

Algorithmes d'implantation et tracé des connexions par fils isolés

Ion Floricica

▶ **To cite this version:**

Ion Floricica. Algorithmes d'implantation et tracé des connexions par fils isolés. Modélisation et simulation. Université Joseph-Fourier - Grenoble I; Institut National Polytechnique de Grenoble - INPG, 1974. Français. NNT : . tel-00284623

HAL Id: tel-00284623

https://theses.hal.science/tel-00284623

Submitted on 3 Jun 2008

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.

THESE

présentée à

UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

par

pour obtenir le grade de

Docteur de troisième cycle

Spécialité : INFORMATIQUE

Ion FLORICICA

Algorithmes d'implantation et tracé des connexions par fils isolés

Thèse soutenue le 28 juin 1974 devant la Commission d'Examen :

Président : Monsieur J. KUNTZMANN

Examinateurs Madame G. SAUCIER
Monsieur D. RANDET
Monsieur M. CHEIN

UNIVERSITE SCIENTIFIQUE
ET MEDICALE DE GRENOBLE

LISTE DES PROFESSEURS

Président : Monsieur Michel SOUTIF
Vice-Président : Monsieur Gabriel CAU

PROFESSEURS TITULAIRES

MM.	ANGLES D'AURIAC Paul	Mécanique des fluides
	ARNAUD Georges	Clinique des maladies infectieuses
	ARNAUD Paul	Chimie
	AUBERT Guy	Physique
	AYANT Yves	Physique approfondie
Mme	BARBIER Marie-Jeanne	Electrochimie
MM.	BARBIER Jean-Claude	Physique expérimentale
	BARBIER Reynold	Géologie appliquée
	BARJON Robert	Physique nucléaire
	BARNOUD Fernand	Biosynthèse de la cellulose
	BARRA Jean-René	Statistiques
	BARRIE Joseph	Clinique chirurgicale
	BENOIT Jean	Radioélectricité
	BERNARD Alain	Mathématiques Pures
	BESSON Jean	Electrochimie
	BEZES Henri	Chirurgie générale
	BLAMBERT Maurice	Mathématiques Pures
	BOLLIET Louis	Informatique (IUT B)
	BONNET Georges	Electrotechnique
	BONNET Jean-Louis	Clinique ophtalmologique
	BONNET-EYMARD Joseph	Pathologie médicale
	BONNIER Etienne	Electrochimie Electrométallurgie
	BOUCHERLE André	Chimie et Toxicologie
	BOUCHEZ Robert	Physique nucléaire
	BOUSSARD Jean-Claude	Mathématiques Appliquées
	BRAVARD Yves	Géographie
	BRISSONNEAU Pierre	Physique du solide
	BUYLE-BODIN Maurice	Electronique
	CABANAC Jean	Pathologie chirurgicale
	CABANEL Jean	Clinique rhumatologique et hydrologie
	CALAS François	Anatomie
	CARRAZ Gilbert	Biologie animale et pharmacodynamie
	CAU Gabriel	Médecine légale et Toxicologie
	CAUQUIS Georges	Chimie organique
	CHABAUTY Claude	Mathématiques Pures
	CHARACHON Robert	Oto-Rhino-Laryngologie
	CHATEAU Robert	Thérapeutique
	CHENE Marcel	Chimie papetière
	COEUR André	Pharmacie chimique
	CONTAMIN Robert	Clinique gynécologique
	COUDERC Pierre	Anatomie Pathologique
	CRAYA Antoine	Mécanique

Mme	DEBELMAS Anne-Marie	Matière médicale
MM.	DEBELMAS Jacques	Géologie générale
	DEGRANGE Charles	Zoologie
	DESRE Pierre	Métallurgie
	DESSAUX Georges	Physiologie animale
	DODU Jacques	Mécanique appliquée
	DOLIQUE Jean-Michel	Physique des plasmas
	DREYFUS Bernard	Thermodynamique
	DUCROS Pierre	Cristallographie
	DUGOIS Pierre	Clinique de Dermatologie et Syphiligraphie
	FAU René	Clinique neuro-psychiatrique
	FELICI Noël	Electrostatique
	GAGNAIRE Didier	Chimie physique
	GALLISSOT François	Mathématiques Pures
	GALVANI Octave	Mathématiques Pures
	GASTINEL Noël	Analyse numérique
	GEINDRE Michel	Electroradiologie
	GERBER Robert	Mathématiques Pures
	GIRAUD Pierre	Géologie
	KLEIN Joseph	Mathématiques Pures
Mme	KOFLER Lucie	Botanique et Physilogie végétale
MM.	KOSZUL Jean-Louis	Mathématiques Pures
	KRAVTCHENKO Julien	Mécanique
	KUNTZMANN Jean	Mathématiques appliquées
	LACAZE Albert	Thermodynamique
	LACHARME Jean	Biologie végétale
	LAJZEROWICZ Joseph	Physique
	LATREILLE René	Chirurgie générale
	LATURAZE Jean	Biochimie pharmaceutique
	LAURENT Pierre-Jean	Mathématiques appliquées
	LEDRU Jean	Clinique médicale B
	LLIBOUTRY Louis	Géophysique
	LOUP Jean	Géographie
Mle	LUTZ Elisabeth	Mathématiques Pures
MM.	MALGRANGE Bernard	Mathématiques Pures
	MALINAS Yves	Clinique obstétricale
	MARTIN-NOEL Pierre	Seméiologie médicale
	MASSEPORT Jean	Géographie
	MAZARE Yves	Clinique médicale A
	MICHEL Robert	Minéralogie et Pétrographie
	MOURIQUAND Claude	Histologie
	MOUSSA André	Chimie nucléaire
	NEEL Louis	Physique du solide
	OZENDA Paul	Botanique
	PAUTHENET René	Electrotechnique
	PAYAN Jean-Jacques	Mathématiques Pures
	PEBAY-PEYROULA Jean-Claude	Physique
	PERRET René	Servomécanismes
	PILLET Emile	Physique industrielle
	RASSAT André	Chimie systématique
	RENARD Michel	Thermodynamique
	REULOS René	Physique industrielle
	RINALDI Renaud	Physique
	ROGET Jean	Clinique de pédiatrie et de puériculture
	SANTON Lucien	Mécanique
	SEIGNEURIN Raymond	Microbiologie et Hygiène
	SENGEL Philippe	Zoologie
	SILBERT Robert	Mécanique des fluides
	SOUTIF Michel	Physique générale

MM.	TANCHE Maurice	Physiologie
	TRAYNARD Philippe	Chimie générale
	VAILLAND François	Zoologie
	VALENTIN Jacques	Physique nucléaire
	VAUQUOIS Bernard	Calcul électronique
Mme	VERAIN Alice	Pharmacie galénique
M.	VERAIN André	Physique
Mme	VEYRET Germaine	Géographie
MM.	VEYRET Paul	Géographie
	VIGNAIS Pierre	Biochimie médicale
	YOCCOZ Jean	Physique nucléaire théorique

PROFESSEURS ASSOCIES

MM.	BULLEMER Bernhard	Physique
	HANO JUN-ICHI	Mathématiques Pures
	STEPHENS Michaël	Mathématiques appliquées

PROFESSEURS SANS CHAIRE

MM.	BEAUDOING André	Pédiatrie
Mme	BERTRANDIAS Françoise	Mathématiques Pures
MM.	BERTRANDIAS Jean-Paul	Mathématiques appliquées
	BIAREZ Jean-Pierre	Mécanique
	BONNETAIN Lucien	Chimie minérale
Mme	BONNIER Jane	Chimie générale
MM.	CARLIER Georges	Biologie végétale
	COHEN Joseph	Electrotechnique
	COUMES André	Radioélectricité
	DEPASSEL Roger	Mécanique des fluides
	DEPORTES Charles	Chimie minérale
	GAUTHIER Yves	Sciences biologiques
	GAVEND Michel	Pharmacologie
	GERMAIN Jean-Pierre	Mécanique
	GIDON Paul	Géologie et Minéralogie
	GLENAT René	Chimie organique
	HACQUES Gérard	Calcul numérique
	JANIN Bernard	Géographie
Mme	KAHANE Josette	Physique
MM.	MULLER Jean-Michel	Thérapeutique
	PERRIAUX Jean-Jacques	Géologie et Minéralogie
	POULOUJADOFF Michel	Electrotechnique
	REBECQ Jacques	Biologie (CUS)
	REVOL Michel	Urologie
	REYMOND Jean-Charles	Chirurgie générale
	ROBERT André	Chimie papetière
	DE ROUGEMONT Jacques	Neurochirurgie
	SARRAZIN Roger	Anatomie et chirurgie
	SARROT-REYNAULD Jean	Géologie
	SIBILLE Robert	Construction mécanique
	SIROT Louis	Chirurgie générale
Mme	SOUTIF Jeanne	Physique générale

MAITRES DE CONFERENCES ET MAITRES DE CONFERENCES AGREGES

Mle	AGNIUS-DELORD Claudine	Physique pharmaceutique
	ALARY Josette	Chimie analytique
MM.	AMBLARD Pierre	Dermatologie
	AMBROISE-THOMAS Pierre	Parasitologie
	ARMAND Yves	Chimie
	BEGUIN Claude	Chimie organique
	BELORIZKY Elie	Physique
	BENZAKEN Claude	Mathématiques appliquées
	BILLET Jean	Géographie
	BLIMAN Samuel	Electronique (EIE)
	BLOCH Daniel	Electrotechnique
Mme	BOUCHE Liane	Mathématiques (CUS)
MM.	BOUCHET Yves	Anatomie
	BOUVARD Maurice	Mécanique des fluides
	BRODEAU François	Mathématiques (IUT B)
	BRUGEL Lucien	Energétique
	BUISSON Roger	Physique
	BUTEL Jean	Orthopédie
	CHAMBAZ Edmond	Biochimie médicale
	CHAMPETIER Jean	Anatomie et organogénèse
	CHIAVERINA Jean	Biologie appliquée (EFP)
	CHIBON Pierre	Biologie animale
	COHEN-ADDAD Jean-Pierre	Spectrométrie physique
	COLOMB Maurice	Biochimie médicale
	CONTE René	Physique
	COULOMB Max	Radiologie
	CROUZET Guy	Radiologie
	DURAND Francis	Métallurgie
	DUSSAUD René	Mathématiques (CUS)
Mme	ETERRADOSSI Jacqueline	Physiologie
MM.	FAURE Jacques	Médecine légale
	GENSAC Pierre	Botanique
	GIDON Maurice	Géologie
	GRIFFITHS Michaël	Mathématiques appliquées
	GROULADE Joseph	Biochimie médicale
	HOLLARD Daniel	Hématologie
	HUGONOT Robert	Hygiène et Médecine préventive
	IDELMAN Simon	Physiologie animale
	IVANES Marcel	Electricité
	JALBERT Pierre	Histologie
	JOLY Jean-René	Mathématiques Pures
	JOUBERT Jean-Claude	Physique du solide
	JULLIEN Pierre	Mathématiques Pures
	KAHANE André	Physique générale
	KUHN Gérard	Physique
	LACOUME Jean-Louis	Physique
Mme	LAJZEROWICZ Jeannine	Physique
MM.	LANCIA Roland	Physique atomique
	LE JUNTER Noël	Electronique
	LEROY Philippe	Mathématiques
	LOISEAUX Jean-Marie	Physique nucléaire
	LONGEQUEUE Jean-Pierre	Physique nucléaire
	LUU DUC Cuong	Chimie organique
	MACHE Régis	Physiologie végétale
	MAGNIN Robert	Hygiène et Médecine préventive
	MARECHAL Jean	Mécanique
	MARTIN-BOUYER Michel	Chimie (CUS)

MM.	MAYNARD Roger	Physique du solide
	MICHOULIER Jean	Physique (IUT A)
	MICOUD Max	Maladies infectieuses
	MOREAU René	Hydraulique (INP)
	NEGRE Robert	Mécanique
	PARAMELLE Bernard	Pneumologie
	PECCOUD François	Analyse (IUT B)
	PEFFEN René	Métallurgie
	PELMONT Jean	Physiologie animale
	PERRET Jean	Neurologie
	PERRIN Louis	Pathologie expérimentale
	PFISTER Jean-Claude	Physique du solide
	PHELIP Xavier	Rhumatologie
Mlle	RIERY Yvette	Biologie animale
MM.	RACHAIL Michel	Médecine interne
	RACINET Claude	Gynécologie et obstétrique
	RENAUD Maurice	Chimie
	RICHARD Lucien	Botanique
Mme	RINAUDO Marquerite	Chimie macromoléculaire
MM.	ROMIER Guy	Mathématiques (IUT B)
	SHOM Jean-Claude	Chimie générale
	STIEGLITZ Paul	Anesthésiologie
	STOEBNER Pierre	Anatomie pathologique
	VAN CUTSEM Bernard	Mathématiques appliquées
	VEILLON Gérard	Mathématiques appliquées (INP)
	VIALON Pierre	Géologie
	VOOG Robert	Médecine interne
	VROUSSOS Constantin	Radiologie
	ZADWORNY François	Electronique

MAITRES DE CONFERENCES ASSOCIES

MM.	BOUDOURIS Georges	Radioélectricité
	CHEEKE John	Thermodynamique
	GOLDSCHMIDT Hubert	Mathématiques
	SIDNEY STUARD	Mathématiques Pures
	YACOUD Mahmoud	Médecine légale

CHARGES DE FONCTIONS DE MAITRES DE CONFERENCES

Mme	BERIEL Hélène	Physilogie
Mme	RENAUDET Jacqueline	Microbiologie

Fait le 30 mai 1972.

Je tiens à exprimer ma reconnaissance à Monsieur le Professeur J. KUNTZMANN pour l'attention qu'il a portée à ce travail et pour m'avoir fait l'honneur de présider le Jury.

Je remercie vivement

- Madame G. SAUCIER, Maître de Conférences à l'ENSIMAG, qui a dirigé ce travail et dont l'aide m'a permis de le mener à bien.

- Monsieur D. RANDET, Directeur du LETI/EPA, qui fut à l'origine de cette thèse et m'a assuré l'environnement nécessaire pour la mener à bien.

- Monsieur M. CHEIN, Maître de Conférences à l'Université de Paris VI, qui a bien voulu faire partie du Jury.

Je remercie sincèrement Mlle C. TURCAT, dont la coopération et les remarques m'ont été très utiles, ainsi que MM. D. MAUDUIT, G. NICOLAS avec lesquels j'ai eu de nombreuses et fructueuses discussions.

Je tiens aussi à remercier toutes les personnes qui ont contribué à la réalisation matérielle de cet ouvrage.

INTRODUCTION

Dans la conception des calculateurs électroniques, l'optimisation des circuits devient un problème de plus en plus important, au fur et à mesure de l'évolution des composants en nombre et complexité.

Les procédures d'optimisation rentrent dans un processus d'étude automatique, apparu dès le développement des calculateurs, dont les principaux niveaux sont :

1) - passage du schéma électrique au circuit intégré
2) - réalisation de la plaquette à l'aide de circuits intégrés et (ou) de composants discrets
3) - assemblage des plaquettes sur des bâtis.

Notre travail a pour objet le deuxième niveau, c'est-à-dire l'élaboration de procédures pour :

a) - l'implantation de modules de circuits intégrés sur la plaquette
b) - le tracé des connexions.

Les méthodes appliquées, ainsi que les buts suivis dans le cadre des deux étapes a) et b) sont fortement influencés par l'évolution technologique de fabrication des plaquettes.

La matérialisation des connexions par circuits imprimé, procédé utilisé actuellement, interdit les croisements des liaisons sur une même couche, ce qui soulève des problèmes complémentaires tels que : la minimisation du nombre des couches, la répartition des liaisons entre les couches, la définition du tracé des connexions sur chaque couche, problèmes qui sont rendus encore plus complexes dans le cas d'une grande densité de composants.

La technique du tracé des connexions par fils isolés, améliorée et reprise par de nouvelles machines à tracer, ouvre de nouvelles perspectives à la construction des plaquettes.

Du point de vue de matériel, par l'utilisation d'un fil très fin (jusqu'à 30 microns de diamètre) et par l'implantation directe des "puces" de circuits intégrés, en éliminant le "boîtier" englobant, on arrive à de hautes densités d'implantation ou à des plaquettes réduites comme dimensions.

D'autre part, l'étude de réalisation des connexions se trouve simplifiée : les croisements étant possibles, les problèmes complémentaires précédents disparaissent. Le tracé dépend étroitement de l'implantation des modules sur la plaquette et la plupart des contraintes imposées à la réalisation de la plaquette sont résolues par l'algorithme d'implantation. L'implantation acquiert donc une importance particulière.

Les algorithmes d'implantation que nous donnons gardent un caractère général, tandis que le tracé des connexions est spécifique à la technique des fils isolés ; un programme efficace pour plusieurs formes de technologies, comporterait un nombre de paramètres optionnels élevé et pratiquement il serait inexploitable.

Le travail présenté est divisé en deux parties :

- La première partie traite le problème d'implantation de modules. Nous cherchons à placer les modules sur la plaquette, tels que la longueur totale de leurs liaisons soit plus courte possible.

Dans le chapitre I, nous réduisons le problème d'implantation à un problème d'affectation quadratique, en considérant le graphe complet de chaque équipotentielle. Le problème ainsi associé est résolu par une méthode heuristique en deux étapes :

. on construit d'abord une implantation initiale
. on améliore cette implantation, par un algorithme de "changement de paires".

Dans le chapitre II, la méthode d'implantation précédente est étendue, pas l'utilisation des sous-réseaux minimaux, à des plaquettes de grande taille (∿ 100 modules ou plus). L'ensemble de tous les modules est partagé en plusieurs sous-ensembles, les liaisons étant peu nombreuses entre les sous-ensembles, mais nombreuses à l'intérieur de chaque sous-ensemble. L'algorithme d'implantation initiale et d'amélioration de l'implantation initiale est appliqué pour chaque sous-réseau ainsi trouvé.

- La deuxième partie est consacrée au tracé des connexions.

Dans le chapitre I, nous cherchons l'arbre minimal de l'équipotentielle, qui en fait sera une chaîne à cause de la contrainte technologique imposée sur les sommets de l'arbre (degré $\leq$ 2). Le problème revient à la recherche des circuits hamiltoniens minimaux et sera résolu par la méthode combinatoire "branch-and-bound" de Little et ... [20].

Après avoir trouvé les équipotentielles minimales, nous cherchons, dans le chapitre II, à minimiser le nombre total de coudes des liaisons du réseau, par rotation du module dans son emplacement. On appliquera un algorithme à solution optimale, adapté de la méthode combinatoire "branch-and-bound" et une méthode heuristique.

Pour les 2 algorithmes, nous élaborons aussi une procédure simple et rapide qui donnera une bonne configuration initiale en simulant les liaisons par des "moments de rotation".

Dans le chapitre III, nous abordons le problème du tracé proprement dit, en définissant le chemin de chaque connexion. Nous cherchons à éviter les croisements, bien qu'ils soient possibles, et respecter les contraintes imposées pour la réalisation des connexions : éviter les virages interdits, etc. La réalisation effective des connexions étant faite par un fil continu, il faut inclure les connexions dans une chaîne hamiltonienne à liaisons supplémentaires plus courtes que possibles.

TABLE DES MATIERES

PARTIE A

IMPLANTATION DE MODULES DE CIRCUITS INTEGRES

CHAPITRE I - IMPLANTATION EN MINIMISANT LA LONGUEUR TOTALE DES LIAISONS ... 3.

I.0. - Introduction ..... 3.

I.1. - Notations et définitions ..... 4.

I.2. - Rappel de certaines méthodes ..... 7.

I.3. - Conclusion ..... 9.

I.4. - Algorithmes d'implantation ..... 10.

I.4.0. - Définitions et notations ..... 10.

I.4.1. - Détermination d'une implantation initiale ..... 11.

I.4.1.1. - Description de l'algorithme ..... 11.

I.4.1.2. - Algorithme ..... 12.

I.4.2. - Amélioration de l'implantation initiale ..... 14.

I.4.2.1. - Principes de calcul ..... 15.

I.4.2.2. - Algorithme ..... 17.

I.4.2.3. - Estimation du nombre de calculs ..... 18.

I.4.2.4. - Exemple ..... 19.

CHAPITRE II - IMPLANTATION DE MODULES PAR SOUS-RESEAUX MINIMAUX ..... 22.

II.1. - Définitions et propriétés ..... 23.

II.2. - Etapes du déroulement de l'algorithme ..... 24.

II.3. - Algorithme ..... 26.

PARTIE B

TRACE DES CONNEXIONS

CHAPITRE I - RECHERCHE DES EQUIPOTENTIELLES MINIMALES ..... 31.

I.1. - Formulation du problème..... 31.

I.2. - Méthodes de résolution ..... 32.

I.3. - Répartition uniforme des liaisons sur la plaquette ..... 33.

I.4. - Equipotentielles minimales et répartition uniforme des liaisons ..... 34.

CHAPITRE II - MINIMISATION DU NOMBRE DE COUDES D'UN RESEAU DE MODULES DE CIRCUITS INTEGRES..... 36.

II.0. - Présentation du problème ..... 36.

II.1. - Définitions et tableaux des nombres minimum de coudes ..... 38.

II.2. - Algorithme à solution optimale pour minimiser le nombre de coudes dans un réseau..... 41.

II.2.1. - Généralités et définitions ..... 41.

II.2.2. - Description de l'algorithme et simplifications de calcul..... 45.

II.2.3. - Algorithme..... 46.

II.3. - Construction d'une configuration initiale ..... 50.
II.3.1. - Définitions et propriétés ..... 50.
II.3.2. - Algorithme et estimation du nombre moyen de calculs 53.
II.4. - Algorithme heuristique pour minimiser le nombre de coudes dans un réseau ..... 56.
II.4.1. - Préliminaires ..... 57.
II.4.2. - Notations, définitions et propriétés ..... 60.
II.4.3. - Algorithme, exemple ..... 70.
II.4.4. - Calcul de α, γ, δ 72
II.4.5. - Estimation du nombre de calculs ..... 74.

CHAPITRE III - REALISATION DES EQUIPOTENTIELLES ..... 75.

III.1. - Critique des algorithmes de tracé connus ..... 75.
III.1.1. - Classification suivant le type de la plaquette .. 76.
III.1.2. - Classification suivant le principe du tracé d'une connexion ..... 77.
III.1.3. - Algorithmes à discrétisation de la plaquette .... 79.
III.1.4. - Algorithmes utilisant des segments ..... 81.
III.1.5. - Conclusion ..... 81.
III.2. - Réalisation des connexions par fils isolés ..... 82.
III.3. - Tracé des connexions ..... 85.
III.3.0. - Justification du choix du tracé en horizontal et vertical ..... 85.
III.3.1. - Génération de la plaquette et introduction des données ..... 87.
III.3.2. - Elaboration des connexions ..... 89.
III.3.2.1. - Contraintes du tracé ..... 90.
III.3.2.2. - Virages interdits et croisements ... 91.
III.3.2.3. - Zones des canaux ..... 93.
III.3.2.4. - Chemins équivalents ..... 94.
III.3.3. - Tracé effectif des connexions ..... 98.
III.3.3.1. - Recherche de chaines de connexions . 99.
III.3.3.2. - Liaisons supplémentaires et exemple. 100.
III.3.3.3. - Génération des commandes supplémentaires ..... 102.

III.3.4. - Sortie des résultats ..... 103.

CONCLUSION GENERALE ..... 104.

ANNEXE 1- ..... 105.

ANNEXE 2- ..... 107.

BIBLIOGRAPHIE -

oOo

A - IMPLANTATION DE MODULES DE CIRCUITS INTEGRES SUR LA PLAQUETTE

Introduction

La difficulté de l'implantation de modules sur la plaquette provient des contraintes imposées pour la réalisation du circuit total. Les contraintes sont nombreuses et variées : elles sont de nature électrique :

- éviter les capacités ou les inductances parasites qui perturbent le fonctionnement des modules,
- assurer une grande vitesse de propagation des signaux dans les fils en diminuant leur longueur,
- éviter les croisements des fils, etc...,

ou imposées par le concepteur de telle façon que l'algorithme du tracé, utilisé ultérieurement, donne un bon tracé des connexions :

- minimiser la longueur totale des liaisons,
- minimiser le nombre de coudes des fils tracés en horizontal et vertical sur la plaquette, etc ...

En pratique, il est impossible de tenir compte de toutes les contraintes. Généralement, la minimisation de la longueur totale des liaisons est le critère le plus sûr pour satisfaire d'une manière acceptable la plupart des contraintes de nature électrique et aussi pour faciliter le tracé des connexions.

Pour le problème pratique que nous résolvons ici, nous cherchons à :

1) minimiser la longueur totale des liaisons
2) minimiser le nombre de coudes
3) éviter les croisements.

Le problème d'implantation de modules sera résolu dans le cas où les emplacements de la plaquette, devant recevoir les modules, sont connus et fixés à l'avance. Nous nous intéresserons au cas le plus fréquent : celui où les emplacements sont fixés de façon régulière suivant des lignes et des colonnes, mais notre algorithme possède aussi une option pour des emplacements non régulièrement distribués sur la plaquette.

CHAPITRE I

IMPLANTATION EN MINIMISANT LA LONGUEUR TOTALE DES LIAISONS

I.0. INTRODUCTION

Au problème d'implantation en minimisant la longueur totale des liaisons peuvent être associés (cf. [1]) deux problèmes mathématiques.

1) le problème d'affectation simple : étant donné le "coût" a_{ij} de l'affectation de l'objet i à l'objet j, où $i,j \in \{1,2,\ldots,n\}$, trouver la permutation p sur $\{1,2,\ldots,n\}$ telle que :
$\sum_i a_{ip(i)}$ soit minimum.

Ce problème est résolu par des méthodes qui donnent soit la solution optimale ([2], [3]), soit des solutions approchées ([15]).

2) le problème d'affectation quadratique : étant donné deux matrices : la matrice "coût" $C = (c_{ij})$ et la matrice "distances" $D = (d_{k\ell})$ à dimensions $n \times n$, trouver la permutation p sur $\{1,2,\ldots,n\}$ telle que
$\sum_{i,j} c_{ij}\, d_{p(i)p(j)}$ soit minimum.

En pratique, ce problème est résolu de façon approchée, car pour trouver la solution optimale, le temps de calcul croît très vite avec n.

Dans le problème d'implantation, il s'agit de placer des groupes de modules reliés entre eux. Le problème d'affectation quadratique concerne uniquement des paires d'éléments et peut être considéré comme un cas particulier du problème d'implantation.

Le problème d'affectation simple est un cas particulier du problème d'affectation quadratique (éléments indépendants).

Le problème d'implantation est fréquemment résolu comme un problème d'affectation quadratique.

I.1. NOTATIONS ET DEFINITIONS

Soit un réseau $\mathcal{R} = (M, \mathcal{E})$ constitué par :

- $M = \{m_1, m_2, \ldots, m_n\}$ appelé ensemble de modules
- $\mathcal{E}$ une famille de parties de M : $\mathcal{E} = \{E_i\}_{i \in \mathbb{N}}$ appelée ensemble des équipotentielles.

S'il existe un $E_i \in \mathcal{E}$, $E_i = \{m_{i_1}, m_{i_2}, \ldots, m_{i_p}\}$, toute paire (m_j, m_k) telle que m_j et m_k appartiennent à E_i, est appelée liaison ou connexion, et notée ℓ_{ij}.

La longueur d'une liaison ℓ_{ij} est la distance entre le module m_i et le module m_j. On pourra utiliser soit la distance euclidienne : $d(m_i, m_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, soit la distance orthogonale : $d'(m_i, m_j) = |x_i - x_j| + |y_i - y_j|$, où (x_i, y_i), (x_j, y_j) sont respectivement les coordonnées des emplacements des modules m_i, m_j.

Soit $E_i = \{m_{i_1}, m_{i_2}, \ldots, m_{i_p}\}$ une équipotentielle. On considère le graphe complet G_{E_i} engendré par l'ensemble de sommets E_i, appelé aussi "équipotentielle complète".

Un arbre maximal de G_{E_i} sera appelé arbre de réalisation de l'équipotentielle E_i (ou réalisation de E_i).

Pour un arbre de réalisation donné T, on appellera longueur de la réalisation, la somme des longueurs des liaisons qui apparaissent dans T. Par abus de langage, nous dirons aussi "longueur de l'équipotentielle".

La construction de l'équipotentielle de longueur minimale revient donc à la recherche d'un arbre de longueur minimale. Si les emplacements des modules sont fixés, la construction d'un arbre minimal est connue ([4], [5], [6]).

Pour implanter des modules tels que la somme des longueurs des équipotentielles soit minimale, les modules pouvant changer de place, il faut tenir compte de toutes les possibilités dans la construction de toutes les réalisations d'équipotentielles.

En considérant toutes les équipotentielles complètes, la matrice $n \times n$ des liaisons $\mathcal{L} = (p_{ij})$ est définie par :

- $p_{ii} = 0 \quad \forall\, i \in \{1,2,\ldots,n\}$
- $p_{ij} = p_{ji} \quad \forall\, i,j \in \{1,\ldots,n\}$
- p_{ij} est le nombre des liaisons du module m_i au module m_j.

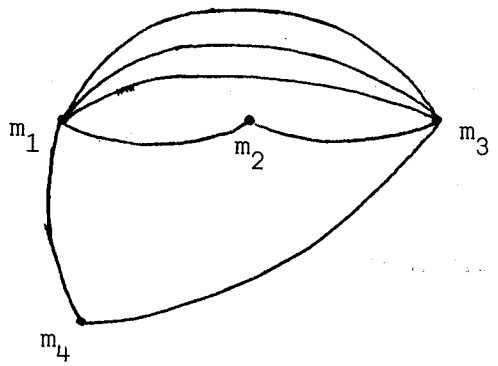
On associe à $\mathcal{R}$ le graphe des équipotentielles complètes, dont les sommets sont les modules de M et les arêtes sont les liaisons des équipotentielles complètes.

EXEMPLE

Soit $\mathcal{R} = (M, \mathcal{E})$ où :

$M = \{m_1, m_2, m_3, m_4\}$

$\mathcal{E} = (\{m_1, m_3\}, \{m_1, m_2, m_3\}, \{m_1, m_3, m_4\})$.



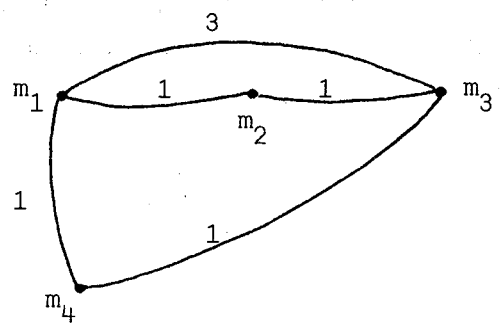
Graphe à équipotentielles complètes

$$\mathcal{L} = \begin{array}{c} \\ m_1 \\ m_2 \\ m_3 \\ m_4 \end{array} \begin{array}{c} \begin{array}{cccc} m_1 & m_2 & m_3 & m_4 \end{array} \\ \begin{bmatrix} 0 & 1 & 3 & 1 \\ 1 & 0 & 1 & 0 \\ 3 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

Matrice des liaisons

Pour simplifier, on représentera le graphe des équipotentielles complètes par le graphe pondéré, correspondant à $\mathcal{L}$, tel que :

- m_i est relié à m_j si $p_{ij} \neq 0$ dans
- (m_i, m_j) porte le "poids" p_{ij}.



Soit {1,2,...,q} l'ensemble des emplacements des modules sur la plaquette, où $q \geq n$. Pour simplifier les notations, on supposera dans la suite $q = n$ (si $q > n$ on peut introduire $q-n$ modules tels que le poids de leurs liaisons soit nul).

Les coordonnées des emplacements disponibles sur la plaquette sont connues et fixées : $(x_i, y_i)_i$.

On note par d_{ij} la distance de l'emplacement i à l'emplacement j. ($d_{ij} = \sqrt{(x_i-x_j)^2 + (x_j-y_i)^2}$ ou $d_{ij} = |x_i-x_j| + |y_i-y_j|$).

Soit p la permutation sur $\{m_1,\ldots,m_n\}$ telle que p(i) soit l'emplacement occupé par le module m_i.

Le problème d'implantation des modules en vue de minimiser la longueur totale des liaisons se ramène au problème d'affectation quadratique : trouver la permutation p telle que :

$$\sum_{i<j} p_{ij}\, d_{p(i)p(j)} \quad \text{soit minimum.}$$

L'énumération, pour n assez grand, de toutes les permutations n'est évidemment pas possible.

I.2. RAPPEL DE CERTAINES METHODES

De nombreuses approches du problème d'implantation ainsi posé ont été faites :

Steinberg [7] réduit le problème d'affectation quadratique associé au problème d'affectation simple en déterminant les sous-réseaux déconnectés maximaux du réseau initial. En enlevant un tel sous-réseau de la plaquette, il le replace de façon à ce que la longueur totale des liaisons diminue.

Rutman [8] améliore l'algorithme de Steinberg en effectuant de plus des changements de places entre modules interconnectés. La méthode dépend du nombre des sous-réseaux déconnectés maximaux et de leur taille, donc n doit être généralement assez grand ($n > 50$). De plus, elle demande beaucoup de temps de calcul : pour des grandes plaquettes (jusqu'à 100 modules), il faut plus de 2 heures de calcul sur un IBM 7090.

Gilmore [9] et Lawler [10] donnent une solution optimale du problème quadratique si n n'est pas trop grand ($n \leq 15$). De manière semi-énumérative, leur algorithme construit progressivement des implantations et compare les coûts à celui d'une implantation initiale. Gilmore [9] donne aussi deux algorithmes heuristiques pour lesquels le nombre d'opérations est de l'ordre de n^4 et n^5, donc assez grand.

Par un autre algorithme heuristique, Mamelak [11] utilise une méthode basée sur l'existence de groupes de modules, appelés "chaînes". A chaque "chaîne" est affectée une ligne d'emplacements sur la plaquette ; la minimisation du nombre des croisements est réalisée en mettant un module "significatif" à chaque bout de la "chaîne", et la minimisation des longueurs est obtenue par changement de places des modules. L'algorithme est rapide (quelques minutes de calcul pour $n \leq 50$), mais il est particulier aux réseaux du type "chaîne".

Garside et Nicholson [12] améliorent une implantation initiale en considérant tous les sous-ensembles de k modules, où k = 2,3,..., et en effectuant sur chacun toutes les k! permutations. Ils arrivent aux conclusions suivantes :

- l'algorithme est rapide
- il donne de bons résultats par rapport à la méthode stochastique de Monte-Carlo
- la meilleure amélioration de l'implantation initiale est obtenue pour k = 2 ; elle devient faible pour k > 2.

Gavett [13] réduit, aussi, le problème d'affectation quadratique au problème d'affectation simple, en construisant la matrice C des coûts associés à toute implantation de chaque paire de modules sur la plaquette.

EXEMPLE

Soit {A,B,C,D} l'ensemble des modules et soit {1,2,3,4} l'ensemble de places disponibles sur la plaquette ; alors la matrice C est :

	(1,2)	(1,3)	(1,4)	(2,3)	(2,4)	(3,4)
(A,B)	c_{11}	c_{12}	. . . . .			c_{16}
(A,C)	c_{21}	c_{22}	. . . . .			c_{26}
(A,D)	.					.
(B,C)	.					.
(B,D)	.					.
(C,D)	c_{61}	c_{62}	. . . . .			c_{66}

Toutes les solutions du problème d'affectation simple ne sont pas des solutions valables pour le problème d'implantation (2,3,... modules peuvent être à la même place). Pour éliminer ces cas d'impossibilité, l'algorithme devient lourd. De plus, la matrice des coûts étant de dimension $C_n^2 \times C_n^2$, on est limité à des n assez petits (n < 15).

I.3. CONCLUSION

Il est difficile d'obtenir une solution optimale de l'implantation de modules, en minimisant les longueurs, surtout si le nombre de modules n est grand, car :

- le nombre de calculs croît exponentiellement avec n, donc la recherche d'une telle solution devient très coûteuse
- même si la solution obtenue pour le problème quadratique associé est optimale, elle n'est pas nécessairement optimale pour le problème d'implantation ([1]).

Pour la résolution du problème d'implantation, nous donnerons des algorithmes heuristiques, pour les deux raisons suivantes :

- ils sont rapides, donc moins chers
- la solution obtenue est assez "proche" de la solution optimale ; en effet, ce type de problèmes n'admet pas de minimum aigu ([14]).

Hanan et Kurtzberg ([12]) montrent statistiquement que de bons résultats au problème d'implantation sont obtenus rapidement par des algorithmes qui construisent d'abord une implantation initiale, puis améliorent cette implantation initiale.

Nous proposons maintenant une méthode de ce type pour implanter des modules.

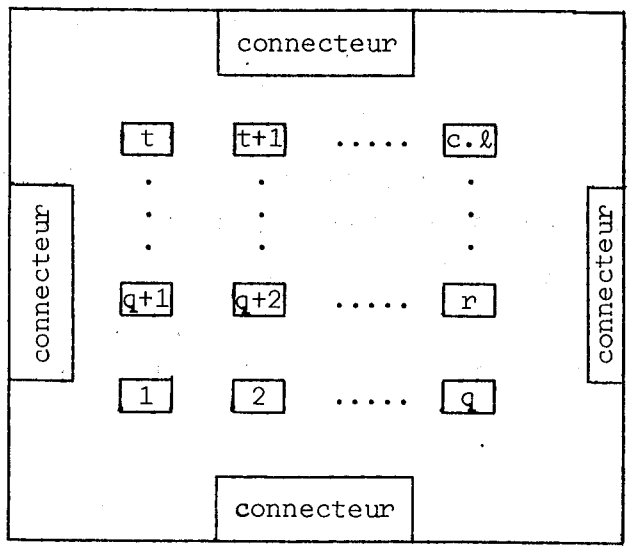
I.4. ALGORITHMES D'IMPLANTATION

I.4.0. Définitions, notations

Soit E la matrice ($\ell \times c$) des emplacements où peuvent se placer les modules sur la plaquette.

On note par : 1, 2, ..., c.ℓ les emplacements de la plaquette ; généralement c.ℓ = n, où n est le nombre de modules, ce que l'on supposera à la suite pour simplifier les notations.

Soit p $\in$ {1,2,3,4} le nombre de connecteurs (ils assurent les liaisons de la plaquette avec l'extérieur). Leurs emplacements sont fixés au bord de la plaquette.



Un module ayant r liaisons avec les autres est dit de degré r.

Soit $\mathcal{L} = (p_{ij})$ la matrice des liaisons, de dimension (n+p) × (n+p) associée au réseau d'équipotentielles complètes.

Soit $\mathcal{D} = (d_{ij})$ la matrice de "distances" de même dimension que $\mathcal{L}$, telle que d_{ij} soit la distance (euclidienne, orthogonale) entre les centres des emplacements i et j ($\mathcal{D}$ est symétrique et $d_{ii} = 0$).

I.4.1. Détermination d'une implantation initiale

Une méthode de "dispersion barycentrique" plaçant un module "libre" au barycentre des emplacements occupés par des modules qui lui sont reliés, permet d'obtenir rapidement une implantation initiale ([16]). Mais ces méthodes supposent qu'il existe des modules possédant des emplacements fixes, connus à l'avance, ce qui n'est pas toujours le cas.

PROPRIETE

Soir r l'emplacement (ou un des emplacements) du centre de la plaquette et soit u,v deux emplacements tels que $d_{ru} \leq d_{rv}$.

Alors : $\sum_{k=1}^{n+p} d_{uk} \leq \sum_{k=1}^{n+p} d_{vk}$.

I.4.1.1. Description de l'algorithme

Nous utilisons la propriété précédente pour obtenir une implantation initiale en ajoutant un à un les modules sur la plaquette.

On choisit le premier module m_{i_1} qui a le plus grand nombre de liaisons avec les connecteurs et s'il y a plusieurs modules dans ce cas on choisit un de degré maximum. Si $p \neq 1$, m_{i_1} est placé au barycentre des connecteurs. Si $p = 1$ il est placé en fonction du nombre de ses liaisons avec les autres modules : s'il est relié à tous les autres modules, il sera placé vers le centre de la plaquette, sinon il sera placé plus près du connecteur.

On choisit ensuite un module ayant le plus grand nombre de liaisons avec les modules déjà placés et s'il y a plusieurs, on prend encore un de degré maximum. Il est mis à un emplacement libre de façon à ce que la somme des longueurs de ses liaisons avec les modules placés soit minimale ; s'il existe plusieurs possibilités, on choisit un emplacement tel qu'il soit le plus près possible du centre de l'ensemble des

emplacements encore libres. On continue ainsi jusqu'à l'épuisement de tous les modules.

1.4.1.2. Algorithme

Prenons d'abord p = 1, ce qui est fréquent en pratique. Le connecteur sera le $(n+1)^{ième}$ "module", donc représenté par la $(n+1)^{ième}$ ligne de la matrice $\mathcal{L} = (p_{ij})$.

On note par :

- $L = \{L_1, L_2, \ldots, L_{n+1}\}$ l'ensemble des lignes de $\mathcal{L}$.
- $C = \{C_1, C_2, \ldots, C_{n+1}\}$ l'ensemble des colonnes de $\mathcal{L}$.
- $\ell_1, \ell_2, \ldots, \ell_\ell$ les lignes de E, tel que ℓ_1 se trouve le plus près du connecteur. c est le nombre de colonnes de E.

1) Détermination du module m_{i_1} : trouver i_1 tel que :

$$p_{i_1,n+1} = \max_{1 \leq j \leq n} (p_{j,n+1})$$

$$\sum_{k=1}^{n} p_{i_1,k} = \max_{1 \leq j \leq n} \left(\sum_{k=1}^{n} p_{j,k} \right)$$

Soient $P = \{L_{i_1}\} \cup \{L_{n+1}\}$, $Q = \{C_{i_1}\} \cup \{C_{n+1}\}$

2) Implantation de m_{i_1} en fonction du nombre de modules auxquels il est relié :

Calculer S le nombre d'éléments non nuls de L_{i_1}.

- si $S < 2.c$ alors placer le module m_{i_1} dans la ligne ℓ_1 de E, à l'emplacement j_1 tel que la somme de ses distances aux autres emplacements de la ligne ℓ_1 soit minimum :

$$\sum_{k \in \ell_1} d_{j_1,k} = \min_{j \in \ell_1} \left(\sum_{k \in \ell_1} d_{jk} \right)$$

- si $2.c \leq S < 4.c$ alors placer le module m_{i_1} dans la ligne ℓ_2 de E, à l'emplacement j_1, tel que :

$$\sum_{k \in \ell_1 \cup \ell_2} d_{j_1 k} = \min_{j \in \ell_1 \cup \ell_2} \left(\sum_{k \in \ell_1 \cup \ell_2} d_{jk} \right)$$

- si $(\ell-2).c \leq S < \ell.c$ alors placer le module m_{i_1} dans la ligne $\ell_{En(\ell/2)}$ de E (où En(ℓ/2) est la partie entière de ℓ/2) à l'emplacement j_1 tel que :

$$\sum_{k \in \ell_1 \cup \ell_2 \cup \ldots \cup \ell_{\ell-2}} d_{j_1 k} = \min_{j,k \in \ell_1 \cup \ell_2 \cup \ldots \cup \ell_{\ell-2}} (\Sigma\, d_{jk})$$

3) Détermination du module m_{i_r} le plus relié aux modules déjà placés : trouver i_r tel que :

$$\sum_{k \in Q} p_{i_r k} = \max_{j \in L-P} \left(\sum_{k \in Q} p_{jk} \right)$$

$$\sum_{k \in C-Q} p_{i_r k} = \max_{j \in L-P} \left(\sum_{k \in C-Q} p_{jk} \right)$$

Faire $P = P \cup \{L_{i_r}\}$, $Q = Q \cup \{C_{i_r}\}$.

4) Implantation du module m_{i_r} à l'emplacement j_r tel que :

- la somme des longueurs des liaisons de m_{i_r} avec les modules déjà placés soit minimale :

$$\sum_{k \in Q} p_{i_r k} \cdot d_{j_r j_k} = \min_{q>r} \left(\sum_{k \in Q} p_{i_r k} \cdot d_{j_q j_k} \right)$$

• la somme des distances aux emplacements libres soit minimale :

$$\sum_{q>r} d_{j_r j_q} = \min_{t>r} \left(\sum_{u>r} d_{j_t j_u} \right)$$

5) Reprendre successivement 3) et 4) tant que P ≠ L.

REMARQUES

1. Aux pas 1), 3) et 4), la deuxième condition est appliquée seulement s'il existe plusieurs modules qui vérifient la première condition.

2. Si p > 1, nous commençons l'algorithme en 3) avec P contenant l'ensemble des lignes correspondantes aux connecteurs et Q contenant l'ensemble des colonnes correspondantes aux connecteurs.

Le concepteur a la possibilité de fixer l'emplacement de certains modules sur la plaquette. Dans ce cas, l'algorithme commence aussi en 3), P,Q contenant de plus les lignes et les colonnes correspondantes à ces modules.

I.4.2. Amélioration de l'implantation initiale

Nous cherchons à diminuer la longueur totale des liaisons entre modules par une méthode dite de "changements des paires", méthode qui donne rapidement de bons résultats ([1], [19]).

Le principe est de considérer successivement chaque module m_i, $i = 1,2,\ldots,n$ et d'essayer d'échanger sa place avec celle du module m_j, $j = 1,2,\ldots,n$, et $j \neq i$, l'échange étant effectivement réalisé si l'on constate une réduction de la longueur totale des fils de liaison.

Divers algorithmes basés sur le "changement des paires" ([1], [12]) cherchent à réduire le nombre de calculs.

Nous donnons une procédure qui par des calculs simples nous donne rapidement une solution.

I.4.2.1. Principes de calcul

L'algorithme est basé sur les 3 remarques suivantes :

REMARQUE 1

Soient $\{m_i, m_j\}$ une paire de modules pour lesquels on envisage un échange de places, $\mathcal{Y}_i$, $\mathcal{Y}_j$ la somme des longueurs des liaisons de m_i et m_j avec tous les autres modules avant l'échange et $\mathcal{Y}'_i$, $\mathcal{Y}'_j$ les quantités correspondantes après l'échange.

Comme la longueur totale des liaisons avant l'échange est $\mathcal{Y}_i + \mathcal{Y}_j + \mathcal{Y}_r$ (à un terme près : la longueur des liaisons communes à m_i, m_j) et après l'échange devient $\mathcal{Y}'_i + \mathcal{Y}'_j + \mathcal{Y}_r$ (au même terme près), où $\mathcal{Y}_r$ est la longueur des liaisons des modules sauf $\{m_i, m_j\}$, il suffit d'avoir : $\mathcal{Y}'_i + \mathcal{Y}'_j < \mathcal{Y}_i + \mathcal{Y}_j$ pour que l'échange soit effectivement fait.

REMARQUE 2

Soient :

- $\mathcal{L}$ la matrice des liaisons, dont les lignes sont : $L_1, L_2, \ldots, L_{n+p}$
- $\mathcal{D}$ la matrice des distances, dont les lignes sont : $D_1, D_2, \ldots, D_{n+p}$
- f une application bijective de $\{L_1, L_2, \ldots, L_{n+p}\}$ dans $\{D_1, D_2, \ldots, D_{n+p}\}$ telle que :

  pour tout $i \in \{1,2,\ldots,n+p\}$ $f(L_i) = D_i$, c'est-à-dire si le module m_i correspond à la $i^{ème}$ ligne de $\mathcal{L}$, alors son emplacement correspond aussi à la $i^{ème}$ ligne de $\mathcal{D}$.

Dans ce cas, le calcul de $\mathcal{S}_i + \mathcal{S}_j$ et de $\mathcal{S}'_i + \mathcal{S}'_j$ se réduit au calcul de produits scalaires de vecteurs lignes de $\mathcal{L}$ et $\mathcal{D}$ à un terme près.

En effet :

$$\mathcal{S}_i + \mathcal{S}_j = p_{i1}d_{i1} + \dots + p_{i,n+p}d_{i,n+p} + p_{j1}d_{j1} + \dots + p_{j,n+p}d_{j,n+p} -$$

$$- p_{ij}d_{ij} = \langle L_i, D_i \rangle + \langle L_j, D_j \rangle - p_{ij}d_{ij} .$$

$$\mathcal{S}'_i + \mathcal{S}'_j = p_{i1}d_{j1} + \dots + p_{ij}d_{ji} + \dots + p_{i,n+p}d_{j,n+p} +$$

$$+ p_{j1}d_{i1} + \dots + p_{ji}d_{ij} + \dots + p_{j,n+p}d_{i,n+p} - p_{ij}d_{ij} .$$

Mais :

$$p_{ij}d_{ji} = p_{ij}d_{ij} = p_{ij}d_{jj} + p_{ij}d_{ij}, \quad (d_{jj} = 0)$$

$$p_{ji}d_{ij} = p_{ij}d_{ij} = p_{ji}d_{ii} + p_{ij}d_{ij}, \quad \text{donc :}$$

$$\mathcal{S}'_i + \mathcal{S}'_j = \langle L_i, D_j \rangle + \langle L_j, D_i \rangle + p_{ij}d_{ij} .$$

Après chaque échange effectivement réalisé, nous cherchons à réordonner la matrice $\mathcal{D}$

afin que les calculs des longueurs puissent être effectués avec des produits scalaires.

Nous aurions :

$$\mathcal{D}' = \mathcal{P}^{-1}\mathcal{D}\mathcal{P} ,$$

où $\mathcal{P}$ est une matrice de permutation sur l'ensemble des emplacements. Mais comme nous ne considérons que des transpositions, nous éviterons de faire des produits de matrices et de calculer une matrice inverse ; on obtient $\mathcal{D}'$ à partir de $\mathcal{D}$ par l'échange de la $i^{ème}$ ligne avec la $j^{ème}$ et de la $i^{ème}$ colonne avec la $j^{ème}$ (les 4 éléments aux intersections des lignes et des colonnes restent en place, car $d_{ii} = d_{jj} = 0$ et $d_{ij} = d_{ji}$).

REMARQUE 3

Pour ne pas calculer deux fois $\mathcal{P}_i + \mathcal{P}_j$, $\mathcal{P}'_i + \mathcal{P}'_j$ (ce qui arrive si les modules ne changent pas de place) on simplifiera comme ci-dessous :

- on ordonne l'ensemble de modules M par ordre croissant des indices

$$M = \{m_1, m_2, \ldots, m_n\}.$$

- on considère l'application g de M dans {1,2,...,n} telle que :

$$g(m_i) = \begin{cases} i \text{ si le module } m_i \text{ n'a pas changé de place} \\ j \text{ si on a échangé la place du module } m_i \text{ avec celle de } m_j . \end{cases}$$

- les tests de changement de places débutent avec le module de plus petit indice.

  Soit le module m_j et supposons que l'on ait testé le changement possible de places avec des modules $m_1, m_2, \ldots, m_{j-1}$:

- si $g(m_i) \leq j$ cela veut dire que le calcul de $\mathcal{P}_i + \mathcal{P}_j$ et de $\mathcal{P}'_i + \mathcal{P}'_j$ pour la paire $\{m_i, m_j\}$ a déjà été fait en étudiant le module m_i, $i < j$ et que l'on n'échangera plus m_i et m_j.

- si $g(m_i) > j$ le calcul des longueurs en vue de l'échange des emplacements de la paire $\{m_i, m_j\}$ a été fait au moment de l'étude du module m_i, mais m_i a été changé de place avec m_k, où $k > j$.

En définitive, si $i < j$ on n'essaiera un échange de places entre m_i et m_j que si $g(m_i) > j$.

I.4.2.2. Algorithme

On note par F l'ensemble des modules dont les emplacements sont fixes et connus (F peut être vide).

1) Ordonner $M = \{m_i \mid i=1,2,\ldots,n\}$ par ordre croissant des indices et faire pour $i=1,2,\ldots,n$ $g(m_i) = i$. $\mathcal{L}$ étant donnée, changer la matrice $\mathcal{D}$ telle que $f(L_i) = D_i$, où L_i, D_i sont les lignes de $\mathcal{L}$, $\mathcal{D}$ pour $i=1,2,\ldots,n$.

2) $j = 0$.

3) $J = J+1$. Considérer $m_j \in M-F$ et tous les $m_i \in M-F$, avec $m_i \neq m_j$.

- Si $(i > j)$ ou $(g(m_i) > j)$, calculer :

$$\mathcal{Y}_i + \mathcal{Y}_j = \langle L_i, D_i \rangle + \langle L_j, D_j \rangle - p_{ij}d_{ij}$$

$$\mathcal{Y}'_i + \mathcal{Y}'_j = \langle L_i, D_j \rangle + \langle L_j, D_i \rangle + p_{ij}d_{ij} .$$

- Sinon, aller à 5).

4) Si $\mathcal{Y}'_i + \mathcal{Y}'_j < \mathcal{Y}_i + \mathcal{Y}_j$ alors échanger les emplacements de m_i et m_j, échanger les lignes i,j et les colonnes i,j de $\mathcal{D}$ et faire $g(m_j) = i$.

5) Si $j < n$, alors aller à 3). S'il y a au moins un échange effectué, aller à 2) sinon fin de l'algorithme.

I.4.2.3. Estimation du nombre de calculs

En négligeant le pas 1) qui n'est pas inclus dans la boucle itérative et les quelques comparaisons, on a :

- $4(n+p)+2$ multiplications et $4(n+p-1)+2$ additions au pas 3), où n est le nombre de modules et p le nombre des connecteurs.

- $4(n+p)-4$ échanges au plus au pas 4) (4 éléments (remarque 3) restant sur place).

Approximativement on a : $n^2[4(n+p)+2+4(n+p-1)+2+4(n+p)-4] =$
$= 12\, n^2(n+p-4)$ "opérations", où "opération" désigne multiplication, addition, échange des valeurs de 2 nombres (pour un pas d'itération).

L'ordre de grandeur du nombre d'opérations est en n^3 ce qui montre que l'algorithme est très rapide.

REMARQUES

1) Si certaines liaisons doivent être les plus courtes possibles (par exemple des liaisons de haute tension ou des liaisons de masse) on peut ajouter des poids supplémentaires aux liaisons correspondantes pour que leur poids devienne plus grand que celui des autres liaisons.

2) Les emplacements de certains modules peuvent aussi être partiellement fixés par rapport à un élément fixe : connecteur ou tout autre module. On procède de la manière suivante :

- soit $H \subset E$ l'ensemble des emplacements dans lequel m_i doit être placé. On diminue les distances, dans la matrice $\mathcal{D}$ de l'emplacement de l'élément fixe aux emplacements de H à $d_{min} < \min_{k,\ell} (d_{k\ell})$
- on augmente, dans la matrice $\mathcal{L}$, le poids de la liaison de m_i avec l'élément fixe à $p_{max} > \max_{i,j} (p_{ij})$.

Par le "changement des paires", m_i trouvera évidemment sa place dans H, car les modules très reliés (grand poids de liaison) cherchent à occuper des emplacements aux plus courtes distances.

I.4.2.4. Exemple

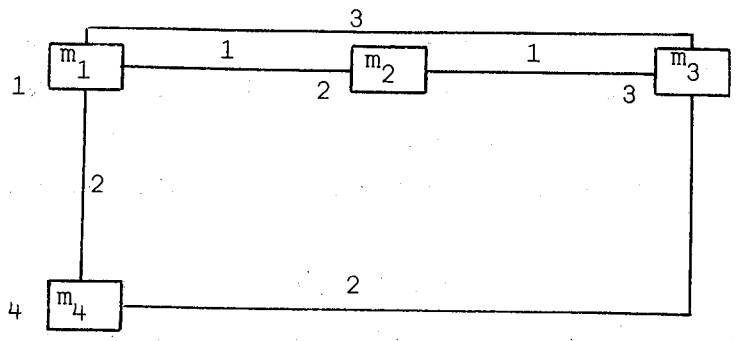
Soit $M = \{m_1, m_2, m_3, m_4\}$ avec les équipotentielles :

$$E_1 = \{m_1, m_2, m_3\},$$
$$E_2 = \{m_1, m_3, m_4\},$$
$$E_3 = \{m_1, m_3, m_4\}.$$

La matrice des liaisons est :

$$\mathcal{L} = \begin{array}{c} \\ m_1 \\ m_2 \\ m_3 \\ m_4 \end{array} \begin{array}{c} \begin{array}{cccc} m_1 & m_2 & m_3 & m_4 \end{array} \\ \begin{bmatrix} 0 & 1 & 3 & 2 \\ 1 & 0 & 1 & 0 \\ 3 & 1 & 0 & 2 \\ 2 & 0 & 2 & 0 \end{bmatrix} \end{array}$$

Soit l'implantation initiale figurée ci-dessous :



La distance entre 2 modules voisins (sur l'horizontale et la verticale) est l'unité de distance.

En considérant la distance orthogonale, la somme des longueurs des liaisons est 16 et la matrice des distances $\mathcal{D}$ est :

$$\mathcal{D} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 1 & 2 & 1 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 3 \\ 1 & 2 & 3 & 0 \end{bmatrix} \end{array}$$

Initialement on a : $g(m_1) = 1$, $g(m_2) = 2$, $g(m_3) = 3$, $g(m_4) = 4$.
On commence par le module m_1.

a) $\mathcal{S}_1 + \mathcal{S}_2 = \langle L_1, D_1 \rangle + \langle L_2, D_2 \rangle - p_{12} d_{12} = 9 + 2 - 1 = 10$

$\mathcal{S}'_1 + \mathcal{S}'_2 = \langle L_1, D_2 \rangle + \langle L_2, D_1 \rangle + p_{12} d_{12} = 7 + 2 + 1 = 10.$

Les emplacements de m_1 et m_2 ne sont pas échangés.

b) $\mathcal{S}_1 + \mathcal{S}_3 = \langle L_1, D_1 \rangle + \langle L_3, D_3 \rangle - p_{13} d_{13} = 16$

$\mathcal{S}'_1 + \mathcal{S}'_3 = \langle L_1, D_3 \rangle + \langle L_3, D_1 \rangle + p_{13} d_{13} = 16.$

Les emplacements de m_1 et m_3 ne sont pas échangés.

c) $\mathcal{Y}_1 + \mathcal{Y}_4 = 15$, $\mathcal{Y}'_1 + \mathcal{Y}'_4 = 17$, donc l'implantation reste inchangée.

d) On continue avec le module m_2 et comme $g(m_1) = 1 < 2$ on ne teste plus $\{m_2, m_1\}$.

$\mathcal{Y}_2 + \mathcal{Y}_3 = 14$, $\mathcal{Y}'_2 + \mathcal{Y}'_3 = 10$, donc les emplacements de m_2 et m_3 sont échangés.

$g(m_2) = 3$ et la matrice des distances devient :

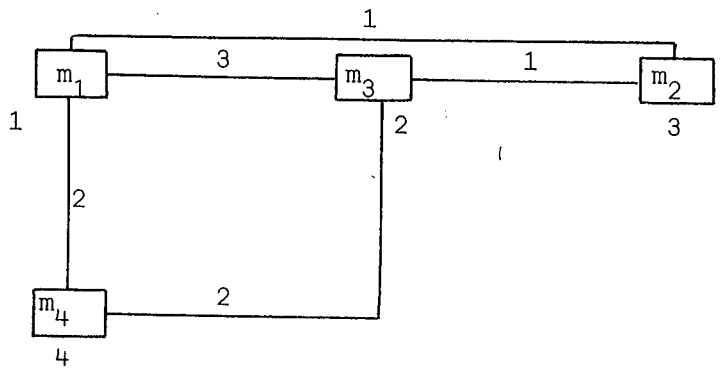
$$\mathcal{D} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 2 & 1 & 1 \\ 2 & 0 & 1 & 3 \\ 1 & 1 & 0 & 2 \\ 1 & 3 & 2 & 0 \end{bmatrix} \end{array}$$

e) $\mathcal{Y}_2 + \mathcal{Y}_4 = 9$, $\mathcal{Y}'_2 + \mathcal{Y}'_4 = 9$, donc les emplacements de m_2 et m_4 ne sont pas échangés.

f) On continue avec le module m_3. Comme $g(m_1) = 1 < 3$ et $g(m_2) = 3$ on ne teste plus $\{m_3, m_1\}$ et $\{m_3, m_2\}$.

$\mathcal{Y}_3 + \mathcal{Y}_4 = 10$, $\mathcal{Y}'_3 + \mathcal{Y}'_4 = 12$, donc les emplacements de m_3 et m_4 ne sont pas échangés.

Le schéma final est :



la somme des longueurs étant 12.

CHAPITRE II

IMPLANTATION DE MODULES PAR SOUS-RESEAUX MINIMAUX

Cette méthode s'applique pour des réseaux à un nombre élevé de modules ($n > 50$). Dans ce cas, les modules forment des groupements à l'intérieur du réseau (chaque groupement réalisant une certaine fonction logique), tels que les liaisons sont nombreuses à l'intérieur d'un groupe et peu nombreuses entre groupes distincts.

La méthode de "changements de paires" perd de son efficacité si elle est appliquée à l'ensemble de tous les modules du réseau ; il y a peu de chances qu'un modules appartenant à un groupement change sa place avec celle d'un module d'un autre groupement.

Un algorithme plus efficace, que nous donnons à la suite, décompose le réseau initial en sous-réseaux tels que le nombre de liaisons entre sous-réseaux soit faible, place les sous-réseaux sur la plaquette en affectant des emplacements à chacun des modules d'un sous-réseau, et applique ensuite à l'intérieur de chaque sous-réseau l'algorithme de "changement de paires" (amélioration de l'implantation initiale).

Nous réduisons ainsi le problème d'implantation de n modules à s problèmes d'implantation de n_i modules, $i=1,2,\ldots,s$, tels que $n_i < n$ pour tout i.

$$\left(\sum_{i=1}^{s} n_i = n \right).$$

II.1. DEFINITIONS ET PROPRIETES

Soit un réseau $\mathcal{R} = \{M, \mathcal{E}\}$, où $M = \{m_1, m_2, \ldots, m_n, m_{n+1}, \ldots, m_{n+p}\}$.

Un sous-réseau de $\mathcal{R}$ est constitué d'un ensemble de modules $M_1 \subset M$ et de toutes les liaisons d'équipotentielles complètes, ayant au moins une extrémité dans M_1.

Une liaison externe du sous-réseau G relie un module de G à un module n'appartenant pas à G.

Un sous-réseau G est minimal ([17]) si pour tout sous-réseau $H \subset G$ on a $h > g$, où h et g sont les nombres de liaisons externes respectivement des sous-réseaux H et G.

REMARQUES

1) Tout sous-réseau constitué par un seul module de M est un sous-réseau minimal.

2) Si M n'est pas réduit à un seul module, alors il existe toujours des sous-réseaux minimaux S_i de $\mathcal{R}$ tels que $S_i \neq M$.

   En effet, il existe au moins les $S_i = \{m_i\} \neq M$.

THEOREME 1 [17]

Soient K,L deux sous-réseaux minimaux. Si K n'est pas inclus dans L et L n'est pas inclus dans K, alors $K \cap L = \emptyset$.

Donc, si des sous-réseaux minimaux se recouvrent, ils sont nécessairement emboîtés.

II.2. ETAPES DU DEROULEMENT DE L'ALGORITHME

1 - Détermination des sous-réseaux minimaux : nous appliquons un algorithme simple et rapide, élaboré par F. Luccio et M. Sami ([17]), qui trouve l'ensemble de tous les sous-réseaux minimaux noté S.

2 - Extraction dans l'ensemble S d'un sous-ensemble $S' = \{S'_1,\ldots,S'_q\}$ tel que S' soit un recouvrement disjoint de M.

S' existe bien, d'après ce qui précède.

REMARQUE

On a intérêt à réaliser le recouvrement de M par un nombre réduit de sous-réseaux minimaux : de 2 à 5, en pratique. Dans le cas limite où Card(S') = n, c'est-à-dire où chaque sous-réseau minimal de S' ne contient qu'un seul module, le problème revient à l'implantation de modules séparés. Par contre, si le nombre de sous-réseaux minimaux est réduit, on a plus de possibilités d'améliorer l'implantation initiale des sous-réseaux.

En pratique, nous cherchons à couvrir M par des sous-réseaux minimaux comportant un nombre de modules ne dépassant pas n/2 ; autrement on peut avoir un sous-réseau minimal de très grande taille (proche de M) et le problème d'implantation n'est pas simplifié.

3 - Implantation des sous-réseaux minimaux sur la plaquette.

Chaque sous-réseau est considéré comme un seul module dont les liaisons sont les liaisons externes du sous-réseau. Pour implanter les sous-réseaux ainsi considérés, nous utilisons un algorithme proche de l'algorithme décrit auparavant pour déterminer une implantation initiale (I.3.1.). Il se décompose en deux étapes :

a) Choix d'un sous-réseau à placer.

Initialement, on choisit le sous-réseau qui contient le plus d'éléments fixes : connecteurs et modules (s'il en existe). Puis, à chaque itération on choisira le sous-réseau qui a le plus de liaisons externes avec les sous-réseaux déjà placés ; pour ce choix on utilisera la matrice T des liaisons externes, construite à partir de la matrice $\mathcal{L}$ des liaisons :

$$\begin{cases} T_{r,t} = \sum_{m_i \in S'_r} \sum_{m_j \in S'_t} \mathcal{L}(m_i, m_j) \ , \quad r,t < q \text{ et } r \neq t \\ T_{r,r} = 0 \ , \quad T_{r,t} = T_{t,r} \end{cases}$$

b) Implantation des modules d'un sous-réseau minimal

Une fois choisi un sous-réseau, on place tous ses modules de manière à ce que les modules les plus reliés occupent des emplacements voisins, comme dans l'algorithme d'implantation initiale. Mais, comme la plaquette est remplie par groupements de modules, on choisit parmi les modules les plus reliés aux modules déjà placés celui dont le nombre de liaisons avec les autres modules est le plus petit.

4 - Amélioration de l'implantation initiale dans chaque sous-réseau minimal.

Une amélioration peut évidemment être faite en énumérant pour chaque sous-réseau les $n_i!$ implantations possibles. Mais, la solution optimale pour l'implantation de chaque sous-réseau ne donne pas nécessairement une solution optimale pour l'implantation du réseau global (pour améliorer l'implantation initiale de S'_i, on suppose que les modules n'appartenant pas à S'_i sont fixés).

D'autre part, si n_i est assez grand ($n_i > 15$) l'énumération de toutes les $n_i!$ implantations n'est pas possible.

Nous préférons utiliser un algorithme donnant une solution approchée mais rapide, dont le principe est lié au "changement de paires".

Dans l'algorithme de changement de paires appliqué pour chaque sous-réseau minimal, nous utilisons les matrices : des liaisons $\mathcal{L}$ et des distances $\mathcal{D}$, définies comme auparavant (I.1 et I.3). Les calculs sont faits de la même manière qu'au chapitre précédent, c'est-à-dire par produits scalaires de vecteurs lignes de $\mathcal{L}$ et $\mathcal{D}$. L'algorithme de changement de paires est donc le même, seulement l'ensemble M est remplacé successivement par les sous-ensembles S'_1, S'_2, ..., S'_q.

Le processus est itéré tant qu'il y a au moins un échange d'emplacements de modules. La méthode est convergente, car à chaque échange effectivement réalisé, la longueur totale des liaisons diminue et elle est bornée inférieurement.

REMARQUE

Si le nombre de sous-réseaux minimaux est grand (card(S') $\geq$ n/2, donc il y a au plus 2 modules par sous-réseaux minimal), nous appliquerons l'algorithme d'amélioration par changement de paires à l'ensemble de tous les modules M, sans distinction de sous-réseaux minimaux.

II.3. ALGORITHME

Supposons $S = \{S_1, S_2, \ldots, S_m\}$ étant l'ensemble de tous les sous-réseaux minimaux de $M = \{m_1, m_2, \ldots, m_n, m_{n+1}, \ldots, m_{n+p}\}$, où n est le nombre de modules et p est le nombre de connecteurs ; S est déterminé par l'algorithme de F. Luccio et M. Sami (voir annexe 1).

On utilise les mêmes notations qu'auparavant :

- $L = \{L_1, L_2, \ldots, L_n, L_{n+1}, \ldots, L_{n+p}\}$ est l'ensemble de lignes de $\mathcal{L}$,
- $C = \{C_1, C_2, \ldots, C_n, C_{n+1}, \ldots, C_{n+p}\}$ est l'ensemble de colonnes de $\mathcal{L}$,
- $\{1, 2, \ldots, n\}$ est l'ensemble des emplacements de la plaquette, qui doivent recevoir les modules.

1) Détermination d'un recouvrement disjoint de M, par la famille S.

a) Construire la matrice $R = (r_{ij})$ à m lignes et n+p colonnes telle que :

$$r_{ij} = \begin{cases} 1 \text{ si } S_i \text{ contient le module } m_j \\ 0 \text{ sinon} \end{cases}$$

b) $j = 0$, $S' = \emptyset$, $M' = M$.

c) $j = j+1$. Considérer $m_j \in M'$ et trouver $S'_i \in S-S'$, $1 \leq i \leq m$, tel que :

- S'_i contient m_j : $r_{ij} = 1$
- $\text{Card}(S_i) \leq \frac{n+p}{2}$: $\sum_{k=1}^{n+p} r_{ik} \leq (n+p)/2$.
- S'_i est le plus grand sous-réseau satisfaisant les 2 conditions précédentes, c'est-à-dire si S_p contient m_j ($r_{pj} = 1$) et card $(S_p) \leq (n+p)/2$ alors :

$$\sum_{k=1}^{n+p} r_{pk} \leq \sum_{k=1}^{n+p} r_{ik}$$

Faire $S' = S' \cup S'_i$.

d) Faire $M' = M-S'_i$. Si $M' \neq \emptyset$ alors aller à c).

2) Implantation des sous-réseaux de S'

Soit $S' = \{S'_1, S'_2, \ldots, S'_q\}$ l'ensemble de sous-réseaux qui recouvrent M, $q \leq m$.

a) Construire la matrice des liaisons externe des sous-réseaux de S', $T = (t_{ij})$, à q lignes et q colonnes, telle que :

- $t_{ii} = 0$, $t_{ij} = t_{ji}$
- $t_{ij} = \sum_{k=1}^{n+p} \sum_{\ell=1}^{n+p} r_{ik} r_{i\ell} p_{k\ell}$, où $i < j$ et $i,j \in \{1,2,\ldots,q\}$.

b) Trouver le sous-réseau S'_{i_1}, $1 \leq i_1 \leq q$, qui contient le plus grand nombre d'éléments fixes (connecteurs et modules éventuels) :

- $Q = C' \cup \{C_{n+1}, C_{n+2}, \ldots, C_{n+p}\}$ où C' est le sous-ensemble de colonnes de C correspondant aux modules fixés à l'avance (C' peut être vide).
- $\sum_{k \in Q} r_{i_1 k} = \max_{1 \leq j \leq q} \left(\sum_{k \in Q} r_{jk} \right)$.

Faire $P = \{L^1_{i_1}, L^2_{i_2}, \ldots, L^u_{i_1}\}$, $U = \emptyset$, $V = 0$, où P est le sous-ensemble de lignes de $\mathfrak{L}$ correspondantes aux modules de S'_{i_1} (card $(S'_{i_1}) = u$).

2.1.) Implantation des modules de S'_{i_1}

a) Trouver $m_{i_r} \in S'_{i_1}$ tel que :

• m_{i_r} a le plus grand nombre de liaisons avec les modules déjà placés :

$$\sum_{k \in Q} p_{i_r k} = \max_{j \in P} \left(\sum_{k \in Q} p_{jk} \right).$$

• s'il y a plusieurs, choisir celui dont le degré est minimum :

$$\sum_{k \in C} p_{i_r k} = \min_{j \in P} \left(\sum_{k \in C} p_{jk} \right)$$

Faire $P = P - \{L_{i_r}\}$, $Q = Q \cup \{C_{i_r}\}$.

b) Placer m_{i_r} à l'emplacement j_r, tel que la somme des longueurs de m_{i_r} avec les modules déjà placés soit minimum :

$$\sum_{k \in Q} p_{i_r k} \, d_{j_r j_k} = \min_{q > r} \left(\sum_{k \in Q} p_{i_r k} \cdot d_{j_q j_k} \right).$$

c) Reprendre les pas a) et b) (de 2.1.) tant que $P \neq \emptyset$.

2.2.) Recherche d'un nouveau sous-réseau de S'

a) Faire V = V+1. Si V > card(S') alors aller à 3).

$S' = S' - \{S_{i_1}\}$, $U = U \cup \{S_{i_1}\}$.

Trouver $S'_{i_\ell} \in S'$ tel qu'il ait le plus grand nombre de liaisons externes avec les sous-réseaux déjà placés :

$$\sum_{t \in U} t_{i_\ell k} = \max_{j \in S'} \left(\sum_{k \in U} t_{jk} \right).$$

b) Faire $P = \{L^1_{i_\ell}, L^2_{i_\ell}, \ldots, L^{u'}_{i_\ell}\}$: le sous-ensemble de lignes de $\mathcal{L}$ correspondantes aux modules de S'_{i_ℓ} (card (S'_{i_ℓ}) = u').

Aller à 2.1. en remplaçant S'_{i_1} par S'_{i_ℓ}.

3) Amélioration de l'implantation initiale

a) Si card(S') $\geq$ n/2 appliquer l'algorithme de changement de paires pour l'ensemble M de tous les modules et aller à 4).

b) Appliquer l'algorithme de changement de paires pour $S'_1, S'_2, \ldots, S'_q$.

c) S'il y a au moins un échange de places effectivement réalisé, alors aller à 3.b.).

4) Fin.

REMARQUE

A part le fait de fixer à l'avance l'emplacement de certains modules, on peut influencer l'implantation des modules en ajoutant des poids supplémentaires aux liaisons. En effet, en mettant des poids forts aux liaisons d'une paire de modules, on augmente les chances que les 2 modules envisagés fassent partie d'un même sous-réseau minimal et après, à l'implantation du sous-réseau minimal, ils se retrouveront aussi près que possible.

B - TRACE DES CONNEXIONS

On distingue deux étapes dans le tracé des connexions :

1) Recherche de l'arbre minimal de l'équipotentielle, qui satisfait aussi à des contraintes de nature technologique.

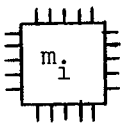
2) Réalisation effective des connexions de l'arbre en tenant compte de divers obstacles de la plaquette.

CHAPITRE I

RECHERCHE DES EQUIPOTENTIELLES MINIMALES

I.1. FORMULATION DU PROBLEME

Soit $M = \{m_1, m_2, \ldots, m_n\}$ l'ensemble de modules. Le module m_i est relié aux autres par l'intermédiaire des pattes de connexion sur ses faces :



Dans le problème d'implantation, on ne s'intéressait qu'au nombre de liaisons de m_i avec m_j ; maintenant, pour la construction effective des équipotentielles nous devons tenir compte de liaisons non plus entre modules, mais entre pattes de modules.

Soit $P = \{p_1, p_2, \ldots, p_m\}$ l'ensemble de toutes les pattes de tous les modules. Une <u>équipotentielle</u> sera maintenant considérée comme un élément de l'ensemble des parties de P.

Réaliser une équipotentielle, c'est relier électriquement toutes les pattes qui la constituent ; donc, le tracé ou réalisation d'une équipotentielle $E_i = \{p_{i_1}, p_{i_2}, \ldots, p_{i_r}\}$ revient à trouver un arbre A_i dans le graphe d'ensemble de sommets E_i et dont l'ensemble de sommets contient $p_{i_1}, p_{i_2}, \ldots, p_{i_r}$.

Le nombre de choix possibles pour un arbre est évidemment très grand. Mais parmi tous les A_i possibles, nous en cherchons un qui satisfait aux contraintes suivantes :

a) le degré dans A_i des sommets ne doit pas dépasser 2, donc l'arbre est en fait une "chaîne".

b) la somme des longueurs des arêtes de l'arbre est minimale, d'où la notion d'arbre "minimal".

La contrainte a) est imposée par le procédé technologique utilisé pour la réalisation des connexions (chapitre III.2.), et la contrainte b) est évidemment nécessaire pour une réalisation économique de l'équipotentielle.

I.2. METHODES DE RESOLUTION

Le problème que nous avons à résoudre, pour une équipotentielle donnée, correspond en théorie des graphes à la recherche d'une chaîne hamiltonienne de longueur minimale. Des solutions à ce problème existent, par exemple :

Pertuiset [21] applique un procédé heuristique qui réduit itérativement les degrés des sommets d'un arbre minimal tels qu'ils soient ≤ 2 ; la méthode conduit à des résultats proches de l'optimum.

Altman [22] propose une méthode de construction directe d'une chaîne, en ajoutant à des chaînes partielles la plus courte arête compatible avec les degrés imposés aux sommets et ne formant pas de cycle. La solution est évidemment approximative.

Kurtzberg [23] donne une méthode simple et rapide de construction progressive de la chaîne en ajoutant aux extrémités d'une chaîne partielle déjà construite une des plus courtes arêtes. Mais les chaînes ainsi obtenues ne sont pas optimales.

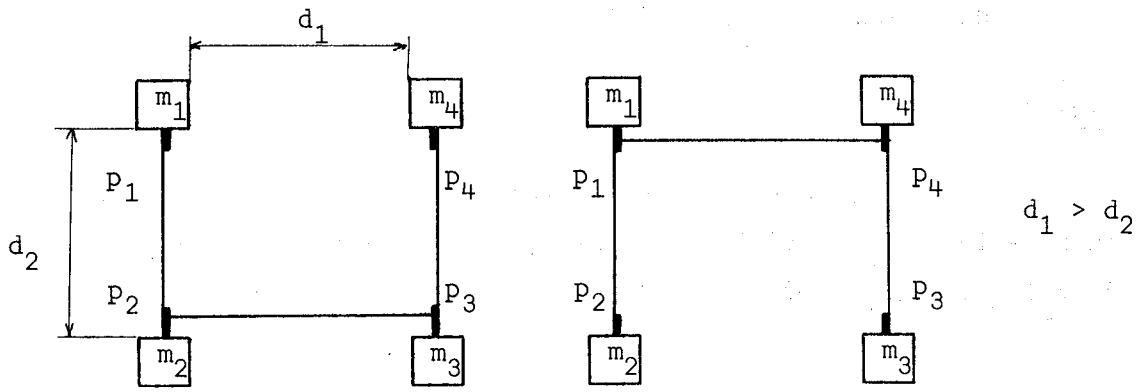
Nous ne retiendrons aucune de ces méthodes, donnant une solution approchée mais qui sont rapides, car nous nous imposons un critère supplémentaire pour le tracé des équipotentielles minimales.

I.3. REPARTITION UNIFORME DES LIAISONS SUR LA PLAQUETTE

En vue d'améliorer la répartition des liaisons sur la plaquette, c'est-à-dire d'éviter des zones trop denses en liaisons, nous choisissons parmi plusieurs chaînes minimales, celle pour laquelle la somme du nombre de liaisons des modules extrémités de la chaîne est la plus grande.

EXEMPLE

Pour l'équipotentielle $\{p_1,p_2,p_3,p_4\}$ ci-dessous, on a deux solutions minimales possibles :



(1) (2)

$$\text{Si } \sum_{j=1}^{n+p} [\mathcal{L}(m_1,m_j) + \mathcal{L}(m_4,m_j)] > \sum_{j=1}^{n+p} [\mathcal{L}(m_2,m_j) + \mathcal{L}(m_3,m_j)]$$

alors on préfère la première solution.

I.4. EQUIPOTENTIELLES MINIMALES ET REPARTITION UNIFORME DES LIAISONS

Soient $P_1,P_2,\ldots,P_n$, n points dans le plan et considérons le graphe complet dont les sommets sont $P_1,P_2,\ldots,P_n$ et chacune de ses arêtes porte un "poids" égal à la distance (euclidienne, orthogonale, ...) entre les sommets qu'elle relie.

D'après Hanan et Kurtzberg [28], rechercher la chaîne minimale revient à rechercher le circuit hamiltonien minimal. En effet, en ajoutant un point de plus, P_{n+1}, relié à tous les autres, tel que chacune de ses arêtes (qui lui sont incidentes) porte un même poids, D, arbitrairement choisi, le circuit hamiltonien minimal K_{n+1} a une "longueur" $\ell(C_n)+2.D$, où $\ell(C_n)$ est la longueur de la chaîne C_n reliant les n points. C_n est la chaîne minimale, car la "longueur" de tout circuit hamiltonien contient 2.D et K_{n+1} est le circuit minimal.

Nous donnons la propriété suivante :

PROPRIETE

Soit K_{n+1} le circuit hamiltonien minimal et notons par P_i,P_j les points reliés à P_{n+1}. Si $\{P_k,P_\ell\}$ est une arête de K_{n+1} et $k,\ell \leq n$, alors $d(P_i,P_j) \geq d(P_k,P_\ell)$ [où $d(P_u,P_v)$ et la distance entre les points P_u,P_v, $u,v \leq n$].

DEMONSTRATION évidente, car s'il existait $\{P_k,P_\ell\} \in K_{n+1}$, $k,\ell \leq n$, tel que $d(P_k,P_\ell) > d(P_i,P_j)$, alors K_{n+1} ne serait pas minimal ; le circuit K'_{n+1} obtenu à partir de K_{n+1} :

- en supprimant les liaisons de P_{n+1} à P_i et P_j et en le reliant à P_k, P_ℓ,
- en reliant P_i et P_j et en supprimant la liaison $\{P_k,P_\ell\}$, serait de "longueur" plus petite que K_{n+1}.

CONSEQUENCE

Considérons le circuit hamiltonien K_n obtenu à partir de K_{n+1} (qui est minimal) en éliminant P_{n+1} et en reliant P_i à P_j.

Une chaîne hamiltonienne minimale est obtenue en éliminant une des plus grandes arêtes de K_n.

Le problème posé en I.1. est résolu de la manière suivante :

1) - A partir de l'équipotentielle complète à n éléments et en ajoutant un élément de plus, le $(n+1)^{ième}$ comme précédemment, on cherche les circuits hamiltoniens minimaux, les K_{n+1}.

La recherche des circuits hamiltoniens minimaux dans un graphe complet est le problème du "voyageur de commerce" pour la résolution duquel il existe de nombreux algorithmes. Ils donnent soit la solution optimale, soit une solution approchée [18].

Les méthodes donnant la solution optimale sont coûteuses en temps de calcul, mais comme dans notre cas particulier le nombre de sommets d'une équipotentielle n'est pas grand (moins de 10 pattes) nous appliquerons une telle méthode (celle de Little, etc ... [20] (voir annexe 2)).

2) - Pour chaque circuit K_{n+1} trouvé on construit K_n et on élimine une connexion ayant la plus grande longueur. S'il y a plusieurs de telles connexions, en éliminant chacune séparément, on obtient plusieurs chaînes minimales ; on retient celle qui donne une meilleure répartition des connexions.

REMARQUE

Basés sur le calcul des chaînes minimales en 2), nous ne retiendrons au pas 1) que des circuits hamiltoniens minimaux K_{n+1} qui nous donnent des K_n différents.

CHAPITRE II

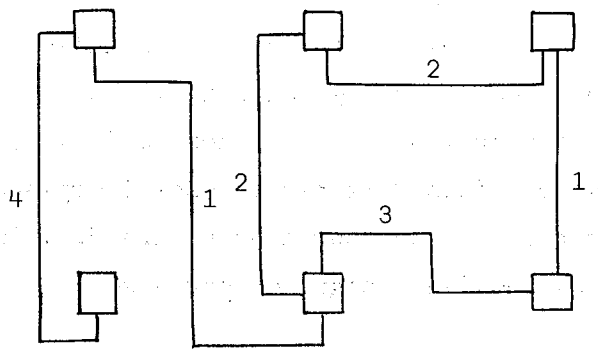
MINIMISATION DU NOMBRE DE COUDES D'UN RESEAU DE MODULES DE CIRCUITS INTEGRES

II.0. PRESENTATION DU PROBLEME

L'implantation des modules de circuits intégrés en minimisant la longueur totale des liaisons a été réalisée en considérant les modules comme ponctuels. En pratique, les liaisons présentent des coudes dont le nombre peut diminuer par des rotations appropriées des modules.

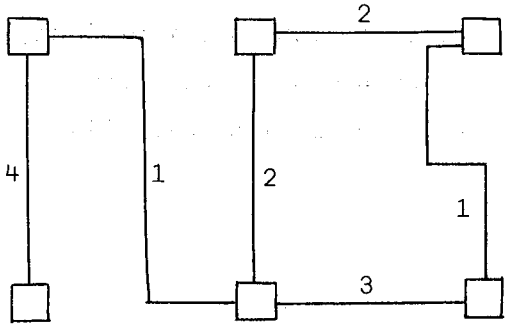
EXEMPLE

Supposons qu'après minimisation des longueurs des liaisons on obtienne l'implantation suivante :



Il y a :
4×3 + 1×4 + 2×2 + 2×2 + 3×3 = 33
coudes dans le réseau.

Par des rotations convenables des modules, on peut avoir le schéma suivant :



Le réseau présente :
1×2 + 1×3 = 5 coudes.

On a intérêt à diminuer le nombre de coudes dans le réseau pour les raisons suivantes :

a) la longueur totale des liaisons diminue.

b) on gagne de la place autour des modules.

c) dans la phase du tracé des connexions le nombre de segments à mémoriser est plus faible.

d) le réseau est plus explicite, plus facile à "lire" et plus esthétique.

Il n'existe pas encore d'algorithmes résolvant ce problème par rotation des modules : jusqu'à maintenant, les modules de circuits intégrés étaient des "boîtiers" que les liaisons touchaient seulement sur deux faces. L'orientation de chaque boîtier sur la plaquette était généralement fixée à l'avance.

Pomentale [24] donne une procédure pour minimiser le nombre de coudes dans le réseau en échangeant les modules de place, afin que les modules reliés entre eux soient sur une même ligne ou une même colonne sur la plaquette. En partant d'une implantation initiale, il échange les positions des modules jusqu'à ce que le nombre de coudes du réseau ne diminue plus. Le résultat dépend de l'implantation initiale qui doit être assez bonne.

Pour résoudre ce problème en faisant tourner les modules, on peut énumérer toutes les configurations possibles et retenir celle pour laquelle le nombre de coudes est minimum. Si n est le nombre de modules, comme chaque module peut avoir 4 positions différentes dans l'emplacement qu'il occupe, il y a 4^n configurations, donc pour n assez grand, l'énumération de toutes les configurations devient longue et coûteuse.

Dans ce qui suit, nous présentons un algorithme semi-énumératif ("branch-and-bound") qui nous donne la solution optimale, c'est-à-dire la configuration pour laquelle le nombre de coudes est minimum, puis un algorithme heuristique rapide, qui donne une solution acceptable, le "graphe des solutions" ne présentant pas un minimum aigu [14].

Dans ce chapitre, ainsi que nous l'avons fait pour l'exemple précédent, nous supposons que le tracé des connexions est réalisé par des segments horizontaux et verticaux.

II.1. DEFINITIONS ET TABLEAUX DES NOMBRES MINIMUM DE COUDES

Nous supposons maintenant les modules placés à l'intersection de lignes horizontales et verticales.

Pour un module donné m_i, on distingue plusieurs sous-ensembles de modules dans le réseau.

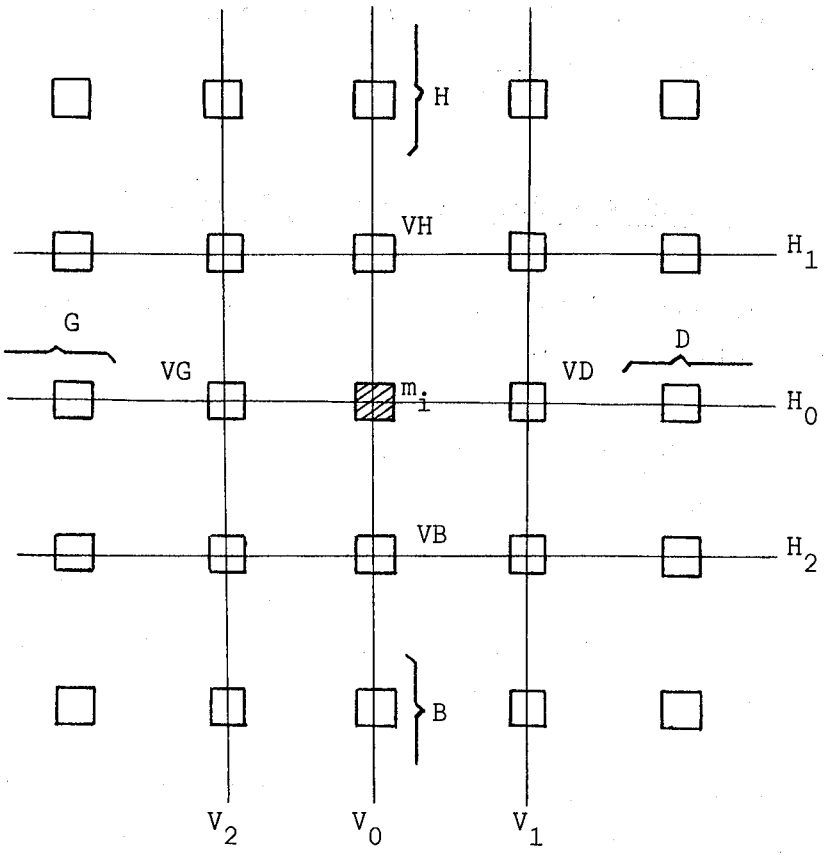
- V_0 : le sous-ensemble des modules qui se trouvent sur la même verticale que m_i. On note par VH le module voisin de m_i par le haut et par VB le module voisin de m_i par le bas.

- H_0 : le sous-ensemble de modules qui se trouvent sur la même horizontale que m_i. On note par VG le module voisin de m_i à gauche et par VD le module voisin de m_i à droite.

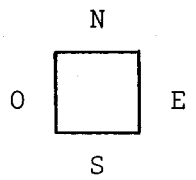
- V_1, V_2 : représentent les sous-ensembles de modules situés sur la même verticale voisine de la verticale de m_i, verticales voisines respectivement à gauche et à droite.

- H_1, H_2 : sont les sous-ensembles de modules situés sur l'horizontale voisine de l'horizontale de m_i, respectivement en haut et en bas.

Dans V_0, on note par H l'ensemble de modules situés au-dessus de VH et par B les modules situés au-dessous de VB. De même, dans H_0, on note par G les modules situés à gauche de VG et par D les modules situés à droite de VD.



Les 4 faces de tous les modules sont notées de la manière suivante :



Considérons le module m_i envisagé précédemment. Le nombre minimum de coudes de la liaison $\{m_i, m_j\}$ où $m_j \in V_0 \cup H_0$ est donné par le tableau suivant, dans lequel les faces de m_i sont en tête des lignes et les faces de m_j en tête des colonnes :

H/V :

m_i \ m_j	N	S	E	O
N	4 H,B 2 G,D	0 VH 4	3	3
S	0 VB 4	4 H,B 2 G,D	3	3
E	3	3	4 G,D 2 H,B	0 VD 4
O	3	3	0 VG 4	4 G,D 2 H,B

EXEMPLE

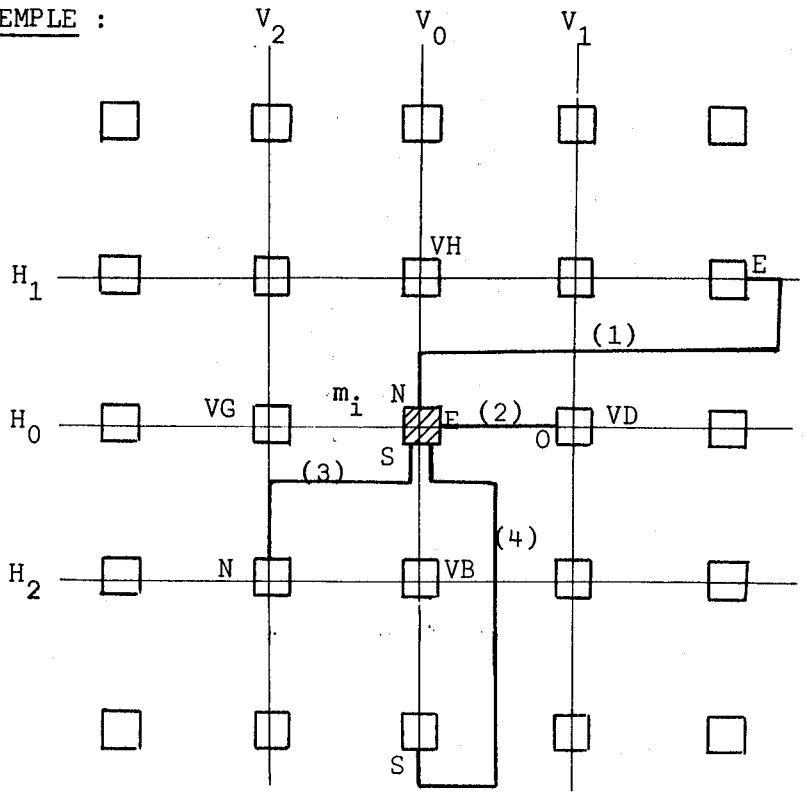
Si la face N de m_i est reliée à la face N de m_j alors le nombre minimum de coudes de la liaison est 4 si $m_j \in H \cup B$, ou 2 si $m_j \in G \cup D$.

Si $m_j \notin H_0 \cup V_0$, alors le nombre minimum de coudes de la liaison $\{m_i, m_j\}$ est donné par le tableau suivant :

D :

m_i \ m_j	N	S	E	O
N	4	2 H1 4	3	3
S	2 H2 4	4	3	3
E	3	3	4	2 V1 4
O	3	3	2 V2 4	4

EXEMPLE :



Le nombre de coudes de la liaison (1) est donné par le tableau D : N de m_i relié à E de m_j : 3 coudes. Le nombre de coudes de la liaison (2) est donné par le tableau H/V : E de m_i relié à 0 de m_j = VD, donc aucun coude ; etc ....

II.2. ALGORITHME A SOLUTION OPTIMALE POUR MINIMISER LE NOMBRE DE COUDES DANS UN RESEAU

II.2.1. Généralités et définitions

L'algorithme que nous donnons utilise le principe combinatoire, connu sous le nom de "branch-and-bound" ([1], [25]).

Gilmore [9] et Lawler [10] utilisent ce même principe pour minimiser la longueur totale des liaisons, mais ceci demande beaucoup de temps de calcul, car :

- il y a n! solutions possibles
- les calculs intermédiaires sont assez laborieux.

Appliquée à l'orientation des modules afin de minimiser le nombre de coudes des liaisons, la méthode est plus rapide, car $4^n < n!$ pour $n > 8$, mais le nombre de calculs croît exponentiellement avec n, donc elle convient pour n pas trop grand ($n < 20$).

Soit le réseau $\mathcal{R} = (M, \mathcal{C}')$ où :

- $M = \{m_1, m_2, \ldots, m_n\}$ est l'ensemble de modules
- $\mathcal{C}'$ est la famille des équipotentielles minimales.

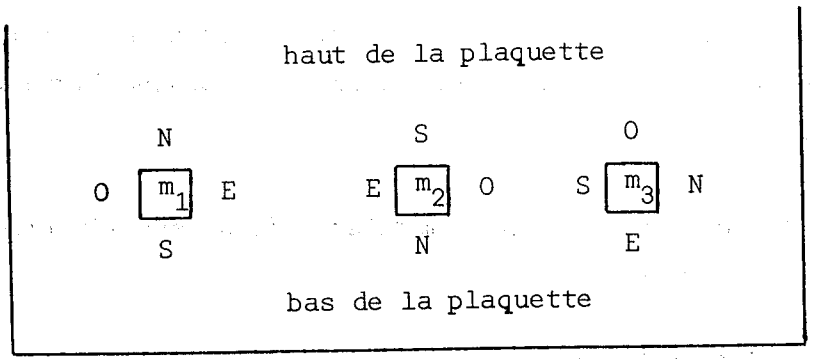
Pour simplifier, on représentera plusieurs liaisons qui relient les mêmes faces d'une paire de modules par le <u>poids</u> d'une seule liaison.

On note par $F = \{N,S,E,O\}$ <u>l'ensemble de faces</u> d'un module.

Une face $F_j \in F$ du module m_i est dite <u>significative</u> si elle occupe une position précise par rapport à un repère fixe externe au réseau.

<u>EXEMPLE</u> :

La face est significative si elle est orientée vers le haut de la plaquette :



Les faces significatives sont N,S,O respectivement pour m_1, m_2, m_3.

On appelle <u>configuration</u> de M (les emplacements des modules de M étant fixés) l'application $f : M \longrightarrow F$

$$m_i \longrightarrow f(m_i)$$

où $f(m_i)$ est la face significative de m_i.

f étant une configuration de M, alors <u>le nombre de coudes</u> associé à cette configuration (ou plus brièvement, le nombre de coudes de f) est :

$$C(f) = \sum_{i=1}^{q} c_i p_i \quad \text{où :}$$

- q est le nombre de liaisons du réseau
- c_i est le nombre de coudes de la liaison i, dans la configuration f
- p_i est le poids de la liaison i.

Si $\mathcal{C}$ est l'ensemble de toutes les configurations, il existe une <u>borne inférieure</u> C_o des C(f), pour $f \in \mathcal{C}$, telle que :

$$C_o = \sum_{i=1}^{q} \min(c_i).p_i \quad \text{où :}$$

$\min(c_i)$ est le minimum de coudes de la liaison i quand les deux modules qu'elle relie peuvent tourner librement. Il est donné par les tableaux H/V et D en considérant la valeur minimale de chaque ligne (ou colonne) des tableaux :

a) $\min(c_i) = 0$ si i relie deux modules voisins sur l'horizontale ou la verticale

b) $\min(c_i) = 2$ si i relie :
- deux modules voisins sur la diagonale
- deux modules qui ne sont pas voisins sur l'horizontale ou la verticale
- deux modules sur des lignes ou colonnes voisines

c) $\min(c_i) = 3$ dans les autres cas.

f étant une configuration de M, si M' $\subset$ M alors g = f/M' (la restriction de f à M') est appelée <u>configuration partielle</u> définie sur M', et f peut être considérée comme un prolongement de g.

Si g est une configuration partielle définie sur M', alors pour tout prolongement f de g à M", M' $\subset$ M", le nombre de coudes de f admet une <u>borne inférieure, B(g)</u> :

$$B(g) = \sum_{i \in (M',M')} c_i.p_i + \sum_{i \in [(M-M'),(M-M')]} \min(c_i).p_i + \sum_{i \in (M',M-M')} \min'(c_i).p_i \quad \text{où}$$

- $i \in (X,Y)$ signifie que la liaison i relie un module $m_i \in X$ à un module $m_j \in Y$
- $\min(c_i)$ et c_i sont définis comme précédemment
- $\min'(c_i)$ est le minimum du nombre de coudes d'une liaison partant d'une face fixée de $m_i \in M'$, aux faces de $m_j \in M-M'$, m_j tournant encore librement.

<u>REMARQUE</u>

Si card(M') = k alors pour g donnée sur M', il existe 4^{n-k} prolongements qui sont des configurations.

Nous utiliserons la propriété évidente suivante :

<u>PROPRIETE</u>

Soient k une configuration de M et g une configuration partielle de M, définie sur M'. Si $B(g) \geq C(k)$ alors pour tout prolongement f de g on a : $C(f) \geq C(k)$.

<u>Le problème</u> que nous posons revient alors à trouver une configuration f telle que :

$$C(f) = \min_{g \in \mathcal{C}} (C(g))$$

De plus, nous disons que f est une configuration acceptable si $\tau.C(f) \leq C_o$, où $0 \leq \tau \leq 1$ est appelé taux d'approximation.

II.2.2. Description de l'algorithme

Nous utilisons la propriété précédente. Etant donné une configuration initiale f, pour laquelle le nombre de coudes est C(f), nous vérifions d'abord si f est une configuration acceptable et sinon nous cherchons, si possible, une succession de configurations :

$g_1, g_2, \ldots, g_k, \ldots$ telles que :

$$C(f) > C(g_1) > C(g_2) > \ldots > C(g_k) > \ldots$$

Les configurations g_k sont construites de la manière suivante : on considère une configuration partielle $h_1^j : M_1 \to F$, où $1 \leq j \leq 4$ et $M_1 = \{m_1\}$, puis un prolongement partiel h_2^j, $1 \leq j \leq 4$ défini sur $M_2 = M_1 \cup \{m_2\}$, etc .... Pour chaque prolongement partiel, on calcule $B(h_i^j)$ et si $B(h_i^j) \geq C(f)$ et $M_i \neq M$ on revient en arrière en choisissant une autre configuration à partir de g_i.

Si $M_i = M$, cela veut dire que h_i^j est une configuration g_i^j de M. Si $C(g_i^j) < C(f)$, on remplace f par g_i^j et on recommence en choisissant une autre configuration à partir de $g_i(h_i)$. L'algorithme s'arrête après l'énumération de toutes les configurations.

SIMPLIFICATIONS DE CALCUL

Soit g une configuration partielle définie sur $M' \subset M$ et soit le prolongement partiel h^j de g à $M'' = M' \cup \{m_i\}$, $1 \leq j \leq 4$, tel que $h(m_i) = F_j$. Alors :

$$B(h^j) = B(g) + \sum_{k \in (\{m_i\}, M')} (c_k - \min'(c_k)).p_k + \sum_{k \in (\{m_i\}, M-M')} (\min'(c_k) - \min(c_k)).p_k ,$$

où : c_k, $\min(c_k)$, $\min'(c_k)$, p_k sont définis comme précédemment. Donc, dans le calcul de $B(h^j)$ nous ne considérerons que les liaisons de m_i avec les autres modules et nous calculons le nombre de coudes ajoutés par chacune de ses liaisons, lorsque la face significative de m_i est fixée à F_j.

M' étant construit en lui ajoutant des modules par ordre croissant de leurs indices, les modules reliés à m_i sont obtenus facilement à partir de la matrice des liaisons $\mathcal{L}$ (éléments non nuls dans la ligne de m_i dans la matrice) et l'appartenance ou non appartenance de m_k à M' est facile à vérifier :

- si $k > i$ alors $m_k \notin M'$
- si $k < i$ alors $m_k \in M'$.

II.2.3. Algorithme

Soient les nouvelles notations suivantes :

- $F = \{0,1,2,3\}$ l'ensemble des faces d'un module où 0,1,2,3 sont respectivement N,S,E,O.
- $M = \{1,2,\ldots,N\}$ l'ensemble de modules et $M_1 = \{1,2,\ldots,I\}$ sous-ensemble de M, où $I \leq N$.

1) Considérer une configuration initiale f sur M et un taux d'approximation τ. Faire $I = 0$, et $M_1 = \emptyset$.

2) Calculer C_o et C(f). Si $\tau.C(f) \leq C_o$, alors f est une solution acceptable ; aller à 7).

3) $I = I+1$; $M_1 = M_1 \cup \{I\}$; $J(I) = 0$; on spécifie donc une configuration partielle $g_I^{J(I)}$ sur M_I.

4) Calculer $B(g_I^{J(I)})$. Si $B(g_I^{J(I)}) < C(f)$ aller à 6).

5) $J_{(I)} = J_{(I)}+1$. Si $J(I) \leq 3$ aller à 4).

Si $J_{(I)} = 4$, alors :

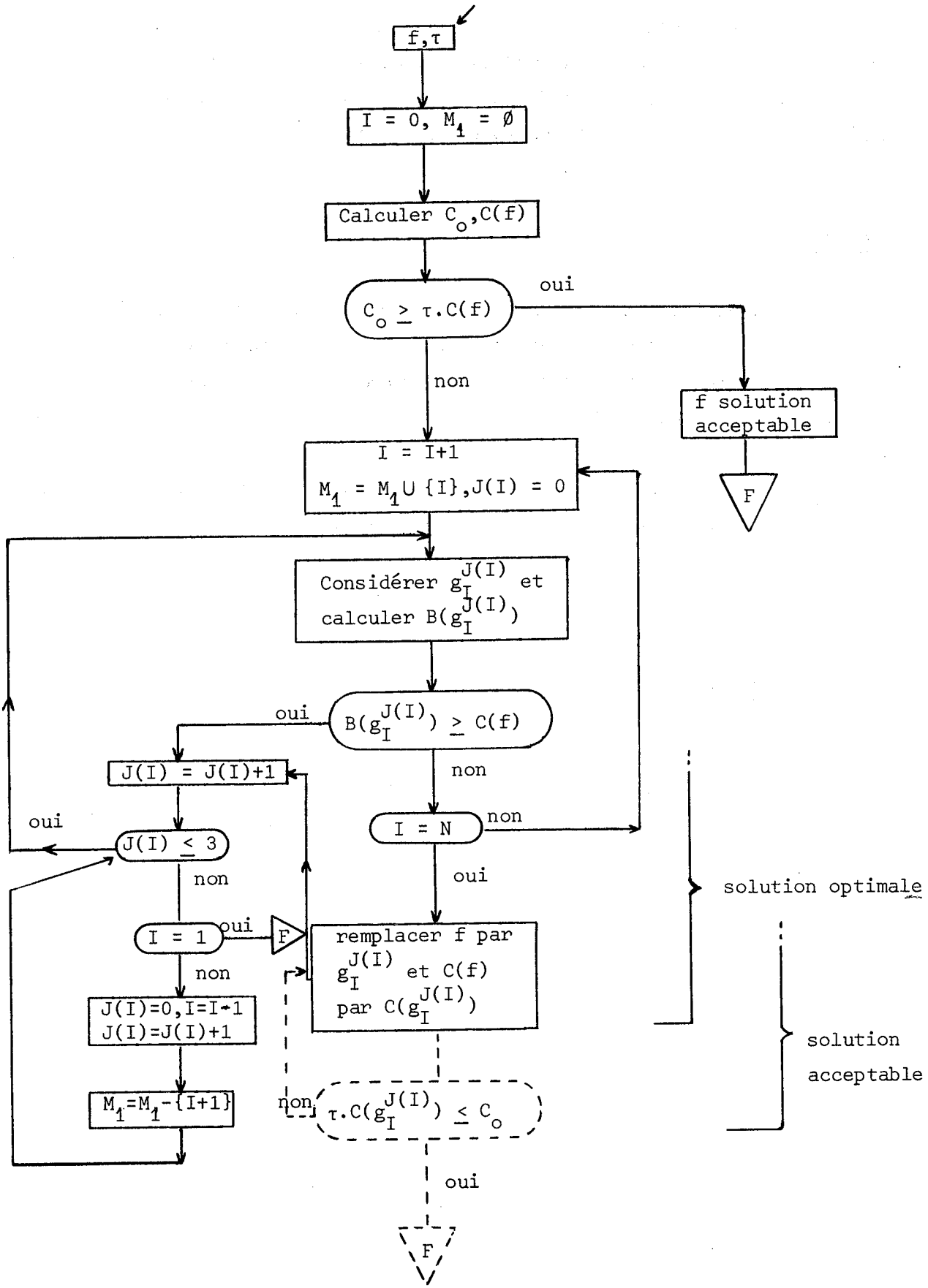
a - si $I = 1$ alors aller à 7)

b - si $I \neq 1$ alors : $J(I) = 0$; $M_1 = M_1 - \{I\}$;
$I = I-1$; aller à 5).

6) Si I = N, alors remplacer f par $g_I^{J(I)}$ et C(f) par $C(g_I^{J(I)})$ et aller à 5) ; sinon aller à 3).

7) fin.

ORGANIGRAMME



REMARQUE 1

Pour accélérer l'algorithme mais en perdant la solution optimale, on peut comparer au pas 6) le nombre de coudes de la configuration g_I^J, qui est "meilleure" que f, à C_o ; si $\tau.C(g_I^J) \leq C_o$ alors on arrête les calculs en ayant une solution acceptable g_I^J.

REMARQUE 2

La rapidité de calcul de l'algorithme dépend de la configuration initiale f ; si f est proche de la configuration optimale, alors le nombre de configurations envisagées par l'algorithme sera réduit, beaucoup de g_I^J étant éliminés dès le début.

Nous nous proposons donc, de construire une bonne configuration initiale au moyen d'un algorithme simple et rapide.

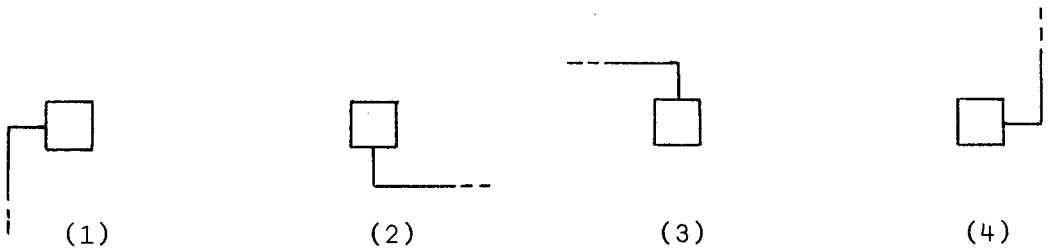
II.3. CONSTRUCTION D'UNE CONFIGURATION INITIALE

II.3.1. Définitions et propriétés

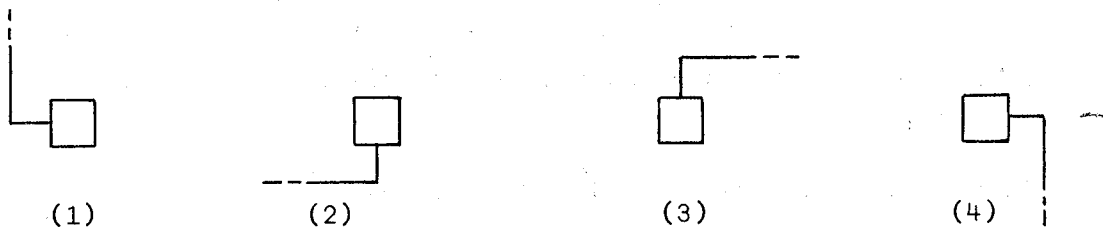
Nous simulons une liaison $\{m_i, m_j\}$ par deux "moments de rotation" M_i, M_j, appliqués respectivement à m_i, m_j, sous l'action desquels chaque module de la paire prend une position correspondante au nombre minimum de coudes de la liaison.

Le plan de la plaquette étant orienté dans le sens trigonométrique, les moments de rotation sont définis de la manière suivante :

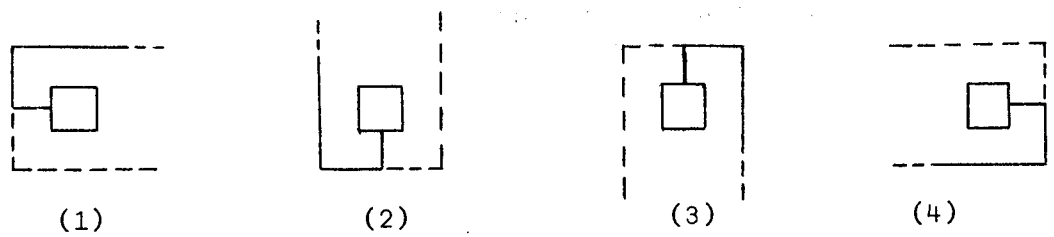
- $\vec{g_1}$ est le moment unité de rotation d'un module de $+\frac{\Pi}{2}$, généré par une liaison dans les 4 cas suivants :



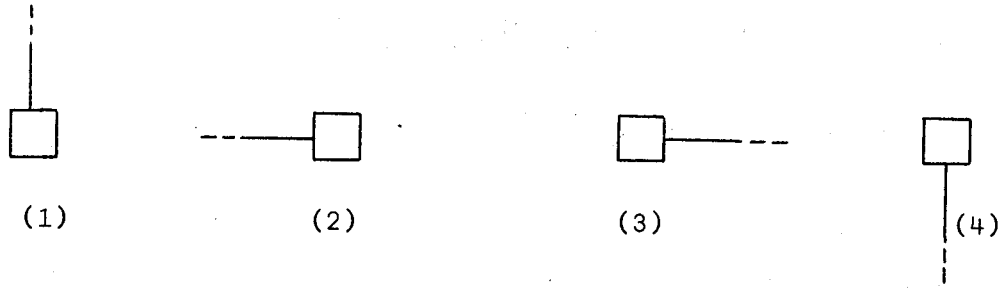
- $\vec{d_1}$ est le moment unité de rotation d'un module de $-\frac{\Pi}{2}$, généré par une liaison dans les 4 cas suivants :



- $\vec{g_2}$ est le moment unité de rotation d'un module de $\pm \Pi$, généré par une liaison dans les 4 cas suivants :



• $\vec{op}$ est le moment unité qui s'oppose à toute rotation d'un module, généré par une liaison dans les 4 cas suivants :



REMARQUE :

$\vec{g_1}$, $\vec{d_1}$, $\vec{g_2}$, $\vec{op}$ correspondent à des liaisons de poids unité.

On note par $\vec{G1}$, $\vec{D1}$, $\vec{G2}$, $\vec{OP}$ les moments résultants, tels que :

$$\vec{G1} = (\sum_i p_i^{g_1})\, \vec{g_1} \quad , \quad \vec{D1} = (\sum_i p_i^{d_1})\, \vec{d_1}$$

$$\vec{G2} = (\sum_i p_i^{g_2})\, \vec{g_2} \quad , \quad \vec{OP} = (\sum_i p_i^{op})\, \vec{op} \quad , \quad \text{où :}$$

$p_i^{g_1}$, $p_i^{d_1}$, $p_i^{g_2}$, p_i^{op} sont respectivement les poids des liaisons qui génèrent des moments de rotation $\vec{g_1}$, $\vec{d_1}$, $\vec{g_2}$, ou s'opposent à la rotation du module.

On note par G1, D1, G2, OP les valeurs absolues des moments résultants.

En associant des moments de rotation à toutes les liaisons incidentes à un module, on obtient le moment résultant qui agit sur le module de la manière suivante :

(1) on calcule d'abord $\vec{R1} = \vec{G1} + \vec{D1} = |G1-D1|\ \vec{u}$ où :

$$\vec{u} = \begin{cases} \vec{g_1} & \text{si } G1 > D1 \\ \vec{d_1} & \text{si } G1 < D1 \\ \vec{0} & \text{si } G1 = D1 \end{cases}$$

(2) on calcule $\overrightarrow{R2} = \max (R1, G2)\vec{v}$, où :

$$\vec{v} = \begin{cases} \vec{g_2} & \text{si } G2 > R1 \\ \vec{r_1} & \text{si } G2 < R1 \\ \vec{0} & \text{si } G2 = R1 \end{cases}$$

(3) on calcule $\vec{R3} = \max (R2, OP)\ \vec{x}$, où :

$$\vec{x} = \begin{cases} \vec{r_2} & \text{si } R2 > OP \\ \vec{op} & \text{si } R2 < OP \\ \vec{0} & \text{si } R2 = OP \end{cases}$$

La justification des 3 étapes est évidente :

en (1) on fait la somme vectorielle habituelle, car $\vec{g_1} + \vec{d_1} = \vec{0}$.

en (2) et (3) on retient le maximum des moments (en valeur absolue) car il n'existe pas de valeur intermédiaire entre $-\frac{\Pi}{2}$, $\frac{\Pi}{2}$ et Π pour l'orientation du module.

Si $\vec{x} \neq \vec{op}$ on fera la rotation du module suivant $\vec{r_2}$.

PROPRIETE 1.

Soient m_i un module quelconque, i_1, i_2, ..., i_x les liaisons sans coudes partant de m_i, j_1, j_2, ..., j_y les liaisons ayant des coudes et partant de m_i. Nous notons :

$$p_s = p_{i_1} + p_{i_2} + \ldots + p_{i_x} \quad \text{et}$$

$$p_c = p_{j_1} + p_{j_2} + \ldots + p_{j_y} \text{ , où } p_k \text{ est le poids de la liaison k.}$$

Si $3 \cdot p_s > 4 \cdot p_c$ alors toute rotation du module m_i fait augmenter le nombre de coudes des liaisons incidentes à m_i.

DEMONSTRATION

Le nombre de coudes des liaisons incidentes à m_i est :

$$N_c = c_1 \cdot p_{j_1} + c_2 \cdot p_{j_2} + \ldots + c_y \cdot p_{j_y}$$

où c_k est le nombre de coudes de la liaison de poids p_{j_k}. Mais :

$$N_c \leq 4 \cdot (p_{j_1} + p_{j_2} + \ldots + p_{j_y}) = 4 \cdot p_c$$

car pour une liaison ayant un nombre minimum de coudes, ce nombre est au plus 4 (voir tableaux des nombres minimum de coudes H/V, D).

Toute rotation du module m_i introduit au moins 3 coudes pour chaque liaison sans coudes, car ces liaisons relient nécessairement des modules voisins sur une même horizontale ou verticale et le nombre minimum des lignes de H/V, est 3 (sauf zéro).

Supposons qu'il existe une rotation qui supprime les coudes des liaisons qui en avaient. Donc, N_c après cette rotation devient au moins.

$$N'_c = 3 \cdot p_{i_1} + 3 \cdot p_{i_2} + \ldots + 3 \cdot p_{i_x} = 3 \cdot p_s$$

Comme $3 \cdot p_s > 4 \cdot p_c$ il résulte $N'_c > N_c$.

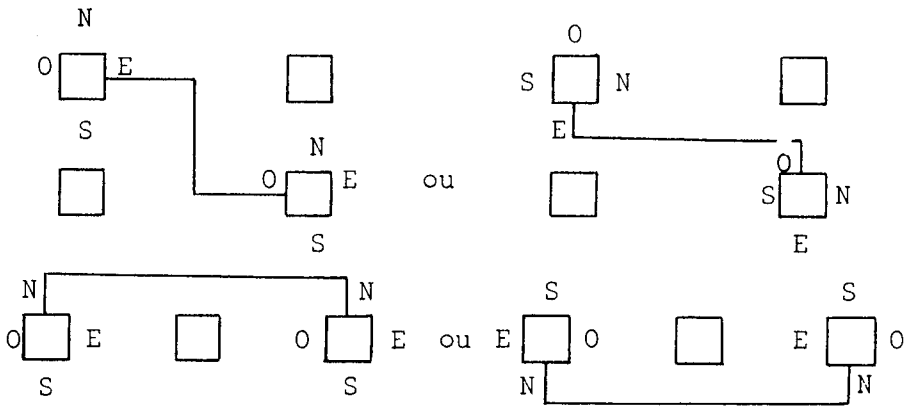
II.3.2. Algorithme

Supposons la somme des longueurs des liaisons minimisée. Dans la construction d'une configuration initiale on s'intéresse seulement aux liaisons qui relient des modules voisins sur l'horizontale et la verticale et cela pour les raisons suivantes :

- l'implantation des modules étant telle que la somme des longueurs des liaisons soit la plus petite possible, alors la plupart des liaisons de poids fort relient des modules voisins sur l'horizontale et la verticale

- la réduction du nombre de coudes d'une liaison reliant deux modules voisins sur l'horizontale ou la verticale peut aller jusqu'à 4, tandis que pour les autres liaisons elle est au maximum de 2 (voir tableau D et H/V pour H, B, G, D).

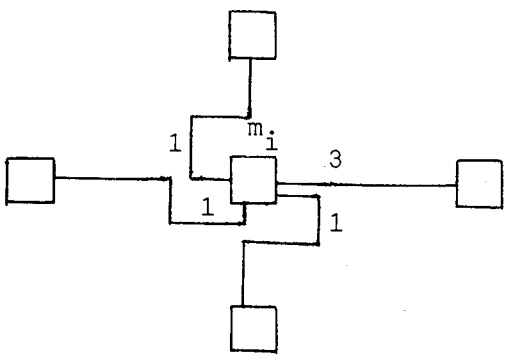
• pour les liaisons entre modules voisins sur l'horizontale ou la verticale les moments de rotation associés dépendent seulement des emplacements physiques des modules et non de leurs positions dans les emplacements ; pour les autres paires de modules il peut correspondre un nombre minimum de coudes pour des orientations différentes des deux modules à la fois :



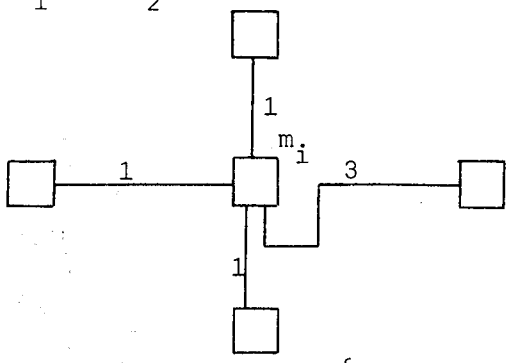
Afin d'éviter des calculs inutiles et pour accélérer l'algorithme, on utilise la propriété 1 pour chaque module, avant le calcul des moments de rotation : on fait la somme p_c des poids des liaisons ayant des coudes et la somme p_s des poids des liaisons sans coudes et si $3 \cdot p_s > 4 \cdot p_c$, alors le module reste inchangé et on passe au module suivant.

Pour gagner de la place autour du module, si R3 = OP alors on décide de faire tourner le module dans le sens de $\vec{R3}$, à condition que le nombre de liaisons ayant des coudes soit plus grand que le nombre de liaison sans coudes.

EXEMPLE :



Nous avons : $\vec{R3} = \vec{D1} = 3 \cdot \vec{d_1}$ et $\vec{OP} = 3 \cdot \vec{op}$. Donc R3 = OP, mais comme il y a 3 liaisons coudées et une liaison sans coudes, en effectuant la rotation de m_i de $-\frac{\Pi}{2}$ nous obtenons la configuration suivante :



L'ensemble de module $M = \{m_1, m_2, \ldots, m_n\}$ étant ordonné d'une manière quelconque, l'agorithme est le suivant :

(1) Considérer le premier module de M : I = 1.

(2) Calculer : • X : le nombre de liaisons sans coudes et p_s
• Y : le nombre de liaisons coudées et p_c.

(3) Si $3 \cdot p_s > 4 \cdot p_c$ alors aller à (7).

(4) Calculer $\vec{G1}$, $\vec{D1}$, $\vec{G2}$, $\vec{OP}$ à l'aide du tableau H/V.

(5) Faire le calcul de $\vec{R1}$, $\vec{R2}$, $\vec{R3}$ comme indiqué précédemment.

(6) Si (R3 > OP) ou [(R3 = OP) et (X < Y)] alors faire la rotation du module I au sens de $\vec{R3}$.

(7) Considérer le module suivant de M : I = I + 1.
Si I ≤ N alors aller à (2) sinon fin de l'algorithme.

Estimation du nombre moyen de calculs.

Soit n le nombre de modules ; on les considère rangés sur p lignes et q colonnes sur la plaquette. Le nombre d'opérations est maximum pour le graphe qui contient toutes les liaisons qui relient des modules voisins sur l'horizontale ou la verticale, les modules étant considérés comme ponctuels.

Dans ce cas, le nombre total de moments à calculer, engendré par de telles liaisons est :

$$F = 4\,[n-2p-2(q-2)] + 3(2p-4) + 3(2q-4) + 4x2, \text{ car :}$$

- les modules non sur le bord dont le nombre est n-2p-2(q-2) sont reliés à 4 modules voisins.

- les 2p-4 modules des 2 colonnes des bords et les 2q-4 modules des 2 lignes des bords sont reliés à 3 modules voisins.

- les 4 modules dans les coins sont reliés à 2 modules voisins.

On obtient :

$$F = 4\ n-2p-2q = 2(2n-p-q).$$

En passant à la représentation réelle des modules et en considérant en moyenne K liaisons différentes entre 2 modules voisins (les liaisons qui relient les mêmes faces de 2 modules voisins sont comprises dans le poids d'une seule liaison significative), on a :

$$F = 2K\ (2n-p-q)$$

Si pour le calcul d'un moment on fait Z opérations, on en déduit : 2KZ (2n-p-q), donc le nombre d'opérations est de l'ordre de grandeur de n, ce qui montre que l'algorithme est très rapide. (En pratique, $K \leq 4$).

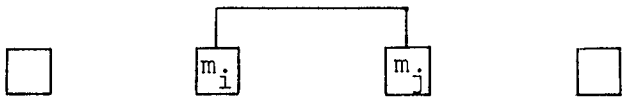
II.4. ALGORITHME HEURISTIQUE POUR MINIMISER LES COUDES DANS UN RESEAU

Généralement, l'algorithme optimal demande un temps de calcul important, donc il est coûteux. On peut résoudre le problème de la minimisation de coudes, d'une manière satisfaisante, à l'aide d'un algorithme heuristique, dont le temps de calcul est réduit.

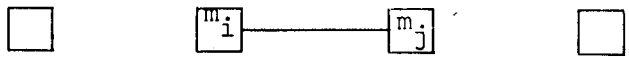
L'algorithme précédent utilisant les moments de rotation peut donner une bonne solution au problème de minimisation du nombre de coudes (en associant des moments de rotation à toutes les liaisons)

s'il est itéré plusieurs fois, c'est-à-dire, si on reprend l'itération tant qu'un module bouge. On ne peut pas prévoir le nombre des itérations.

Nous présentons un autre algorithme heuristique, dont le nombre d'itérations est limité et qui déduit à chaque pas des calculs une configuration meilleure que la précédente. Le principe est de considérer chaque liaison et de chercher à réduire, si c'est possible, le nombre de coudes de cette liaison en tournant à la fois les deux modules. Le fait de considérer à la fois les deux modules, découle de l'exemple simple suivant :



Si on tourne les modules isolément, la liaison l_{ij} possédant déjà un nombre minimum de coude, toute rotation de m_i ou de m_j augmentera le nombre de coudes ; donc on n'est pas sûr que la configuration finale obtenue soit meilleure que la précédente. Par contre, si l'on tourne à la fois m_i et m_j on arrive au résultat suivant :

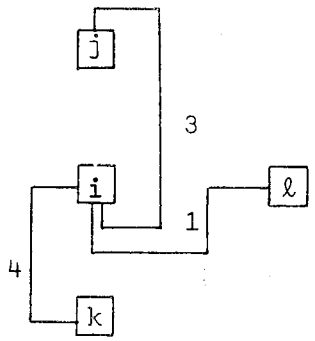


II.4.1. Préliminaires

"Minimiser" une liaison signifie faire tourner les deux modules qu'elle relie de façon à ce que le nombre de coudes devienne minimum.

Si i, j sont les deux modules considérés on notera par l_{ij} la liaison, par p_{ij} le poids de cette liaison et par t_{ij} le nombre de coudes de la liaison (tournants).

a) - Considérons le réseau suivant :

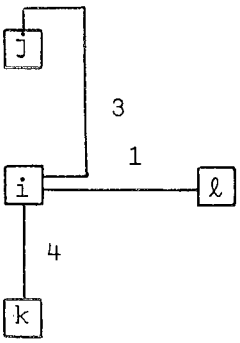


Si on minimise d'abord la liaison l_{ik} on réduit le nombre de coudes du réseau de : (2x4) + (1x3) + (3x1) = 14.

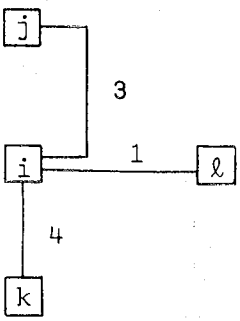
Si on minimise d'abord l_{ij} le nombre de coudes est réduit de :

(4x3) - (2x4) = 4.

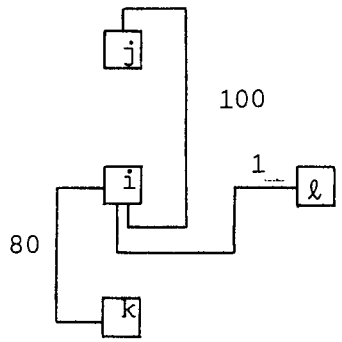
On choisit donc de minimiser l_{ik}. On obtient le réseau suivant :



On continue en cherchant à minimiser l_{ij}. La réduction dans le nombre de coudes est : (3x3) - (3x4) - (3x1) = -6, c'est-à-dire le nombre de coudes augmente de 6 si on minimise l_{ij}. En faisant tourner seulement le module j, on obtient le réseau à nombre de coudes minimum :



b) - Considérons maintenant le même réseau, mais avec des poids différents des liaisons :



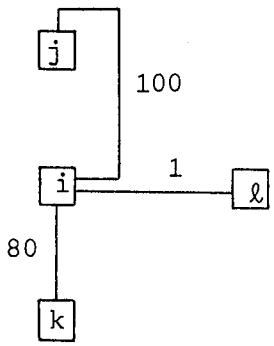
Si on minimise d'abord l_{ik} on réduit le nombre de coudes de :

$$(2x80) + (1x100) + (3x1) = 263$$

Si on minimise d'abord l_{ij}, la réduction est de :

$$(4x100) - (2x80) = 240$$

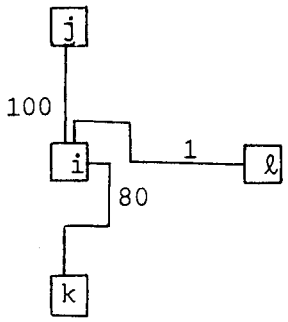
La minimisation de l_{ik} étant la plus forte, on choisit de la faire d'abord.



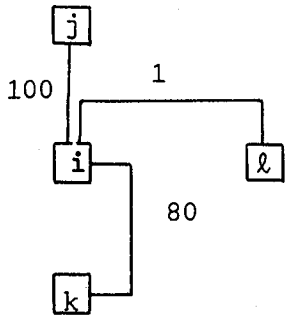
On continue en cherchant à minimiser l_{ij}. La réduction est :

$$(3x100) - (3x80) - (3x1) = 57,$$

donc on minimise l_{ij} et le réseau obtenu devient :



En continuant à minimiser l_{ik} et l_{il} on obtiendra le réseau suivant :

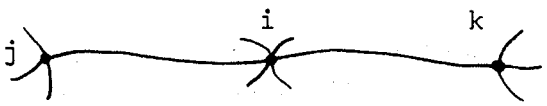


Dans les exemples a) et b) nous remarquons que bien que dans la recherche de l'orientation du module m_i on ait toujours choisi l'orientation de i donnant une réduction maximale du nombre de coudes des liaisons incidentes à i, ce module ne garde pas toujours cette orientation quand on passe à l'étude des liaisons avec les autres modules voisins. Donc, en principe nous pourrons toujours réexaminer l'orientation des modules, ce qui donne plusieurs itérations de l'algorithme, jusqu'à ce qu'aucun module ne change d'orientation.

Nous déduisons maintenant une condition suffisante pour que l'orientation du module i reste inchangée quand on passe aux modules voisins de i. En appliquant cette condition on évite donc le retour en arrière et l'algorithme n'itère pas.

II.4.2. Notations, définitions, propriétés

Soient trois modules i, j, k reliés comme ci-dessous :



Nous faisons les notations suivantes :

- N_{ij}^{k} : réduction du nombre de coudes des liaisons de i, sans compter l_{ij} et l_{ik}, si l'on minimise l_{ij}. Pareil par N_{ik}^{j}.

- N_{ij} : réduction du nombre de coudes des liaisons de j, sans compter l_{ij}, si l'on minimise l_{ij}. Pareil par N_{ik}.

En considérant le poids égal à 1 (liaison simple), nous définissons :

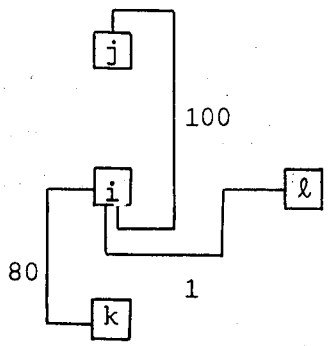
- α_{ij}^{k} : nombre de coudes introduits dans l_{ij} si on minimise l_{ik}. Pareil pour α_{ik}^{j}.

γ_{ij}^{k} : nombre de coudes introduits dans l_{ij} quand elle est minimisée et on minimise l_{ik}. Pareil pour γ_{ik}^{j}

δ_{ij}^{k} : nombre de coudes introduits dans l_{ij} si on tourne j seul à la minimisation correspondante de l_{ij}, la liaison l_{ik} étant minimisé Pareil pour δ_{ik}^{j}.

EXEMPLE :

Pour le réseau donné précédemment on a :



- $t_{ij} = 4$, $t_{ik} = 2$
- $N_{ij}^{k} = 0$, $N_{ik}^{j} = 3$
- $N_{ij} = 0$, $N_{ik} = 0$
- $\alpha_{ij}^{k} = -1$, $\gamma_{ij}^{k} = 3$, $\delta_{ij}^{k} = 0$
- $\alpha_{ik}^{j} = 2$, $\gamma_{ik}^{j} = 3$, $\delta_{ik}^{j} = -1$.

PROPRIETE 1.

Si dans la minimisation de l_{ik} et de l_{ij} on doit tourner le module i, alors la minimisation de l_{ik} (l_{ij}) introduit α_{ij}^{k} (α_{ik}^{j}) coudes dans la liaison l_{ij} (l_{ik}), où α_{ij}^{k} (α_{ik}^{j}) prend des valeurs entières dans l'intervalle [-2, 2].

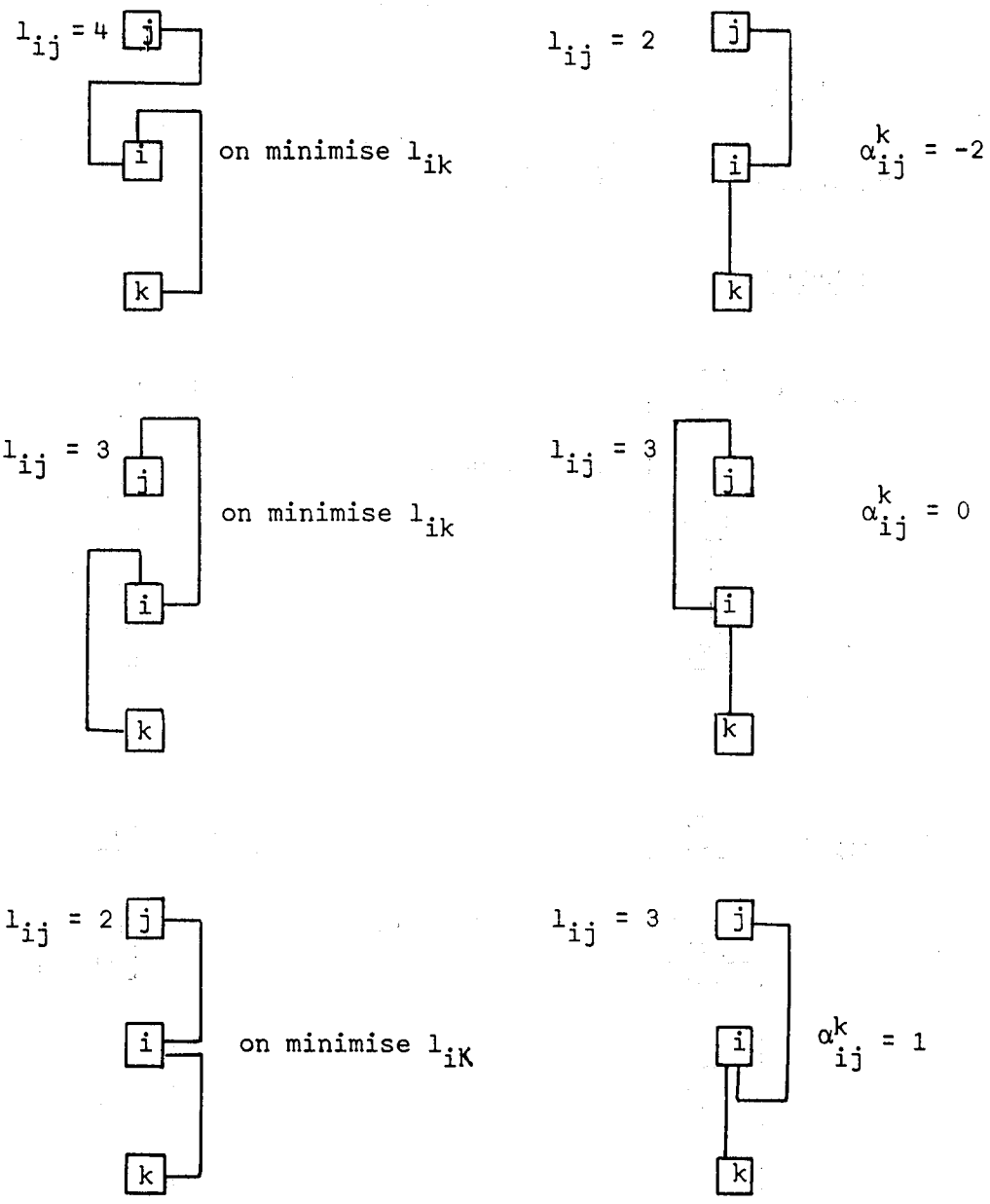
DEMONSTRATION.

Evidente d'après les tableaux H/V, D. l_{ij} (l_{ik}) n'étant pas minimisée, on supprime les minimum des tableaux H/V ou D. En minimisant l_{ik} (l_{ij}) le nombre des coudes de l_{ij} (l_{ik}) passe de p à q où p, q $\in$ {2, 3, 4}.

REMARQUE.

Pour les fils "diagonaux", en supprimant les minimum de D, on a : p, q $\in$ {3, 4}, donc α^k_{ij} (α^j_{ik}) se réduit aux valeurs entières de [-1, 1].

EXEMPLES :



PROPRIETE 2.

Si l_{ik} (l_{ij}) est minimisée et pour la minimisation de l_{ij} (l_{ik}) il faut tourner i, alors on introduit γ^{j}_{ik} (γ^{k}_{ij}) coudes dans la liaison l_{ik} (l_{ij}) si l'on minimise l_{ij} (l_{ik}), où $\gamma \in \{1, 2, 3, 4\}$.

DEMONSTRATION.

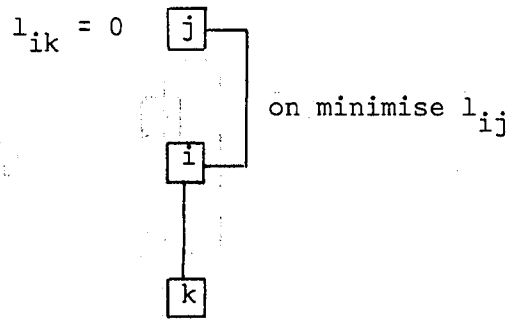
La liaison l_{ik} (l_{ij}) étant minimisée elle peut avoir 0 coudes o 2 coudes (d'après les tableaux H/V et D). Comme la minimisation correspon à une orientation unique de i, quand on tourne i pour la minimisation de l_{ij} (l_{ik}) le nombre des coudes passe à p, où $p \in \{3, 4\}$.

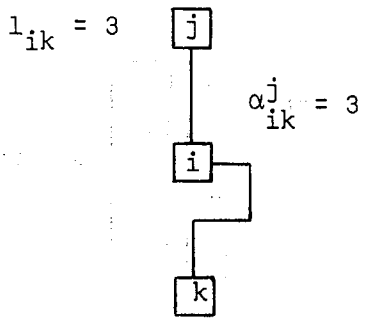
Donc $\gamma \in \{1, 2, 3, 4\}$.

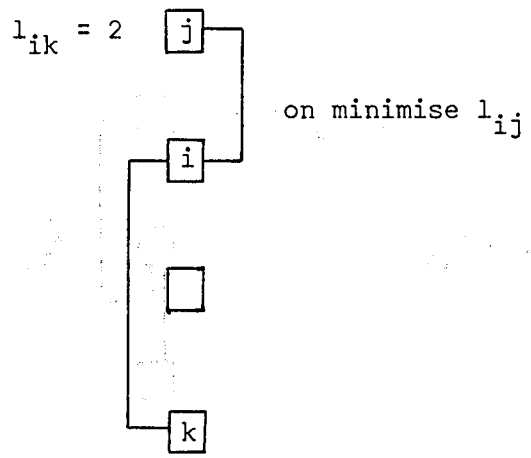
REMARQUE.

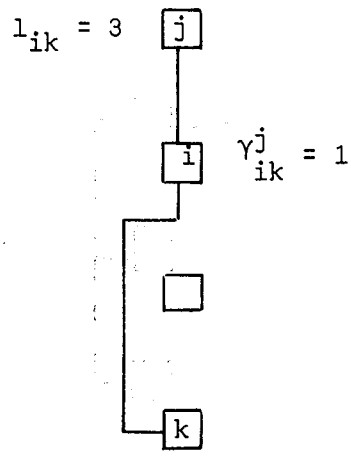
Pour les fils sans coudes (dont le minimum est zéro dans le tableau H/V), $\gamma \in \{3, 4\}$.

EXEMPLES :









PROPRIETE 3.

Si l_{ik} (l_{ij}) est minimisée et pour la minimisation de l_{ij} (l_{ik}) il faut tourner i et j (k), alors on introduit δ^k_{ij} (δ^j_{ik}) coudes dans la liaison l_{ij} (l_{ik}) si l'on tourne seulement j (k) à la position correspondante au minimun de l_{ij} (l_{ik}),

où δ^k_{ij} (δ^j_{ik}) $\in \{0, 1, -1\}$.

DEMONSTRATION.

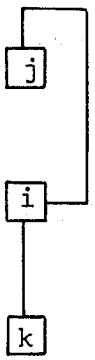
En effet, si l_{ij} (l_{ik}) n'est pas minimisée, elle peut avoir 2, 3 ou 4 coudes, d'après les tableaux H/V, D. Dans chaque ligne des tableaux, le nombre minimum des coudes qui ne correspond pas à la minimisation de l_{ij} (l_{ik}) est 3, donc on a la variation de δ^k_{ij} (δ^j_{ik}) de -1 à 1.

REMARQUE.

Si l_{ij} (l_{ik}) est un fil "diagonal", alors s'il n'est pas minimisé il peut avoir 3 ou 4 coudes (d'après le tableau D); donc, δ^k_{ij} (δ^j_{ik}) $\in \{0, -1\}$.

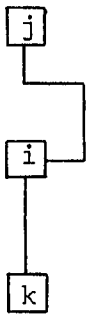
EXEMPLES :

$l_{ij} = 3$



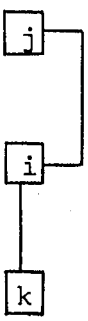
on tourne j à la minimisation correspondante de l_{ij}

$l_{ij} = 3$

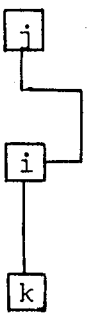


$\delta^k_{ij} = 0$

$l_{ij} = 2$



$l_{ij} = 3$



$\delta^k_{ij} = 1$

Soient :

C : le nombre de coudes du réseau

$N_i^j = N_{ij}^k + t_{ij} \cdot p_{ij} - \alpha_{ik}^j \cdot p_{ik}$: réduction du nombre de coudes du réseau si l'on minimise ℓ_{ij}

$N_i^k = N_{ik}^j + t_{ik} \cdot p_{ik} - \alpha_{ij}^k \cdot p_{ij}$: réduction du nombre de coudes du réseau si l'on minimise ℓ_{ik}.

On réduit le nombre de coudes du réseau de deux manières différentes :

(1) • on minimise d'abord $\underline{\ell_{ij}}$. Le nombre de coudes du réseau devient :

$$C' = C - N_i^j - N_{ij}$$

• après, on tourne le module K à la minimisation correspondante de ℓ_{ik}. Le nombre de coudes du réseau devient :

$$C_{ij}^k = C' - N_{ik} + \delta_{ik}^j \cdot p_{ik}$$

(2) • on minimise d'abord $\underline{\ell_{ik}}$. Le nombre de coudes du réseau devient :

$$C'' = C - N_i^k - N_{ik}$$

• après, on tourne le module j à la minimisation correspondante de ℓ_{ij}. Le nombre de coudes du réseau devient :

$$C_{ik}^j = C'' - N_{ij} + \delta_{ij}^k \cdot p_{ij}$$

THEOREME 1

$$\text{Si } N_i^j - N_i^k \geq p_{ij} \, (t_{ij} + \alpha_{ij}^k - \gamma_{ij}^k) + \delta_{ik}^j \cdot p_{ik} \text{ alors } C_{ij}^k \leq C_{ik}^j.$$

$$\text{Si } N_i^j - N_i^k < p_{ij} \, (t_{ij} + \alpha_{ij}^k - \gamma_{ij}^k) + \delta_{ik}^j \cdot p_{ik} \text{ alors } C_{ij}^k > C_{ik}^j.$$

DEMONSTRATION

Il suffit de remarquer que (2) est équivalent à (2') suivant :

(2') • on minimise d'abord ℓ_{ij}. Le nombre de coudes du réseau, comme en (1), devient :

$$C' = C - N_i^j - N_{ij}$$

• on minimise après ℓ_{ik}. Le nombre de coudes du réseau devient :

$C_{ik}^j = C' + N_{ij}^k + \gamma_{ij}^k \cdot p_{ij} - N_{ik}^j - (t_{ik}+\alpha_{ik}^j) \cdot p_{ik} - N_{ik}$, car :

à C' il faut ajouter la réduction faite auparavant par la minimisant de ℓ_{ij} et, en plus, le nombre de coudes introduits $\gamma_{ij}^k \cdot p_{ij}$ par la rotation de i à la minimisation correspondante de ℓ_{ik}. Après, il faut soustraire la réduction correspondante à la minimisation de ℓ_{ik}, dont le nombre de coudes a été modifié par α_{ik}^j à la suite de la minimisation précédente de ℓ_{ij}. Donc :

$$C_{ik}^j - C_{ij}^k = C' + N_{ij}^k + \gamma_{ij}^k \cdot p_{ij} - N_{ik}^j - (t_{ik}+\alpha_{ik}^j) \cdot p_{ik} - N_{ik} - C' + N_{ik} - \delta_{ik}^j \cdot p_{ik}.$$

En ajoutant et soustrayant des termes pour obtenir N_i^j et N_i^k on a :

$$C_{ik}^j - C_{ij}^k = (N_{ij}^k + t_{ij} \cdot p_{ij} - \alpha_{ik}^j \cdot p_{ik}) - (N_{ik}^j + t_{ik} \cdot p_{ik} - \alpha_{ij}^k \cdot p_{ij})$$

$$+ \gamma_{ij}^k \cdot p_{ij} - t_{ij} \cdot p_{ij} - \alpha_{ij}^k \cdot p_{ij} - \delta_{ik}^j \cdot p_{ik} =$$

$$= N_i^j - N_i^k - p_{ij}(t_{ij} + \alpha_{ij}^k - \gamma_{ij}^k) - \delta_{ik}^j \cdot p_{ik}.$$

Si $N_i^j - N_i^k \geq p_{ij} (t_{ij} + \alpha_{ij}^k - \gamma_{ij}^k) + \delta_{ik}^j \cdot p_{ik}$, alors on a évidemment :

$C_{ik}^j - C_{ij}^k \geq 0$ donc $\underline{C_{ij}^k \leq C_{ik}^j}$.

Si $N_i^j - N_i^k < p_{ij} (t_{ij} + \alpha_{ij}^k - \gamma_{ij}^k) + \delta_{ik}^j \cdot p_{ik}$, on obtient :

$$C_{ik}^j - C_{ij}^k < 0 \quad \text{donc} \quad \underline{C_{ij}^k > C_{ik}^j}.$$

On fait les notations suivantes :

$$r_{ij}^{k} = p_{ij} \, (t_{ij} + \alpha_{ij}^{k} - \gamma_{ij}^{k}) + \delta_{ik}^{j} \cdot p_{ik}$$

$$r_{ik}^{j} = p_{ik} \, (t_{ik} + \alpha_{ik}^{j} - \gamma_{ik}^{j}) + \delta_{ij}^{k} \cdot p_{ij} \, .$$

COROLLAIRE 1

$$\text{Si } N_i^j - N_i^k < r_{ij}^k \quad \text{alors} \quad N_i^k - N_i^j \geq r_{ik}^j \, .$$

DEMONSTRATION

En effet, supposons $N_i^k - N_i^j < r_{ik}^j$, donc, d'après le théorème précédent on a : $C_{ik}^j > C_{ij}^k$. D'autre part, par hypothèse on a : $C_{ij}^k > C_{ik}^j$, ce qui est une contradiction, donc $N_i^k - N_i^j \geq r_{ik}^j$.

COROLLAIRE 2

$$\text{Si } N_i^j - N_i^k \geq 3 \cdot p_{ij} + p_{ik}, \text{ alors } C_{ij}^k \leq C_{ik}^j \, .$$

En effet : $3 \cdot p_{ij} + p_{ik} \geq p_{ij} \, (t_{ij} + \alpha_{ij}^k - \gamma_{ij}^k) + \delta_{ik}^j \cdot p_{ik}$, car :

$t_{ik} + \alpha_{ik}^j$ est évidemment au plus 4

γ_{ij}^k est au moins 1 (propriété 2)

δ_{ik}^j est au plus 1 (propriété 3), donc :

$N_i^j - N_i^k \geq 3 \cdot p_{ij} + p_{ik} \geq p_{ij} \, (t_{ij} + \alpha_{ij}^k - \gamma_{ij}^k) + \delta_{ik}^j \cdot p_{ik}$ et d'après le théorème précédent il résulte $C_{ij}^k \leq C_{ik}^j$.

REMARQUES

- si ℓ_{ij} est un fil dont le nombre minimum de coudes est zéro (fil "sans coudes"), il suffit d'avoir :

$$N_i^j - N_i^k \geq p_{ij} + p_{ik}$$

car, dans ce cas γ_{ij}^k est au moins 3 (remarque propriété 2), donc

$1 \geq t_{ij} + \alpha_{ij}^{k} - \gamma_{ij}^{k}$.

• si le poids de tous les fils est 1 (un seul fil par face de module), alors la condition du corollaire 2 devient :

$N_i^j - N_i^k > 3$ pour les fils ℓ_{ij} dont le minimum du nombre de coudes est 2

$N_i^j - N_i^k > 1$ pour les fils dont le minimum du nombre de coudes est zéro.

CONSEQUENCES.

(1) Soit le module i relié aux modules j, k, l, etc.... Supposons :

$$N_i^j \geq N_i^k + r_{ij}^k$$

$$N_i^j \geq N_i^l + r_{ij}^l, \text{ etc....}$$

Supposons que l'on minimise la liaison l_{ij}. D'après le théorème 1, le nombre des coudes du réseau augmente si l'on cherche à minimiser n'importe quelle autre liaison du module i. Donc i garde l'orientation correspondant à la minimisation de l_{ij}.

(2) Soient les relations suivantes :

$$(a) \quad N_i^j < N_i^k + r_{ij}^k$$

$$(b) \quad N_i^k < N_i^l + r_{ik}^l$$

En vertu du théorème 1, on a :

$$C_{ij}^k > C_{ik}^j$$

$$C_{ik}^l > C_{il}^k$$

Le nombre minimum des coudes du réseau correspond à $\min(C_{ik}^j, C_{il}^k)$. On remarque le fait suivant :

C_{ik}^j : correspond à la minimisation de l_{ik}, et l_{ij} contenant $\delta_{ij}^k \cdot p_{ij}$ coudes de plus.

C_{il}^k : correspond à la minimisation de l_{ik} et l_{il} contenant $\delta_{il}^k \cdot p_{il}$ coudes de plus.

Donc C_{ik}^j diffère de C_{il}^k par au plus $p_{ij} + p_{il}$ coudes, δ_{ij}^k, δ_{il}^k étant au plus 1 et au moins -1.

Donc, avec une erreur d'au plus $p_{ij} + p_{il}$ coudes on peut orienter le module i pour la minimisation de l_{il}.

II.4.3. Algorithme

A chaque pas de l'algorithme on doit s'assurer que le nombre de coudes du réseau n'augmente pas.

Au $i^{ème}$ pas (ou au $i^{ème}$ module) on procède de la manière suivante :

(1) On partage les modules reliés à i en deux classes :

- la classe des modules fixes (marqués) : F_i
- la classe des modules à tourner librement (non marqués) : L_i

et pour chaque liaison l_{ij} de i on calcule N_i^j en tournant à la fois i et j si $j \in L_i$, ou en tournant seulement i si $j \in F_i$, de façon à minimiser le nombre de coudes de l_{ij}.

(2) On retient les N_i^j tels que : $N_i^j > 0$ et $N_{ij} \geq 0$ (les réductions du nombre de coudes du réseau).

(3) On construit le vecteur :

$$W = (N_i^j, N_i^k, N_i^l, \ldots)$$

des valeurs retenues précédemment, ordonné par ordre décroissant de ses composantes.

(4) On considère (N_i^j, N_i^k) et on calcule $3 \cdot p_{ij} + p_{ik}$.

Si $N_j^i - N_i^k \geq 3 \cdot p_{ij} + p_{ik}$ alors on retient N_i^j et on considère (N_i^j, N_i^l), etc....

Si $N_i^j - N_i^k < 3\ p_{ij} + p_{ik}$ on calcule r_{ij}^k.

Si $N_i^j - N_i^k \geq r_{ij}^k$ on considère (N_i^j, N_i^l), etc....

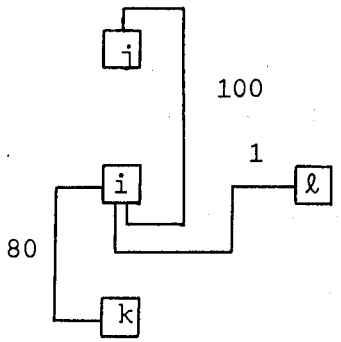
sinon on retient N_i^k et on considère (N_i^k, N_i^l), etc....

On fait l'orientation de i suivant la dernière composante retenue du vecteur W et on marque i.

L'algorithme s'arrête quand tous les modules sont marqués.

EXEMPLE :

On reprend le réseau donné en début du chapitre :



Supposons que l'on commence par le module i. En appliquant l'algorithme, on fait les opérations suivantes :

(1) $F_i = \emptyset$, $L_i = \{j, k, l\}$

$N_i^j = 4x100 - 2x80 = 240$, $N_{ij} = 0$

$N_i^k = 2x80 + 100 + 3 = 263$, $N_{ik} = 0$

$N_i^l = 3x1 + 100 - 2x80 = -57$, $N_{il} = 0$

(2) on retient N_i^j et N_i^k.

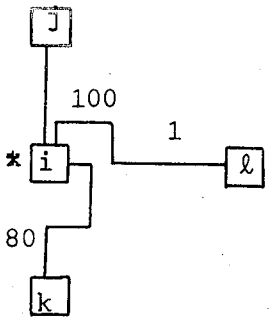
(3) $W = (N_i^k, N_i^j)$

(4) $3 \cdot p_{ik} + p_{ij} = 340$

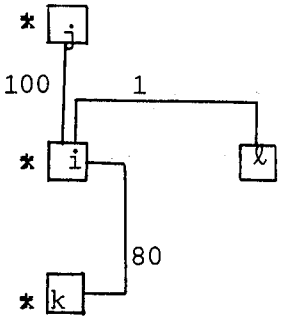
$N_i^k - N_i^j = 23$, donc $N_i^k - N_i^j < 3 \cdot p_{ik} + p_{ij}$. On calcule r_{ik}^j

$r_{ik}^j = p_{ik} (t_{ik} + \alpha_{ik}^j - \gamma_{ik}^j) + \delta_{ij}^k \cdot p_{ij} = 80 (2 + 2 - 3) + 0 \times 100 =$

$N_i^k - N_i^j < r_{ik}^j$, donc on retient N_i^j, et on fait l'orientation suivant la minimisation de l_{ij}, en marquant i.



En continuant pour les autres modules non marqués, on obtient facilement la configuration suivante :



dont le nombre de coudes est plus petit que celle si l'on avait minimisé l_{ik}, dont $N_i^k > N_i^j$.

II.4.4. Calcul de α, γ, δ

Sur l'ensemble F = {N, E, S, O} de faces d'un module on définit les permutations suivantes :

g_1 :				d_1 :				g_2 :			
N	E	S	O	N	E	S	O	N	E	S	O
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
O	N	E	S	E	S	O	N	S	N	O	E

a) Les coefficients α sont calculés à l'aide des tableaux H/V et D.

Soit à calculer α_{ij}^k : nombre de coudes introduits dans la liaison l_{ij} si on minimise l_{ik}. On note par c_{ij} le nombre de coudes de l_{ij}. Si f_i est la face du module i reliée à la face f_j du module j, alors :

$$C_{ij} = T(f_i, f_j), \quad \text{où} \quad T \in \{H/V, D\}.$$

On cherche r_i, $r_k \in \{g_1, d_1, g_2\}$ appliqués respectivement aux faces de i, k, tels que :

$T(r_i(f_i'), r_k(f_k))$ soit minimum, où f_i', f_k sont les faces de i, k reliées par l_{ik}.

On calcule :

$C'_{ij} = T\ (r_i\ (f_i),\ f_j)$, et on obtient :

$\alpha^k_{ij} = C'_{ij} - C_{ij}$.

b) Le calcul de γ est plus rapide. Soit à calculer γ^k_{ij} : nombre de coud introduits dans l_{ij} si elle était minimisée et on minimise l_{ik}.

- Si le module k $\in$ {VH, VB, VG, VD} alors :

  • si $r_i = g_2$ (en cherchant comme précédemment r_i, r_k), alors $\gamma^k_{ij} = 4$
  • $\gamma^k_{ij} = 3$, autrement.

- Si le module k $\notin$ {VH, VB, VG, VD}, alors :

  • si $r_i = g_2$, alors $\gamma^k_{ij} = 2$
  • $\gamma^k_{ij} = 1$ autrement.

C) Les coefficients δ sont calculés aussi à l'aide des tableaux H/V, D.

Soit à calculer δ_{ij} : nombre de coudes introduits dans la liaison l_i si on tourne j seulement, pour la minimisation de l_{ij}. On note par C_{ij} nombre de coudes de l_{ij}, initial. On cherche r_i, $r_j \in \{g_1, g_2, d_1\}$ appliqués à i, j respectivement, tels que :

$T\ (r_i\ (f_i),\ r_j\ (f_j)$ soit minimum, où $T \in \{H/V, D\}$

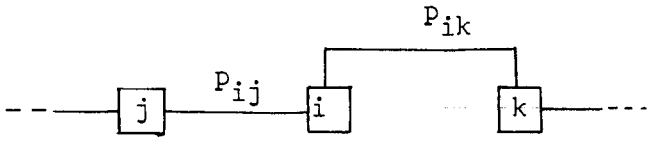
et f_i, f_j sont les faces de i, j reliées par l_{ij}. Alors :

$C'_{ij} = T\ (f_i,\ r_j\ (f_j))$ et

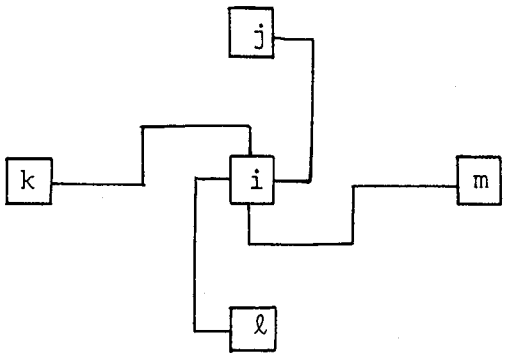
$\delta_{ij} = C'_{ij} - C_{ij}$

REMARQUES SUR L'ALGORITHME.

1) - L'algorithme donne évidemment la solution optimale si le nombre de liaisons (à poids) incidentes à un module n'est pas plus grand que deux :



2) - On peut augmenter les performances de l'algorithme en appliquant d'abord l'algorithme de construction de la configuration initiale, décrit précédemment (II.3.), qui résoud facilement des cas particuliers, comme ci-dessous :



II.4.5. Estimation du nombre de calculs

Le nombre d'opérations de l'algorithme est maximum pour un réseau complet. Supposons les n modules ponctuels et le graphe complet associé. Si par "opération" on désigne le calcul du nombre de coudes d'une liaison, on a :

- n éléments (modules) à étudier
- n-1 liaisons par élément
- n-2 liaisons à étudier pour chaque liaison d'un élément.

Au total on a : n(n-1) (n-2) "opérations" à faire. En supposant qu'il y a C liaisons différentes (à poids différent) par module ($C \leq 4$, en pratique) on a : C•n•(n-1)•(n-2) "opérations", donc le nombre de calculs est de l'ordre de grandeur de n^3.

CHAPITRE III

REALISATION DES EQUIPOTENTIELLES

Les équipotentielles ayant été déterminées, leur réalisation effective dépend étroitement de la technologie utilisée. Généralement, les liaisons sont matérialisées par pistes de cuivre imprimé, ce qui interdit les croisements des connexions sur une même surface. Dans ce cas, le problème des croisements devient un facteur décisif dans la réalisation de la plaquette ; actuellement, pour la plupart des algorithmes de tracé par circuit imprimé, le pourcentage de réussite du tracé de toutes les connexions est rarement de 100 %.

Afin d'améliorer les performances des algorithmes du tracé, les constructeurs utilisent des plaquettes à deux (bicouche) ou plusieurs couches (multicouches).

III.1. CRITIQUE DES ALGORITHMES DE TRACE CONNUS

Il est quasi-impossible de trouver un algorithme donnant une solution quelque soit la technologie utilisée [29], surtout si le nombre de modules et le nombre de liaisons sont grands. D'autre part, pour une technologie donnée la recherche d'une solution optimale étant trop coûteuse, on utilise divers algorithmes qui donnent des solutions satisfaisantes.

Le grand nombre des algorithmes de tracé est dû à la grande variété de technologies utilisées qui entraînent des contraintes variées pour la réalisation des connexions.

III.1.1. Suivant le type de plaquette utilisé on peut les classer en :

a) - Algorithmes pour plaquettes bicouche [30].

Ces algorithmes adoptent des politiquent différentes :

- soit essayer de tracer le plus de connexions possibles sur une couche, l'autre couche étant réservée aux connexions ne pouvant être placées sur la première,

- soit tracer la connexion sur une couche, puis sur l'autre si une contrair ne peut être respectée et ainsi pour toutes les connexions.

Si la première politique a le désavantage de surcharger la première couche par rapport à l'autre, la deuxième politique présente l'inconvénient de trous de traversée trop nombreux.

b) - Algorithmes pour plaquettes multicouche [29, 31].

Ont pour principe :

- soit appliquer la première politique des algorithmes précédents si le nombre de connexions et très grand,

- soit créer des classes de fils suivant leur tendance directionnelle et affecter à chaque classe une couche ; c'est la meilleure politique.

c) - Algorithmes à "blocs denses" [31].

La densité des boîtiers sur la plaquette est élevée et, donc, la surface disponible au tracé des connexions est réduite. Le tracé de la connexion consiste à trouver des canaux disponibles parmi les boîtiers en respectant, aussi, certaines contraintes : distance minimale entre deux fils, minimiser le nombre de croisements, etc....

III.1.2. Suivant le principe du tracé d'une connexion :

On peut les classer en :

a) - Algorithmes de tracé global [32] :

Le traitement de toutes les connexions est fait en même temps. Un ensemble de critères étant choisi, la construction des liaisons évolue à chaque itération.

Les méthodes présentent l'avantage de ne pas traiter un fil comme s'il était seul sur la plaquette ; mais il est difficile de définir sa rapidité de convergence, chaque itération n'améliorant pas forcément le tracé. De même, en cas d'échec on n'a pas de solution partielle.

b) - Algorithmes de tracé fil à fil :

Ce sont les algorithmes les plus utilisés actuellement. La méthode consiste à tracer isolement chaque fil, qui devient après un obstacle pour les autres. Elle présente les avantages suivants [30] :

- modularité du programme du tracé
- extension facile du programme du tracé
- donne toujours une solution (partielle en cas d'échec)
- contraintes faciles à respecter
- le tracé est esthétique
- offre la possibilité d'intervention humaine au cours du tracé.

Pour notre problème particulier nous utilisons un tel type d'algorithme pour le tracé des connexions.

Les algorithmes de tracé fil à fil doivent résoudre les deux problèmes suivants :

(1) Choisir l'ordre dans lequel sont tracées les connexions.

Bien que L.C. Abel [33] montre expérimentalement, pour un algorithme particulier (algorithme de Lee [34]), que la longueur totale moyenne des liaisons ne dépend pas de l'ordre dans lequel sont tracées les connexions (ordre suivant les longueurs des connexions ou les longueurs de leurs projections sur horizontale et verticale), le problème d'ordonnacement des connexions garde une importance particulière. Par exemple, le fait de tracer les fils courts d'abord a pour effet d'augmenter la longueur des fils longs et le cheminement de ceux-ci est plus sinueux. Mais, d'autre part, pour un tracé multicouche, si on groupe les connexions en classes de tendance directionnelle, en affectant à chaque classe une couche, on a intérêt à tracer les fils courts d'abord, car ils n'ont pas une grande influence sur la direction du tracé de la couche.

Il n'existe pas encore de méthode rigoureuse pour déterminer l'ordre du tracé des connexions ; le problème est résolu de façon approximative pour chaque cas particulier. Un remède serait la méthode dite de "l'arbre dynamique" [29] : l'impossibilité de réaliser entièrement une équipotentielle pourrait être levée pour un choix différent de l'arbre initial.

(2) Trouver une méthode de liaison entre deux points (tracé du chemin de la connexion), compte-tenu des diverses contraintes :

- on n'autorise que certaines directions des liaisons
- interdire ou minimiser les croisements
- minimiser la longueur des connexions
- minimiser le nombre de coudes des connexions, etc....

Il existe une grande variété d'algorithmes pour le tracé d'une connexion. On peut les grouper en deux classes :

(a) algorithmes à discrétisation de la plaquette

(b) algorithmes utilisant des segments.

Discrétiser la plaquette consiste à diviser sa surface en "cellules" par un réseau de droites horizontales et verticales et de décrire l'état de chaque cellule : libre, occupée, trou, fil, isolant, etc....

Nous donnons rapidement un aperçu des principaux algorithmes de type (a) et (b) en mettant en évidence leurs qualités et leurs défauts.

III.1.3. Algorithmes à discrétisation de la plaquette

Ce sont les premiers algorithmes appliqués au tracé des connexions par circuit imprimé : l'idée d'utiliser un quadrillage du plan vient naturellement à l'esprit, d'une part du fait de l'emploi du calculateur, d'autre part afin de pouvoir respecter les contraintes de distance portants sur les fils.

L'algorithme de base est le plus souvent l'algorithme de Lee [34], remontant à 1961.

Majorani [35] a donné une variante simplifiée de cet algorithme. L'algorithme de LEE travaille avec deux listes : L_1, L_2. Le processus de recherche du chemin optimal (longueur minimale, nombre de croisements minimum, etc...) peut être exprimé de façon imagée de la manière suivante : on simule le développement d'une onde à partir de la cellule de départ (contenue initialement dans L_1). La progression de l'onde est obtenue en ajoutant à la liste L_2 chaque nouvelle cellule des voisinages des cellules contenues dans L_1 et en attribuant une "coordonnée" (qui permet de retrouver le chemin déjà construit) aux cellules qui satisfont les contraintes imposées. On transfère ensuite les cellules de L_2 dans L_1 et on vide L_2. Le processus continue jusqu'à ce que la cellule "arrivée" apparaît dans L_2, auquel cas on a trouvé la solution, ou L_2 est vide auquel cas il n'existe pas de solution.

L'algorithme présente les qualités suivantes :

- s'il existe des solutions il donne la "meilleure" (chemin optimal)
- il est simple et se prête bien à la programmation.

Mais, le découpage de la surface de la plaquette en cellules, nécessite un certain nombre de bits par case, d'où un gros encombrement de mémoire, même quand les problèmes à traiter sont simples. D'autre part, la recherche s'effectue cellule par cellule, d'où un temps de calcul impor tant ; pour joindre un point situé en haut à droite de la plaquette à un point situé en bas à gauche, il faut explorer toute la plaquette.

Il existe des variantes de l'algorithme de Lee qui cherchent à le rendre plus rapide, mais au détriment de son efficacité :

- algorithme de Lee avec Shunt : le Shunt est un paramètre qui fixe la "largeur" du front de l'onde, ce qui donne à l'algorithme une directivit plus ou moins marquée selon la valeur du shunt.
- algorithme de "tir" [36] : l'avancement se fait cellule par cellule suivant une de 8 directions possibles, vers la cellule d'arrivée.
- algorithme de "visée" à retour en "épi" [29] : cherche à regrouper les qualités de l'algorithme de Lee avec Shunt (chemin proche de l'optimum) et de l'algorithme de "tir" (rapidité), en effectuant un "lissage automatique" par un développement en épi, à la rencontre d'un obstacle.

Toutes ces variantes sont plus rapides en temps d'exécution, mais le chemin trouvé est plus long, plus sinueux et elles ne le trouvent pas toujours.

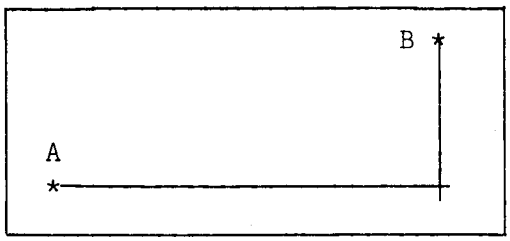
Chein (37), en utilisant aussi un quadrillage du plan, trouve le tracé pour chaque connexion en minimisant le nombre de croisements et la longueur de chaque équipotentielle. Du graphe de croisements il extrait un arbre minimal auquel il associe un graphe de liaisons à longueur minimale.

L'algorithme présente l'avantage (théorique) et l'inconvénient (pratique) d'examiner toutes les configurations possibles. La meilleure solution est obtenue au prix de calculs longs et nombreux (par exemple, pour la construction des graphes de croisements et de liaisons il faut exa miner plusieurs fois les cases du quadrillage, ce qui prend un temps de calcul important).

III.1.4. Algorithmes utilisant des segments [38, 39].

La recherche du chemin ne se fait plus cellule par cellule, mais par chaînes rectilignes de cellules, appelées "segments".

Dans des cas simples, ces algorithmes présentent un net avantage sur ceux de discrétisation de la plaquette ; par exemple, pour les points les plus éloignés de la plaquette, Lee doit effectuer l'exploration de toute la plaquette, tandis que la recherche par segments donne immédiatement la solution :



Il s'ensuit un gain de temps considérable et le codage des chemins étant fait par segments (au lieu de cellules) il résulte une importante économie de mémoire.

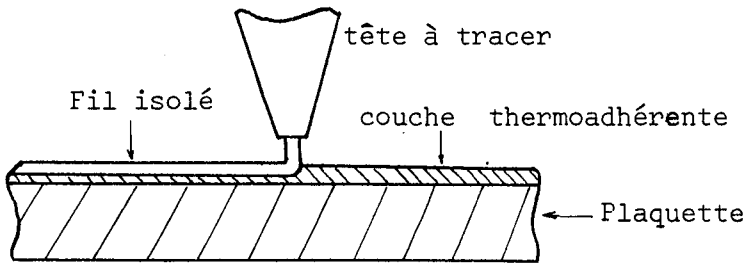
III.1.5. Conclusion

Les algorithmes à discrétisation de la plaquette nécessitent généralement des temps de calcul importants et demandent beaucoup de mémoire du calculateur pour garder l'image de toute la plaquette. De plus, ils ne garantissent pas le tracé de toutes les équipotentielles.

Nous avons préféré en fin de compte utiliser une méthode heuristique de tracé par segments qui a l'avantage d'être plus économique au point de vue temps d'exécution et encombrement mémoire.

III.2. REALISATION DES CONNEXIONS PAR FILS ISOLES [40]

Les connexions entre "puces" de semi-conducteurs sont réalisées au moyen de fils de cuivre isolés, de faible diamètre (30 microns). Le fil est enrobé d'une couche thermoadhérante et est collé, par l'intermédiaire d'une tête (plume) à tracer, à un support revêtu d'une autre couche thermo adhérante :



Le support est placé sur une table à commande programmée. Le pas de déplacement de la table est très faible (5 microns). Les commandes de déplacement se trouvent sur une bande perforée et lues soit une par une (l'ordre de passage au bloc suivant est donné par l'opérateur) soit de façon automatique, du début à la fin de la bande.

La structure du bloc de commande est la suivante :

X
N, Y, SIGNE de X et Y, Incrément, Fonction spéciale.
D

N : numéro du bloc

X, Y, D : indique le déplacement : suivant OX, OY ou diagonal (45°)

SIGNE de X et Y : donne le sens du déplacement

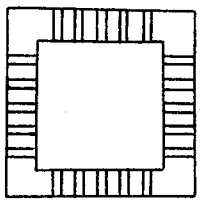
Incrément : coordonnée relative du nouveau point par rapport au précédent.

Fonction spéciale : est exécuté après le déplacement et on peut avoir arrêt, dénudage du fil, lever la plume, baisser la plume, rien.

Les fils étant isolés peuvent se croiser autant de fois qu'il est nécessaire. On peut placer les fils très près les uns des autres, à un pas voisin de leur diamètre, et faire, ainsi, des réseaux très denses.

Contact entre le fil et les pattes des circuits intégrés

L'intérêt d'une méthode d'interconnexion par fil dépend beaucoup de la façon dont on peut réaliser les contacts. Actuellement, il est assuré par un petit cadre (0,5 cm x 0,5 cm), comportant des rainures de positionnement des fils conducteurs sur ses quatre côtés :



Le nombre des rainures d'un cadre varie selon le nombre des points de connexion du module semiconducteur, qu'il recevra à son centre. Le cadre est immobilisé sur la plaquette, grâce à la couche thermoadhérante, qui la recouvre.

On distingue trois étapes dans le processus de réalisation de la plaquette :

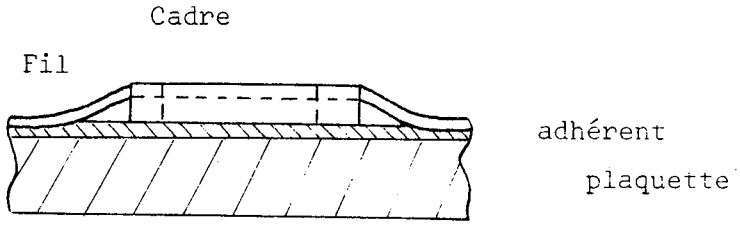
1) - Le tracé par déroulement de fil -

L'origine du tracé est repérée par une croix sur la plaquette et la tête est positionnée au centre de cette croix. A partir de là, on exécute le tracé avec du fil isolé, suivants les incréments imposés par les coordonnées des connexions. Le fil est dénudé thermiquement, si cela est nécessaire, au passage dans les rainures des cadres. Les cadres ayant une certaine hauteur, la tête est levée avant la traversée du cadre et abaissée après. Comme il n'existe pas encore la possibilité de couper le fil au cours du tracé, il sera réalisé par le déroulement d'un seul fil ; donc, pour commencer une nouvelle équipotentielle, on doit, en principe, sortir en dehors de la plaquette et rentrer à un certain point de l'équipotentielle. A la suite on verra que ce n'est pas toujours nécessaire de quitter la plaquette, car on peut employer des rainures non utilisées des cadres.

Dans le tracé des connexions on doit tenir compte des contraintes suivantes :

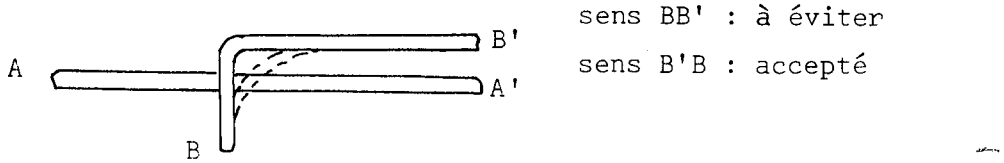
• changements de direction au passage du cadre sont interdits, car par la levée de la tête le fil n'adhère pas à la plaquette.

• les changements de direction trop près du cadre (moins de 2mm) sont interdits, car le fil, après la baissée de la tête n'adhère pas encore totalement à la plaquette :



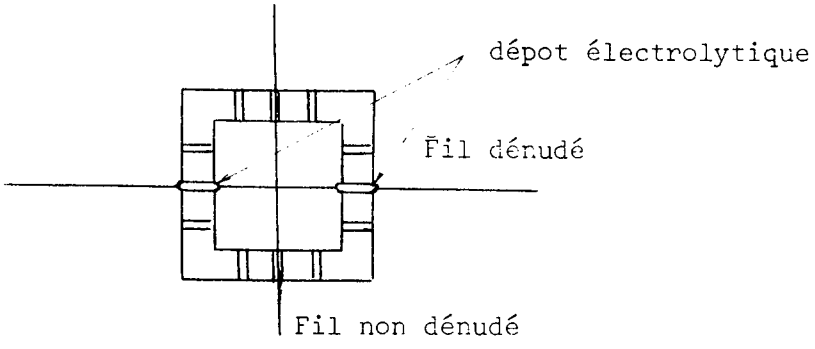
• le passage de plus de deux fils par rainure est interdit, à cause de la largeur trop petite de la rainure.

• éviter le changement de direction après le croisement d'un fil, car le fil n'étant pas bien collé, risque d'être tiré.



2) - Réalisation du contact électrique entre fils et cadres -

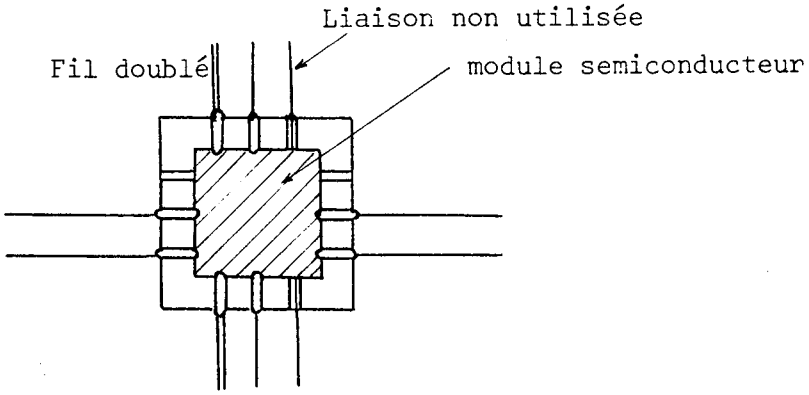
Les portions de fils dénudées sont raccordées collectivement aux cadres par croissance électrolytique :



3) - La mise ne place des modules semiconducteurs -

Après l'étape 2) on procède au découpage mécanique des fils se trouvant au centre des cadres. On coupe aussi les fils qui dépassent les dimensions de la plaquette.

On introduit les modules semiconducteurs à l'intérieur des cadres et on réalise les contacts avec les fils, un par un ou collectivement, par soudure.

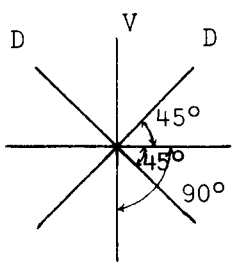


En utilisant un fil de liaison très fin, (ce qui permet l'élimination des "boitiers" englobants des modules, qui assuraient l'espacement des pattes de connexion), on arrive à des grandes réductions des dimensions des plaquettes qui existent actuellement dans les calculateurs, ou à des hautes densités de modules de circuits intégrés sur des plaquettes de même dimension.

III.3. - TRACE DES CONNEXIONS

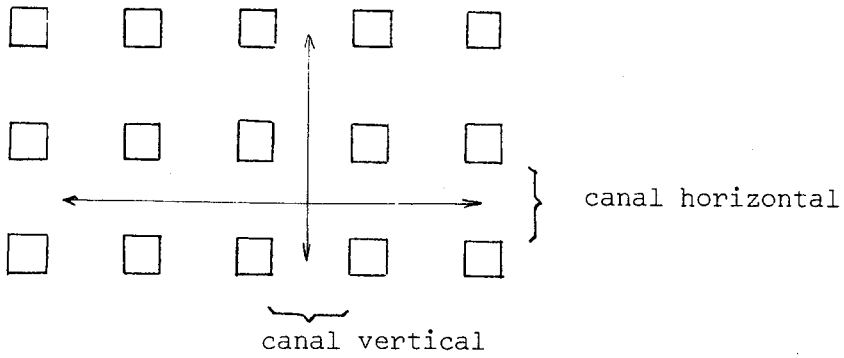
III.3.0. Justification du choix du tracé en horizontal et vertical

Par la technique du tracé des connexions, présentée précédemment, on a la possibilité de tracer des fils suivant deux directions perpendiculaires (horizontal et vertical) et une direction à 45° par rapport à deux précédentes (diagonal) :

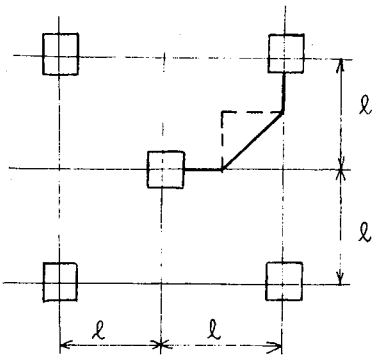


Pour le tracé effectif des connexions, nous avons choisi les deux directions H et V pour les raisons suivantes :

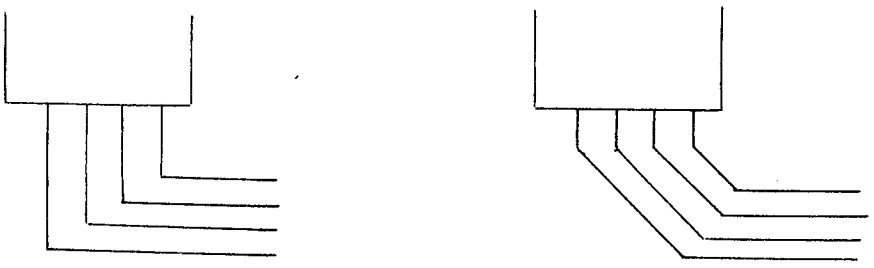
- l'implantation géométrique des modules sur la plaquette définit des canaux (espaces libres) horizontaux et verticaux :



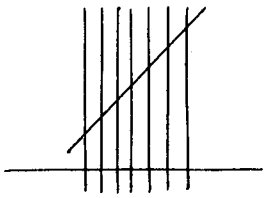
Pratiquement, la direction diagonale, n'est efficace que pour le tracé des fils qui relient deux modules voisins, dont les emplacements se trouvent sur les bissectrices ; cela suppose, donc, un espacement régulier des lignes et des colonnes de la matrice des emplacements physique



- Généralement, le tracé en diagonal, même s'il assure une certaine économi d'espace, il fait augmenter le nombre de coudes des fils et, aussi, le nombre de segments à mémoriser :



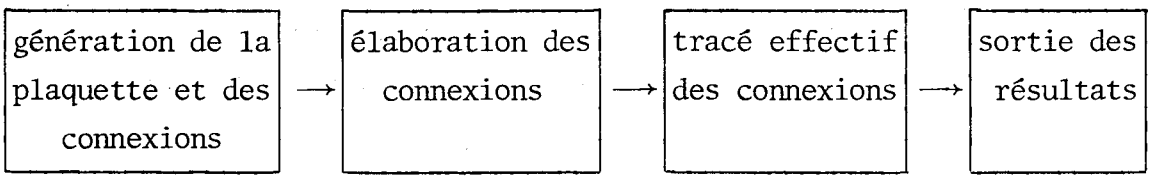
• l'adhérence des fils diagonaux est moins bonne à la traversée d'un paquet de fils horizontaux et verticaux :



• le tracé est uniforme, donc plus explicite et aussi esthétique.

Organigramme général du programme de tracé

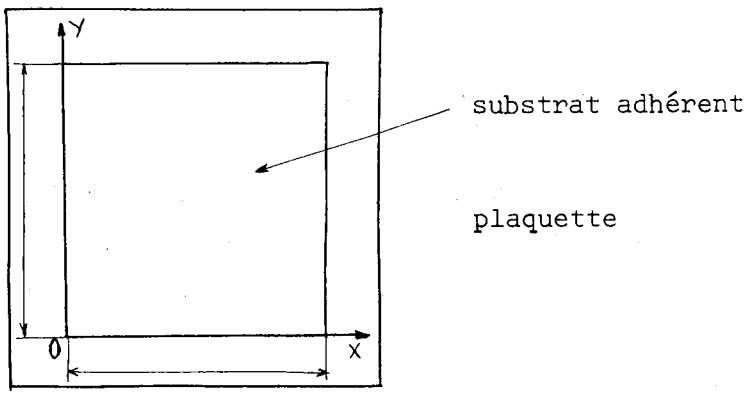
La chaîne des programmes qui réalisent le tracé des connexions est la suivante :



III.3.1. Génération de la plaquette et introduction des données

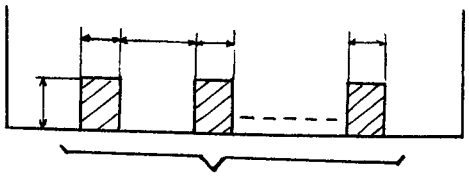
Il s'agit d'avoir en mémoire du calculateur une image réelle de la plaquette et de ses modules. Pour cela, on doit avoir :

• les dimensions de la plaquette et l'origine du système d'axes des coordonnées. Généralement, l'origine est au coin gauche, en bas, de la plaquette :



• le nombre des connecteurs et leurs emplacements sur la plaquette. Dans la description du connecteur on doit spécifier :

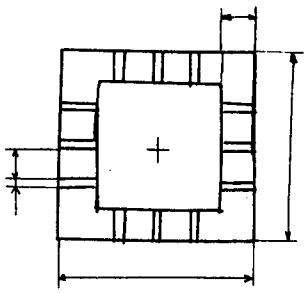
  . le nombre des plots
  . le pas d'espacement des plots
  . les dimensions des plots.



connecteur

• les données relatives aux cadres, c'est-à-dire :

  . les coordonnées du centre du cadre
  . les dimensions du cadre et son épaisseur
  . le nombre des rainures de chaque face
  . la largeur de la rainure et son pas d'espacement.



En ce qui concerne la connexion il faut spécifier :

  . les modules qu'elle relie
  . les faces des modules
  . les numéros des pattes dans chaque face.

Introduction des données

Un langage de description très simple, va permettre une analyse syntaxique et détecter, aussi, certaines erreurs de perforation des cartes. Les données sont entrées au moyen de plusieurs types de cartes, le premier caractère de la carte étant imposé et spécifiant la nature des données.

Comme la plupart des cadres possèdent les mêmes caractéristiques, on construit une bibliothèque des cadres qui contient toutes les données d'un certain type : dimensions, nombre des rainures des faces, largeur et pas d'espacement des rainures. Pour la description d'un cadre il suffira de donner les coordonnées et son type.

Les types de cartes des données sont les suivants :

- carte P, pour les données de la plaquette
- carte C, pour la caractérisation des connecteurs
- carte M, pour les données des cadres (modules)
- carte B, pour la construction de la bibliothèque des cadres
- carte T, pour l'introduction des données relatives aux connexions.

REMARQUE.

A la bibliothèque des cadres on ajoute, aussi, l'orientation du cadre après l'exécution du programme d'implantation.

III.3.2. Elaboration des connexions

Les segments horizontaux et verticaux sont représentés en machine par les coordonnées de leurs extrémités. La largeur du fil n'intervient pas, car :

- on utilise le même fil pour toutes les connexions
- même si deux segments ont des coordonnées identiques, par l'extension de la tête du traceur, le fil tracé le dernier, glisse à côté du fil tracé premièrement et adhère à la plaquette. Cependant, on cherchera à éviter cette situation.

Le problème posé à ce niveau est de trouver le chemin de chaque connexion, c'est-à-dire, les coordonnées des segments qui la constituent.

III.3.2.1. Contraintes du tracé

Dans la recherche des chemins des connexions on doit tenir compt des contraintes suivantes :

1) - tracer les connexions dans les zones libres de la plaquette, c'est-à-dire :

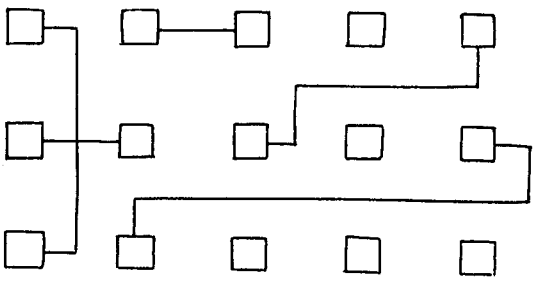
- le chemin d'une connexion ne doit pas traverser des cadres
- éviter de superposer des segments.

2) - au départ du cadre ou à l'arrivée au cadre la connexion doit être sans coudes, sur une distance de ℓ mm au moins (ℓ = 2 mm, en pratique).

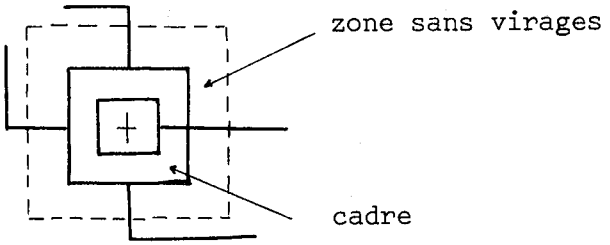
3) - éviter les virages interdits, c'est-à-dire les changements de directi tout de suite après le croisement d'un segment ; on doit parcourir au moins 1 mm (200 pas machine) après le croisement.

4) - minimiser le nombre de croisements.

Pour respecter la contrainte 1) les chemins sont définis par les canaux auxquels appartiennent les faces des modules qui sont reliées :

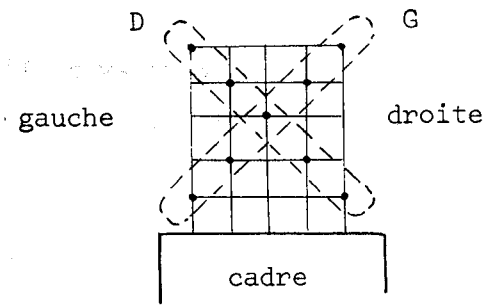


En raison de la contrainte 2), on définit une zone autour du cadre : "Zone sans virages". La direction du fil, définie par le contact au cadre, ne change pas à l'intérieur de cette zone :



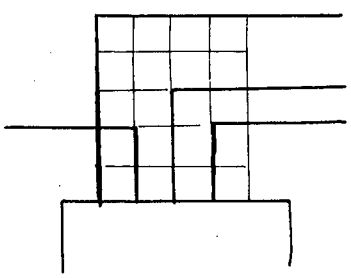
III.3.2.2. Virages interdits et croisements

a) - Cas des cadres voisins. Pour éviter les virages interdits au départ ou à l'arrivée des cadres et pour réduire le nombre des croisements, on adopte le procédé suivant : pour chaque paire de cadres, qui sont reliés, on applique à un des cadres, le départ ou l'arrivée en "escalier" des connexions, c'est-à-dire, selon le sens auquel le fil doit aller, il tourne à une certaine distance du cadre :



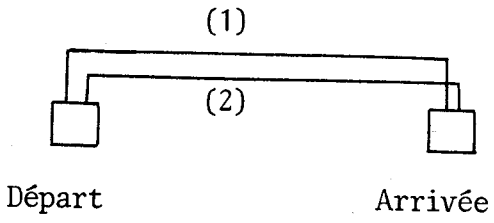
Pour une connexion allant à droite, le point où elle tourne appartient à D, et si elle tourne à gauche, le point correspondant appartient à G.

EXEMPLE :



Pour l'autre cadre de la paire, on distingue 2 cas :

1) si la connexion arrive au cadre sans autre coude supplémentaire (sauf coude correspondant au contact à la patte), le chemin est établi sans ten compte de la disposition en escalier :



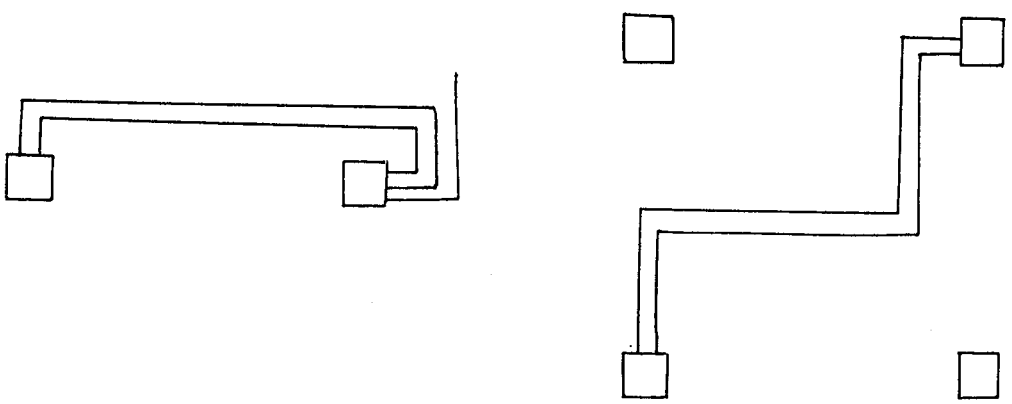
On remarque que, dans ce cas, on peut avoir un virage interdit s

. la connexion (1) est tracée avant la connexion (2), et

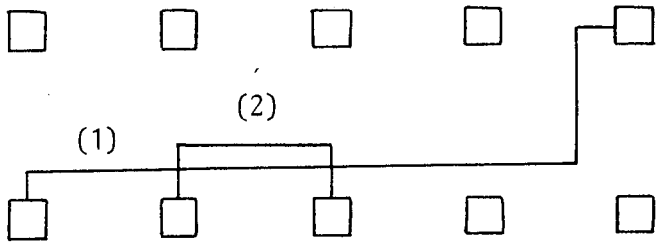
. la connexion (2) est tracée à partir du module "départ" vers le module "arrivée".

Dans les autres cas, le virage interdit disparaît

2) si la connexion possède un autre coude avant l'arrivée au cadre. On applique, aussi, pour ce deuxième cadre, le principe de liaisons en escalie



b) - Cas des cadres éloignés. Les connexions pour ces cadres sont tracées vers le centre des canaux pour éviter des cas comme celui-ci :



Si la connexion (1) est tracé en premier, on aura forcément un virage interdit pour la connexion (2).

III.3.2.3. Zones des canaux

On définit trois zones dans les canaux de la plaquette :

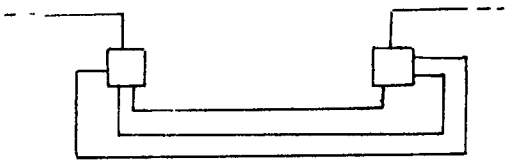
- une zone centrale destinée aux connexions qui relient des cadres éloignés
- deux zones latérales, d'un côté et de l'autre de la zone centrale, le tracé des connexions des cadres voisins.

Généralement, la zone centrale est réduite (0,5 mm), car il n'y a pas beaucoup de longues liaisons.

a) - Chemins des connexions dans la zone latérale

Les chemins des connexions dans cette zone seront disposés autant que possible en escalier. Si l'on considère 64 pattes par face de module (ce qui est le cas limite en pratique) et le fil à diamètre de 30 µ, on a la largeur maximale de la zone latérale de 64 x 30 = 1920 µ, soit 2 mm.

Pour établir les chemins des connexions dans la zone centrale on applique la règle suivante : les connexions à plus de coudes seront poussées vers la zone centrale; ainsi on réduit en quelque sorte le nombre de croisements :

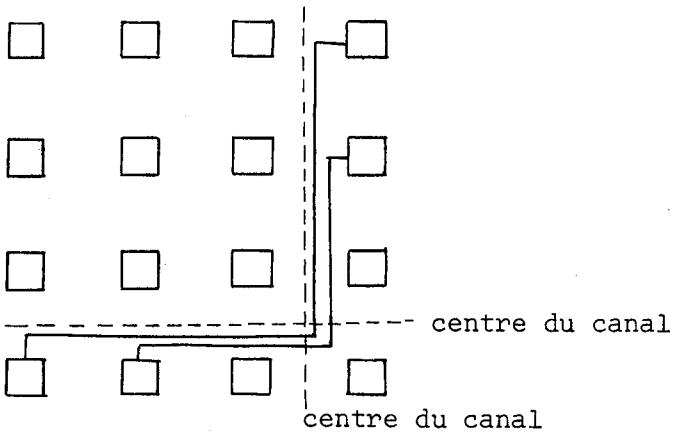


b) - Chemins des connexions dans la zone centrale

Les chemins dans cette zone sont établis d'après les règles suivantes :

• les connexions dont la longueur à l'intérieur du canal est plus grande plus près du centre de la zone par rapport à celles qui le parcourent moi (disposition en escalier suivant la longueur rectiligne),

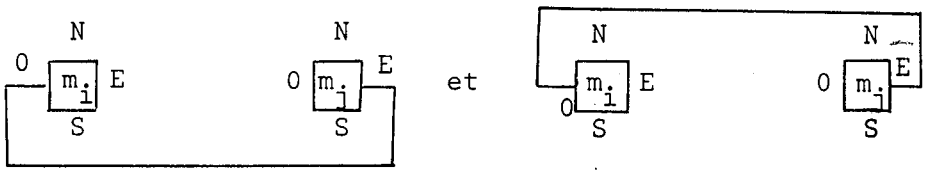
• une connexion de la zone centrale tourne toujours à l'intersection des zones centrales des canaux perpendiculaires.



III.3.2.4. Chemins équivalents

Deux chemins sont dits équivalents s'ils ont la même longueur e même nombre de coudes.

EXEMPLE :



Les chemins équivalents pour une même paire de modules sont les suivants :

• sur la verticale

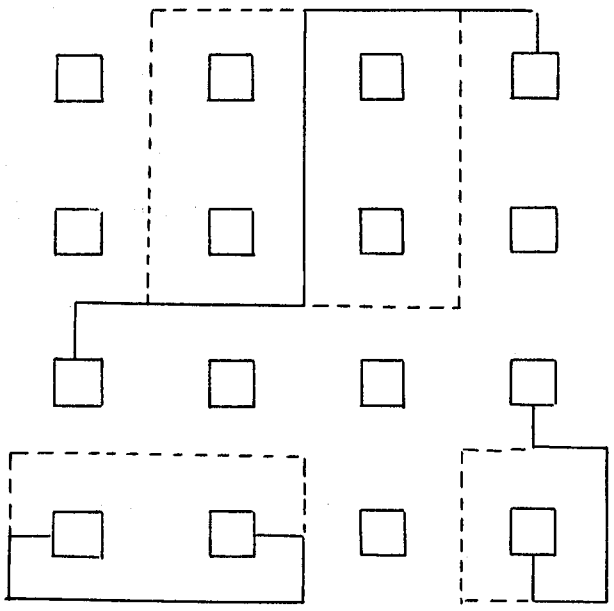
- modules voisins : liaisons des faces : N-N, S-S
- modules non voisins : liaisons des faces : N-N, N-S, S-N, S-S

• sur la verticale

- modules voisins : liaisons des faces : O-O, E-E
- modules non voisins : liaisons des faces : O-O, O-E, E-O, E-E

• modules éloignés, séparés par au moins une rangée de modules horizontale et une rangée de modules verticale : liaisons des faces : N-N, N-S, S-N, S-S, O-O, O-E, E-O, E-E.

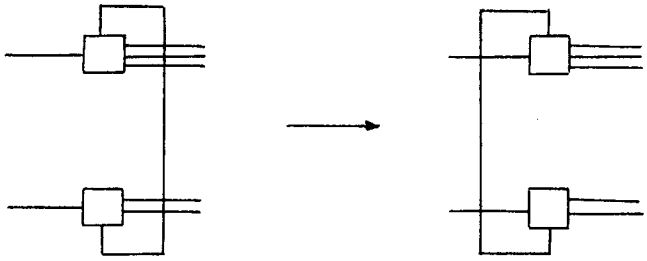
EXEMPLE :



En pointillé, chemins équivallents au chemin tracé

On cherche à réduire le nombre de croisements, en choisissant pa les chemins équivalents celui qui croise moins de connexions. Pour cela, le connexions à chemins équivalents sont élaborées à la fin, après les connexi dont le chemin est bien déterminé.

EXEMPLE :



5 croisements 2 croisements

REMARQUES

1) Pour les liaisons au connecteur, on utilise toujours les plots les plus près du module en question.

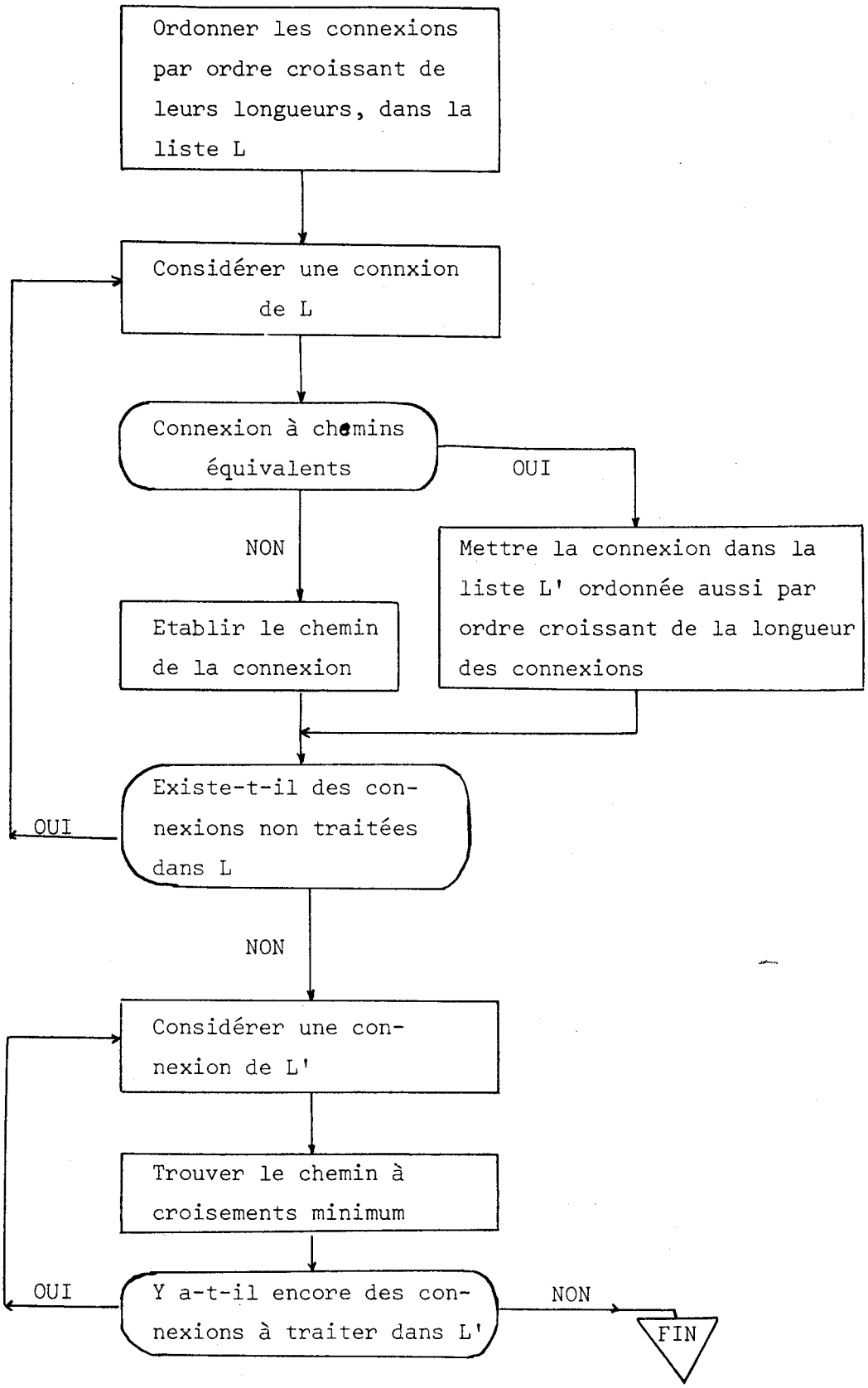
2) En tenant compte du nombre maximum de fils que peut recevoir un canal (il dépend de l'espacement de l'emplacement des modules, mais généralement il est plus grand de 150 en pratique) il faut vérifier à chaque connexion que la portion du canal qu'elle traverse n'est pas saturée.

Dans ce cas, on applique une procédure de recherche du chemin par des canaux non saturés; la longueur de la connexion augmente, dans ce cas, mais cela intervient rarement en pratique.

3) L'algorithme commence l'élaboration des connexions les plus courtes, don les canaux sont remplis de la zone latérale vers la zone centrale.

4) La délimitation des zones n'est pas stricte.

ORGANIGRAMME GENERAL



III.3.3. Tracé effectif des connexions

La difficulté majeure du tracé effectif des connexions provient de l'impossibilité de couper le fil; actuellement des études sont en cours pour l'utilisation d'un mince faisceau laser.

Soit G le graphe dont les sommets sont les pattes des modules et les arêtes les connexions elles-mêmes. Le problème du tracé des connexions peut se formuler de la manière suivante :

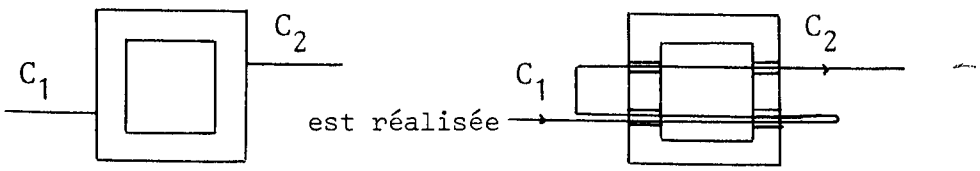
Trouver la chaîne hamiltonienne H à connexions supplémentaires, telle que la somme des liaisons ajoutées soit minimale.

Dans la construction de la chaîne H on a en vue les contraintes suivantes :

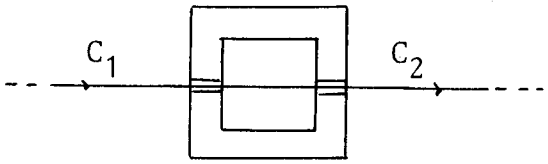
a) • ne pas changer de direction à la traversée du cadre; donc la traversé est faite par des rainures opposées des deux faces parallèles.

b) • ne pas passer plus de 2 fils par rainure (voir description tracé tech logique). Les deux contraintes ont pour conséquences les résultats suivants

- si la connexion C_1 vient d'être tracée et la rainure opposée n'est pas utilisée alors on sort du cadre par cette rainure et on peut revenir en ar rière par les mêmes rainures et tracer une autre connexion C_2, du même cad



- si C_1 vient d'être tracée et la rainure opposée est utilisée (extrêmité la connexion C_2), alors C_2 sera tracée obligatoirement à la suite de C_1, autrement on aura plus de 2 fils dans les rainures correspondantes :



De nature combinatoire, la taille du problème élimine dès le début les solutions de nature exhaustive. De plus, la solution optimale ne respecte pas toujours les contraintes de nature technologique et son tracé est impossible à réaliser. A défaut de méthode optimale il s'agit de trouver une méthode qui donne une solution satisfaisante. Dans cette optique il est possible de scinder le problème en deux parties :

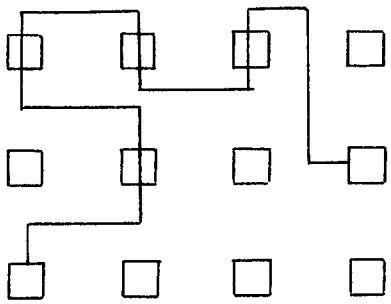
• recherche des chaînes de connexions

• liaisons des chaînes trouvées précédemment par des arêtes supplémentaires

III.3.3.1. Recherche des chaînes de connexions

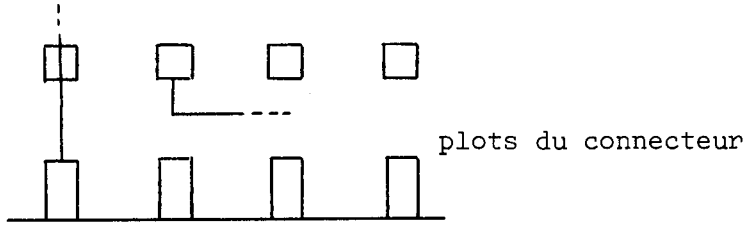
On définit la chaîne de connexions comme une suite de connexions qui mises dans un certain ordre peuvent être tracées ensemble, sans autre liaisons supplémentaires que celles des traversées des cadres :

EXEMPLE :



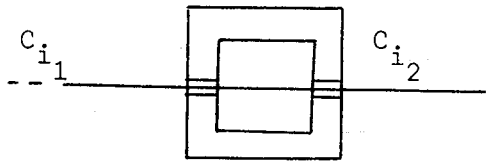
Une chaîne de connexions est construite de la manière suivante :

• on part d'une connexion C_{i_1}, dont une extrêmité est reliée au connecteur, ou, si les 2 extrêmités sont reliées aux cadres, la rainure opposée d'une des extrêmités n'est pas utilisée.



On calcule le nombre des coudes de C_{i_1} et on le met dans la liste L_i.

• on considère l'autre extrêmité de C_{i_1} et on ajoute à la liste L_i la connexion C_{i_2}, qui relie la patte opposée de la face opposée.



On calcule le nombre des coudes de C_{i_2} et on l'ajoute au nombre de coudes des connexions de L_i.

• on répète le processus tant qu'il existe des connexions telles que C_{i_2}.

Les chaînes de connexions, avec leur nombre de coudes C_i, sont obtenues en itérant la procédure précédente tant qu'il existe des connexio qui n'appartiennent pas aux listes L_i.

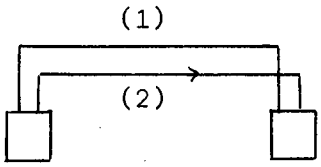
A la fin du processus on fait un classement des chaînes par l'or décroissant de leur nombre de coudes : L'_1, L'_2, L'_3,... tel que $C'_1 \geq C'_2 \geq$...

III.3.3.2. Liaisons supplémentaires

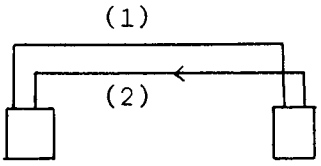
La chaîne finale est obtenue en reliant les chaines des listes L par des connexions supplémentaires, dans l'ordre de leur classement précédent.

Dans la construction de la connexion supplémentaire, on cherche à réduire le nombre des virages interdits. D'après une remarque précédente, on a vu que si une connexion présente un virage interdit quand elle est pa courue dans un sens, ce virage interdit disparaît si la connexion est parc rue dans l'autre sens :

(1) tracé premièrement



virage interdit pour (2)



pas de virage interdit

Pour chaque chaîne L'_i, où $i \geqslant 2$, on définira un sens de parcours qui correspond au nombre minimum de virages interdits de ses connexions, sachant que L'_1, L'_2,..., L'_j, où $j < i$ listes de connexions sont déjà tracées.

Ainsi orientée, la chaîne présente une extrêmité "début" et une extrêmité "fin". Dans la construction de la chaine finale H, on relie l'extrêmité "fin" de L'_{i-1} à l'extrêmité "début" de L'_i.

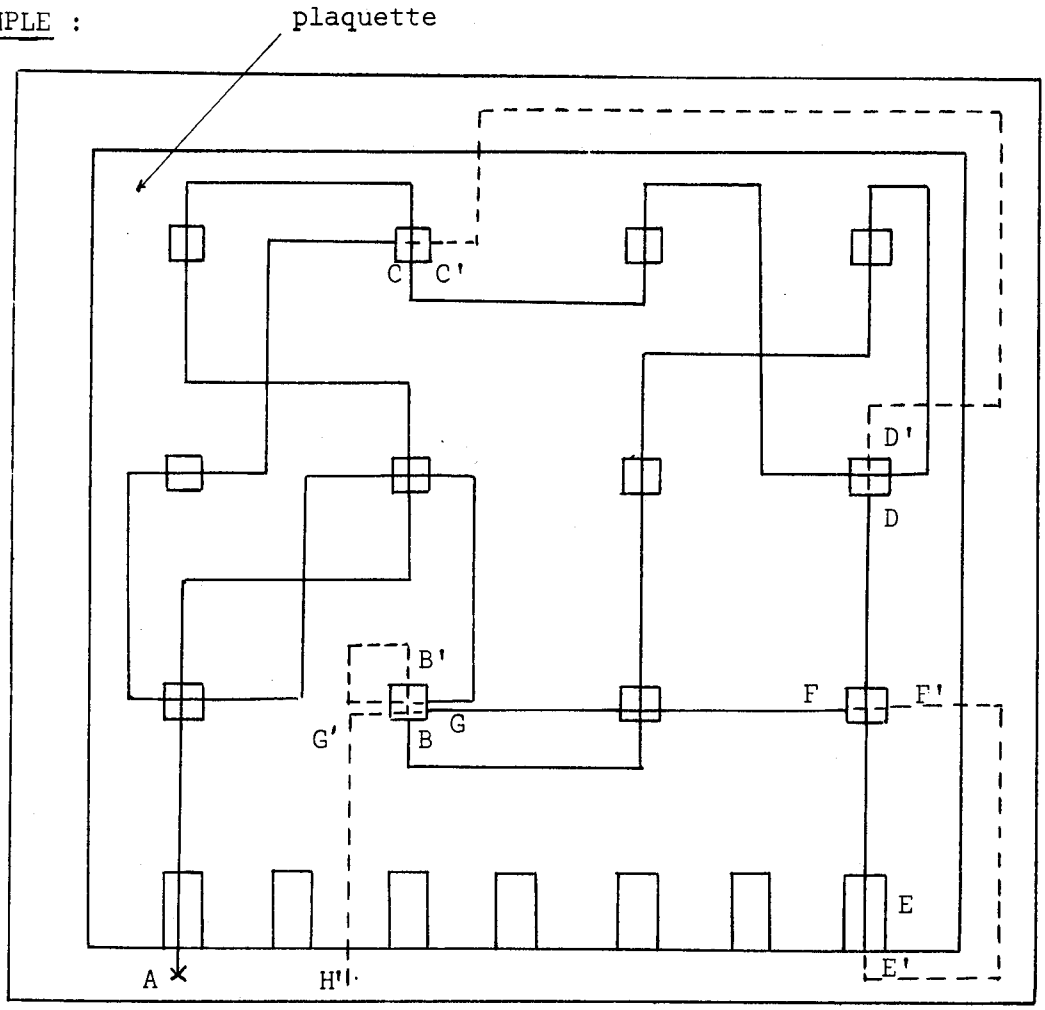
REMARQUES

1) La liaison supplémentaire est construite de façon qu'elle soit plus courte possible. Elle est définie aussi par les coordonnées de ses segments.

Les connexions supplémentaires très longues (longueur plus grande que la moitié du périmètre de la plaquette) sont tracées en dehors de la plaquette.

2) Dans l'établissement du chemin de la liaison supplémentaire, on peut se servir des rainures non utilisées de certains cadres.

EXEMPLE :



Les chaînes de connexions sont : AB, B'C, DE, FG, et les liaisons supplémentaires : BB', C'D', E'F', G'H'.

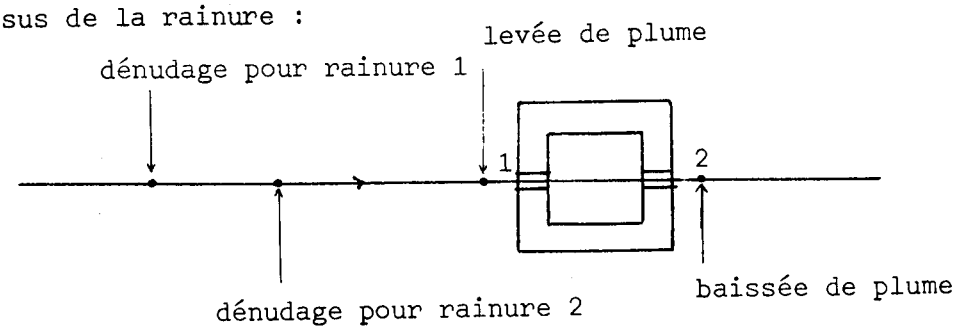
Les connexions supplémentaires qui se trouvent en dehors de la plaquette seront éliminées par le découpage de la plaquette; celles qui se trouvent à l'intérieur de la plaquette y resteront comme des fils inutiles (leurs extrêmités n'étant pas dénudées - voir paragraphe suivant -).

III.3.3.3. Génération des commandes supplémentaires

Il s'agit de reprendre la chaîne finale H et de générer les coordonnées des points des commandes suivantes :

- levée de plume à la traversée d'un cadre
- baissée de plume après la traversée du cadre

- dénudage du fil à l'endroit des rainures. Pour chaque traversée du cadre, si les deux rainures sont utilisées, il faut générer deux points correspondants aux commandes de dénudage. En raison du procédé technologique, la commande de dénudage est donnée avant l'arrivée de la tête du traceur, au dessus de la rainure :



Les distances des points définis plus haut, au cadre, sont les mêmes quelque soit le type du cadre.

A partir de la chaîne finale parcourue dans le sens de son tracé, on génère les blocs de commandes pour le tracé des connexions et pour les commandes supplémentaires qui seront perforés sur bande. A la lecture de cette bande, la machine à tracer, réalise automatiquement le tracé des connexions.

III.3.4. Sortie des résultats

Les blocs des commandes sont sortis sur listing sous la forme où ils se trouvent sur la bande perforée :

N°.bloc	X / Y / D	Signe	Incrément	Fonction Spéciale

Le tracé, avant d'être réalisé effectivement, est vérifié en le traçant à l'aide d'une table traçante (Benson).

CONCLUSION GENERALE

Les algorithmes qui viennent d'être décrits ont été elaborés à partir de préoccupations industrielles bien précisées. Cela explique pourquoi nous avons cherché des méthodes de résolution qui nous donnent rapidement et économiquement une bonne solution.

La confrontation de nos idées initiales avec l'expérience des algorithmes existants nous a conduit à abandonner certains principes; nous avons évité le dialogue homme-machine pour deux raisons :

• il est coûteux en temps de calcul, le temps de réponse de l'homme étant très long par rapport à celui du calculateur,

• la technique du tracé utilisée (fils isolés) ne nécessite pas l'intervention humaine; le calculateur trouve les chemins optimaux (le plus court, nombre de coudes minimum, nombre de croisements réduit) pour toutes les connexions.

Cependant, l'algorithme d'implantation est assez souple, le concepteur ayant la possibilité de guider l'implantation des modules, simplement par modifications des éléments de la matrice des liaisons et de la matrice des distances entre les emplacements de la plaquette. De plus, le choix de certaines options permet l'adaptation des programmes à des problèmes un peu différents (par exemple, emplacements distribués sur des lig et des colonnes, ou non régulièrement distribués sur la plaquette).

Les programmes sont écrits en PL/1 et ils forment une chaine de procédures appelées et exécutées à partir de l'implantation, jusqu'au trac des connexions sans intervention humaine.

ANNEXE 1

ALGORITHME DE RECHERCHE DES SOUS-RESEAUX MINIMAUX (17)

-o-o-o-o-

On utilise quatre listes : L_1, L_2, L_3, L_4. A la fin de l'algorithme L_4 contient tous les sous-réseaux minimaux, exceptés les sous-réseaux réduits à un seul module. L'algorithme est le suivant :

1) L_1 contient initialement tous les modules et L_2, L_3, L_4 sont vides.

2) On examine toutes les paires des éléments de L_1. Si on trouve une paire M, N, telle que K = MUN est un sous-réseau minimal, on introduit K dans L_2 et L_4 et on élimine M, N de la liste L_1 (K est minimal si k < m, n où k, m, n sont les nombres des terminaux de K, M, N respectivement).

3) Si L_2 est vide, aller à 8).

4) On examine toutes les paires de L_2 et les paires formées d'un élément de L_1 et un élément de L_2. Si une paire M,N est trouvée telle que K = MUN est un sous-réseau minimal, on introduit K dans L_3 et L_4 ET M,N sont éliminées de leurs listes d'origine.

5) L_2 est ajoutée aux éléments de L_1 et après vidée.

6) Si L_3 est vide, alors aller à 8).

7) L_3 est transférée dans L_2 et après vidée. Aller à 4).

8) Faire J = 2.

9) Si J ⩾ au nombre des éléments de L_1, aller à 13).

10) Faire J = J+1.

11) On examine tous les groupements de J éléments : $(M_1,\ldots, M_J)$ de L_1. Si $K = M_1 \cup M_2 \cup \ldots \cup M_J$ est un sous-réseau minimal ($k < m_1, m_2, \ldots m_J$), introduit K dans les listes L_2, L_4 et on élimine $M_1,\ldots, M_J$ de L_1.

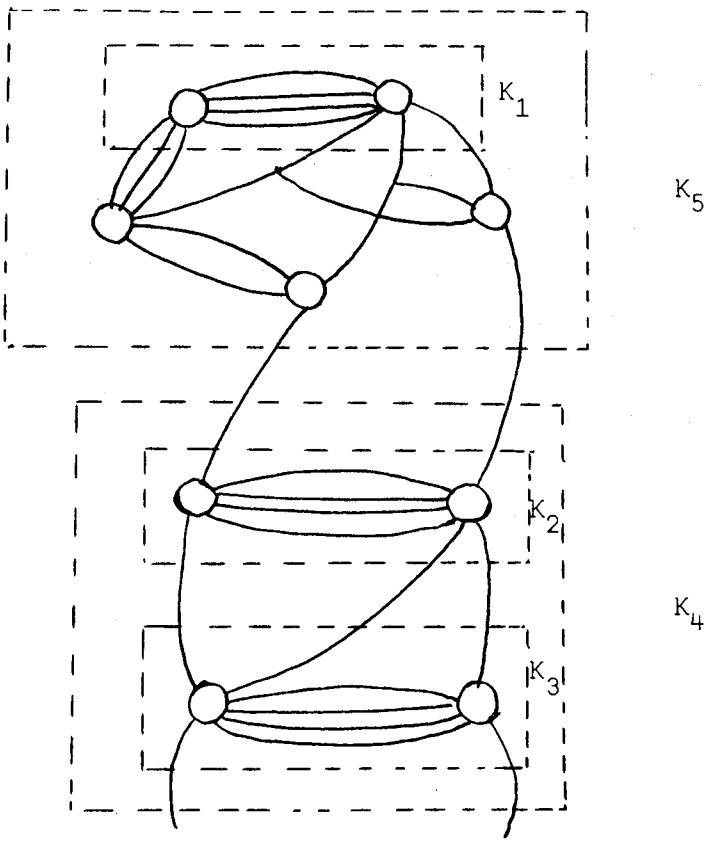
12) Si L_2 est vide alors aller à 9), sinon aller à 4).

13) Fin de l'algorithme.

REMARQUE

A la fin de l'algorithme, la liste L contient les sous-réseaux minimaux de plus grande taille, c'est à dire ceux qui ne sont pas inclus d'autres sous-réseaux minimaux.

En reprenant l'exemple des auteurs, par l'application de l'algorithme on obtient les sous-réseaux minimaux suivants :



ANNEXE 2

UN ALGORITHME POUR LE PROBLEME DU VOYAGEUR DE COMMERCE (20)

-o-o-o-o-

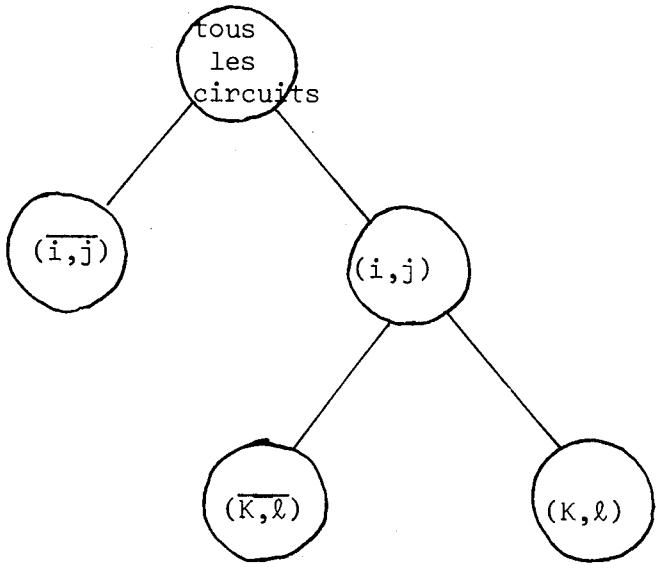
L'idée de base de l'algorithme est la suivante : partager l'ensemble de tous les circuits hamiltoniens en sous-ensembles de plus en plus réduits et calculer pour chaque sous-ensemble une borne inférieure des longueurs des circuits qu'il contient. Cette borne inférieure guide le partitionnement des sous-ensembles et identifie un circuit hamiltonien minimal, si un sous-ensemble contient un seul circuit dont la longueur est plus petite ou égale aux bornes inférieures de tous les autres sous-ensembles.

L'algorithme utilise une matrice de "coûts", $C = (c_{ij})$ où c_{ij} est le coût d'aller de i à j (C n'est pas forcément symétrique).

C est <u>réduite</u> si elle a au moins un zéro dans chaque ligne et chaque colonne; la réduction de C est obtenue en soustrayant le plus petit élément de chaque ligne de tous les éléments de la ligne, et de même pour chaque colonne.

Le partage de l'ensemble de tous les circuits en plusieurs sous-ensembles sera représenté par un arbre, dont le noeud initial représente "tous les circuits". Le noeud (i,j) représente "tous les circuits qui contiennent la paire {i,j} ". Le noeud $(\overline{i,j})$ représente "tous les circuits qui ne contiennent pas la paire {i,j} ".

EXEMPLE :



Soit t un circuit hamiltonien. On notera par :

- z(t) : le coût du circuit t
- X, Y, $\overline{Y}$: noeuds de l'arbre, tels que X est relié à Y et $\overline{Y}$
- W(X) : borne inférieure des coûts de tous les circuits de X
- zo : coût du circuit minimal trouvé par l'algorithme.

L'algorithme est le suivant :

1) $C = (c_{ij})_{i,j}$, $zo = \infty$

2) Réduire C et affecter au noeud initial de l'arbre ("tous les circuits") la somme des constantes de réduction de C.

3) Choisir {K, l}, pour l'extension de l'arbre, tel que :

$\theta(K,l) = \max \theta(i,j)$, où

$\theta(i,j) = [$le plus petit coût dans la ligne i, en supprimant $c_{ij}\ ] + [$le plus petit coût dans la colonne j, en supprimant c_{ij}

4) Faire la liaison de X à $\overline{Y} = (\overline{k,\ l})$ et affecter à $\overline{Y}$ la borne inférieure $W(\overline{Y}) = W(X) + \theta(K,\ l)$.

5) Faire la liaison de X à Y = (K, l). Supprimer la ligne K et la colonne l dans C. Trouver le point de départ p, et le point d'arrivée m du circuit contenant (K, l). Faire $C_{pm} = \infty$. Réduire C. Affecter à Y, W(Y) = W(X) + |somme des constantes de réduction de C| .

6) Si C est devenue une matrice 2 x 2 aller à 11).

7) Considérer le nouveau noeud X pour la prochaine extension de l'arbre, celui qui a le plus petit W(X).

8) Si zo $\leq$ W(X) alors aller à 13).

9) Si X est égal à Y trouvé au pas 3), alors aller à 3).

10) • Considérer la matrice originale des coûts : C
- Trouver (i,j) qui font partie des circuits qui contiennent X et faire $g = \Sigma C_{ij}$
- Supprimer pour chaque (i,j) la ligne i et la colonne j de C. Pour chaque chemin contenant les (i,j) noeuds, trouver le point de départ p et le point d'arrivée m et mettre $C_{mp} = \infty$. Pour chaque (K,l) qui ne fait pas partie de circuits contenant X, mettre $C_{Kl} = \infty$.
- Réduire C.
- Affecter à X, W(X) = g + |somme des constantes de réduction| . Aller à 3).

11) Si W(Y) < zo alors aller à 7).

12) zo = W(Y) et garder le circuit. Aller à 7).

13) Fin.

REMARQUES

1) - Dans notre cas, pour la recherche des équipotentielles minimales, la matrice C étant symétrique (matrice des distances), pour éviter d'obtenir un circuit minimal deux fois (parcouru dans les deux sens), nous mettrons ainsi que les auteurs l'indiquent, $e_{Kl} = \infty$ si Y = (K,l) a été choisi.

2) - Ayant besoin de tous les circuits hamiltoniens minimaux, au lieu d'éli miner des noeuds X tels que W(X) = zo, on les considèrera comme de nouvelle sources d'extension de l'arbre pour obtenir, éventuellement, de nouveaux circuits hamiltoniens minimaux.

3) - Il existe des algorithmes plus récents et plus performants que l'algorithme de Little et .... (20) , que nous avons présenté ici (27). Mais, dans notre cas, le nombre d'éléments de l'équipotentielle étant petit (moir de 10 pattes), l'algorithme de Little et .... est aussi rapide que les autres. Etant plus connu, nous l'avons préféré.

BIBLIOGRAPHIE

1) - M. HANAN et J. KURTZBERG : "A review of the placement and quadratic assignment problems", SIAM Review, vol. 14, n° 2, april 1972.

2) - H.W. KUHN : "The Hungarian method for the assignment problem", Naval Research Logistics Quarterly, 2, 1955, p. 83-97.

3) - J. MUNKRES : "Algorithms for the Assignment and Transportation Problems", SIAM Journal, vol. 5, march 1957, p. 32-38.

4) - KRUSKAL J.B. : "On the shortest spanning subtree of a graph", Proc. Amer. Math. Soc., 7, 1956, p. 48-50.

5) - G. DEMOUCRON : "Optimisation du câblage", Gestion, Fr. Février, 1968, N° 11, p. 186-197.

6) - C. BERGE : "Théorie des graphes et ses applications", Ed. Dunod, Paris.

7) - L. STEINBERG : "The backboard wiring problem : a placement algorithm", SIAM Journal, 1961, 3, p. 37-50.

8) - R.A. RUTMAN : "An algorithm for placement of interconnected elements based on minimum wire length", Proc. AFIPS, 1964, vol. 25, p. 471-491.

9) - P.C. GILMORE : "Optimal and suboptimal algorithms for the quadratic assignment problem", SIAM Journal, 10, 1962, p. 305-313.

10) - E.L. LAWLER : "The quadratic assignment problem", Management SCI, 9, 1963, p. 586-599.

11) - J.S. MAMELAK : "The placement of computer logic modules", J. Ass. Comp. Mach., 1966, 13, p. 615-629.

12) - R.G. GARSIDE et T.A. NICHOLSON : "Permutation procedure for the back board wiring problem", Proc. IEE, 115, p. 27-30.

13) - J.W. GAWETT, ... : "The optimal assignment of facilities to location by branch and bound", Operations Res., 1966, p. 210.

14) - L. COOPER : "Heuristic methods for location-allocation problems", SI Review, 6, 1964, p. 37-53.

15) - J.M. KURTZBERG : "On approximation methods for the assignment proble J. Assoc. Comp. Mach., 9, 1962, p. 419-439.

16) - M. SARRET : "Problèmes d'implantation, le programme DESMAG", Thèse d 3e cycle, 1969, Grenoble.

17) - F. LUCCIO et M. SAMI : "On the Decomposition of Networks in Minimall Interconnected Subnetworks", IEEE Trans. on Circ. Theory, may, 1 pp. 184-188.

18) - M. BELLMORE et G.L. NEMHAUSER : "The traveling Salesman problem : a survey", Operations Res. Jan.Fevr. 1968, p. 539-558.

19) - C.E. NUGENT :"An experiment comparaison of techniques for the assignm of the facilities to locations", Operations Res. 16, 1968, p. 15(173.

20 - J.D.C. LITTLE,... : "An algorithm for the traveling Salesman problem" Operations Res. 11, 1963, p. 979-989.

21) - R. PERTUISET : "Optimisation du câblage des ordinateurs par l'applica tion de la théorie des graphes", Thèse d'Ingénieur, C.N.A.M., Nov. 1965.

22) - G.W. ALTMAN : "Automation of Computer Panel Wiring", Communications and Electronics, 48, May, 1960, p. 118-125.

23) - J.M. KURTZBERG : "Algorithms for Backplane Formation", Symposium of Microelectronics and Larges Systems, Nov. 1964.

24) - T. POMENTALE : "An algorithm for minimizing backboard wiring functions", Com. ACM, 8, 1965, p. 696-703.

25) - N. SUGIYAMA,... : "An integrated circuit layout design system", Computer Aided Design, Vol. 6, 2, April 1974.

26) - H. LOBERMAN and A. WEINBERGER : "Formal procedures for connecting terminals with a minimum total wire length", J. Assoc. Comp. Mach., 4, 1957, p. 428-437.

27) - M. HELD and R.M. KARP : "The traveling salesman problem and minimal spanning trees", Op. Res., 18, 1970, 1138-1162.

28) - M.A. BREUER : "Design Automation of Digital Systems, Theory and Techniques", Ed. Englewood Cliffs.

29) - D. LITAIZE : "Tracé automatique de plaquettes de circuits imprimés en technologie multicouche haute densité", Thèse Docteur de Spécialité, Toulouse 1971.

30) - J.F. TRICHARD : "Tracé automatique de plaquettes de circuits imprimés", Thèse Docteur - Ingénieur, Toulouse 1968.

31) - A. GRESSE : "Tracé automatique de plaquettes en technologie multicouche et blocs denses", Thèse Docteur - Ingénieur, Toulouse, 1970.

32) - C.J. FISK, D. CASKEY, L.E. WEST : "ACCEL - Automated Circuit Card Etching Layout", Proc. IEEE, 1, Nov. 1967, p. 1971-1982.

33) - L.C. ABEL : "On the ordering of Connections for Automatic Wire Routir IEEE Trans. On Computers, Nov. 1972, p. 1227,1233.

34) - C.Y. LEE : "An algorithm for path connections and its applications", IRE Trans. on Electr. Computers, Vol. EC. 10, 5, Sept, 1961.

35) - E. MAJORANI : "Simplification of Lee's algorithm for special problems Calcolo, 1, 1964, p. 246-256.

36) - J.P. AVENIER, J.M. GUEGAN : "Artemis : Algorithme de Recherche et de Tracé d'Equipotentielles pour les Masques de circuits Intégrés", Projet 3e année, IP Grenoble, 1970.

37) - M. CHEIN : "Sur des problèmes de décomposition d'un graphe liés à l'implantation", Thèse Doctorat d'Etat, Grenoble, 1970.

38) - K.MIKANI et K. TABUCHI : "A computer program for optimal routing of printed circuit conductors", IFIPS, Proc, 1968.

39) - I. ARAMAKI,... : "Automation of Etching Pattern Layout", Com. ACM, Nov. 1971.

40) - G. NICOLAS : "Interconnexion de circuits intégrés par fils isolés", Rapport final, LETI/EPA, CENG, juillet 1973.

ERRATA

Page 63
- lignes 4 et 10 du haut: $\Upsilon \in \{0,1,2,3,4\}$.
- ligne 7 du haut: 2 ou 3 coudes (...

Page 67
- lignes 11, 12, 16 du haut: $4p_{ij} + p_{ik}$
- ligne 9 du bas: Υ est au moins 0

Page 70
- lignes 5, 7, 8 du bas: $4p_{ij} + p_{ik}$

Page 71
- ligne 5 du bas: $4p_{ik} + p_{ij} = 320$
- ligne 4 du bas: $4p_{ik} + p_{ij}$

Page 73
- ligne 10 du haut: si le module $k \in \{V_0, V_1, V_2, H_0, H_1, H_2\}$