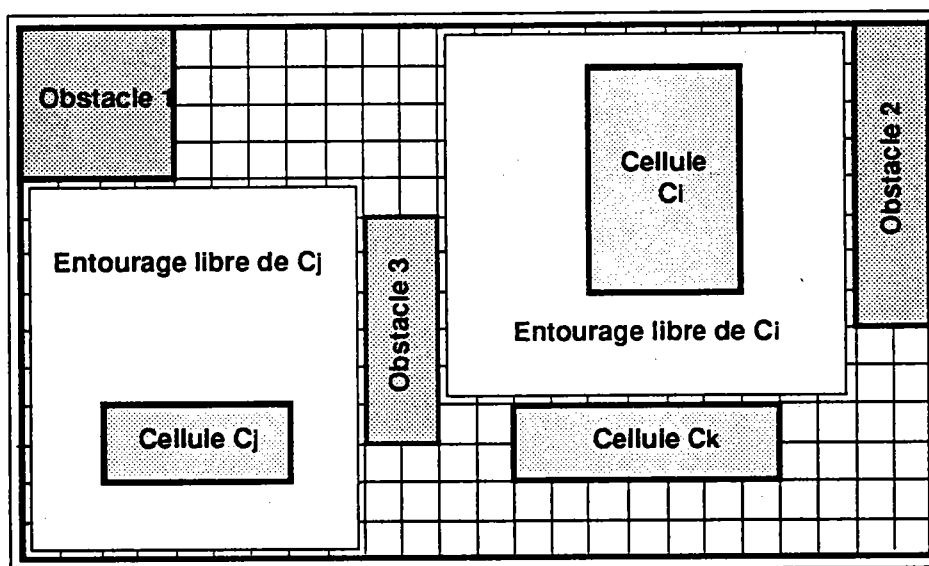


Figure B9 : Exemple d'entourage libre de cellule

L'expression 1 (resp. 2) exprime le fait que la longueur (resp. la largeur) de la cellule i à laquelle nous ajoutons de part et d'autre la largeur DI de l'entourage libre minimum doit être plus petite que la longueur (resp. la largeur) de l'entourage libre de la cellule j . Les expressions 3 et 4 expriment la même chose pour la cellule j et l'entourage de la cellule i .

A.2. Réalisation de la permutation

Si les inégalités ci-dessus sont vérifiées, le système passe à la phase de réalisation de la permutation. Nous proposons un algorithme permettant de réaliser cela (lire l'algorithme tout en suivant sur la figure B9) :

1. Détermination des entourages libres respectifs CE_i et CE_j des deux cellules C_i et C_j .
 2. Détermination de l'orientation des deux cellules C_i et C_j par rapport à l'axe x (voir figure B11).
 3. Détermination des orientations $Orien(CE_i)$ et $Orien(CE_j)$ des entourages des cellules (même règle).
 4. Détermination de la position relative des deux entourages parmi les cinq positions possibles de la figure B10.
 5. Positionnement des cellules dans leur nouvel environnement, comme indiqué dans la figure B10.
 6. Re-positionnement de l'entrée et de la sortie de chaque cellule.
- Fin.

L'opération de détermination des entourages libres des cellules est utilisée dans la plupart des transformations élémentaires. Nous lui consacrons l'annexe D.

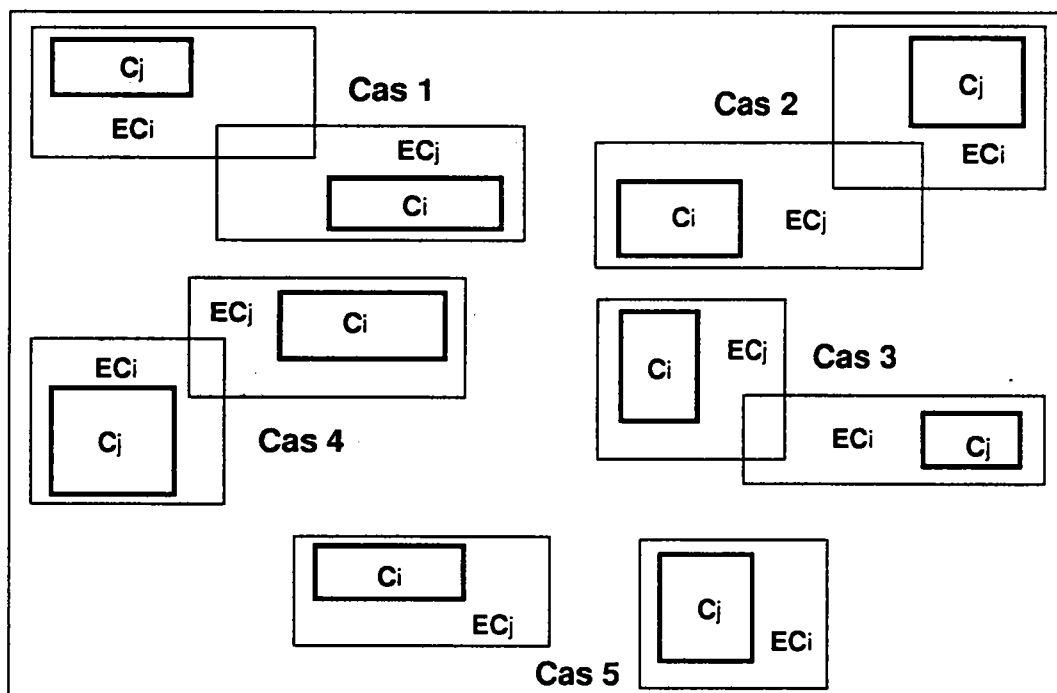


Figure B10 : Les cinq cas de positions relatives entre les entourages des cellules et re-positionnement des cellules après permutation

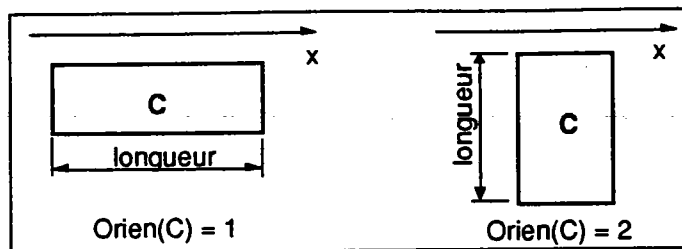


Figure B11 : L'orientation d'un rectangle (cellule ou entourage)

B. Déplacement d'une cellule vers une zone libre de l'atelier

Le système choisit aléatoirement la cellule à déplacer et procède comme suit :

B.1. Test d'admissibilité

Pour que cette opération soit réalisable, il faut que le système arrive à trouver une zone libre rectangulaire de l'atelier qui puisse recevoir la cellule choisie (le processus de détermination d'une zone libre rectangulaire est expliqué ci-dessus). Si le système arrive à trouver plusieurs zones libres, il en choisit une aléatoirement.

Une zone libre rectangulaire ZL de dimensions dx et dzy peut recevoir une cellule C_i de dimensions $dx(i)$ et $dy(i)$ si :

$$\begin{aligned} & \max (dx(i), dy(i)) + 2 \cdot DI \leq \max (dxx, dzy), \\ \text{et} & \quad \min (dx(i), dy(i)) + 2 \cdot DI \leq \min (dxx, dzy). \end{aligned}$$

Ceci exprime le fait que la longueur (resp. la largeur) de la cellule à laquelle nous ajoutons la largeur DI de l'entourage libre minimum doit être plus petite ou égale à la longueur (resp. la largeur) de la zone libre ZL qui la recevra.

B.2. Réalisation du déplacement

La réalisation de cette opération se fait conformément à l'algorithme suivant :

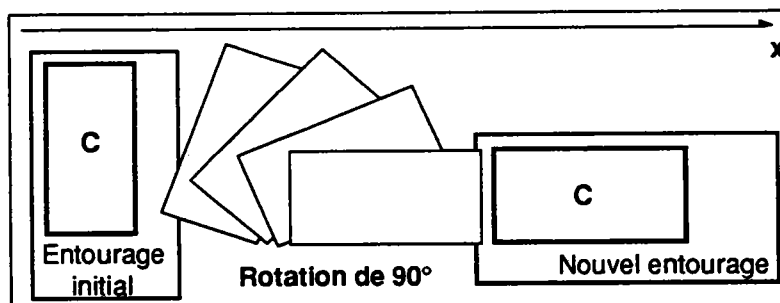


Figure B12 : Ré-orientation de la cellule dans son nouvel entourage

1. Détermination de l'orientation de la cellule (règle sur figure B11).

2. Détermination d'une zone libre rectangulaire pouvant recevoir la cellule :

2.1. détermination de toutes les zones libres de l'atelier,

2.2. sélection de celles qui peuvent recevoir la cellule,

2.3. si (le nombre de ces zones est nul) :

Fin.

sinon : - tirer aléatoirement une de ces zones,

- détermination de l'orientation de cette zone.

3. Installation de la cellule dans son nouvel entourage :

Si (la cellule et son nouvel entourage ont la même orientation) :

Installer la cellule sans changement d'orientation.

Sinon, si (la longueur de la cellule est inférieure à celle de son nouvel entourage) :

Installer la cellule sans changer son d'orientation.

sinon : Installer la cellule après une rotation de 90° comme indiqué sur la figure B12.

4. Re-positionnement de l'entrée et de la sortie de la cellule.

Fin.

C. Translation d'une cellule dans son entourage

Cette opération s'effectue également en deux étapes :

C.1. Test d'admissibilité

Pour vérifier la possibilité de déplacement d'une cellule dans son entourage, il faut d'abord trouver ce dernier. La connaissance de cet entourage nous permettra, en plus, de connaître les différentes directions et les différents sens de translation possibles ainsi que les longueurs des translations. Le test d'admissibilité est le suivant :

En reprenant les mêmes notations que précédemment, nous dirons que la translation de la cellule C_i dans son entourage EC_i est possible si :

$$dx(i) + 2 \cdot DI < dex(i) \text{ et/ou } dy(i) + 2 \cdot DI < dey(i).$$

Cette condition exprime le fait que la dimension de la cellule suivant l'un des deux axes ou suivant les deux, à laquelle on ajoute de part et d'autre la largeur de l'entourage libre minimum, doit être plus petite ou égale à la dimension de son entourage suivant le même axe.

C.2. Réalisation de la translation

Dans son entourage une cellule peut disposer d'une translation horizontale (suivant l'axe des abscisses et dans les deux sens), d'une translation verticale (suivant l'axe des ordonnées et dans les deux sens) ou d'une translation oblique suivant une résultante des deux mouvements précédents. La possibilité de réalisation de l'une ou l'autre de ces translations dépend de la position initiale de la cellule et de son entourage (voir figure B13).

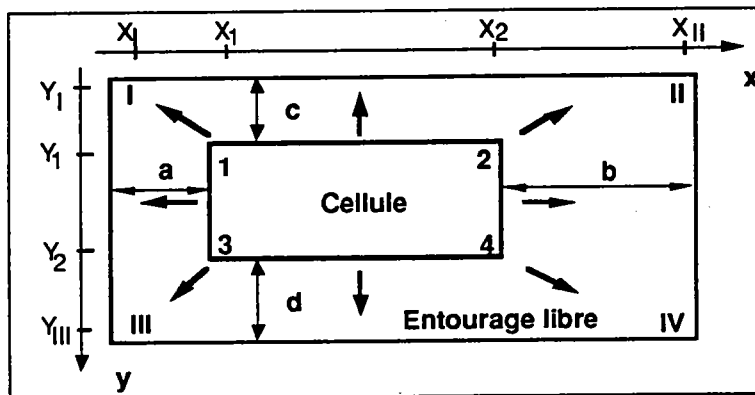


Figure B13 Position de la cellule dans son entourage et possibilités de translation

L'algorithme permettant de trouver les possibilités de translation et de les réaliser, est le suivant :

-
1. Translation horizontale : $a = X_1 - X_I$, $b = X_{II} - X_2$,
 - Si ($a > DI$ et $a > b$) :
 - translation de la cellule vers la gauche de $\Delta x = a - DI$,
 - re-positionnement de l'entrée et de la sortie de la cellule.
 - Sinon, si ($b > DI$ et $b \geq a$) :
 - translation de la cellule vers la droite de $\Delta x = b - DI$,
 - re-positionnement de l'entrée et de la sortie de la cellule.
 - Sinon, la translation horizontale est impossible.
 2. Translation verticale : $c = Y_1 - Y_I$, $d = Y_{III} - Y_2$,
 - Si ($c > DI$ et $c > d$) :
 - translation de la cellule vers le haut de $\Delta y = c - DI$,
 - re-positionnement de l'entrée et de la sortie de la cellule.
 - Sinon, si ($d > DI$ et $c \geq d$) :
 - translation de la cellule vers le bas de $\Delta y = d - DI$,
 - re-positionnement de l'entrée et de la sortie de la cellule.
 - Sinon, la translation verticale est impossible.

Fin.

Si deux translations, l'une verticale et l'autre horizontale, sont réalisables, nous obtenons la translation oblique qui est la résultante des deux.

D. Rotation d'une cellule dans son entourage

Les deux étapes de cette transformation élémentaire sont :

D.1. Teste d'admissibilité

En reprenant les mêmes notations que précédemment, la rotation d'une cellule C_i d'un angle de 90° dans son entourage EC_i est possible si et seulement si l'inégalité suivante est vérifiée :

$$\max (dx(i), dy(i)) + 2 \cdot DI \leq \min (dex(i), dey(i)).$$

Ceci est une expression du fait que lorsqu'on ajoute DI de part et d'autre de la longueur de la cellule, ceci doit être plus petit que la largeur de l'entourage de la cellule.

La rotation de la cellule d'un angle de 90° est suivie d'un re-positionnement de la cellule comme l'indique la figure B14. Ce n'est pas le cas pour la rotation à 180° , puisque la cellule n'est pratiquement pas touchée, seules les positions de l'entrée et de la sortie de la cellule sont modifiées.

D.2. Réalisation de la rotation

Le déroulement de cette transformation est conforme à l'algorithme suivant :

Choix aléatoire de l'angle de rotation de la cellule :

Si (angle de rotation = 90°) :

- Si (rotation est possible) :
 - tourner la cellule de 90° ,
 - re-positionner l'entrée et la sortie de la cellule.
- Sinon :
 - (rotation à 180°) re-positionner l'entrée et la sortie de la cellule en conséquence.

Sinon si (angle de rotation = 180°) :

- re-positionner l'entrée et la sortie de la cellule en conséquence.

Fin.

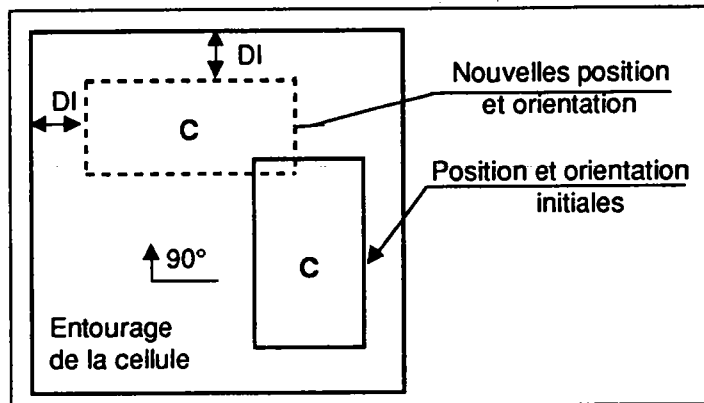


Figure B14 : Rotation d'une cellule dans son entourage

2.3.2.3. Coût d'une configuration

Le coût d'une configuration du système de fabrication (ou la valeur du critère qui lui est associé) est calculé à l'aide de la formule (30), si le système est soumis à des contraintes de proximité. Il peut également être calculé à l'aide de la formule (24) si ce type de contraintes n'existe pas. Les variables du problème, modifiant ce coût lors du passage d'une configuration à une autre, sont les longueurs des plus courts chemins parcourus par le transporteur entre les entrées et les sorties des cellules et les distances entre les cellules lorsque le problème comporte des contraintes de proximité.

Le paragraphe qui suit est consacré au calcul des longueurs des plus courts chemins.

2.4. Calcul des longueurs des plus courts chemins

Le mode de calcul de ces distances dépend exclusivement du type de système de transport utilisé. Par exemple, si on utilise un pont roulant capable de réaliser simultanément les deux mouvements horizontaux, la distance est Euclidienne. Si, par contre, ce pont roulant effectue les deux mouvements consécutivement, nous aurons à calculer une distance de Manhattan. Dans ces deux cas, le calcul des distances est aisé comparé au cas où on utilise un système de Véhicules Auto-Guidé (VAG). Les distances à calculer sont, à ce moment là, celles des plus courts chemins évitant tous les obstacles qui se trouvent entre les points de départ et les points d'arrivée.

Dans ce qui suit nous posons le problème du cheminement. Nous passons en revue certains algorithmes de calcul des plus courts chemins

existants, et enfin nous présentons les deux algorithmes que nous avons utilisés dans le programme d'agencement.

2.4.1. Position du problème

Soit un atelier de fabrication possédant N_e entrées et N_s sorties. Cet atelier contient N_c ressources (cellules, machines, stocks, etc...). Le transport des produits dans l'atelier se fait entre les N_d points de départ (sorties des ressources et entrées de l'atelier) et les N_a points d'arrivée (les entrées des ressources et sorties de l'atelier). En supposant que chaque ressource dispose d'une entrée et d'une sortie, nous avons : $N_d = N_c + N_e$ et $N_a = N_c + N_s$.

Les déplacements des VAGs dans l'atelier sont parallèles aux axes. La figure B15 donne des exemples de chemins à trouver pour une configuration donnée du système de fabrication.

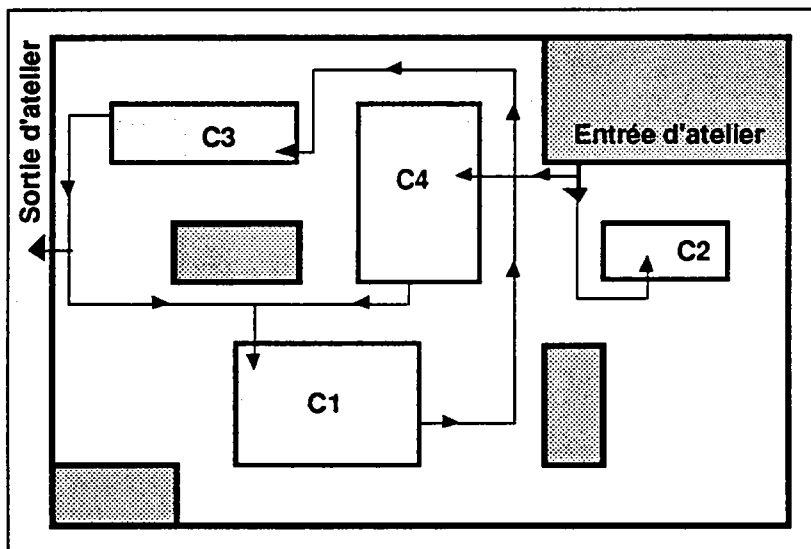


Figure B15 : Des chemins possibles pour une configuration donnée de l'atelier

Notons :

$SC_{i,j}$ l'ensemble des chemins reliant deux points i et j et évitant tous les obstacles.

$d(i, j)$ la longueur du chemin $Ch(i, j) \in SC_{i,j}$.

Le problème consiste à trouver les chemins les plus courts, $Ch^*(i, j)$, entre tous les couples de points $(i, j) \in \{1, 2, \dots, N_d\} \times \{1, 2, \dots, N_a\} / i \neq j$ et $F(i, j) \neq 0$; tels que :

$$d^*(i, j) = \min_{Ch(i,j) \in SC_{i,j}} (d(i, j)). \quad (31)$$

$F(i, j)$ est le flux de matière entre les points i et j .

2.4.2. Etat de l'art

La recherche des plus courts chemins est un problème qui se pose souvent et dans des domaines variés : l'agencement des systèmes de production, le transport, les réseaux de télécommunication, l'informatique, l'électronique, etc. Parmi la multitude d'algorithmes proposés dans la littérature, nous nous contentons de citer les deux algorithmes les plus récents applicables dans notre cas :

- L'algorithme de Moffat et Takaoka proposé en 1987 : C'est un algorithme basé sur la théorie des graphes. Il détermine les plus courts chemins entre tous les couples de points dans un graphe complet avec des arcs pondérés non-négativement. Cet algorithme nécessite un temps de calcul de l'ordre de $O(n^2 \log n)$ en moyenne et $O(n^3)$ dans le pire des cas (n est le nombre de sommets du graphe). Cet algorithme suppose que le graphe a été préalablement construit. La construction de ce graphe revient à établir un diagramme qui entoure tous les obstacles et qui passe par tous les points d'intérêt (points de départ et d'arrivée). Un tel diagramme est appelé "diagramme de Voronoï". Sa construction se fait en un temps proportionnel à $(n \log n)$ (voir le livre de R. Sedgewick "Algorithmes en langage C", chapitre 28).

Une présentation complète de tous les algorithmes de recherche des plus courts chemins dans un graphe se trouve dans le livre de Gondran et Minoux (1990), chapitres 2 et 3.

- L'algorithme de Rezende et coll. (1989) : Cet algorithme trouve le plus court chemin reliant un point donné à tout autre point du plan, en évitant les n obstacles rectangulaires et parallèles aux axes. La complexité de cet algorithme est en $O(n \log n)$ en temps moyen de traitement. Cet algorithme ne se base pas sur un graphe mais sur une représentation géométrique qui dépend du point de départ considéré. Il doit donc être recalculé pour chacun des points de départ, ce qui augmente considérablement le temps de traitement.

2.4.3. Algorithmes proposés

Pour résoudre ce problème, nous proposons deux algorithmes. Le premier calcule le plus court chemin d'un point du graphe vers tous les autres. Le second explore les solutions par séparation et évaluation (voir annexe B). Nous présentons les deux algorithmes que nous évaluons ensuite sur des applications.

2.4.3.1. Algorithme 1

Cet algorithme, comme le suivant, est basé sur une représentation géométrique de base simple qui n'est autre que celle que nous avons illustrée dans le paragraphe 2.3.1.3. de cette partie B. Elle est composée de la grille recouvrant l'atelier et de la matrice binaire associée.

Le but de cet algorithme est de trouver la séquence de points qui correspond au chemin le plus court reliant un point de départ à un point d'arrivée donnés (la figure B16 montre deux exemples de chemins). Etant donné que les obstacles sont rectangulaires et que les déplacements sont parallèles aux axes, les points intermédiaires ne peuvent être que les points voisins des sommets des obstacles à contourner (comme l'indique la figure B17).

L'algorithme commence par localiser tous les points intermédiaires de la configuration donnée. Il construit ensuite un graphe dont les nœuds représentent les points de départ et d'arrivée et les points intermédiaires. Les arcs directs du graphe relient parmi ces points ceux qui correspondent aux couples de sommets

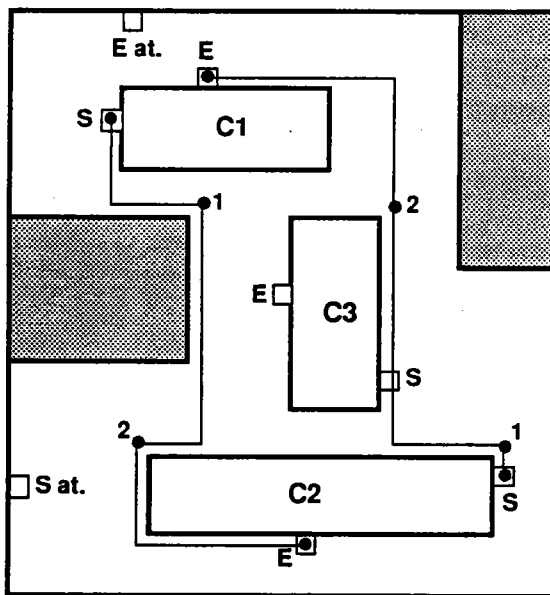


Figure B16 : Exemple de chemins à déterminer

non séparés par un obstacles, dits adjacents. Dans la figure B17, (8, 21), (2, 23), (10, 11), ... sont des couples de points adjacents, contrairement aux couples (8, 1), (22, 16), Il est bien clair que la distance la plus courte entre deux points adjacents i et j est : $d^*(i, j) = |x_i - x_j| + |y_i - y_j|$. Ces distances sont les poids des arcs correspondants. Les éléments de la matrice associée au graphe ont pour valeurs initiales les distances minimales pour les couples de nœuds reliés par

un arc direct et -1 pour les couples dont la distance minimale reste à déterminer. Cette matrice est symétrique.

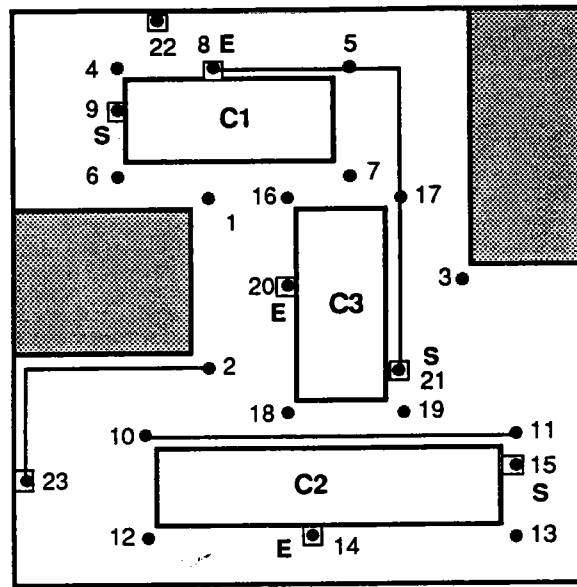


Figure B17 : Points intermédiaires et couples de points adjacents

Nous utilisons dans cette méthode l'algorithme de "Moore-Dijkstra" de recherche des plus courts chemins dans un graphe dont les arcs sont pondérés positivement (Gondran et Minoux, 1990). Cet algorithme calcule la longueur du plus court chemin d'un sommet donné du graphe vers tous les autres. Dans notre application, les sommets de départ sont les sorties des cellules et les entrées de l'atelier. Pour chaque sommet de départ, le temps requis par l'algorithme de Moore-Dijkstra est en $O(N^2)$, dans le pire des cas (N étant le nombre de sommets proportionnel au nombre d'obstacles). Le calcul des longueurs des chemins entre les points d'intérêt prendra un temps en $O(M \times N^2)$ (M étant le nombre de nœuds de départ qui est nettement inférieur à N).

Soient :

$P = \{1, 2, \dots, N\}$ l'ensemble des nœuds du graphe et

$L_{i,j}$ le poids de l'arc (i, j) , s'il existe.

Notons $SP^*(i)$ la longueur du chemin le plus court entre le point de départ 1 et un point i ; avec $SP^*(1) = 0$.

L'algorithme se déroule en $N - 1$ itérations. Au début de chacune d'elles, l'ensemble P est partagé en deux sous-ensembles ; un sous-ensemble S des

sommets dont la plus courte distance jusqu'au point 1 est connue et \bar{S} le sous-ensemble contenant les autres points.

A chaque nœud du graphe, on affecte la valeur $SP(i)$ de la longueur du chemin le plus court entre 1 et i trouvé jusqu'à présent. $SP(i)$ vérifie les propriétés suivantes :

- Si $i \in S$ alors $SP(i) = SP^*(i)$
 - Sinon, $SP(i) = \underset{\substack{k \in S \\ k \in \Gamma_i^{-1}}}{\text{Min}}(SP(k) + L_{k,i})$
- (32)

avec Γ_i l'ensemble des successeurs de i et Γ_i^{-1} l'ensemble de ses prédécesseurs. Dans notre cas ces deux ensembles sont identiques ; ils constituent l'ensemble des points adjacents au point i définis ci-dessus (la matrice associée au graphe est symétrique).

L'algorithme est le suivant :

-
1. Calcul de la matrice $[L]$ associée au graphe (distances entre les points adjacents),
 2. Initialisation : $\bar{S} = \{2, 3, \dots, N\}$ $SP(1) = 0$ et $SP(i) = \begin{cases} L_{1,i} & \text{si } i \in \Gamma_1, \\ +\infty & \text{sinon.} \end{cases}$
 3. Pour $c = 0$ à $N - 1$, Faire :
 - 3.1. Sélectionner $j \in \bar{S}$ tel que : $SP(j) = \min_{i \in \bar{S}} (SP(i))$
 - 3.2. $\bar{S} = \bar{S} - \{j\}$
Si $\text{Card}(\bar{S}) = 0$ Alors FIN.
 - 3.3. Pour tout $i \in \Gamma_j \cap \bar{S}$
 $SP(i) = \min (SP(i), SP(j) + L_{j,i})$ (voir (32)).
- FIN.
-

2.4.3.2. Algorithme 2

Cet algorithme est entièrement différent du précédant. Il repose sur une toute autre représentation géométrique. Dans ce qui suit, nous décrivons la représentation géométrique de l'atelier utilisée, puis nous montrons la manière dont a été utilisé l'algorithme d'exploration par séparation et évaluation (ESE) pour résoudre ce problème.

a. Représentation géométrique

La surface de l'atelier est découpée horizontalement en blocs rectangulaires adjacents de différentes tailles. Nous distinguons deux types de blocs (voir figure B18) :

- **Type 1** : les blocs occupés qui représentent les ressources ou les zones interdites de l'atelier (grisés sur la figure),
- **Type 2** : les blocs non occupés (blancs) qui représentent les zones libres de l'atelier où le transporteur peut se déplacer.

Le découpage de la surface de l'atelier en blocs rectangulaires est réalisé en se basant sur la matrice binaire [MATEL], décrite ci-dessus dans la section 1.3.1.

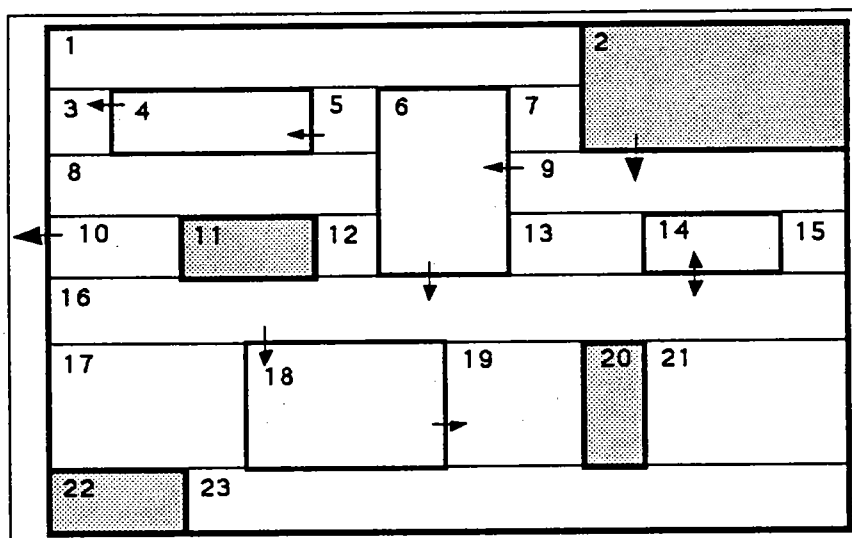


Figure B18 : Représentation géométrique de l'atelier

On traite simultanément deux lignes consécutives ($i - 1$) et i ($i = 2, \dots, n_l$) de la matrice [MATEL] composée de n_l lignes et n_c colonnes. La progression se fait ligne par ligne.

```

ligne i-1  00111111110001100
ligne i    000000000000001100

```

L'algorithme de découpage est le suivant :

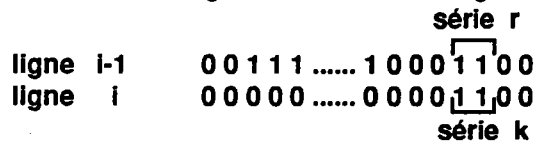
Pour $(i = 2, 3, \dots, nl)$, faire :

a. On décompose chaque ligne en segments (séries d'éléments) de même valeur (0 ou 1). (Dans l'exemple ci-dessus, la ligne $i - 1$ comporte 5 segments, 3 d'entre eux sont du type 2 et 2 du type 1).

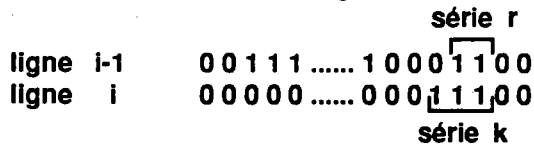
b. On examine ensuite les positions relatives des séries r de la ligne $i - 1$ et k de la ligne i . Si les deux séries sont de même nature, plusieurs cas sont possibles :

b.1. Si les extrémités droites de la série r de la ligne $i - 1$ et de la série k de la ligne i sont sur la même colonne, alors :

b.1.1. Si les extrémités gauches des deux séries sont aussi sur la même colonne, alors : les deux séries appartiennent au même bloc. On passe alors aux séries $r + 1$ de ligne $i - 1$ et $k + 1$ de la ligne i .

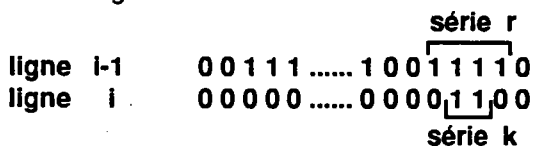


b.1.2. Sinon, si les extrémités gauches des deux séries ne sont pas sur la même colonne, alors : le bloc contenant la série r de la ligne $i - 1$ se termine et commence un nouveau bloc avec la série k de la ligne i . On passe ensuite aux séries $r + 1$ de la ligne $i - 1$ et $k + 1$ de la ligne i .



b.2. Sinon, si les extrémités droite de la série r de la ligne $i - 1$ et la série k de la ligne i ne sont pas sur la même colonne, alors :

b.2.1. Si le numéro de colonne de l'extrémité gauche de la série r de la ligne $i + 1$ est inférieur à celui de l'extrémité gauche de la série k de la ligne i , alors : le bloc contenant la série r se termine et commence un nouveau bloc contenant la série k . On passe ensuite à la série $r+1$ de la ligne $i - 1$ et la série $k + 1$ de la ligne i .



b.2.2. Sinon, si le numéro de colonne de l'extrémité gauche de la série k de la ligne $i - 1$ est supérieur à celui de l'extrémité gauche de la série k de la ligne i , alors : le bloc contenant la série k se termine et commence un nouveau bloc contenant la série k . On passe ensuite aux séries $r + 1$ de la

ligne $i - 1$ et $k + 1$ de la ligne i .

									série r				
ligne $i-1$	0	0	1	1	1	1	0	0	1	1	0	0
ligne i	0	0	0	0	0	0	0	0	1	1	1	0
													série k

Fin

Au cours du balayage de la matrice [MATEL], et au fur et à mesure qu'un bloc rectangulaire est construit, on lui affecte les paramètres suivants :

- un numéro (la numérotation va de gauche à droite et de haut en bas),
- un indice de couleur (0 pour libre et 1 pour occupé),
- les coordonnées de ses sommets,
- la liste des blocs qui lui sont adjacents.

b. Position du problème

En se basant sur cette représentation géométrique, nous posons le problème de recherche des plus courts chemins de la manière suivante :

Trouver une séquence $S_{i,j}$ de blocs $B_{i,j}^k$ (k est un entier non nul), correspondant au chemin le plus court joignant la sortie de cellule i (points de départ) et l'entrée de cellule j (points d'arrivée), tels que :

- 1 • $\forall k, B_{i,j}^k$ est libre (indice de couleur 0),
- 2 • $B_{i,j}^1$ contient le point de départ i ,
- 3 • $\forall l, m \in \{1, 2, \dots, n\}$ si $l \neq m$ alors $B_{i,j}^l \neq B_{i,j}^m$,
- 4 • $\forall k \in \{2, 3, \dots, n\}$ $B_{i,j}^{k-1}$ est adjacent à $B_{i,j}^k$,
- 5 • $B_{i,j}^n$ contient le point d'arrivée j .

La condition 1 exprime le fait que les déplacements ne se font que sur les blocs libres, la troisième exprime le fait que l'on ne passe jamais deux fois sur le même bloc libre.

Lors du passage d'un bloc à un autre, les mouvements du transporteur sont parallèles aux axes. Ainsi, à chaque transition entre deux blocs adjacents de la séquence, on calcule la distance élémentaire correspondante. La somme de toutes ces distances nous donne la longueur du chemin recherché.

Donnons d'abord quelques définitions utiles concernant le processus de calcul de ces portions de chemin élémentaires :

- A désigne la position courante appartenant au bloc i et dont les coordonnées sont (x_A, y_A) ,
- A' désigne le point de destination appartenant au bloc i + 1 et dont les coordonnées sont $(x_{A'}, y_{A'})$,
- (x_1, y_1) , (x_2, y_2) , (x_3, y_3) et (x_4, y_4) désignent respectivement les coordonnées des quatre sommets du bloc destination (i + 1), avec bien entendu $(x_1 = x_3, x_2 = x_4, y_1 = y_2$ et $y_3 = y_4)$.

Ces portions de chemins élémentaires sont de trois types. Nous montrons dans ce qui suit la manière dont elles sont calculées.

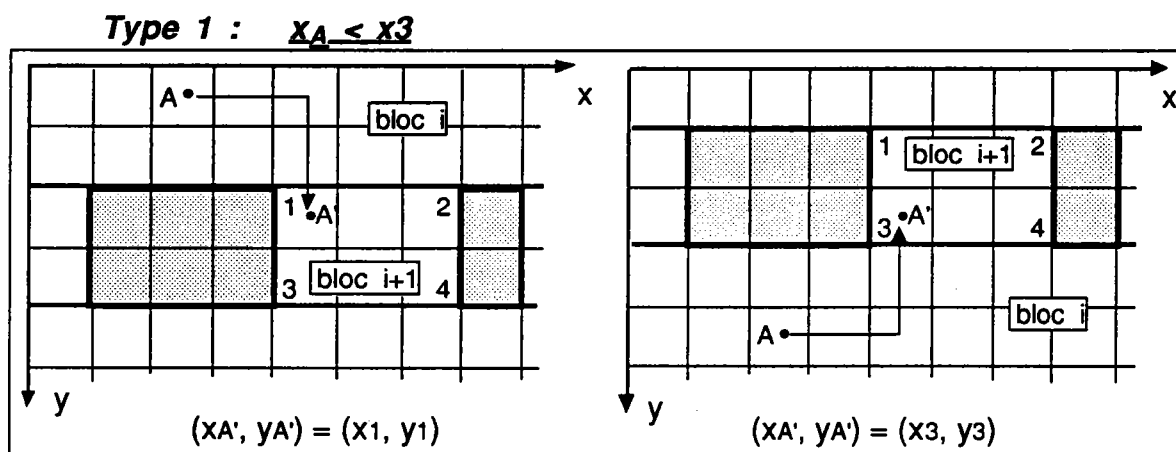


Figure B19 : Mouvement élémentaire du type 1

La distance parcourue dans ce cas est $D_i = |x_A - x_{A'}| + |y_A - y_{A'}|$.

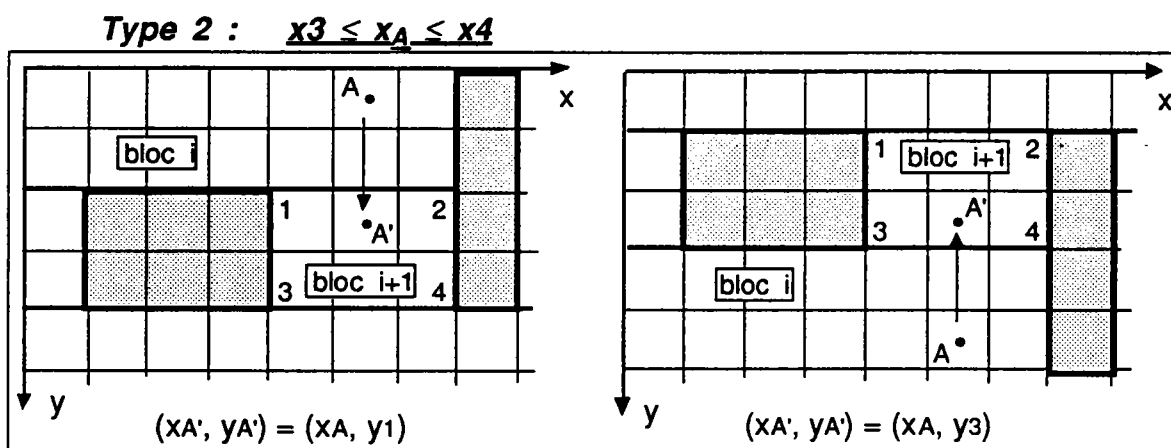


Figure B20 : Mouvement élémentaire du type 2

La distance parcourue dans ce cas est $D_i = |y_A - y_{A'}|$.

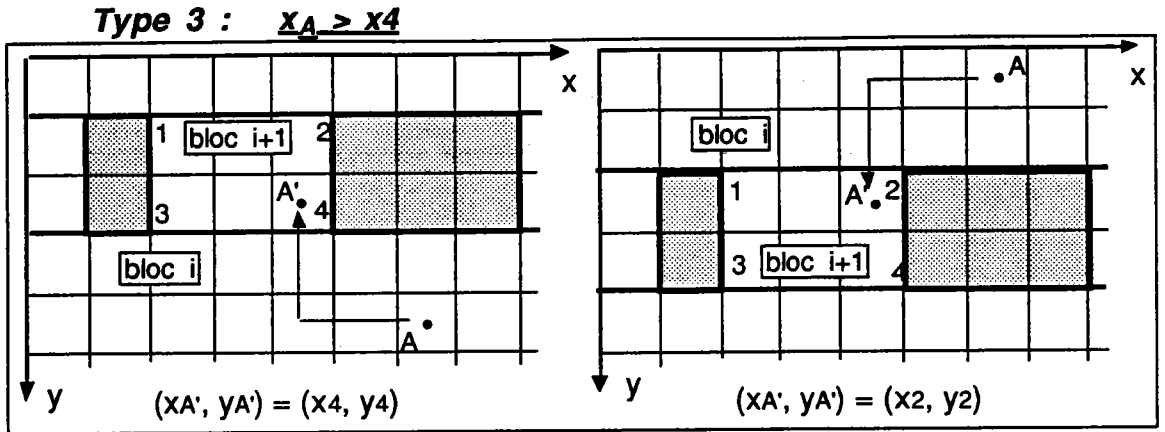


Figure B21 : Mouvement élémentaire du type 3

La distance parcourue dans ce cas est $D_i = |x_A - x_{A'}| + |y_A - y_{A'}|$.

c. Méthode de résolution

Dans ce qui suit, nous expliquons la manière dont l'algorithme ESE est appliqué à ce problème. Cet algorithme développe un arbre de recherche, dans lequel :

- chaque nœud exploré représente un bloc et contient la séquence partielle des blocs joignant le bloc de départ au bloc considéré,
- et chaque arc représente la portion de chemin entre le bloc considéré et le bloc antécédent direct. L'arc est pondéré par la longueur de cette parcelle de chemin.

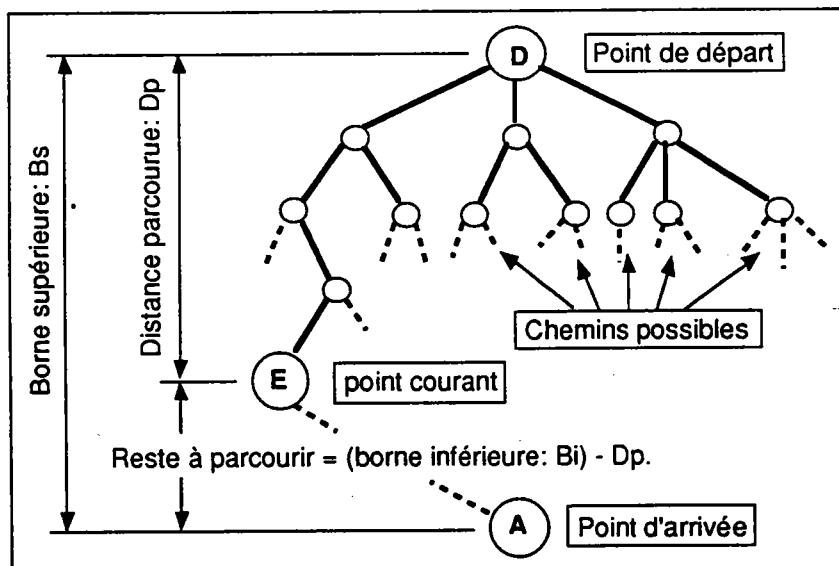


Figure B22 : Arbre de recherche du chemin le plus court

Dans la figure B22 présentant un exemple d'arbre de prospection, D est le point de départ, A le point d'arrivée et E un point intermédiaire. Nous y illustrons la borne supérieure B_s , la borne inférieure B_i et la distance parcourue D_p .

L'algorithme ESE appliqué à ce problème est le suivant :

Pour chaque couple de points (i, j) , faire:

1. Calculer une borne supérieure.

2. Faire :

- 2.1. Identifier le nœud courant (celui qui donne la plus petite valeur du critère),
- 2.2. Identifier les descendants du nœud (tous les blocs qui lui sont adjacents et de même couleur),
- 2.3. Eliminer les nœuds dominés (correspondant aux blocs prospectés dans une étape antérieure et ayant donné une distance plus grande que la distance actuelle),
- 2.4. Calculer la borne inférieure de chacun des nœuds restants et mettre à jour la liste des nœuds actifs (les nœuds qui ne vérifient aucune des deux relations suivantes : $D_p \geq B_s$ ou $B_i \geq B_s$; (voir figure 21)).

Si le nœud actif est le nœud d'arrivée et $D_p < B_s$, alors :

$$B_s = D_p;$$

Tant que (le nombre de points non prospectés est non nul),

3. $d^*(i, j) = B_s$;

Fin.

d. La borne supérieure

La borne supérieure B_s est la plus petite distance réalisable et calculable, supérieure à la longueur du plus court chemin entre un point de départ D et un point d'arrivée A (voir figure B22). La qualité de la borne supérieure affecte fortement le temps de calcul ; plus cette borne est proche du minimum recherché plus l'opération d'évaluation est sélective et par conséquent le nombre de nœuds à prospector est faible. Il est donc important de disposer d'un algorithme rapide donnant des bornes supérieures de bonne qualité.

Dans l'heuristique que nous proposons ci-dessous $D = (x_D, y_D)$ désigne le point de départ, $E = (x_E, y_E)$ le point courant et $A = (x_A, y_A)$ le point d'arrivée.

L'algorithme est le suivant :

A partir du bloc libre contenant le point D et à chaque point intermédiaire K, faire :

1. Chercher le ou les blocs libres qui lui sont adjacents et qui ont la distance $\Delta y = |y_K - y_A|$ la plus petite possible,
2. Vérifier si l'un de ces blocs contient le point A :
 - 2.1. Si oui, ce bloc est le dernier de la séquence et nous obtenons ainsi la borne supérieure,
 - 2.2. Sinon, nous prenons comme bloc intermédiaire celui qui donne la plus petite distance $\Delta x = |x_K - x_A|$ et qui n'a pas été utilisé antérieurement.

Tant que (A n'est pas atteint).

Fin.

e. La borne inférieure

La borne inférieure B_i est obtenue par relaxation de la contrainte de contournement des objets se trouvant dans l'atelier. Nous obtenons la valeur suivante :

$$B_i = D_p + |x_E - x_A| + |y_E - y_A|.$$

C'est une valeur qui est calculable à tout moment et qui ne coûte pratiquement rien en temps de calcul.

Cette valeur est utilisée, avec B_s , pour abandonner un chemin en construction si $D_p < B_s$ et $B_i > B_s$.

2.4.3.3. Evaluation des méthodes proposées

Nous présentons dans ce paragraphe deux séries d'applications dont les résultats serviront de base pour comparer les deux méthodes ci-dessus, afin d'évaluer le domaine d'utilisation de chacune d'elles. Ces deux séries sont faites sur cinq exemples différents que nous présentons ci-dessous.

Exemple 1

Atelier à 5 cellules, 4 zones interdites, 2 entrées et 2 sorties. La représentation géométrique de sa configuration ainsi que la matrice des flux associée sont données dans l'annexe E.2.1.

Exemple 2

Atelier à 10 cellules, 4 zones interdites et sans entrées/sorties. La représentation géométrique de sa configuration et la matrice des flux sont données dans l'annexe E.2.2.

Exemple 3

Atelier à 15 cellules, 5 zones interdites et sans entrées/sorties. La représentation géométrique de sa configuration et la matrice des flux sont données dans l'annexe E.2.3.

Exemple 4

Atelier à 20 cellules, 5 zones interdites, 2 entrées et 2 sorties. La représentation géométrique de sa configuration et la matrice des flux sont données dans l'annexe E.2.4.

Exemple 5

Atelier à 25 cellules, 6 zones interdites, 2 entrées et 2 sorties. La représentation géométrique de sa configuration et la matrice des flux sont données dans l'annexe E.2.5.

La première avec les configurations données ci-dessus et des matrices des flux remplies à 100% (c'est à dire que tous les flux sont non nuls). Les résultats de cette série sont donnés par le tableau B1. La seconde série est faite avec des matrices de flux plus réalistes (données dans les énoncés des exemples 1 à 5, voir annexe E.2) où certains flux sont nuls. Les taux de remplissage de ces matrices des flux, notées TRF, sont donnés dans le tableau B2 avec les résultats de cette série.

a. Interprétation des résultats

Afin d'évaluer les deux algorithmes proposés, pour chaque exemple, nous donnons le temps d'initialisation (construction du modèle et/ou initialisation de la matrice des distances) et le temps d'exécution de l'algorithme pour le calcul des longueurs des chemins les plus courts (voir tableau B1 et B2). Des données et/ou résultats supplémentaires sont donnés pour chaque série afin de faciliter les interprétations. Les calculs ont été réalisés sur un SUN sparc (station 330) et les temps sont donnés en unités CPU.

La première constatation que l'on peut faire à la vue ces deux tableaux est que, pour tous les exemples, le temps d'initialisation dans le premier algorithme est nettement supérieur à celui du deuxième algorithme. Ceci est dû à la nature du modèle géométrique utilisé dans chaque algorithme et à la manière dont ses éléments sont générés. Dans la partie initialisation du premier algorithme, nous trouvons d'abord les sommets des obstacles, puis nous calculons les distances entre les couples de points adjacents. Tandis que dans le second, nous ne faisons que le partage de la surface de l'atelier, le calcul des distances entre les

blocs libres adjacents se fait au cours de la résolution selon les besoins. Une amélioration du processus d'initialisation du premier algorithme reste envisageable.

Première série

	Algorithme 1			Algorithme 2			$T_{\text{trai.2}}/T_{\text{trai.1}}$	$T_{\text{tot.2}}/T_{\text{tot.1}}$
	$T_{\text{Init.}}$	$T_{\text{trai.1}}$	$T_{\text{tot.1}}$	$T_{\text{Init.}}$	$T_{\text{trai.2}}$	$T_{\text{tot.2}}$		
5 cellules	0.200	0.033	0.233	0.017	0.133	0.150	4.030	0.644
10 cellules	0.233	0.100	0.333	0.033	0.316	0.349	3.160	1.048
15 cellules	0.450	0.350	0.800	0.033	1.417	1.460	4.049	1.825
20 cellules	0.700	0.883	1.567	0.050	4.266	4.333	4.831	2.765
25 cellules	0.883	1.533	2.416	0.067	8.133	8.200	5.305	3.394

Tableau B1 : Résultats de la première série : Matrices des flux pleines à 100%

Pour un même nombre de chemins à calculer (ici le nombre maximum), le temps de traitement du deuxième algorithme est dans la plupart des cas plus grand que celui du premier. Ceci est illustré par le rapport $T_{\text{trai.2}}/T_{\text{trai.1}}$ qui augmente avec le nombre de cellules. Le temps de traitement du deuxième algorithme est jusqu'à 5.305 fois plus grand. Cette différence s'atténue mais persiste en ce qui concerne le temps total cumulé. Le rapport $T_{\text{tot.2}}/T_{\text{tot.1}}$ atteint 3.394. Nous constatons qu'il est inférieur à 1 pour l'exemple à 5 cellules et légèrement supérieur à 1 pour l'exemple à 10 cellules (ce qui montre que pour un nombre de cellules de l'ordre de 5, le deuxième algorithme reste plus intéressant même si la matrice des flux est pleine à 100%).

Deuxième série

	TRF	Algorithme 1			Algorithme 2			$T_{\text{tot.1}}/T_{\text{tot.2}}$
		$T_{\text{Init.}}$	$T_{\text{trai.1}}$	$T_{\text{tot.1}}$	$T_{\text{Init.}}$	$T_{\text{trai.2}}$	$T_{\text{tot.2}}$	
5 cellules	50,00%	0.200	0.033	0.233	0.017	0.050	0.067	3.477
10 cellules	41.11%	0.233	0.100	0.333	0.033	0.100	0.133	2.504
15 cellules	21.90%	0.450	0.350	0.800	0.033	0.350	0.383	2.088
20 cellules	20.79%	0.700	0.883	1.567	0.050	1.016	1.066	1.470
25 cellules	11.49%	0.883	1.533	2.416	0.067	1.267	1.334	1.811

Tableau B2 : Résultats de la deuxième série : Matrices des flux non pleines

Nous constatons en premier lieu que, pour le premier algorithme, les temps ne changent pas entre la première et la deuxième série. Ceci est dû au fait que cet algorithme calcule, de toute façon, tous les chemins entre les sorties et les entrées des cellules. Malgré des taux de remplissage des matrices des flux TRF ne dépassant pas les 50% les temps de traitement des deux algorithmes sont assez comparables. Seulement, le deuxième algorithme a l'avantage d'avoir un temps de traitement très réduit par rapport au premier. Cela rend le temps total cumulé de ce deuxième algorithme plus intéressant (voir dernière colonne du tableau) dans certains cas. Nous constatons aussi que le rapport $T_{tot.1}/T_{tot.2}$ est plus grand pour les petits systèmes (< 10 cellules) bien que les taux de remplissage des matrices des flux sont les plus importants.

b. Consignes d'utilisation

A partir des remarques faites ci-dessus, nous déduisons que le temps total de calcul des deux algorithmes augmente avec le nombre de cellules. Celui du deuxième algorithme augmente en plus avec le taux de remplissage de la matrice des flux.

Ceci nous suggère donc de trouver les limites de ces deux paramètres pour lesquels il est plus avantageux d'utiliser l'une ou l'autre des deux méthodes. Nous avons effectué pour cela un grand nombre d'applications en utilisant la deuxième méthode pour des systèmes ayant un nombre de cellules N_c égal à 5, 10, 15, 20 et 25. Nous avons fait varier le TRF (taux de remplissage des matrices des flux) entre 10% et 100% avec un pas de 5%. Nous avons comparé pour chaque application le temps total cumulé de calcul des chemins avec celui de la première méthode. Les résultats obtenus sont donnés ci-dessous :

Afin d'avoir les temps de traitement les plus réduits possibles lors du calcul des longueurs des chemins les plus courts au moment de la résolution du problème de l'agencement, nous utiliserons le deuxième algorithme dans les situations données dans le tableau B3. Dans toutes les autres situations, c'est le premier algorithme qui est utilisé.

Nc	TRF
$N_c \leq 5$	$\leq 100\%$
$5 < N_c \leq 10$	$\leq 90\%$
$10 < N_c \leq 15$	$\leq 50\%$
$15 < N_c \leq 20$	$\leq 35\%$
$20 < N_c \leq 25$	$\leq 30\%$

Tableau B3 : Utilisation des méthodes de calcul des chemins

2.5. Conclusion du deuxième chapitre

Ce chapitre a été consacré au problème de l'agencement des cellules dans un atelier de fabrication. Nous y avons décrit le problème global et proposé une méthode de résolution basée sur l'algorithme du recuit simulé. Les différents sous-problèmes qui le constituent ont été décrits et résolus. Nous nous sommes ensuite intéressés au problème le plus épineux, à savoir celui du calcul des longueurs des plus courts chemins entre les points de départ et d'arrivée des transporteurs. Ce dernier prend la plus grande part du temps de calcul lors de la résolution du problème global d'agencement des cellules. Deux algorithmes complémentaires ont été proposés pour le résoudre. L'utilisation de l'un ou l'autre de ces deux algorithmes dépend du nombre de cellules et du taux de remplissage de la matrice des flux dans l'atelier.

Un logiciel a été développé pour ce problème. Nous le décrivons entièrement dans la partie C de cette thèse.

3. INTRODUCTION A L'AGENCEMENT DYNAMIQUE DES SYSTEMES DE FABRICATION

Dans ce chapitre, nous ne faisons qu'introduire le problème de l'agencement dynamique des systèmes de fabrication. Nous montrons la manière dont pourraient être appliquées les techniques existantes pour résoudre ce problème dans le cas des systèmes cellulaires de fabrication. Nous expliquons également la manière dont peut être utilisé le recuit simulé pour améliorer les performances des méthodes existantes. Enfin, nous posons les questions liées à ce problème et qui restent à résoudre totalement ou partiellement.

3.1. Introduction

Le monde industriel vit de nos jours un phénomène d'accélération de la réduction du cycle de vie des produits. Par conséquent, le cycle de vie des systèmes de fabrication qui réalisent ces produits subit le même sort. Ceci est dû à une haute fréquence d'apparition de nouveaux produits, aux fortes fluctuations de la demande, et à l'évolution rapide des techniques de fabrication. Cette

nouvelle situation impose une mutation du mode d'agencement des systèmes de production : on passe d'un mode traditionnel réactif à un mode d'agencement pré-actif qui s'adapte mieux aux exigences actuelles.

Dans un mode réactif, il est simplement demandé au concepteur de donner un agencement des ressources qui soit admissible et qui, tout en minimisant un critère donné, respecte certaines contraintes physiques imposées au système. Ce processus est répété chaque fois que l'on constate une forte augmentation du coût de fabrication due à un agencement mal adapté. Dans le mode pré-actif d'agencement, le concepteur doit prévoir l'évolution du système de fabrication, en se basant sur l'historique du comportement du système et sur les prévisions faites en fonction des tendances du marché.

Une décision stratégique concernant le mode d'agencement (réactif ou pré-actif) doit donc être prise. Les éléments de décision sont, principalement, la vitesse d'évolution du système et la différence relative qui existe entre les coûts moyens de fonctionnement du système (manutention) et les coûts moyens de réagencement. Ces derniers sont constitués du coût de déplacement des ressources (qui dépend du nombre de ressources à déplacer, des distances de déplacement et des portions du système de manutention à changer) (Rosenblatt 1986), et par le coût de la production non réalisée durant le temps d'arrêt du système (ce coût croît avec le temps d'arrêt) (Afentakis et al. 1990).

a. Mode d'agencement réactif des systèmes dynamiques

On choisit le mode d'agencement réactif lorsqu'il s'agit d'un système stable pour lequel les coûts moyens de réagencement des ressources sont négligeables par rapport aux coûts moyens de fonctionnement du système. Dans ce cas, le concepteur réalise des réagencement statiques et indépendants, sans aucune prévision, pour chaque nouvel état de la demande nécessitant un changement de configuration.

b. Mode d'agencement pré-actif des systèmes dynamiques

Le mode d'agencement pré-actif est adopté lorsque les coûts moyens de réagencement du système ne sont plus négligeables devant les coûts moyens de fonctionnement du système. Deux situations différentes sont possibles :

b.1. Première situation

Celle où le coût de réagencement est trop important. Dans ce cas le principal objectif du concepteur est de trouver un agencement robuste, c'est-à-

dire un agencement fixe durant tout l'horizon de planification qui résiste aux fluctuations de la demande. La demande varie sous forme de scénarios (un scénario est l'ensemble des couples (type de produit, demande) à réaliser durant une période donnée) et à chaque scénario correspond une configuration optimale. La configuration fixe, à trouver, doit être proche des configurations optimales des différents scénarios sans être nécessairement égale à l'une d'elles. Les lignes automatisées de peinture, les machines de mesure et de contrôle de qualité (banc d'essais) et les machines de nettoyage des produits sont des exemples de ressources à fort coût d'installation.

Dans cette situation l'agencement dynamique du système de fabrication est appelé **mono-périodique** (cette période étant l'horizon de planification).

b.2. Deuxième situation

Les coûts de réagencement ne sont ni négligeables ni prohibitifs. Dans ce cas, le concepteur vise un agencement flexible du système de fabrication. Il ne faut pas entendre par cela la flexibilité du système de fabrication, qui est son aptitude à fabriquer plusieurs types de produits à des quantités variables. Il faut plutôt entendre par cela la flexibilité de l'agencement du système qui est son aptitude à changer de configuration afin de produire un plus grand nombre de types de produits et d'accroître, sinon maintenir, la rentabilité du système. Le concepteur doit donc trouver une configuration du système pour chaque scénario de demande, à maintenir durant une période donnée.

Dans cette situation, l'agencement dynamique du système de fabrication est appelé **multi-périodique** (la somme de ces différentes périodes constituent l'horizon de planification).

3.2. Etat-de-l'art

La nécessité d'un traitement dynamique de l'agencement a été soulignée, pour la première fois par Nicol et Hollier en 1983. Ils ont effectué une étude sur 33 systèmes de fabrication localisés en Grande Bretagne. Ils ont constaté que si l'on considère que la durée de vie d'un agencement est le temps qui s'écoule entre l'heure de sa première mise en marche et celle où le tiers des ressources ont changé de position, alors pratiquement la moitié des systèmes de production étudiés avait une stabilité moyenne de leur agencement égale à deux ans. Cette enquête prouve que l'agencement d'un système de fabrication varie dans le temps (voir Nicol et Holier 1983).

Nous donnons dans ce qui suit les conclusions de quelques travaux les plus marquants qui retracent l'évolution de l'étude de ce problème entre 1986 et aujourd'hui.

En 1986, Rosenblatt a développé le premier modèle mathématique pour résoudre le problème dynamique d'affectation quadratique multi-périodique avec des scénarios de demande déterministes. Il a résolu ce problème en procédant comme suit :

Pour chaque période, le concepteur résout un problème d'agencement statique et propose un ensemble de configurations potentielles. Il se fixe ensuite l'objectif de sélectionner la séquence optimale des configurations, une pour chaque période, qui minimise la somme totale des coûts de fonctionnement et des coûts de transformation du système pour passer d'une configuration à une autre. L'auteur utilise la méthode de programmation dynamique pour résoudre ce deuxième problème. Cette approche est moyennement intéressante si le nombre total de configurations à trouver, par période, n'est pas faible. Pour limiter ces nombres Rosenblatt utilise une méthode basée sur deux bornes, une borne supérieure et une borne inférieure, du coût total de fonctionnement du système (manutention et réagencement) (voir Rosenblatt 1986).

Une note de Batta (1987) complète la proposition de Rosenblatt. Il propose un théorème stipulant que si le même agencement est conservé durant tout l'horizon de planification du système de fabrication (agencement dynamique mono-périodique), le problème dynamique peut être résolu comme un problème d'agencement statique dans lequel la matrice des flux entre les ressources est obtenue par une combinaison linéaire des matrices des flux des différents scénarios probables. Les coefficients de cette combinaison sont les proportions de temps pris par chaque scénario sur l'horizon de planification (ou la probabilité d'occurrence des scénarios, si ces derniers ont la même durée). Le coût de cette configuration fixe est une borne supérieure pour le problème de limitation du nombre de configurations potentielles ci-dessus (voir Batta 1987).

En 1990, Afentakis, Millen et Solomon reconsidèrent le problème d'agencement multi-périodique pour étudier le problème de la durée de vie d'une configuration. Ils proposent deux stratégies. La première est destinée aux systèmes de nature périodique. Le problème est alors de trouver la période de réagencement. Dans la deuxième stratégie, destinée aux autres systèmes, ils s'intéressent à la détermination du moment où doit s'effectuer le réagencement. Ce moment est lié à la position d'une variable d'intérêt par rapport à un seuil fixé. Trois variables d'intérêt sont envisageables : le nombre de types de

produits, le changement des routages et la variation du volume total des flux. L'étude a été effectuée par simulation et a montré que l'on peut arriver à un gain de 36% en effectuant des réagencements souvent minimes en utilisant la bonne stratégie de réagencement (voir Afentakis et al. 1990).

Montreuil et Venkatadri (1991) proposent un modèle d'agencement dynamique multi-périodique pour des systèmes cellulaires de production se trouvant dans la phase d'expansion (arrivée de nouvelles ressources dans les cellules) ou de déclin (retrait de certaines ressources des cellules) de leur cycle de vie. Le nombre de cellules étant fixe, ce sont les surfaces des cellules qui varient. Dans ce modèle l'objectif est de concevoir par interpolation les configurations du système lors de son évolution entre l'état initial et l'état final connus. Les auteurs supposent aussi l'existence de contraintes sur les positions des entrées/sorties de certaines cellules dites rigides (des cellules dont le coût de déplacement est prohibitif) (voir Montreuil et Venkatadri, 1991).

Kouvelis et Kiran (1991) reposent les deux problèmes d'agencement mono- et multi-périodiques, dans les cas d'existence d'incertitudes dans les scénarios de demande, les gammes de fabrication et les routages des produits. Les auteurs ont développé des règles de dominance permettant de sélectionner les configurations potentielles pour chaque période. Ce nouveau concept leur permet de trouver la ou les solutions optimales dans le cas mono-périodique et de réduire le nombre de solutions alternatives par période dans le cas multi-périodique (voir Rosenblatt 1986). Ces règles sont utiles dans ce dernier cas, sachant que le temps global de traitement de ce type de systèmes varie linéairement avec le nombre de périodes et exponentiellement avec le nombre de solutions alternatives par période. Cette méthode permet de résoudre des problèmes de taille moyenne (entre 4 et 8 stations) dans des temps raisonnables (voir Kouvelis et Kiran 1991).

Balakrishnan et coll. (1992) reposent le problème d'agencement dynamique multi-périodique posé par Rosenblatt en 1986, en ajoutant une contrainte supplémentaire. Cette contrainte consiste à limiter le coût global des réagencements. Pour résoudre ce problème, ils utilisent une méthode de recherche du plus court chemin dans un graphe sous contraintes. Ce modèle présente le même inconvénient que celui de Rosenblatt ; le grand nombre de configurations potentielles à trouver se traduit dans cette approche par le grand nombre de nœuds du graphe (voir Balakrishnan et al., 1992).

Kouvelis et coll. (1992) reconsidèrent le problème de l'incertitude des données lors de l'agencement dynamique des systèmes dans le cas d'un

agencement par affectation quadratique. Ils cherchent à trouver un agencement robuste du système, qui peut résister aux variations des scénarios de demandes et rester rentables. Ils se basent sur la définition de la robustesse donnée par Rosenblatt et Lee en 1987 (voir Rosenblatt et Lee, 1987) qui dit que : la robustesse d'un agencement donné est mesurée par le nombre de fois où la différence relative entre le coût de l'agencement fixé soumis aux différents scénarios et le coût des agencements optimaux des scénarios reste inférieur à un pourcentage fixé par le concepteur. Cette définition considère que la connaissance des probabilités de réalisation des différents scénarios, dont parle Batta (1987), n'est pas nécessaire. Les contributions de Kouvelis et coll. sont la généralisation de ce problème d'incertitude aux systèmes multi-périodiques et l'utilisation d'une procédure de séparation et évaluation (Branch and Bound) modifiée pour générer des agencements robustes pour les systèmes mono-périodiques (voir Kouvelis et coll. 1992).

Lacksonen et Enscore (1993) considèrent le problème d'agencement dynamique multi-périodique et proposent des modifications dans cinq algorithmes existants pour résoudre ce problème modélisé sous forme d'un problème d'affectation quadratique. Les cinq méthodes sont : CRAFT, "Cutting plan", Séparation et évaluation (Branch & Bound), Programmation dynamique et "Cut trees". Les auteurs concluent que le "Cutting plan" trouve les solutions les plus proches de l'optimum dans tous les exemples traités (voir Lacksonen et Enscore, 1993).

3.3. Agencement dynamique mono-périodique

Dans ce qui suit, nous présentons ce problème et donnons une formulation mathématique aussi simple que possible. Nous nous plaçons dans le cas de l'agencement des cellules dans l'atelier cellulaire de fabrication. Ce cas est typique pour ce mode d'agencement, car le coût de déplacement d'une cellule avec toutes ses machines, son système de manutention et ses accessoires est évidemment assez élevé.

Au moment de la conception du système de fabrication, les types de produits et leurs demandes ne sont pas parfaitement connus. Après la réalisation du système et sa mise en fonctionnement, ces paramètres évoluent en passant par un certain nombre de scénarios probables, plus ou moins lents. Il s'agit donc de trouver une configuration robuste du système de fabrication, à

conserver durant tout l'horizon de planification. Nous adoptons dans ce qui suit la définition de la robustesse proposée par Batta en 1987 (voir état-de-l'art).

3.3.1. Position du problème

Nous proposons le modèle suivant :

Soit un système dynamique de fabrication qui vérifie les hypothèses suivantes :

1. les ressources du système sont capables de fabriquer un ensemble P de plusieurs types de produits. Chaque type de produit est défini par son routage à travers les cellules (macro-routage).

2. le système peut être soumis à un nombre K de scénarios probables. Le scénario k est défini par le sous-ensemble de produits P_k à fabriquer durant une période donnée, la demande D_q^k et la taille du lot U_q^k (considérée constante entre toutes les ressources) associées à chaque type de produit q ($q \in P_k$) du scénario k ,

3. les scénarios des demandes arrivent au système d'une manière séquentielle et aléatoire. p_k est la probabilité d'occurrence du scénario k durant l'horizon de planification du système. Cette probabilité peut être exprimée par la proportion de temps que prend le scénario k sur tout l'horizon de planification.

Batta (1987) (voir état de l'art ci-dessus) montre que ce type de problème peut être résolu comme un problème d'agencement statique. Ceci après avoir calculé la matrice $[FE]$ des flux moyens globaux entre les ressources durant tout l'horizon de planification.

Dans le cas du problème d'agencement des cellules dans un atelier de fabrication (voir chapitre 2 de cette partie B), la fonction objectif est :

$$\min \sum_{i=1}^M \sum_{\substack{j=1 \\ i \neq j}}^M FE_{i,j} \times d_{i,j} \quad (33)$$

$d_{i,j}$ est la distance entre la sortie de la cellule C_i et l'entrée de la cellule C_j ,

$FE_{i,j}$ est le flux moyen global de la cellule C_i vers la cellule C_j .

$$FE_{i,j} = \sum_{k=1}^{NS} p_k \times F_{i,j}^k \quad (34)$$

p_k est la probabilité d'occurrence du scénario k ,

$F_{i,j}^k$ est le flux de C_i vers C_j pour le scénario k ,

NS est le nombre de scénarios probables.

$$F_{i,j}^k = \sum_{q=1}^P \frac{D_q^k}{TL_q^k} \times T_{i,j}^q \quad (35)$$

D_q^k est la demande du produit q durant le scénario k ,

U_q^k est la taille du lot du produit q transporté durant le scénario k ,

$T_{i,j}^q$ est le trafic du produit q entre les cellules C_i et C_j .

Cette fonction objectif est soumise aux mêmes contraintes de proximité entre les cellules, fortes et molles, que dans le cas statique. Ces contraintes ne dépendent pas du scénario (voir le point 2.2.2 de cette partie B).

3.3.2. Résolution du problème

Pour résoudre ce problème, le recuit simulé peut être utilisé très aisément et sans aucune modification par rapport au problème d'agencement statique. L'algorithme est le suivant :

-
1. Acquisition des données géométriques (identiques à ceux du problème statique).
 2. Acquisition des différents scénarios de demandes possibles :
 - 2.1. Acquérir l'ensemble des types de produits avec leurs routages respectifs.
 - 2.2. Pour chaque scénario, acquérir :
 - la probabilité d'occurrence du scénario,
 - le sous-ensemble des produits à fabriquer,
 - pour chaque type de produit du sous-ensemble, acquérir la demande et la taille du lot.
 3. Calculer, pour chaque scénario, la matrice des flux associée (à l'aide de la formule (35)).
 4. Calculer la matrice globale des flux.
 5. Utiliser la méthode du recuit simulé pour trouver la configuration fixe du système (voir chap. 2).

FIN.

3.3.3. Conclusion

Deux définitions de la robustesse d'un agencement ont été proposées jusqu'à présent, celle de Batta (1987) suivie par celle de Rosenblatt et Lee (1987). Nous avons posé le problème d'agencement dynamique mono-périodique dans le cas de l'agencement des cellules dans l'atelier de

fabrication. Nous avons donné ensuite une formulation mathématique basée sur la première définition de la robustesse. Nous montrons, à travers l'algorithme de résolution du problème, que l'utilisation de l'algorithme du recuit simulé reste toujours valable. Les résultats donnés sont d'aussi bonne qualité que pour le cas statique.

3.4. Agencement dynamique multi-périodiques

Dans ce paragraphe, nous commençons par poser le problème d'agencement dynamique des systèmes multi-périodiques sous sa forme la plus simple que nous appliquons au problème de l'agencement interne des cellules dans un système cellulaire de fabrication. Nous montrons enfin les différentes possibilités de généralisation.

3.4.1. Position du problème et méthode de résolution

Soit une cellule de fabrication soumise à un certain nombre de scénarios de demande durant l'horizon de planification. Nous considérons les hypothèses suivantes :

1. L'horizon de planification est partagé en périodes de temps égales dont la durée se mesure en mois, en trimestres ou en années selon le type de fabrication.
2. Les scénarios de demande Sc_i ($i = 1, 2, \dots, H$) sont parfaitement connus et s'appliquent au système d'une manière séquentielle sans chevauchement dans le temps.
3. Un réagencement de la cellule est réalisé entre la fin d'un scénario et le début du suivant.

Chaque scénario de demande i dispose de plusieurs configurations potentielles $\{Ag_i^1, Ag_i^2, \dots, Ag_i^{K-1}, Ag_i^K\}$ (voir figure B23). Il s'agit donc de sélectionner pour chacun des scénarios de demande donné une configuration Ag_i^* convenable en ayant pour objectif la minimisation du coût total de la fabrication durant tout l'horizon de planification. Ce coût total est la somme des coûts de fonctionnement du système pendant les durées des scénarios et des coûts de passage d'une configuration à une autre.

Pour résoudre ce problème de base, une méthode en deux étapes a été proposée par Roseblatt (1986) (voir état-de-l'art ci-dessus).

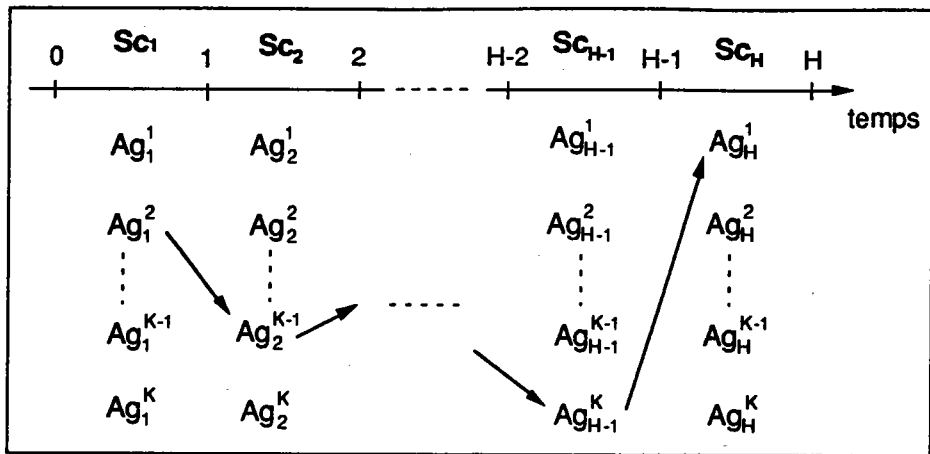


Figure B23 : Problème simple d'agencement dynamique multi-périodique

Etape 1

Le processus de résolution commence par la génération, pour chaque scénario, d'un ensemble de solutions potentielles. Ce problème est exactement identique aux sous-problèmes d'agencement statique posés dans le problème d'agencement des systèmes cellulaires de fabrication que nous avons résolu (voir les paragraphes 2.4 de la partie A "agencement intra-cellulaire"). Le réagencement interne des cellules est une pratique courante dans les systèmes cellulaires car les fluctuations de la demande en types et en quantités de produits se répercutent plus sur l'agencement interne des cellules que sur l'agencement des cellules dans l'atelier. En plus de cela, il est beaucoup moins coûteux de réagencer l'intérieur d'une cellule qu'un atelier tout entier.

Nous donnons donc, dans ce qui suit, la formulation de ce problème appliquée à l'agencement interne d'une cellule C donnée.

Pour chaque scénario l ($l = 1, 2, \dots, H$), on résout un problème d'agencement statique. Ce problème consiste à affecter les machines de la cellule aux différents sites fixés sur la configuration de base de la cellule, dans le but de minimiser le coût de la maintenance dans la cellule.

La méthode du recuit simulé que nous avons utilisée pour résoudre le problème d'agencement statique a justement un avantage qui nous est précieux dans ce cas. Cet avantage est le fait qu'elle peut donner plusieurs solutions différentes toutes aussi bonnes les unes que les autres.

Etape 2

Dans la deuxième étape, on résout un problème de programmation dynamique. En utilisant la terminologie de cette technique, on peut dire que les étapes correspondent aux périodes (scénarios) et les états correspondent aux configurations potentielles de la cellule à trouver pour chaque période. Le problème peut s'énoncer comme suit (Rosenblatt 1986).

Soient :

- R_l le nombre de configurations potentielles à considérer pour la période l . Ces configurations sont classées dans l'ordre croissant de leur coût de manutention.

- C_{km} le coût du réagencement pour passer de la configuration Ag_k à la configuration Ag_m , $C_{kk} = 0$. Nous considérons que ce coût ne dépend que de la distance entre les deux sites de la machine i dans les deux configurations successives k et m :

$$C_{km} = \sum_{i=1}^{N_m} \mu \times d(s^k(i), s^m(i)) \quad (38)$$

μ est le coût de déplacement d'une machine par unité de distance,

N_m est le nombre de machines dans la cellule.

- Z_{lk} est le coût de manutention de la $k^{\text{ième}}$ meilleure configuration de la période l . Ce coût a été calculé dans la première étape.

- L_{lk}^* est le coût minimum total (manutention et réagencement) de toutes les périodes jusqu'à la période l où la configuration Ag_k^l est sélectionnée. Sa valeur est calculée à l'aide de la formule récursive suivante :

$$L_{l,m}^* = \min_k \{L_{l-1,k}^* + C_{k,m}\} + Z_{l,m}^* \quad l = 1, 2, \dots, H \quad (39)$$

et $L_{0,1}^* = 0$, en supposant qu'il n'y a qu'un seul agencement initial (fictif).

La séquence optimale et son coût L^* sont obtenus à la fin du processus de récurrence correspondant à la dernière période H .

Pour résoudre ce problème d'une manière exacte, on calcule pour chaque période l les valeurs de $L_{l,m}^*$ pour tout m ($m = 1, 2, \dots, R_l$).

Le temps de traitement augmente alors linéairement avec H et exponentiellement avec R_l . Il est donc nécessaire de limiter ce nombre de configurations potentielles par période. Cette limitation est basée sur ce qui suit.

Soient :

• Z^{BI} une borne inférieure pour L^* le coût total minimum de la séquence optimale des configurations potentielles à trouver. Une valeur possible de cette borne est donnée par :

$$Z^{BI} = \sum_{i=1}^H Z_{i,1} \quad (40)$$

Une autre expression de cette borne inférieure a été proposée par T. Urban (1992).

• Z^{BS} une borne supérieure pour L^* . Une expression possible de cette borne est :

$$Z^{BS} = \sum_{i=1}^H Z_{i,1} + \sum_{i=2}^H C_{Ag_{i,1}, Ag_i} \quad (41)$$

Une meilleure borne supérieure a été proposée par R. Batta (1987). Elle est égale au coût de la configuration trouvée pour le problème d'agencement mono-périodique que nous avons présenté ci-dessus.

Sweeney et Tathan (1986) prouvent que :

$$\text{Si } K = Z^{BS} - Z^{BI}, \quad (42)$$

alors la valeur de R_i est donnée par :

$$\begin{cases} Z_{i, R_i} - Z_{i,1} \leq K \\ Z_{i, R_i + 1} - Z_{i,1} > K \end{cases} \quad (43)$$

et aucune solution dont le rang est supérieur à R_i ne peut faire partie de la séquence optimale des configurations potentielles à trouver.

3.4.2. Conclusion et remarques

Nous avons présenté le problème d'agencement dynamique multi-périodique sous sa forme la plus simple. Nous avons donné une méthode de résolution qui suppose certaines hypothèses assez fortes. Pour se rapprocher de la réalité de ce type de systèmes, plusieurs de ces hypothèses doivent être reconsidérées. Les problèmes qui restent posés actuellement et qui méritent plus d'attention dans un travail ultérieur sont :

1. Les incertitudes dans les scénarios de demandes (les demandes ne sont pas connues d'une manière certaine).
2. Cas où plusieurs scénarios sont probables pour chaque période.
3. Les incertitudes dans les durées des scénarios causées par les aléas de la fabrication.

4. Les incertitudes dans les dates d'arrivée des scénarios pouvant causer des chevauchements dans le temps entre les scénarios.
5. Trouver une méthode précise permettant le choix entre l'agencement mono-périodique et l'agencement multi-périodique pour un système donné .

3.5. Conclusion du troisième chapitre

Ce chapitre a été consacré à l'introduction du problème de l'agencement dynamique de fabrication. Deux modes d'agencement dynamique des systèmes de fabrication sont possibles : l'agencement mono-périodique et l'agencement multi-périodique. Nous avons présenté une formulation simple du problème pour chacun de ces deux modes. Nous avons ensuite appliqué le mode mono-périodique à l'agencement des cellules dans l'atelier de fabrication et le mode multi-périodique à l'agencement des machines dans les cellules. Pour chacun de ces deux problèmes, nous avons présenté une méthode de résolution où le recuit simulé joue un grand rôle.

Un certain nombre de problème restent ouverts dans ce domaines, tels que la prise en compte des incertitudes dans les demandes, dans leurs temps d'arrivée et dans leurs délais, Ces problèmes peuvent faire l'objet de travaux de recherche futurs.

4. CONCLUSION DE LA PARTIE B

L'agencement des cellules dans l'atelier de fabrication est le principal thème de cette partie. Nous avons posé ce problème d'optimisation combinatoire et proposé une méthode pour le résoudre. Cette méthode est basée sur l'algorithme du recuit simulé et sur une représentation géométrique dédiée à un agencement par affectation des cellules aux surfaces libres de l'atelier.

Nous avons développé, sur cette base, un logiciel d'agencement des systèmes de production (voir la partie C), pouvant être utilisé non seulement pour l'agencement des cellules mais aussi pour l'agencement de tout autre système du type job-shop.

Dans le troisième chapitre de cette partie, nous avons introduit le problème de l'agencement dynamique des systèmes de fabrication. Nous avons

identifié ses deux variantes qui trouvent des applications très intéressantes dans les systèmes cellulaires de fabrication. Plusieurs problèmes reliées à l'aspect aléatoire des données restent sans solutions ou avec des solutions partielles.

PARTIE C

1. Introduction

2. Fonctionnement du logiciel

2.1. Matrice des flux

2.2. Contraintes

2.3. Configuration initiale

2.4. Paramètres de contrôle du recuit simulé

3. Architecture du logiciel

3.1. Arbre des appels

3.2. Description des sous-programmes

4. Résultats

4.1. Résultats intermédiaires

4.2. Résultat final

5. Exemples d'application

5.1. Exemple détaillé

5.2. Série d'exemples

6. Conclusion

1. INTRODUCTION

Ce logiciel permet de faire l'agencement des ressources dans un atelier de fabrication du type job-shop. Il est utilisé également dans le processus de résolution du problème d'agencement dynamique des systèmes de fabrication discuté dans la partie B. Nous prévoyons dans ce logiciel, en plus de la partie centrale qui résout le problème d'agencement proprement dit, deux interfaces. La première interface permet à l'utilisateur d'introduire les données utilisées par le système. La seconde interface permet à l'utilisateur d'intervenir pour modifier la configuration proposée par le logiciel à la fin du processus du recuit simulé. Il a alors la possibilité de modifier ce résultat manuellement, de relancer le processus pour obtenir un résultat meilleur, ou d'arrêter les calculs si le résultat est satisfaisant. Le système visualise à l'écran tous les résultats intermédiaires et finaux.

Ce programme a été réalisé en langage C sur Sun / SPARC330. Nous avons utilisé les bibliothèques standards du langage C qui sont : ctype.h, stdio.h, math.h et time.h et nous avons programmé les procédures suivantes (dans l'ordre alphabétique) :

- act_ope.c
- agencement21.c
- calcflux2.c
- calcout3.c
- cheminc1.c
- confinit_contr3.c
- contraintes.c
- datas1.c
- distmin.c
- give_mat_trafic1.c
- manip_res.c
- modif_cont.c
- param_control.c
- platini_contr1.c
- rec_conf.c
- visu_ways.c
- visu_ways1.c
- zone_libre_cont.c

dont l'exécutable est mk_agenaticdem.

2. FONCTIONNEMENT DU LOGICIEL

Nous décrivons dans ce qui suit le fonctionnement de ce logiciel, en commençant par une description globale du déroulement du processus résumé par l'organigramme présenté dans la figure C1 ci-dessous. La partie centrale de cet organigramme représente le processus du Recuit Simulé (RS). Ce processus est entouré par les interfaces et procédures suivantes : l'acquisition des données, la construction ou l'acquisition d'une configuration initiale, et en fin l'intervention de l'utilisateur.

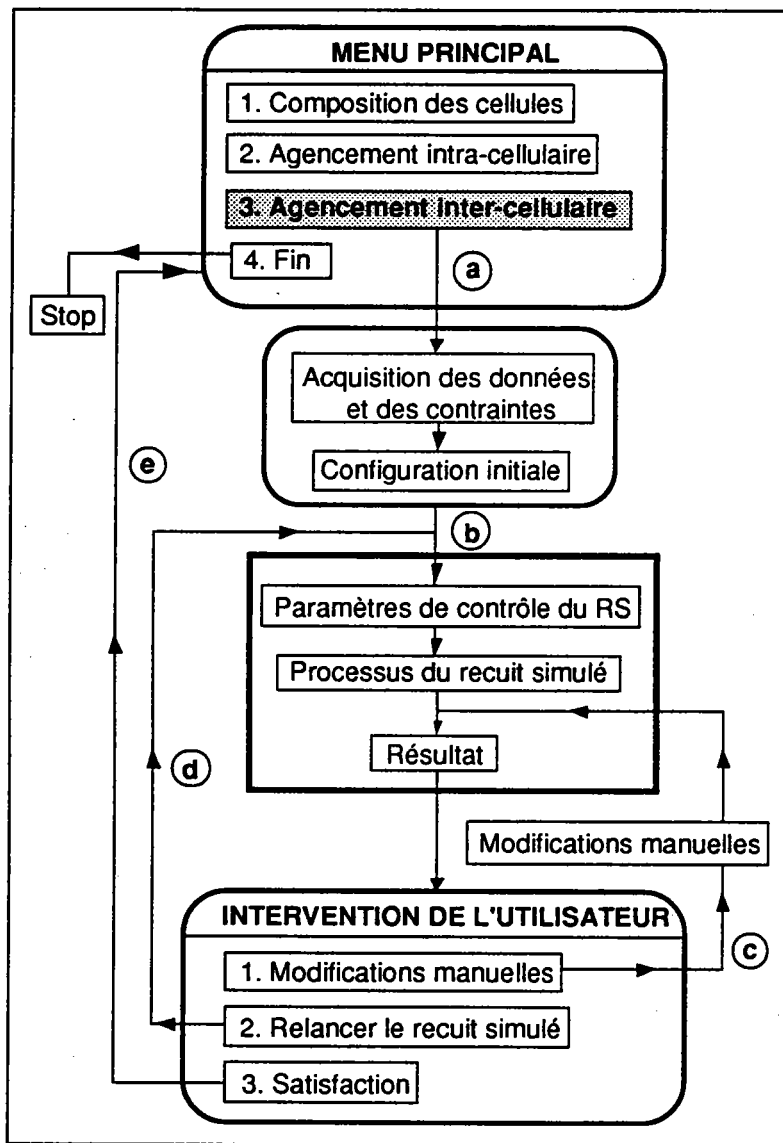


Figure C1 : Organigramme du fonctionnement du programme d'agencement inter-cellulaire

Nous décrivons le déroulement de la procédure comme suit :

Etant donné un atelier de fabrication doté d'un parc de cellules (ou ressources en général), le système commence par suivre le chemin (a) représenté dans la figure C1. Il demande à l'utilisateur de lui fournir les données (les dimensions de l'atelier et des cellules, la matrice des flux inter-cellulaires et les contraintes). Il passe ensuite à la construction ou à l'acquisition d'une configuration initiale. En suivant le chemin (b), le système passe à l'exécution du processus de résolution du problème qui se fait en trois étapes ; l'acquisition des paramètres de contrôle du RS, le RS proprement dit, et la présentation du résultat. Alors, le système donne à l'utilisateur la possibilité d'intervenir. Trois décisions sont possibles : (1) faire quelques modifications dans la configuration fournie par le RS (chemin c), (2) relancer le processus du RS (chemin d), et (3) satisfaction et retour au menu principal où l'utilisateur peut faire un autre agencement ou arrêter.

Nous allons, dans ce qui suit, suivre le fonctionnement du logiciel en s'attardant sur chaque étapes de ce processus.

Les données nécessaires sont :

- La matrice des flux inter-cellulaires. Cette matrice peut être donnée par l'utilisateur, ou bien donnée par le module de formation des cellules, ou bien calculée dans ce module.
- Les contraintes qui ne peuvent être fournies que par l'utilisateur, car lui seul peut les déduire des caractéristiques des cellules. L'automatisation de cette opération nécessiterait le développement d'un système expert.
- Les données géométriques concernant l'atelier, les ressources de fabrication, et le moyen de transport utilisé. Dans le cas où ces ressources sont des cellules, leurs dimensions et les positions de leurs entrées/sorties sont directement fournies par le module de l'agencement intra-cellulaire.

2.1. Matrice des flux inter-cellulaires

Le processus débute par la donnée de la matrice des flux inter-cellulaires. L'utilisateur a le choix entre introduire lui même cette matrice élément par élément, ou la faire calculer par le système à partir du fichier des gammes et de la composition des cellules. Dans ce cas, le système lui demande les données suivantes, qu'il peut fournir manuellement ou prendre dans un fichier :

- Les routages des produits à travers les machines du système,

- les compositions des cellules,
- les demandes de produits,
- et les tailles des lots de produits transportés entre les cellules.

Les données, lorsqu'elles sont fournies manuellement, sont sauvegardées automatiquement dans un fichier. Il en est de même pour la matrice des flux une fois qu'elle est calculée. La figure C2 résume cette opération entièrement.

2.2. Contraintes

Les contraintes du problème sont décrites dans le paragraphe 2.2.2 de la partie B de la thèse. Lorsqu'elles existent, elles sont données par l'utilisateur, manuellement ou à partir d'un fichier. La figure C3 résume la manière dont ces contraintes sont acquises par le système.

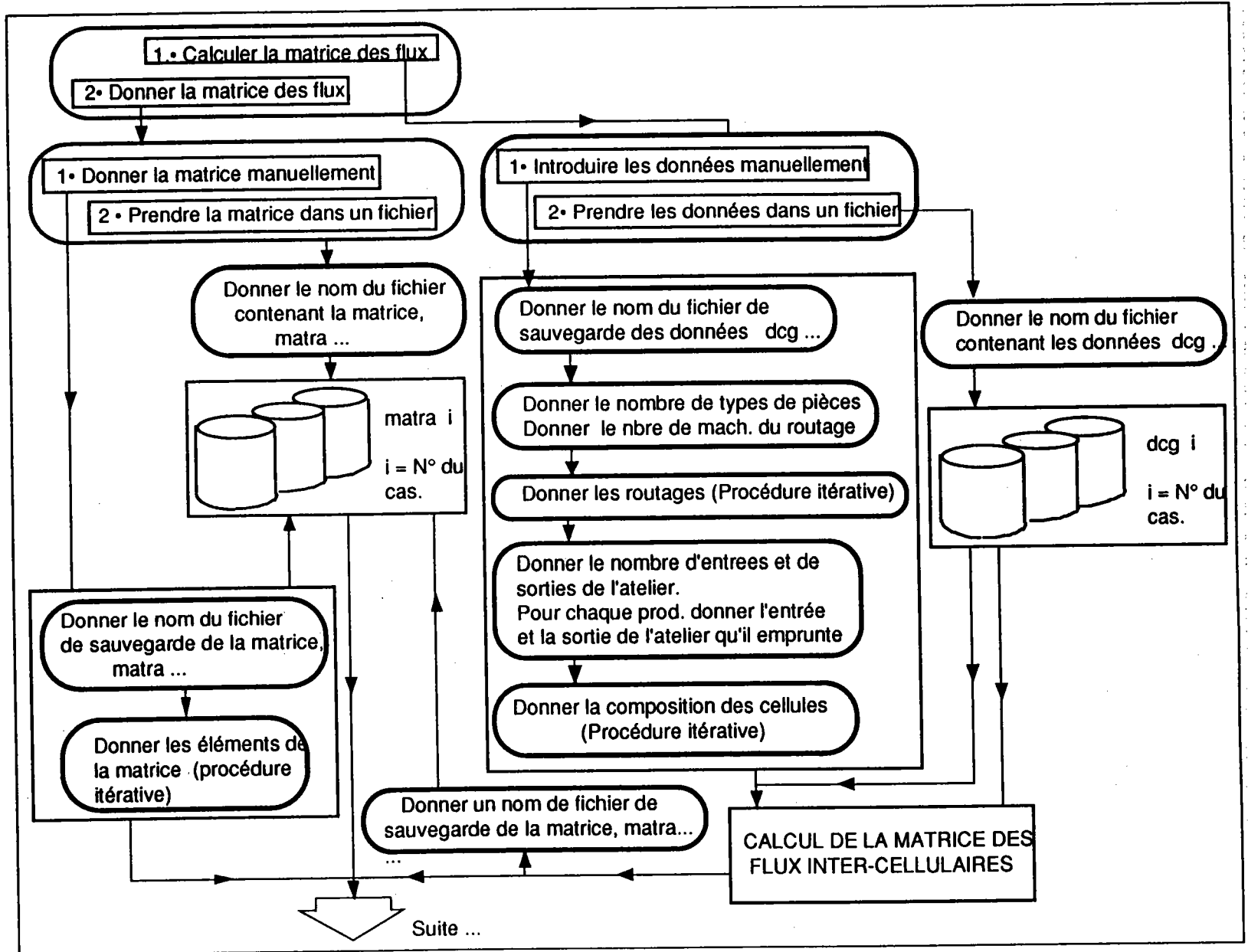
2.3. Configuration initiale

La configuration initiale peut être donnée par l'utilisateur, manuellement ou à partir d'un fichier. Elle peut également être construite par le système et l'utilisateur d'une manière inter-active. Dans le cas où l'utilisateur préfère donner lui même une configuration initiale pré-construite, le système lui demande de fournir les dimensions de l'atelier et les positions de ses entrées/sorties. Il lui demande aussi de donner les dimensions et les positions des zones interdites. Les dimensions des cellules ainsi que les positions de leurs entrées/sorties sont extraites par le système au fur et à mesure de leur installation. Si l'utilisateur décide de construire la configuration initiale avec le système, celui-ci lui demande, en plus des dimensions de l'atelier et celles des zones interdites, les dimensions des cellules, et ceci avant d'entamer le processus de construction. Le placement des cellules se fait alors de la manière suivante :

Le système choisit une cellule ainsi qu'un rectangle libre sur la surface de l'atelier pouvant la recevoir et les propose à l'utilisateur. Si celui-ci refuse cette proposition, il aura la possibilité de choisir et d'installer une cellule lui même.

Les figures C4, C5, C6 et C7 décrivent ce processus.

Figure C2 : Calcul ou acquisition de la matrice des flux inter-cellulaires



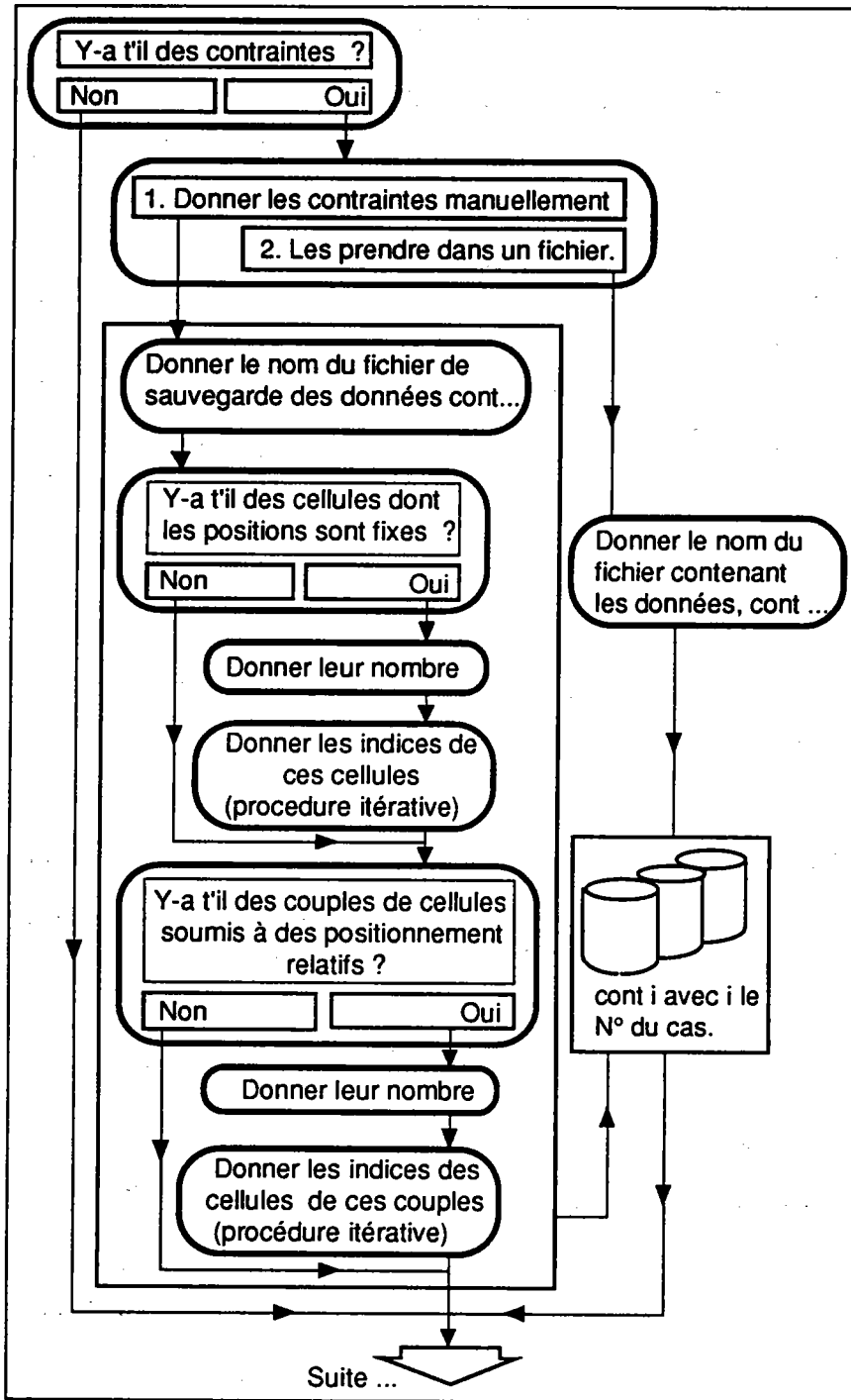


Figure C3 : Les contraintes

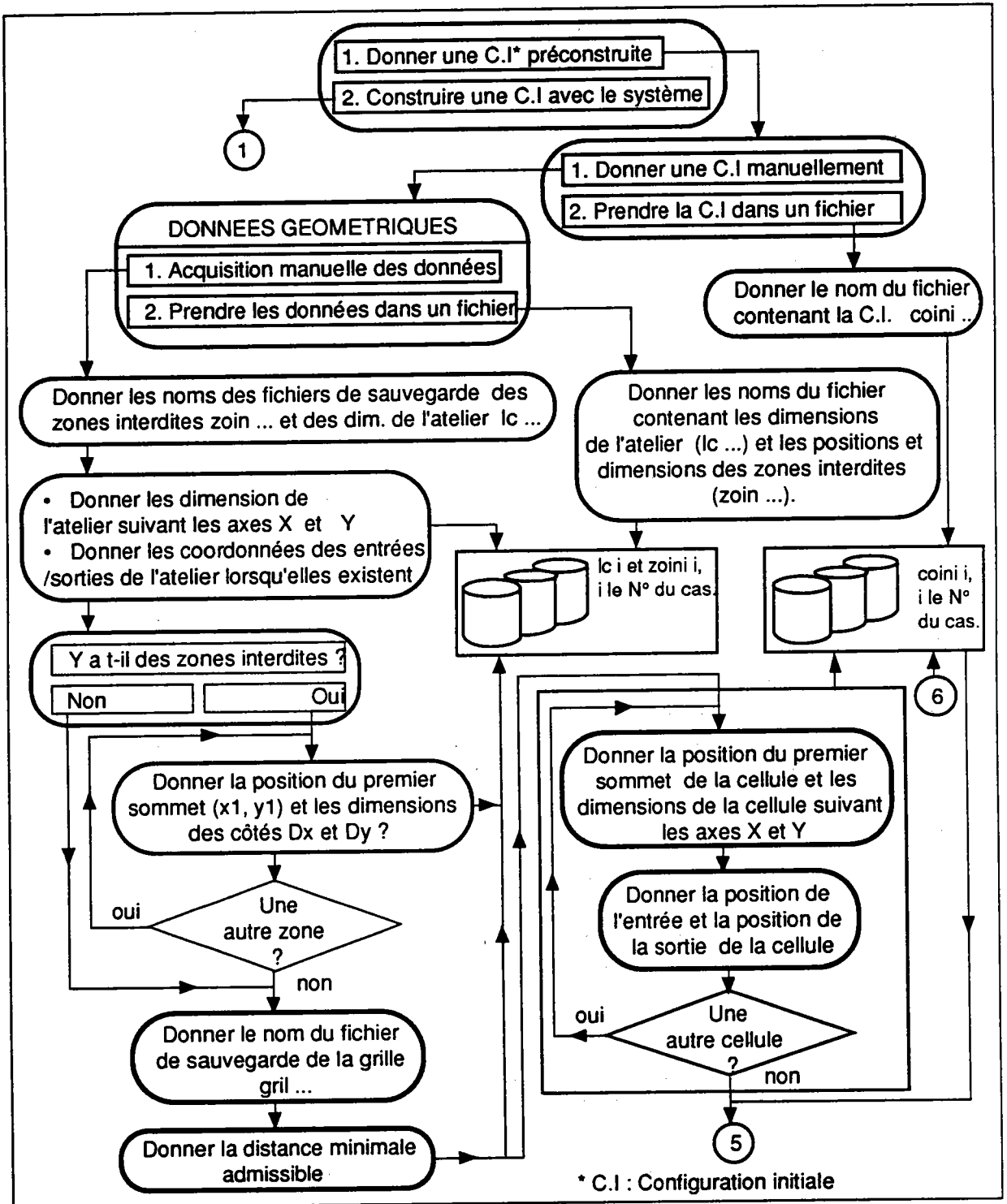


Figure C4 : Une configuration initiale donnée par l'utilisateur

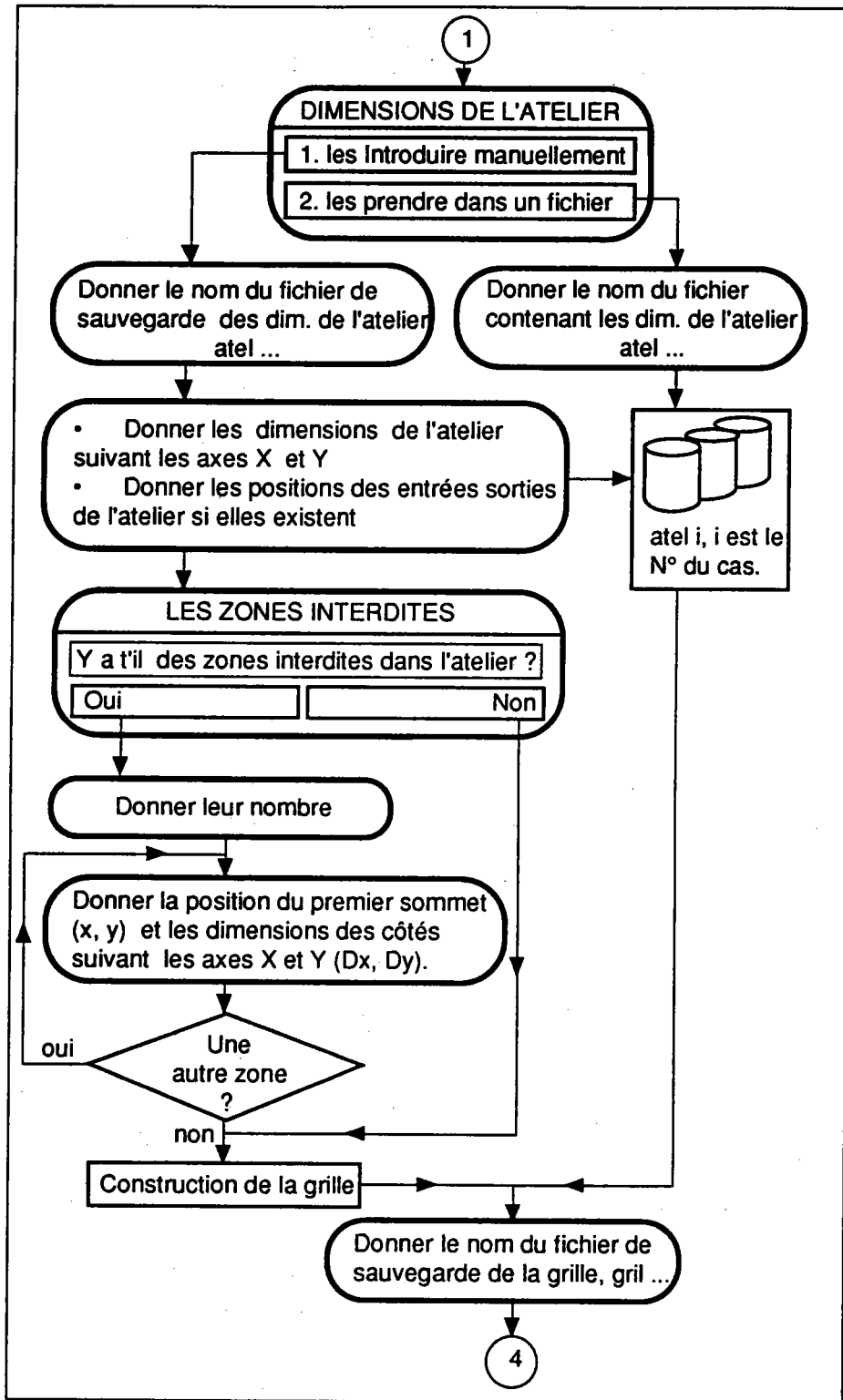


Figure C5 : Acquisition des données géométriques concernant l'atelier

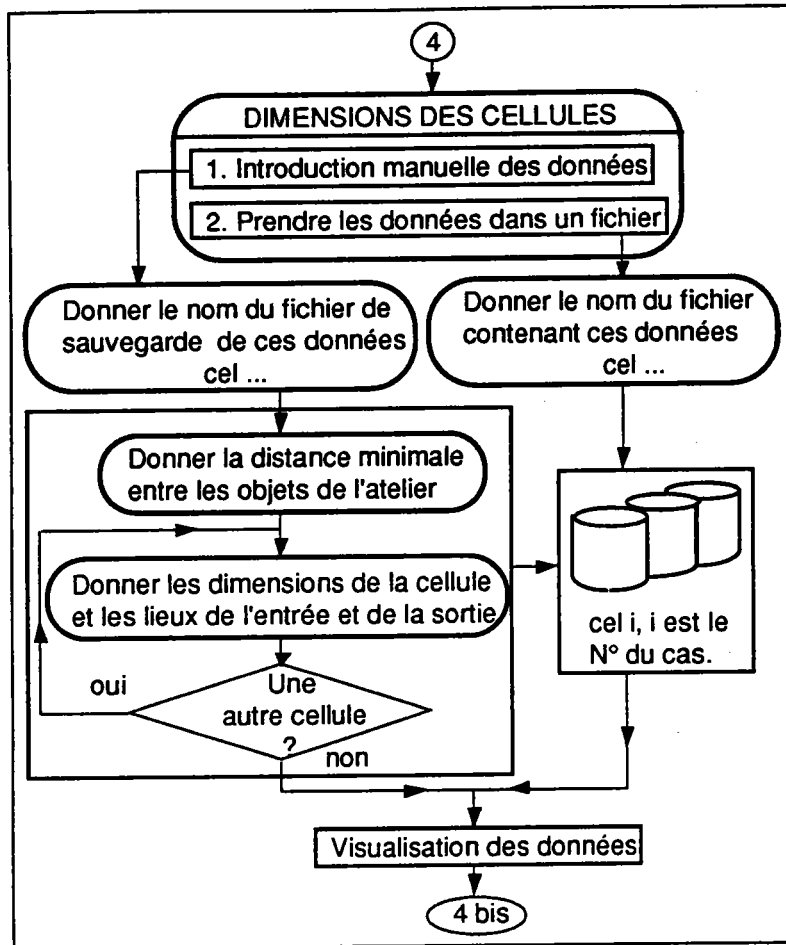


Figure C6 : Dimensions des cellules et positions de leurs entrées/sorties

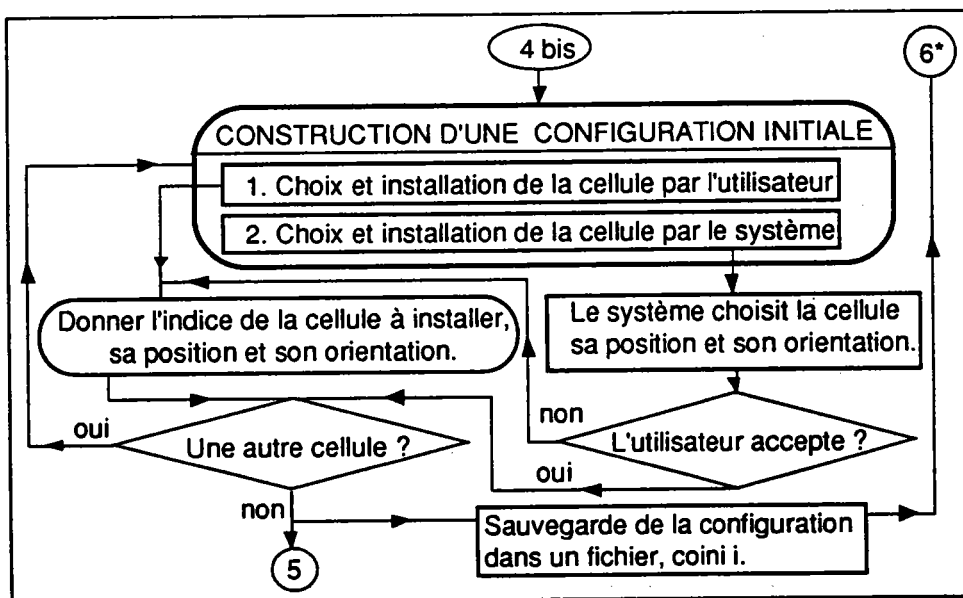


Figure C7 : Construction d'une configuration initiale par le système et l'utilisateur

2.4. Paramètres du recuit simulé

Pour déclencher le processus du recuit simulé, le système doit disposer des paramètres permettant son contrôle. Ces paramètres sont soit fournis par l'utilisateur, manuellement, soit lus dans un fichier, soit choisis par le système en fonction de l'importance et de la complexité du problème (voir annexe A).

La figure C8 illustre le choix des paramètres. Pour la commodité de la représentation, nous avons ajouté dans cette figure un bloc représentant le processus du RS, bien qu'il n'en fasse pas partie.

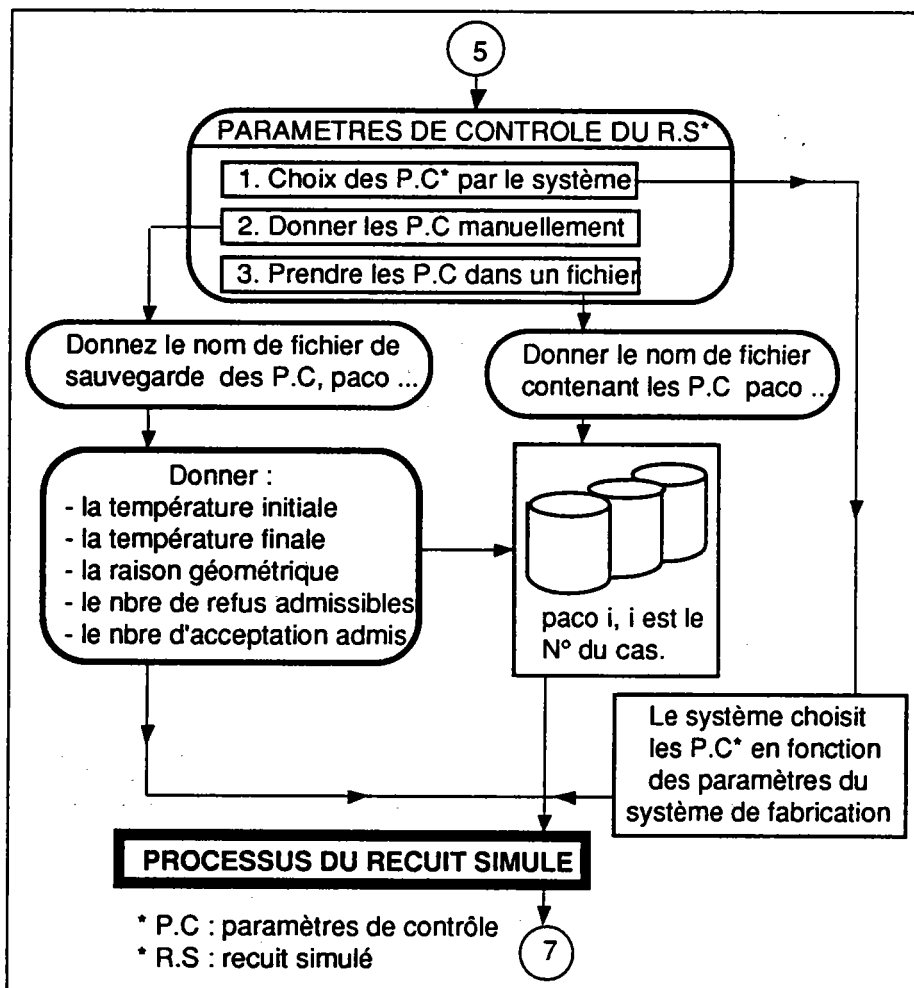


Figure C8 : Paramètres de contrôle du recuit simulé

2.5. L'intervention de l'utilisateur

A la fin du processus du recuit simulé, le système présente à l'utilisateur la configuration correspondant à la plus petite valeur du critère trouvée.

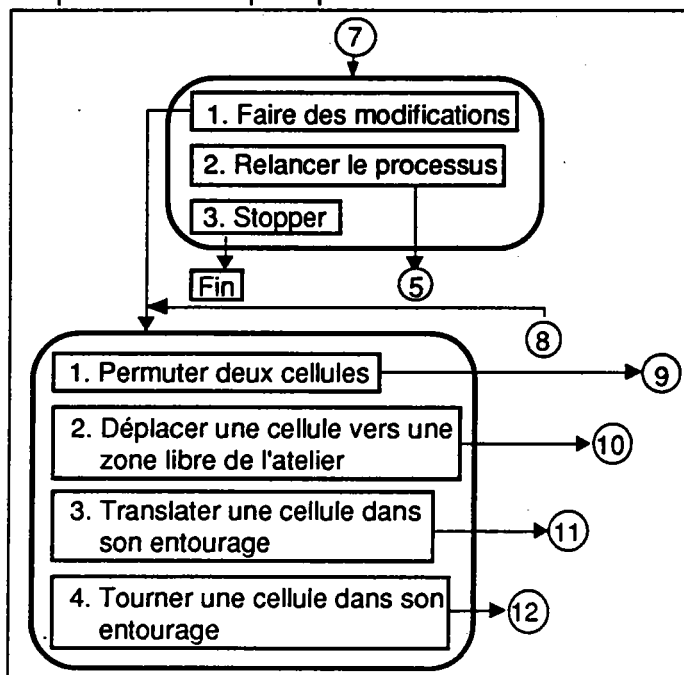


Figure C9 : Intervention de l'utilisateur (menu principal)

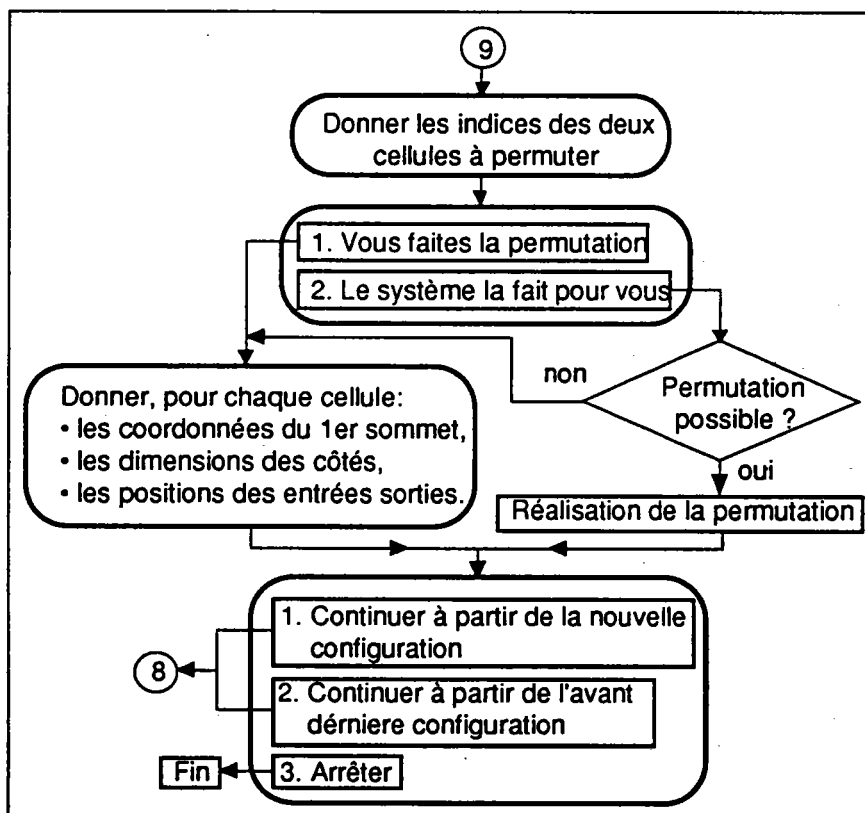
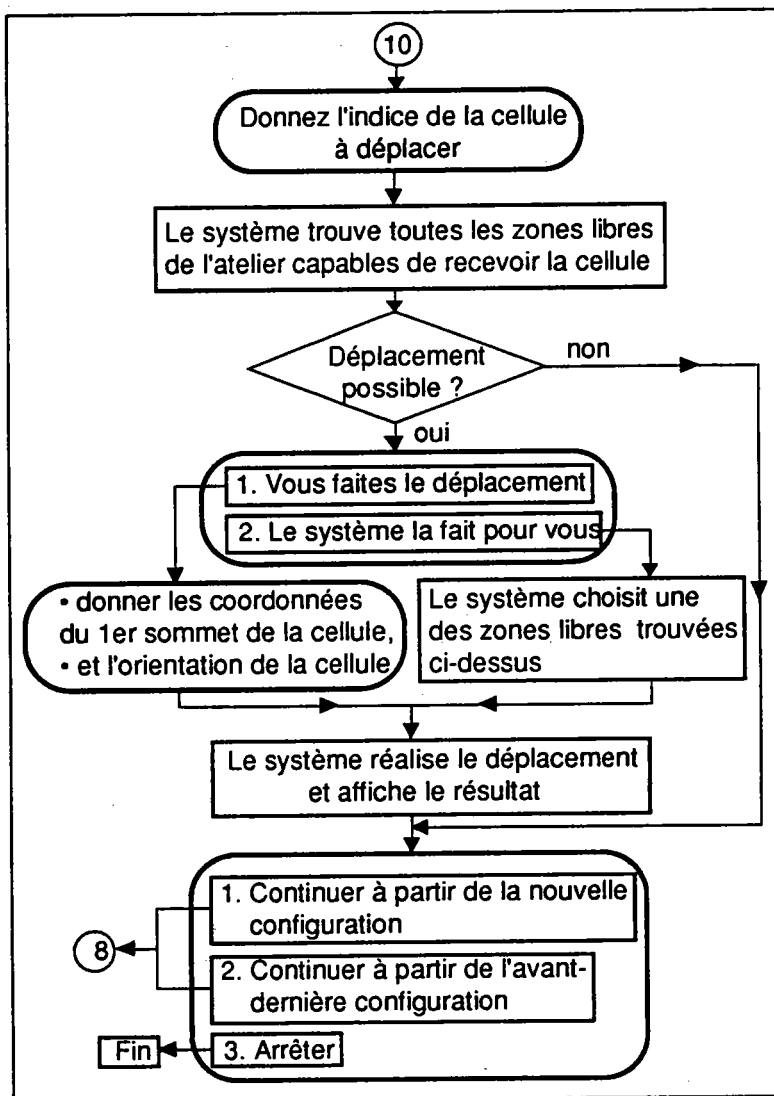


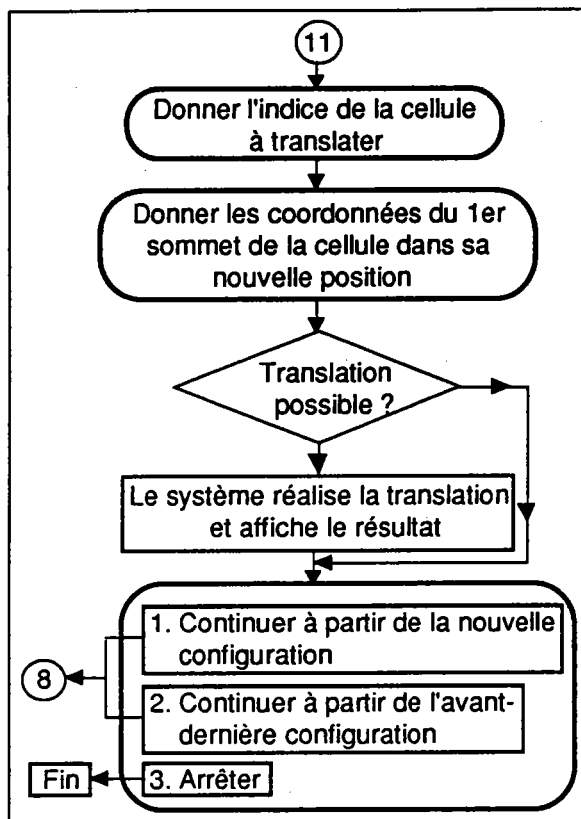
Figure C10 : Permutation de deux cellules

Il lui donne, immédiatement après, la possibilité d'intervenir pour améliorer la configuration trouvée, en lui proposant une palette de transformations élémentaires : la permutation de deux cellules, le déplacement d'une cellule vers une zone libre de l'atelier, la translation d'une cellule dans son entourage libre ainsi que la rotation d'une cellule dans son entourage.

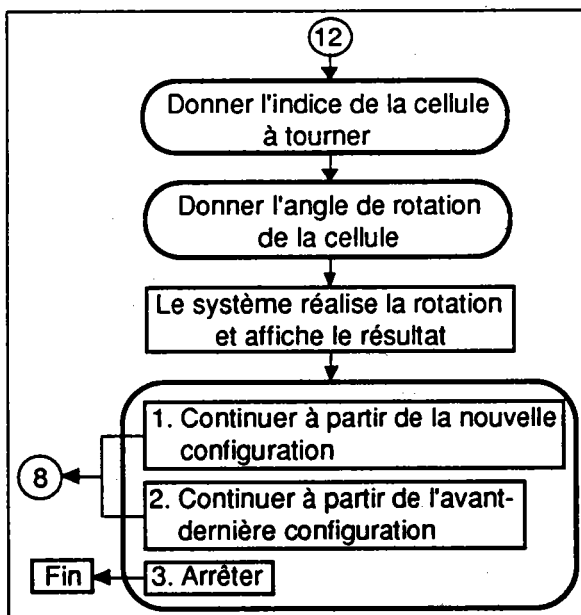
A la fin de chaque transformation élémentaire et en fonction de la valeur du critère de la nouvelle configuration, l'utilisateur peut continuer les modifications à partir de la dernière configuration ou à partir de la précédente. Les figures C9, C10, C11, et C12 illustrent cela.



FigureC11 : Déplacement d'une cellule vers une zone libre de l'atelier



FigureC12 : Translation d'une cellule dans son entourage libre



FigureC13 : Rotation d'une cellule dans son entourage libre

3. ARCHITECTURE DU LOGICIEL

Nous donnons dans ce qui suit l'architecture du logiciel. La figure C14 présente l'arbre des appels entre les différentes procédures. Nous donnons le rôle global de chaque procédure ainsi que le rôle de chacun de ces paramètres.

3.1. Arbre des appels

L'arbre des appels (figure C14) illustre les liens qui existent entre les différents procédures du programme. Certaines procédures sont utilisées à plusieurs reprises dans le programme et sont suivies par le même groupe de procédures. Pour éviter l'encombrement de l'arborescence, nous tronquons cette suite et nous la remplaçons par une étoile.

3.2. Description des sous-programmes

Nous expliquons ci-dessous la fonction de chaque sous-programme et donnons la signification de ses paramètres. Les sous-programmes sont listés dans l'ordre alphabétique.

A

actopera() : Procédure permettant l'intervention de l'utilisateur à la fin du processus du RS. Elle permet à l'utilisateur de choisir entre relancer le processus, faire des modifications dans la meilleure configuration obtenue, ou arrêter.

adj_a_gauche(k, i) : Procédure de détermination d'un bloc adjacent à gauche à un bloc donné. k est le numéro d'une série de mailles élémentaires de même couleur sur la ligne i (voir calcul des chemins, partie B).

adj_dessous(k, r, i) : Procédure de détermination d'un bloc adjacent en dessous d'un bloc donné. k est le numéro d'une série de mailles élémentaires de même couleur (occupées ou libres) de la ligne i . r est le numéro d'une autre série de la ligne $i + 1$ (voir calcul des chemins, partie B).

adj_dessus(k, r, i) : Procédure de détermination d'un bloc adjacent au dessus d'un bloc donné. k est le numéro d'une série de la ligne i et r celui d'une série de la ligne $i + 1$ (voir calcul des chemins, partie B).

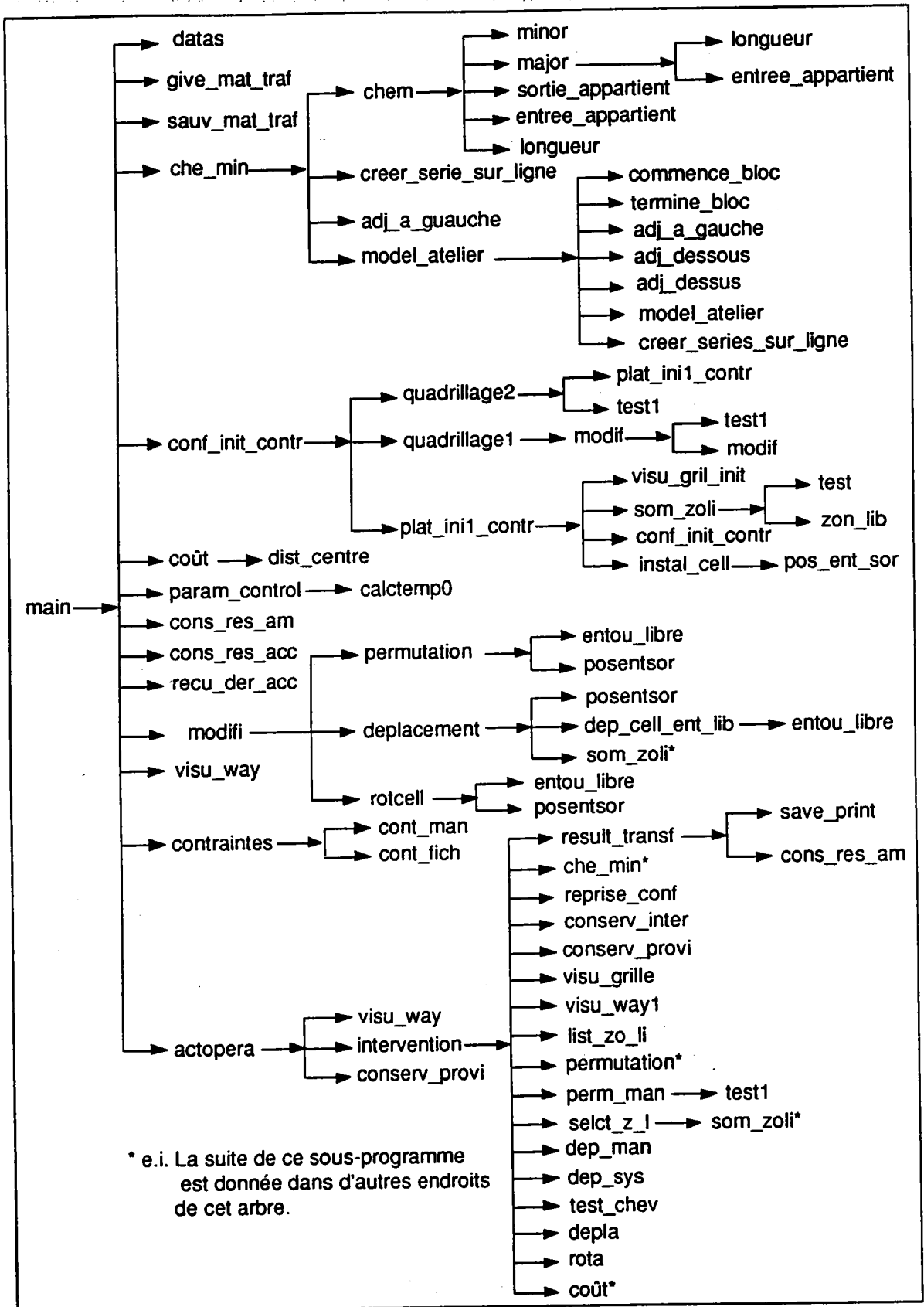


Figure C14 : Arbre d'appel du logiciel d'agencement inter-cellulaire

C

calctemp0() : Procédure de calcul des paramètres de contrôle du RS par défaut.

chem() : Procédure principale de détermination des chemins inter-cellulaires les plus courts, après la modélisation géométrique de l'atelier.

che_min() : Procédure secondaire de modélisation de l'atelier et de détermination de chemins inter-cellulaires.

commence_bloc(k, i) : Procédure de localisation de la première série de mailles élémentaires d'indice k d'un bloc, se trouvant sur la ligne i de la grille (voir calcul des chemins, partie B).

conf_init_contr() : Procédure de construction ou d'acquisition d'une configuration initiale construite en tenant compte des contraintes du problème.

conserv_inter(adad) : Procédure de conservation d'une configuration intermédiaire pouvant éventuellement être reprise dans l'étape suivante. adad est le numéro de la configuration conservée.

cons_res_acc() : Procédure de sauvegarde d'une configuration acceptée (qu'elle améliore la valeur du critère ou qu'elle ne l'améliore pas) lors du processus du recuit simulé.

cons_res_am() : Procédure de conservation d'une configuration qui améliore la valeur du critère lors du processus du recuit simulé.

cont_fich() : Procédure de récupération des contraintes dans un fichier.

cont_man() : Procédure d'introduction manuelle des contraintes.

contraintes() : Procédure d'acquisition des contraintes de positionnement définitif de certaines cellules ou relatif entre certaines cellules.

coût() : Procédure de calcul de la valeur du critère pour une configuration donnée en tenant compte des contraintes fonctionnelles du système de fabrication considéré.

creer_serie_sur_ligne(i) : Procédure de répartition d'une ligne en séries de mailles élémentaires de même couleur. i est le numéro de la ligne (voir calcul des chemins, partie B).

D

datas() : Procédure d'acquisition des données concernant les routages des produits et la composition des cellules .

dep_cell_ent_lib(c) : Procédure de réalisation de la transformation élémentaire de translation d'une cellule c dans son voisinage libre par le système.

depla(c, x, y) : Procédure de translation d'une cellule c vers une position où son premier sommet se trouverait en (x, y).

dep_man(c) : Procédure de translation d'une cellule c vers une zone libre. L'opération est réalisée par l'utilisateur.

dep_sys(c, z) : Procédure de translation d'une cellule c vers la zone libre z. L'opération est réalisée par le système.

deplacement(c) : Procédure de réalisation de la transformation élémentaire de déplacement d'une cellule c. L'opération est réalisée par le système.

dist_centre(i, j) : Procédure de calcul de la distance entre les centres de deux cellules i et j soumises à une contrainte de positionnement relatif.

E

entree_appartient(i, bloc_adj) : Cette procédure retourne un 1 si l'entrée de la cellule i se trouve dans le bloc "bloc_adj" qui lui est adjacent (voir calcul des chemins, partie B).

entou_libre(c) : Procédure de détermination de l'entourage libre de la cellule d'indice c (voir annexe D).

G

give_ma_traf() : Procédure d'acquisition de la matrice des flux inter-cellulaires, manuellement ou à partir d'un fichier.

I

instale_cell(k, c, x, y, d_i_c_min, L_c, l_c, or_c, or_z) : Procédure d'installation d'une cellule d'indice c sur une zone libre de l'atelier. k est le nombre de cellules installées, d_i_min la distance minimale entre les objets se trouvant dans l'atelier, L_c la longueur de la cellule, l_c la largeur de la cellule, or_c l'orientation de la cellule et or_z l'orientation de la zone libre.

intervention() : Procédure d'intervention de l'utilisateur à la fin du recuit simulé pour faire des modifications ou pour relancer le processus du RS.

L

list_zo_li(pm) : Procédure de détermination de la liste des zones libres pouvant recevoir une cellule donnée. pm est le nombre de zones libres possibles.

longueur(r, s) : Procédure de calcul de la longueur du chemin entre deux blocs adjacents r (bloc de départ) et s (bloc d'arrivé) (voir calcul des chemins, partie B).

M

main() : Programme principal d'agencement de cellules dans un atelier de fabrication.

major(i, bloc_i, r) : Procédure de calcul de la borne supérieure de la distance du plus court chemin entre deux points donnés. i est l'indice du bloc de départ, bloc_i est l'indice du bloc qui lui est adjacent et qui contient son entrée et r l'indice de la cellule d'arrivée (voir calcul des chemin, partie B).

minor(r) : Procédure de calcul d'une borne minimale pour la distance entre la position courante et l'entrée de la cellule d'indice r (la distance de Manhattan) (voir calcul des chemin, partie B).

model_atelier(k, r, i) : Procédure récursive de modélisation géométrique de l'atelier, qui génère des blocs rectangulaires libres et occupés, à partir des séries libres et des séries occupées (voir calcul des chemin, partie B).

modif() : Procédure récursive de modification ou d'ajout d'une cellule ou d'une zone interdite lors de l'acquisition manuelle d'une configuration initiale.

modifi() : Procédure de passage de la dernière configuration acceptée à une configuration voisine par une modification élémentaire. Opération réalisée par le système.

P

param_control() : Procédure d'acquisition des paramètres de contrôle du recuit simulé.

perm_man(c1, c2) : Procédure permettant à l'opérateur de réaliser la permutation de deux cellules c1 et c2.

permutation(c1, c2) : Procédure de réalisation de la permutation de deux cellules c1 et c2 si cela est possible.

plat_ini1_contr() : Procédure de construction d'une configuration initiale à partir des données géométriques en tenant compte des contraintes.

posensor(cause, c, ent, sor, pos_ent, pos_sor) : Procédure de positionnement de l'entrée et de la sortie d'une cellule d'indice c après une modification élémentaire. "cause" indique si la cellule est à tourner ou pas lors de la modification, ent (resp. sor) est le numéro du côté où se trouve l'entrée

(resp. la sortie), et `pos_ent` (resp. `pos_sor`) la position de l'entrée (resp. la sortie) sur ce côté. Cette procédure est utilisée par le système lors du processus du RS.

pos_ent_sor(c, ent, sor, or_z, or_c) : Procédure de positionnement de l'entrée ou de la sortie d'une cellule `c` après son installation. "ent" (resp. `sor`) est le numéro du côté de la cellule où se trouve l'entrée (resp. la sortie), "or_z" est l'orientation de la zone libre et "or_c" est l'orientation de la cellule. Cette procédure est utilisée lors du processus d'intervention de l'utilisateur.

Q

quadrillage1() : Procédure d'acquisition des données géométriques concernant l'atelier, les zones interdites et de la configuration initiale, manuellement ou à partir d'un fichier.

quadrillage2() : Procédure d'acquisition des données géométriques concernant l'atelier, des zones interdites, et construction d'une configuration initiale d'une manière interactive avec l'opérateur.

R

recu_der_acc() : Procédure de récupération de la dernière configuration acceptée.

reprise_conf() : Procédure permettant la reprise de l'avant dernière configuration lors du processus d'intervention de l'opérateur.

result_transf() : Procédure donnant les résultats d'une transformation réalisée par l'opérateur.

rota(c, x, y, angle) : Procédure de rotation d'une cellule d'indice `c` d'un angle "angle". Le premier sommet de la cellule sera en (x, y) . Cette opération est commandée par l'opérateur.

rotcell(c) : Procédure de réalisation de la transformation élémentaire de rotation d'une cellule `c`. Cette opération est réalisée par le système.

S

sauv_mat_traf() : Procédure de sauvegarde de la matrice des flux inter_cellulaires dans un fichier.

save_print() : Procédure de sauvegarde d'une configuration dans fichier afin de l'imprimer.

select_z_l(c) : Procédure de sélection des zones libre pouvant accueillir la cellule d'indice `c`.

som_zoli() : Procédure de détermination d'un des sommets de chacune des zones libres de l'atelier (voir annexe C).

sortie_appartient(i, bloc_adj) : Cette procédure retourne un 1 si la sortie de la cellule d'indice i se trouve dans le bloc "bloc_adj" qui lui est adjacent, et 0 sinon.

T

test(m, n) : Procédure permettant de vérifier, lors de la détermination des zones libres, si le sommet de cette zone a déjà été trouvé. (m, n) sont les coordonnées du sommet.

test1(pt_chaine) : Procédure de test de la nature d'une chaîne de caractères "pt_chaine".

test_chev(c, x, y) : Procédure permettant de vérifier si le point (x, y) se trouve sur une zone occupée. Ce test se fait lors du déplacement d'une cellule d'indice c.

termine_bloc(r, i) : Procédure de détermination de la dernière série r de mailles de même couleur, d'un bloc donné qui se trouve sur la ligne i de la grille.

V

visu_grille() : Procédure de visualisation de la grille de la configuration initiale.

visu_gril_init(K) : Procédure de visualisation de la grille couvrant l'atelier avant et après toute modification lors de l'intervention de l'utilisateur. K est le nombre de cellules déjà installées dans l'atelier.

visu_way() : Procédure de visualisation des chemins entre les cellules de la meilleure configuration obtenue à l'issu du processus du recuit simulé.

visu_way1() : Procédure de visualisation des chemins entre les cellules de la configuration obtenue après intervention de l'opérateur.

Z

zon_lib() : Procédure de détermination des zones libres, connaissant leur premier sommet (voir annexe C).

4. RESULTATS

Le système fournit non seulement le résultat final, mais aussi des résultats intermédiaires tout au long du déroulement du processus de résolution du problème.

4.1. Résultats intermédiaires

Lors du déroulement du processus d'acquisition des données, le système les montre à l'utilisateur afin de lui permettre de vérifier leur exactitude. Le système donne également des résultats intermédiaires durant le déroulement du processus de résolution du problème. Ces résultats sont :

- a. Le système visualise la matrice des flux inter-cellulaires après l'avoir calculée.
- b. Lors de la construction de la configuration initiale, le système visualise l'atelier après chaque opération d'installation de cellule.
- c. La configuration initiale étant construite, le système visualise, à la demande de l'opérateur, tout chemin reliant les entrées/sorties des cellules.
- d. Tout au long du processus du RS, le système visualise les valeurs des paramètres permettant le suivi de l'évolution du processus de recherche de la solution optimale. Il donne, dans l'ordre :
 - le numéro de l'itération,
 - la valeur du critère de la configuration,
 - la plus petite valeur du critère obtenue jusque là,
 - le durée de la transformation élémentaire,
 - le durée du calcul des chemins de cette configuration,
 - et le temps total depuis le début des calculs.
- e. Après avoir fourni un résultat et lors du processus d'intervention de l'opérateur, le système :
 - visualise l'atelier,
 - donne la valeur du critère pour la configuration obtenue,
 - donne les taux d'amélioration ou de détérioration de la solution par rapport à la configuration initiale et par rapport au meilleur résultat précédant.

4.2 Résultat final

Comme résultat final, le système donne :

- une visualisation de la meilleure configuration de l'atelier,
- la valeur du critère correspondant à cette configuration,
- les taux d'amélioration de cette valeur par rapport à la configuration initiale,
- le nombre d'itérations,
- le temps de traitement total,
- la possibilité d'impression de ce résultat.

5. EXEMPLES D'APPLICATION

Nous commençons par traiter entièrement un petit exemple illustratif à 5 cellules. Nous donnerons ensuite les résultats du traitement de 4 autres exemples avec 10, 15, 20 et 25 cellules. Les configurations initiales et les matrices des flux sont données dans l'annexe E.3.1. et les configurations des solutions finales sont données dans l'annexe E.3.2.

5.1. Exemple détaillé

Dans ce qui suit, nous présentons une application de notre logiciel sur un petit système de fabrication que nous avons choisi à titre illustratif.

Nous décrivons la manière dont sont préparées les données fonctionnelles et géométriques ainsi que les contraintes. Nous montrons ensuite comment est construite la configuration initiale par le système. Enfin, nous donnons le résultat final.

5.1.1. Présentation du système de fabrication

(préparation des données)

a. Les données fonctionnelles

Notre système comporte 12 machines réparties en 5 cellules. 10 types de produits y sont fabriqués. La composition des cellules est donnée dans le tableau C1 et celle des routages des produits dans le tableau C2 (le rapport demande/taille du lot, de tous les produits, est égal à 1).

# de cellule	Composition		
1	1	12	4
2	2	9	6
3	7	10	
4	3	11	
5	5	8	

Tableau C1 : Composition des cellules

Indice du produit	Routage					Entrée dans l'atelier	Sortie de l'atelier
1	5	3	11	1	12	1	1
2	2	6	7			0*	0
3	2	11	9	6	10	7	2
4	6	2	12	11		2	2
5	5	8				1	0
6	1	12	4	7	10	2	1
7	4	1	6	11	3	2	2
8	5	8	2			1	0
9	9	2	11			0	1
10	6	11	12			2	0

Tableau C2 : Routages des produits

Dans ce qui suit, nous décrivons les données géométriques de l'atelier.

b. Les données géométriques

b.1. L'atelier

Les figures C15 et C16 montrent la transition entre l'atelier réel tel qu'il est donné et le modèle tel qu'il est utilisé par le système. L'atelier a une longueur de 25 unités de longueur (u) et une largeur de 20 u. Il dispose de 2 entrées et 2 sorties. Les coordonnées de ces points sont données dans le tableau C3 par rapport à un repère orthonormé (voir figure C16).

	Abscisse	Ordonnée
Entrée 1	18	1
Entrée 2	18	20
Sortie 1	5	1
Sortie 2	5	20

Tableau C3 : Coordonnées de entrées/sorties de l'atelier

Cet atelier contient 4 zones interdites dont les dimensions et les positions sont données dans le tableau C4.

Indice de la zone	Dimension du côté // à l'axe X	Dimension du côté // à l'axe Y	Position du 1er sommet	
			l'abscisse	l'ordonnée
1	2	4	24	1
2	2	5	1	8
3	2	4	24	17
4	5	2	1	19

Tableau C4 : Dimensions et positions des zones interdites

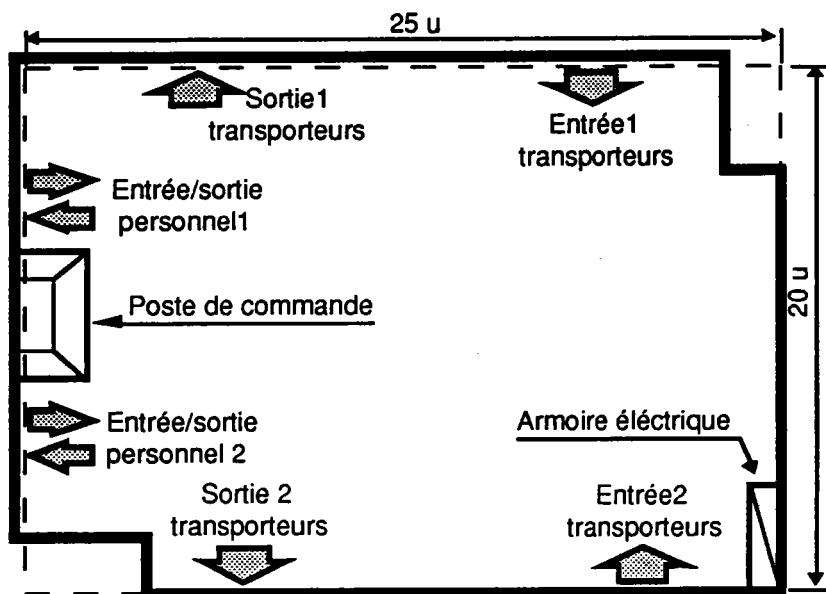


Figure C15 : L'atelier de fabrication vide

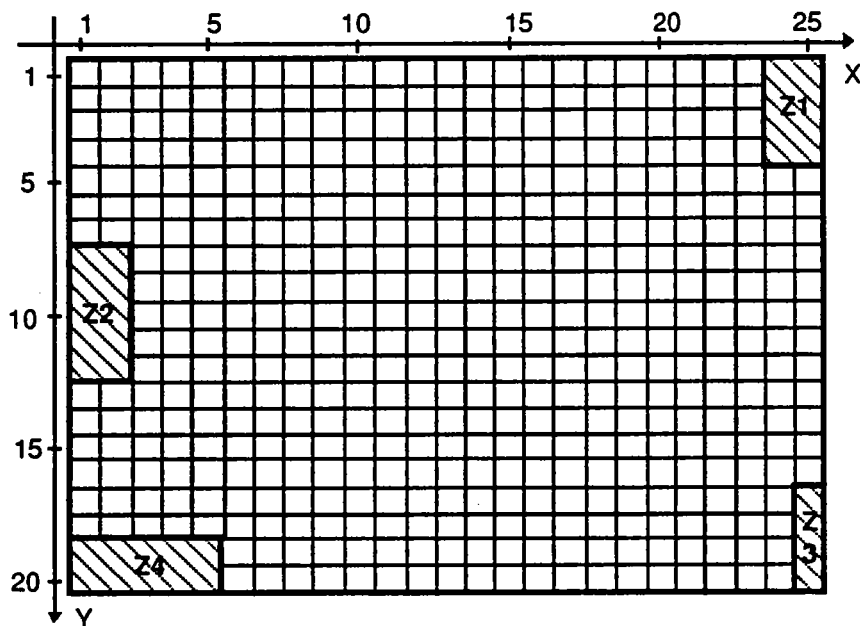


Figure C16 : Modèle de l'atelier de fabrication

b.2. Les cellules

Le passage d'une cellule réelle à sa représentation géométriques telle qu'elle est utilisée par le système est illustré par la figure C17.

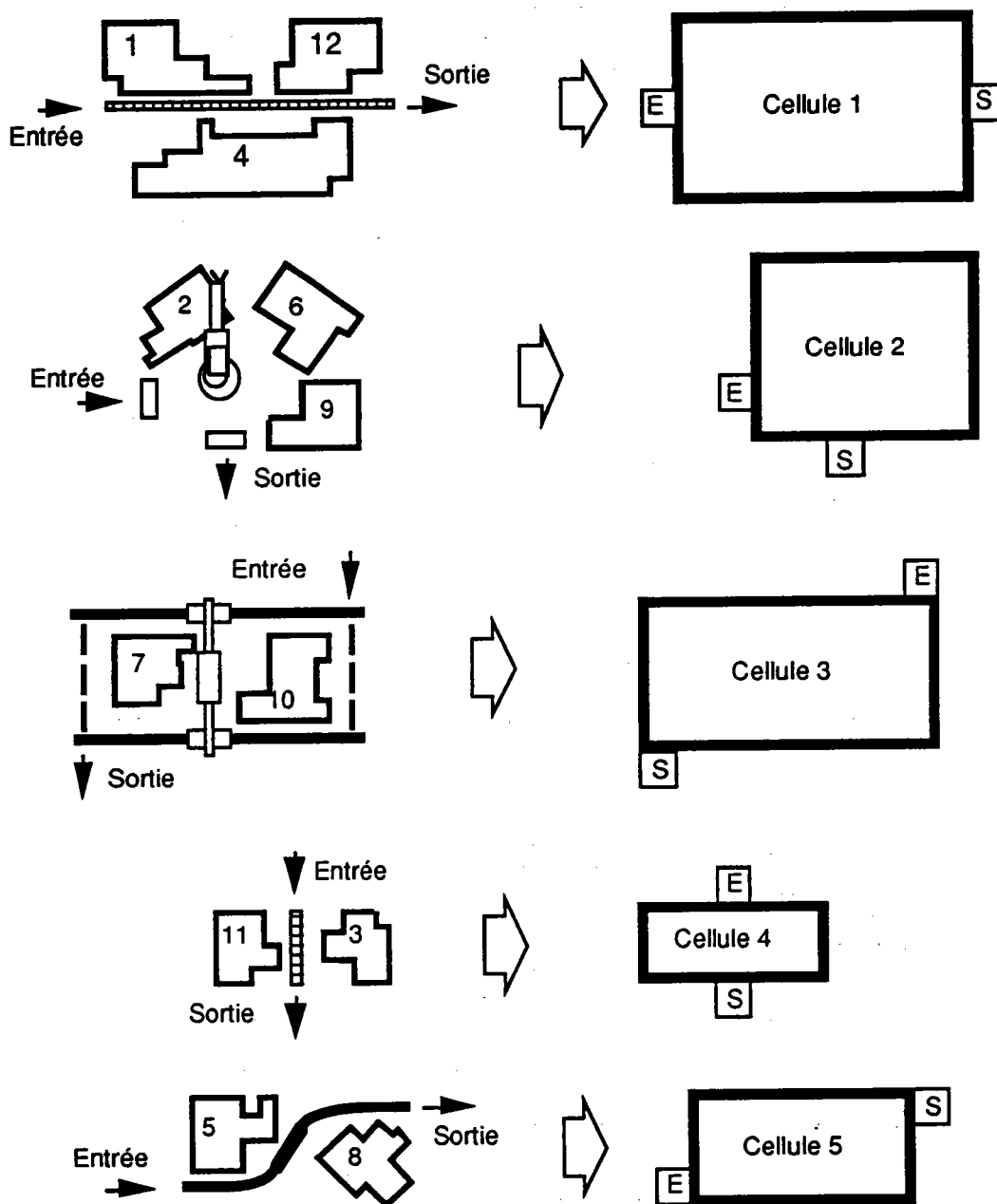


Figure C17 : Passage de la cellule à sa représentation géométrique

Le tableau C5 donne les dimensions des cellules et les positions de leurs entrées/sorties. **Lieu** indique le numéro du côté de la cellule où se trouve l'entrée ou la sortie et **Position** indique le nombre d'unités de longueur (u) entre

la première extrémité de ce côté et la position de cette entrée ou sortie (voir paragraphe 2.3.1 de la partie B).

Cellule	Dimension du côté // à l'axe X	Dimension du côté // à l'axe Y	Entrée		Sortie	
			Lieu	Position	Lieu	Position
1	8	5	4	3	2	3
2	6	5	4	2	3	4
3	8	4	1	8	3	8
4	5	2	1	3	3	3
5	6	3	4	1	2	1

Tableau C5 : Dimensions des cellules et positions de leur entrée/sortie

b.3. Les contraintes

Ce système est soumis à une contrainte de positionnement définitif et à une contrainte de positionnement relatif :

- La cellule 5 contient la machine 5 qui est une machine de réception, de lavage, et d'écroûtage de brut. Cette cellule doit être mise devant l'entrée 1 de l'atelier et ne doit pas être déplacée lors du processus de résolution du problème. Elle est orientée de sorte que son entrée soit en face de l'entrée de l'atelier. Sa position et son orientation sont données dans le tableau C6.

Remarquons que l'orientation de cette cellule change par rapport à son orientation initiale dans les données (rotation de 90°).

Cellule	Position du 1er sommet		Orientation de la cellule			
	Abscisse	Ordonnée	Sommet 1 et 2 sur la même		1 après 2	2 après 1
			horizontale	verticale		
5	21	3		*		*

Tableau C6 : Contrainte de positionnement définitif

- Il est souhaitable aussi que les cellules 3 et 5 soient installées le plus près possible l'une de l'autre, sans que la distance entre leur centres ne dépasse 15 unités de longueur (u). Le poids de cette contrainte est 1 dans l'échelle proposée au paragraphe 2.3 de la partie A. Le tableau C7 récapitule cela.

1ère cellule	2ème cellule	Type de contrainte		Distance max / min	Poids de la contrainte
		Rapprochement	Eloignement		
3	5	*		15	1

Tableau C7 : Contraintes de positionnement relatif entre les cellules

5.1.2. Agencement

Rappelons que, lors de l'exécution du programme, l'utilisateur est entièrement guidé par le système.

a. Matrice des flux inter-cellulaires

Dans le menu principal, nous choisissons de faire un premier lancement qui nous relie automatiquement au module de calcul de la matrice des flux. Nous choisissons alors l'option permettant le calcul de cette matrice. Le système nous demande d'introduire les données concernant la composition des cellules et les routages des produits. Nous les introduisons manuellement à partir des tableaux C1 et C2.

Le système nous montre les données fournies, pour vérification, puis calcule la matrice des flux qu'il conserve dans un fichier. Nous donnons cette matrice dans l'annexe E.3.a.

b. Acquisition des contraintes

Le système passe ensuite à l'acquisition des contraintes que nous avons décrites ci-dessus. Nous les donnons manuellement à partir des tableaux C6 et C7 et nous les stockons dans un fichier.

c. Configuration initiale

Il passe alors à la configuration initiale. Nous choisissons de la construire avec le système.

c.1. Introduction des données

Le système demande, pour cela, les dimensions de l'atelier, les positions de ses entrées / sorties ainsi que les positions et dimensions de ses zones interdites (voir tableaux C3 et C4). Nous les donnons manuellement et nous les stockons dans un fichier.

Il demande ensuite les dimensions des cellules et les positions de leurs entrées/sorties que nous introduisons manuellement à partir du tableau C5 et stockons dans un fichier.

c.2. Construction de la configuration

Avant de construire une configuration initiale, le système nous demande d'installer la cellule 5 qui est soumise à une contrainte de positionnement définitif. Nous donnons sa position et son orientation à partir du tableau C6.

Nous entamons ensuite l'installation des autres cellules, l'une après l'autre. A chaque fois, nous demandons au système de choisir et d'installer la cellule lui même. Il nous renvoie la main en cas d'impossibilité.

Les cellules sont installées comme suit :

Cellule 1 :	Sommet 1 : (5, 3)	Cellule 2 :	Sommet 1 : (12, 3)
	Sommet 4 : (9, 10)		Sommet 4 : (16, 8)
	Entrée : (7, 11)		Entrée : (15, 9)
	Sortie : (7, 2)		Sortie : (17, 6)
Cellule 3 :	Sommet 1 : (12, 11)	Cellule 4 :	Sommet 1 : (8, 13)
	Sommet 4 : (19, 14)		Sommet 4 : (9, 17)
	Entrée : (19, 10)		Entrée : (7, 15)
	Sortie : (12, 15)		Sortie : (10, 15)

Les coordonnées de la cellule 5 après son installation sont :

Sommet 1 : (5, 3)
Sommet 4 : (9, 10)
Entrée : (7, 11)
Sortie : (7, 2)

La valeur du critère pour la configuration initiale que nous donnons dans l'annexe E.3.a. est de 350.

Le système propose une visualisation des chemins entre les cellules. Nous donnons des exemples de chemins pour la configuration initiale de l'atelier (voir annexe E.3.a).

d. Les paramètres de contrôle du recuit simulé

Nous passons alors aux paramètres de contrôle du recuit simulé. Le système choisit les paramètres suivants :

Température initiale : $T_i = 500$,
Température finale : $T_f = 0.1$,
Raison géométrique : $r_g = 0.95$,
Longueur maximale du palier : $L_p \text{ max} = 15$.

e. Premier traitement

Après ce choix, le système entame le processus itératif du recuit simulé pour la recherche d'une solution optimale ou proche de l'optimum. Après avoir effectué 1289 itérations en 96 unités CPU, le système donne une configuration finale dont le coût est de 266. D'où une amélioration de 24%. La configuration finale est donnée dans l'annexe E.3.a.

f. Relance du processus

Nous relançons le processus du recuit simulé à plusieurs reprises et nous constatons que le résultat ne s'améliore pas.

5.2. Série d'exemples

Nous donnons dans la suite un tableau récapitulatif des résultats d'une série d'applications dont l'exemple traité ci-dessus. Les matrices des flux et les configurations initiales sont données dans l'annexe E.3.

Nombre de cellules	Ent./Sor. atelier		Coût de la Config. initiale	Paramètres de contrôle du RS				Résultats			
	Nbre d'Entrées	Nbre de Sorties		Temp. initiale	Temp. finale	Raison géométrique	Longueur du palier	Coût final	% d'amélioration	Nombre d'itérations	durée de traitement
5	2	2	350	500	0,1	0,95	15	266	24.00	1289	242
10	0	0	781	500	0,1	0,95	15	658	15.74	1244	255
15	0	0	1089	600	0,1	0,95	15	654	39.95	1263	322
20	2	2	2520	700	0,1	0,95	15	1854	26.43	1317	1386
25	2	2	109006	800	0,1	0,95	15	70514	35.31	1797	2923

Tableau C8 : Résultats de l'agencement

Les configurations des résultats finaux de ces applications sont données dans l'annexe E.3.

6. CONCLUSION DE LA PARTIE C

Nous avons consacré cette partie C à la présentation du logiciel d'agencement des cellules dans un atelier de fabrication. Ceci n'exclut pas la possibilité de l'utiliser pour l'agencement de machines dans un job shop ou d'autres types de ressources reliées par des flux dans un environnement donné. Nous avons développé autour de la procédure d'optimisation proprement dite une interface simple et souple. Elle permet de fournir les données à l'entrée de plusieurs manières différentes, et à la sortie les résultats du traitement et de permettre à l'utilisateur d'intervenir pour relancer le processus ou pour faire des modifications locales dans la configuration finale. Pour illustrer d'avantage le fonctionnement du logiciel, nous avons présenté un exemple illustratif simple. D'autres applications de laboratoire avec des nombres de cellules plus grands ont été faites. Nous avons également utiliser ce logiciel pour l'agencement de systèmes réels que nous n'avons pas exposés dans la thèse. Deux d'entre eux pour des entreprises françaises (le premier concerne le réagencement d'un atelier de montage d'articles ménagers de la compagnie CALOR et le second concerne le réagencement d'un atelier de fabrication de turbines et de

compresseurs d'automobiles dans le cadre du projet ESPRIT "COALA"). La troisième application a été faite pour une entreprise Américaine (agencement d'un atelier de fabrication mécanique de la compagnie WESTINGHOUSE dans le cadre d'une collaboration avec l'université du Maryland). Les résultats obtenus ont été déclarés très satisfaisants par les utilisateurs.

Nous avons comparé notre logiciel avec les logiciels les plus connus du marché, à savoir :

CRAFT (USA 1964) : Buffa et al., 1964.

CORELAP (USA 1967) : Lee et Moore, 1967.

SHAPE (USA 1986) : Hassan et al., 1986.

KBML (Canada 1988) : Kusiak, 1990.

PLOOT (IBM France 1990) : Beziat, 1990.

Les tableaux qui suivent comparent notre logiciel (désigné par MSLD "Manufacturing Systems Layout Design") à ces logiciels.

1. Caractéristiques générales

	CRAFT	CORELAP	SHAPE	KBML	PLOOT	MSLD
Résolution par construction		*	*			
Résolution par amélioration	*			*	*	*
Intelligence artificielle				*		
Capacité	< 40	<45	<50	?	?	< 60
Prend en compte le système de manutention	*			*	*	*
Calcul les chemins réels						*
Optimise un critère donné	*		*	*	*	*

2. Les données prises en compte

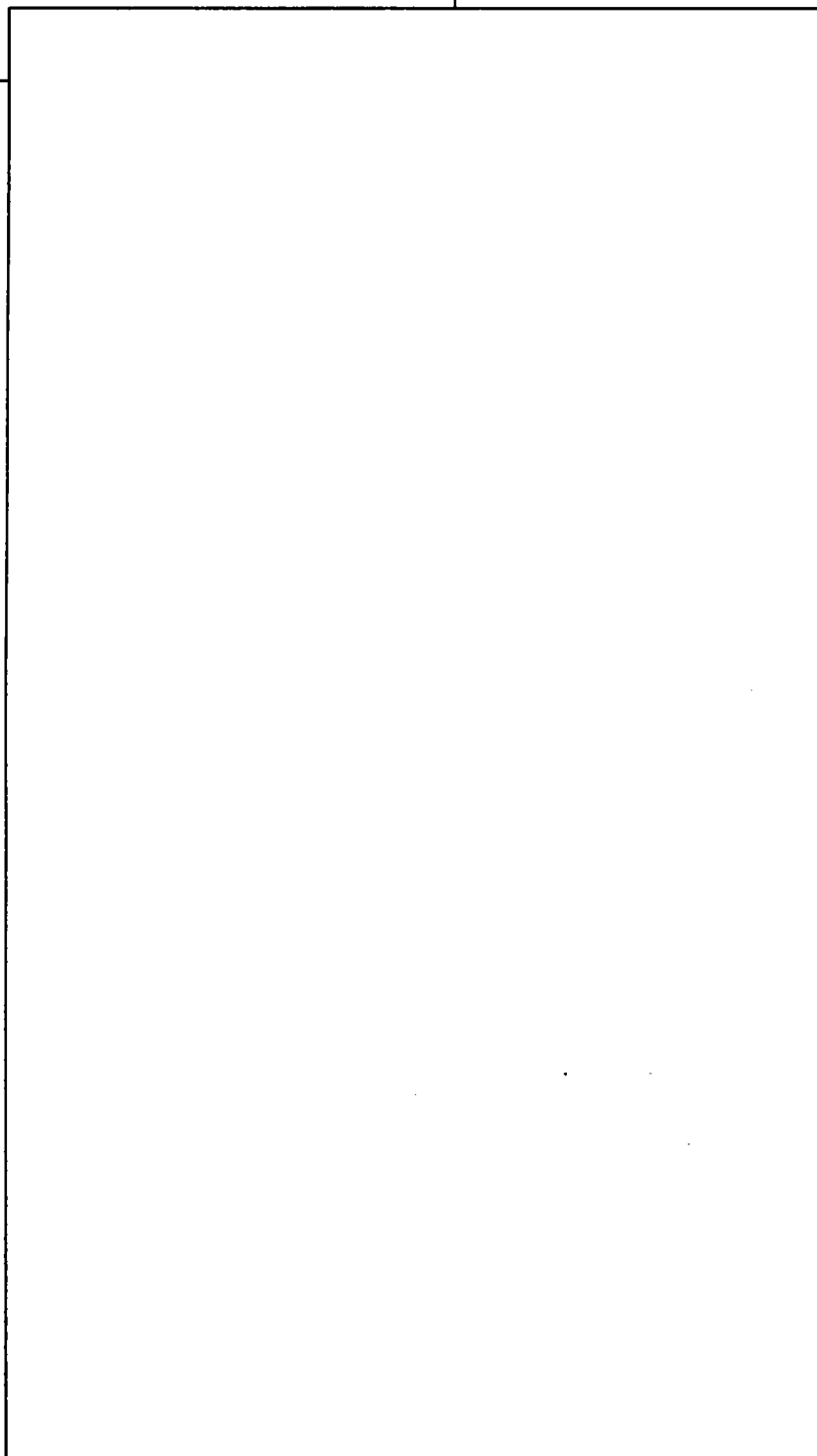
	CRAFT	CORELAP	SHAPE	KBML	PLOOT	MSLD
Flux	*		*	*	*	*
Tableau des relations qualitatives entre les ressources		*				
Surfaces des ressources	*		*		*	
Dimensions des ressources		*		*		*
Dimensions de l'atelier	*	*		*	*	*
Ressources fixes		*		*	*	*
Contraintes de proximité		*	*	*	*	*
Coût de réagencement des ressources	*					
Données sur le système de manutention	*			*		*
Configuration initiale	*			*	*	*
Paramètres relatifs à la méthode de résolution				*	*	*

3. Les résultats

	CRAFT	CORELAP	SHAPE	KBML	PLOOT	MSLD
Représentation de l'atelier	*	*	*	*	*	*
Le coût de fonctionnement	*		*	*	*	*
Les plus courts chemins entre les ressources						*
Evaluation de la solution		*				*

Nous pouvons dire que, relativement aux autres logiciels, MSLD est assez proche de la réalité des systèmes de fabrication et qu'il donne des résultats conformes aux besoins des utilisateurs.

CONCLUSION

A large, empty rectangular box with a black border, intended for writing the conclusion of a document. It occupies the majority of the page below the title.

CONCLUSION GÉNÉRALE

Face à l'intensification de la concurrence internationale, l'économie mondiale bascule de plus en plus d'une économie d'échelle vers une économie de diversité. La production se caractérise par une grande variété de produits, des systèmes de fabrication complexes et un marché fortement fluctuant. Les efforts déployés pour concevoir des systèmes bien adaptés à ces exigences ont abouti aux systèmes cellulaires de fabrication. Ces systèmes sont une application de la technologie de groupe, ils ont la cadence des systèmes linéaires et la flexibilité des job-shops.

L'un des principaux problèmes qui se posent lors de la conception des systèmes cellulaires de fabrication est l'agencement. Nous avons exposé dans la partie A de cette thèse une méthode globale de résolution de ce problème. Cette méthode comporte quatre modules : (i) le module de pré-traitement des données, où s'effectue l'équilibrage de la production et l'affectation des produits aux ressources, (ii) le module de composition des cellules orientés produits, (iii) le module de l'agencement interne des cellules et enfin (iv) le module d'agencement des cellules dans l'atelier de fabrication. La partie B de la thèse a été consacré à la résolution du problème qui se pose dans ce dernier module. Ce problème étant combinatoire, nous avons développé, pour le résoudre, une méthode basée sur l'algorithme du recuit simulé. Nous avons également étudié, dans cette partie B, le problème de l'agencement dynamique des systèmes de fabrication. Nous avons montré où et comment se pose t-il dans les systèmes cellulaires de fabrication et présenté des méthodes utilisant les solutions issues de la résolution du problème d'agencement statique.

La méthode, mise au point pour résoudre ce problème d'agencement inter-cellules, a débouché sur un logiciel (MSLD). Comparé aux logiciels les plus connus dans la littérature, MSLD semble assez proche de la réalité et donne des résultats relativement conformes aux exigences des systèmes manufacturiers actuels. En effet, des applications industrielles ont été réalisées

Conclusion

et ont donné des résultats appréciés des utilisateurs, aussi bien du point de vue temps de traitement que celui de la qualité des résultats.

Pour l'avenir, nous proposons les axes suivants (dans l'ordre d'importance décroissant) :

- Revoir plus en détail le problème de l'agencement dynamique des systèmes cellulaires de fabrication.

- Trouver d'autres algorithmes de calcul des plus courts chemins dans l'atelier ou améliorer les algorithmes existants afin de réduire d'avantage les temps d'exécution.

- Améliorer le modèle géométrique afin de pouvoir accepter des formes géométriques quelconques ne pouvant pas être ramenées à des formes rectangulaires.

ANNEXES

- A. Méthode du recuit simulé**
- B. Méthode d'exploration par séparation et évaluation**
- C. Délimitation des zones libres de l'atelier**
- D. Délimitation des entourages libres des cellules**
- E. Exemples numériques**

A. MÉTHODE DU RECUIT SIMULÉ

Le Recuit Simulé (RS) a été introduit pour la première fois par Kirkpatrick et coll. (1983). Il s'agit d'une technique d'optimisation qui exploite l'analogie entre les problèmes complexes d'optimisation et les phénomènes étudiés en thermodynamique statistique.

A.1. Thermodynamique statistique

La thermodynamique statistique est le noyau central de la physique de la matière condensée. Elle regroupe un ensemble de méthodes pour l'analyse du comportement de grands nombres de particules (atomes ou molécules) formant un échantillon de matière. Seul le comportement des particules dans un état d'équilibre à une température donnée peut être observé lors des expériences ; la thermodynamique statistique cherche à étudier le comportement d'un système de particules durant les changements d'état. Elle cherche en particulier à trouver les conditions sous lesquelles, lors de la diminution de la température, un matériau donné reste fluide ou se solidifie, et le cas échéant s'il tend vers un état cristallin stable d'énergie minimale ou vers un état amorphe (vitreux) métastable d'énergie supérieure.

A.2. Recuit physique

Un des procédés étudiés en thermodynamique statistique pour faire approcher un matériau de son état cristallin optimum est le **recuit**. Ce procédé est une stratégie de contrôle de la température du matériau en question. Il consiste à chauffer le matériau en le portant à sa température de liquéfaction (qui correspond à une énergie élevée), puis à le refroidir très lentement, en marquant des paliers de température de durée suffisante (pour l'établissement d'un état d'équilibre à cette température), jusqu'à ce que la température de solidification du matériau soit atteinte. Celui-ci aura à ce moment là une structure cristalline optimale où proche de l'optimum théorique. La technique opposée à celle du recuit est le **trempe**. Elle consiste à réduire très rapidement la température du matériau en fusion. On obtient alors une structure amorphe

(vitreuse) à un état métastable qui correspond à un minimum local d'énergie. La figure Aa.1 donne une illustration de ces deux phénomènes.

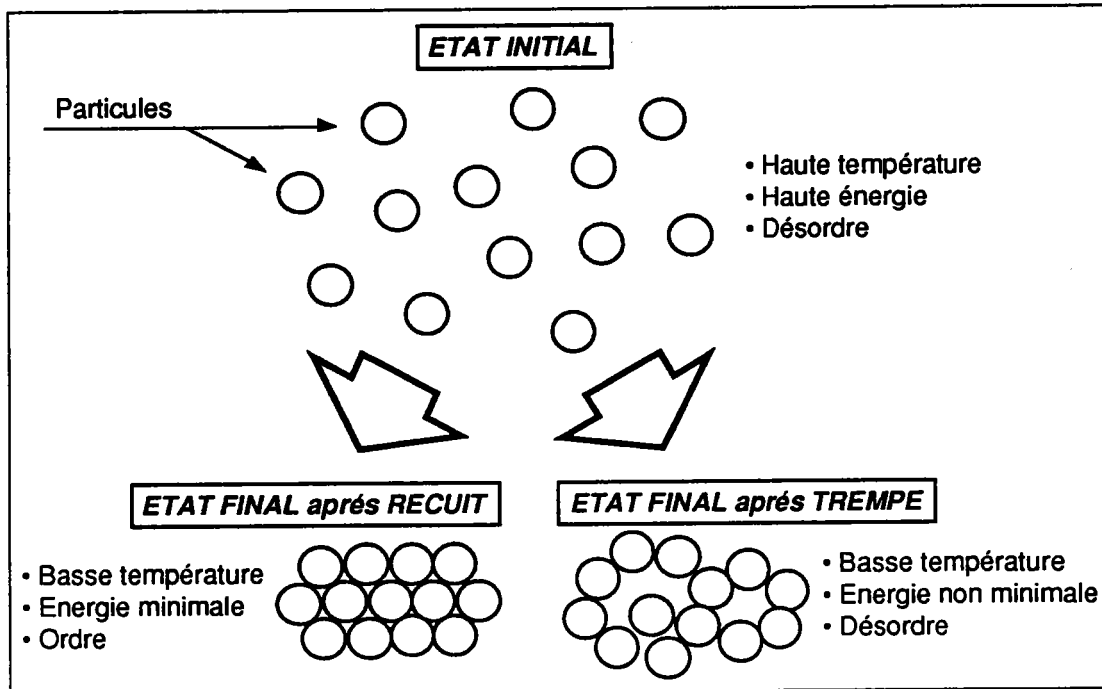


Figure Aa1: Le recuit et la trempe

A.3. Analogie entre le problème de cristallisation des matériaux et le problème d'optimisation combinatoire

L'analogie entre le problème physique de cristallisation des matériaux et le problème d'optimisation combinatoire réside principalement dans le fait que chacun de ces deux problèmes de minimisation peut aboutir à un minimum local d'où il ne pourra pas sortir si le processus d'optimisation (refroidissement) ne tient pas compte de la "vitesse" avec laquelle évolue le système. Cela suggère donc l'utilisation d'une procédure qui simule le comportement du recuit physique afin de résoudre les problèmes d'optimisation combinatoire. D'où la méthode du Recuit Simulé. Le tableau Aa.1 présente les différents points d'analogie entre les deux problèmes. Pour plus d'information sur les développements théoriques concernant cette analogie on pourra consulter le livre de Van Laarhoven et Aarts (1987), celui de Siarry et Dreyfus (1988), ainsi que celui de Azencott (1992).

Problèmes d'optimisation combinatoire	Problèmes de cristallisation des matériaux
solution réalisable du problème	état du matériau
variables du problème	positions des particules
coût de la solution	énergie libre

Figure Aa1 : Analogie entre problèmes physique et problèmes d'optimisation

A.4. Le Recuit Simulé

Du point de vue mathématique, le RS est un algorithme stochastique utilisé pour la minimisation d'une fonction numérique ayant un grand nombre de variables ou degrés de liberté. Notons que le même algorithme s'applique pour un problème de maximisation. C'est un algorithme qui procède par améliorations successives et peut être considéré comme une amélioration des méthodes d'optimisation locale.

Les méthodes d'optimisation locale évoluent d'une manière itérative en générant, à chaque itération, une solution voisine de la précédente. Si la valeur du critère correspondant à la nouvelle solution est meilleure, la nouvelle solution est sélectionnée, sinon elle est rejetée et c'est la solution précédente qui est retenue. Dans les deux cas, on continue le processus en choisissant une solution dans le voisinage de la solution sélectionnée. Le processus s'arrête lorsqu'il n'y a plus de possibilités d'amélioration ou lorsque le nombre total d'itérations choisi par l'utilisateur est atteint. L'inconvénient de l'optimisation locale est qu'elle aboutit souvent dans un minimum local et que la qualité de la solution finale dépend fortement de la solution initiale.

La différence du recuit simulé avec les algorithmes d'optimisation locale est qu'une solution B dérivant d'une solution A est acceptée non seulement lorsqu'elle est meilleure que A, mais également lorsqu'elle ne l'est pas avec une certaine probabilité. Cette probabilité est calculée comme suit :

$$p = \exp\left(-\frac{\Delta f}{T}\right) \quad (\text{Aa1})$$

où T est un paramètre donné (équivalent à la température dans le recuit physique) dont la valeur diminue avec le nombre d'itérations (i.e. le volume des calculs déjà effectués),

$$\text{et} \quad \Delta f = f(B) - f(A) > 0, \quad (\text{Aa2})$$

où $f(\cdot)$ est le critère à minimiser. Cette règle, dite *règle d'acceptation de Metropolis*, implique que : (i) plus petite est l'augmentation de la valeur du critère, plus facilement est acceptée la configuration qui lui correspond, et (ii) plus petite est la valeur de la température (i.e. plus le nombre d'itérations réalisées est important), plus faibles sont les chances d'acceptation de cette nouvelle solution.

L'algorithme du RS débute par une configuration initiale et une valeur initiale de la température T assez importante pour éviter de tomber prématurément dans un optimum local. Sur chaque palier de température (voir figure Aa2) l'algorithme effectue plusieurs itérations en produisant des solutions, qui sont acceptées ou refusées selon la règle d'acceptation de Metropolis. A chaque itération, le processus se poursuit à partir de la dernière solution acceptée. La température diminue, lorsqu'un état d'équilibre est atteint, c'est-à-dire qu'un certain nombre d'itérations a été réalisé. Ce nombre est appelé longueur du palier de température. Plusieurs critères d'arrêt du processus sont possibles. A titre d'exemple, un nombre total d'itérations, une valeur minimale de la température T , un minimum d'amélioration moyenne durant les "n" dernières itérations sont des critères utilisés.

La valeur initiale de la température, la longueur du palier, le processus de réduction de la température, et le critère d'arrêt du processus sont décrits dans le paragraphe qui suit.

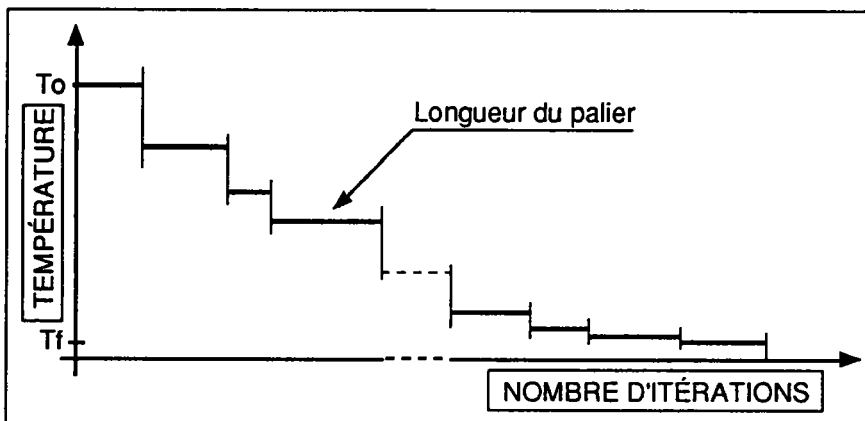


Figure Aa2 : Variation de la température

A.5. L'algorithme général du RS

L'algorithme du RS comporte deux boucles imbriquées : la boucle extérieure contrôle le processus de diminution de la température, et la boucle intérieure contrôle la longueur des paliers de température.

```

Générer une solution initiale i ;
Choix de la valeur initiale de la température  $T_0 > 0$  ;
Choix du processus de réduction de la température ;
Choix de la fonction donnant la longueur du palier ;
Faire :
    Mettre la longueur du palier à zéro,  $n = 0$  ;
    Faire :
        Générer une solution j dans le voisinage de i ;
        Calculer  $\Delta f = f(j) - f(i)$  ;
        Si  $\Delta f < 0$  alors  $i := j$  ;
        Sinon si  $\text{random}(0, 1)^* < \exp(-\Delta f / T)$  alors  $i := j$  ;
    Tant que (l'équilibre sur ce palier n'est pas atteint)
    Calculer la nouvelle température T ;
Tant que (le critère d'arrêt n'est pas vérifié).
```

* $\text{random}(0, 1)$ génère un nombre aléatoire compris entre 0 et 1.

Le processus de génération d'une solution initiale, le processus de passage d'une solution à une autre, ainsi que celui du calcul de la valeur du critère d'une solution donnée dépendent de l'application. Pour chacune des applications décrites ci-dessus, nous examinons chacun de ces points.

A.6. Les paramètres de contrôle du RS

Avec les premières études théoriques et expérimentales sérieuses (E. Aarts et J. Van Laarhoven (1987), B. Hajek (1988), J. Rose, W. Klebsch et J. Wolf (1990)), il est devenu clair que la réussite d'une application du RS nécessite un ajustement minutieux des paramètres de contrôle. Quatre choix génériques sont à faire : la température initiale, la fonction donnant la longueur d'un palier de

température, la fonction de variation de la température, et le critère d'arrêt du processus de résolution.

A.6.1. Température initiale

Kirkpatrick et al. (1983) ont proposé une température initiale T_0 suffisamment grande pour que la configuration qui provoque la plus grande détérioration de la valeur du critère sur le premier palier soit acceptée avec une probabilité P_0 généralement égale à 0.8. En reprenant les expressions Aa1 et Aa2, si Δf_{\max} est cette détérioration maximale du critère, la température initiale est donnée par la formule suivante :

$$T_0 = -\frac{\Delta f_{\max}}{\log(P_0)} \quad (\text{Aa3})$$

Cette température initiale doit également dépendre, dans certains cas, de la nature du système et de sa complexité (c'est-à-dire de la difficulté de passage d'une configuration à une autre). Dans le cas de l'agencement inter-cellulaire, par exemple, la complexité du problème provient essentiellement de l'encombrement des objets se trouvant dans l'atelier, c'est-à-dire de la facilité de mouvement de ces objets pour passer d'une configuration à une autre. Nous proposons donc de majorer la température donnée par (Aa3) comme suit :

$$T_0^* = E \cdot T_0 \quad (\text{Aa4})$$

où E est le degré de liberté des cellules dans l'atelier :

$$E = \frac{S_l}{\sum_{i=1}^{N_c} S_{c_i}} \quad (\text{Aa5})$$

S_l est la surface libre de l'atelier et S_{c_i} est la surface de la cellule i .

A.6.2. Longueur du palier de température

Le RS peut être représenté théoriquement sous forme d'une chaîne de Markov non homogène (Harhalakis et coll. 1990). La série de transformations réalisées sur chaque palier de température doit être suffisante pour qu'un équilibre puisse s'établir. La longueur de la chaîne de Markov du $k^{\text{ième}}$ palier de température est notée N_k et est dite longueur du palier. Plusieurs lois de variation de N_k ont été utilisées dans la littérature. Nous donnons les plus simples :

- a. Constante $N_k = \text{Cte.} ;$
- b. Arithmétique $N_k = N_{k-1} + \text{Cte.} ;$
- c. Géométrique $N_k = N_{k-1} / a, (a \text{ est une constante } < 1) ;$
- d. Logarithmique $N_k = \text{Cte} / \text{Log}(T_k) ;$
- e. Exponentielle $N_k = (N_{k-1})^{(1/a)}, (a \text{ est une constante } < 1).$

Dans nos applications du RS, le contrôle de la longueur du palier se fait à l'aide de deux paramètres : le nombre maximum d'acceptations et le nombre maximum de refus par palier. Sur chaque palier, nous comptons le nombre d'acceptations et le nombre de refus, et si l'un de ces nombres dépasse la valeur maximale qui lui est accordée, on dira qu'un équilibre s'est établi et on change de palier en diminuant la température.

A.6.3. Fonction de variation de la température

Rappelons que le rôle de la température est de contrôler la probabilité d'acceptation des configurations qui détériorent la valeur du critère tout au long de la résolution du problème. Le but est de faire en sorte que le nombre d'acceptations des configurations de ce type diminue progressivement. Plusieurs lois de variation de la température ont été proposées dans la littérature, les plus simples sont :

- a. Arithmétique $T_k = T_{k-1} - \text{Cte.} ;$
- b. Géométrique $T_k = \text{rg} \times T_{k-1} [\text{rg est une constante inférieure à } 1] ;$
- c. Fonction inverse $T_k = \text{Cte}/(1 + k) ;$
- d. Logarithmique $T_k = \text{Cte}/(\text{Log}(1 + k)).$

Nous avons utilisé la deuxième lois ci-dessus. Le coefficient rg est choisi plus petit que 1 mais proche de 1. Van Laarhoven (1987) propose des valeurs comprises entre 0,85 et 0,95.

A.6.4. Critère d'arrêt du processus

Reste à choisir le critère d'arrêt du processus du RS. Plusieurs critères sont possibles. Les suivants sont les plus simples :

- a. un nombre total fixé d'itérations est atteint,
- b. une température finale fixée est atteinte,
- c. le nombre souhaité d'acceptations pour un nombre donné d'itérations est atteint.

Dans nos applications, nous utilisons le second critère.

A.7. Conclusion

Le recuit simulé a été utilisé dans divers domaines avec plus ou moins de succès. Les chercheurs tentent actuellement de résoudre le problème du choix des paramètres de contrôle et de la solution initiale afin d'augmenter les chances d'aboutir aux résultats les meilleurs le plus rapidement possible. Il faut noter aussi que cette méthode, bien qu'elle soit satisfaisante en général, reste relativement inefficace pour certains problèmes où des heuristiques spécifiques donnent des résultats bien meilleurs. C'est le cas des problèmes de taille modeste. Cependant, le RS garde l'avantage de ne pas dépendre du critère à optimiser, ce dernier pouvant être modifié même à l'intérieur d'un même problème. Il a également l'avantage, par rapport aux autres heuristiques, de pouvoir fournir plusieurs bons résultats, l'utilisateur aura alors la possibilité de choisir celui qui lui convient le mieux en fonction de critères plus qualitatifs.

Lors de l'utilisation du RS, nous avons constaté que la difficulté de résolution d'un problème donné dépend non seulement de sa taille, mais aussi de sa nature et de ses caractéristiques particulières. Il faut donc tenir compte de cela lors du choix des paramètres de contrôle. Une manière de les choisir est de procéder par amélioration, c'est-à-dire, de commencer par les valeurs communément utilisées, puis de les modifier jusqu'à obtenir des paramètres donnant les meilleurs résultats.

Il est établi dans la théorie que pour avoir de bon résultats, il faut laisser tourner le RS assez longtemps (idéalement, pendant un temps infini). Cependant, nous avons constaté que pour les problèmes ayant un degré de liberté important (c'est-à-dire, un nombre de solutions possibles important), il valait mieux relancer le processus à plusieurs reprises avec un nombre d'itérations pas trop grand à chaque fois, que de lancer un long processus une seule fois. En procédant ainsi, nous donnons périodiquement au système la possibilité de repartir sur des pistes qui commencent par des solutions qui détériorent la valeur du critère. Nous augmentons les chances d'atteindre des solutions non atteignables autrement.

B. EXPLORATION PAR SÉPARATION ET ÉVALUATION

Les techniques d'exploration par séparation et évaluation sont destinées à résoudre les problèmes NP-complets. Elles consistent à décrire l'ensemble des solutions représentées par les nœuds d'une arborescence, et à parcourir celle-ci en examinant le moins de nœuds possible pour arriver à la solution optimale. Nous donnons dans ce qui suit une brève description de la théorie afférente. Pour plus de détails consulter le livre de Gondran et Minoux (1990), chapitre 11. Consulter également le livre de Proth et Hillion (1990), chapitre 5.

On considère le problème suivant :

$$\text{Min}_{x \in S} F(x),$$

S étant l'ensemble des solutions réalisables.

A l'aide des procédures d'exploration par séparation et évaluation, nous pouvons obtenir pour ce problème : une solution optimale, toutes les solutions optimales, ou une bonne solution. Nous ne parlons dans la suite que de la première possibilité.

Le principe d'une procédure d'exploration par séparation et évaluation est de construire progressivement des solutions admissibles jusqu'à isoler une solution optimale. Chaque nœud de l'arborescence correspond à un ensemble de solutions, lequel ensemble peut ne pas contenir de solutions admissibles. La racine de l'arborescence représente toutes les solutions S du problème (voir figure Ab4). Pour définir plus précisément la procédure d'exploration, il faut définir une règle de séparation qui permet de construire l'arborescence à explorer et une règle d'évaluation qui permet de se déplacer dans cette arborescence.

Soit S_1, S_2, \dots, S_n des ensembles de solutions, pouvant être non-admissibles, tels que :

$$S_1 \cup S_2 \cup \dots \cup S_n \supseteq S$$

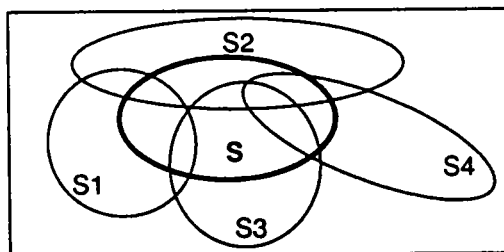


Figure Ab1 : Séparation de l'ensemble S, cas général

On distingue le cas particulier où les sous-ensembles ne contiennent que des solutions admissibles, où : $S_1 \cup S_2 \cup \dots \cup S_n = S$

et $\forall (i, j) \in \{1, 2, \dots, n\}^2 / i \neq j; \quad S_i \cap S_j = \emptyset.$

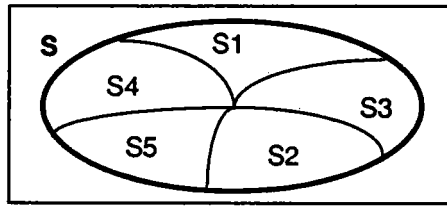


Figure Ab2 : Cas particulier de séparation de l'ensemble S

Nous supposons que :

1. Pour tout sous-ensemble S_i ($i = 1, 2, \dots, n$) on peut trouver une borne inférieure pour le problème :

$$\text{Min}_{x \in S \cap S_i} F(x)$$

Cette borne est notée " b_i ".

2. On peut également trouver dans S une borne supérieure pour la valeur optimale $F(x^*)$ de la fonction objectif. Cette borne est notée " a ".

A partir des hypothèses ci-dessus, on peut dire que x^* se trouve certainement dans l'ensemble S^1 donné par (voir figure Ab3) :

$$S^1 = \bigcup_{i \in \bar{E}} (S \cap S_i) \quad \text{avec } \bar{E} = \{i / b_i \leq a\}$$

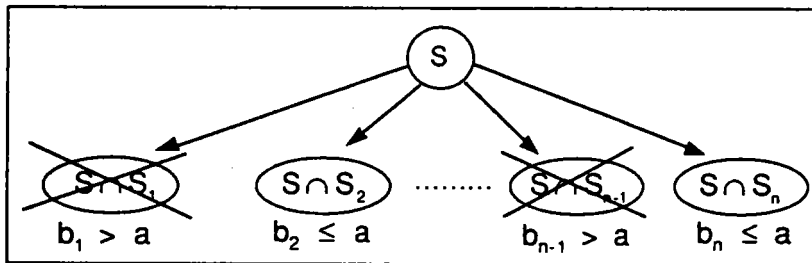


Figure Ab3 : Arbre de prospection par séparation et évaluation : première itération

Plus la borne supérieure " a " est proche de l'optimum du problème, plus S^1 est petit, et plus vite la solution optimale est localisée.

Le même processus est repris en séparant l'ensemble S^1 . On obtient les ensembles S'_1, S'_2, \dots, S'_n tels que :

$$S_1^1 \cup S_2^1 \cup \dots \cup S_{n_1}^1 \supseteq S^1.$$

Après évaluation des sous-ensembles $(S^1 \cap S_i^1)$ ($i = 1, 2, \dots, n_1$), on obtient l'ensemble S^2 , tel que :

$$S^2 = \bigcup_{i \in \bar{E}_1} (S^1 \cap S_i^1) \quad \text{avec } \bar{E}_1 = \{i / b_i^1 \leq a\};$$

où b_i^1 est la borne inférieure de : $\text{Min}_{x \in S^1 \cap S_i^1} F(x)$

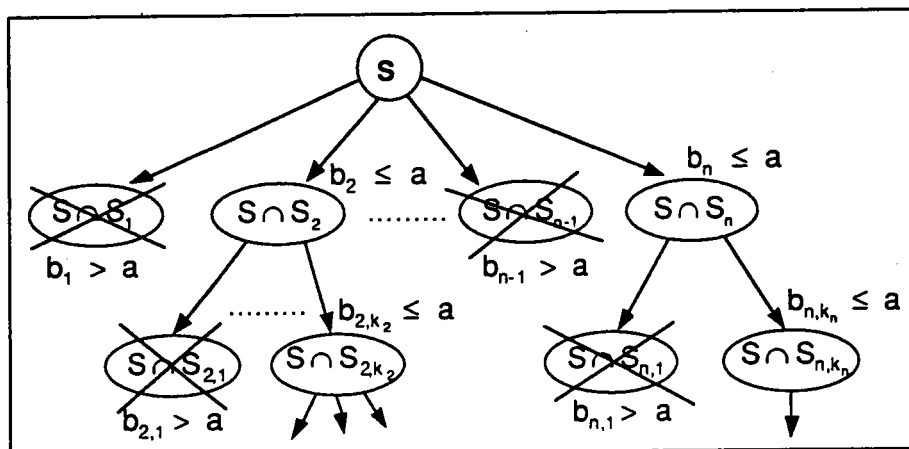


Figure Ab4 : Arbre d'exploration par séparation et évaluation

Les sous-ensembles $S_1^1, S_2^1, \dots, S_{n_1}^1$ sont déduits après la séparation des sous-ensembles S_1, S_2, \dots, S_n non éliminés après évaluation dans l'étape précédente.

Le processus continue de la même manière en calculant S^3, S^4, \dots jusqu'à ce que l'on atteigne l'ensemble S^N suffisamment petit pour permettre la localisation de la solution optimale x^* , telle que :

$$F(x^*) = \text{Min}_{x \in S^N} F(x)$$

Le résultat le plus simple est d'arriver à trouver en fin de processus un ensemble $S^N = \{x^*\}$.

Les déplacements dans l'arborescence se font généralement de deux manières différentes, (i) par exploration en largeur d'abord (elle sépare le sommet de l'arborescence ayant donné la meilleure évaluation) et (ii) par exploration en profondeur d'abord (elle sépare le sommet de l'arborescence le plus proche du dernier sommet séparé).

CONCLUSION

Nous pouvons dire que les principales conditions qui rendent possible l'utilisation d'une procédure d'exploration par séparation et évaluation pour résoudre un problème d'optimisation NP-complet sont :

- de trouver une méthode qui donne une borne supérieure de bonne qualité (proche de l'optimum) dans l'ensemble des solutions admissibles ;
- d'avoir une procédure de séparation de l'ensemble des solutions admissibles en sous-ensembles ;
- d'avoir une méthode de calcul de la borne inférieure de la solution du problème pour tous les sous-ensembles issus de la séparation.

C. DÉLIMITATION DES ZONES LIBRES RECTANGULAIRES DE L'ATELIER

Nous présentons dans cette annexe un utilitaire largement utilisé dans la procédure de construction d'une configuration initiale, et dans les procédures de transformations élémentaires. Cet utilitaire permet de trouver les zones libres rectangulaires de l'atelier sous une configuration donnée. Un exemple de ces zones est donné dans la figure Ac1. Nous constatons sur cette figure (Ac1) que ces zones libres rectangulaires se superposent. Les superpositions sont possibles du fait que nous ne manipulons qu'une seule cellule à la fois lors d'une installation ou d'un déplacement.

La détermination de ces zones se fait en deux phases. Dans la première phase, on détermine un point de chacun des rectangles libres (le sommet du haut à gauche, voir figure Ac1). Puis dans la deuxième phase, on procède par balayage en partant de chacun de ces sommets, pour déduire les autres limites du rectangle. Les détails de ces deux phases sont les suivants :

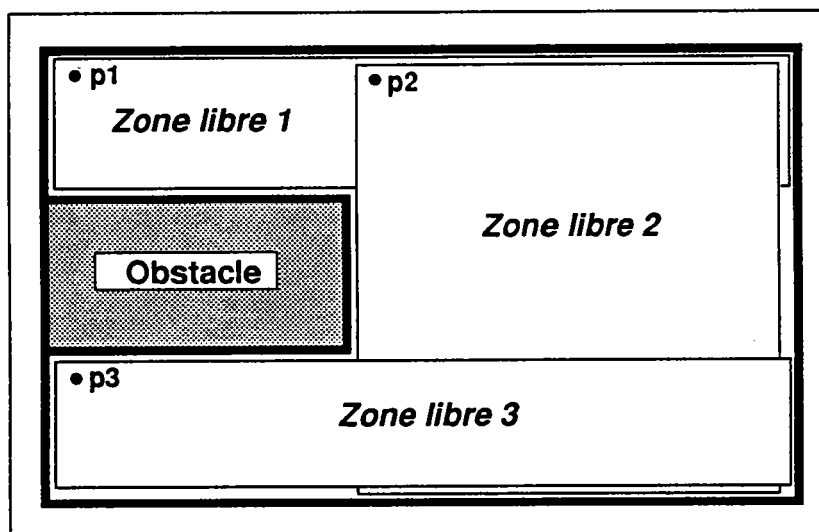


Figure Ac1 : Exemple de zones libres rectangulaires à déterminer

Remarque : Dans ce qui suit, tous les objets se trouvant à l'intérieur de l'atelier (zones interdites et cellules) sont appelés obstacles sans aucune distinction.

Phase 1: Détermination des sommets des rectangles

Les sommets à déterminer vérifient la propriété suivante (voir figure Ac2) :

Chacun de ces sommets (numérotés 1, 2, 3, ...) est l'intersection de deux segments l'un horizontal [les segments (1, 3), (4, a), (5, b) et (7, d) de la figure Ac2] et l'autre vertical [les segments (a, 1), (c, 2), (d, 3) et (7, 4) de la figure Ac2] :

(a). Les segments horizontaux sont : (i) les segments formés par les mailles, libres et adjacentes, de la première ligne de la grille ou la matrice [MATEL], ou (ii) les segments joignant deux points, A et B, définis comme suit :

- **A** (l'extrémité droite du segment) se trouve :
 - sur la maille libre de l'angle du bas à droite d'un obstacle (les points a et c de la figure Ac2).
 - ou sur la maille de l'angle entre le côté bas d'un obstacle et le côté droit de l'atelier.
- **B** (l'extrémité gauche du segment) se trouve :
 - sur une maille située sur le prolongement vers la gauche du point A et adjacente au côté droit d'un obstacle
 - ou au bord gauche de l'atelier (les points 4, 5 et 7 sur la figure Ac2).

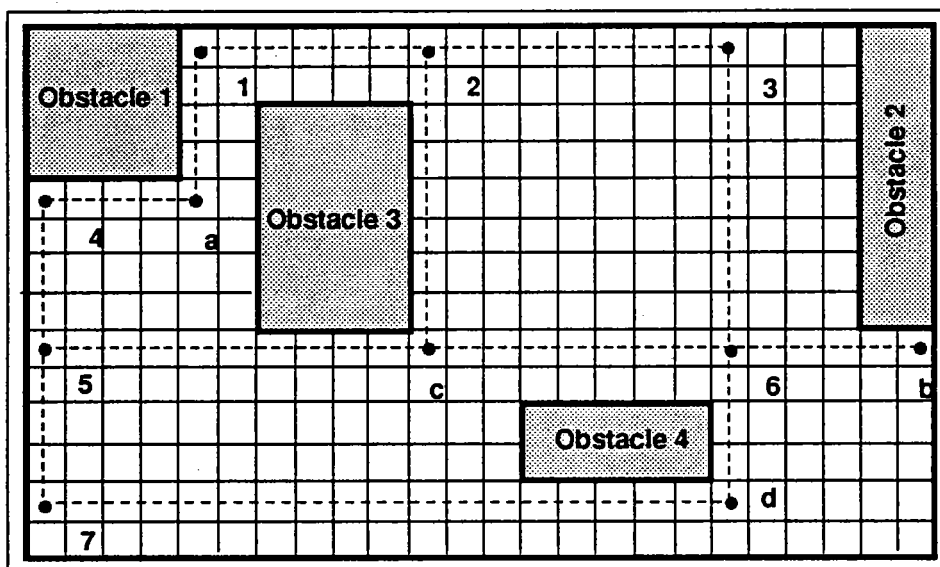


Figure Ac2 : Détermination des sommets des rectangles libres

(b). Les segments verticaux sont : (i) les segments formés par les mailles, libres et adjacentes, de la première colonne de la grille ou de la matrice [MATEL], ou (ii) les segments joignant les deux points, C et D, définis comme suit :

- **C** (l'extrémité inférieure du segment), se trouve sur la maille libre de l'angle du bas à droite d'un obstacle (les points a et c de la figure B10). Ce point peut être également le point A du segment horizontal. Il se trouve aussi sur la maille de l'angle entre le côté gauche d'un obstacle et le côté bas de l'atelier.
- **D**, l'extrémité supérieure du segment, se trouve sur une maille située à l'extrémité du prolongement du point C vers le haut et adjacente au côté inférieur d'un obstacle ou au côté supérieur de l'atelier (les points 1, 2 et 3 sur la figure B10).

Pour trouver ces sommets, on considère les obstacles les uns après les autres. On distingue alors trois cas de figure. Nous donnons dans ce qui suit le point de départ relié à l'obstacle et qui nous permet de trouver un ou deux des sommets recherchés :

- Cas 1 : Le côté droit de l'obstacle est contre le côté droit de l'atelier. Le point de départ est alors le point de l'angle entre le côté bas de l'obstacle et le côté droit de l'atelier.
- Cas 2 : Le côté bas de l'obstacle est contre le côté bas de l'atelier. Le point de départ est alors le point de l'angle entre le côté bas de l'atelier et le côté droit de l'obstacle.
- Cas 3 : L'obstacle n'est contre aucun de ces deux côtés de l'atelier. Le point de départ est alors le point se trouvant sur la maille libre du bas à droite de l'obstacle.

Les coordonnées de ces points de départ sont notées (xd, yd).

L'algorithme de détermination des sommets est le suivant (à suivre sur la figure Ac2) :

Pour chaque obstacle, faire :

- | | |
|-------------------|--------------------------------------|
| si cas 1 : | PROSPECTION HORIZONTALE, |
| sinon, si cas 2 : | PROSPECTION VERTICALE, |
| sinon, si cas 3 : | PROSPECTION HORIZONTALE, |
| | et PROSPECTION VERTICALE, |
| sinon : | l'obstacle n'est pas pris en compte. |

Fin faire.

Rappelons que Dx et Dy sont respectivement les dimensions de l'atelier suivant l'axe des abscisses et celui des ordonnées.

L'algorithme de la procédure de prospection horizontale est le suivant :

1. Considérer un point de départ (x_d, y_d)

2. Pour $(i = x_d \text{ à } 0)$ faire :

Si (le point $(i-1, y_d)$ appartient à un obstacle) :

Si (le point (i, y_d) a été déjà pris comme sommet) :

Ignorer ce point.

Sinon : Le conserver comme sommet.

Passer à un autre point de départ.

Sinon : Pour $(j = y_d + 1 \text{ à } D_y)$ faire :

(D_y : dimension de l'atelier suivant l'axe y de l'atelier).

Si (le point $(i-1, j)$ appartient à un obstacle) :

Si (le point (i, j) a déjà été pris comme sommet) :

Ignorer ce point et continuer.

Sinon : le conserver comme sommet et continuer.

Si (le point $(i, j+1)$ appartient à un obstacle) :

Arrêter la descente et passer à un autre point de départ.

Fin faire.

Fin faire.

L'algorithme de la procédure de prospection verticale est le suivant :

1. Considérer un point de départ (x_d, y_d) .

2. Pour $j = y_d \text{ à } 0$, faire :

Si (le point $(x_d, 1)$ est atteint sans avoir rencontré d'obstacle) :

Si $((x_d, 1)$ n'a jamais été considéré comme sommet) :

Conserver le point comme sommet.

Sinon, Ignorer le point.

Passer à un autre point de départ.

Sinon, si (le point $(x_d, j-1)$ appartient à un obstacle) :

Passer à un autre point de départ.

Fin faire.

Le nombre de ces sommets est noté N_s (dans l'exemple de la figure Ac3, on compte 7 sommets, numérotés de 1 à 7). Ces sommets sont considérés un à un dans la phase suivante pour la détermination des rectangles libres.

Phase 2 : Détermination des rectangles libres

Chacun de ces N_s sommets donne au moins un rectangle libre (voir l'exemple de la figure B11, le point 5 donne les deux rectangles libres 6 et 7).

Parmi ces points, ceux qui se trouvent sur la dernière ligne constituent un cas particulier pour lequel nous utilisons la procédure B. Nous décrirons chacune de ces procédures après avoir donné la procédure principale.

Remarque : dans ce qui suit, les coordonnées des sommets sont notées (x_s, y_s) , avec : $0 \leq x_s < Dx$ et $0 \leq y_s < Dy$.

Pour $s = 1$ à N_s , faire :

 Si $(y_s < Dy - 1)$: PROCEDURE A

 Sinon, si $(y_s = Dy - 1)$: PROCEDURE B

Fin faire.

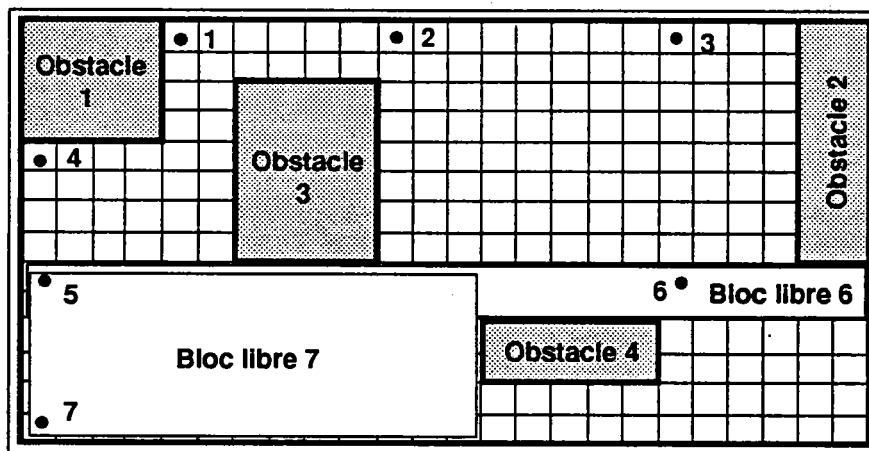


Figure Ac3 : Détermination des blocs rectangulaires libres
(cas des points n'appartenant pas à la dernière ligne)

Algorithme de la procédure A (voir figure Ac3) :

1. Considérer le sommet de coordonnées (x_s, y_s) tel que $0 < y_s < Dy - 1$.
2. Fixer la limite initiale suivant l'axe des abscisses à $(limix = Dx)$.
3. Pour $(j = y_s \text{ à } Dy)$ faire :
 - Pour $(i = x_s \text{ à } limix)$ faire :
 - Si, $(i+1, j)$ appartient à un obstacle, alors
 - Conserver cette limite comme l'abscisse du côté droit d'un nouveau rectangle libre, sous réserve qu'il soit accepté* ensuite.
 - Modifier la limite sur l'axe x, si $(i < limix)$: $limix = i + 1$.
 - Si, le point $(i, j + 1)$ appartient à un obstacle, alors
 - Si la limite latérale droite de ce bloc a été atteinte ci-dessus, et qu'il est acceptable* : Conserver le bloc.
 - Sinon : Ignorer ce bloc.

Fin faire.

Fin faire.

Algorithme de la procédure B (voir figure Ac4) :

1. Considérer le sommet de coordonnées (x_s, y_s) tel que $y_s = Dy - 1$.
 2. Pour $(i = x_s \text{ à } Dx)$ faire :
 - Si (le point (i, y_s) appartient à un obstacle)
 - Conserver le bloc comme rectangulaire libre.
- Fin faire.

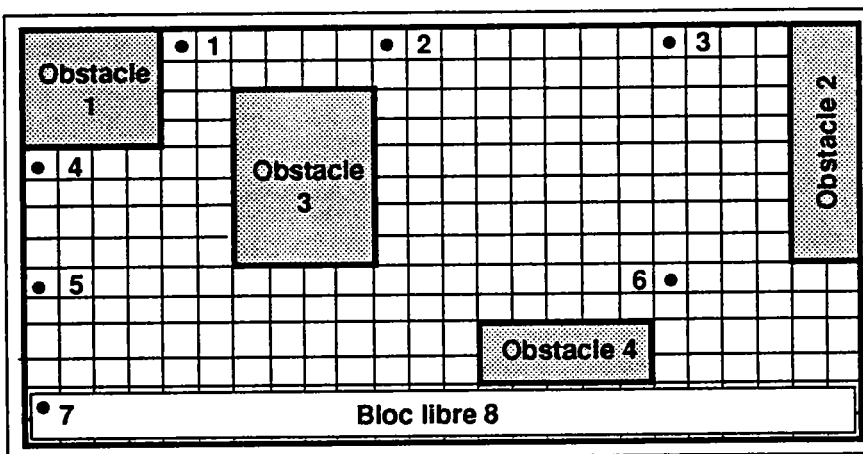


Figure Ac4 : Détermination des blocs rectangulaires libres
(cas des points appartenant à la dernière ligne)

*** Règle d'acceptation d'un bloc**

A chaque fois qu'un bloc est créé, le système vérifie si chacun de ses côtés est adjacent à au moins un obstacle ou côté de l'atelier. Dans l'exemple de la figure B11, le bloc libre 7 est adjacent aux obstacles 3 et 4 et aux côtés de gauche et du bas de l'atelier.

D. DÉLIMITATION DES ENTOURAGES LIBRES DES CELLULES

Dans trois des quatre transformations élémentaires (la permutation, la translation et la rotation), nous avons eu besoin de connaître les entourages rectangulaires libres des cellules. Ces entourages sont déterminés de la manière suivante :

Comme il est montré dans la figure Ad1, le rectangle libre contenant la cellule C noté RL(C) est obtenu à partir de quatre rectangles libres situés autour de la cellule. Chacun de ces rectangles, numérotés de 1 à 4, donne une, deux ou aucune des limites du rectangle libre recherché.

- Le rectangle R_1 peut donner une limite supérieure Lh_1 et/ou une limite droite Ld_1 de RL(C). Il est déterminé par balayage vers le haut et vers la droite à partir du point d_1 .

- Le rectangle R_2 peut donner une limite droite Ld_2 et/ou une limite inférieure Lb_2 de RL(C). Il est déterminé par balayage vers le bas et vers la droite à partir du point d_2 .

- Le rectangle R_3 peut donner une limite inférieure Lb_2 et/ou une limite gauche Lg_1 de RL(C). Il est déterminé par balayage vers le bas et vers la gauche à partir du point d_3 .

- Le rectangle R_4 peut donner une limite gauche Lg_2 et/ou une limite supérieure Lh_2 de RL(C). Il est déterminé par balayage vers le haut et vers la gauche à partir du point d_4 .

Ainsi, nous déduisons les limites du rectangle RL(C) comme suit :

- limite supérieure : $Lh = \min (Lh_1, Lh_2),$
- limite droite : $Ld = \max (Ld_1, Ld_2),$
- limite inférieure : $Lb = \max (Lh_1, Lb_2),$
- limite gauche : $Lg = \min (Lg_1, Lg_2).$

Par conséquent, les coordonnées des sommets du rectangle libre contenant la cellule C sont :

- sommet I : (Lg, Lh), sommet II : (Ld, Lh),
- sommet III : (Lg, Lb), sommet IV : (Ld, Lb).

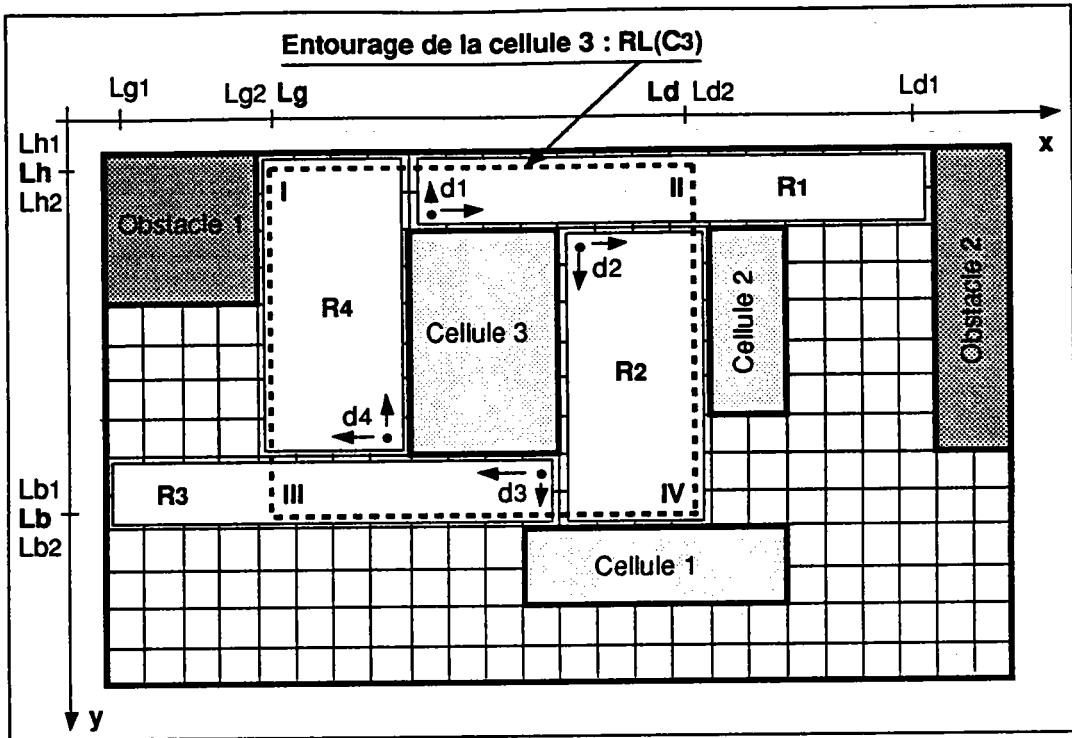


Figure Ad1 : Détermination de l'entourage libre d'une cellule

E. EXEMPLES NUMÉRIQUES

Cette annexe contient un exemple d'illustration pour chacun des modules de formation des cellules, d'agencement interne des cellules et d'agencement des cellules dans l'atelier de fabrication. Pour le module d'agencement intra-cellulaire, nous ne donnons d'exemple que pour la partie concernant le problème d'affectation des machines aux sites.

E.1. Formation des cellules de fabrication

L'exemple proposé pour illustrer la solution que peut donner le module de conception des cellules concerne un système de fabrication de 23 machines produisant 24 types de produits. Nous considérons que le ratio de production est le même pour tous les types.

Le tableau Ae1 donne les routages des produits à travers les machines. Chaque produit visite au plus trois machines. Les machines sont données dans l'ordre dans lequel elles sont visitées.

	PRODUITS																							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1st mach.	18	20	12	12	19	22	20	7	7	3	14	21	4	9	21	10	14	5	22	18	13	1	6	8
2nd mach.	17	4	16	13	16	2	23	1	18	21	21	14	15	3	9	5	21	11	11	1	19	18		
3rd mach.			19						7			21						10			13			

Tableau Ae1 : Les routages des produits

L'algorithme de conception des cellules donne une partition de 7 cellules.

Le tableau Ae2 donne la composition de ces cellules.

	CELLULES						
	1	2	3	4	5	6	7
The cells composition	11	1	15	8	14	6	12
	2	17	4		9		13
	10	18	23		21		16
	5	7	20		3		19
	22						

Tableau Ae2 : La solution finale

E.2. Agencement intra-cellulaire

Nous avons appliqué l'algorithme d'agencement intra-cellulaire à une cellule composée de 8 machines, produisant 17 types de produits. Le tableau condensé Ae3 donne les dimensions des machines, les tailles des lots, les quantités à produire durant un horizon de temps T et les routages des produits.

Par exemple, 10 produits de type 2 doivent être produits durant l'horizon T, avec une taille du lot de 1 (les produits de type 2 évoluent 1 à 1 dans la cellule) et doivent visiter dans l'ordre les machines 4, 8, 7 puis 8 une seconde fois.

MACHINES			PRODUITS																	
			Produit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Quantité			8	10	20	40	10	4	30	100	10	20	12	10	2	30	5	50	5	
Ind	Lon	Lar	Taille du lot	1	1	4	20	5	2	30	10	5	5	4	10	1	6	5	10	5
1	20	20	Routage	1			2,4			1,4		1		4	1		2			2
2	10	10					1	1,5	4			2,4		2		1				
3	15	15		2											1,3			1		
4	10	10			1	2		3	2	2	2		1							1
5	15	15				1				3		3	2,4		2			3	2	
6	15	15						2	3				3		3,5			2	1,3	
7	10	10			3			4	1						4	2				
8	10	10			2,4		3				1				6			1		

Tableau Ae3 : Caractéristiques des machines et des produits

Le SdM utilisé est un pont roulant, la cellule est du type multi-lignes, le nombre maximum de lignes est 3, le nombre maximum de sites par ligne est 3 aussi et la distance entre deux lignes adjacentes est 25.

La valeur du critère correspondant à cette configuration initiale (voir figure Ae1) est de 3812,40. Après 1145 itérations faites durant 20,36 sec. CPU (sur un SUN sparc station 330), le système donne une configuration finale (voir figure Ae1) pour laquelle la valeur du critère est 2210,12 (une amélioration de 42,03%).

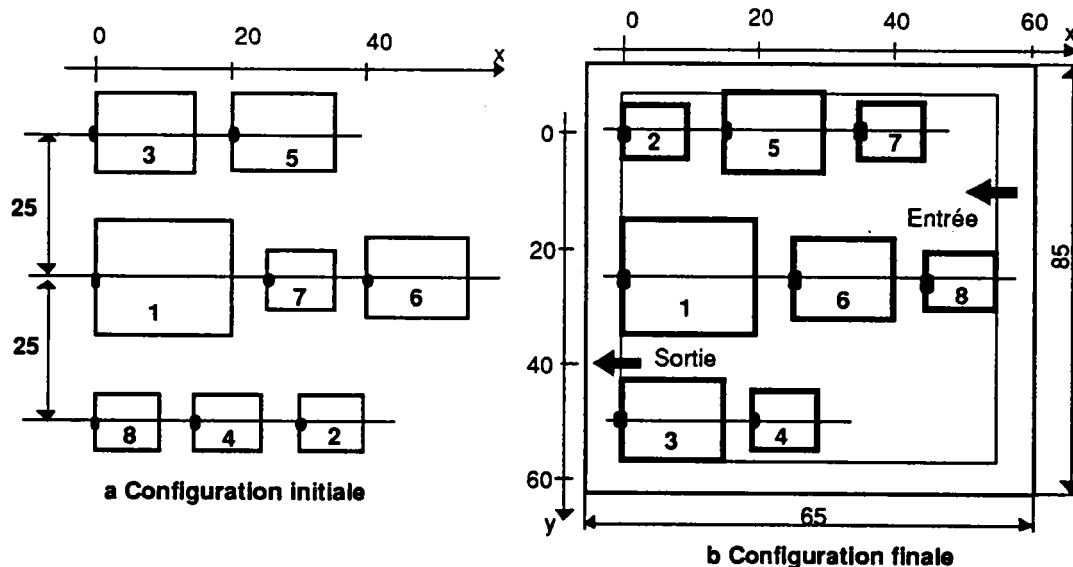


Figure Ae1 : Agencement intra-cellulaire : configurations initiales et finales

E.3. Agencement inter-cellulaire

a. Exemple à 5 cellules

	1	2	3	4	5	Sa1	Sa2
1	0	1	1	1	0	1	0
2	1	0	2	4	0	0	0
3	0	0	0	0	0	1	1
4	2	1	0	0	0	1	2
5	0	1	0	1	0	0	0
Ea1	0	0	0	0	3	0	0
Ea2	2	2	0	0	0	0	0

Sa : sortie de l'atelier ; Ea : entrée de l'atelier

Tableau Ae4 : Matrice des flux (atelier à 5 cellules)

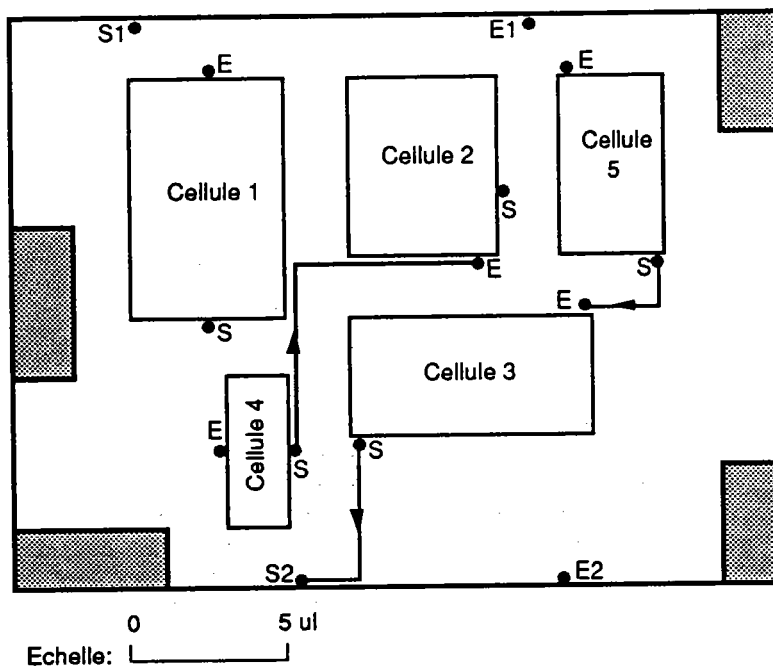


Figure Ae2 : Configuration initiale de l'atelier à 5 cellules

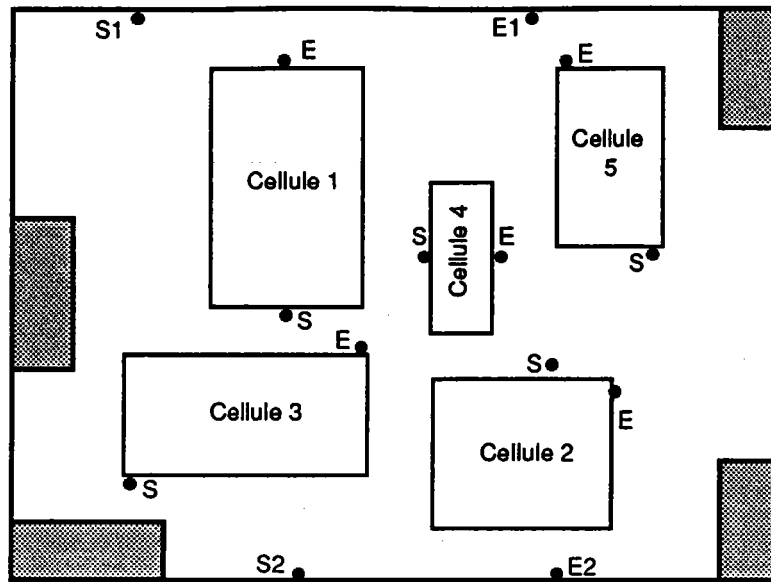


Figure Ae3 : Résultat final (5 cellules)

b. Exemple à 10 cellules

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	1	1	0	1	2	0	0
2	0	0	1	0	2	1	0	0	1	0
3	1	0	0	0	2	0	2	0	0	1
4	0	0	2	0	0	1	0	0	2	0
5	1	0	1	1	0	1	1	0	0	0
6	0	1	0	1	1	0	0	2	0	0
7	0	0	2	0	0	1	0	10	0	0
8	0	0	1	0	1	1	0	0	0	0
9	1	0	1	1	0	1	0	1	0	0
10	0	0	0	1	0	0	0	1	3	0

Tableau Ae5 : Matrice des flux (atelier à 10 cellules)

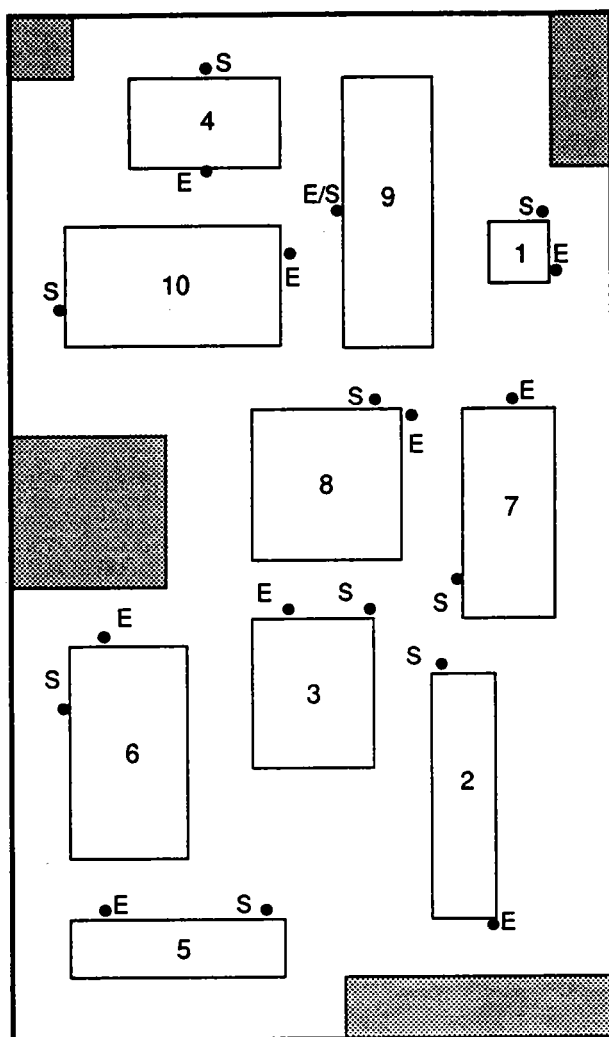


Figure Ae4 : Atelier à 10 cellules (configuration initiale)

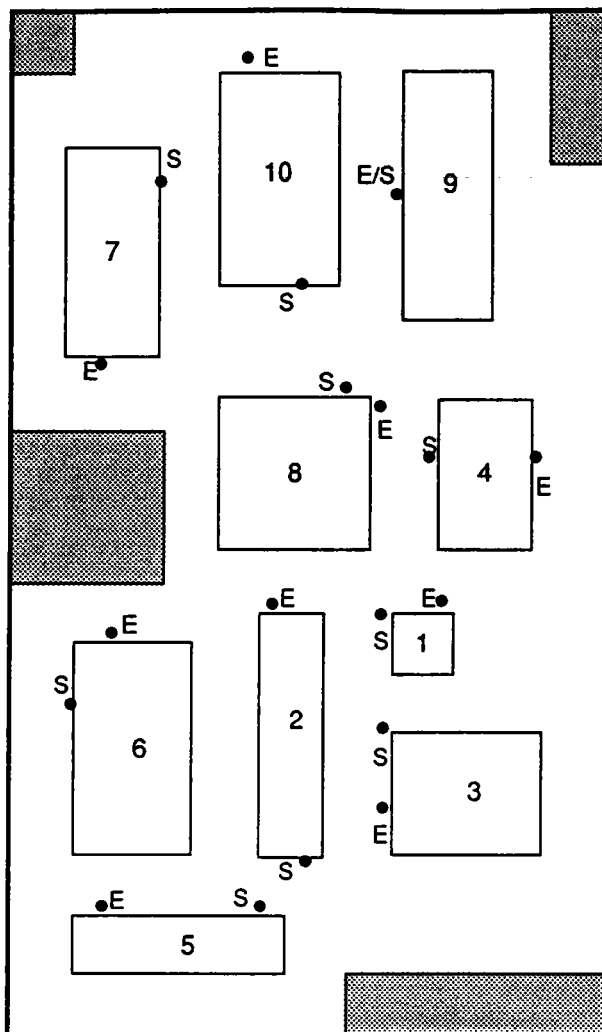


Figure Ae5 : Résultat final (10 cellules)

c. Exemple à 15 cellules

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
2	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	1	1	0	0	0	1	1	0	0	0	0	0	0
6	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
7	0	1	0	0	0	0	0	1	0	1	1	0	1	0	1
8	1	0	0	0	2	0	1	0	0	0	0	1	0	0	0
9	0	0	0	0	1	2	0	1	0	1	0	0	0	1	0
10	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0
12	0	0	0	0	0	0	1	0	0	0	0	0	0	2	0
13	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0
14	0	0	0	0	0	0	0	1	0	0	1	1	0	0	1
15	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0

Tableau Ae6 : Matrice des flux (atelier à 15 cellules)

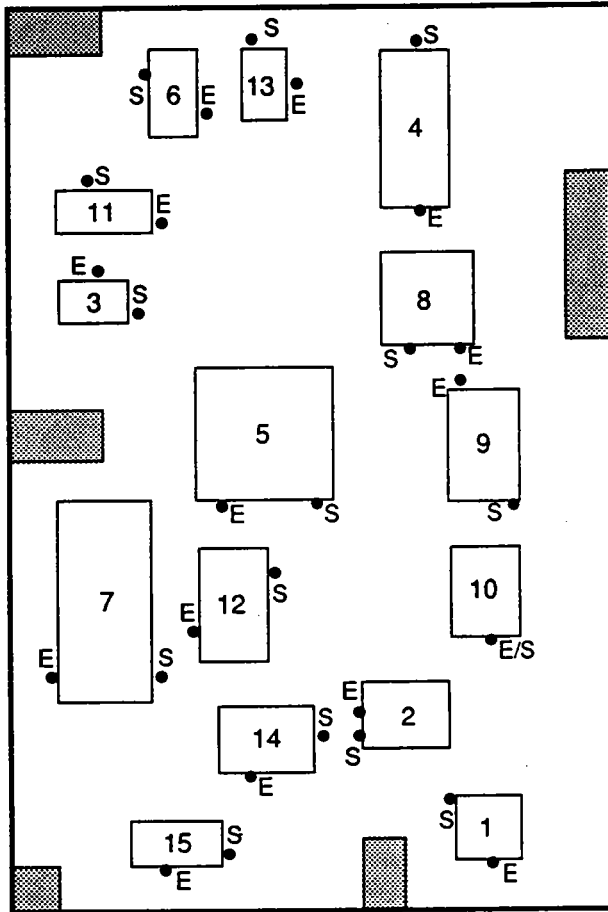


Figure Ae6 : Atelier à 15 cellules (configuration initiale)

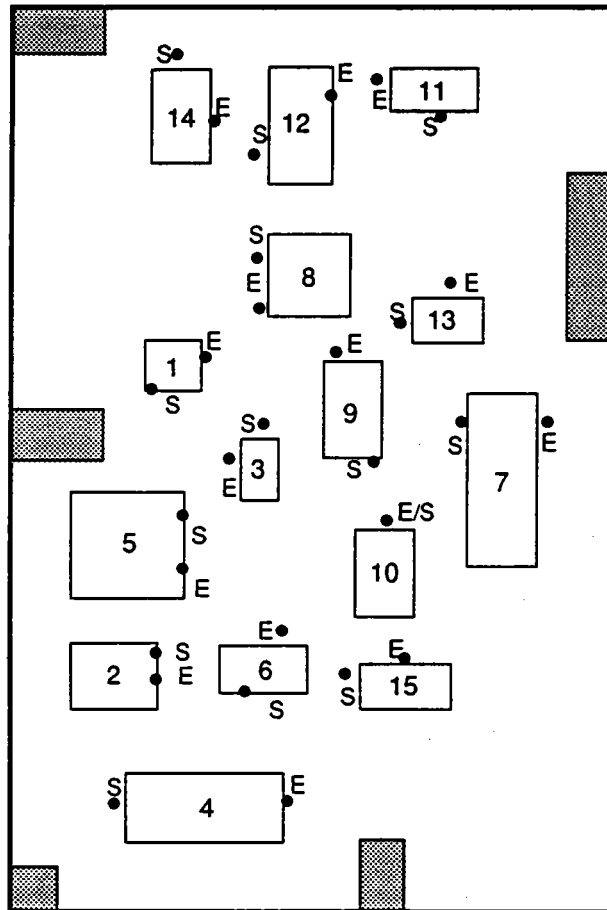


Figure Ae7 : Résultat final (15 cellules)

d. Exemple à 20 cellules

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	S1	S2
1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1	1
2	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
4	0	0	0	0	0	1	0	0	0	1	0	0	1	1	1	0	0	0	0	0	1	0
5	2	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1
6	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2	0
7	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0
8	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
9	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
11	0	0	0	1	0	0	0	0	1	0	0	1	0	1	0	0	1	0	1	0	1	0
12	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	1	0	0
13	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
14	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
15	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
16	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
19	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
20	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
Ea1	1	0	0	0	0	0	2	0	0	0	0	0	2	0	0	1	0	0	0	0	0	0
Ea2	2	0	0	1	1	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0

Tableau Ae7 : Matrice des flux (atelier à 20 cellules)

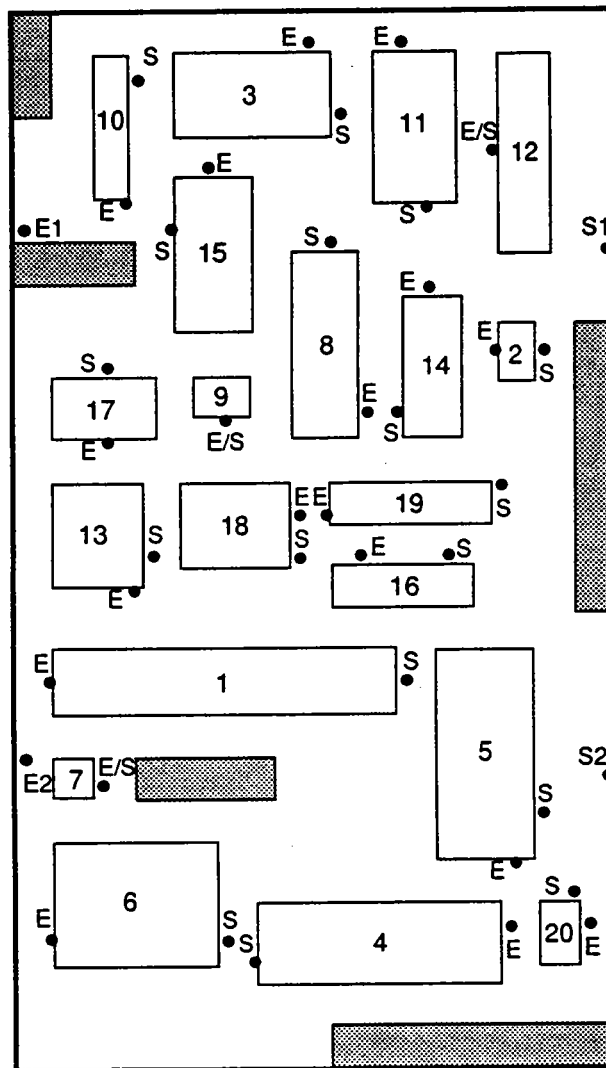


Figure Ae8 : Atelier à 20 cellules (configuration initiale)

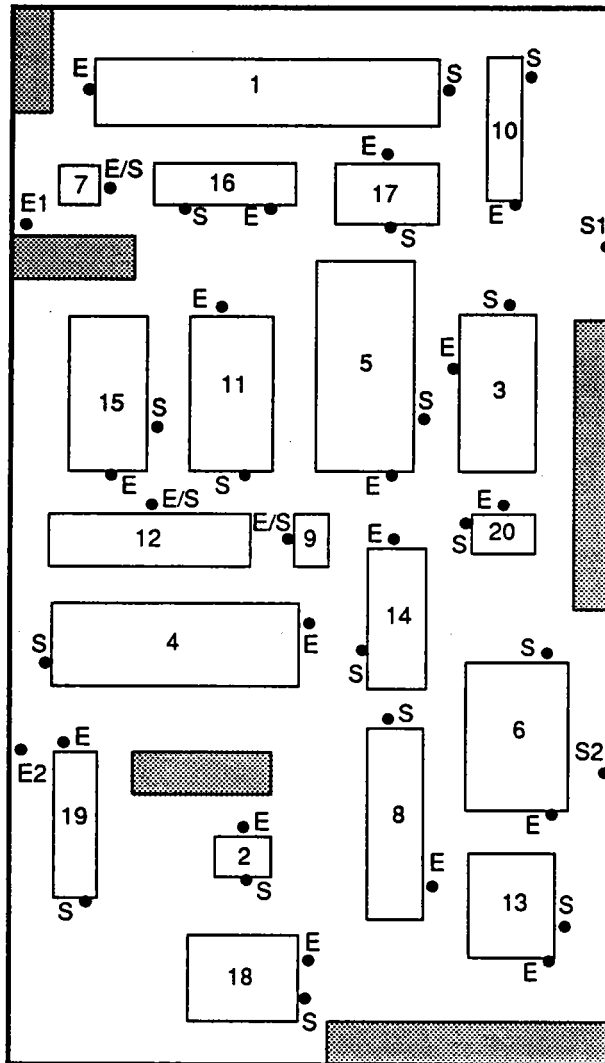
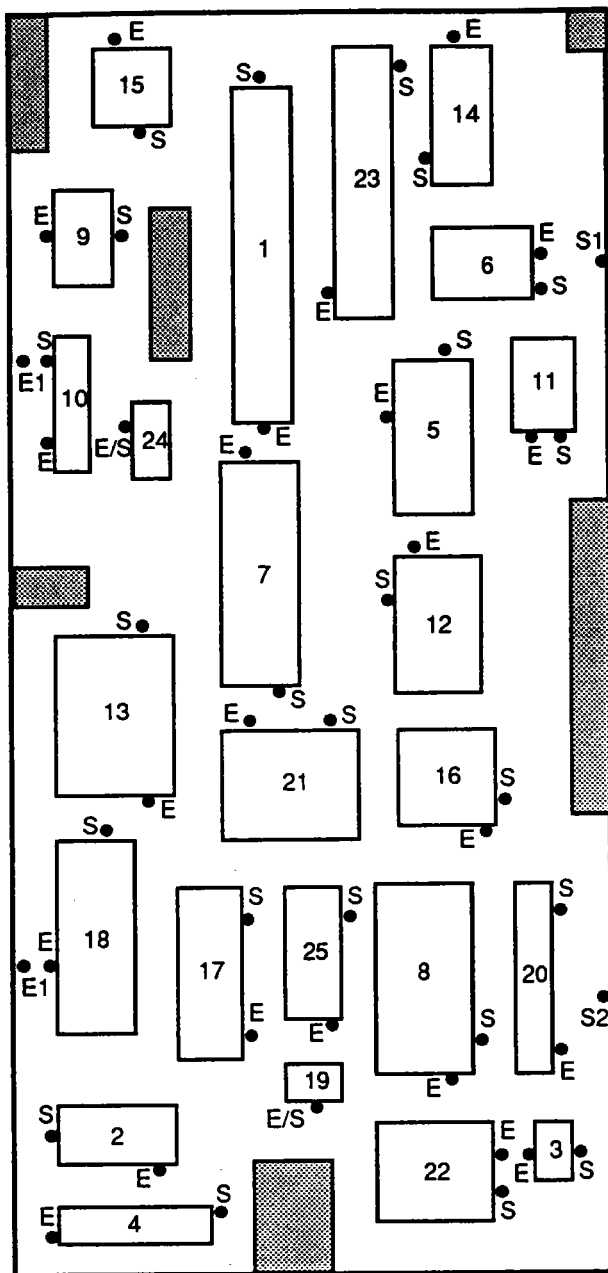


Figure Ae9 : Résultat final (20 cellules)

e. Exemple à 25 cellules

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	S1	S2
1	0	0	0	0	0	0	0	0	68	0	0	0	0	0	0	0	0	20	0	20	0	0	0	100	0	0	50
2	50	0	0	60	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	60
4	0	0	0	0	0	60	0	0	0	15	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0
5	20	0	90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0
6	0	50	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80	0
7	0	0	0	0	0	0	0	10	0	0	200	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	20	0	0	0	0	0	10	0
9	0	0	0	0	0	50	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	30	200	0	0	0	0	0
12	18	0	0	0	0	0	0	10	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	20
14	0	0	0	0	0	0	10	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	15	0
15	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
16	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0
18	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	100	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	10
20	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20
21	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	40	0	0	0	0	0	0	0	0	0	10	0
22	0	0	0	0	30	0	0	0	0	0	0	0	15	0	0	0	0	0	0	20	0	0	0	0	0	0	200
23	0	0	0	0	20	0	0	0	0	0	0	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100
25	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	10	0
Ea1	50	0	0	15	0	0	20	0	0	0	30	0	0	0	0	0	0	0	0	0	0	30	38	20	0	0	0
Ea2	0	60	0	0	0	0	200	10	0	0	10	0	0	0	100	10	0	0	0	20	0	0	0	0	0	0	0

Tableau Ae8 : Matrice des flux (atelier à 25 cellules)



0 8 ul
Echelle: []

Figure Ae10 : Atelier à 25 cellules

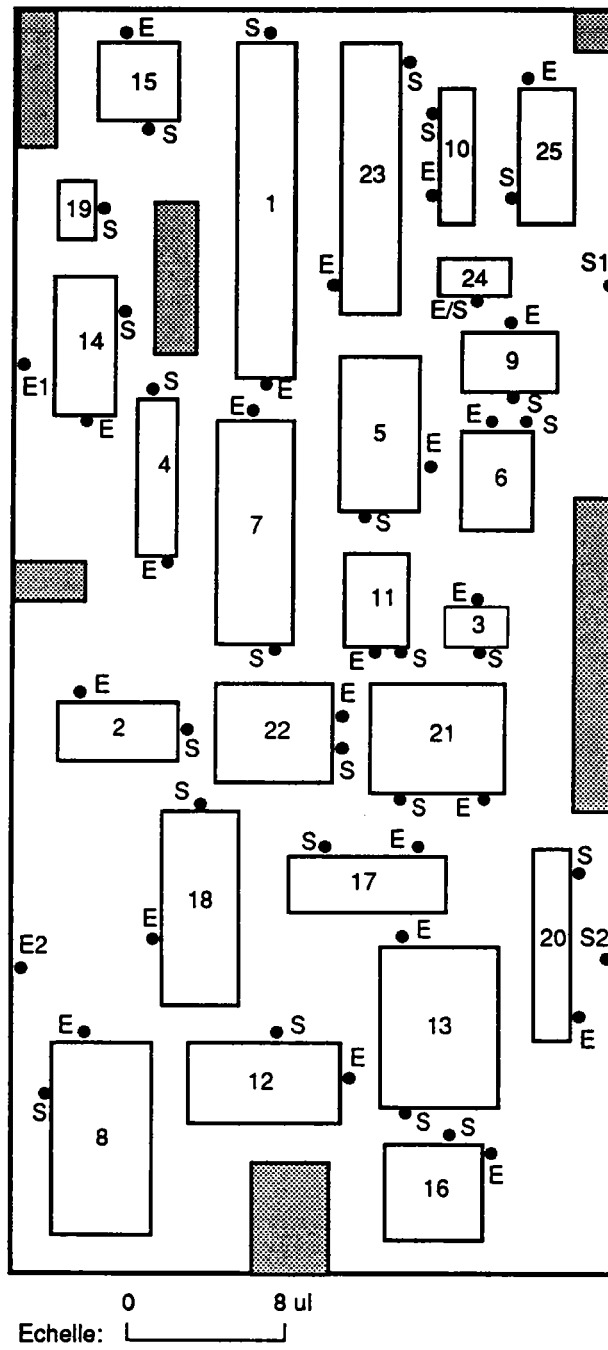


Figure Ae11 : Résultat final (25 cellules)

BIBLIOGRAPHIE

A large, empty rectangular box with a thin black border, occupying the lower two-thirds of the page. It is positioned to the right of a smaller rectangular box that contains the title. This area is intended for the user to list their references.

BIBLIOGRAPHIE

- Afentakis P., Millen R. A. et Solomon M. M. (1990), "*Dynamic Layout Strategies for Flexible Manufacturing Systems*", International Journal of Production Research, Vol. 28, N°2, pp. 311-323.
- Anciaux D. (1990), "*Etude d'agencement spacial dans un atelier de production selon le concept de technologie de groupe*", Thèse de doctorat de l'université de Metz, France.
- Azencott R. (1992), "*Simulated annealing : parallelization techniques*", John wiley, Inc. .
- Baker B.S., Coffman E.G. et Rivest R.L. (1980), "*Orthogonal packing in two dimensions*", Society of Industrial and Applied Mathematics Journal, Vol. 9, N° 4, pp. 846-855.
- Balas E. et Mazzola J.B. (1984), "*Nonlinear 0-1 Programming : I. Linearization Techniques*", Mathematical Programming, Vol. 30, pp. 1-22.
- Balakrishnan J., Jacobs. F. R. and Venkataramanan M. A. (1992), "*Solution for the constrained dynamic facility layout problem*", European Journal of Operational Research, Vol. 57, pp. 280-286.
- Batta R. (1987), "*Comment on the Dynamics of Plant Layout*", Management Science, Vol. 33, N°8, p1065.
- Bazaraa et El-shafei (1979), "*An exact branch-and-bound procedure for the quadratic assignment problem*", Naval Research Logistics Quarterly, Vol. 26, n°1, pp. 109-121.
- Bazaraa M.S. et Kirna O. (1983), "*A Branch-and-Bound-based heuristic for solving the QAP*", Naval Research Logistic Quarterly, Vol. 30, pp. 287-304.
- Bazaraa M.S. et Sherali. M.D. (1980), "*Benders' partitioning scheme applied to a new formulation of the quadratic assignement problem*", Naval Research Logistic Quarterly, Vol. 27, N° 1, pp. 29-41.

- Berkey J.O. et Wang P.Y. (1987), "*Two-dimensional finite bin-packing algorithms*", Journal of Operational Research Society, Vol. 38, N° 5, pp. 423-429.
- Buffa E.S., Armour G.C. et Vollman T.E. (1964), "*Allocating facilities with CRAFT*", Harvard Business Review, Vol. 42, pp. 136-158.
- Burkard R.E. et Bonninger T. (1983), "*A heuristic for quadratic Boolean program with applications to quadratic assignment problems*", European Journal of Operational Research, N°13, pp.374-386.
- Cerny V. (1985), "*Thermodynamical approach to the travelling salesman problem : An efficient simulation algorithm*", Journal of Optimization Theory and Applications, Vol. 45, pp. 41-51.
- Chu C.-H. (1989), "*Cluster Analysis in Manufacturing Cellular Formation*", OMEGA International Journal of Management Science, Vol. 17 N° 3 pp. 289-295.
- Dangelmaier W. (1984), "*Algorithmen und Verfahren zur Erstellung innerbetrieblicher Anordnungspläne*", Habilitationsschrift genehmigt von der Universität Stuttgart, Fakultät Fertigungstechnik, Hauptberichter : Prof. Dr.-Ing. H.-J. Warnecke, Stuttgart, Germany.
- Deisenroth. M.P. et Appel J.M. (1972), "*A computerized plant layout analysis and evaluation technique*", Annual AIIE Conference, Norcross, GA.
- Eglese R.W. (1990), "*Simulated annealing : a tool for operational research*", European Journal of Operational Research, Vol. 46, pp. 271-281.
- El-shefai (1977), "*Hospital layout as a quadratic assignment problem*", Operations Research Quarterly, Vol. 28, N° 1, pp. 167-197.
- Finke G., Burkard R.E. et Rendl F. (1985), "*Quadratic assignment Problem, Working paper*", Dept. of Applied Mathematics, Technical University of Nova Scotia, Halifax, NS, Canada.
- Fisher E.L. et Nof S.Y. (1984), "*FADES : Knowledge-based facility design*", Proc. Spring Industrial Engineering Conference, Chicago, IL, pp. 74-82.
- Gavett J.W. et Plyter N.V (1966), "*The optimal assignment of facilities to locations by branch-and-bound*", Operational Research, Vol. 14, pp. 210-232.

- Gilmore P.C. (1962), "*Optimal and suboptimal algorithms for the quadratic assignment problem*", Journal of the Society for Industrial and Applied Mathematics, Vol. 10, pp. 305-313.
- Gondran M. et Minoux M. (1990), "*Graphes et Algorithmes*", 2e édition, EDITIONS EYROLLES, Paris.
- Hamma S.B. et Bonnemoy C. (1991), "*La méthode du recuit simulé : Optimization globale dans \mathbb{R}^n* ", APPII, Vol. 25, pp. 477-496.
- Hamann T. (1992), "*Le problème d'agencement des ressources à l'intérieur des cellules des systèmes de fabrication*", Thèse de Doctorat, Université de Metz, France.
- Hamann T. et Vernadat F. (1992), "*The Intra-cell Layout Problem in Automated Manufacturing Systems*", CARs & FOF '92 8th International Conference on CAD/CAM, Robotics and Factories of the Future, Metz, France, pp. 695-708.
- Hamann T., Proth J.M., Souilah A., Vernadat F. et Xie X.L. (1992), "*COALA : A Manufacturing Plant Layout Approach*", The Eighth International IFIP WG5.3 Conference "PROLAMAT' 92" Man in CIM Systems, Tokyo, Japan, June 24-26.
- Harhalakis G., Proth J.M. et Xie X.L. (1990), "*Manufacturing cell design using simulated annealing : an industrial application*", Journal of Intelligent Manufacturing, Vol. 1, pp. 185-191.
- Harhalakis G. et Proth J.M. (1991), "*Some open problems in the design and use of modern production systems*", Applied Stochastic Models and Data Analysis, Vol. 7, N° 1, pp. 33-45.
- Hassan M.M.D. et Hogg G.L. (1987), "*A Review of Graph Theory Application to the Facilities Layout Problem*", OMEGA, International Journal of Management Science, Vol. 15, N° 4, pp. 291-300.
- Hassan M.M.D, Hogg G.L. et Smith D.R. (1986), "*SHAPE : A construction algorithm for area placement evaluation*", International Journal of production research, Vol. 24, N° 5, pp. 1283-1295.
- Heragu S. and Kusiak A. (1987), "*Analysis of expert systems in manufacturing design*", IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-17, N° 6, pp. 898-912.

- Hillier F.S. (1963), "*Quantitative tools for plant layout analysis*", Journal of Industrial Engineering, Vol. 14, pp. 33-40.
- Hillier F.S. et Connors M.M. (1966), "*Quadratic assignment problem algorithm and the location of indivisible facilities*", Management Science, Vol. 13, pp. 42-57.
- Kaku B.K. et Thompson G.L. Baybar I. (1986), "*A heuristic method for the multi-story layout problem*", European Journal of Operational Research, Vol. 37, pp. 384-397.
- Kaufman et Broeckx (1978), "*An algorithm for the quadratic assignment problem using Benders' decomposition*", European Journal of Operational Research, Vol. 2, pp. 204-211.
- Kirkpatrick S., Gelatt C.D. et Vecchi M.P. (1983), "*Optimization by simulated annealing*", SCIENCE, Vol. 220, N° 4598, pp. 671-680.
- Koopmans T.C. et Beckman M. (1957), "*Assignment problems and the locations of economic activities*", Econometrica, Vol. 25, N° 1, pp. 53-76.
- Kouvelis P. et Kiran A. S. (1991), "*Single and multiple period layout models for automated manufacturing systems*", European Journal of Operational Research, Vol. 52, pp. 300-314.
- Kouvelis, Kurawarwala et Gutiérrez (1992), "*Algorithms for robust single and multiple period layout planning for manufacturing systems*", European Journal of Operational Research, Vol. 63, pp. 287-303.
- Kusiak A. (1987), "*Artificial intelligence and operations research in flexible manufacturing systems*", INFOR, Vol. 25, N° 1, pp. 2-11.
- Kusiak A. (1990), "*Intelligent Manufacturing systems*", Prentice Hall, Englewood Cliffs, NJ.
- Lacksonen T. A. et Ensore E. E. (1993), "*Quadratic assignment algorithms for dynamic layouts problem*", International Journal of Production Research, Vol. 31, N°3, pp. 503-517.
- Lawler E.L. (1963), "*The quadratic assignment problem*", Management Science, Vol. 9, pp 586-599.

- Lee R.C. et Moore J.M (1967), "*CORELAP : COmputerized RElationship LAYout Planning*", Dept. of Industrial Engineering, Northeastern University, U.S.A. and the Journal of Industrial Engineering, Vol. 18, N° 3, pp. 195-200.
- Malakooti B. (1987), "*Computer-aided facility layout selection (CAFLAS) with applications to multiple criteria manufacturing planning problems*", Large Scale Systems, Vol. 12, pp. 109-123.
- Minten B. (1977), "*Beitrag zur rechnergestützten Fabrikplanung*", Diss. Universität Stuttgart, Stuttgart, Germany.
- McAuley J. (1972), "*Machine grouping for efficient production*", The production Engineer, Vol. 52, pp. 53-57.
- Montreuil B. et Venkatadri U. (1991), "*Strategic Interpolative Design of Dynamic Manufacturing Systems Layout*", Management Science, Vol. 37, N°6, pp. 682-694.
- Montreuil B. et Laforge A. (1992), "*Dynamic layout design given a scenario tree of probable futures*", Euroean Journal of Operation Research, Vol. 63, pp. 271-286.
- Nozari A. et Ensore Jr.E.E (1981), "*Computerized facility layout with graph theory*", Computer & Industrial Engineering, Vol. 5, N° 3, pp. 183-193.
- Nugent C.E., Vollmann T.E. et Ruml R. (1968), "*An experimental comparison of techniques for the assignment of facilities to locations*", Operations Research, Vol. 16, pp. 150-173.
- Proth J.M. et Souilah A. (1992), "*Near-Optimal Layout Algorithm Based on Simulated Annealing*", International Journal of Systems Automation : Research and Applications (SARA), Vol. 2, pp. 227-243.
- Rajagopalan, R. et Batra J. L. (1975), "*Design of Cellular Production Systems - A Graph Theoretic Approach*", International Journal of Production Research, Vol. 13, N°6, pp. 567, 579.
- Rose J., Klebsch W. et Wolf J. (1990), "*Temperature Measurement and equilibrium dynamics of Simulated Annealing placement*", IEEE Transactions on Computer Aided Design, Vol. 9, N° 3, pp. 253-259.
- Rosenblatt M. J. (1986), "*The Dynamics of Plant Layout*", Management Science, Vol. 32, N°1, pp. 76-86.

- Rosenblatt M. J. et Lee H. L. (1986), "*A robustness approach to facilities design*", International Journal of Production Research, Vol. 25, N°4, pp. 479-486.
- Sahni S. et Gonzalez T. (1976), "*P-complete approximation problem*", Journal of Associated Computing Machinery, Vol. 23, N°3, pp. 555-565.
- Sechen (1990), "*Chip-Planning, Placement, and Global Routing of Macro-cell Integrated Circuits Using Simulated Annealing*", International Journal of Computer Aided VLSI Design, Vol. 2, pp. 127-158.
- Scriabin M. et Vergin R. C. (1985), "*A cluster analysis approach to facility layout*", Management Science, Vol. 31, N°1, pp. 33-49.
- Sedgewick R. (1991), "*Algorithmes en langage C*", InterEditions, Paris.
- Seehof J.M. et Evens W.O. (1967), "*Automated layout design program*", The Journal of Industrial Engineering, Vol. 18, N° 2, pp. 690-695.
- Seifoddini H., (1989), "*Duplication Process in Machine Cells Formation in Group Technology*", IIE Transactions, Vol. 21, N° 4, pp. 382-388.
- Siarry P. et Dreyfus G. (1988), "*La méthode du recuit simulé : théorie et applications*", I.D.S.E.T, Paris.
- Singh N. (1993) "*Editorial of the special issue on Cellular Manufacturing Systems*", European Journal of Operational Research, Vol. 69, N° 3, pp. 281-283.
- Tam K. Y. (1992) "*A simulated annealing algorithm for allocating space to manufacturing cells*", International Journal of Production Research, Vol. 30, N° 1, pp. 63-87.
- Tirumpatikumara S.R., Kashyap R.L. et Moodie C.L. (1985), "*Artificial intelligence techniques in facilities layout planning : The development of an expert system*", Technical report TR-ERC-86-1, Purdue University, West Lafayette, Ind.
- Tompkins J.A. et Reed R. Jr. (1976), "*An applied model for the facilities design problem*", International Journal of Production Research, Vol. 14, N° 5, pp. 583-595.
- Urban T. (1981), "*A multiple criteria model for the facility layout problem*", International Journal Production Research, Vol. 25, N° 12, pp. 183-193.

- Urban T. (1992), "*Computational performance and efficiency of lower-bound procedures for the dynamic facility layout problem*", European Journal of Operation Research, Vol. 57, pp. 271-279.
- Van. Laarhoven P. J. M. et Aarts E. H. (1987), "*Simulated annealing : Theory and Application*", (Dordrecht : D. reidel Publishing Co.).
- Warnecke H.J., Steinhilper R. Roth H.-P. et Weber T. (1981), "*Planning and Realization of FMS : Case studies, experiences and recommendations*", Fraunhofer Institute for Manufacturing Engineering and automation (IPA) Stuttgart, Germany.
- Xie X-L, (1993) "*Manufacturing Cell Formation under Capacity Constraints*", À paraître dans Applied Stochastic Models and Data Analysis.

LES SYSTÈMES CELLULAIRES DE PRODUCTION : L'AGENCEMENT INTER-CELLULES

Résumé

Le but de cette thèse est d'apporter une contribution à la résolution du problème de l'agencement des systèmes cellulaires de fabrication. Nous introduisons d'abord le concept de systèmes cellulaires de fabrication. Puis, nous décrivons l'approche globale proposée pour l'agencement de ce type de systèmes. Cette démarche se déroule en quatre étapes. Nous donnons les trois premières étapes (pré-traitement des données, formation des cellules et agencement intra-cellulaire) et expliquons les méthodes utilisées. Nous consacrons la deuxième partie de cette thèse à la quatrième étape qui est l'agencement des cellules dans l'atelier de fabrication. Il s'agit, dans cette étape, de trouver une configuration du système qui minimise la somme des produits (coût x distance) entre les entrées/sorties des cellules, tout en respectant certaines contraintes physiques. Ce problème combinatoire est résolu en utilisant une méthode par améliorations basée sur l'algorithme du recuit simulé. Nous avons besoin, pour cela, de calculer les longueurs des plus courts chemins entre les entrées/sorties des cellules. Deux algorithmes sont proposés pour résoudre ce problème. A la fin de la deuxième partie, nous présentons une introduction au problème de l'agencement dynamique des systèmes de fabrication et montrons où et comment se pose ce problème dans les systèmes cellulaires. La troisième partie est consacrée au logiciel réalisé.

Mots clés

Systèmes cellulaires, Agencement, Optimisation, Recuit simulé, Plus court chemin, Agencement dynamique.

CELLULAR MANUFACTURING SYSTEMS: INTER-CELL LAYOUT DESIGN

Abstract

The aim of this thesis is to examine the cellular manufacturing system layout design problem. First, the concept of cellular manufacturing systems is introduced. Then, a four step approach proposed for the layout design of this kind of systems is introduced. The first three steps (data pre-processing, cell design and intra-cell layout design) are introduced and the methods which are used are provided. The second part of this thesis is dedicated to the fourth step, namely the inter-cell layout design. The goal is to find a layout which minimises the sum of the products (flow x distance) between the inputs/outputs of the cells, with respect to some physical constraints. This combinatorial problem is solved using an iterative method based on the simulated annealing method. For that purpose, we have to compute the lengths of the shortest paths between the inputs/outputs of the cells. Two algorithms are proposed. At the end of the second part, we introduce the dynamic layout design concept and define its fields of application. The third part is devoted to the software developed.

Key words

Cellular manufacturing systems, Layout design, Optimization, Simulated Annealing, Shortest path computation, Dynamic layout.