



HAL
open science

Placement automatique de circuits intégrés

Eric Chotin

► **To cite this version:**

Eric Chotin. Placement automatique de circuits intégrés. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1992. Français. NNT: . tel-00341773

HAL Id: tel-00341773

<https://theses.hal.science/tel-00341773>

Submitted on 26 Nov 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.

Thèse

Présentée par

Eric CHOTIN

Pour obtenir le grade de **Docteur**

de l'Institut National Polytechnique de Grenoble

(arrêté ministériel du 30 Mars 1992)

Spécialité : **Microélectronique (CAO)**

Placement automatique de circuits intégrés

Soutenue le 20 Novembre 1992

Composition du Jury:

Madame	G. Saucier	
Messieurs	J. Mossière	Président
	G. Dreyfus	Rapporteur
	R. Gerber	Rapporteur
	P. Magarshack	

Thèse préparée au sein du Laboratoire Conception de Systèmes Intégrés

PRESIDENT DE L'INSTITUT
Monsieur Maurice RENAUD



Année 1991/1992

PROFESSEURS DES UNIVERSITES

BARIBAUD	Michel	ENSERG
BARRAUD	Alain	ENSIEG
BAUDELET	Bernard	ENSPG
BAUDIN	Gérard	UFR PGP
BEUFILS	Jean-Pierre	ENSIEG/ILL
BOIS	Philippe	ENSHMG
BONNET	Guy	ENSPG
BOUYIER	Gérard	ENSERG
BRISSONNEAU	Pierre	ENSIEG
BRUNET	Yves	CUEFA
CAILLERIE	Denis	ENSHMG
CAYAIGNAC	Jean-François	ENSPG
CHARTIER	Germain	ENSPG
CHENEYIER	Pierre	ENSERG
CHERADAME	Hervé	UFR/PGP
CHERUY	Arlette	ENSIEG
CHOYET	Alain	ENSERG
COGNET	Gérard	ENSHMG
COLINET	Catherine	ENSEEG
COMMAULT	Christian	ENSIEG
CORNUT	Bruno	ENSIEG
COULOMB	Jean-Louis	ENSIEG
CROWLEY	James	ENSIMAG
DALARD	Francis	ENSEEG
DARVE	Félix	ENSHMG
DELLA DORA	Jean	ENSIMAG
DEPEY	Maurice	ENSERG
DEPORTES	Jacques	ENSPG
DEROO	Daniel	ENSEEG
DESRE	Pierre	ENSEEG
DIARD	Jean-Paul	ENSEEG
DOLMAZON	Jean-Marc	ENSERG
DURAND	Francis	ENSEEG
DURAND	Jean-Louis	ENSPG
FAUTRELLE	Yves	ENSHMG
FOGGIA	Albert	ENSIEG
FONLUPT	Jean	ENSIMAG
FOULARD	Claude	ENSIEG
GALERIE	Alain	ENSEEG
GANDINI	Alessandro	UFR/PGP
GAUBERT	Claude	ENSPG
GENTIL	Pierre	ENSERG
GENTIL	Sylviane	ENSIEG
GREYEN	Hélène	CUEFA
GUERIN	Bernard	ENSERG
GUYOT	Pierre	ENSEEG
IYANES	Marcel	ENSIEG
JALLUT	Christian	ENSEEG



JANDT	Marie-Thérèse	ENSERG
JAUSSAUD	Pierre	ENSIEG
JOST	Rémy	ENSPG
JOUBERT	Jean-Claude	ENSPG
JOURDAIN	Geneviève	ENSIEG
KUENY	Jean-Louis	ENSHMG
LACHENAL	Dominique	UFR PGP
LACOUME	Jean-Louis	ENSIEG
LADET	Pierre	ENSIEG
LESIEUR	Marcel	ENSHMG
LESPINARD	Georges	ENSHMG
LIENARD	Joël	ENSIEG
LONGEQUEUE	Jean-Pierre	ENSPG
LORET	Benjamin	ENSHMG
LOUCHET	François	ENSEEG
LUCAZEAU	Guy	ENSEEG
LUX	Augustin	ENSIMAG
MASSE	Philippe	ENSIEG
MASSELOT	Christian	ENSIEG
MAZARE	Guy	ENSIMAG
MICHEL	Gérard	ENSIMAG
MOHR	Roger	ENSIMAG
MOREAU	René	ENSHMG
MORET	Roger	ENSIEG
MOSSIÈRE	Jacques	ENSIMAG
OBLED	Charles	ENSHMG
OZIL	Patrick	ENSEEG
PANANAKAKIS	Georges	ENSERG
PAULEAU	Yves	ENSEEG
PERRET	Robert	ENSIEG
PIAU	Jean-Michel	ENSHMG
PIC	Etienne	ENSERG
PLATEAU	Brigitte	ENSIMAG
POUPOT	Christian	ENSERG
RAMEAU	Jean-Jacques	ENSEEG
REINISCH	Raymond	ENSPG
RENAUD	Maurice	UFR/PGP
ROBERT	François	ENSIMAG
ROSSIGNOL	Michel	ENSPG
ROYE	Daniel	ENSIEG
SABONNADIÈRE	Jean-Claude	ENSIEG
SAGUET	Pierre	ENSERG
SAUCIER	Gabrièle	ENSIMAG
SCHLENKER	Claire	ENSPG
SCHLENKER	Michel	ENSPG
SILVY	Jacques	UFR/PGP
SIRIEYS	Pierre	ENSHMG
SOHM	Jean-Claude	ENSEEG
SOLER	Jean-Louis	ENSIMAG
SOUQUET	Jean-Louis	ENSEEG
TICHKIEWITCH	Serge	ENSHMG
TROMPETTE	Philippe	ENSHMG
TRYSTRAM	Denis	ENSGI
YEILLON	Gérard	ENSIMAG
VERJUS	Jean-Pierre	ENSIMAG
VINCENT	Henri	ENSPG
ZADWORNÝ	François	ENSERG

SITUATION PARTICULIERE

PROFESSEURS D'UNIVERSITE

DETACHEMENT

ENSERG	BLIMAN	Samuel	Mutation	
ENSPG	BLOCH	Daniel	Recteur	21/12/93
ENSIMAG	LATOMBE	Jean-Claude	Détachement	01/05/93
ENSHMG	PIERRARD	Jean-Marie	Disponible	

RETRAITE

ENSEEG	BONNETAIN	Lucien	
--------	-----------	--------	--

DIRECTEURS DE RECHERCHE CNRS

ABELLO	Louis	MEUNIER	Gérard
ALDEBERT	Pierre	MICHEL	Jean-Marie
ALEMANY	Antoine	NAYROLLES	Bernard
ALLIBERT	Colette	PASTUREL	Alain
ALLIBERT	Michel	PEUZIN	Jean-Claude
ANSARA	Ibrahim	PHAM	Antoine
ARMAND	Michel	PIAU	Monique
AUDIER	Marc	PIQUE	Jean-Paul
AUGOYARD	Jean-François	POINSIGNON	Christiane
AYIGNON	Michel	PREJEAN	Jean-Jacques
BERNARD	Claude	RENOUARD	Dominique
BINDER	Gilbert	SENATEUR	Jean-Pierre
BLAISING	Jean-Jacques	SIFAKIS	Joseph
BONNET	Roland	SIMON	Jean-Paul
BORNARD	Guy	SUERY	Michel
BOUCHERLE	Jean-Xavier	TEODOSIU	Christian
CAILLET	Marcel	YACHAUD	Georges
CARRE	René	YAUCLIN	Michel
CHASSERY	Jean-Marc	WACK	Bernard
CHATILLON	Christian	YAYARI	Ali-Reza
CIBERT	Joël	YONNET	Jean-Paul
CLERMONT	Jean-Robert		
COURTOIS	Bernard		
CRICUI	Patrick		
CRISTOLOVEANU	Sorin		
DAVID	René		
DION	Jean-Michel		
DOUSSIERE	Jacques		
DRIGLE	Jean		
DUCHET	Pierre		
DUGARD	Luc		
DURAND	Robert		
ESCUDIER	Pierre		
EUSTATHOPOULOS	Nicolas		
FINON	Dominique		
FRUCHARD	Robert		
GARNIER	Marcel		
GIRDD	Jacques		
GLANGEAUD	François		
GUELIN	Pierre		
HOPFINGER	Emil		
JORRAND	Philippe		
JOUD	Jean-Charles		
KAMARINOS	Georges		
KLEITZ	Michel		
KOFMAN	Walter		
LACROIX	Claudine		
LANDAU	Ioan		
LAULHERE	Jean-Pierre		
LEGRAND	Michel		
LEJEUNE	Gérard		
LEPROVOST	Christian		
MADAR	Roland		
MARTIN	Jean-Marie		
MERMET	Jean		

PERSONNES AYANT OBTENU LE DIPLOME
D'HABILITATION A DIRIGER DES RECHERCHES

BALESTRA	Francis
BALME	Louis
BECKER	Monique
BIGEON	Jean
BINDER	Zdeneck
BOE	Louis-Jean
CANUDAS DE WIT	Carlos
CHOLLET	Jean-Pierre
COEY	Jean-Pierre
CORNUEJOLS	Gerard
COURNIL	Michel
CRASTES DE PAULET	Michel
DALLERY	Yves
DESCOTES-GENON	Bernard
DUGARD	Luc
DURAND	Madeleine
FERRIEUX	Jean-Paul
FEUILLET	René
FREIN	Yannick
GAUTHIER	Jean-Paul
GHIBAUDO	Gérard
GUILLEMOT	Nadine
GUYOT	Alain
HAMAR	Sylviane
HAMAR	Roger
HORAUD	Patrice
JACQUET	Paul
LATOMBE	Claudine
LE HUY	Hoang
LE GORREC	Bernard
LOZANO-LEAL	Rogelio
MACOVSCHI	Mihail
MAHEY	Philippe
METAIS	Olivier
MONMUSSON-PICQ	Georgette
MORY	Mathieu
MULLER	Jean
MULLER	Jean-Michel
NGUYEN TRONG	Bernadette
NIEZ	Jean-Jacques
PERRIER	Pascal
PLA	Fernand
RECHENMANN	François
ROGNON	Jean-Pierre
ROUGER	Jean
ROUX	Jean-Claude
TCHUENT	Maurice

PERSONNES AYANT OBTENU LE DIPLOME

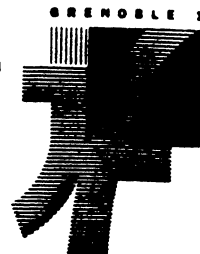
DE DOCTEUR D'ETAT INPG

ABDEL-RAZEK	Adel
AKSAS	Haris
ALLA	Hassane
AMER	Ahmed
ANCELLE	Bernard
ANGENIEUX	Gilbert
ATMANI	Hamid
AYEDI	Hassine Feri
A.BADR	Osman
BACHIR	Aziz
BALANZAT	Emmanuel
BALTER	Roland
BARDEL	Robert
BARRAL	Gérard
BAUDON	Yves
BAUSSAND	Patrick
BEAUX	Jacques
BEGUINOT	Jean
BELLISSANT née FUNEZ	Marie-Claire
BELLON	Catherine
BEN RAIS	Abdejettah
BERGER-SABBATEL	Gilles
BERNACHE-ASSOLANT	Didier
BEROVAL	Abderrahmane
BERTHOD	Jacques
BILLARD	Dominique
BLANC épouse FOULETIER	Mireille
BOCHU	Bernard
BOJO	Gilles
BOKSENBAUM	Claude
BOLOPION	Alain
BONNARD	Bernard
BORRIONE	Dominique
BOUCHACOURT	Michel
BRINI	Jean
BRION	Bernard
CAIRE	Jean-Pierre
CAMEL	Denis
CAPERAN	Philippe
CAPLAIN	Michel
CAPOLINO	Gérard
CASPI	Paul
CHAN-TUNG	Nam
CHASSANDE	Jean-Pierre
CHATAIN	Dominique
CHEHIKIAN	Alain
CHIRAMELLA	Yves
CHILO	Jean
CHUPIN	Jean-Claude
COLONNA	Jean-François
COMITI	Jacques
CORDET	Christian
COUDURIER	Lucien

COUTAZ	Jean-Louis
CREUTIN	Jean-Dominique
DAO	Trongtich
DARONDEAU	Philippe
DAYID	Bertrand
DE LA SEN	Manuel
DELACHAUME	Jean-Claude
DENAT	André
DESCHIZEAUX née CHERUY	Marie-Noëlle
DIJON	Jean
DOREMUS	Pierre
DUPEUX	Michel
EL ADHAM	Karim
EL OMAR	Fovaz
EL-HENNAWY	Adel
ETAY	Jacqueline
FABRE épouse MAXIMOVITCH	Suzanne
FAURE-BONTE épouse MARET	Mireille
FAYIER	Denis
FAYIER	Jean-Jacques
FELIACHI	Movlound
FERYAL	Haj Hassan
FLANDRIN	Patrick
FOREST	Bernard
FORESTIER	Michel
FOSTER	Panayolis
FRANC	Jean-Pierre
GADELLE	Patrice
GARDAN	Yvon
GENIN	Jacques
GERYASON	Georges
GILORMINI	Pierre
GINOUX	Jean-Louis
GOUMIRI	Louis
GROC	Bernard
GROSJEAN	André
GUEDON	Jean-Yves
GUERIN	Jean-Claude
GUESSOUS	Anas
GUIBOUD-RIBAUD	Serge
HALBWACHS	Nicolas
HAMMOURI	Hassan
HEDEIROS SILIYEIRA	Hamilton
HERAULT	Jeanny
HONER	Claude
HUECKEL	Tomasz
IGNAT	Michel
ILIADIS	Athanasios
JANIN	Gérard
JERRAYA	Ahmed Amine
JUTTEN	Christian
KAHIL	Hassan
KHUONGQUANG	Dong
KILLIS	Andreas
KONE	Ali
LABEAU	Michel
LACAZE	Alain
LACROIX	Jean-Claude
LANG	Jean-Claude
LATHUILLERE	Chantal

LATY	Pierre
LAUGIER	Christian
LE CADRE	Jean-Pierre
LE GARDEYR	René
LE NEST	Jean-François
LE THIESSE	Jean-Claude
LEMAIGNAN	Clement
LEMUET	Daniel
LEVEQUE	Jean-Luc
LONDICHE	Henry
L'HERITIER	Philippe
MAGNIN	Thierry
MAISON	François
MAMWI	Abdullah
MANTEL épouse SIEBERT	Elisabeth
MARCON	Guy
MARTINEZ	Francis
MARTIN-GARIN	Lionel
MASSE	Dominique
MAZER	Emmanuel
MERCKEL	Gérard
MEUNIER	Jean
MILI	Ali
MOALLA	Mohamed
MODE	Jean-Michel
MONLLOR	Christian
MONTELLA	Claude
MORET	Frédéric
MRAYATI	Mohammed
M'SAAD	Mohammed
M'SIRDI	Kouider Nace
NEPOMIASTCHY	Pierre
NGUYEN	Trong Khoi
NGUYEN-XUAN-DANG	Michel
ORANIER	Bernard
ORTEGA MARTINEZ	Roméo
PAIDASSI	Serge
PASSERONE	Alberto
PEGON	Pierre
PIJOLAT	Christophe
POGGI	Yves
POIGNET	Jean-Claude
PONS	Michel
POU	Tong Eck
RAFINEJAD	Paiviz
RAGAIE	Harie Fikri
RAHAL	Salah
RAMA SEABRA SANTOS	Fernando
RAYAINE	Denis
RAZBAN-HAGHIGHI	Tchanguiz
RAZZOUK	Micham
REGAZZONI	Gilles
RIQUET	Jean-Pierre
ROBACH	Chantal
ROBERT	Yves
ROGEZ	Jacques
ROHMER	Jean
ROUSSEL	Claude
SAAD	Abdallah
SAAD	Youcef

SABRY	Mohamed Nabi
SALON	Marie-Christine
SAUBAT épouse MARCUS	Bernadette
SCHMITT	Jean-Hubert
SCHOELLKOPF	Jean-Pierre
SCHOLL	Michel
SCHOLL	Pierre-Claude
SCHOULER	Edmond
SCHWARTZ	Jean-Luc
SEGUIN	Jean
SIWY	Jacques
SKALLI	Abdellatif
SKALLI HOUSSEYNI	Abdelali
SOUCHON	Alain
SUETRY	Jean
TALLAJ	Nizar
TEDJAR	Farouk
TEDJINI	Smail
TEYSSANDIER	Francis
THEVENODFOSSE	Pascale
TMAR	Mohamed
TRIDLLIER	Michel
TUFFELIT	Denis
TZIRITAS	Georges
VALLIN	Didier
VELAZCO	Raoul
VERDILLON	André
YERMANDE	Alain
YIKTOROVITCH	Pierre
YITRANT	Guy
WEISS	François
YAZAMI	Rachid



Président de l'Université :

M. NEMOZ Alain

ANNEE UNIVERSITAIRE 1990 - 1991

MEMBRES DU CORPS ENSEIGNANT DE SCIENCES ET DE GEOGRAPHIE

PROFESSEURS DE 1ERE CLASSE

ADIBA Michel
ANTOINE Pierre
ARVIEU Robert
AURIAULT Jean Louis
BARA Jean René
BECKER Pierre
BEGUIN Claude
BELORISKY Elie
BENZAKEN Claude
BERARD Pierre
BERNARD Alain
BERTRANDIAS Françoise
BERTRANDIAS Jean Paul
BILLET Jean
BLANCHI Jean Pierre
BOEHLER Jean Paul
BOITET Christian
BORNAREL Jean
BRUANDET Jean François
CARLIER Georges
CASTAING Bernard
CHARDON Michel
CHIBON Pierre
COHEN ADDAD Jean Pierre
COLIN DE VERDIERE Yves
CYROT Michel
DEBELMAS Jacques
DEMAILLY Jean Pierre
GENEUVILLE Alain
DEPORTES Charles
DOLIQUE Jean Michel
DOUCE Roland
DUCROS Pierre
FINKE Gerde
GAUTRON René
GENIES Eugène
GERMAIN Jean Pierre
GIDON Maurice
GIGNOUX Claude
GILLARD Roland
GUITTON Jacques

Informatique
Géologie I.R.I.G.M.
Physique Nucléaire I.S.N.
Mécanique
Statistiques - Mathématiques Appliquées
Physique
Chimie Organique
Physique
Mathématiques Pures
Mathématiques Pures
Mathématiques Pures
Mathématiques Pures
Mathématiques Pures
Géographie
A.P.S.
Mécanique
Informatique et Mathématiques Appliquées

Physique
Biologie Végétale
Physique
Géographie
Biologie Animale
Physique
Mathématiques Pures
Physique du Solide
Géologie Générale
Mathématiques Pures
Physique
Chimie Minérale
Physique des Plasmas
Physiologie Végétale
Cristallographie
Informatique
Chimie
Chimie
Mécanique
Géologie
Sciences nucléaires
Mathématiques
Chimie

.. / ...

HERAULT Jeanny
HICTER Pierre
JANIN Bernard
JOLY Jean René
JOSELEAU Jean Paul
KAHANE André
KAHANE Josette
KRAKOWIAK Sacha
LAJZEROWICZ Jeanine
LAJZEROWICZ Joseph
LAURENT Pierre Jean
LEBRETON Alain
DE LEIRIS Joël
LHOMME Jean
LOISEAUX Jean Marie
LONGEQUEUE Nicole
LUNA Domingo
MACHE Régis
MASCLE Georges
MAYNARD Roger
NEMOZ Alain
OMONT Alain
PELMONT Jean
PERRIER Guy
PIERRE Jean Louis
RENARD Michel
RICHARD Jean Marc
RIEDTMANN Christine
RINAUDO Marguerite
ROBERT Jean Bernard
ROSSI André
SAXOD Raymond
SENGEL Philippe
SERGERAERT Francis
SOUCHIER Bernard
STUTZ Pierre
TRILLING Laurent
VALLADE Marcel
VAN CUTSEM Bernard
VIALON Pierre
VIDAL Micheal

Physique
Chimie
Géographie
Mathématiques Pures
Biochimie
Physique
Physique
Mathématiques Appliquées
Physique
Physique
Mathématiques Appliquées
Mathématiques Appliquées
Biologie
Chimie
Sciences Nucléaires I.S.N.
Physique
Mathématiques Pures
Physiologie Végétale
Géologie
Physique du Solide
Physique
Astrophysique
Biochimie
Géophysique
Chimie Organique
Thermodynamique

Mathématiques
Chimie C.E.R.M.A.V.

Biologie
Biologie Animale
Biologie Animale
Mathématiques Pures
Biologie
Mécanique
Mathématiques Appliquées
Physique
Mathématiques Appliquées
Géologie

PROFESSEURS DE 2EME CLASSE

APPARU Marcel
ARMAND Gilbert
ARNAUD Hubert
ARTRU Marie Christine
ATTANE Pierre
BARATE Robert
BARET Paul
BARGE Jean
BARLET Roger
BERTIN José
BLOCK Marc
BLUM Jacques
BOITET Christian
BORRIONE Dominique
BOULON Marc
BOUTRON Claude
BOUVET Jean
BROSSARD Jean
BRUGAL Gérard
CAMPILLO Michel
CAVILLE Jean Yves
CERFF Rudiger
CHIARAMELLA Yves
CHOLLET Jean Pierre
COLOMBEAU Jean François
COTTET Georges-Henri
COURT Jean
CUNIN Pierre Yves
DAVID Jean
DEROUARD Jacques
DHOUILLY Danièle
DUFRESNOY Alain
DUPUY Claude
DURAND Mireille
FONTECAVE Marc
FOURNIER Jean Marc
GASPARD François
GIDON Maurice
GIORNI Alain
GONZALEZ SPRINBERG Gérardo
GOURC Jean Pierre
GUIGO Maryse
GUMUCHIAN Hervé
HACQUES Gérard
HAMMOU Abdelkader
HERBIN Jacky
HERINO Roland
HERZOG Michel
JARDON Pierre
JUTTEN Christian
KERCKHOVE Claude
KOSAREW Siegmund
KLINGER Jurgen
LAURENT Christine
MANDARON Paul
MARTINEZ Francis
MERCHEZ Fernand
MILAS Michel
MOREL Alain
MORIN Pierre
NGUYEN HUY Xuong-
OUDET Bruno

Chimie
Géographie
Géologie
Physique
Mécanique
Sciences Nucléaires
Chimie
Mathématiques
Chimie
Mathématiques
Biologie
Mathématiques Appliquées
Mathématiques Appliquées
Automatique informatique
Mécanique
Glaciologie
Biologie
Mathématiques
Biologie
Géophysique
Chimie
Biologie
Mathématiques Appliquées
Mécanique
Mathématiques (ENSL)
Modélisation, calcul scientifique, statis.
Chimie
Informatique
Géographie
Physique
Biologie
Mathématiques Pures
Chimie
Sciences Nucléaires
Chimie
Physique
Physique
Géologie
Sciences Nucléaires
Mathématiques Pures
Mécanique
Géographie
Géographie
Mathématiques Appliquées
Chimie
Géographie
Physique
Biologie
Chimie
Physique
Géologie
Math. fondamentales et appliquées
Glaciologie
Mathématiques
Biologie
Mathématiques Appliquées
Physique
Chimie
Géographie
Physique
Informatique
Mathématiques Appliquées

PAUTOU Guy
PECHER Arnaud
PELLETIER Guy
PERRIN Claude
PFISTER Claude
PIBOULE Michel
PORTESEIL Jean Louis
PUECH Laurent
RAYNAUD Hervé
REGNARD Jean René
ROBERT Claudine
ROBERT Danielle
ROBERT Gilles
SAJOT Gérard
SARROT REYNAULD Jean
SAYETAT Françoise
SERVE Denis
STOECKEL Frédéric
SCHOLL Pierre Claude
SUBRA Robert
TEMPERVILLE André
TISSUT Michel
TOURNIER Evelyne
VALLADE Marcel
VALLON Michel
VICAT Jean
VINCENS Maurice
VINCENT Gilbert
VIVIAN Robert
VOTTERO Philippe
WITOMSKI Patrick

Biologie
Géologie
Astrophysique
Sciences Nucléaires I.S.N.
Biologie
Géologie
Physique
Physique
Mathématiques Appliquées
Physique
Didactique des disciplines scientifiques
Chimie
Mathématiques Pures
Physique
Géologie
Physique
Chimie
Physique
Mathématiques Appliquées
Chimie
Mécanique
Biologie
Informatique et Mathématiques appliquées
Physique
Glaciologie
Physique
Chimie
Physique
Géographie
Chimie

Avant-Propos

Je tiens à remercier tout d'abord Madame Gabrièle SAUCIER, Professeur à l'ENSIMAG et directrice du laboratoire Conception de Systèmes Intégrés, pour m'avoir accueilli dans son laboratoire et pour avoir guidé mes recherches. Le travail présenté ici doit beaucoup à ses remarques et suggestions.

Je remercie également:

Monsieur Jacques MOSSIERE, Professeur et directeur de l'ENSIMAG, de me faire l'honneur de présider ce jury,

Messieurs Gérard DREYFUS, Professeur à l'ESPCI, et Roland GERBER, directeur de la section CCI du CNET, d'avoir accepté d'être rapporteurs de cette thèse,

Monsieur Philippe MAGARSHACK, Ingénieur à Thomson Composants Militaires et Spatiaux, pour avoir accepté de faire partie de mon jury.

Enfin, je remercie les membres du CSI, actuels et anciens, avec qui j'ai partagé de très bons moments et plus particulièrement Anne MIGNOTTE pour sa bonne humeur et ses bons conseils, ainsi que tous ceux qui, de près ou de loin, ont contribué aux travaux présentés dans cette thèse.

Je ne saurais terminer sans remercier ma famille et les copains, pour leur réconfort et pour avoir supporté mes humeurs pas toujours faciles (surtout ces derniers temps!...).

Résumé

Cette thèse présente l'étude et l'implantation de deux méthodes pour le placement automatique de circuits intégrés. Un circuit intégré peut être considéré comme un ensemble de blocs et une liste d'interconnexions entre ces blocs. Le problème du placement consiste à disposer les blocs sur la surface hôte en respectant diverses contraintes et en optimisant des critères comme la surface occupée et la longueur totale de connexions. Les méthodes présentées ici sont toutes les deux guidées par l'optimisation de la connectique. La première fait appel à une technique d'analyse de données, l'analyse d'un tableau de proximités. Dans un premier temps, des proximités sont définies entre les blocs de façon à refléter un agencement idéal en fonction de la connectique. L'utilisation de l'ATP permet alors d'obtenir une disposition planaire des blocs respectant au mieux les proximités qui ont été définies. L'analyse effectuée fait le point sur les diverses façons de définir les proximités entre les blocs, ainsi que sur les traitements ultérieurs destinés à l'obtention d'un placement réalisable. Les qualités et les limitations de cette approche sont ensuite discutées.

La seconde méthode est connue sous le nom de placement par bipartitionnements successifs. L'ensemble des blocs du circuit et la surface hôte sont ainsi bipartitionnés récursivement jusqu'à ce que l'emplacement de chaque bloc soit déterminé. A partir des algorithmes existants, des heuristiques ont été mises au point afin de permettre la prise en compte de contraintes supplémentaires comme le traitement des plots d'entrées-sorties ou des blocs pré-fixés. L'expérimentation a permis de valider ces heuristiques et de comparer les résultats du placement par bipartitionnements successifs à ceux fournis par la première méthode.

Mots clefs : placement de circuit, analyse de données, bipartitionnement de graphe.

Summary

This thesis presents the study and the implantation of two methods for the automated placement of integrated circuits. An integrated circuit is considered as a set of blocks and a set of connections between these blocks. The placement problem is to determine a position for each block of the circuit, in order to respect given constraints and to optimize some criteria such as the total area or the total wirelength. The two methods presented here are both directed by the wirelength minimisation. The first one is based upon a statistical data analysis technique called proximity analysis. In a first time, proximities are defined between each pair of block in order to reflect a perfect arrangement of the blocks according to their connectivity. Then the proximity analysis gives the planar arrangement of the blocks that tends to respect the defined proximities. This thesis study the different ways to define proximities between blocks, and focuses on compaction, decompaction and orientation of the blocks, which are used to obtain a placement respecting the constraints. The advantages and drawbacks of this approach are discussed.

The second method is known as a min-cut placement. The set of blocks of the circuit and the area of the chip are recursively partitioned until each block position is known. From existing algorithms, some heuristics have been developed to allow new constraints such as I/O pads location or pre-fixed blocks processing. Practical experiments show the efficiency of these heuristics and allow a comparison between the two placement methods.

Keywords : floorplan design, exploratory data analysis, min-cut partitioning.

Table des matières

Avant-Propos	1
Résumé	3
Summary	5
Table des matières	7
Introduction	13
Le problème du placement	17
I. Présentation du problème.....	19
I.1 Le réseau logique du circuit.....	19
I.2 Les contraintes de placement.....	21
I.3 Les critères à optimiser	22
I.4 Le coût d'un placement.....	24
I.5 Modélisation du problème.....	24
II. Les méthodes de placement.....	25
II.1 Les méthodes constructives	25
II.1.1 Construction incrémentale	25
II.1.2 Dualisation rectangulaire du graphe de connectivité.....	26
II.1.3 Optimisation analytique	28
II.1.4 Bipartitionnements successifs	29
II.1.5 Immersion planaire.....	30
II.2 Les méthodes itératives.....	30
II.2.1 Modélisation par des forces.....	31
II.2.2 Le recuit simulé.....	31
Placement automatique par une méthode d'analyse de données	35
I. L'analyse en composantes principales (ACP).....	37
II. L'analyse d'un tableau de proximités (ATP).....	40
III. Lien entre placement et analyse de données	42
IV. Placement de blocs par ATP.....	43
IV.1 Choix d'une distance entre les blocs	44
IV.1.1 Uniformité des distances.....	44
IV.1.2 Prise en compte de la surface des blocs	45
IV.2 Distance entre blocs non interconnectés.....	50

IV.3	Pondération des blocs.....	54
IV.4	Calcul des vecteurs propres.....	56
V.	Obtention d'un placement réalisable	60
V.1	Décompaction par homothétie	61
V.2	Décompaction par arbre binaire de placement.....	64
V.3	Décompaction excentrique partant du barycentre	67
V.4	Décompaction alternée horizontale et verticale.....	68
V.5	Décompaction par itération de l'ATP.....	71
V.6	Contrainte de faisabilité du routage.....	74
VI.	Amélioration d'un placement.....	74
VI.1	Orientation des blocs.....	74
VI.1.1	Orientation après placement.....	74
VI.1.2	Prise en compte de l'orientation lors de l'ATP.....	76
VI.2	Compaction du circuit.....	78
VI.2.1	Approche analytique	78
VI.2.2	Algorithme implanté.....	82
VII.	Expérimentation.....	83
VIII.	Conclusion	85

Placement automatique par bipartitionnements successifs87

I.	Bipartitionnement d'un ensemble de modules	90
I.1	Modélisation.....	90
I.2	Amélioration d'une bipartition initiale.....	92
I.2.1	Contrainte d'équilibrage.....	93
I.2.2	Gain d'un module	94
I.2.3	Mise à jour des gains.....	96
I.2.4	Amélioration itérative d'une bipartition	99
I.2.5	Choix du module de base	102
I.2.6	Algorithme d'amélioration d'une bipartition initiale	104
I.2.7	Extension du concept de gain d'un module.....	105
	a) Initialisation des gains généralisés	109
	b) Remise à jour des gains généralisés.....	110
I.3	Construction d'une bipartition initiale.....	112
I.3.1	Bipartitionnement du graphe de connectivité sans contrainte de taille	113
I.3.2	Utilisation de l'algorithme d'amélioration itérative.....	115
I.3.3	Approche simple basée sur un tri des connexions.....	117
I.3.4	Contraction du graphe de connectivité par regroupement de sommets	118
II.	Stratégie du placement.....	119

II.1	Divisions successives du circuit	120
II.1.1	Estimation de l'enveloppe du circuit	120
II.1.2	Division des boxes	121
II.2	Bipartitionnement des modules d'un noeud	122
II.3	Orientation des blocs et ajustement des positions	126
II.4	Prise en compte des plots d'entrées-sorties du circuit.....	132
II.5	Prise en compte de blocs fixes sur le circuit.....	134
II.6	Prise en compte des blocs déformables.....	138
II.6.1	Ajustement du rapport de forme des blocs.....	139
II.6.2	Côtés d'attachement des connexions	141
III.	Expérimentation.....	143
III.1	Comparaison avec les résultats du placement par A.T.P	144
III.2	Traitement des blocs fixes et des plots d'E/S	145
IV.	Conclusion.....	145
	Conclusion générale.....	147
	Bibliographie.....	151

Introduction

Depuis le début des années quatre-vingts, les techniques de conception assistée par ordinateur ont permis de créer des systèmes propres à la conception des circuits intégrés. Au départ, ces systèmes ne permettaient d'effectuer que des tâches simples mais fastidieuses pour le concepteur comme le dessin des masques d'un circuit, puis au fur et à mesure des années, ces environnements de C.A.O sont devenus de plus en plus "intelligents", permettant d'automatiser chaque fois un peu plus les tâches de la conception. C'est ainsi qu'à l'heure actuelle, il existe de nombreux outils capables d'effectuer certaines étapes de la conception d'un circuit, comme la synthèse de fonctions booléennes ou de machines d'états finis.

Alors que les résultats fournis par de tels outils sont dans l'ensemble plus que satisfaisants, certaines étapes restent encore peu automatisées.

C'est notamment le cas du placement du circuit. Cette étape est une des dernières qui précède la génération des masques. Elle consiste à disposer les éléments du circuit les uns par rapport aux autres sur le plan de façon à optimiser certains critères. Selon la technologie utilisée pour la réalisation du circuit, le problème du placement peut être différent. Il peut s'agir du placement sur réseau prédiffusé, auquel cas les éléments à placer sont tous de la même taille et doivent être immergés sur une grille. Dans un tel cas, la recherche d'une solution est avant tout guidée par la nécessité de pouvoir effectuer le routage des connexions, l'espace de routage sur le circuit étant imposé par la structure de la grille.

Le problème de la génération de plan de masse pour les circuits "classiques" est lui beaucoup plus général. Il est bien souvent guidé par la minimisation de la surface occupée. Les éléments peuvent avoir des formes et des tailles différentes et leur placement, tout comme le routage des connexions, n'est guidé par aucune structure.

Dans cette thèse, nous nous intéressons au problème de la génération de plan de masse. Un premier chapitre expose de façon générale le problème du placement et passe en revue l'ensemble des méthodes qui peuvent être appliquées à la résolution de ce problème.

Dans le second chapitre, nous nous intéressons à une méthode originale faisant appel à des techniques d'analyse des données. Cette méthode n'ayant pas auparavant fait l'objet d'une recherche très approfondie, nous nous sommes attachés à en faire une étude complète afin de lui apporter toutes les améliorations possibles en vue d'un jugement critique.

La méthode précédente présentant certaines lacunes, nous nous sommes intéressés dans le troisième chapitre à une méthode radicalement différente et

procédant par des divisions successives de l'ensemble des modules. L'état de l'art et l'implantation de cette méthode sont détaillés et des extensions à certains cas particuliers sont proposées.

Enfin une expérimentation effectuée sur le même ensemble de circuits pour les deux méthodes permet de juger de la qualité des résultats obtenus par chaque approche.

CHAPITRE I

Le problème du placement

I. PRÉSENTATION DU PROBLEME

Le problème abordé ici est celui du placement des différents blocs constituant un circuit intégré. Un circuit intégré peut être considéré comme un ensemble de blocs (ou *modules*) et une liste d'interconnexions (ou *équipotentiels*) entre ces blocs.

Le but du placement est alors de disposer les blocs sur la surface hôte du circuit, en respectant diverses contraintes et en optimisant certains critères.

L'obtention d'un bon placement est la condition impérative d'une implantation complète et satisfaisante d'un circuit. Sa qualité dépend de la minimisation des critères considérés, parmi lesquels on peut citer la surface totale occupée, la longueur totale des connexions ou la longueur de certaines connexions critiques après leur routage effectif.

Ces critères ne peuvent être connus précisément qu'après la phase de tracé des connexions; or celle-ci s'effectue toujours après le placement. Ceci rend difficile l'évaluation d'une solution lors du placement, et en fait le domaine de l'implantation automatique "dans lequel les échecs sont les plus courants et les plus frappants au regard des solutions trouvées manuellement" [BELLON 84].

I.1 Le réseau logique du circuit

Un circuit est vu comme un réseau logique dont les éléments sont:

- La liste des blocs à placer :

Chaque bloc est décrit par sa topologie (dimensions, position des points de contacts pour les connexions) et par la liste des équipotentiels connectés à chacun de ses points de contact.

La forme des blocs peut être rectanguloïde dans le cas général (assemblage de rectangles) mais afin de simplifier le problème, nous supposons dans la suite qu'ils sont tous de forme rectangulaire. Les dimensions des blocs sont connues de façon précise dans le cas de blocs durs (blocs déjà synthétisés ou faisant partie d'une bibliothèque, par exemple ROM,PLA,UAL ...), alors que dans le cas de blocs "mous", consistant en un ensemble de cellules standards dont le placement n'est pas déterminé, seule une valeur approximative de la surface est connue. Ainsi les blocs mous peuvent prendre comme forme tout rectangle dont le rapport des côtés ou *ratio* est compris entre des valeurs extrêmes admises pour le bloc. C'est le cas des blocs constitués de cellules standards.

- La liste des équipotentiels :

Chaque équipotentielle du circuit est décrite par la liste des blocs qu'elle connecte. Il existe ensuite plusieurs façons de décrire la connectique globale du circuit :

L'ensemble des équipotentiels peut être représenté sous la forme d'un tableau (t_{ij}) blocs x connexions tel que: $t_{ij}=1$ si la connexion j est reliée à l'élément i , 0 sinon.

Une seconde représentation de la connectique du circuit peut être faite sous la forme d'un graphe dont les sommets sont les blocs. A chaque connexion correspond un ensemble d'arêtes reliant deux à deux les blocs touchés par la connexion, ces arêtes forment donc une clique. Ce type de graphe est appelé *graphe de connectivité* du circuit. Dans le cas d'un circuit dont les connexions peuvent relier plus de deux blocs, cette représentation entraîne une perte d'informations car il est alors impossible d'identifier la liste des connexions à partir du graphe. En fait dans ce cas, il conviendrait de définir un hypergraphe [GONDRAN&MINOUX 85] afin de ne pas se limiter à des relations binaires sur l'ensemble des blocs et permettre à des arêtes de connecter plus de deux sommets. Ainsi on peut définir une bijection qui, à chaque connexion, fait correspondre une arête dans l'hypergraphe et donc déterminer sans ambiguïté la liste des connexions à partir du graphe.

Mais la représentation sous forme d'hypergraphe n'est presque jamais utilisée dans la littérature, on lui préfère la représentation sous forme de graphe biparti suivante :

L'ensemble des blocs et l'ensemble des connexions forment chacun une des classes de sommets du graphe biparti. Chaque arête entre les deux classes correspond à la liaison d'une connexion sur un bloc. Ceci est en fait la représentation sous forme de graphe du tableau (t_{ij}) , chaque t_{ij} non nul correspond à une arête entre les deux classes de sommets et vice versa.

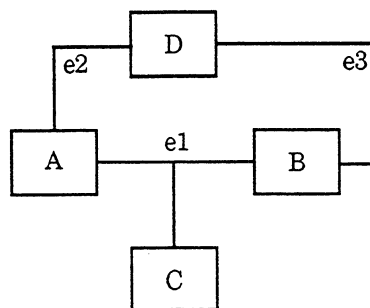


Figure 1 : Exemple de circuit composé de quatre blocs et trois équipotentiels.

	e1	e2	e3
A	1	1	0
B	1	0	1
C	1	0	0
D	0	1	1

Figure 2a

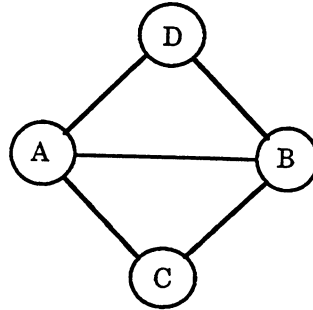


Figure 2b

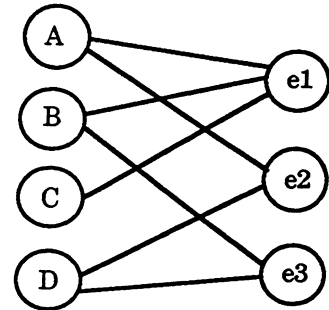


Figure 2c

Figure 2a : Tableau blocs \times connexions correspondant au circuit.

Figure 2b : Graphe de connectivité du circuit, les sommets sont les blocs et les arêtes représentent les connexions entre ces blocs.

Figure 2c : Représentation du circuit sous forme de graphe biparti.

I.2 Les contraintes de placement

- Contraintes de faisabilité du placement :

- Les blocs ne doivent pas se chevaucher. La prise en compte de cette contrainte est grandement facilitée dans le cas de blocs rectangulaires. De plus les côtés des blocs doivent tous être parallèles à un des côtés du circuit, ce dernier ayant une forme rectangulaire. Ceci implique que chaque bloc a 8 orientations possibles si l'on considère les retournements (4 sinon).

- Si une surface hôte est définie pour le circuit, alors le placement final doit pouvoir être contenu à l'intérieur. Cette contrainte est imposée lorsque le boîtier à l'intérieur duquel sera placé le circuit, est déterminé avant le placement.

- Certains blocs peuvent avoir des positions plus ou moins prédéterminées, c'est notamment le cas des blocs d'entrées-sorties (ou *plots d'E/S*) reliés aux broches du boîtier, ou les blocs du circuit fortement connectés à ces plots. Dans le cas des plots d'E/S, ceux-ci doivent être placés à la périphérie du circuit, ils peuvent avoir des positions fixes si la surface hôte du circuit est définie ou avoir simplement des affectations sur les faces du circuit avec éventuellement un ordre.

- Contraintes de faisabilité du routage :

- Le tracé des connexions (ou *routage*) se fait dans des espaces laissés libres entre les blocs et appelés *canaux*. Ces canaux doivent être suffisamment larges pour pouvoir tracer toutes les connexions qui les empruntent. L'estimation des largeurs nécessaires pour les canaux est rendue difficile par le fait que l'itinéraire emprunté par chaque connexion est déterminé après le placement, durant la phase de routage global. Pour cette raison une largeur

arbitraire est fixée pour chaque canal lors du placement et c'est à la suite du routage global que l'on détermine les largeurs suffisantes, une phase de décompaction est alors utilisée afin d'espacer les blocs en conséquence.

I.3 Les critères à optimiser

- Minimisation de la surface du circuit :

La minimisation de la surface du circuit se traduit par la minimisation du rectangle englobant celui-ci, dans le cas où aucune surface hôte n'a été définie. Dans le cas contraire, le placement final doit pouvoir être contenu à l'intérieur de la surface hôte.

- Minimisation de la longueur totale de tracé des connexions :

En minimisant la surface de tracé des connexions, on minimise la surface occupée par le routage dans le circuit, et donc par voie de conséquence la surface globale. Cette minimisation ne peut être faite qu'à travers la minimisation de la longueur totale estimée des connexions car la longueur de tracé ne peut être connue de façon fiable qu'après le routage global. Ceci pose le problème d'une méthode d'estimation rapide et efficace.

Pour estimer la longueur d'une connexion, on ne dispose que des coordonnées de ses points de contact.

Lorsqu'une équipotentielle a deux extrémités, on utilise de préférence à la distance euclidienne entre ces points, la distance de Manhattan qui est plus proche du tracé réel.

Dans le cas d'une équipotentielle ayant plus de deux extrémités, une évaluation couramment utilisée est la mesure du demi-périmètre du rectangle englobant les points de contact.

Un autre estimation, dite du "tronc unique", a l'avantage d'offrir un modèle simple, plus proche de la réalité. Des modèles plus complexes peuvent être utilisés au détriment du temps de calcul: construction d'un arbre de Steiner rectiligne (la recherche de celui de longueur minimum étant un problème NP-complet [GAREY&JOHNSON 77]) ou construction de l'arbre complet minimum (algorithme polynomial).

Évidemment ces méthodes ne tiennent pas compte des possibilités réelles de passage entre les blocs et les détours pour contourner les blocs peuvent être importants.

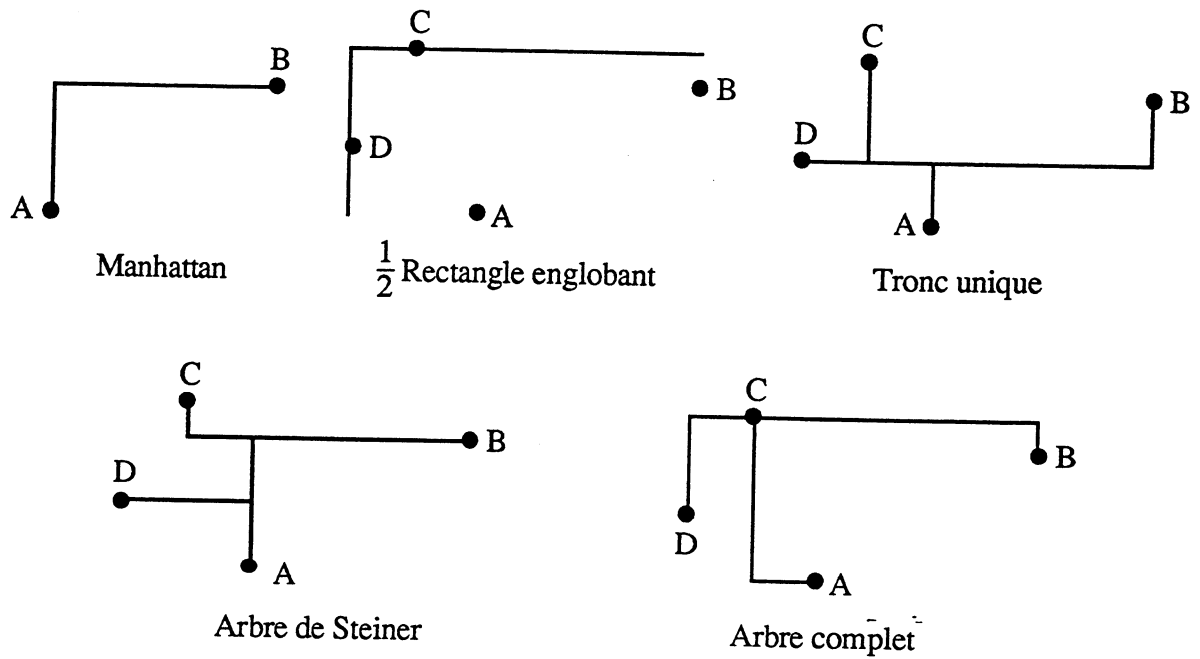


Figure 3 : Les différentes estimations de tracé d'une connexion.

- Minimisation de la longueur maximale des connexions :

Dans ce cas, le but est d'optimiser la propagation des signaux électriques sur les équipotentielles et ainsi d'améliorer les performances du circuit. Ce critère peut paraître analogue au précédant, mais ils sont parfois antinomiques.

Exemple :

Les points B, C, D sont fixes.

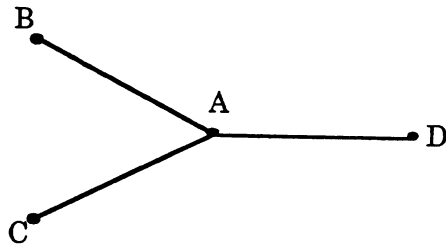


Figure 4a

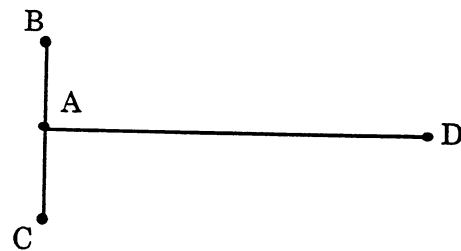


Figure 4b

Figure 4a : position de A pour minimiser la longueur maximale des connexions.

Figure 4b : position de A pour minimiser la longueur totale de connexions.

Lorsqu'il existe des blocs fortement connectés et des blocs faiblement connectés, l'incompatibilité des deux critères devient plus importante. En effet, réduire la longueur totale des connexions conduira à rapprocher entre eux les blocs fortement interconnectés, au détriment des blocs faiblement interconnectés qui pourront se retrouver très éloignés les uns des autres. Ceci est évidemment contraire au critère de minimisation de la longueur maximale

des connexions.

- Minimisation et uniformisation des densités estimées de connexions dans les canaux :

Ce critère est destiné à faciliter le tracé effectif des connexions à l'intérieur de chaque canal et à ne pas trop perturber le placement lors de l'élargissement des canaux. Généralement, son optimisation a tendance à augmenter la longueur des connexions, car elle conduit à déplacer des éléments placés dans des zones denses du circuit vers des zones moins denses.

I.4 Le coût d'un placement

Le coût d'un placement est évalué à partir des critères d'optimisation et une fonction coût est construite, dont les composantes sont les coûts élémentaires associés à chaque critère.

La forme la plus utilisée pour cette fonction est la suivante:

$$F(P) = \sum \alpha_i \cdot C_i$$

où les C_i représentent les coûts des différents critères choisis et les α_i sont des réels caractérisant le *poids* accordé à chaque critère.

Selon les méthodes, on peut être amené à calculer le coût d'un placement en cours de processus, pour juger par exemple d'une modification de celui-ci. Dans ce cas, le placement final dépend du choix de cette fonction de coût, car la réalisation des objectifs d'optimisation est liée au poids qui est accordé aux différents critères.

Nous verrons par la suite que les critères définis au paragraphe précédent ne sont généralement pas tous pris en compte durant le placement et que lorsqu'ils le sont, c'est rarement par une évaluation du coût.

I.5 Modélisation du problème

Si l'on fait l'hypothèse que la surface hôte du circuit est divisée en k localisations possibles, alors le problème du placement peut se modéliser de la façon suivante :

Soit $M = \{m_1, m_2, \dots, m_n\}$ l'ensemble des blocs à placer et soit $S = \{s_1, s_2, \dots, s_k\}$ l'ensemble des k localisations possibles d'un bloc sur le circuit (avec $n \leq k$). Déterminer le placement $P = M \times S$ qui optimise le coût $F(P)$.

Dans le problème ainsi formulé, les contraintes de faisabilité du placement doivent être introduites dans $F(P)$ en tant que critères à minimiser. Ainsi la

surface totale de chevauchement des blocs peut constituer un critère et le débordement de la surface hôte peut en constituer un autre. L'aspect prioritaire de ces critères est alors caractérisé par des poids plus importants dans $F(P)$.

La discrétisation du plan en k localisations possibles fait de l'espace des solutions un ensemble fini et le problème devient un problème d'optimisation combinatoire.

II. LES MÉTHODES DE PLACEMENT

Étant donné la difficulté du problème, il n'existe aucune méthode universelle qui conduise à l'optimum sans faire une recherche exhaustive sur l'espace des solutions. La nécessité d'obtenir des solutions satisfaisantes en des temps raisonnables a donc conduit à rechercher des méthodes approchées, ou heuristiques.

Celles-ci se répartissent en deux principales classes : les méthodes constructives et les méthodes itératives. Les premières construisent pas à pas la configuration du circuit, une disposition globale n'étant donnée qu'à la fin de la phase de construction. Les méthodes de la seconde classe partent d'une configuration initiale des blocs du circuit et tentent de l'améliorer itérativement par des modifications successives de la configuration courante.

II.1 Les méthodes constructives

Ces méthodes ne remettent jamais en cause la position d'un bloc, elles sont entièrement déterministes et pour un problème donné, elles conduisent au même placement à chaque exécution. Elles peuvent considérer le problème localement en plaçant les blocs les uns après les autres ou procéder à une optimisation globale avant de placer tous les blocs simultanément.

II.1.1 Construction incrémentale

C'est la plus simple des méthodes constructives [HANAN, WOLF & AGULE 76], [TSITSIMIS 84]. Le placement est construit pas à pas, à chaque pas un bloc est sélectionné et une position lui est attribuée par rapport aux blocs déjà placés, de façon à minimiser une fonction objective, par exemple la longueur estimée des connexions. Le procédé est répété jusqu'à ce que tous les blocs soient placés. Il est possible d'initialiser la procédure au moyen d'un petit ensemble de blocs jugés critiques ou simplement au moyen des plots d'E/S qui peuvent avoir une position imposée. L'ordre de sélection des blocs est déterminant pour le placement, il peut être fixé avant le lancement de la procédure ou le choix du bloc peut être effectué à chaque itération selon un critère à définir.

Les critères de placement optimisés sont la longueur totale des connexions et la surface, selon la fonction objective choisie. Ils sont évalués sur le placement partiel à chaque itération.

Ce type d'approche est généralement utilisé pour générer un placement initial qui sera ensuite amélioré par une méthode itérative.

II.1.2 Dualisation rectangulaire du graphe de connectivité

Cette méthode considère le graphe de connectivité du circuit dont les sommets sont les blocs. Ce graphe est rendu planaire par l'ajout d'un nombre minimum de sommets sur les croisements d'arêtes. Ensuite la méthode consiste à générer ce qui est appelé le graphe dual rectangulaire, ce graphe correspond à une configuration du circuit car chacune de ses faces correspond à un sommet du graphe de connectivité. La construction du dual peut être réduite à un problème de couplage dans un graphe biparti [LAI&LEINWAND 88].

Lorsque le graphe planaire satisfait certaines conditions, il est fréquent qu'il admette plusieurs duals rectangulaires. Le meilleur dual est celui qui donnera le circuit de taille minimum, une méthode rapide d'estimation est donnée dans [LOKANATHAN&KINNEN 89].

Les conditions d'existence du dual rectangulaire d'un graphe planaire sont que celui-ci soit triangulaire [KOZMINSKI&KINNEN 84] et qu'il n'admette pas de triangles complexes (triangles contenant des sommets) [LEINWAND&LAI 84]. Pour cela des arêtes sont rajoutées au graphe planaire jusqu'à ce que toutes les régions soient triangulaires, ensuite l'élimination des triangles complexes se fait par l'ajout de sommets et d'arêtes. Une fois les conditions satisfaites, la construction du dual rectangulaire peut être réalisée par un algorithme de complexité linéaire [BHASKER&SAHNI 86].

L'exemple suivant montre les différentes étapes de cette méthode.

Exemple:

Figure 5a: graphe de connectivité du circuit, le sommet X est un sommet fictif pour les connexions avec l'extérieur du circuit.

Figure 5b: La triangularisation du graphe entraîne l'ajout des arêtes (A,I) et (I,X). Pour supprimer les triangles complexes D-E-I et I-B-E, le sommet L est ajouté.

Figure 5c: Les angles du circuit peuvent être attribués aux sommets D, E, P, B, I. On décide de placer P et B chacun dans un angle, et D sur les deux restants.

Figure 5d: Dual rectangulaire correspondant à une configuration du circuit. La face L correspond à un canal d'interconnexion.

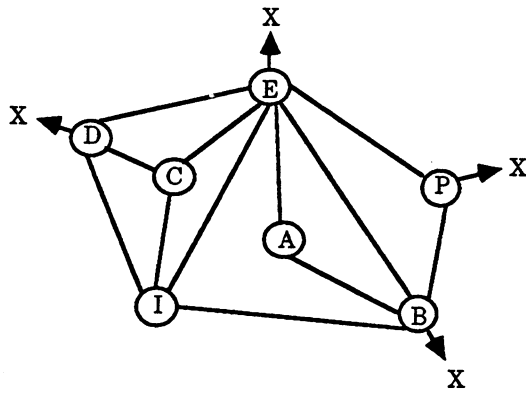


Figure 5a

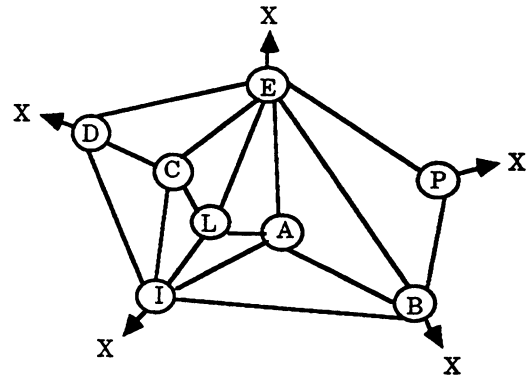


Figure 5b

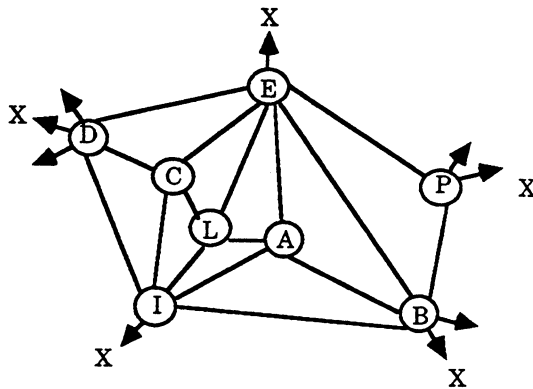


Figure 5c

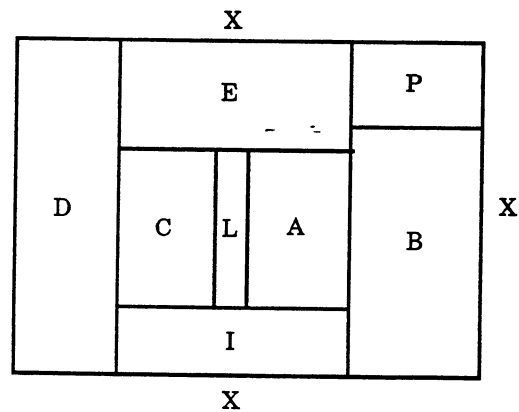


Figure 5d

Il n'y a pas, dans cette méthode, minimisation d'une fonction coût représentant les critères de placement à optimiser. Néanmoins, l'objectif visé est de placer sur des positions adjacentes les éléments interconnectés, de ce fait le critère d'optimisation de la longueur totale des connexions est minimisé, mais sans évaluation directe durant l'algorithme.

On constate que les conditions pour la génération du dual rectangulaire sont très restrictives. Tout d'abord le graphe de connectivité du circuit doit être planaire, or ceci n'arrive que dans de très rares cas, quand le circuit est extrêmement simple. Il est donc nécessaire, dans la plupart des cas, de planariser ce graphe en rajoutant un nombre minimum de sommets, ceci constitue la première difficulté, d'autant plus que le résultat dépendra de la représentation initiale du graphe de connectivité que l'on aura faite sur le plan. Ensuite, le processus de triangularisation du graphe conduit également à rajouter des sommets ainsi que des arêtes.

Selon le graphe de connectivité du circuit, on risque d'aboutir, après ces différentes opérations, à un graphe triangulaire ne correspondant plus du tout à la spécification du circuit. Le dual rectangulaire est alors difficile à exploiter

pour obtenir le placement du circuit.

Cette méthode est donc à réserver aux circuits ayant une connectique "aérée", dont les équipotentielles connectent peu de blocs à la fois, car leur représentation sous forme de clique dans le graphe de connectivité pose de gros problèmes pour la planarisation (une clique de cinq sommets n'est pas planaire).

Un second désavantage réside dans le fait que les dimensions des blocs ne sont pas prises en compte, de ce fait il n'est pas forcément possible de reproduire le graphe rectangulaire avec les blocs du circuit si ceux-ci ont des dimensions figées. Il est donc préférable que l'ensemble des blocs du circuit soient des blocs déformables.

II.1.3 Optimisation analytique

Le problème du placement de blocs rectangulaires est formulé comme un problème d'optimisation analytique sous contraintes. Les contraintes de faisabilité du placement, comme le non recouvrement des blocs et leur positionnement à l'intérieur de la surface hôte, sont décrites analytiquement sous forme de systèmes d'inéquations faisant apparaître les coordonnées des blocs et leurs dimensions. Une fonction objective représentant la longueur de connexion est construite. Le tout forme généralement un problème quadratique d'optimisation. Celui-ci peut être minimisé en résolvant un système d'équations linéaires par des algorithmes itératifs [RANKE&JOHANNES 89], [KLEINHANS,SIGL&JOHANNES 88 et 89], [HERRIGEL &FICHTNER 89].

Le problème du placement de blocs rectangulaires peut également être formulé analytiquement comme un problème de minimisation sans contrainte [SHA&DUTTON 85],[YING&WONG 89] en incorporant les contraintes de placement dans la fonction objective. La fonction est alors minimisée par des méthodes itératives dites "quasi-Newton" dont les algorithmes SSVM (Self-Scaling Variable Metric) [OREN 73].

Dans ces deux cas, seul le critère de longueur totale des connexions est minimisé, la fonction objective constituant une évaluation directe de ce dernier tout au long de l'algorithme.

Des problèmes d'instabilité numérique peuvent se poser lors de la minimisation itérative.

Une modélisation différente est proposée dans [SUTANTHAVIBUL, SHRAGOWITZ&ROSEN 90], elle conduit à un problème de programmation linéaire dont certaines variables sont entières et les autres réelles. Le temps de résolution d'un tel système est exponentiel; pour cette raison le problème est décomposé en sous-problèmes selon une stratégie définie et chaque sous-problème est résolu de façon optimum. Cette modélisation permet, quant à

elle, de prendre en compte la minimisation de la surface du circuit.

Généralement les méthodes d'optimisation analytique doivent être suivies d'un déchevauchement des blocs.

II.1.4 Bipartitionnements successifs

Plus connue sous le nom de Min_Cut Partitioning [BREUER 77],[LAUTHER 79],[FIDUCCIA&MATTHEYSES 82],[LAPOTIN&DIRECTOR 86], cette technique consiste en une division successive de la surface rectangulaire du circuit en sous-rectangles, au moyen de lignes de coupe. Pour cela la surface du circuit doit être estimée, si elle n'est pas donnée comme une contrainte, en prenant par exemple une somme pondérée des surfaces des blocs composant le circuit.

L'ensemble des blocs est bipartitionné en deux classes de façon à minimiser le nombre de connexions passant d'une classe à l'autre et l'aire du circuit est découpée en deux zones au moyen d'une ligne de coupe horizontale ou verticale. A chacune des deux zones est affectée une classe issue de la bipartition des modules. La position de la ligne de coupe est déterminée par la taille totale des modules de chaque classe.

Le même processus est récursivement appelé au sein de chaque zone jusqu'à ce que chaque bloc se trouve isolé dans une zone. Les lignes de coupe correspondent alors aux canaux d'interconnexions du placement ainsi obtenu.

Lors du bipartitionnement d'un ensemble de modules, la fonction objective à minimiser est le nombre de connexions entre les deux classes de la bipartition. Bien que ne correspondant à aucun des critères de placement à optimiser, ceci conduit à une minimisation des densités de connexions dans les canaux et à une minimisation de la longueur totale de connexions du placement final.

Un tel placement peut être représenté par un arbre binaire. Chaque noeud de l'arbre est constitué d'un ensemble de modules et de la position de cet ensemble sur le circuit, la racine de l'arbre étant l'ensemble de tous les modules. Un noeud ne peut avoir que deux fils, ceux-ci correspondent aux classes de modules issues de la bipartition des modules du noeud père. Chaque feuille de l'arbre est constituée d'un bloc et de son affectation sur le circuit.

L'arbre ainsi obtenu et le placement qui lui est associé sont connus dans la littérature sous les noms de "slicing tree" et "slicing placement".

Exemple :

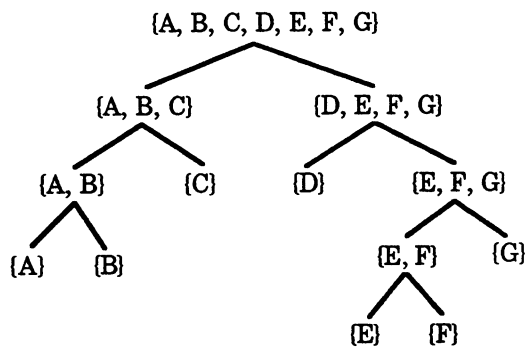


Figure 6a

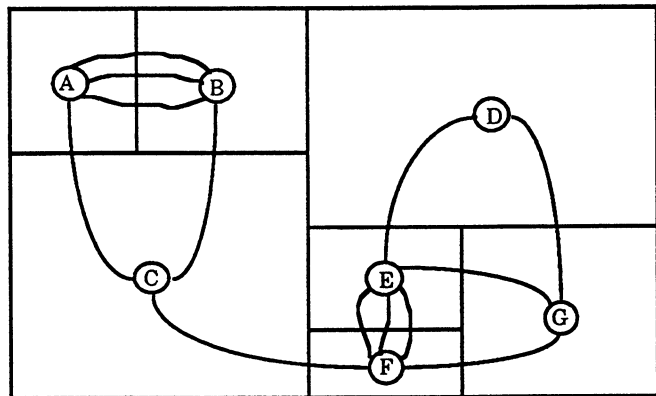


Figure 6b

Figure 6a : arbre binaire de placement.

Figure 6b : placement par bipartitionnements successifs et graphe de connectivité associé.

II.1.5 Immersion planeaire

Les modules sont considérés comme des points positionnés dans l'espace euclidien de dimension $n-1$ (si n est le nombre total de modules). Les proximités des modules les uns par rapport aux autres sont définies par rapport à leur connectivité selon le principe suivant: deux points sont d'autant plus proches que les modules correspondants ont des connexions en commun. La disposition des modules dans cet espace est jugée idéale par rapport à la connectivité et le but de la méthode est d'en donner une image planeaire aussi fidèle que possible [OTTEN 82].

Une technique statistique d'analyse de données (l'analyse factorielle d'un tableau de distances) [BARRA&Co. 87], [KOUKA&SAUCIER 87] peut être utilisée pour projeter sur un plan le nuage des points disposés dans l'espace, de façon à perturber le moins possible les distances d'origine. Les coordonnées des centres des blocs sur ce plan sont obtenues à la suite d'un calcul de vecteurs propres, sans l'optimisation d'aucune fonction objective. Une propriété théorique assure que le plan de projection est celui qui conserve le mieux les distances.

Ceci fournit un placement relatif pour l'optimisation de la longueur totale de connexion, qu'il convient de transformer en positionnement définitif tenant compte de la taille des modules, afin d'éviter leur chevauchement.

Cette méthode fait l'objet d'une partie de cette thèse et sera étudiée en détail dans le deuxième chapitre.

II.2 Les méthodes itératives

Comme il a été dit précédemment ces méthodes opèrent de la façon suivante:

partant d'une configuration initiale des blocs, on choisit à chaque itération un ensemble de blocs et un ensemble de transformations à appliquer à ces blocs (translations, rotations, échanges de positions, permutations ...). Un nouveau placement est alors obtenu et son coût est calculé afin de décider si cette nouvelle configuration est conservée ou non. Si c'est le cas, le processus de modification recommence en partant de ce nouveau placement, sinon, de nouvelles transformations sont faites sur l'ancien placement jusqu'à ce qu'un nouveau placement soit accepté.

Les itérations se poursuivent jusqu'à satisfaction de critères d'arrêt définis à l'avance.

II.2.1 Modélisation par des forces

Les méthodes dirigées par des forces [QUINN&BREUER 79] appliquent un modèle physique. Pour chaque paire de blocs connectés, les connexions entre ces deux blocs sont assimilées à un ressort dont la raideur est proportionnelle au nombre de connexions en commun. Ceci définit donc des forces d'attraction entre les blocs. Il suffit de les laisser se déplacer librement selon les forces pour converger vers un équilibre où la résultante des forces sur chaque bloc est nulle, et donc où tout mouvement cesse. Cela est réalisé itérativement en déplaçant les modules un par un. A chaque itération le module sur lequel s'exerce la plus importante résultante des forces est sélectionné, il est déplacé vers le point qui minimise cette résultante.

Pour éviter les chevauchements de blocs il est nécessaire d'introduire des forces de répulsion entre ceux-ci, en pratique ceci est fait après quelques itérations afin de laisser s'établir les positions relatives des blocs [FORBES 87].

Cette approche n'effectue pas d'optimisation de la surface du circuit, seule la connectique est prise en compte. Néanmoins, il est difficile de dire quel type de minimisation est faite sur celle-ci. Si l'on reprend l'exemple du paragraphe I.3, il apparaît qu'avec un poids égal des trois connexions, le système de ressorts prendra une configuration proche de celle qui minimise la longueur maximale des connexions. En revanche, si le nombre de connexions entre A et B, et entre A et C est plus important que celui entre A et D alors la configuration prise se rapprochera de celle minimisant la longueur totale des connexions.

II.2.2 Le recuit simulé

Le recuit simulé est une technique stochastique générale utilisée dans beaucoup de problèmes d'optimisation combinatoire [KIRKPATRICK,GELATT &VECCHI 83], [NAHAR,SAHNI&SHRAGOWITZ 86]. Il repose sur une analogie avec la cristallisation des solides.

Le système des blocs et de leurs interactions dues à la connectique est assimilé

à un système de particules élémentaires en agitation thermique, interagissant entre elles par leurs forces d'attraction. La recherche d'un placement optimal correspond alors à l'obtention d'une disposition cristalline des particules. Or cette dernière est obtenue physiquement en abaissant très progressivement la température des particules, et en l'augmentant parfois lorsque des défauts sont apparus.

Ceci se traduit par les points suivants pour le placement: les modifications apportées à la disposition courante lors d'une itération doivent être de moins en moins importantes au fur et à mesure de l'algorithme (l'agitation diminue avec la température). Pour cela l'évolution de l'algorithme est paramétrée par une fonction température qui est abaissée selon une stratégie donnée. Pour chaque valeur de ce paramètre un nombre suffisant d'itérations doit être réalisé afin de permettre une stabilisation. Enfin l'algorithme doit permettre de s'extraire des minima locaux. Ceci n'est pas réalisé comme dans le processus physique par une élévation de la température, mais simplement en acceptant avec une certaine probabilité les configurations qui dégradent le placement courant. Cette probabilité doit décroître avec la baisse de la température.

D'où la modélisation suivante [OTTEN&VAN GINNEKEN 84]:

- L'énergie interne du système de particules est assimilé au coût du placement.
- l'ampleur d des modifications globales (longueur maximale d'une translation, distance maximale entre deux éléments échangés) dépend du coefficient T de température.
- Pour chaque modification du placement:
 - Si l'énergie E du système (le coût du placement) est abaissée, la transformation est acceptée systématiquement,
 - Sinon, la transformation est acceptée avec une probabilité $e^{(-\Delta E/kT)}$ où k est une constante définie empiriquement.

L'optimisation du placement est obtenue en générant aléatoirement des modifications selon ce principe, lors de phases successives de coefficient T décroissant [CHAISEMARTIN&Co 89]. Tout au long de l'algorithme, la fonction objective devant être minimisée correspond au coût du placement tel qu'il est donné au paragraphe I.4. Elle est calculée après chaque modification par évaluation des différents critères.

La prise en compte des contraintes de faisabilité du placement, comme le non chevauchement des blocs ou le non débordement de la surface du circuit, est réalisée en rajoutant des critères à minimiser dans la fonction objective. Ceux-ci reflètent la surface totale de chevauchement dans la configuration courante, ou le débordement de la surface hôte. Les poids qui leurs sont attribués sont plus importants que les autres, de façon à favoriser le respect des contraintes.

Le recuit simulé nécessite des temps d'exécution très longs. Il est très sensible aux valeurs des coefficients T , k et d , or celles ci sont fixées de façon expérimentale. En particulier, l'évolution de la température conditionne énormément le bon fonctionnement de l'algorithme, aussi bien en durée d'exécution qu'en qualité du résultat [ROSE,KLEBSCH&WOLF 90]. Enfin comme il a déjà été dit au paragraphe I.4, la fonction de coût d'un placement doit être bien ajustée (choix des différents critères et de leur poids associé) car c'est elle qui détermine les objectifs du placement.

Tout cela fait que le recuit simulé, bien qu'ayant été démontré théoriquement comme une méthode convergeant vers l'optimum, reste très délicat à manipuler.

Après ce bref panorama des principales méthodes utilisées pour résoudre le problème du placement, une remarque intéressante peut être faite:

On constate que les différents critères à optimiser lors de la construction d'un placement (surface, longueur totale des connexions, longueur maximale d'une connexion ...) ne sont estimés que par trois techniques. Parmi elles, seule la construction incrémentale et le recuit simulé visent à une minimisation de la surface du circuit. L'optimisation analytique n'utilise généralement dans sa fonction objective que le critère de longueur totale des connexions.

Les autres méthodes conduisent à des minimisations des critères de connectivité mais sans en faire une minimisation directe durant l'élaboration du placement. Ce n'est pas pour autant qu'elles donnent forcément de moins bons résultats.

CHAPITRE II

**Placement automatique par une méthode
d'analyse de données**

L'analyse de données est une branche de la statistique qui concerne l'étude d'un grand nombre d'individus décrits par un ensemble de caractères. Son but est de permettre une étude globale des individus et des caractères en utilisant généralement des représentations graphiques suggestives.

Les fondements de l'analyse de données ont été élaborés au tout début du siècle par C.Spearman et K.Pearson. H.Hotelling dans les années 30 développa leurs travaux pour donner naissance à la plus ancienne des méthodes d'analyse de données, l'analyse en composantes principales.

Aujourd'hui les méthodes d'analyse de données sont utilisées dans des domaines aussi différents que les sciences physiques, naturelles, sociales ou humaines. Elles permettent d'obtenir des résumés synthétiques de l'information présente à l'intérieur de grands tableaux de données.

En ce qui nous concerne nous utiliserons une technique proche de l'analyse en composantes principales afin d'obtenir un placement par la méthode dite "d'immersion planaire" (Cf chapitre précédent). Dans un premier temps nous décrirons brièvement l'analyse en composantes principales et sa dérivée, l'analyse d'un tableau de proximités; pour une étude plus approfondie de ces techniques le lecteur pourra se référer aux ouvrages [BOUROCHE & SAPORTA 80], [DIDAY & Co. 83].

Ensuite, nous détaillerons l'utilisation de l'analyse d'un tableau de proximités dans le problème du placement de blocs. Nous étudierons les différentes possibilités de définir les distances entre les blocs, en fonction de leurs connexions et de leur surface. Nous justifierons le choix pris pour l'implantation, puis la recherche d'une bonne pondération des blocs sera abordée. Enfin, les techniques de compaction-décompaction et d'orientation pour l'obtention d'un placement réalisable seront étudiées et comparées. Une expérimentation sur plusieurs circuits permettra de juger des résultats obtenus.

I. L'ANALYSE EN COMPOSANTES PRINCIPALES (ACP)

C'est la méthode de base de l'analyse de données, elle a pour objet la description de l'information contenue dans un tableau (individus x caractères) où p caractères sont mesurés sur n individus. Les caractères doivent être tous quantitatifs, c'est-à-dire ne prendre que des valeurs numériques sur l'ensemble des individus.

On note X le tableau à n lignes et p colonnes représentant les valeurs prises par les p caractères sur chacun des n individus :

$$X = \begin{bmatrix} x^1(1) & \dots & x^p(1) \\ \vdots & & \vdots \\ x^1(n) & & \vdots \end{bmatrix}$$

Les p caractères sont notés x^1, \dots, x^p . La valeur prise par le caractère x^j sur l'individu i est notée $x^j(i)$. Donc la ligne i de X représente l'ensemble des valeurs des p caractères sur le i -ème individu, tandis que la colonne j représente les valeurs du j -ème caractère sur l'ensemble des n individus.

On voit que chaque individu est défini par p coordonnées, il est donc considéré comme un vecteur d'un espace vectoriel \mathbb{R}^p à p dimensions appelé *espace des individus*. Ainsi chaque individu e_i est identifié au vecteur e_i de composantes $(x^1(i), \dots, x^p(i))$.

Lorsqu'il n'y a que deux caractères on peut représenter l'ensemble des individus sur un plan. Il est alors facile d'analyser les liaisons entre les deux caractères et de repérer les individus ou groupes d'individus présentant des caractéristiques voisines.

S'il y a trois caractères, l'étude visuelle du nuage des individus est encore possible en faisant de la géométrie dans l'espace. Mais dès que le nombre de caractères devient supérieur ou égal à quatre, cela devient impossible. Si l'on désire représenter tous les individus sur un plan, on introduira forcément des distorsions par rapport au nuage, ces distorsions devront être minimales.

La représentation planaire du nuage s'obtiendra en projetant les points individus e_1, \dots, e_n sur un plan. Afin de réduire au minimum les distorsions on choisira le plan de projection sur lequel les distances entre les points seront en moyenne le mieux conservées. Comme l'opération de projection raccourcit toujours les distances, on se fixera pour but de rendre maximale la moyenne des carrés des distances entre les projections des individus. Le plan de projection qui maximise ce critère est appelé le plan principal.

La distance entre deux individus est mesurée par une métrique M , qui peut être n'importe quelle matrice symétrique définie positive, de dimension p . Cette métrique définit le produit scalaire entre deux individus e_1 et e_2 par:

$$\langle e_1, e_2 \rangle_M = e_1 \cdot M \cdot e_2$$

La distance entre e_1 et e_2 est alors définie comme:

$$d(e_1, e_2) = \sqrt{\langle e_1 - e_2, e_1 - e_2 \rangle_M}$$

Si l'on veut se ramener à la distance classique dérivée de la formule de Pythagore il suffit de prendre comme métrique M la matrice identité I . Les

Alors on a:

$$\sum_{i,j=1}^n p_i \cdot p_j \cdot d^2(c^J(i), c^J(j)) = 2 \sum_{j \in J} \lambda_j$$

La partie gauche de l'égalité constitue le critère à maximiser lors de la projection sur le plan principal. Or en projetant sur un plan on a: $|J|= 2$. Donc afin de maximiser le critère on doit prendre les deux plus grandes valeurs des λ_j .

On en conclut que les composantes principales c^1 et c^2 sont les deux vecteurs propres associés aux deux plus grandes valeurs propres de la matrice $X.M.^tX.D$. Ceci définit les coordonnées des projections des n individus sur le plan principal, et donc maximise le critère:

$$\sum_{i,j=1}^n p_i \cdot p_j \cdot d^2(f_i, f_j)$$

où les f_i représentent les projections des individus e_i sur ce plan.

Si l'on ordonne par ordre décroissant les valeurs propres de la matrice $X.M.^tX.D$ ($\lambda_1 > \lambda_2 > \dots > \lambda_p$) alors le quotient $(\lambda_1 + \lambda_2) / (\lambda_1 + \lambda_2 + \dots + \lambda_p)$ est appelé *part d'inertie cumulée des deux premiers axes principaux*. Il mesure l'aplatissement du nuage sur le plan principal. Plus sa valeur est grande, meilleure est la représentation du nuage sur ce plan. Une part d'inertie ayant pour valeur 1 signifie une représentation exacte du nuage sur le plan, c'est-à-dire que l'espace des individus peut être ramené à un espace de dimension 2.

II. L'ANALYSE D'UN TABLEAU DE PROXIMITÉS (ATP)

Suivant les domaines d'application on ne connaît pas forcément les valeurs prises par les caractères sur l'ensemble des individus mais seulement les *distances* entre individus.

Les données constituent donc un *tableau de distances* entre les n individus. Supposons que ces distances soient euclidiennes, c'est-à-dire que les n individus peuvent être considérés comme des points d'un espace vectoriel de dimension p (inconnu) muni d'une métrique M telle que les distances entre les points vérifient:

$$\forall i, j \quad d_{ij} = \sqrt{\langle e_i - e_j, e_i - e_j \rangle_M}$$

Si on connaissait les coordonnées des points dans un repère arbitraire de l'espace vectoriel, alors on pourrait se ramener à un tableau X (individus \times caractères) et ainsi effectuer une analyse en composantes principales. Nous

venons de voir que les composantes principales qui constituent la liste des coordonnées des individus sur les axes principaux sont alors les vecteurs propres de la matrice $X.M.^tX.D$.

Or il est possible d'obtenir la matrice $X.M.^tX$ à partir du tableau des distances entre individus.

En effet, le produit de matrices $X.M.^tX$ est une matrice W dont les éléments w_{ij} sont les produits scalaires $\langle e_i, e_j \rangle_M$, et les éléments diagonaux w_{ii} ont pour valeur $\|e_i\|^2$.

D'après la relation du triangle on a:

$$\begin{aligned} d_{ij}^2 &= \|e_i\|^2 + \|e_j\|^2 - 2\|e_i\|\|e_j\|\cos(e_i, e_j) \\ &= w_{ii} + w_{jj} - 2w_{ij} \end{aligned}$$

En appliquant cette relation à tous les couples d'individus on arrive à la formule:

$$w_{ij} = \frac{1}{2} (d_{i*}^2 + d_{j*}^2 - d_{ij}^2 - d_{**}^2)$$

où

$$d_{i*}^2 = \sum_{j=1}^n p_j \cdot d_{ij}^2 \quad \text{et} \quad d_{**}^2 = \sum_{i,j=1}^n p_i \cdot p_j \cdot d_{ij}^2$$

On voit donc que l'on peut obtenir les composantes principales uniquement à partir du tableau des distances entre individus et de la matrice diagonale D des poids des individus, et ainsi obtenir une représentation de ceux-ci sur un plan. Il suffit pour cela de calculer la matrice W puis de déterminer les vecteurs propres du produit $W.D$.

Si la distance entre les individus est réellement une distance euclidienne alors toutes les valeurs propres de la matrice $W (= X.M.^tX)$ sont positives ou nulles. S'il existe des valeurs propres négatives, alors l'hypothèse que l'on se trouve dans un espace euclidien est fautive, et le tableau de données ne peut plus être considéré comme un tableau de distances, mais plutôt comme un tableau définissant des proximités entre les individus. Pour obtenir quand même une représentation graphique on transforme les mesures de proximités en distances euclidiennes en respectant la contrainte d'ordre suivante:

Si d est la mesure de proximité et $f(d)$ sa modification alors on doit avoir:

$$d_{ij} < d_{kl} \quad \Rightarrow \quad f(d_{ij}) < f(d_{kl}) \quad \forall i, j, k, l$$

Un exemple d'une telle transformation est donné par:

$$\forall i \neq j \quad [f(d_{ij})]^2 = [f(d_{ji})]^2 = d_{ij}^2 + 2 \cdot |\lambda_r|$$

$$\forall i \quad f(d_{ii}) = 0$$

où λ_r désigne la plus petite valeur propre négative de la matrice W.D .

Il est ensuite possible d'obtenir une représentation planaire des individus décrits par le nouveau tableau de distances en recommençant une ATP sur celui-ci.

III. LIEN ENTRE PLACEMENT ET ANALYSE DE DONNÉES

Comme nous l'avons vu dans le chapitre précédent, le seul critère utilisable par une méthode constructive pour guider le placement des blocs les uns par rapport aux autres est le critère de la connectivité. Ceci vient du fait qu'il n'y a pas de rapport direct entre le placement d'un bloc par rapport à un autre et la conséquence que cela aura sur les autres critères, notamment sur la surface finale du circuit.

L'utilisation de l'analyse de données dans le problème de placement ne fera pas exception: si l'on considère la description du réseau logique du circuit sous la forme d'un tableau (blocs x connexions) alors le rapprochement avec une description (individus x caractères) est évident. En assimilant chaque bloc à un individu on obtient une disposition des blocs en nuage de points de l'espace $\{0,1\}^p$, où p désigne le nombre d'équipotentiels. Les coordonnées du i -ème bloc dans cet espace sont données par le p -uplet correspondant à la ligne i de la matrice (blocs x connexions).

Cette représentation des blocs sous la forme d'un nuage reflète une disposition idéale de ceux-ci par rapport à la connectivité du circuit. A partir de là, l'utilisation de l'analyse en composantes principales semble s'imposer.

Malheureusement la particularité des caractères de ne prendre comme valeur que 0 ou 1 quel que soit l'individu n'est pas adaptée à l'analyse en composantes principales. En effet cette particularité est caractéristique de caractères qualitatifs que l'on aurait rendus quantitatifs en les décomposant selon chacune de leurs modalités (par exemple le caractère couleur des yeux pourrait se décomposer selon ses modalités: rouge, bleu, vert, marron; chacune d'entre elles étant un caractère quantitatif ne prenant comme valeur que 0 ou 1, et éventuellement toujours 0 ...).

D'où l'utilisation dans notre cas, d'une approche consistant à définir un tableau de proximités entre les blocs, en fonction de leur connectivité, puis à effectuer l'analyse de ce tableau par la méthode décrite dans le paragraphe précédent. Cette approche permet en outre d'intervenir sur les proximités pour agir sur le résultat de la projection.

IV. PLACEMENT DE BLOCS PAR ATP

Comme il vient d'être dit, l'analyse d'un tableau de proximités est utilisée dans le cadre du placement de blocs, comme méthode d'optimisation du critère de longueur totale des connexions: on cherche à rapprocher les blocs en fonction du nombre d'équipotentiels qui les relie. L'ATP comme tout autre méthode d'analyse de données considère chaque individu comme un point, de ce fait chaque bloc du circuit sera assimilé à son centre.

Cette technique n'étant que l'outil qui nous permettra d'obtenir une représentation planaire des centres des blocs à partir d'une matrice de proximités de ceux-ci, il convient avant tout de définir de façon adéquate cette matrice.

En premier lieu, les proximités que l'on va définir doivent respecter les propriétés de base d'une distance, à savoir:

$$\begin{aligned}d_{ij} &\geq 0 & \forall i, j \\d_{ij} &= d_{ji} & \forall i, j \\d_{ii} &= 0 & \forall i\end{aligned}$$

Ceci étant établi nous devons respecter la règle suivante: deux éléments doivent être d'autant plus proches qu'ils ont d'équipotentiels en commun.

La première idée qui vient à l'esprit est de définir une distance qui soit inversement proportionnelle au nombre de connexions entre les blocs:

$$\begin{aligned}d_{ij} &= \frac{1}{N_{ij}} & \forall i \neq j \\d_{ii} &= 0 & \forall i\end{aligned}$$

où N_{ij} est le nombre de connexions entre les blocs i et j .

Afin d'éviter une valeur infinie de d_{ij} quand les blocs i et j n'admettent pas de connexion commune, on rajoute une constante au dénominateur.

$$d_{ij} = \frac{1}{k + N_{ij}} \quad \forall i \neq j, \quad k = \text{constante choisie empiriquement.}$$

Cette distance permet de valider la méthode de placement par ATP en prenant $k=1$. On se trouve alors dans le cas où toutes les distances sont comprises dans l'intervalle $[0,1]$.

La matrice $W.D$ (avec $W=X.M.^tX$) dont les vecteurs propres sont les composantes principales est obtenue à partir de la matrice des distances inter-individuelles et de la matrice diagonale des poids attribués aux blocs (voir partie II). Aucun critère n'étant défini concernant l'attribution des poids aux blocs, on choisit la configuration équi-répartie suivante:

$$p_i = \frac{1}{n} \quad \forall i,j$$

où n désigne le nombre de blocs dans le circuit.

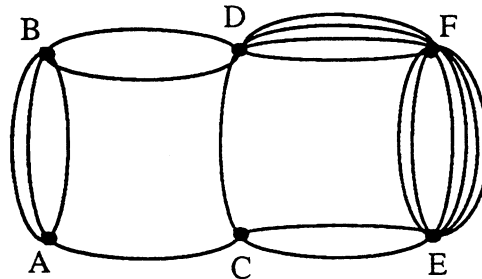
Il est évident que la disposition des blocs sur le plan dépend de façon cruciale de la matrice des distances constituée, de ce fait nous allons étudier, dans ce qui suit, les différentes modifications envisageables.

IV.1 Choix d'une distance entre les blocs

IV.1.1 Uniformité des distances

La formule que nous venons de voir définit la distance entre deux blocs de façon absolue par rapport au nombre de connexions qu'ils partagent. Ceci conduit à rapprocher d'autant plus les blocs qu'ils ont d'équipotentielles en commun, mais certains cas particuliers soulèvent une équivoque. Considérons l'exemple suivant:

Exemple :



$$d_{ij} = \frac{1}{1 + N_{ij}} \quad \forall i \neq j \quad \Rightarrow \quad d(A,B) = \frac{1}{4} \quad d(E,F) = \frac{1}{7}$$

Les blocs A et B partagent la moitié des connexions qui leur sont incidentes, il en est de même pour les blocs E et F, il peut alors paraître normal que pour chaque couple les distances soient identiques. Pourtant si l'on utilise la formule évoquée précédemment alors les distances obtenues ne sont pas égales. Pour remédier à cela on peut utiliser une mesure de proximité différente, se rapprochant de la distance du khi-2 utilisée en statistique, et définie comme il suit:

$$d_{ij} = \frac{\text{nombre de connexions incidentes à A ou B}}{\text{nombre de connexions reliant A et B}}$$

$$\Leftrightarrow d_{ij} = \frac{N_{*i} + N_{*j} - N_{ij}}{N_{ij}}$$

où N_{*i} = nombre de connexions reliées au bloc i .

N_{ij} = nombre de connexions partagées par les blocs i et j .

Avec cette mesure les distances entre les blocs A et B, et les blocs E et F

deviennent identiques.

Cependant on peut se demander si le fait de rapporter le nombre de connexions partagées par deux blocs au nombre total de connexions incidentes, sera bénéfique pour l'optimisation de la longueur totale des connexions. En effet, si l'on considère le placement final, il est préférable que E et F soient proches et A et B éloignés, plutôt que l'inverse.

Or lors de la projection sur le plan principal, l'ATP cherche à maximiser le critère $\sum_{i,j=1}^n p_i \cdot p_j \cdot d^2(f_i, f_j)$ où les f_i sont les projections des points sur le plan.

Comme l'opération de projection raccourcit toujours les distances, ceci revient à réduire le moins possible les grandes distances.

Si les distances $d(A,B)$ et $d(E,F)$ sont égales dans la matrice des proximités alors l'optimisation du critère de projection risque de réduire davantage la distance entre A et B que la distance entre E et F, ce qui n'est pas souhaitable. En revanche, si $d(A,B) > d(E,F)$ dans la matrice de proximité, alors l'optimisation du critère de projection conduira à rechercher un plan de projection qui réduit moins la distance entre A et B que entre E et F. Il en résultera que la minimisation de la distance entre E et F sera prioritaire sur la minimisation de la distance entre A et B, ce qui paraît plus approprié.

Il apparaît donc que l'utilisation de cette distance n'est pas justifiée en ce qui concerne l'optimisation globale de la connectique.

IV.1.2 Prise en compte de la surface des blocs

Comme nous l'avons dit précédemment, lors du traitement par analyse du tableau de proximités, les blocs sont considérés comme des points dans l'espace. Une fois la projection sur le plan principal effectuée, ces points représenteront les positions des centres des blocs sur le circuit. Un traitement ultérieur sera alors nécessaire afin d'obtenir la disposition finale des blocs sur le circuit. Dans ce traitement sera effectuée une phase de décompaction afin de tenir compte de la contrainte de non recouvrement des blocs (voir chapitre précédent). Cette décompaction sera facilitée si les distances entre les centres sont du même ordre de grandeur que les dimensions des blocs, d'où l'utilité d'une prise en compte de la surface des blocs lors de la définition des proximités.

Considérons les centres des blocs positionnés sur le plan, alors le respect de la contrainte suivante implique le non recouvrement des blocs:

$$\forall i \neq j \quad d(i,j) \geq r_i + r_j \quad (1)$$

où $d(i,j)$ désigne la distance entre les centres des blocs i et j sur le plan et r_i représente le rayon du bloc i . Si l et h sont la longueur et la hauteur du bloc,

alors r_i est donné par:

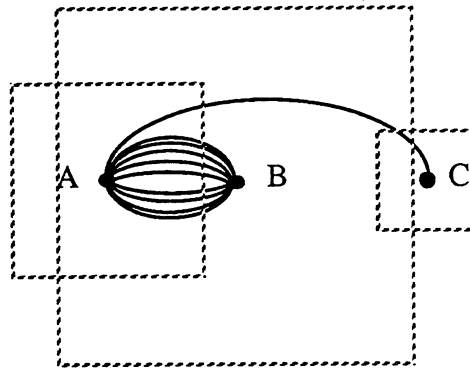
$$r_i = \frac{1}{2} \sqrt{h^2 + l^2}$$

Dans le but de faciliter la décompaction, on pourrait modifier les distances de la matrice de proximités, de la façon suivante:

$$d_{ij} = r_i + r_j + \frac{1}{k + N_{ij}} \quad \forall i \neq j$$

Mais ceci n'est pas satisfaisant car les valeurs de r_i et r_j sont généralement beaucoup plus grandes que les valeurs initiales des distances. De ce fait, les rapports des distances sont complètement perturbés et ne reflètent plus la proximité des blocs selon la connectique mais plutôt les contraintes de non chevauchement. L'exemple suivant en donne une illustration.

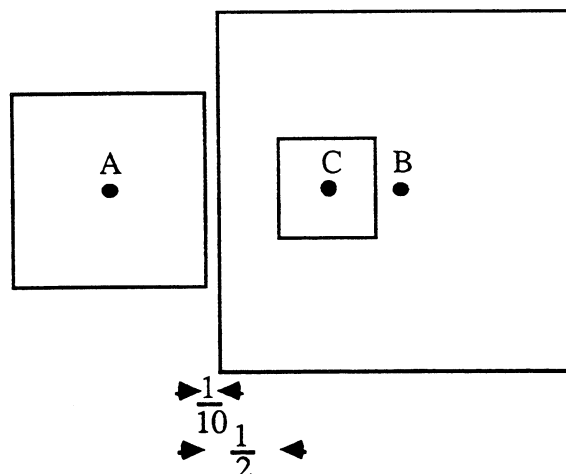
Exemple :



$$d_{ij} = r_i + r_j + \frac{1}{1 + N_{ij}} \quad \forall i \neq j$$

$$\Rightarrow d(A,B) = r_A + r_B + \frac{1}{10} \quad \text{et} \quad d(A,C) = r_A + r_C + \frac{1}{2}$$

On obtient la configuration suivante:



D'une manière générale, le seul moyen de modifier l'ordre de grandeur des distances, sans en changer les rapports, est de les multiplier toutes par le même

facteur. Le choix de ce facteur dépend de l'objectif à atteindre. Si l'on désire que toutes les distances de la matrice vérifient la condition (1) alors on peut tenir le raisonnement suivant:

En prenant comme distance entre i et j $d_{ij} = \frac{1}{1 + N_{ij}}$ on a

$$\forall i, j \text{ tq } i \neq j \text{ alors } \frac{1}{1 + \max_{i \neq j} (N_{ij})} \leq d_{ij} \leq 1$$

De ce fait en choisissant comme facteur $\max_{i \neq j} (r_i + r_j)(1 + \max_{i \neq j} (N_{ij}))$ la condition (1) sera vérifiée. Mais cependant les couples de blocs qui admettaient des distances proches de 1 vont se retrouver très éloignés et alors le problème de décompaction des blocs risque de se transformer en problème de compaction.

Une autre façon de prendre en compte la surface des blocs lors de la définition des distances est utilisée dans [KOUKA 89]:

$$d_{ij} = \frac{(N_{*i} - N_{ij})S_i + (N_{*j} - N_{ij})S_j}{(S_i + S_j)(1 + N_{ij}^2)} = \frac{R_{ij}}{A_{ij}} \quad \forall i, j$$

Avec $S_i =$ périmètre du bloc i .

Cette formule peut être interprétée comme le quotient d'une force de répulsion R_{ij} entre les blocs, par une force d'attraction A_{ij} .

La force de répulsion entre les blocs tient compte de leurs dimensions et du nombre de connexions qui ne sont pas en commun, ce qui doit permettre aux blocs de ne pas se chevaucher. En revanche, l'attraction entre deux blocs ne dépendant que du nombre d'équipotentiels partagées, la force associée ne devrait pas faire intervenir les dimensions. Étudions les valeurs de cette formule dans différents cas:

- Si $N_{ij} = 0$, la valeur de d_{ij} est maximale et on obtient $d_{ij} = \frac{N_{*i} S_i + N_{*j} S_j}{S_i + S_j}$
ce qui implique $\min(N_{*i}, N_{*j}) \leq d_{ij} \leq \max(N_{*i}, N_{*j})$

- Si $N_{ij} = \frac{N_{*i}}{2} = \frac{N_{*j}}{2}$ on a $d_{ij} = \frac{N_{ij} S_i + N_{ij} S_j}{(S_i + S_j)(1 + N_{ij}^2)} = \frac{N_{ij}}{1 + N_{ij}^2} \ll 1$

- Si $N_{ij} = N_{*i} = N_{*j}$ alors $d_{ij} = 0$

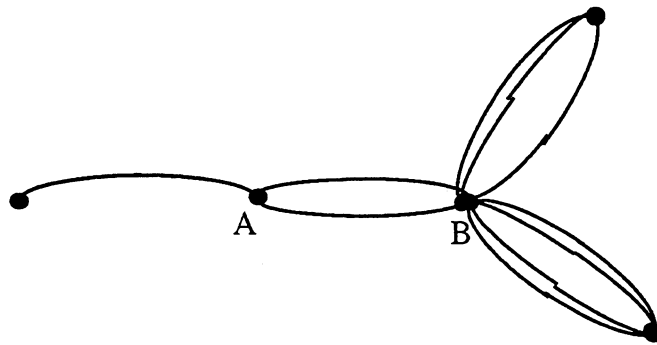
Il apparaît donc que les valeurs de d_{ij} appartiennent à l'intervalle $[0, \max_i (N_{*i})]$ et que l'objectif d'obtenir des distances du même ordre de grandeur que les dimensions des blocs n'est pas atteint.

Considérons maintenant un autre cas particulier:

$$\text{- Si } N_{ij} = \frac{N_{*i}}{2} = N_{*j} \quad \text{on a } d_{ij} = \frac{N_{ij} S_i}{(S_i + S_j)(1 + N_{ij}^2)} = \frac{N_{ij}}{1 + N_{ij}^2} \times \frac{S_i}{S_i + S_j}$$

On observe que d_{ij} est d'autant plus petit que S_j est grand par rapport à S_i , or ceci n'a aucun sens. L'exemple suivant montre que la présence des dimensions des blocs dans cette définition des distances conduit à des incohérences.

Exemple :



$$d_{AB} = \frac{(3 - 2)S_A + (9 - 2)S_B}{(S_A + S_B)(1 + 2^2)} = \frac{S_A + 7S_B}{5(S_A + S_B)}$$

$$\text{Si } S_A = S_B \text{ on a } d_{AB} = \frac{8}{10} = 0,8$$

$$\text{Si } S_A = 2 \text{ et } S_B = 9 \text{ on a } d_{AB} = \frac{65}{55} = 1,181\dots$$

$$\text{Si } S_A = 9 \text{ et } S_B = 2 \text{ on a } d_{AB} = \frac{23}{55} = 0,418\dots$$

Il n'y a aucune raison d'avoir des distances différentes entre A et B selon que S_A est plus grand que S_B ou l'inverse, d'autant plus que la variation est supérieure à 100% d'un cas sur l'autre.

Il en ressort que l'utilisation qui est faite des dimensions des blocs dans cette formule est inadaptée au problème.

Lorsque les valeurs de S_i et S_j sont égales, on a:

$$d_{ij} = \frac{(N_{*i} - N_{ij}) + (N_{*j} - N_{ij})}{2(1 + N_{ij}^2)}$$

Les dimensions n'intervenant plus, il devient plus simple d'étudier le comportement de cette distance selon la connectique.

A partir d'un certain seuil de connectivité entre deux blocs, la distance devient

inférieure à 1. Pour cela on doit avoir:

$$\begin{aligned} N_{*i} + N_{*j} - 2N_{ij} &< 2 + 2N_{ij}^2 \\ \Leftrightarrow 2N_{ij}^2 + 2N_{ij} - N_{*i} - N_{*j} + 2 &> 0 \end{aligned} \quad (2)$$

Ce polynôme du second degré en N_{ij} admet deux racines lorsque:

$$N_{*i} + N_{*j} > \frac{3}{2}$$

Si l'on suppose que le circuit ne comprend pas de blocs isolés alors cette condition est toujours réalisée et l'inégalité (2) est vérifiée lorsque:

$$N_{ij} > \frac{-1 + \sqrt{1 + 2(N_{*i} + N_{*j} - 2)}}{2} = r_2 \quad (3)$$

En résumé, la distance entre i et j devient inférieure à 1 dès que l'inégalité (3) est vérifiée. Le tableau suivant permet de voir quand ceci est réalisé:

$N_{*i} + N_{*j}$	2	3	4	5	6	7	8	9	15	30	60	120
r_2	0	0,3	0,6	0,8	1	1,1	1,3	1,4	2,09	3,2	4,9	7,1

On voit donc que si la quantité $N_{*i} + N_{*j}$ est inférieure à 6 alors il suffit d'une connexion entre i et j pour que la distance soit inférieure à 1. Lorsque $N_{*i} + N_{*j} = 60$ alors la distance entre i et j devient inférieure à 1 dès que $N_{ij} \geq 5$.

Ce tableau montre que l'inégalité (3) est facilement vérifiée pour deux blocs interconnectés. De plus quand $N_{ij} > r_2$ la distance entre les blocs i et j est rapidement proche de zéro pour N_{ij} suffisamment grand. En revanche, lorsque l'inégalité (3) n'est pas vérifiée, les distances sont comprises entre 1 et $\max_i (N_{*i})$ ce qui entraîne de grandes disproportions avec le cas précédent.

Ceci se traduit par une répartition des distances en deux classes principales, la première comprenant les distances inférieures à 1 et la seconde regroupant les grandes distances correspondant aux blocs non interconnectés. Évidemment une telle répartition n'est pas souhaitable car elle masque les cas intermédiaires et constitue d'une certaine façon une perte d'informations. De plus la projection risque fortement d'aggraver ce phénomène, c'est pourquoi il est préférable d'avoir des distances uniformément réparties.

La raison de cette mauvaise répartition revient au terme N_{ij}^2 dans le dénominateur. On voit qu'en supprimant le carré on se ramène à une formule proche de celle étudiée dans le paragraphe IV.1.1.

Nous venons de voir diverses façons de définir les proximités entre les blocs, dans le but de les rapprocher en fonction du nombre d'équipotentiels qu'ils partagent. De toutes, la plus satisfaisante est en fait la plus simple:

$$d_{ij} = \frac{1}{k + N_{ij}} \quad \forall i \neq j$$

Les distances ainsi obtenues sont toutes inférieures à $\frac{1}{k}$ et donc ne prennent pas en compte les dimensions des blocs. Nous avons vu que ceci ne peut être fait qu'approximativement en les multipliant toutes par un même facteur reflétant les dimensions moyennes d'un bloc.

Nous nous sommes essentiellement attachés jusqu'à présent à définir des distances entre blocs interconnectés, sans préoccupation pour les blocs ne partageant pas d'équipotentiels. Le seul but étant de leur affecter des distances supérieures à celles des blocs interconnectés, une seule et même valeur leur était attribuée.

Or nous l'avons déjà remarqué, le but de l'ATP revient à essayer de réduire le moins possible les grandes distances sur le plan. On peut donc dire que les distances entre les blocs non interconnectés jouent un rôle primordial, c'est pourquoi il est important de ne pas les négliger.

Il est évident qu'après la projection sur le plan, toutes ces distances ne seront pas identiques, il n'y a d'ailleurs aucun intérêt à ce qu'elles le soient. Il convient donc de définir leurs valeurs d'une façon plus adéquate. C'est ce que nous allons voir dans la suite.

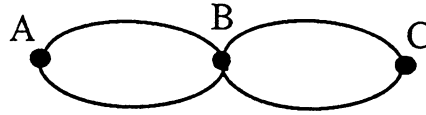
IV.2 Distance entre blocs non interconnectés

La distance que nous avons retenue permet d'exprimer des proximités entre les blocs interconnectés de façon idéale par rapport à la connectique qui les relie. Il n'en est pas de même entre les blocs ne partageant pas d'équipotentiels. Dans ce cas la distance entre deux blocs dépend de la présence ou pas d'autres blocs connectés à chacun d'eux. La formule adoptée précédemment ne permet pas de prendre en compte ce critère. Ceci va créer des conflits inutiles dans la matrice des proximités, qui ne pourront être résolus que par des déformations des distances.

En effet, nous avons vu lors de la présentation de l'ATP que cette méthode nécessite de partir d'une matrice de distances euclidiennes entre les individus afin de procéder à leur projection sur le plan principal. Quand ce n'est pas le cas, la matrice de proximités doit être réajustée de façon à la rendre euclidienne. Dans notre situation, les distances obtenues ne sont pas

euclidiennes car elles conduisent fréquemment à la violation de l'inégalité triangulaire comme le montre l'exemple suivant:

Exemple :



En prenant comme mesure de proximité $d_{ij} = \frac{1}{1 + N_{ij}} \quad \forall i \neq j$

on a:

$$d(A,B) = \frac{1}{3} = d(B,C) \quad \text{et} \quad d(A,C) = 1$$

Ce qui ne vérifie pas:

$$d(A,C) \leq d(A,B) + d(B,C)$$

Il est possible d'éliminer un certain nombre de ces violations en affectant des distances adéquates entre les blocs non interconnectés, sans pour autant perturber les proximités liées à des connexions.

Considérons pour cela les contraintes de placement sur les blocs de l'exemple précédent, nous avons:

A doit être proche de B

B doit être proche de C

A peut être aussi éloigné que l'on veut de C

Donc pour respecter l'inégalité triangulaire on prend:

$$d(A,C) = d(A,B) + d(B,C)$$

c'est-à-dire la plus grande distance autorisée par l'inégalité triangulaire.

Dans le cas où il y a k blocs connectés à la fois à A et C, il existe k inégalités triangulaires à respecter. On prend alors:

$$d(A,C) = \min_{i=1 \dots k} (d(A,B_i) + d(B_i,C))$$

Cela nous conduit à construire la matrice des distances de la façon suivante:

1. Calculer toutes les distances entre blocs connectés, à l'aide de la formule choisie. Les distances entre blocs non connectés sont fixées à l'infini.

2. Pour chaque couple (i,j) de blocs non connectés prendre comme distance

$$d_{ij} = \min_{k=1 \dots n} (d_{ij}, d_{ik} + d_{kj})$$

3. Si des distances viennent d'être modifiées retourner en 2., sinon fin.

On remarque que la définition des proximités peut être révisée. En effet, la constante rajoutée au dénominateur en cas d'absence de connexions communes

devient caduque. On peut donc définir la distance entre deux blocs connectés par:

$$d_{ij} = \frac{1}{N_{ij}} \quad \forall i \neq j \text{ connectés.}$$

Cet algorithme permet de réduire le nombre de conflits à l'inégalité triangulaire mais n'assure pas qu'il n'en restera aucun. En effet si un conflit est créé à l'étape 1. entre trois blocs tous connectés les uns aux autres, alors la deuxième étape de l'algorithme n'autorise pas que ces distances soient modifiées. Il est possible de résoudre cela en autorisant l'étape 2. à modifier toutes les distances, comme il est fait dans [CHAISEMARTIN 86], mais on perd alors la maîtrise des distances sur les blocs interconnectés et ceci peut mener à des contradictions vis à vis de la connectique.

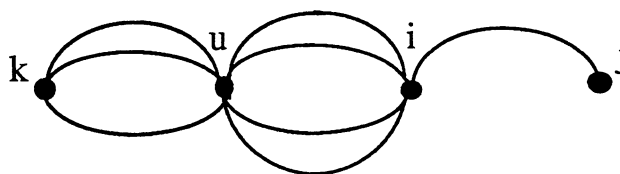
Le reste des violations de l'inégalité triangulaire sera résolu par l'algorithme de réajustement de la matrice, dont le principe a été donné lors de l'exposé de la méthode d'analyse d'un tableau de proximités. Ce réajustement a l'avantage de ne pas modifier les relations d'ordre entre les distances, contrairement à l'algorithme précédent, si on l'appliquait à toutes les distances.

Lors de la définition des distances, le but est de rapprocher les blocs interconnectés par rapport à ceux qui ne le sont pas. Ceci implique que pour un bloc donné, les blocs qui lui sont connectés doivent lui être plus proches que ceux qui ne lui sont pas connectés:

$$\forall i,j,k \text{ tq } i,j \text{ connectés et } i,k \text{ non connectés, on doit avoir } d_{ij} \leq d_{ik}$$

Or il arrive, dans certaines situations, que l'algorithme précédent conduise au non respect de cette inégalité.

Exemple :



$$d_{ku} = \frac{1}{3} \quad d_{ui} = \frac{1}{4} \quad d_{ij} = 1$$

Avec l'algorithme précédent on obtiendra $d_{ik} = \frac{1}{3} + \frac{1}{4}$
donc $d_{ik} \leq d_{ij}$

Afin d'éviter ce genre de déformations inadéquates, il est nécessaire de stocker pour chaque bloc, sa distance maximale avec un bloc qui lui est connecté. Ainsi

la modification d'une distance de ce bloc ne sera autorisée que si elle reste supérieure à la distance maximale stockée.

L'algorithme de construction de la matrice des distances est alors le suivant:

1. Calculer toutes les distances entre blocs connectés, à l'aide de la formule choisie. Les distances entre blocs non connectés sont fixées à l'infini.
2. Pour chaque bloc i , stocker dans $\text{dist-max}[i]$ sa distance maximale avec un bloc qui lui est connecté.
3. Pour chaque couple (i,j) de blocs non connectés faire


```

      { Pour chaque  $k \neq i,j$  faire
        {  $d_{ikj} \leftarrow d_{ik} + d_{kj}$ 
          Si ( $d_{ikj} \leq d_{ij}$  et  $d_{ikj} > \text{dist-max}[i]$  et  $d_{ikj} > \text{dist-max}[j]$ )
            {  $d_{ij} \leftarrow d_{ikj}$ 
               $d_{ji} \leftarrow d_{ikj}$ 
            }
          }
        }
      
```
4. Si des distances viennent d'être modifiées retourner en 3., sinon fin.

La matrice des distances ainsi constituée est soumise à une ATP.

Pour cela le produit de matrices $W.D$ est calculé et ses valeurs propres sont déterminées. Si la matrice des distances a une structure euclidienne alors les valeurs propres sont toutes positives et les vecteurs propres associés donnent les coordonnées des centres des blocs sur les axes principaux.

L'algorithme précédent ne modifiant pas toutes les distances, il peut subsister des violations de l'inégalité triangulaire. Dans ce cas il existe des valeurs propres négatives et les vecteurs propres ne correspondent pas aux coordonnées des blocs sur les axes principaux. Pour obtenir une représentation planaire des blocs on doit réajuster la matrice des distances de façon à transformer les proximités en distances euclidiennes, puis effectuer de nouveau une ATP sur la nouvelle matrice des distances. Ce réajustement est réalisé selon la méthode suggérée dans la partie II.

Lors de la présentation de l'analyse en composantes principales nous avons introduit la part d'inertie cumulée des deux premiers axes principaux qui permet de mesurer la qualité de la représentation des distances sur le plan par rapport à celles de la matrice.

Cette part d'inertie n'a de sens pour nous que si elle reflète la déformation des

distances sur le plan par rapport à la matrice initiale. Or dans le cas d'un réajustement, la part d'inertie cumulée est donnée par rapport à la nouvelle matrice des distances, elle n'est donc d'aucune utilité pour déterminer la déformation des distances initiales.

IV.3 Pondération des blocs

Nous avons vu, durant la présentation de l'ATP, que les composantes principales sont données par les vecteurs propres de la matrice W.D. La matrice D est une matrice diagonale, dont les éléments diagonaux constituent une pondération des individus:

$$D = \begin{bmatrix} p_1 & & & 0 \\ & \dots & & \\ & & \dots & \\ 0 & & & p_n \end{bmatrix} \quad \text{telle que } \sum_{i=1}^n p_i = 1 \quad \text{et } p_i \geq 0 \quad \forall i$$

Cette pondération des blocs intervient dans la recherche des composantes principales, puisque celles-ci sont déterminées de façon à maximiser le critère:

$$\sum_{i,j=1}^n p_i \cdot p_j \cdot d^2(f_i, f_j) \quad \text{où les } f_i \text{ sont les projections des points sur le plan.}$$

Il est donc intéressant d'étudier l'influence des différentes pondérations possibles sur la déformation des distances lors de la projection.

On voit qu'en fait, la pondération des blocs se traduit par une pondération des distances dans le critère à maximiser. Chaque distance entre un bloc i et un bloc j est pondérée par le produit $p_i \cdot p_j$, qui caractérise l'importance de d_{ij} dans le critère. De ce fait plus $p_i \cdot p_j$ est important et moins d_{ij} risque d'être réduite lors de la projection.

Il apparaît donc que si certaines distances sont "critiques" et doivent rester faibles, il faut assurer que les distances qui leur sont supérieures avant la projection le soient encore sur le plan.

- La pondération par défaut consiste à donner une importance égale à tous les blocs:

$$p_i = \frac{1}{n} \quad \forall i,j$$

$$\text{D'où } p_i \cdot p_j = \frac{1}{n^2} \quad \forall i,j$$

Aucune distance n'est privilégiée par rapport aux autres.

Dans le cas où les poids affectés aux distances ne sont pas tous identiques, l'objectif à atteindre est de pondérer plus fortement les grandes distances, afin qu'elles restent supérieures aux autres dans la représentation planaire.

Etudions si une pondération permet d'atteindre cet objectif:

- Pondération relative à la connectique:

Chaque bloc est pondéré selon le nombre d'équipotentiels qu'il admet.

$$p_i = \frac{\text{nombre d'équipotentiels adjacentes à } i}{\sum_{i=1}^p t_i} \quad \forall i, j$$

où p représente le nombre total d'équipotentiels et t_i représente le nombre de terminaisons de l'équipotentielle i .

Alors les distances de poids forts sont celles entre deux blocs ayant chacun un grand nombre d'équipotentiels, mais n'en partageant pas forcément. Ceci ne conduit pas à une pondération telle que nous la désirons puisque les distances de poids forts sont aussi bien petites que grandes.

- Pondération relative à la surface:

Chaque bloc est pondéré selon sa surface.

$$p_i = \frac{\text{surface du bloc } i}{\text{surface totale des blocs}} \quad \forall i, j$$

Les distances de poids forts sont celles entre blocs de surfaces importantes. Nous avons vu, lors de l'étude des distances, que les surfaces des blocs i et j ne peuvent pas être prises en compte pour définir d_{ij} . Puisque les distances reflètent uniquement la connectique, il est inadapté de les pondérer selon la surface des blocs. De plus ceci ne conduit pas à une forte pondération des grandes distances.

Essayons de définir directement une pondération adéquate des distances.

On désire que le poids soit d'autant plus important que la distance est grande, l'idéal est donc d'avoir:

$$\forall i, j, n, m \text{ tq } i \neq j \text{ et } n \neq m \quad \frac{p_i \cdot p_j}{p_n \cdot p_m} = \frac{d_{ij}}{d_{nm}}$$

Cette condition est réalisée ssi:

$$\exists K = \text{constante tq } \forall i, j \text{ tq } i \neq j \quad p_i \cdot p_j = K \cdot d_{ij}$$

Sur l'ensemble des valeurs de i et j , on obtient un système de $\frac{n^2 - n}{2}$ équations à n inconnues.

Fixons i et considérons les équations concernant d_{ik} et d_{kl} . On obtient:

$$p_i = \frac{K \cdot d_{ik}}{p_k} \quad \text{et} \quad p_i = \frac{K \cdot d_{il}}{p_l}$$

$$\text{d'où} \quad \frac{d_{ik}}{d_{il}} = \frac{p_k}{p_l}$$

En prenant j à la place de i on obtient également: $\frac{d_{jk}}{d_{jl}} = \frac{p_k}{p_l}$

On en déduit que le système n'admet de solutions que si:

$$\text{Pour tout couple } (i,j) \text{ avec } i \neq j \text{ on a } \frac{d_{ik}}{d_{il}} = \frac{d_{jk}}{d_{jl}} \quad \forall k, l \neq i, j$$

Il est évident qu'une telle condition n'a aucune chance d'être réalisée sur les distances telles qu'on les a définies.

Ceci confirme la difficulté d'obtenir une bonne pondération des blocs.

Il est important de préciser que nous venons de considérer les déformations des distances lors de la projection, or cette projection n'est effectuée que si les distances sont euclidiennes. Ce n'est généralement le cas qu'après le réajustement de la matrice, et durant ce réajustement il n'est pas possible de contrôler l'évolution des distances.

Il ressort de cette analyse que parmi les différentes pondérations étudiées, la plus adaptée est de donner à tous les blocs un poids identique, égal à $\frac{1}{n}$.

IV.4 Calcul des vecteurs propres

Lorsque la matrice des distances est définie et que les poids des blocs sont attribués, tous les paramètres du problème sont fixés et l'analyse du tableau de proximités peut être effectuée. Celle-ci consiste à calculer les vecteurs propres de la matrice $W.D$.

La constitution des matrices W et D a été détaillée auparavant et l'obtention du produit ne pose aucun problème. C'est donc le calcul des vecteurs propres que nous allons aborder dans cette partie, ce calcul constitue en effet l'essentiel de l'ATP.

Toutes les méthodes utilisées en pratique pour calculer les valeurs propres et les vecteurs propres d'une matrice carrée A , sont des méthodes itératives qui

Il est démontré dans [GOLUB&VAN LOAN 83] que la suite de matrices $A_1, A_2, \dots, A_r, \dots$ ainsi constituée, converge vers une matrice diagonale ayant les mêmes valeurs propres que A . En pratique cette convergence est assez rapide, il a d'ailleurs été démontré (Schonage (1964) et Van Kempen (1966)) que la convergence de $\text{off}(A_j)$ vers zéro, devient quadratique pour i assez grand.

Pour p et q fixés, les valeurs $\cos\theta$ et $\sin\theta$ de la matrice $\Omega_{pq\theta}$, qui annulent l'élément (p,q) de A , sont obtenues de la façon suivante:

$$\cos\theta = \frac{1}{\sqrt{1 + t^2}}$$

$$\sin\theta = t \cdot \cos\theta$$

avec
$$t = \frac{\text{signe}(\tau)}{|\tau| + \sqrt{1 + \tau^2}} \quad \text{et} \quad \tau = \frac{a_{qq} - a_{pp}}{2a_{pq}}$$

Dès que l'on atteint un indice L pour lequel la valeur $\text{off}(A_L)$ devient négligeable, on a:

$$A_L = (\Omega_{pq\theta}^L)^t \cdot (\Omega_{pq\theta}^{L-1})^t \cdot \dots \cdot (\Omega_{pq\theta}^1)^t \cdot A \cdot \Omega_{pq\theta}^1 \cdot \dots \cdot \Omega_{pq\theta}^L = D_{vp}$$

où D_{vp} est la matrice diagonale des valeurs propres. La matrice des vecteurs propres associés est alors la matrice produit: $\Omega_{pq\theta}^1 \cdot \dots \cdot \Omega_{pq\theta}^L$.

L'algorithme de Jacobi par balayage sur les lignes, pour une matrice A carrée, symétrique, de dimension n , est le suivant:

1. $M \leftarrow \text{Id}$, calculer $\text{off}(A)$.
2. Tant que ($\text{off}(A) > \varepsilon$) faire
 - { Pour $p=1$ à $n-1$ faire
 - Pour $q=p+1$ à n faire
 - Si $|a_{pq}| > 0$ alors
 - { Calculer $\Omega_{pq\theta}$
 - $A \leftarrow (\Omega_{pq\theta})^t \cdot A \cdot \Omega_{pq\theta}$
 - $M \leftarrow M \cdot \Omega_{pq\theta}$
 - }
 - Calculer $\text{off}(A)$
 - }
3. Les valeurs propres sont les éléments diagonaux de A , et les vecteurs propres associés sont les colonnes de la matrice M .

Id désigne la matrice unité de dimension n et ε correspond à la précision à partir de laquelle on considère que la valeur de $\text{off}(A)$ est négligeable.

Lorsque l'on calcule A_{k+1} à partir de A_k avec la formule (4), seules les lignes et les colonnes p et q sont modifiées dans A_{k+1} par rapport à A_k .

On en déduit un algorithme simplifié pour effectuer les transformations $A \leftarrow (\Omega_{pq\theta})^t \cdot A \cdot \Omega_{pq\theta}$ et $M \leftarrow M \cdot \Omega_{pq\theta}$.

1. Modification des lignes et des colonnes p et q de A:

$$a_{pq} \leftarrow 0$$

$$a_{qp} \leftarrow 0$$

$$a_{pp} \leftarrow a_{pp} - t \cdot a_{pq}$$

$$a_{qq} \leftarrow a_{qq} + t \cdot a_{pq}$$

Pour j=1 à n faire

Si ($j \neq p$ et $j \neq q$) alors

$$\{ r \leftarrow a_{jp}$$

$$a_{jp} \leftarrow r \cdot \cos\theta - a_{jq} \cdot \sin\theta$$

$$a_{jq} \leftarrow r \cdot \sin\theta + a_{jq} \cdot \cos\theta$$

$$a_{pj} \leftarrow a_{pj}$$

$$a_{qj} \leftarrow a_{qj}$$

}

2. Modification des colonnes p et q de M

Pour j=1 à n faire

$$\{ r \leftarrow m_{jp}$$

$$m_{jp} \leftarrow r \cdot \cos\theta - m_{jq} \cdot \sin\theta$$

$$m_{jq} \leftarrow r \cdot \sin\theta + m_{jq} \cdot \cos\theta$$

}

V. OBTENTION D'UN PLACEMENT RÉALISABLE

La méthode de placement par analyse du tableau de proximités fournit une disposition des centres des blocs sur le plan de façon à minimiser globalement la connectique. Cette disposition des blocs les uns par rapport aux autres n'admet pas d'orientation par rapport au circuit, néanmoins nous supposons que le repère orthogonal dans lequel sont données les coordonnées des points correspond à l'orientation du circuit par rapport au nuage planaire.

Le placement des blocs qui en découle n'est pas forcément réalisable, de plus l'orientation des blocs n'est pas déterminée et la disposition n'optimise pas la surface.

Afin d'obtenir un placement réalisable nous devons faire en sorte que les contraintes de faisabilité du placement, énoncées au chapitre précédent, soient vérifiées. Elles sont au nombre de trois, dont deux ne peuvent pas être prises en compte lors du placement par ATP. Heureusement, ces dernières ne sont pas systématiquement imposées lors des problèmes de placement : il s'agit d'imposer une position à certains blocs ainsi qu'une surface hôte au circuit.

Evidemment les plots d'entrées-sorties du circuit doivent forcément être placés sur la périphérie de celui-ci, mais on peut ne pas les considérer lors du

placement par ATP. En revanche, si des blocs internes au circuit ont une position imposée à l'intérieur de la surface hôte, alors le problème du placement ne peut pas être traité par ATP: il n'est pas possible, à l'aide de la matrice de proximité, d'imposer une position à un bloc. Le même problème se pose avec la définition d'une surface hôte.

L'obtention d'un placement réalisable se réduit donc dans notre cas à assurer le respect de la contrainte de non recouvrement des blocs.

Malgré les efforts réalisés lors de la construction de la matrice de proximités pour obtenir des distances prenant en compte la taille des blocs, il est habituel que des blocs se chevauchent. La raison essentielle en est que l'ATP, dont le principe consiste à projeter le nuage de points sur un plan, ne peut que réduire les distances et ceci conduit inévitablement à des chevauchements de certains blocs.

A partir de là, une décompaction ultérieure visant à éliminer les chevauchements est nécessaire.

Cette décompaction doit permettre de préserver les positions relatives des blocs fournies par l'ATP. Pour cela plusieurs méthodes ont été étudiées, nous allons les exposer dans ce qui suit.

V.1 Décompaction par homothétie

C'est la plus instinctive des méthodes de décompaction, d'autant plus qu'elle permet de conserver exactement les positions relatives des blocs les uns par rapport aux autres. Le principe est d'effectuer une homothétie sur les centres des blocs. Le rapport de l'homothétie est déterminé de façon à ce que les centres s'espacent suffisamment pour que les blocs ne se chevauchent plus.

Une homothétie est définie par son centre et son rapport de la façon suivante: Soient le point $O(x_0, y_0)$ centre de l'homothétie et k son rapport, alors l'homothétie est une application du plan sur lui même:

$$h: P \rightarrow P$$

$$M(x, y) \rightarrow M'(x', y') \quad \text{tel que} \quad \vec{OM'} = k \cdot \vec{OM}$$

Propriétés :

Soient A et B deux points quelconques du plan et A', B' leurs images par l'homothétie, alors quel que soit le centre de l'homothétie on a:

$$d(A', B') = k \cdot d(A, B)$$

où $d(A, B)$ désigne la distance sur le plan entre les points A et B .

Si on appelle $(x_A, y_A), (x_B, y_B), (x'_A, y'_A)$ et (x'_B, y'_B) les coordonnées respectives des points A, B, A', B' dans un repère quelconque, alors on a :

$$\frac{y'_A - y'_B}{x'_A - x'_B} = \frac{y_A - y_B}{x_A - x_B}$$

Ce qui signifie que les droites (A,B) et (A',B') ont la même inclinaison quels que soient le centre et le rapport de l'homothétie.

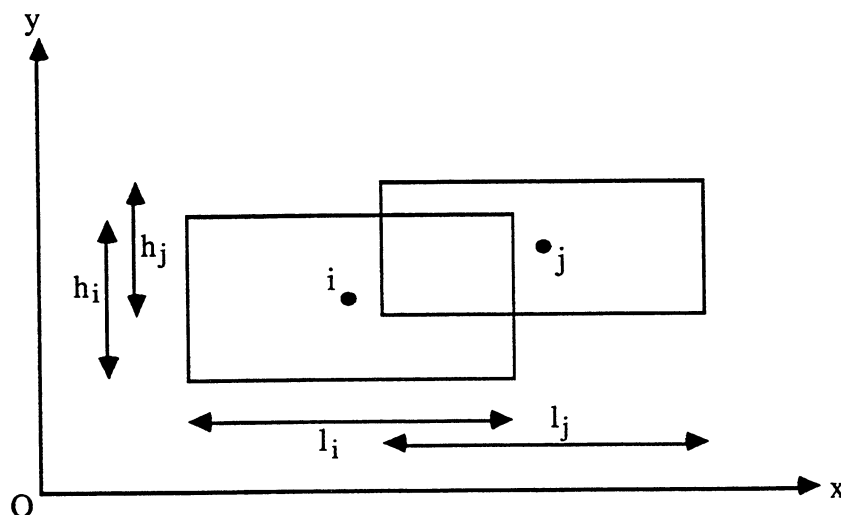
De ces propriétés on retire que l'image du nuage planaire par l'homothétie ne dépend pas du centre de l'homothétie, seule sa position sur le plan variera en fonction de la position du centre. Autrement dit, pour deux homothéties de même rapport, les images du nuage sont identiques à une translation près.

Si l'on désire que l'image du nuage reste dans la même région du plan que le nuage initial, alors on peut prendre comme centre de l'homothétie le centre de gravité du nuage.

Pour ce qui est du rapport de l'homothétie, nous allons voir dans la suite comment le déterminer de façon à espacer suffisamment les centres des blocs.

Considérons deux blocs i et j qui se chevauchent. Soient (x_i, y_i) et (x_j, y_j) les coordonnées de leur centre dans le repère orthonormé lié au plan. Puisque les blocs i et j se chevauchent, alors nécessairement leurs côtés verticaux se chevauchent sur l'axe Oy et leurs côtés horizontaux se chevauchent sur l'axe Ox, comme l'illustre l'exemple suivant :

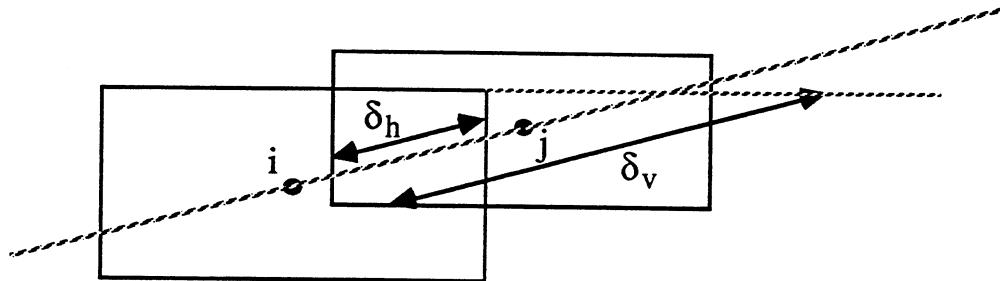
Exemple :



Pour décompacter les deux blocs il suffit de supprimer un des deux chevauchements. D'après la propriété précédente, lors de l'homothétie les deux blocs se déplaceront selon la droite qui relie leurs centres. Appelons δ_h et δ_v les déplacements selon cette droite pour supprimer respectivement le

chevauchement horizontal et le chevauchement vertical. Le déplacement minimum selon cette droite pour éliminer le chevauchement des blocs est $\min(\delta_h, \delta_v)$ comme on le voit sur l'exemple.

Exemple :



On en déduit, d'après la propriété précédente, que le rapport minimum de l'homothétie pour décompacter les deux blocs est:

$$k_{\min} = \frac{d(i,j) + \min(\delta_h, \delta_v)}{d(i,j)}$$

où $d(i,j)$ est la distance entre les centres des blocs i et j .

Dans le cas de l'exemple on a $x_i < x_j$. Si on appelle x_{d_i} et x_{g_j} les abscisses du côté droit de i et du côté gauche de j on a:

$$\delta_h = \sqrt{(x_{g_j} - x_{d_i})^2 + a^2 \cdot (x_{g_j} - x_{d_i})^2} \quad \text{avec } a = \frac{y_j - y_i}{x_j - x_i}$$

$$\Leftrightarrow \delta_h = |x_{g_j} - x_{d_i}| \cdot \sqrt{1 + a^2}$$

De la même façon, comme $y_i < y_j$, on obtient:

$$\delta_v = |x_{b_j} - x_{h_i}| \cdot \sqrt{1 + a^2}$$

Le rapport minimum de l'homothétie pour décompacter tous les blocs est déterminé par l'algorithme suivant:

$k = 1$

Pour chaque couple (i,j) de blocs faire

{ Si i et j se chevauchent alors

{ calculer $\delta = \min(\delta_h, \delta_v)$

calculer la distance d entre le centre de i et le centre de j

$k' \leftarrow (d + \delta)/d$

Si $k' > k$ alors $k \leftarrow k'$

}

}

k contient la valeur du rapport minimum de l'homothétie.

Si l'on choisit comme centre de l'homothétie l'origine du repère orthonormé, alors les nouvelles coordonnées des blocs sont obtenues en exécutant la boucle suivante:

```

Pour chaque bloc  $i$  faire
  {  $x_i \leftarrow k.x_i$ 
     $y_i \leftarrow k.y_i$ 
  }

```

Ces nouvelles coordonnées assurent que la condition de non recouvrement des blocs est respectée tout en préservant exactement les positions relatives des blocs, données par l'ATP.

Néanmoins, cette méthode de décompaction n'est pas satisfaisante car si deux blocs ont initialement des centres très proches, alors le rapport nécessaire pour les déchevaucher devra être très grand. Par voie de conséquence, les blocs qui ne se chevauchent pas vont se retrouver très éloignés les uns des autres, et le circuit va prendre une taille trop grande, voire gigantesque.

D'une façon générale, les blocs ne se chevauchent jamais tous dans les mêmes proportions et donc l'homothétie pour déchevaucher les deux blocs les plus critiques laissera beaucoup trop d'espace entre les autres blocs.

Dans l'outil de placement, la décompaction par homothétie est utilisée pour visualiser le résultat de l'ATP dans une fenêtre à part, dans le but de conseiller le concepteur sur les proximités à respecter entre les blocs.

V.2 Décompaction par arbre binaire de placement

L'idée de cette méthode est de découper le circuit en zones par des bipartitionnements successifs, jusqu'à ce que chaque bloc soit isolé dans une zone.

Le procédé est le suivant:

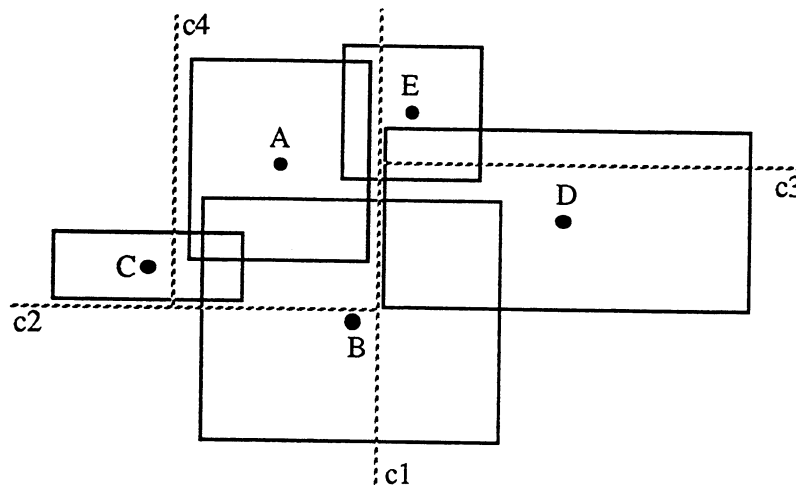
1. L'ensemble des blocs est trié selon l'axe horizontal, et une ligne de coupe verticale est définie entre chaque paire de blocs consécutifs.
2. Une des lignes de coupe est choisie pour bipartitionner le circuit en deux sous-circuits appelés cellules.
3. Les opérations 1. et 2. sont répétées sur chaque cellule avec des lignes de coupe successivement perpendiculaires, jusqu'à ce que chaque cellule ne comporte qu'un seul bloc. On obtient ainsi un arbre binaire de placement qui permettra de reconstituer le circuit sans chevauchement de blocs.

Le choix de la ligne de coupe à l'étape 2. est fait de manière à équilibrer la surface des blocs de chaque côté de cette ligne.

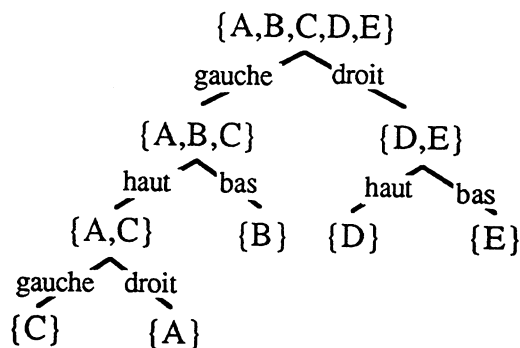
L'exemple suivant détaille l'exécution de cette méthode.

Exemple :

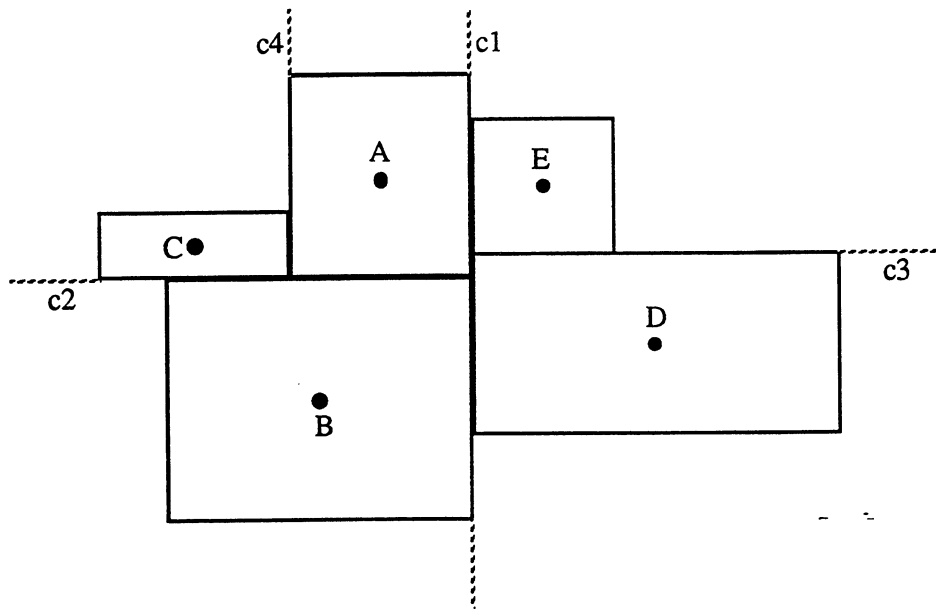
4 lignes de coupe sont nécessaires pour isoler chaque bloc dans une zone :



on obtient l'arbre binaire de placement :



d'où le placement final :



Une fois l'arbre binaire de placement constitué, on peut connaître facilement la taille de chacune des cellules en utilisant des procédures récursives. Ces procédures évaluent comme étant la largeur (ou hauteur) d'une cellule, soit, la somme des deux largeurs (ou hauteurs) des cellules de niveau inférieur, soit, le maximum de ces largeurs (ou hauteurs), ceci en fonction du type de coupe (horizontale ou verticale).

La connaissance des dimensions des cellules à n'importe quel niveau de l'arbre permet de déterminer leurs coordonnées exactes et donc d'obtenir les coordonnées des blocs.

En effectuant la décompaction selon cette méthode on fait en sorte que:

- Lors d'une bipartition horizontale, les blocs situés au Nord de la ligne de coupe soient, dans le placement final, au-dessus des blocs situés au Sud de celle-ci.
- Lors d'une bipartition verticale, les blocs situés à l'Ouest de la ligne de coupe soient, dans le placement final, à gauche des blocs situés à l'Est de celle-ci.

Or la position relative d'un bloc par rapport à un autre s'exprime dans deux directions: position horizontale du premier par rapport au second (ie à gauche ou à droite) et position verticale du premier par rapport au second (ie plus haut ou plus bas)

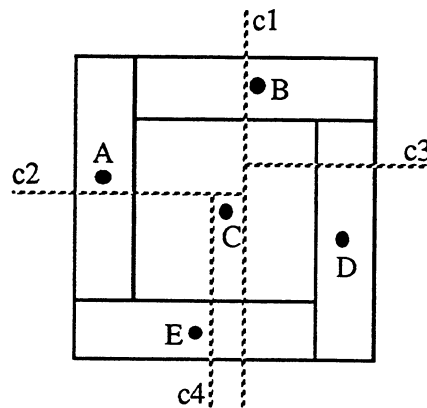
On voit donc que pour chaque couple de blocs la position relative de l'un par rapport à l'autre ne sera conservée que dans une seule direction, cette direction étant déterminée par la ligne de coupe qui a placé les deux blocs dans deux

cellules différentes.

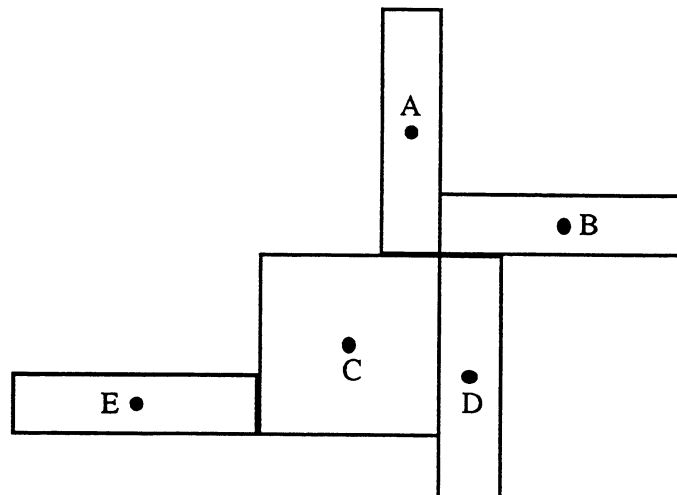
Le second défaut de cette approche est qu'elle part d'une disposition quelconque des blocs sur le plan et qu'elle génère obligatoirement un placement issu de coupes successives (*slicing placement* en anglais), ce qui est évidemment assez restrictif. L'exemple suivant montre ce que l'on obtient en partant d'une disposition classique non issue de coupes successives.

Exemple :

Les surfaces des quatre blocs périphériques étant identiques, plusieurs choix sont possibles pour c_1 et c_2 . Si l'on choisit c_1 entre $\{A,C,E\}$ et $\{B,D\}$, et c_2 entre $\{A\}$ et $\{C,E\}$ alors on obtient:



ce qui donne le circuit:



V.3 Décompaction excentrique partant du barycentre

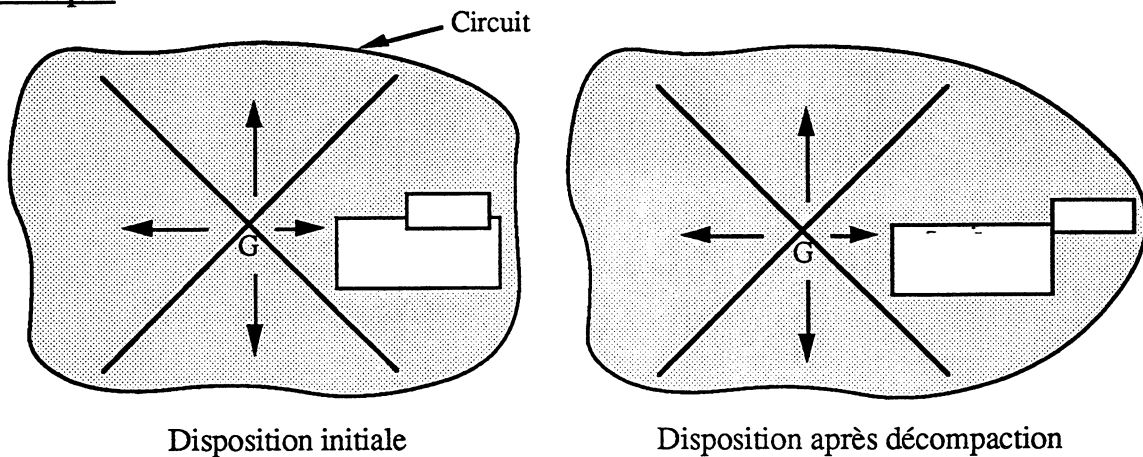
Le barycentre des blocs est déterminé et devient la nouvelle origine du repère orthonormé. Les blocs sont traités dans l'ordre croissant de leur distance au barycentre. A chaque étape, le bloc traité est déplacé excentriquement de façon à supprimer les chevauchements avec les blocs traités précédemment. Pour cela une direction de décompaction doit être choisie. Une possibilité consiste à découper le plan en plusieurs zones, chacune étant associée à une direction de

décompaction. Chaque bloc sera déplacé dans la direction associée à la zone dans laquelle il se trouve.

Si l'on découpe le plan en quatre zones, chacune délimitée par les droites d'équation $(y = x)$ et $(y = -x)$ alors les déplacements sont effectués dans quatre directions: à gauche, à droite, en haut et en bas.

Ceci peut conduire à des décompactions inadaptées comme le montre l'exemple.

Exemple :



La décompaction des blocs est faite horizontalement alors que la disposition des blocs l'un par rapport à l'autre suggère une décompaction verticale.

On voit également que les blocs seront exagérément disposés selon les axes de décompaction, dans le cas précédent le circuit aura tendance à prendre la forme d'une croix.

Cet inconvénient peut être résolu en définissant autant de directions de décompactions, qu'il y a de blocs: chaque bloc est déplacé excentriquement selon la droite reliant son centre au barycentre du circuit. Mais ceci ne permet toujours pas une meilleure prise en compte des positions relatives des blocs pour le décompactage et ne produit aucune amélioration sur l'exemple précédent.

V.4 Décompaction alternée horizontale et verticale

C'est la méthode qui a été développée dans le cadre de l'outil de placement. Elle permet de décompacter les blocs en préservant autant que possible les positions relatives sans que cela mène à une disposition trop éparpillée.

Le principe est d'alterner décompaction verticale et décompaction horizontale des blocs. La décompaction verticale se fait vers le haut et la décompaction horizontale vers la droite.

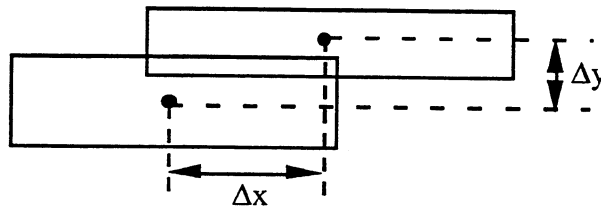
Tout d'abord, un ordre vertical et un ordre horizontal sont établis pour

l'ensemble des blocs. Ces ordres correspondent respectivement au classement croissant des ordonnées des centres des blocs, et au classement croissant des abscisses des centres des blocs, ces coordonnées étant celles fournies par l'ATP. Ils définissent une fois pour toute l'ordre de traitement des blocs pour chacune des décompactations.

Ensuite, toujours à partir des coordonnées fournies par l'ATP, une direction de décompactation est définie pour chaque paire de blocs. Soient (x_1, y_1) et (x_2, y_2) les coordonnées des centres de deux blocs, et posons $\Delta x = |x_1 - x_2|$ et $\Delta y = |y_1 - y_2|$. Si $\Delta x > \Delta y$, la direction de décompactation des deux blocs est l'horizontale, sinon, la verticale.

Néanmoins, lorsque deux blocs se chevauchent initialement et que $\Delta x > \Delta y$, les deux blocs sont effectivement décompactés horizontalement seulement si le déplacement nécessaire à cette décompactation n'est pas deux fois supérieur au déplacement nécessaire verticalement, sinon la direction de décompactation est la verticale. Il en est de même lorsque $\Delta x \leq \Delta y$.

Exemple :



$\Delta x > \Delta y$, néanmoins les deux blocs seront décompactés verticalement

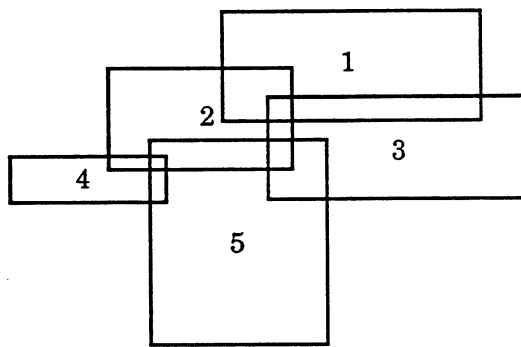
Le point de départ de l'algorithme de décompactation est la configuration initiale des blocs. Chaque bloc peut être déplacé vers le haut et vers la droite selon les directions de décompactation qu'il partage avec les autres blocs.

L'algorithme est le suivant :

1. Déterminer l'ordre de traitement horizontal et l'ordre de traitement vertical.
2. Déterminer les directions de décompaction pour chaque paire de blocs.
3. Effectuer la décompaction verticale en considérant un à un les blocs selon l'ordre de traitement défini en 1. :
Pour chaque bloc considéré, seuls les blocs situés après dans l'ordre sont déplacés vers le haut. Il faut pour cela qu'ils chevauchent le bloc courant et que leur direction de décompaction avec le bloc courant soit la verticale. Le déplacement est alors réalisé de façon à supprimer le chevauchement avec le bloc courant.
4. Effectuer la décompaction horizontale en considérant un à un les blocs selon l'ordre de traitement défini en 1. :
Pour chaque bloc considéré, seuls les blocs situés après dans l'ordre sont déplacés vers la droite. Il faut pour cela qu'ils chevauchent le bloc courant et que leur direction de décompaction avec le bloc courant soit l'horizontale. Le déplacement est alors réalisé de façon à supprimer le chevauchement avec le bloc courant.
5. Si des blocs ont été déplacés en 3. ou en 4. alors retourner en 3., sinon fin.

Le déroulement de l'algorithme est illustré sur un exemple.

Exemple :

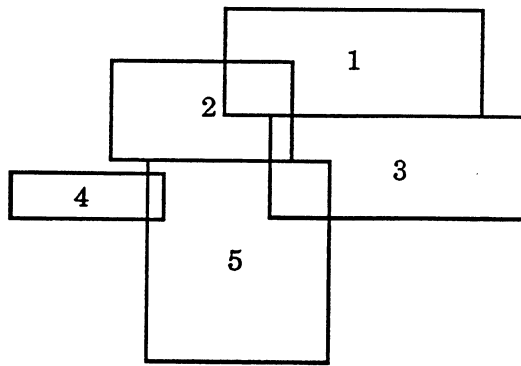


Placement initial

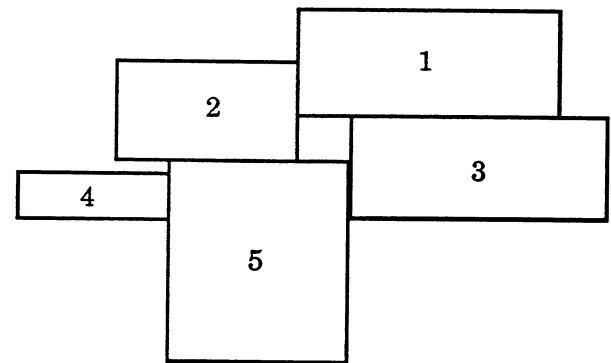
	1	2	3	4	5
1	°	h	v	h	v
2	h	°	h	v	v
3	v	h	°	h	h
4	h	v	h	°	h
5	v	v	h	h	°

Matrice des directions de décompaction

Selon les positions relatives, les blocs 4 et 2 devraient être décompactés horizontalement mais la nature du chevauchement suggère une décompaction verticale.



Décompaction verticale



Décompaction horizontale

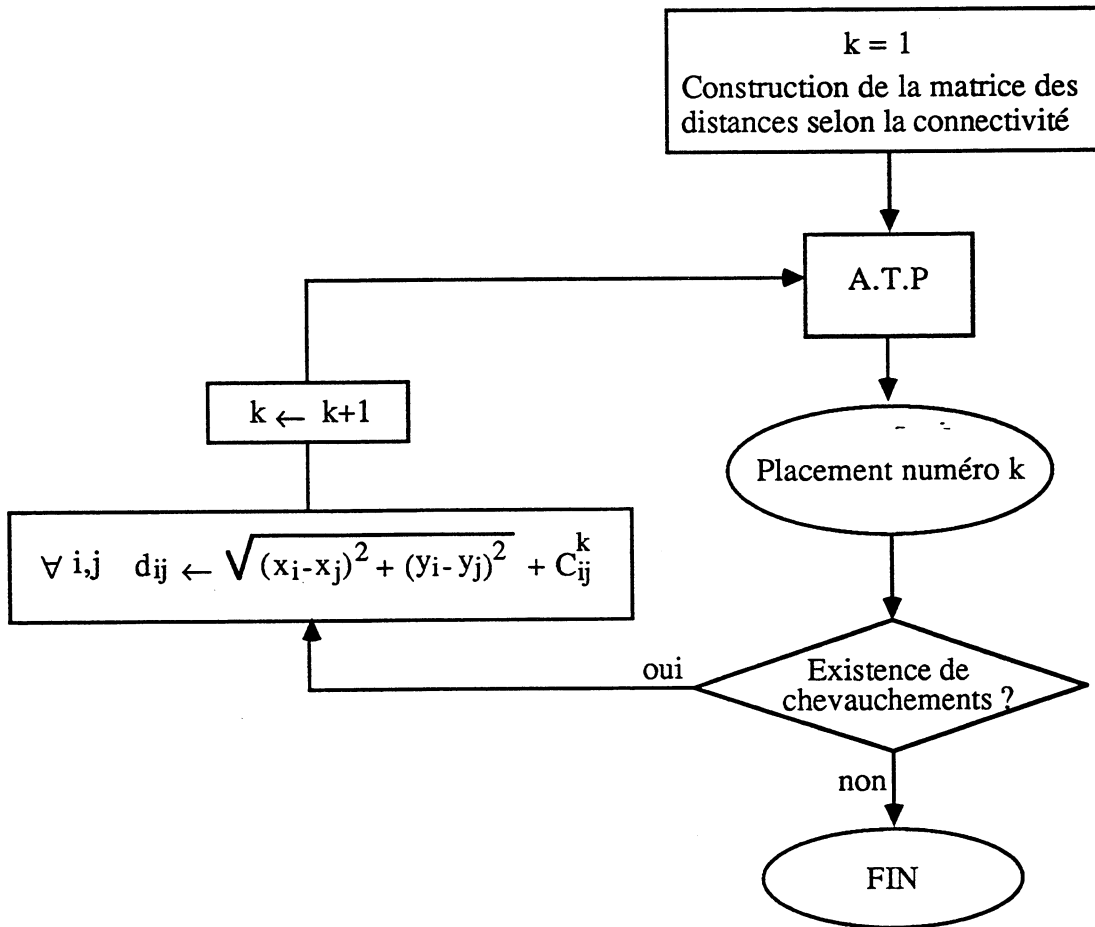
Chaque déplacement se répercute par des déplacements ultérieurs des blocs situés plus haut ou plus à droite. Cet effet de cascade a pour conséquence de maintenir assez bien les positions relatives des blocs: un bloc peut difficilement changer de place radicalement sans que les blocs voisins ne bougent à leur tour.

V.5 Décompaction par itération de l'ATP

Cette approche est basée sur le fait que si l'on fournit à l'ATP une matrice de distances issue d'une disposition des individus sur un plan, alors l'ATP restituera la disposition initiale des individus. Ceci est très bien illustré dans [CHAISEMARTIN 86] où la disposition des villes de France est retrouvée à partir d'un tableau de distances inter-villes.

L'idée consiste donc à effectuer une première ATP comme il est exposé précédemment, puis à réitérer le procédé en construisant la matrice des distances à partir des distances sur le plan. Si deux blocs se chevauchent, la distance qui les sépare sur le plan est augmentée d'un facteur représentatif du chevauchement. Les itérations se poursuivent tant qu'il existe des chevauchements.

On peut schématiser le fonctionnement de l'algorithme par l'organigramme suivant :



Organigramme de l'analyse factorielle itérée

C_{ij}^k = déplacement minimal pour décompacter i et j sur le placement n° k .
Si i et j ne se chevauchent pas alors $C_{ij}^k = 0$.

Une précaution importante est à prendre concernant l'itération de l'ATP. En effet, nous avons vu au début de la partie V que la disposition des centres des blocs sur le plan principal n'admet pas d'orientation particulière. Par convention nous avons décidé que le repère orthogonal dans lequel sont données les coordonnées correspond à l'orientation du circuit par rapport à la projection du nuage. Ceci étant fixé, il est préférable qu'après chaque itération l'orientation du nuage reste la même par rapport au repère.

Or l'orientation du nuage est définie par les deux vecteurs propres associés aux deux plus grandes valeurs propres de la matrice W.D . Si l'ordre des blocs change dans la matrice des distances, cela revient à une permutation des lignes et des colonnes de la matrice W.D, ce qui entraîne un changement des vecteurs

propres, les valeurs propres restant, bien entendu, identiques. Il en résulte un changement d'orientation du nuage par rapport au repère.

Il convient donc de garder un ordre des blocs constant dans la matrice des distances.

C'est également pour cette raison qu'en utilisant le placement par ATP on peut obtenir plusieurs résultats différents pour un circuit donné, ce qui paraît paradoxal car la connectique est évidemment identique. Ceci s'explique par le fait que si l'orientation du nuage diffère, les blocs, eux, conservent toujours la même orientation par rapport au circuit. Dans ce cas, les différents placements ne se déduiront pas les uns des autres par simple rotation. Cette différence est amplifiée par la phase de décompaction.

Après expérimentation, on observe que l'algorithme précédent converge vers un placement sans chevauchement. De plus, dès la seconde itération, la matrice des distances est "quasi-euclidienne" et on observe alors que la part d'inertie des deux premiers axes principaux converge vers 1. Ceci signifie qu'au fur et à mesure des itérations, les distances de la matrice courante sont de mieux en mieux représentées sur le plan.

La question est alors de déterminer si l'objectif principal de ce placement est atteint, à savoir, disposer les blocs sur le plan de façon à respecter le plus possible les positions relatives établies par le premier placement.

L'expérimentation montre que pour arriver à un placement sans chevauchement, l'algorithme éloigne beaucoup trop certains blocs, notamment les blocs qui ne se chevauchaient pas dans le placement initial. Ceci s'explique par le fait que la matrice de proximités constituée entre chaque passe, bien qu'étant "quasi-euclidienne", peut néanmoins contenir des violations de l'inégalité triangulaire. En effet, le fait de rajouter à chaque distance d_{ij} sur le plan, un terme C_{ij}^k spécifique à d_{ij} , conduit à des modifications des rapports des distances, pouvant engendrer de telles violations. Il s'en suit un réajustement de la matrice, consistant à ajouter à chaque distance la quantité $2.|\lambda_r|$ où λ_r désigne la plus petite valeur propre de la matrice $W.D$ (voir paragraphe II.). De ce fait, même les blocs qui ne se chevauchaient pas se trouvent écartés un peu plus.

La répétition de ce phénomène entre chaque passe conduit à des placements très étendus qui nécessitent une phase de compaction. Cette compaction est d'autant plus délicate que les blocs sont très dispersés.

On voit donc que l'itération de l'ATP pour la décompaction du circuit ne procure pas d'avantages par rapport à la méthode précédente. Ceci est d'autant plus vrai que l'accroissement en temps de calcul est loin d'être négligeable.

V.6 Contrainte de faisabilité du routage

Comme il a été dit dans le chapitre précédent, un placement est réalisable si le routage du circuit peut être effectué. Pour cela la largeur des canaux de routage doit être suffisante, or ces largeurs ne peuvent être estimées que lorsque l'on connaît le routage global du circuit.

La contrainte de faisabilité du routage n'est donc pas prise en compte lors de la construction du placement, et des largeurs arbitraires sont données aux canaux. Pour cela les dimensions de chaque bloc sont augmentées proportionnellement à la connectique du bloc, avant la décompaction. Les blocs retrouvent ensuite leur dimension initiale, ce qui génère des espacements pour les canaux.

VI. AMÉLIORATION D'UN PLACEMENT

Nous avons vu lors de la présentation du placement par ATP que chaque bloc est assimilé à un point représentant son centre. Il est clair que cette modélisation ne permet pas de prendre en considération l'orientation des blocs dans la minimisation de la longueur totale de connexion. Or cette minimisation étant l'objectif du placement par ATP, l'amélioration d'une solution consistera essentiellement à orienter les blocs de la meilleure façon possible par rapport à la connectique, et éventuellement, à compacter le circuit si des écarts excessifs sont laissés entre les blocs.

VI.1 Orientation des blocs

L'orientation des blocs peut être guidée par différents critères. Les deux principaux sont la minimisation de la longueur totale de connexion et la minimisation de la surface globale du circuit. Le deuxième critère est très difficile à prendre en compte, nous avons vu dans le chapitre précédent que peu de méthodes de placement conduisent à sa minimisation.

VI.1.1 Orientation après placement

Si le placement est issu de coupes successives alors une approche pour orienter les blocs de façon à minimiser la surface du circuit est développée dans [ZIMMERMAN 88]. Elle consiste à calculer la fonction de forme (*shape function* en anglais) de chacun des noeuds de l'arbre binaire de placement en combinant les fonctions de forme des noeuds fils. Pour un noeud donné, cette fonction représente tous les couples (hauteur, largeur) de dimensions possibles

du noeud. Les fonctions de forme des feuilles étant parfaitement définies par les dimensions de chacun des blocs correspondants, lorsque l'on arrive à la racine, on en déduit la forme adéquate du circuit et l'on peut redescendre dans l'arbre pour déterminer la forme nécessaire de chaque noeud.

Si le placement n'est pas issu de coupes successives, l'orientation des blocs visant à minimiser la surface du circuit ne peut être faite que de façon exhaustive.

Dans notre cas, l'orientation des blocs sera conduite de façon à minimiser la longueur totale de connexion. On peut alors utiliser une modélisation par des forces:

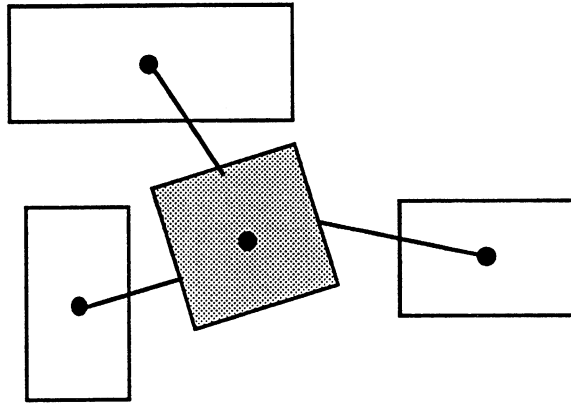
Chaque équipotentielle est représentée par une clique d'arêtes entre les points qu'elle touche sur les blocs. Chacune des arêtes est assimilée à un ressort dont la raideur est définie par le nombre de bit de la connexion associée. Les centres des blocs sont fixes et la liberté de mouvement de chacun des blocs se limite à pivoter autour de son centre.

On a donc un système physique entièrement défini, qui doit converger vers un équilibre stable correspondant à l'orientation optimum de chaque bloc.

Néanmoins la résolution d'un tel système n'est pas aisée car les blocs ne peuvent pas être traités séparément; en effet, l'orientation d'un bloc dépend de l'orientation des blocs qui lui sont connectés.

Afin de simplifier le problème et rendre l'orientation de chaque bloc indépendante des autres, on peut modifier la modélisation physique en fonction du bloc traité: seules les terminaisons des connexions sur le bloc à orienter sont considérées à leur emplacement réel, les autres terminaisons sont placées au centre des blocs. Ainsi les ressorts associés aux connexions du bloc forment un système isolé dont l'équilibre peut être facilement déterminé en utilisant les lois élémentaires de la physique (en particulier: \sum moments d'inertie = 0 à l'équilibre). L'exemple qui suit illustre cette modélisation.

Exemple :



Le bloc à orienter pivote autour de son centre, jusqu'à atteindre sa position d'équilibre.

On constate que la contrainte concernant l'orientation des blocs n'est pas prise en compte, ce qui conduit généralement à une orientation du bloc dont les côtés ne sont pas parallèles aux bords du circuit.

L'alternative serait alors de calculer la résultante des moments d'inertie, pour chacune des quatre orientations possibles, et de choisir l'orientation qui minimise cette résultante. Mais puisque l'on est ramené à une stratégie exhaustive, il est finalement plus simple de calculer la longueur de connexions pour chacune des quatre orientations et de retenir celle qui correspond à la longueur minimale.

Si l'on désire prendre en compte le retournement des blocs, il suffit de considérer également les quatre orientations du bloc retourné.

L'algorithme d'orientation des blocs qui a été implanté dans l'outil de placement fonctionne selon cette approche.

Lorsque le bloc à orienter n'a pas une forme carrée, il peut arriver que l'orientation retenue génère des chevauchements avec les blocs voisins.

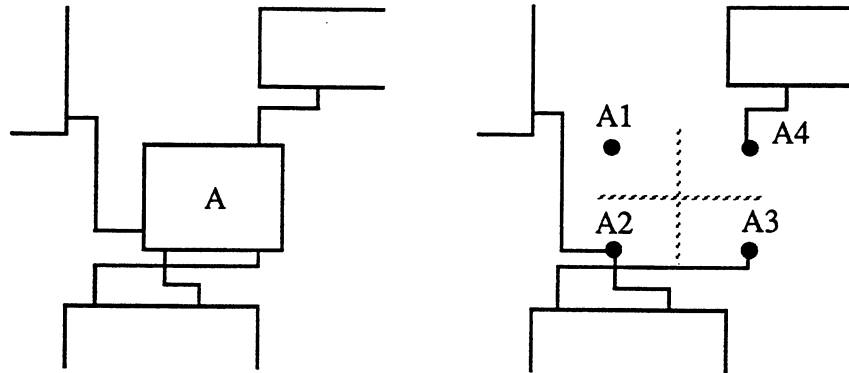
Afin d'éviter cela dans l'outil de placement, chaque bloc est étendu à son carré englobant lors de la décompaction. Ainsi lorsque les blocs retrouvent leur forme initiale, l'espacement entre eux est suffisant pour que aucun chevauchement ne se produise durant l'orientation. Une compaction du circuit peut ensuite être réalisée, de façon à éliminer les écarts excessifs entre les blocs.

VI.1.2 Prise en compte de l'orientation lors de l'ATP

L'idée d'une prise en compte de l'orientation des blocs lors de l'ATP figure dans [CHAISEMARTIN 86]. Pour cela chaque bloc n'est plus considéré comme un seul point représentant son centre, mais comme quatre points. Ces points représentent les quatre angles du bloc et constituent des individus différents

pour l'ATP. A chaque point sont associées les équipotentielles aboutissant sur le secteur correspondant du bloc, ainsi chacun des coins du bloc sera attiré indépendamment vers d'autres blocs du circuit.

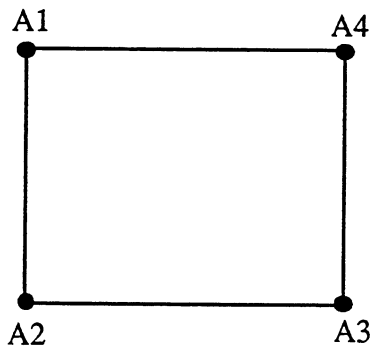
Exemple :



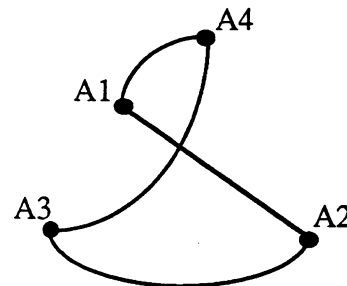
Pour maintenir la cohésion d'un bloc, les distances entre ses quatre points sont déterminées à partir des dimensions réelles du bloc. Ceci pose un problème de cohérence entre les distances inter-blocs et les distances intra-blocs, en particulier la différence d'ordre de grandeur entre les deux catégories doit être réaliste.

A cette difficulté vient s'ajouter l'existence de configurations ne respectant pas la structure du bloc, l'exemple suivant permet de s'en rendre compte.

Exemple :



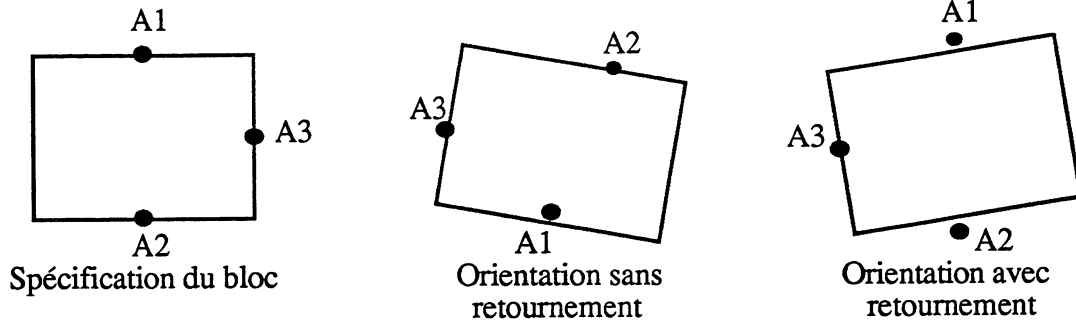
Spécification du bloc



Résultat de la projection

La projection des points sur le plan conduit à une torsion du bloc irréalisable en pratique. Il faut alors décider de la nécessité d'un retournement du bloc en fonction de la position des points.

Afin d'éviter cela, on peut modifier la représentation d'un bloc pour l'ATP, en n'utilisant que trois points situés chacun sur un côté différent du bloc. Les configurations irrégulières sont ainsi éliminées, tout en prenant en compte les retournements de blocs suggérés par l'ATP. Ceux-ci sont détectés par un ordre des points différent de l'initial, dans le sens trigonométrique.

Exemple :

Une fois l'ATP effectuée, on doit déterminer la position et l'orientation de chaque bloc à partir des projections des différents points qui le composent. A ce premier problème s'ajoute le fait que l'orientation obtenue conduit rarement à avoir les côtés des blocs parallèles aux bords du circuit. On doit alors déterminer parmi les huit orientations autorisées, celle qui s'en rapproche le plus pour chaque bloc.

On voit que l'on est finalement ramené à procéder de la même façon que pour l'orientation après placement, avec l'inconvénient supplémentaire de l'accroissement du temps de calcul de l'ATP dû à l'augmentation non négligeable de la taille de la matrice des distances.

Cette approche est peut être avantageuse si l'on ne considère pas que l'orientation du circuit par rapport au nuage est celle du repère orthogonal. Dans ce cas il faut arriver à déterminer l'orientation du circuit qui maximise le nombre de blocs ayant des côtés parallèles à ses bords.

VI.2 Compaction du circuit

La compaction du circuit a pour objectif de réduire sa surface tout en conservant la disposition des blocs les uns par rapport aux autres. Pour cela les écarts entre les blocs sont supprimés quand cela est possible. Afin de conserver des canaux de routage, les dimensions des blocs sont augmentées durant cette phase (voir V.6).

VI.2.1 Approche analytique

Les blocs sont représentés par un ensemble de $2n$ lignes horizontales et $2n$ lignes verticales dans le repère orthonormé (n étant le nombre de blocs). On associe aux lignes verticales des abscisses x_i et aux lignes horizontales des ordonnées y_i . Les contraintes à respecter durant la compaction s'énoncent alors comme deux systèmes, l'un sur les x_i et l'autre sur les y_i . Considérons celui des contraintes horizontales:

Les variables sont les abscisses x_i , leurs valeurs initiales sont données par la

disposition initiale des blocs. On se ramène au cas où tous les x_i sont positifs et on pose $x_{\max} = \max_{i=1..2n} (x_i)$.

Alors les contraintes sont du type:

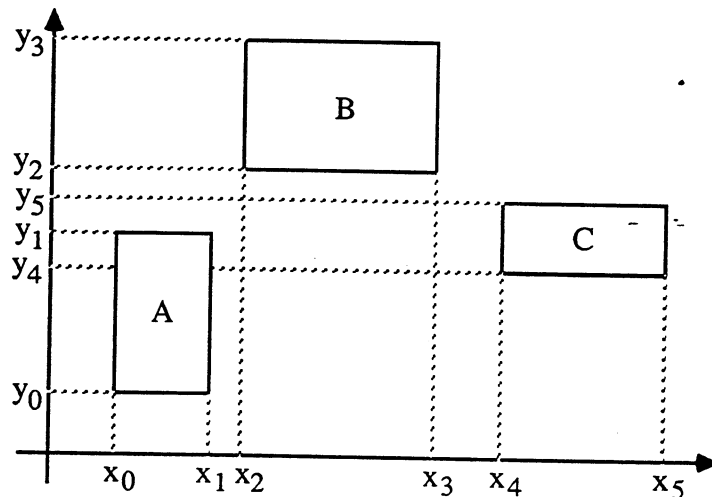
$$0 \leq x_i \leq x_{\max}$$

$$x_j - x_i \geq C_{ij} \quad \text{distance minimale entre deux lignes verticales.}$$

$$x_j - x_i \leq C_{ij} \quad \text{distance maximale entre deux lignes verticales.}$$

$$x_j - x_i = C_{ij} \quad \text{distance exacte entre deux lignes verticales.}$$

Exemple :



On obtient pour les contraintes horizontales:

$$x_1 - x_0 = l_A$$

$$x_4 - x_0 \geq l_A$$

$$x_5 - x_0 \geq l_A + l_C$$

$$x_4 - x_1 \geq 0$$

$$x_5 - x_1 \geq l_C$$

$$x_3 - x_2 = l_B$$

$$x_5 - x_4 = l_C$$

où l_A , l_B , l_C , sont les largeurs respectives des blocs A, B et C.

Il est facile de voir qu'un changement dans les valeurs des y_i (ie un déplacement vertical des blocs) peut entraîner une modification dans les contraintes sur les x_i . Si dans l'exemple on descend le bloc B jusqu'à ce qu'il soit en regard du bloc C alors:

$x_4 - x_1 \geq 0$ devient $x_4 - x_1 \geq l_B$ et de nouvelles contraintes apparaissent, dont la contrainte $x_4 - x_3 \geq 0$.

Or le problème général de compaction implique de traiter simultanément le système des contraintes horizontales et le système des contraintes verticales, ce qui correspond à déplacer simultanément les blocs horizontalement et verticalement. Il a été démontré que générer un placement de surface minimale avec de tels déplacements est un problème NP-complet [SHIN,

SANGIOVANNI-VINCENTELLI&SEQUIN 86].

Pour cette raison, l'approche généralement employée pour résoudre le problème de compaction en dimension 2 consiste à le décomposer en une séquence alternée de compactations en dimension 1, horizontales ou verticales.

Considérons le cas d'une compaction horizontale avec le système de contraintes associé. Alors une solution optimale est un ensemble de valeurs des x_i qui vérifie le système des contraintes et qui minimise $x_{\max} - x_{\min}$, où:

$$x_{\max} = \max_{i=1\dots 2n} (x_i) \text{ et } x_{\min} = \min_{i=1\dots 2n} (x_i).$$

Ceci constitue un problème de programmation en nombres entiers. Ce type de problème est connu pour être NP-complet dans le cas général, mais dans notre application chaque contrainte ne fait intervenir que deux variables, ce qui permet d'utiliser une technique de résolution particulière [JACKSON 89] :

On peut associer un graphe orienté pondéré au système des contraintes.

Les sommets du graphe sont les x_i et chaque arc (x_i, x_j) pondéré par C_{ij} représente la relation $x_j - x_i \geq C_{ij}$. Il est donc nécessaire de transformer le système des contraintes de façon à n'obtenir que des relations du type \geq .

D'après la construction du graphe, pour tout chemin P reliant x_i à x_j on doit avoir $x_j - x_i$ qui est supérieur au poids de P . De même, pour tout cycle C passant par x_i , alors $x_i - x_i$ doit être supérieur au poids de C . Donc le poids d'un cycle doit être négatif ou nul.

Afin de caractériser la solution optimale pour les valeurs des x_i on introduit un point initial O dans le graphe, ainsi que tous les arcs (O, x_i) , chacun de ces arcs ayant un poids nul. Alors la valeur optimale pour chaque x_i est le poids maximum d'un chemin partant de O et finissant à x_i .

D'où l'algorithme suivant pour déterminer les valeurs optimales des x_i :

1. Transformer toutes les contraintes en contraintes du type $x_j - x_i \geq C_{ij}$.
Initialiser toutes les valeurs des x_i à zéro: $\forall i \quad V(x_i) = 0$.
 $k = 0$
2. Pour chacune des contraintes $x_j - x_i \geq C_{ij}$ faire

$$\{ V(x_j) \leftarrow \max(V(x_j), V(x_i) + C_{ij}) \}$$

$$k \leftarrow k + 1$$
3. Si aucune valeur des x_i n'a été modifiée alors la solution courante est optimale. Fin.
4. Si des valeurs des x_i ont été modifiées et que l'on a $k \leq n$, aller en 2. .
Sinon, il existe un cycle de poids positif, donc pas de solution. Fin.

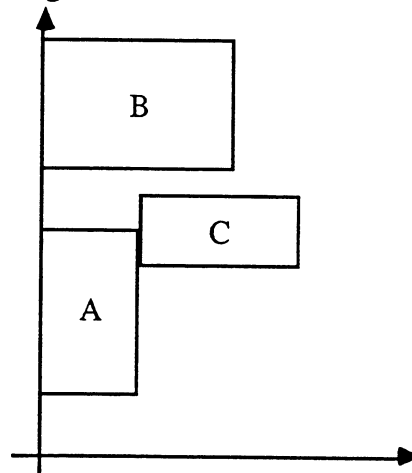
Sur l'exemple précédent on obtient les nouvelles contraintes:

$$\begin{aligned}x_1 - x_0 &\geq l_A \\x_0 - x_1 &\geq -l_A \\x_4 - x_0 &\geq l_A \\x_5 - x_0 &\geq l_A + l_C \\x_4 - x_1 &\geq 0 \\x_5 - x_1 &\geq l_C \\x_3 - x_2 &\geq l_B \\x_2 - x_3 &\geq -l_B \\x_5 - x_4 &\geq l_C \\x_4 - x_5 &\geq -l_C\end{aligned}$$

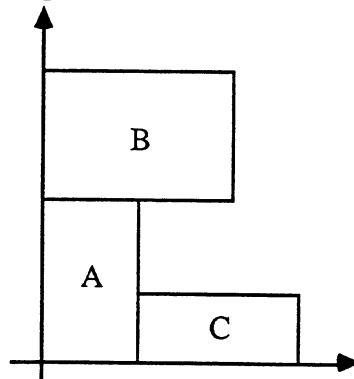
Le déroulement de l'algorithme est le suivant:

	x_0	x_1	x_2	x_3	x_4	x_5
$k = 0$	0	0	0	0	0	0
$k = 1$	0	l_A	0	l_B	l_A	$l_A + l_C$
$k = 2$	0	l_A	0	l_B	l_A	$l_A + l_C$

Ce qui correspond à la configuration suivante des blocs:



Si l'on effectue ensuite la compaction verticale on obtiendra:



Afin de respecter plus fidèlement les positions relatives des blocs il est

préférable d'alterner plusieurs fois compaction horizontale et compaction verticale, en limitant à chaque itération l'amplitude des déplacements.

VI.2.2 Algorithme implanté

L'algorithme de compaction utilisé dans l'outil de placement fonctionne selon le principe précédent : les blocs sont déplacés progressivement vers le point le plus en bas et à gauche du circuit, par des successions de petits déplacements horizontaux et verticaux.

L'algorithme de compaction vers l'angle bas et gauche du rectangle englobant le circuit est donné ci-dessous.

Les valeurs ε et k sont des paramètres de l'algorithme caractérisant l'amplitude des déplacements et par voie de conséquence la rapidité d'exécution de l'algorithme. Expérimentalement ces valeurs ont été fixées à $\varepsilon = 0,05$ et $k = 10$.

Les notations xg_i et yb_i représentent respectivement l'abscisse et l'ordonnée de l'angle bas et gauche du bloc i .

Compaction vers le bas et vers la gauche :

1. Déterminer l'angle bas et gauche du plus petit rectangle englobant le circuit (noté G) et classer les blocs dans l'ordre croissant de leur distance avec ce point ($d(G,i) \leq d(G,i+1)$).

2. Pour $i=1$ à n faire

{Déterminer la plus petite largeur du couloir entre le bloc i et un bloc en regard de lui, à sa gauche. Soit L cette largeur.

Si aucun bloc n'est à gauche de i alors $L = (xg_i - x_G)$.

Si $L > \varepsilon$ alors déplacer le bloc i de $\frac{L}{k}$ unités sur la gauche.

Déterminer la plus petite hauteur du couloir entre le bloc i et un bloc en regard de lui, en dessous. Soit H cette hauteur.

Si aucun bloc n'est en dessous de i alors $H = (yb_i - y_G)$.

Si $H > \varepsilon$ alors déplacer le bloc i de $\frac{H}{k}$ unités vers le bas.

}

3. Si un bloc a été déplacé à l'étape 2. alors recommencer 2. .

Un algorithme identique de compaction vers l'angle haut et droit du rectangle englobant le circuit est utilisé en alternance avec le précédent dans l'algorithme global de compaction. Ceci dans le but d'obtenir le meilleur emboîtement possible des blocs.

Algorithme de compaction :

1. Calculer le plus petit rectangle englobant le circuit.
2. Effectuer une compaction vers l'angle bas et gauche du rectangle englobant.
Mémoriser les positions des blocs. Calculer le nouveau rectangle englobant.
 $S \leftarrow$ surface du rectangle englobant
3. Faire
 - {Surf-min \leftarrow S
 - Etirement du rectangle englobant, vers le haut:
Si ($HR < LR$) alors $HR \leftarrow 1,5 \times HR$
 - Etirement du rectangle englobant, vers la droite:
Si ($LR \leq HR$) alors $LR \leftarrow 1,5 \times LR$
 - Effectuer une compaction vers la nouvelle position de l'angle haut et droit du rectangle englobant.
 - Effectuer une compaction vers l'angle bas et gauche du rectangle englobant.
 - Calculer la surface S du nouveau rectangle englobant.
 - Si ($S < Surf-min$) alors mémoriser les positions des blocs.
 - }
- Tant que ($1,01 \times S < Surf-min$)
4. Positions des blocs \leftarrow mémorisation.

Cet algorithme itère les compactations alternées tant qu'une amélioration significative de la surface est obtenue. L'étirement du rectangle englobant est destiné à donner une plus grande liberté de mouvements aux blocs, afin de favoriser une amélioration éventuelle de la surface.

VII. EXPÉRIMENTATION

Afin d'illustrer les capacités de l'analyse des proximités à pouvoir retrouver d'éventuelles structures sous-jacentes dans la connectique d'un circuit, nous avons expérimenté le placement qu'elle fournit dans le cas où le circuit a une structure de maillage. Le résultat est plus que probant puisque la disposition des blocs fournie par la projection du nuage de points correspond exactement à la structure de la connectique.

Les représentations graphiques des circuits sont données en annexe.

Pour le circuit de 4×4 blocs, deux résultats différents sont obtenus selon l'ordre des blocs dans la matrice des proximités. Dans les deux cas la disposition des blocs au sein de la projection du nuage respecte la structure du

maillage, mais les procédures de décompaction et d'orientation ne permettent pas toujours d'obtenir l'optimum.

Sur le circuit de 5×5 blocs est illustrée l'utilité de l'algorithme de calcul des distances entre blocs non connectés. On voit que lorsque celles-ci sont établies à partir de la formule, la structure du maillage n'est retrouvée que partiellement.

La qualité du placement dépendant étroitement de la distance utilisée lors de la construction de la matrice de proximités, nous avons expérimenté chacune des trois distances étudiées dans la partie IV.1 sur un ensemble de neuf circuits. L'ensemble des résultats est présenté dans le tableau ci-dessous.

Circuit	S ₁	L ₁	S ₂	L ₂	S ₃	L ₃
genc.1.1	211906112	6009952	240649616	6454693	250391616	6963288
genc.1.2	176214320	8650483	148476800	7491858	223559728	9748442
genc.1.3	118005600	4370413	155041824	5456332	151149536	5494954
genc.1.4	137992224	5230919	111163840	4644381	155626784	5248023
genc.1.5	88092016	2294832	58842464	1990362	88964688	2304376
genc.1.6	327711520	15097091	285695040	15476563	277388352	13859879
genc.1.7	5590054	63018	5541698	61538	5285468	68454
genc.1.8	248016672	7123301	210318592	5860384	283240800	8867720
genc.1.9	237608784	5941700	277378528	7368009	174141312	5554510

S₁ et L₁ sont la surface du circuit obtenu et la longueur estimée des connexions, en prenant comme distance : $d_{ij}^1 = \frac{1}{N_{ij}}$

Les représentations graphiques des placements obtenus avec cette distance sont données en annexe.

S₂ et L₂ sont obtenues en prenant comme distance : $d_{ij}^2 = \frac{N_{*i} + N_{*j} - N_{ij}}{N_{ij}}$

S₃ et L₃ sont obtenues avec : $d_{ij}^3 = \frac{(N_{*i} - N_{ij})S_i + (N_{*j} - N_{ij})S_j}{(S_i + S_j)N_{ij}^2}$

Les valeurs encadrées sont les meilleurs résultats pour chaque circuit.

On remarque que concernant la longueur de connexion, les résultats de d_{ij}^3 sont toujours moins bons que ceux de d_{ij}^1 et d_{ij}^2 , sauf pour les circuits genc.1.6 et

genc.1.9. Ces résultats médiocres confirment les observations faites dans la partie IV.1.2 .

En ce qui concerne les deux autres distances, on remarque que d_{ij}^1 donne de meilleurs résultats que d_{ij}^2 sur les circuits genc.1.1, genc.1.3, genc.1.6 et genc.1.9 tandis que d_{ij}^2 est meilleure sur les cinq autres. Nous avons vu, dans la partie IV.1.1, que d_{ij}^2 peut s'avérer moins bonne que d_{ij}^1 dans certains cas particuliers, ce qui semble contraire à l'expérimentation. Il ressort néanmoins que ces deux distances sont globalement équivalentes du point de vue de la qualité des résultats.

VIII. CONCLUSION

Nous avons vu dans ce chapitre, une méthode originale de placement, utilisant une technique statistique : l'analyse d'un tableau de distances. Le placement par ATP avait déjà fait l'objet d'une thèse [CHAISEMARTIN 86], et ses débuts étaient prometteurs. Le but de l'étude menée dans ce chapitre était de faire le point sur cette méthode, afin d'en dégager les points forts et les points faibles, et de réaliser les améliorations possibles en vue d'une implantation dans un outil d'aide à la conception de circuits.

De l'analyse effectuée il ressort que la méthode de placement par ATP peut difficilement fournir un placement final, même suivie des procédures d'orientation et de compaction des blocs.

Tout d'abord, seule la minimisation du routage est prise en compte lors du placement, et ceci par la définition de proximités ne considérant pas la surface des blocs et n'imposant aucune contrainte de régularité. Cela a pour conséquence de donner des circuits très désordonnés qui occupent beaucoup trop de surface.

Afin d'être cohérent, l'orientation des blocs est ensuite effectuée uniquement selon un critère de connectique et la compaction a été élaborée pour conserver autant que possible les positions relatives des blocs, il n'y a donc pas de modifications profondes du placement durant les phases suivantes.

Mais surtout le placement par ATP ne permet pas de traiter les contraintes telles que les blocs fixes et les distances critiques entre deux blocs. Les plots d'entrées-sorties ne peuvent pas non plus être traités car il n'existe aucun moyen d'imposer leur placement sur la périphérie du circuit.

Tout ceci fait du placement par ATP une méthode très rigide, ne permettant aucun contrôle sur le résultat de la projection. La seule liberté réside dans la

définition de la matrice des distances, dont on ne sait pas comment elle sera respectée sur le plan.

Néanmoins le placement par ATP peut être considéré comme un bon outil de conseil sur la disposition des blocs en fonction de la connectique. Les résultats peuvent être excellents là où d'autres techniques échouent, le meilleur exemple en est le maillage régulier. C'est la raison pour laquelle cette méthode reste un des atouts du système d'aide à la conception de circuits que nous avons développé [CHOTIN,MIGNOTTE&SAUCIER 90], le concepteur peut ensuite faire évoluer le placement initial à l'aide des commandes interactives de déplacement et de rotation de blocs.

En dernier point, il est important de remarquer que l'ATP se prête beaucoup plus à un placement en dimension 1, à savoir le placement de cellules standards sur une ligne. En effet, il suffit de procéder de la même façon qu'en dimension 2, et de ne considérer que la projection sur le premier axe principal. Le problème de régularité du placement est alors éliminé, de même que le problème d'optimisation de la surface, la connectique étant le seul critère à optimiser dans un placement 1-D.

Le nombre de cellules est généralement beaucoup plus grand que dans un problème de placement classique, ce qui entraîne des matrices bien plus grosses pouvant occasionner des temps de calcul importants pour l'algorithme de Jacobi. Dans ce cas il pourrait être intéressant de le remplacer par l'algorithme QR, plus performant, ou bien de le paralléliser, puisque c'est une des particularités de cet algorithme.

CHAPITRE III

Placement automatique par bipartitionnements successifs

L'application des techniques de partitionnement au problème du placement des circuits intégrés date de nombreuses années. Ces techniques ont d'abord été appliquées dans le but de décomposer un problème initial en sous-problèmes moins complexes, approche connue sous le nom "Divide and Conquer". L'ensemble des modules du problème initial est ainsi divisé en plusieurs sous-ensembles fortement intraconnectés et faiblement interconnectés, ce qui permet d'appliquer indépendamment les techniques de placement sur chacun des sous-ensembles.

C'est en 1977 que M.A. Breuer présenta en premier une approche du placement entièrement basée sur le bipartitionnement d'un ensemble de modules [BREUER 77], et dénommée "min-cut placement". Le principe est d'adopter une stratégie descendante qui a pour but de traiter globalement tous les blocs du circuit en affinant progressivement leurs positions relatives, ceci jusqu'à ce que chaque bloc soit affecté à une position précise. Pour cela on procède à des bipartitions successives de la surface rectangulaire du circuit et de l'ensemble des modules, comme il est expliqué lors de la présentation générale des méthodes de placement (chapitre I).

Les différentes implantations du placement par bipartitionnements successifs diffèrent selon la façon dont est divisée la surface du circuit: à chaque étape une portion du circuit est divisée en deux, selon une *ligne de coupe*. Celle-ci peut être horizontale ou verticale, et selon sa position les deux sous-portions seront égales ou non.

Le procédé de bipartitionnement est, lui, identique quelle que soit la stratégie de découpe du circuit; il faut à chaque itération bipartitionner un ensemble de modules en deux sous-ensembles faiblement interconnectés. Les techniques de bipartitionnement sont relativement peu nombreuses, et c'est en définitive toujours la même qui est utilisée, bénéficiant de quelques améliorations au cours des années.

Dans une première partie, nous formaliserons le problème de bipartitionnement qui nous intéresse, et nous développerons la technique de résolution utilisée. Cette dernière nécessitant de partir d'une bipartition initiale, nous proposerons une étude de plusieurs méthodes de construction d'une telle bipartition. Nous décrirons ensuite la stratégie de placement choisie pour l'implantation, en justifiant les choix effectués. Enfin nous tirerons profit de la flexibilité de cette méthode de placement en proposant des méthodes pour le traitement de contraintes supplémentaires comme l'introduction de plots

d'entrées-sorties autour du circuit ou de blocs fixes à l'intérieur de celui-ci. Une prise en compte des blocs déformables durant le processus d'optimisation du placement sera également proposée.

Pour conclure, une expérimentation sera effectuée sur l'ensemble des circuits utilisés lors du placement par A.T.P, afin de juger de la qualité des résultats en comparaison de ceux obtenus au chapitre II.

I. BIPARTITIONNEMENT D'UN ENSEMBLE DE MODULES

Le placement énoncé dans [BREUER 77] repose entièrement sur le partitionnement d'un ensemble de modules en deux sous-ensembles, généralement appelé *bipartitionnement*. Celui-ci doit être fait de façon à minimiser le nombre de connexions reliant des modules n'appartenant pas au même sous-ensemble. Cet ensemble de connexions est appelé *coupe* de la bipartition. Les deux sous-ensembles doivent généralement être de tailles équivalentes.

I.1 Modélisation

Le rapprochement entre le bipartitionnement d'un graphe et celui d'un ensemble de modules semble évident. Néanmoins, quelques précisions sont nécessaires.

Lorsque le circuit n'admet que des connexions bi-terminales, c'est-à-dire qui ne connectent que deux blocs, il existe une bijection entre les arêtes du graphe de connectivité et les connexions du circuit (chapitre I). De ce fait lorsqu'une bipartition du graphe est constituée, le nombre d'arêtes entre les deux sous-ensembles correspond exactement au nombre de connexions qui vont passer d'un sous-ensemble à l'autre. Ainsi le bipartitionnement de l'ensemble des modules se ramène au bipartitionnement du graphe de connectivité:

Soit $G = (V, E)$ le graphe de connectivité du circuit, tel que $|V| = n$ (n est le nombre de modules) et $|E| = p$ (p est le nombre de connexions entre les modules).

On cherche $A \subset V$ ($A \neq \emptyset$) et $B \subset V$ ($B \neq \emptyset$) tels que:

- 1) $A \cup B = V$
- 2) $A \cap B = \emptyset$
- 3) $-T \leq |A| - \frac{1}{2} (|A| + |B|) \leq T$

où $T \in \mathbb{R}$ est fixé, et est appelé la *tolérance* de la bipartition.

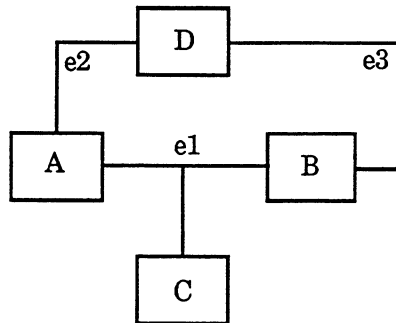
et qui minimisent $|C|$, où:

$$C = \{e \in E / e=(v_i, v_j) \text{ tq } v_i \in A \text{ et } v_j \in B\}$$

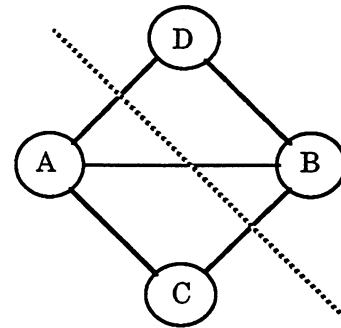
La troisième contrainte est une contrainte sur la taille des deux sous-ensembles issus de la bipartition, également appelée *contrainte d'équilibrage*.

Si le circuit comporte des équipotentiels à plus de deux terminaisons, alors celles-ci sont représentées par des cliques dans le graphe de connectivité. De ce fait il n'y a plus de bijection entre les arêtes du graphe et les connexions du circuit, ce qui entraîne que $|C|$ ne représente plus le nombre de connexions qui vont passer d'une partition à l'autre, mais constitue un majorant du nombre de ces connexions. De ce fait, s'il existe plusieurs bipartitions qui minimisent $|C|$, il n'est pas possible, à partir du graphe, de savoir lesquelles minimisent effectivement le nombre de connexions.

Exemple :



Circuit comportant une connexion tri-terminale.



graphe de connectivité associé.

La coupe sur le graphe intersecte trois arêtes alors que seules les connexions e1 et e2 sont coupées.

Ce problème peut être résolu en considérant l'hypergraphe de connectivité du circuit, dont les sommets sont les blocs, et les arêtes sont les équipotentiels entre ces blocs. Il y a dans ce cas bijection entre les arêtes du graphe et les connexions du circuit. Le problème est alors de regrouper les sommets de l'hypergraphe en deux ensembles disjoints, de façon à minimiser le nombre des arêtes passant d'un ensemble à l'autre:

Soit $H = (V, \mathfrak{E})$ l'hypergraphe de connectivité du circuit. $\mathfrak{E} = \{E_1, E_2, \dots, E_p\}$ est une famille de parties de V dont chaque élément E_i correspond à une équipotentielle distincte et contient l'ensemble des modules touchés par cette équipotentielle.

On cherche $A \subset V$ ($A \neq \emptyset$) et $B \subset V$ ($B \neq \emptyset$) tels que:

- 1) $A \cup B = V$
- 2) $A \cap B = \emptyset$
- 3) $-T \leq |A| - \frac{1}{2}(|A| + |B|) \leq T \quad (T \in \mathbb{R})$

et qui minimisent $|C|$, où:

$$C = \{E_k \in \mathcal{E} / \exists v_i, v_j \in E_k \text{ tq } v_i \in A \text{ et } v_j \in B\}$$

I.2 Amélioration d'une bipartition initiale

On considère l'hypergraphe de connectivité du circuit avec:

n = nombre de modules

et p = nombre de connexions.

Aucun algorithme de temps polynomial n'est connu à ce jour pour trouver la bipartition (A,B) , solution du problème précédent. De plus si l'on impose que deux sommets particuliers s et t de V soient respectivement dans A et B alors le problème a été prouvé comme NP-complet [GAREY&JOHNSON 79]. Cette contrainte supplémentaire est généralement nécessaire, nous nous en rendrons compte ultérieurement.

Pour cette raison, les méthodes utilisées sont toutes des méthodes itératives qui améliorent à chaque pas une bipartition initiale. Nous verrons par la suite comment cette bipartition peut être obtenue.

Définitions :

- Étant donnée une bipartition (A,B) , on appelle *coupe* de cette bipartition, l'ensemble des connexions qui connectent à la fois des modules de A et B .
- On appelle *taille de la coupe* le nombre d'équipotentielles dans la coupe.
- Nous appellerons *segments*, les sous-ensembles A et B de modules, issus de la bipartition.

L'amélioration d'une bipartition consiste à réduire la taille de la coupe, ceci par des modifications de la répartition des modules dans les segments, tout en conservant respectée la contrainte d'équilibrage.

Kernighan et Lin proposèrent en 1970 un algorithme d'amélioration d'une bipartition, basé sur des échanges de modules entre les deux segments [KERNIGHAN&LIN 70]. A chaque itération, une paire de modules, dont chaque élément appartient à un segment différent, est sélectionnée selon une

fonction coût. L'échange de chaque module d'un segment vers l'autre est effectué et le processus peut être répété tant qu'il existe une paire de modules dont l'échange améliore la bipartition, ou bien tant qu'un critère d'arrêt n'est pas satisfait (par exemple tant qu'il existe une paire qui n'a pas été déplacée). Dans le second cas, les échanges qui détériorent la bipartition sont acceptés dans l'espoir de sortir des minima locaux. Une fois que toutes les paires ont été déplacées, la meilleure bipartition rencontrée durant la passe est retenue, et une nouvelle passe peut être effectuée à partir de la bipartition finale de la passe précédente.

Si l'on considère la taille d'un segment comme étant le nombre de modules qu'il contient, alors cet algorithme permet de conserver tout au long du processus, le même rapport de taille entre les deux segments. Il faut donc commencer avec une bipartition initiale respectant la contrainte de taille sur les segments.

L'inconvénient majeur de cet algorithme réside dans sa complexité. En effet à chaque itération, le nombre de paires de modules à évaluer pour le choix de l'une d'entre elles est de l'ordre de $(\frac{n}{2})^2$, si n est le nombre total de modules et si les segments ont des cardinalités identiques.

En 1982, Fiduccia et Mattheyses proposèrent un algorithme d'amélioration d'une bipartition dont la complexité de chaque passe est linéaire par rapport à la taille du circuit [FIDUCCIA&MATTHEYSES 82]. Cet algorithme est celui qui est le plus utilisé de nos jours.

Étant donné une bipartition (A,B) des modules vérifiant la contrainte d'équilibrage, l'idée principale est de déplacer à chaque itération un seul module d'un segment à l'autre, dans le but de réduire le nombre de connexions touchant des modules dans les deux segments. Cette idée avait déjà été suggérée auparavant dans [SHIRAIISHI&HIROSE 80] mais sans algorithme optimisé.

L'approche précédente entraîne inévitablement une modification du rapport de taille des segments et le déplacement d'un module à chaque itération n'est possible que si la contrainte d'équilibrage reste vérifiée.

I.2.1 Contrainte d'équilibrage

Afin de pouvoir traiter les circuits dont les modules n'ont pas tous la même surface, il est nécessaire de reconsidérer la contrainte d'équilibrage.

Définitions :

- A chaque module v_i est associée sa surface $s(v_i)$.

- La taille d'un segment est définie comme la somme des surfaces de ses éléments.

Pour une bipartition (A,B) des modules la contrainte sur la taille des segments devient:

$$-T \leq S(A) - \frac{1}{2} S_t \leq T \quad T \in \mathbb{R} \text{ est la tolérance de la bipartition.}$$

$$\text{avec } S(A) = \sum_{v_i \in A} s(v_i) \quad \text{et} \quad S_t = \sum_{i=1}^n s(v_i)$$

où n désigne le nombre total de modules.

Il est évident que la contrainte d'équilibrage doit être suffisamment lâche pour permettre de déplacer un grand nombre de modules.

Si $S_t \gg \max_{i=1..n} (s(v_i))$ on prend $T = \max_{i=1..n} (s(v_i))$. De cette façon, lorsque la contrainte est respectée, il est toujours possible de déplacer soit n'importe quel module appartenant au segment A, soit n'importe quel module appartenant au segment B:

Si $S(A) \in [\frac{1}{2} S_t - T ; \frac{1}{2} S_t]$ alors au moins chacun des modules de B peut être déplacé vers A sans violer la contrainte d'équilibrage.

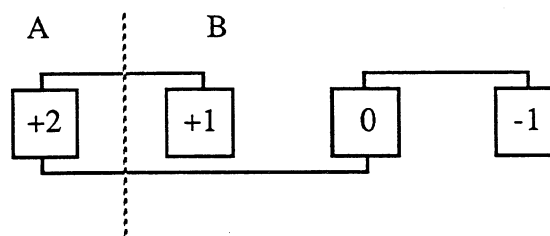
Si $S(A) \in [\frac{1}{2} S_t ; \frac{1}{2} S_t + T]$ alors au moins chacun des modules de A peut être déplacé vers B sans violer la contrainte d'équilibrage.

Le choix du module à déplacer à chaque itération est fait de façon à réduire au maximum la taille de la coupe, tout en respectant la contrainte d'équilibrage. Afin de guider ce choix, un gain représentant la variation éventuelle de la coupe est associé à chaque module.

I.2.2 Gain d'un module

Le *gain* d'un module est égal à la variation de la taille de la coupe si on déplace ce module du segment où il se trouve vers l'autre segment.

Exemple :



Gain des modules

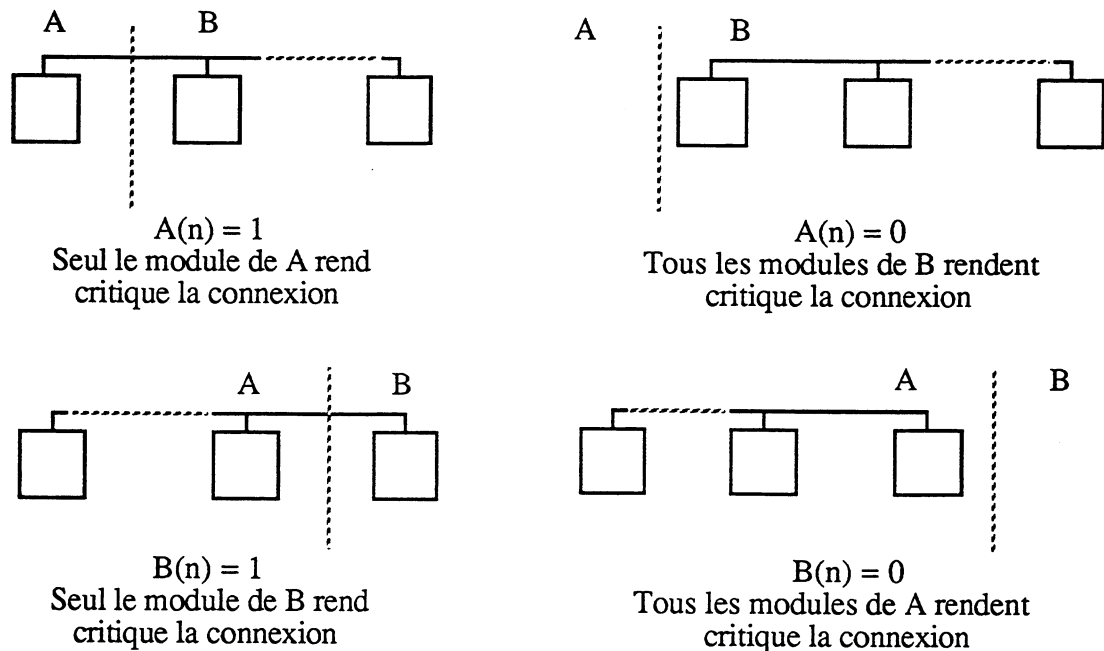
Afin de formaliser la définition du gain pour une cellule nous devons

introduire quelques notions.

Définitions :

- On appelle *incidence* d'une connexion n par rapport à un segment A , le nombre de modules de A sur lesquels n est connectée, noté $A(n)$.
- La coupe d'une bipartition (A,B) étant l'ensemble des connexions qui connectent à la fois des modules de A et B , on dit qu'une connexion n est *critique* par rapport à une bipartition s'il existe un module de cette connexion qui, si on le déplace, fait entrer ou sortir n de la coupe. On dit qu'un tel module rend critique la connexion.

Il est facile de voir qu'une connexion n est critique par rapport à une bipartition (A,B) si au moins une des deux incidences $A(n)$ ou $B(n)$ est égale à 0 ou 1. Les quatre cas possibles sont les suivants:



Propriété :

Le gain d'un module m peut être exprimé en fonction des connexions qu'il rend critiques, chacune de ces connexions contribuant de $+1$ ou -1 au gain de m :

Appelons S le segment source contenant le module m , et D le segment destination. Alors le gain du module m , si on le déplace de S vers D , est donné par:

$$g(m) = NS_1(m) - ND_0(m)$$

où $NS_1(m)$ désigne le nombre de connexions n de m vérifiant $S(n)=1$ et $ND_0(m)$ désigne le nombre de connexions n de m vérifiant $D(n)=0$.

L'algorithme d'initialisation des gains est alors le suivant:

```

Pour chaque module m faire
  {g(m) ← 0
  S ← segment source de m
  D ← segment destination de m
  Pour chaque connexion n de m faire
    {Si S(n) = 1 alors g(m) ← g(m) + 1
    Si D(n) = 0 alors g(m) ← g(m) - 1
    }
  }

```

I.2.3 Mise à jour des gains

Le gain d'un module est attaché à une bipartition, il faut donc opérer une phase de remise à jour des gains après chaque déplacement. Néanmoins, afin d'éviter le bouclage de l'algorithme, chaque module déplacé est ensuite figé dans son segment destination, il n'est donc pas nécessaire de garder à jour le gain des modules déplacés. Seuls les modules dits "libres", c'est-à-dire non encore déplacés doivent avoir un gain à jour.

Il est évident que le gain d'un module libre doit être réajusté seulement si une au moins des connexions de ce module a changé d'incidence sur les segments de la bipartition.

Définition :

- On dit que deux modules sont *voisins* s'ils ont au moins une connexion en commun.

Ainsi lorsqu'un module m est déplacé, seul le gain de ses voisins est susceptible d'être modifié.

Il est important de remarquer que seuls certains voisins nécessitent une remise à jour de leur gain.

En effet, une connexion qui n'est critique ni avant ni après le déplacement ne contribue ni avant ni après au gain des modules (d'après la propriété précédente), elle n'aura donc aucun effet sur la modification des gains des modules qu'elle connecte.

On en déduit qu'un voisin de m dont les connexions avec m ne sont critiques ni avant ni après le déplacement ne verra pas son gain modifié.

Seules les connexions de m qui sont critiques avant ou après le déplacement impliquent un réajustement des gains des modules qu'elles connectent.

De ce fait, on ne considère, pour le réajustement de gain, que les voisins de m

dont une au moins des connexions avec m est critique avant ou après le déplacement.

Étudions les particularités de ces connexions.

Considérons un module m que l'on déplacera du segment S vers le segment D . Avant le déplacement, m est dans S et une connexion n de m est critique si et seulement si:

$$S(n) = 1 \quad \text{ou} \quad D(n) = 0 \quad \text{ou} \quad D(n) = 1$$

Pour chacun des deux premiers cas, m est un module qui rend critique la connexion alors que, si seul le troisième cas est vérifié, c'est un module de D qui rend critique n . Le cas $S(n) = 0$ ne peut pas se produire car m est dans S .

De la même façon, après le déplacement, m est dans D et une connexion n de m est critique si et seulement si:

$$D(n) = 1 \quad \text{ou} \quad S(n) = 0 \quad \text{ou} \quad S(n) = 1$$

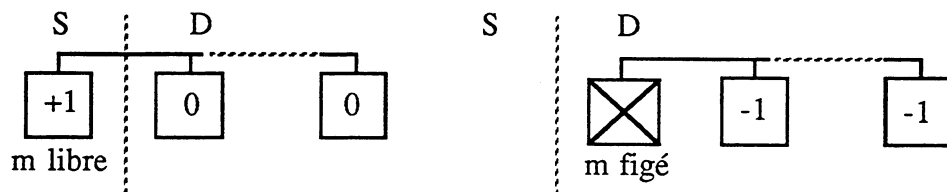
On remarque que puisque n est une connexion de m alors:

$S(n) = 1$ avant le déplacement de $m \Leftrightarrow S(n) = 0$ après le déplacement de m

$D(n) = 0$ avant le déplacement de $m \Leftrightarrow D(n) = 1$ après le déplacement de m

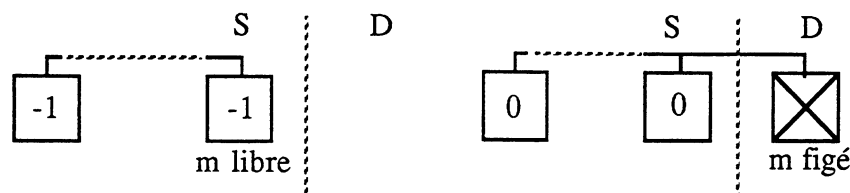
Les six cas de connexions critiques précédents se ramènent donc à quatre cas:

Cas 1 : $S(n) = 1$ avant le déplacement de $m \Leftrightarrow S(n) = 0$ après le déplacement de m



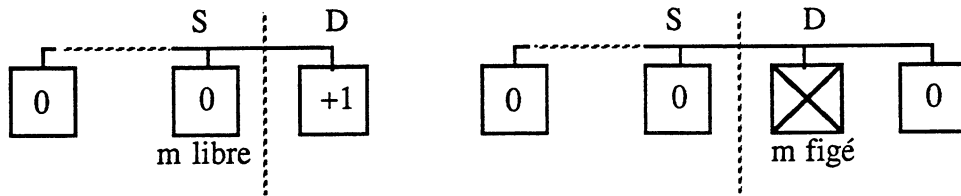
Après le déplacement de m on doit décrémenter de 1 le gain de chaque module libre de n .

Cas 2 : $D(n) = 0$ avant le déplacement de $m \Leftrightarrow D(n) = 1$ après le déplacement de m



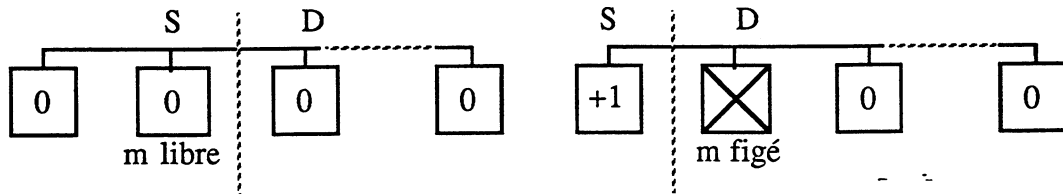
Après le déplacement de m on doit incrémenter de 1 le gain de chaque module libre de n .

Cas 3 : $D(n) = 1$ avant le déplacement de m



Après le déplacement de m on doit décrémenter de 1 le gain du module de n situé dans le segment D, s'il est libre.

Cas 4 : $S(n) = 1$ après le déplacement de m



Après le déplacement de m on doit incrémenter de 1 le gain du module de n situé dans le segment S, s'il est libre.

On remarque que si deux de ces situations se cumulent, on doit effectuer les réajustements nécessaires à chaque cas. Par exemple si les cas 1 et 3 se cumulent on doit décrémenter de 2 le module libre de n situé dans D.

Ceci nous conduit à l'algorithme suivant pour le déplacement du module m et la remise à jour des gains:

```

S ← segment source de m
D ← segment destination de m
Déplacer m et le figer dans le segment D.
Pour chaque connexion n de m faire
    { /* cas 2 et cas 3 */
    Si D(n) = 0 alors incrémenter de 1 le gain de chaque module libre de n
    Sinon Si D(n) = 1 alors décrémenter de 1 le gain du module de n situé dans
    le segment D, s'il est libre.
    /* Mise à jour des incidences de n */
    S(n) ← S(n) - 1
    D(n) ← D(n) + 1
    /* cas 1 et cas 4 */
    Si S(n) = 0 alors décrémenter de 1 le gain de chaque module libre de n.
    Sinon Si S(n) = 1 alors incrémenter de 1 le gain du module de n situé dans
    le segment S, s'il est libre.
    }

```

I.2.4 Amélioration itérative d'une bipartition

Étant donnée une bipartition initiale vérifiant la contrainte d'équilibrage, l'algorithme d'amélioration itérative consiste à déplacer à chaque itération le module de plus fort gain qui respecte la contrainte d'équilibrage; celui-ci est appelé *module de base*.

Si le plus fort gain devient négatif, on se trouve dans un minimum mais celui-ci peut être local. Le processus est donc poursuivi dans l'espoir de sortir de ce minimum pour converger vers un autre que l'on espère meilleur.

La taille de la coupe ne diminue donc pas systématiquement tout au long de l'algorithme et ce n'est pas forcément la dernière bipartition obtenue qui est la meilleure. Pour cette raison on doit tenir à jour, à chaque itération, la plus petite taille de coupe obtenue, afin de pouvoir y revenir éventuellement en fin d'algorithme, en déplaçant les modules dans l'ordre inverse.

Après chaque déplacement de module, la taille de la coupe doit donc être calculée.

Propriété :

Étant donnée une bipartition (A,B) et soit χ la taille de la coupe. Si l'on déplace un module m du segment où il se trouve vers l'autre segment alors la taille χ' de la nouvelle coupe est:

$$\chi' = \chi - g(m)$$

où $g(m)$ désigne le gain du module m .

Afin d'éviter le bouclage de l'algorithme, chaque module déplacé est ensuite figé dans le segment d'arrivée. Ainsi le nombre de modules dits "libres", c'est-à-dire que l'on peut déplacer, décroît au fur et à mesure des déplacements.

Lorsque tous les modules sont figés ou qu'aucun des modules libres ne peut être déplacé sans provoquer la violation de la contrainte d'équilibrage, le processus s'arrête. Néanmoins, le processus peut être stoppé-avant, si l'on est sûr de ne plus améliorer la bipartition lors des déplacements ultérieurs. Pour cela on introduit la notion d'équipotentielle figée par rapport à la bipartition.

Définition :

- Une équipotentielle est dite *figée* par rapport à une bipartition si elle connecte des modules figés dans chacun des segments. Cette équipotentielle fait alors partie de toutes les coupes qui dérivent de la bipartition par déplacements des modules libres.

Ainsi, lorsque le nombre d'équipotentielles figées est supérieur ou égal à la taille de la plus petite coupe rencontrée jusqu'alors, on est sûr que les coupes issues des déplacements ultérieurs auront des tailles supérieures ou égales et le processus s'arrête.

L'algorithme d'amélioration itérative de la bipartition est le suivant:

1. Initialiser tous les modules comme modules libres.
2. Initialiser les gains des modules avec l'algorithme du paragraphe I.2.2.
3. Initialiser à nulle la liste des modules déplacés.
4. Initialiser la taille de la coupe minimale à celle de la coupe initiale.
5. Tant que aucun des critères d'arrêt n'est vérifié Faire
 - {
 - Choisir comme module de base le module libre de gain maximal (positif ou négatif) pouvant être déplacé sans violer la contrainte d'équilibrage.
 - Si ce module existe Alors
 - {
 - Effectuer le déplacement du module et la mise à jour des gains avec l'algorithme du paragraphe I.2.3.
 - Figurer le module de base dans le segment d'arrivée et l'ajouter à la liste des modules déplacés.
 - Calculer la taille de la coupe et, si nécessaire, mettre à jour la taille de la coupe minimum.
 - Calculer le nombre d'équipotentiels figés. S'il est supérieur à la taille de la coupe minimum alors aller en 6. .
 - }
 - Sinon aller en 6. .
 - }
6. Reformuler la bipartition de coupe minimale en déplaçant les modules dans l'ordre inverse.

On remarque que lorsqu'aucune diminution de la coupe n'a été possible, les déplacements sont refaits dans l'ordre inverse jusqu'à ce que l'on arrive à une bipartition dont la coupe est égale à celle de la bipartition initiale.

Théorème : [FIDUCCIA&MATTHEYSES 82]

La complexité de l'algorithme d'amélioration itérative d'une partition est $O(N_c)$ avec:

$$N_c = \text{nombre total de connecteurs sur l'ensemble de modules} \\ = \sum_{v_i \in V} |\mathfrak{E}(v_i)|$$

où $\mathfrak{E}(v_i)$ désigne l'ensemble des connexions attachées au module v_i .

I.2.5 Choix du module de base

Afin d'effectuer rapidement le choix du module de base à chaque itération, Fiduccia et Mattheyses ont proposé une structure spécifique pour le stockage des modules libres. Cette structure s'apparente à un tri par tas des modules libres, les modules d'un même tas ayant des gains identiques.

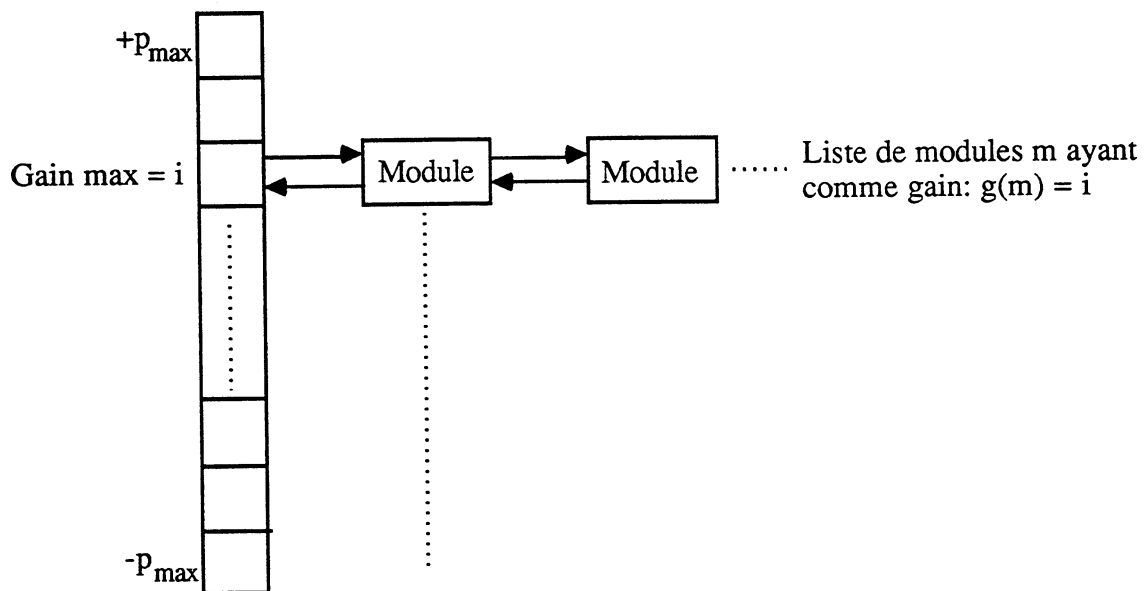
On note p_{\max} le nombre maximum d'équipotentiels sur un module:

$$p_{\max} = \max_{i=1..n} |\xi(v_i)|$$

où $\xi(v_i)$ désigne l'ensemble des connexions attachées au module v_i .

Alors le gain d'un module est toujours compris entre $-p_{\max}$ et $+p_{\max}$.

On utilise donc un tableau indicé de $-p_{\max}$ à $+p_{\max}$ pour stocker l'ensemble des tas. Chaque indice i est un pointeur sur une liste doublement chaînée de modules. Cette liste correspond au tas de modules ayant i pour valeur de gain. On conserve à chaque itération la valeur de l'indice correspondant à la liste de modules de gain maximum.



Structure de tri par tas.

La mise à jour du gain d'un module se fait alors en déplaçant le module de son ancienne liste vers la liste correspondant à la nouvelle valeur de son gain.

Un tel tableau est associé à chacun des segments A et B de la bipartition, de façon à ce que celui-ci ne contienne que les modules libres de son segment. Une fois le module de base déplacé, il est retiré du tableau et placé dans la liste des modules déplacés.

Le choix du module de base se fait de la façon suivante :

- Trouver le module de plus fort gain de A qui respecte la contrainte d'équilibrage.
- Trouver le module de plus fort gain de B qui respecte la contrainte d'équilibrage.
- Prendre parmi les deux modules celui de plus fort gain. Si les gains sont identiques alors prendre celui qui minimise la contrainte d'équilibrage.

Pour trouver dans un segment le module de gain maximal qui respecte la contrainte d'équilibrage, on doit parcourir les modules de son tableau dans l'ordre décroissant des gains, jusqu'à ce que l'un d'entre eux satisfasse la condition.

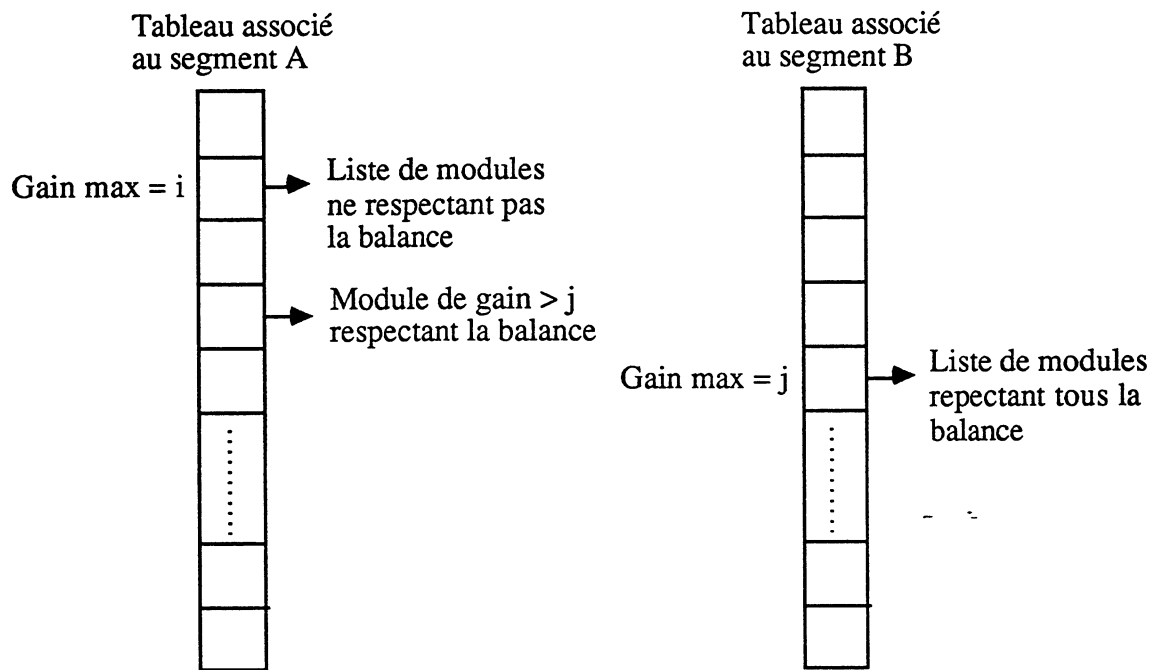
Dans le cas où la tolérance sur la contrainte d'équilibrage est $T = \max_{i=1..n} (s(v_i))$, il est possible de supprimer cette recherche, et donc d'augmenter la rapidité de l'algorithme [KRISHNAMURTHY 84].

En effet, nous avons vu au paragraphe I.2.1 qu'un tel choix de la tolérance permet de déplacer, soit n'importe quel module du segment A, soit n'importe quel module du segment B. S'il existe des modules libres dans les deux segments, alors forcément le module de gain maximal de l'un ou l'autre des segments peut être déplacé. Or, l'utilisation de la structure précédente permet d'obtenir sans aucune recherche le module de gain maximal de chaque segment. Il est donc possible de choisir comme module de base l'un ou l'autre des modules de plus fort gain de chaque segment.

Ainsi, tant qu'il existe des modules libres dans chaque segment, le choix du module de base se fait sans avoir à faire de recherches.

Néanmoins, il est important de noter que ce choix ne correspond pas forcément à celui du module libre de plus fort gain sur l'ensemble des modules libres des deux segments.

Une configuration comme celle illustrée sur la figure peut se produire.



Situation non optimale du choix sans recherche du module de base.

Sur la figure, le segment dont tous les modules peuvent être déplacés est le segment B. Aucun des modules de gain maximal du segment A ne peut être déplacé, mais il peut exister pour le tableau A un indice compris entre i et j dont un des modules de la liste associée ne viole pas la contrainte d'équilibrage durant son déplacement. Ce module a alors un gain supérieur au gain maximum du segment B.

I.2.6 Algorithme d'amélioration d'une bipartition initiale

Soit $H = (V, \mathcal{E})$ un hypergraphe et (A, B) une bipartition initiale de H vérifiant la contrainte d'équilibrage. On désire obtenir pour H une bipartition minimisant la taille de la coupe et vérifiant la contrainte d'équilibrage.

Nous avons présenté dans le paragraphe I.2.4, un algorithme pour l'amélioration itérative d'une bipartition. Cet algorithme peut être appliqué plusieurs fois de suite de la façon suivante:

Lors de la première passe, l'algorithme est appliqué sur la bipartition initiale (A, B) , puis pour chacune des passes suivantes il est appliqué sur le résultat obtenu lors de la passe précédente. Ce procédé est répété tant qu'il y a diminution de la coupe d'une passe sur l'autre.

A chaque début d'une nouvelle passe les modules sont tous réinitialisés comme modules libres, ceci permet des déplacements qui n'étaient pas possibles lors de

la passe précédente et donc peut amener à de nouvelles diminutions de la coupe.

Le nombre de passes est forcément inférieur ou égal à p car p est une borne supérieure à la taille d'une coupe. On en déduit que la complexité de l'algorithme précédent est $O(p.N_c)$ où N_c désigne le nombre total de connecteurs sur l'ensemble de modules.

Lorsque l'algorithme précédent ne peut plus optimiser la taille de la coupe, il est possible d'employer une stratégie initialement proposée par Kernighan et Lin. Il s'agit en quelque sorte de relancer le processus d'optimisation dans une direction "orthogonale" à celle qui vient d'être explorée.

Plus précisément, si l'on désigne par (A', B') la partition optimale obtenue par l'algorithme, on divise A' en deux parties A'_1 et A'_2 , et B' en deux parties B'_1 et B'_2 . On relance alors l'algorithme avec comme partition initiale une des partitions $(A'_1 \cup B'_1, A'_2 \cup B'_2)$ ou $(A'_1 \cup B'_2, A'_2 \cup B'_1)$. Si une bipartition meilleure que (A', B') est trouvée, on la divise en deux parties et on réitère le processus.

Chaque segment est séparé en deux de façon à ce que les deux parties aient le moins de connexions en commun. Pour cela on utilise l'algorithme précédent sans la stratégie d'optimisation, pour limiter l'augmentation du temps d'exécution.

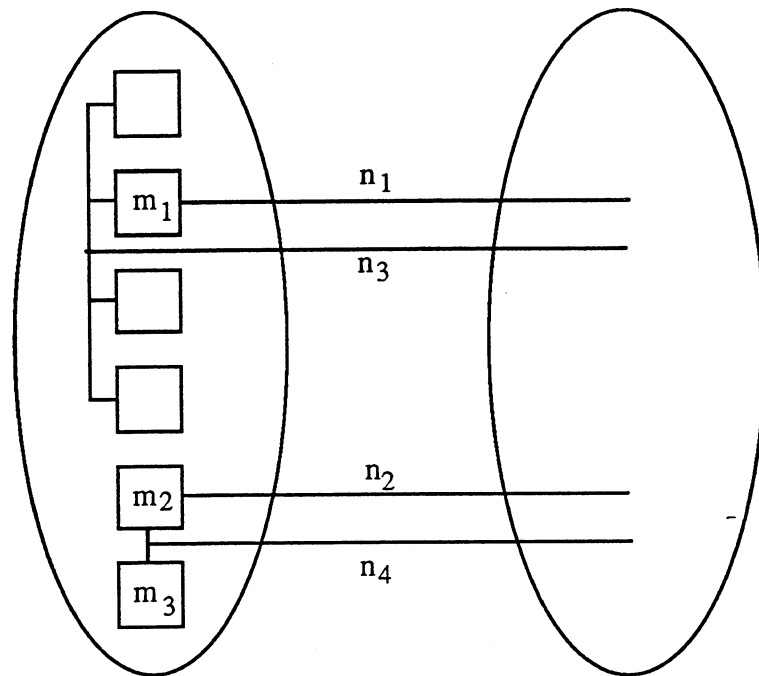
L'expérimentation montre que cette technique permet d'obtenir des tailles de coupe non atteintes par l'algorithme seul, et jusqu'à 30% inférieures, même quand ce dernier est exécuté sur un grand nombre de partitions initiales [KRISHNAMURTHY 84]. Ceci permet de justifier l'importante augmentation du temps d'exécution causée par cette technique.

1.2.7 Extension du concept de gain d'un module

Nous avons vu au paragraphe 1.2.4 l'algorithme d'amélioration d'une bipartition tel qu'il a été présenté par Fiduccia et Mattheyses [FIDUCCIA&MATTHEYSES 82]. Cet algorithme nécessite à chaque itération le choix du module de plus fort gain, le gain d'un module ayant été défini au paragraphe 1.2.2, comme la variation de la taille de la coupe si on déplace ce module du segment où il se trouve vers l'autre segment.

Or, bien souvent, le critère du plus fort gain ne suffit pas à dégager un seul candidat au déplacement. Il n'existe alors pas d'autre critère qui permette de choisir parmi les modules de plus fort gain celui qui aura la meilleure influence sur les diminutions ultérieures de la taille de la coupe.

Cet aspect est illustré sur la figure suivante :



Meilleur choix parmi plusieurs modules de gains identiques.

Les modules m_1 et m_2 sont tous les deux candidats au déplacement et chacun d'entre eux permet de supprimer une connexion de la coupe (respectivement n_1 et n_2). Ces deux modules ont donc le même gain, au sens où nous l'avons défini, et quel que soit le module déplacé les connexions n_3 et n_4 resteront dans la coupe. Néanmoins il est évident que le choix de m_2 est le plus avantageux car il permettra de supprimer n_4 de la coupe dès l'itération suivante par le déplacement de m_3 .

Le gain défini en I.2.2 ne permet pas de prendre en compte l'effet d'un déplacement sur les variations ultérieures de la coupe. Le choix entre des modules de gains identiques est donc fait de façon arbitraire par rapport à la connectique. Les conséquences de tels choix sur la taille de la coupe finale sont loin d'être négligeables.

A titre d'exemple, Krishnamurthy exécuta une centaine de fois l'algorithme décrit en I.2.6 sur un même circuit de 304 cellules et 346 connexions, avec à chaque fois une bipartition initiale obtenue aléatoirement. Chacun des choix arbitraire ayant été rendu parfaitement aléatoire, il trouva des coupes allant de 41 à 88 connexions avec une moyenne de 67 connexions.

En reprenant la même expérience mais avec le choix du module de plus fort gain guidé par les possibilités de diminution ultérieure de la coupe, il obtint des coupes allant de 39 connexions à 68 connexions avec une moyenne de 51

connexions [KRISHNAMURTHY 84].

Pour y parvenir Krishnamurthy généralisa le concept de gain d'un module afin de mieux prendre en compte l'effet d'un déplacement sur la taille de la coupe. Auparavant nous devons définir la notion de lien d'une connexion par rapport à un segment.

Définition :

Etant donné une connexion n quelconque et A un des segments de la bipartition. Le *lien* de la connexion n par rapport au segment A , noté $L_A(n)$, est défini par:

$$L_A(n) = \infty \quad \text{si } n \text{ connecte un module figé de } A$$

$$L_A(n) = A(n) = \text{incidence de } n \text{ sur } A \quad \text{sinon.}$$

Intuitivement le lien représente une mesure de l'attachement d'une connexion à un segment. Une valeur infinie du lien signifie que la connexion est figée par rapport au segment. Une connexion figée par rapport à une bipartition est telle que ses liens sur les deux segments sont infinis.

Nous pouvons maintenant introduire le concept de gain généralisé :

- Gain d'ordre 1 :

Étant donnée une bipartition (A,B) et m un module du segment A adjacent à une connexion n telle que $L_A(n) = 1$ et $L_B(n) > 0$. La connexion n fait partie de la coupe et disparaît de la coupe si on déplace m vers B . A l'inverse, si la connexion n est telle que $L_A(n) > 0$ (ce qui découle de nos hypothèses) et $L_B(n) = 0$ alors n ne fait pas partie de la coupe et le déplacement de m vers B va introduire n dans la coupe. On dit alors que, dans le premier cas, n contribue positivement au gain d'ordre 1 du module m , alors que dans le deuxième cas, n contribue négativement au gain d'ordre 1 de m .

- Gain d'ordre 2 :

Avec les mêmes hypothèses supposons maintenant que nous ayons $L_A(n) = 2$ et $L_B(n) > 0$. Le déplacement de m vers B permettra à n de contribuer positivement au gain d'ordre 1 du second module du segment A qui lui est connecté. Inversement si $L_A(n) > 0$ et $L_B(n) = 1$ alors le déplacement de m empêchera n de contribuer positivement au gain d'ordre 1 du module de B connecté à n . On dit alors que, dans le premier cas, n contribue positivement au gain d'ordre 2 du module m , alors que dans le deuxième cas, n contribue négativement au gain d'ordre 2 de m .

- Gain d'ordre i :

En généralisant les observations ci-dessus on obtient :

Définition :

$$g_i(m) = |\{n \in \mathfrak{E}(m) / L_A(n) = i \text{ et } L_B(n) > 0\}| \\ - |\{n \in \mathfrak{E}(m) / L_A(n) > 0 \text{ et } L_B(n) = i-1\}|$$

où $\mathfrak{E}(m)$ désigne l'ensemble des connexions attachées au module m .

Il est clair que le gain d'ordre i n'est défini que pour $1 \leq i \leq n_{\max}$, où n_{\max} désigne le nombre maximum de modules sur une connexion:

$$n_{\max} = \max_{i=1..p} |V(n_i)|$$

avec $V(n_i)$ = ensemble des modules attachés à la connexion n_i .

On remarque que pour contribuer positivement au gain d'ordre i d'un module une connexion ne doit pas être figée dans le segment de ce module.

Propriété :

Quel que soit le module m et quelle que soit la bipartition considérée, on a :

$$- p_{\max} \leq g_i(m) \leq p_{\max} \quad \text{pour tout } i \text{ tel que } 1 \leq i \leq n_{\max}$$

On rappelle que p_{\max} désigne le nombre maximum de connexions sur un module.

On considère maintenant le gain d'un module comme un vecteur de dimension k ($1 \leq k \leq n_{\max}$) dont les k composantes sont les gains d'ordre i ($i \leq k$) de m :

$$G(m) = (g_1(m), g_2(m), \dots, g_k(m))$$

On peut établir une relation d'ordre " \leq " entre des vecteurs gain de même dimension :

Soient $G(m_1)$ et $G(m_2)$ deux vecteurs gain de dimension k . On dit que $G(m_1) \leq G(m_2)$ si $G(m_1)$ et $G(m_2)$ sont identiques, ou s'il existe un indice i ($i \leq k$) tel que :

$$\forall h < i \text{ on a } g_h(m_1) = g_h(m_2) \\ \text{et } g_i(m_1) < g_i(m_2)$$

On démontre facilement que pour k fixé, la relation " \leq " définit un ordre total sur l'ensemble des gains des modules.

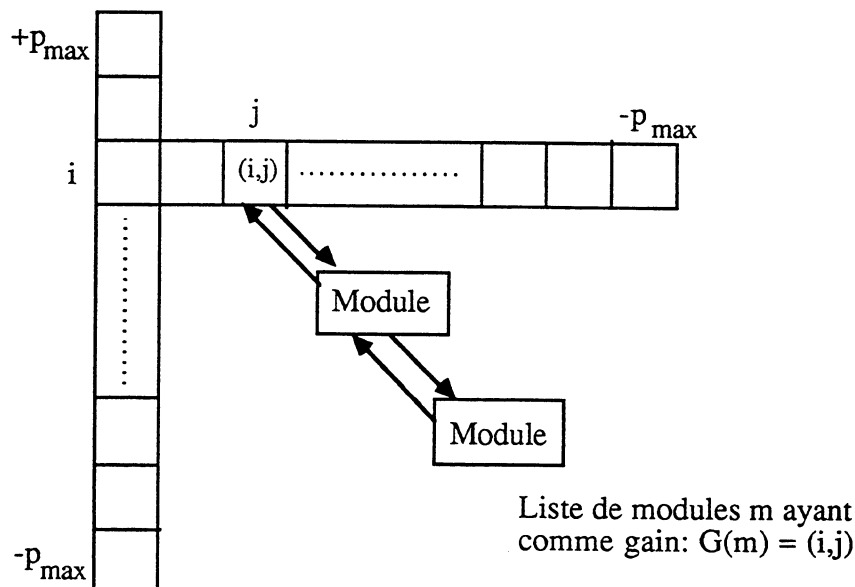
Propriété :

Le gain d'ordre 1 d'un module est égal à la variation de la coupe si on déplace ce module du segment où il se trouve vers l'autre segment. Ainsi si l'on appelle χ la taille de la coupe avant le déplacement d'un module m et χ' la taille de la coupe après son déplacement, alors:

$$\chi' = \chi - g_1(m)$$

Avant de pouvoir utiliser l'algorithme défini au paragraphe I.2.4, il est nécessaire de revoir les algorithmes d'initialisation et de mise à jour des gains, ainsi que la structure de données pour le stockage ordonné des modules libres. Cette dernière est une généralisation de la structure présentée au paragraphe I.2.5). On utilise un tableau à k dimensions, dont les indices vont de $-p_{\max}$ à $+p_{\max}$ pour chacune des dimensions. Chaque case de ce tableau est un pointeur sur une liste doublement chaînée de modules ayant comme valeur de gain d'ordre i pour $1 \leq i \leq k$, l'indice de la case dans la $i^{\text{ème}}$ dimension du tableau.

La figure suivante représente une telle structure pour $k = 2$.



Structure de tri par tas pour des gains de dimension 2.

Comme précédemment, un tel tableau est associé à chacun des segments de la bipartition.

a) Initialisation des gains généralisés

Le calcul des gains généralisés étant fonction des liens des connexions sur chacun des segments de la bipartition, ceux-ci doivent être à jour avant chaque calcul de gain. On fixe l'ordre maximum des gains à $k \leq n_{\max}$.

L'algorithme est le suivant :

```

/* Initialisation des liens de chaque connexion */
Initialiser à zéro le lien de chaque connexion sur les segments.
Pour chaque connexion n du circuit faire
    {Pour chaque module m de la connexion faire
        {Si (m ∈ A) alors
             $L_A(n) \leftarrow L_A(n) + 1$ 
        Sinon
             $L_B(n) \leftarrow L_B(n) + 1$ 
        }
    }
}

/* Initialisation des gains */
Initialiser à zéro le vecteur gain de chaque module
Pour chaque module m du circuit faire
    {S ← segment source de m
    D ← segment destination de m
    Pour chaque connexion n de m faire
        {Si ( $L_S(n) \leq k$  et  $L_D(n) > 0$ ) alors
            incrémenter le gain d'ordre  $L_S(n)$  de m
        Si ( $L_D(n) < k$ ) alors
            décrémenter le gain d'ordre ( $L_D(n) + 1$ ) de m
        }
    Placer m en tête de liste à la position ( $g_1(m), g_2(m), \dots, g_k(m)$ ) du
    tableau associé à S.
    }

```

b) Remise à jour des gains généralisés

On dit qu'une connexion est critique d'ordre i si elle contribue positivement ou négativement au gain d'ordre i d'un de ses modules.

Étant donnée une bipartition (A,B) et une connexion n , étudions les différentes conditions qui font qu'il existe un ordre pour lequel n est critique :

L'ordre maximum des gains est toujours fixé à $k \leq n_{\max}$.

Contribution au gain des modules de A :

- Si $L_A(n) \leq k$ et $L_B(n) > 0$ alors n contribue positivement au gain d'ordre $L_A(n)$ des modules de A qui lui sont connectés.
- Si $L_B(n) < k$ et $L_A(n) > 0$ alors n contribue négativement au gain d'ordre $(L_B(n) + 1)$ des modules de A qui lui sont connectés.

Contribution au gain des modules de B :

- Si $L_B(n) \leq k$ et $L_A(n) > 0$ alors n contribue positivement au gain d'ordre $L_B(n)$ des modules de B qui lui sont connectés.

- Si $L_A(n) < k$ et $L_B(n) > 0$ alors n contribue négativement au gain d'ordre $(L_A(n) + 1)$ des modules de B qui lui sont connectés

Pour les mêmes raisons que dans le paragraphe I.2.3, on ne considère, pour le réajustement des gains, que les voisins du module déplacé dont une au moins des connexions avec ce module est critique d'ordre $i \leq k$ avant ou après le déplacement.

L'algorithme pour le déplacement du module m de S vers D et la remise à jour des gains est :

S ← segment source de m

D ← segment destination de m

Déplacer m et le figer dans le segment D.

Pour chaque connexion n de m faire

{/* Attention, les valeurs de $L_S(n)$ et $L_D(n)$ sont celles avant le déplacement. On a donc $L_S(n) \geq 1$. */

/* Mise à jour des gains des modules de S */

Si $(L_S(n) \leq k$ et $L_D(n) > 0)$ alors

/* On annule l'ancienne contribution positive de n */

Décrémenter le gain d'ordre $L_S(n)$ des modules libres de n situés dans S.

Si $(L_S(n) - 1 \leq k$ et $L_D(n) > 0)$ alors

/* On prend en compte la nouvelle contribution positive */

Incrémenter le gain d'ordre $(L_S(n) - 1)$ des modules libres de n situés dans S.

Si $(L_D(n) < k)$ alors

/* On annule l'ancienne contribution négative de n */

Incrémenter le gain d'ordre $(L_D(n) + 1)$ des modules libres de n situés dans S.

/* Il n'y a pas de nouvelle contribution négative de n car la connexion devient figée par rapport à D */

/* Mise à jour des gains des modules de D */

Si $(L_D(n) \leq k)$ alors

/* On annule l'ancienne contribution positive de n */

Décrémenter le gain d'ordre $L_D(n)$ des modules libres de n situés dans D .

/* Il n'y a pas de nouvelle contribution positive de n car la connexion devient figée par rapport à D */

Si ($L_S(n) < k$ et $L_D(n) > 0$) alors

/* On annule l'ancienne contribution négative de n */

Incrémenter le gain d'ordre ($L_S(n) + 1$) des modules libres de n situés dans D .

Si ($L_S(n) - 1 < k$ et $L_D(n) > 0$) alors

/* On prend en compte la nouvelle contribution négative */

Décrémenter le gain d'ordre $L_S(n)$ des modules libres de n situés dans D .

/* Mise à jour des liens de n */

$L_S(n) \leftarrow L_S(n) - 1$

$L_D(n) \leftarrow \infty$

}

Munis du concept de gain généralisé et des algorithmes d'initialisation et de remise à jour des gains, nous pouvons exécuter l'algorithme donné au paragraphe I.2.4 .

Théorème : [KRISHNAMURTHY 84]

En utilisant des gains généralisés, la complexité de l'algorithme d'amélioration itérative d'une bipartition donnée au paragraphe I.2.4 est $O(k.N_c)$ au lieu de $O(N_c)$ pour les gains d'ordre 1, avec :

k = Ordre maximal des gains généralisés.

N_c = Nombre total de connecteurs sur l'ensemble des modules

$$= \sum_{v_i \in V} |\mathfrak{E}(v_i)|$$

où $\mathfrak{E}(v_i)$ désigne l'ensemble des connexions attachées au module v_i .

I.3 Construction d'une bipartition initiale

Comme nous l'avons dit précédemment, le problème de bipartitionnement est généralement divisé en deux sous-problèmes : la construction d'une bipartition initiale respectant la contrainte d'équilibrage, puis l'amélioration de cette bipartition par une méthode itérative telle que celles que nous avons étudiées

dans la partie I.2 .

Nous allons étudier dans cette partie les différentes stratégies pour obtenir une bipartition initiale.

Une stratégie aléatoire peut être utilisée pour répartir les modules dans deux sous-ensembles de tailles équivalentes, mais il est évident que la bipartition initiale conditionne beaucoup la coupe de la bipartition finale.

Bien qu'il soit difficile de discerner parmi les bipartitions initiales celles qui conduisent à des coupes faibles par l'algorithme d'amélioration itérative, on essaye dans la mesure du possible d'obtenir une bipartition initiale dont la coupe est déjà faible ou dont la répartition des modules favorise les améliorations durant la phase itérative.

I.3.1 Bipartitionnement du graphe de connectivité sans contrainte de taille

Si l'on écarte la contrainte d'équilibrage concernant la taille de chacun des segments, alors on peut considérer le problème de bipartitionnement suivant :

Soient $H = (V, \mathfrak{E})$ l'hypergraphe de connectivité du circuit, $w: \mathfrak{E} \rightarrow \mathbb{R}^+$ une pondération positive des arêtes, et s et p deux sommets particuliers de H ($s, p \in V$).

On cherche $A \subset V$ et $B \subset V$ tels que :

- 1) $s \in A ; p \in B$
- 2) $A \cup B = V$
- 3) $A \cap B = \emptyset$

et qui minimisent $W(C)$, où :

$$C = \{E_k \in \mathfrak{E} / \exists v_i, v_j \in E_k \text{ tq } v_i \in A \text{ et } v_j \in B\}$$

et
$$W(C) = \sum_{E_i \in C} w(E_i)$$

Si l'on fait l'hypothèse qu'en pratique il est fréquent d'avoir des sommets figés dans chacun des segments, alors ce problème correspond à notre problème de bipartitionnement, privé de la contrainte d'équilibrage. De plus, il peut être résolu de façon optimale en se ramenant au problème de la coupe minimale dans un réseau [LAWLER 73].

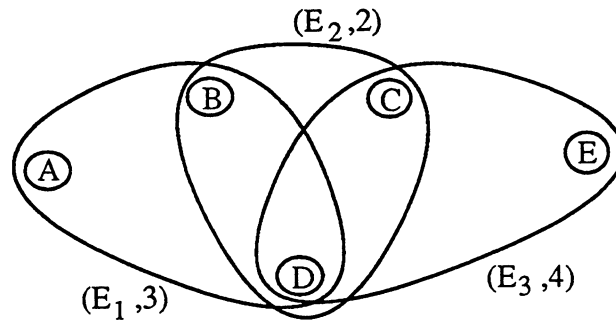
Il faut pour cela construire le réseau $R=(X,U,s,p)$ de la façon suivante :

A chaque sommet x_i de H , correspond un sommet x_i de R et, à chaque arête E_i de H , correspondent deux sommets E'_i et E''_i de R , reliés par un arc (E'_i, E''_i)

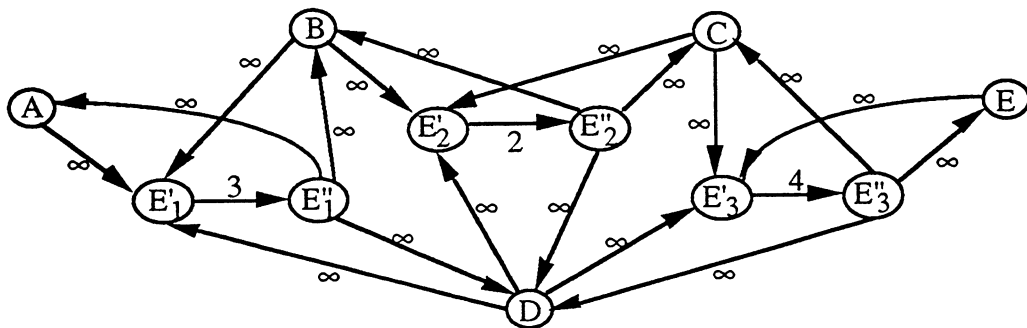
dont la capacité est égale au poids de l'arête initiale (une arête peut en effet représenter plusieurs connexions, dans ce cas elle est pondérée par le nombre de connexions).

Si l'arête E_i est incidente au sommet x_i dans H , alors R contient les deux arcs (x_i, E'_i) et (E''_i, x_i) de capacité infinie.

La figure suivante illustre le cas d'un circuit composé de 5 modules et 3 connexions.



Hypergraphe du circuit.



Réseau associé à l'hypergraphe.

Un arc de retour (p,s) , de capacité infinie doit être rajouté dans R lorsque les sommets s et p sont déterminés.

L'algorithme de Ford et Fulkerson [SAKAROVITCH 84] permet alors d'obtenir le flot maximum dans R , ainsi que la coupe minimale associée. Cette solution est trouvée en un temps polynomial d'ordre $O(M.N^2)$ où N est le nombre de noeuds du réseau et M le nombre d'arcs.

- Un flot dans le réseau est une application $f: U \rightarrow \mathbb{R}$ qui doit vérifier les propriétés suivantes :

$0 \leq \text{flot sur l'arc } u \leq \text{capacité de l'arc } u$ ceci sur tous les arcs.

$\sum_{\text{arcs entrants}} \text{flot sur l'arc} = \sum_{\text{arcs sortants}} \text{flot sur l'arc}$ ceci sur chaque noeud.

- Une coupe séparant s de p est un ensemble C d'arcs du réseau (ie $C \subset U$), tel qu'il existe un sous-ensemble Y de noeuds du réseau ($Y \subset X$) avec $s \in Y$ et $p \notin Y$, qui vérifie :

$$C = \{u \in U / I(u) \in Y \text{ et } T(u) \notin Y\}$$

où $I(u)$ et $T(u)$ désignent respectivement le noeud de départ et le noeud d'arrivée de l'arc u .

La coupe minimale est la coupe dont la somme des capacités des arcs est minimale. Elle conduit à une bipartition de l'ensemble des sommets de l'hypergraphe, séparant les sommets s et p , et telle que la somme des poids des arêtes entre les deux segments soit minimale.

Si le problème est de trouver une coupe minimale entre deux sous-ensembles disjoints de sommets X_1 et X_2 , le réseau précédent doit être adapté. On rajoute les deux nouveaux sommets s et p : le sommet s est connecté à chaque sommet de X_1 par un arc de capacité infinie et chaque sommet de X_2 est connecté à p par un arc de capacité infinie.

Nous venons de voir une méthode optimale pour le bipartitionnement des sommets d'un hypergraphe en deux sous-ensembles tels que le nombre de connexions entre ces deux sous-ensembles soit minimum.

Cette technique de bipartitionnement n'est pas utilisée en pratique car elle conduit rarement à une taille équivalente de chacun des sous-ensembles, ce qui entraîne le non respect de la contrainte d'équilibrage. De plus, bien que la complexité de l'algorithme de recherche d'une coupe minimale soit acceptable, l'ordre de grandeur du réseau augmente très rapidement avec la complexité du circuit.

I.3.2 Utilisation de l'algorithme d'amélioration itérative

Pour obtenir une bipartition équilibrée, on peut utiliser l'algorithme d'amélioration itérative d'une partition, décrit dans la partie I.2, en partant d'une bipartition initiale non équilibrée dont tous les modules sont dans le même segment, l'autre étant vide.

Le but des itérations est alors d'établir la contrainte d'équilibrage tout en

conservant la coupe la plus petite possible.

Supposons que l'on parte de la bipartition (A,B) dont A contient tous les modules et B est vide. La contrainte d'équilibrage est :

$$|S(A) - \frac{1}{2} S_t| \leq T \quad \text{avec } T \in \mathbb{R}, \text{ tolérance de la bipartition.}$$

$$\text{or on a: } S(A) = \sum_{v_i \in A} s(v_i) = \sum_{i=1}^n s(v_i) = S_t$$

Le déplacement d'un module n'est donc autorisé que s'il diminue la valeur $|S(A) - \frac{1}{2} S_t|$ également appelée *valeur d'équilibre*.

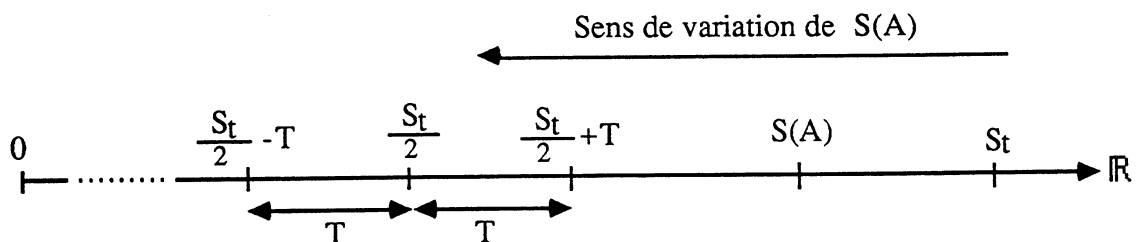
Durant tout le processus, seuls les modules du segment A sont libres, ainsi les déplacements se font toujours de A vers B et la valeur $(S(A) - \frac{1}{2} S_t)$ ne fait que décroître.

A chaque itération, le module de base est le module de A de plus fort gain qui diminue la valeur d'équilibre, ceci permet de maintenir la coupe la plus petite possible. Lorsque le déplacement des modules de A ne permet plus de diminuer la valeur d'équilibre, le processus s'arrête.

Dans l'algorithme décrit dans le paragraphe I.2.4, un second critère d'arrêt est évalué par rapport au nombre d'équipotentiels figés de la bipartition. Ce critère n'est pas gênant ici car il ne peut pas exister d'équipotentiels figés, du fait que les modules de A sont toujours des modules libres.

- Si la valeur de la tolérance est $T \geq \frac{1}{2} \max_{i=1..n} (s(v_i))$ alors tant que la contrainte d'équilibrage n'est pas établie, le module de base est toujours le module de gain maximum, car aucun déplacement de module ne risque de faire chuter la valeur de $S(A)$ en dessous de $(\frac{1}{2} S_t - T)$. Ainsi le choix du module de base se fait sans recherche.

Ceci est illustré sur le schéma suivant :



Quand la contrainte d'équilibrage est établie, il peut être possible de diminuer encore sa valeur mais ceci n'est pas l'objectif visé, d'autant plus que le choix du module de base peut nécessiter une recherche. Pour cette raison on choisit

d'arrêter le processus dès que la contrainte est vérifiée.

La bipartition obtenue peut servir de bipartition initiale pour le processus d'optimisation de la coupe.

- Si la valeur de la tolérance est $T < \frac{1}{2} \max_{i=1..n} (s(v_i))$ alors le module de base est celui de plus fort gain qui permet d'avoir $S(A) \geq \frac{1}{2} S_t - T$ après son déplacement.

Il est possible que le processus arrive dans la situation :

$$S(A) > \frac{1}{2} S_t + T \quad \text{et} \quad \forall v_i \in A: S(A) - s(v_i) < \frac{1}{2} S_t - T$$

Dans ce cas le processus s'arrête sans que la contrainte d'équilibrage soit vérifiée. La seule solution est alors d'augmenter la valeur de la tolérance.

Deux possibilités se présentent :

- Choisir T de façon à ce que la bipartition courante satisfasse la contrainte ($T = S(A) - \frac{1}{2} S_t$).

- Augmenter T jusqu'à pouvoir déplacer le plus petit module v_i de A sans violer la contrainte ($T = \frac{1}{2} S_t - S(A) + s(v_i)$)

On choisit comme solution celle qui conduit à la plus faible augmentation de la tolérance.

I.3.3 Approche simple basée sur un tri des connexions

Partant d'une bipartition dont les deux ensembles sont vides, cette approche consiste à chaque pas, à affecter tous les modules d'une connexion donnée au même ensemble, afin que la connexion ne fasse pas partie de la coupe.

Les connexions doivent donc être triées dans l'ordre préférentiel de leur non apparition dans la coupe. Si la longueur de certaines connexions doit être minimisée prioritairement (raisons électriques ...), celles-ci doivent être placées en premier dans le classement.

L'approche classique consiste à trier les connexions en fonction de leur taille, soit dans le sens croissant soit dans le sens décroissant.

L'algorithme de construction d'une bipartition initiale est le suivant :

- Construire la liste triée des connexions.
- Initialiser A et B à l'ensemble vide.
- Tant que l'un des deux ensembles n'atteint pas la taille souhaitée faire
 - {
 - Sélectionner la première connexion dans la liste.
 - Assigner les modules non encore assignés de cette connexion, à celui des ensembles A et B qui est le moins rempli, dans la limite de la taille souhaitée pour cet ensemble.
 - Supprimer la connexion de la liste.
 - }
- Assigner le reste des modules à l'autre ensemble.

Il est possible de modifier l'algorithme de façon à passer directement à la connexion suivante lorsque la connexion courante connecte des modules déjà assignés dans A et B, ceci dans le but de minimiser la coupe de la bipartition. Néanmoins, si l'on désire réellement qu'une connexion n'apparaisse pas dans la coupe, il est préférable que dès la bipartition initiale ses modules soient essentiellement disposés dans le même ensemble. Ainsi l'algorithme d'amélioration itérative qui suit pourra plus facilement l'éliminer de la coupe.

Dans le cas où les connexions sont triées dans l'ordre croissant de leur taille, les premières connexions traitées seront celles connectant peu de modules. L'algorithme pourra donc traiter beaucoup de connexions avant que la taille de l'un des deux ensembles ne soit atteinte. De ce fait le nombre de connexions dans la coupe sera "relativement" minimisé, mais ces connexions seront essentiellement des grosses connexions. Il sera alors difficile de les éliminer de la coupe par l'algorithme de déplacements itératifs de modules.

En revanche, dans le cas où les premières connexions traitées sont celles connectant beaucoup de modules, les ensembles arrivent rapidement à saturation, et l'algorithme s'exécute plus vite. La taille de la coupe est plus élevée qu'avec la liste croissante mais elle peut être plus facilement minimisée lors de l'algorithme itératif, car beaucoup de ces connexions connectent peu de modules. Ce choix prime généralement sur le précédent.

I.3.4 Contraction du graphe de connectivité par regroupement de sommets

Cette technique ne permet pas d'obtenir une bipartition initiale, mais conduit à

un graphe plus simple, qu'il est ensuite plus facile de bipartitionner.

Le principe consiste à regrouper les modules fortement connectés ou devant appartenir à un même segment, en un seul module qui hérite de toutes leurs connexions avec le reste du circuit.

Les critères de regroupement sont divers : Goldberg et Burstein [GOLDBERG&BURSTEIN 83] ont observé que l'algorithme d'amélioration itérative est plus performant si le degré moyen des sommets du graphe de connectivité est supérieur à six. Ils effectuent donc des regroupements de modules jusqu'à satisfaire cette condition.

Dans [NG,OLDFIELD&PITCHUMANI 87] les auteurs se basent sur les propriétés statistiques des réseaux, observées par Landman et Russo [LANDMAN&RUSSO 71]. Ils en déduisent un algorithme de regroupement de modules par paires, basé sur la loi de probabilité que suit le nombre de connexions d'un ensemble de modules vers les autres modules. Cette loi est une loi de Rent, dont le calcul du coefficient de Rent pour chaque paire de candidats au regroupement, permet de sélectionner la meilleure.

Dans tous les cas on doit ensuite bipartitionner le graphe contracté, par une des méthodes précédentes. La bipartition obtenue constitue alors une bipartition initiale du graphe de départ, une fois celui-ci reconstitué.

II. STRATÉGIE DU PLACEMENT

Le principe du placement par bipartitionnements successifs consiste à bipartitionner récursivement l'ensemble des modules du circuit jusqu'à ce que chaque module soit isolé dans un segment.

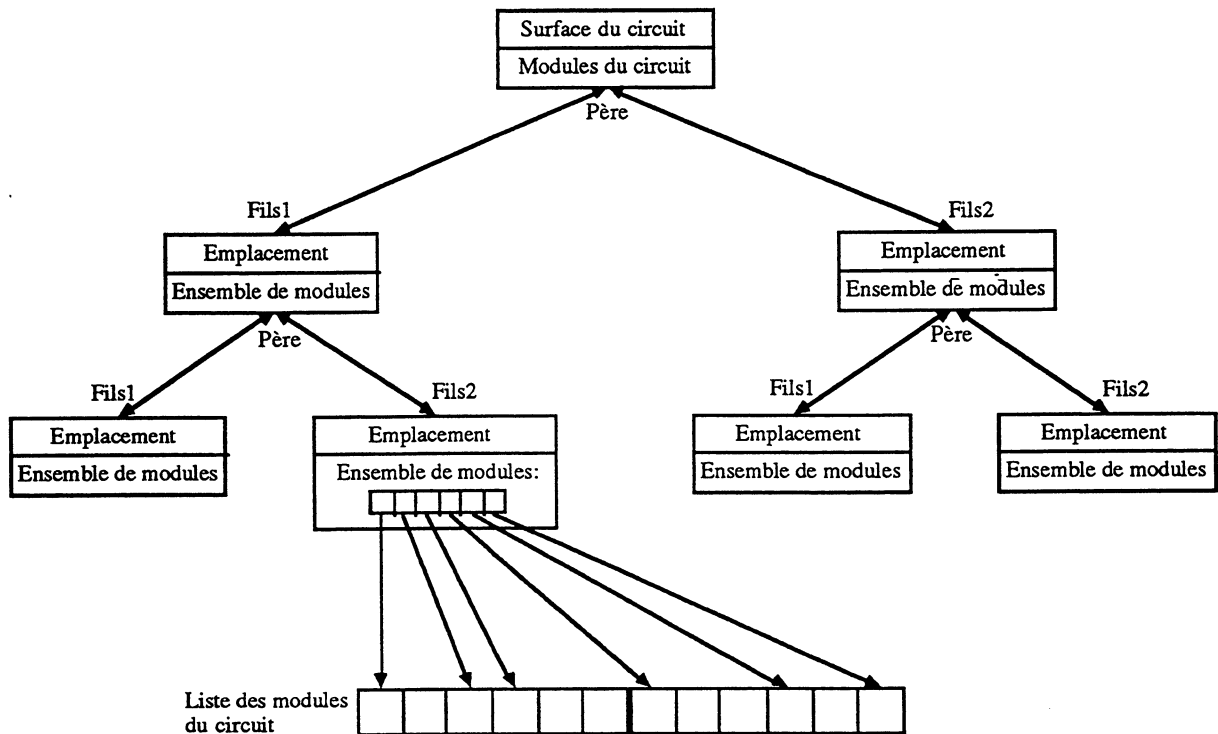
Parallèlement, à chaque segment issu d'une bipartition est associée une zone du circuit, issue elle-même du partage en deux de la zone associée au segment père de la bipartition. Lors de l'initialisation, la zone du circuit associée à l'ensemble de tous les modules est le circuit lui-même. Si la surface du circuit n'est pas donnée en tant que contrainte, elle doit être estimée à partir de la surface des modules.

Ainsi, lorsque chaque module se trouve isolé dans un segment, il lui est associée une zone qui correspond à son emplacement sur le circuit. Il ne reste alors plus qu'à déterminer l'orientation de chaque bloc.

L'élaboration du placement par des bipartitionnements successifs de l'ensemble des modules correspond à la construction d'un arbre binaire de placement en partant de la racine. La racine correspond ici à l'ensemble de tous les modules du circuit et, à chaque étape, l'ensemble des modules associé à une feuille de l'arbre est bipartitionné pour produire deux nouvelles feuilles.

Pour cette raison, la structure de données utilisée pour stocker le placement courant à chaque étape représente l'arbre binaire à cette étape. Chacun des noeuds de l'arbre est constitué d'un tableau de pointeurs sur les modules du noeud et de l'emplacement de cet ensemble sur le circuit, également appelé *box*.

Le schéma suivant donne une idée de cette structure :



Représentation interne du placement.

II.1 Divisions successives du circuit

Il peut arriver, lors de la conception d'un circuit, que le boîtier d'accueil de celui-ci soit déterminé à l'avance. Dans ce cas l'enveloppe du circuit est fixée par le boîtier et peut être considérée comme le *box* associé au noeud racine. Dans le cas où l'on ne dispose pas de l'enveloppe du circuit avant le placement, une estimation de celle-ci doit être effectuée.

II.1.1 Estimation de l'enveloppe du circuit

Pour déterminer une estimation de l'enveloppe du circuit, on doit d'abord effectuer une estimation de la surface qu'il va occuper. Cette dernière est calculée en fonction de la surface totale des modules et d'une estimation de l'aire de routage nécessaire aux connexions du circuit:

On désigne par S_t la surface totale des modules du circuit et par A_r l'aire

estimée pour le routage des connexions.

$$S_t = \sum_{i=1}^n s(v_i)$$

où $s(v_i)$ désigne la surface du module v_i .

$$A_r = A_{r\text{-local}} + A_{r\text{-global}}$$

$$A_{r\text{-local}} = (\text{Largeur d'une piste}) \cdot \frac{1}{2} \sum_{i=1}^n (|\mathfrak{E}(v_i)| \cdot \text{périmètre de } v_i)$$

$$A_{r\text{-global}} = (\text{Largeur d'une piste}) \cdot \frac{1}{2} \cdot n \cdot \sqrt{S_t}$$

où n désigne le nombre total de modules et $\mathfrak{E}(v_i)$ désigne l'ensemble des connexions attachées au module v_i .

On estime alors la surface S_c du circuit par:

$$S_c = 1,3 \cdot S_t + A_r$$

Par défaut, l'enveloppe du circuit est prise carrée. Si l'une des dimensions d'un module est supérieure à la taille de l'enveloppe, on prend pour celle-ci un rectangle dont la plus grande dimension est la plus grande dimension existante pour les modules.

II.1.2 Division des boxes

A chaque étape du placement, un noeud de l'arbre contenant plus d'un module est traité : l'ensemble de ses modules est bipartitionné de façon à minimiser la coupe, puis le box qui lui est associé est divisé en deux nouveaux boxes que l'on affectera à chacun des segments issus de la bipartition de l'ensemble des modules.

Le box est divisé en deux parties au moyen d'une ligne de coupe qui peut être soit horizontale, soit verticale.

La direction de cette ligne de coupe peut être fixée alternativement, horizontale puis verticale, de façon à simplifier l'algorithme. Néanmoins il est préférable de déterminer sa direction de façon à partager le box en deux sous-portions les plus carrées possibles, dans la mesure où cela n'empêche pas un des modules du noeud d'être contenu dans les boxes résultants.

Une fois la direction de la ligne de coupe établie et le bipartitionnement des modules effectué, chaque segment est affecté d'un côté de la ligne.

Selon le côté d'affectation d'un segment par rapport à la ligne de coupe, la longueur de connexions entre les modules de ce segment et les modules

extérieurs au noeud traité est plus ou moins importante.

Nous verrons dans le paragraphe II.2 comment il est possible de tenir compte des connexions vers l'extérieur du noeud afin de déterminer la position des modules par rapport à la ligne de coupe.

La position de la ligne de coupe est fixée de façon à ce que le rapport des surfaces des boxes soit le même que le rapport des surfaces des segments qui leur sont affectés.

Si on note S_A et S_B les surfaces des boxes associés aux segments A et B alors la position de la ligne de coupe est telle que :

$$\frac{S_A}{S_B} = \frac{S(A)}{S(B)}$$

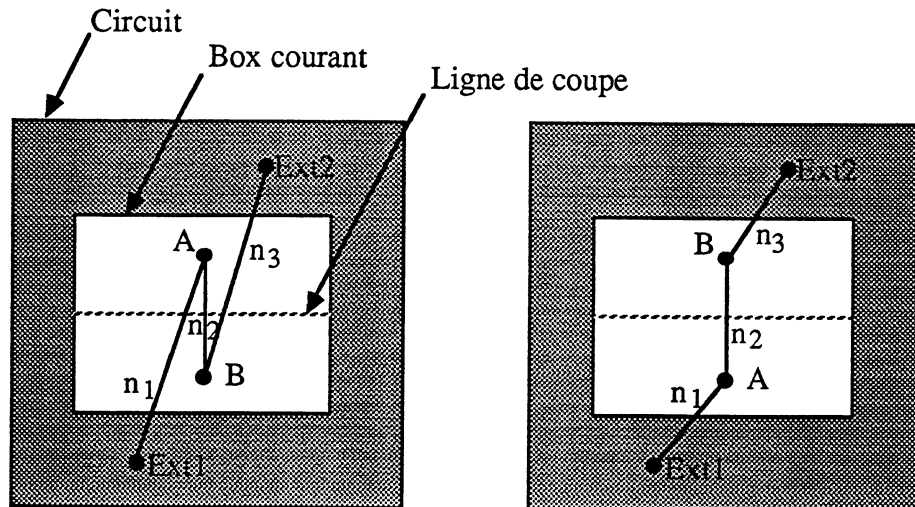
$$\text{avec } S(A) = \sum_{v_i \in A} s(v_i) \quad \text{et} \quad S(B) = \sum_{v_i \in B} s(v_i)$$

II.2 Bipartitionnement des modules d'un noeud

Lors du traitement d'un noeud de l'arbre, l'ensemble de modules qui lui est associé doit être bipartitionné s'il contient plus de deux éléments. Ce bipartitionnement doit être réalisé de façon à minimiser le nombre de connexions qui traverseront la ligne de coupe du box; pour cela, on utilise l'algorithme décrit dans la partie I.

Il faut donc considérer comme hypergraphe de connectivité lors du bipartitionnement, le sous-hypergraphe constitué de l'ensemble des modules du noeud et des connexions entre ces modules.

Cependant, la connectique entre les modules du noeud et ceux qui lui sont extérieurs n'est pas représentée dans ce sous-hypergraphe. Cet aspect n'est pourtant pas négligeable car il est évident que ces connexions devront traverser la ligne de coupe dans le cas où les modules sont disposés de part et d'autre de celle-ci. C'est par exemple le cas si la ligne de coupe est horizontale et qu'une connexion relie un module du noeud situé au Nord de la ligne de coupe avec un module extérieur au noeud situé au Sud du box. La figure page suivante illustre une des pires situations qui puissent arriver.



Les deux dispositions des modules A et B sont identiques du point de vue de la connectique locale au noeud, pourtant celle de droite est bien meilleure au niveau global.

Il faut donc tenir compte des connexions entre les modules du noeud et les modules qui lui sont extérieurs, lors du bipartitionnement.

D.P.Lapotin [LAPOTIN 85] a proposé une méthode pour tenir compte de ces connexions tout en conservant le même algorithme de bipartitionnement. Cette méthode a ensuite été reprise par A.T.Sherman [SHERMAN 89], en voici le principe:

Supposons que l'on divise le box courant à l'aide d'une ligne de coupe horizontale. On recherche alors tous les modules extérieurs au noeud courant qui sont connectés à un module de celui-ci. Ceux qui sont situés au Nord du noeud sont introduits dans le segment haut du box en les figeant dans ce segment et en leur affectant une surface nulle. Ceux qui sont situés au Sud sont introduits de la même manière dans le segment bas du box. De tels modules sont alors appelés modules fictifs car ils n'interviennent dans le processus de bipartitionnement que par leur connectique et n'influencent pas sur le critère d'équilibrage.

A l'aide de cet artifice, l'hypergraphe de connectivité considéré lors du bipartitionnement, est le sous-hypergraphe, dont l'ensemble des sommets est l'union des modules du noeud et des modules fictifs, et dont l'ensemble des arêtes est constitué de toutes les connexions existant entre ces modules.

Ainsi la connectique vers l'extérieur du noeud est prise en compte et les modules fictifs étant figés, ils tirent vers eux les modules du noeud courant qui

leur sont connectés.

On peut imaginer une modification de ce procédé, afin de diminuer le nombre de noeuds fictifs introduits :

Pour cela on considère chaque connexion ayant des modules à l'intérieur et à l'extérieur du noeud.

Si tous les modules extérieurs de la connexion sont situés au Nord du box, alors on en ajoute un de façon fictive (en le figeant et sans tenir compte de sa surface) dans le segment Nord. Il n'est pas nécessaire d'introduire tous les modules extérieurs de façon fictive car, de toute façon, le coût d'une connexion dans la coupe est toujours de 1 quel que soit le nombre de modules qu'elle connecte de part et d'autre.

On procède de la même façon si tous les modules extérieurs sont situés au Sud. Dans le cas où les modules extérieurs sont disposés à la fois au Nord et au Sud du box alors, quoiqu'il arrive, la connexion traversera la ligne de coupe. On introduit dans les segments Nord et Sud, un des modules extérieurs de la connexion situé respectivement au Nord et au Sud du box. Cela permet de figer directement la connexion dans la coupe, et évite de la prendre en compte durant le processus d'optimisation. Sans ces modules fictifs, le processus pourrait être amené à supprimer "localement" cette connexion de la coupe, au détriment d'une autre connexion éventuellement plus avantageuse.

Il faut noter que l'introduction des modules fictifs nécessite une prise en compte des blocs figés lors de l'initialisation des gains.

La position d'un module extérieur, par rapport au box courant, peut être établie en utilisant la direction de la ligne de coupe qui l'a séparé de l'ensemble des modules du noeud courant. Mais si cette ligne de coupe est verticale, le module est considéré uniquement comme étant à l'Ouest ou à l'Est du box courant, même s'il est également situé au Nord ou au Sud de celui-ci. Ceci ne permet donc pas d'en tenir compte dans les modules fictifs d'une bipartition horizontale et peut amener à ignorer certaines connexions vers l'extérieur.

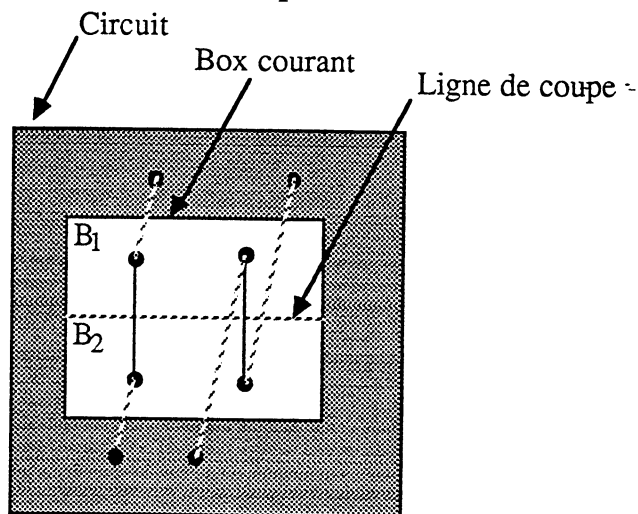
Pour cette raison il est préférable de comparer les coordonnées pour déterminer les positions relatives.

Néanmoins, il est important de noter que la position d'un module extérieur, par rapport au box courant, ne peut être déterminée qu'en utilisant les coordonnées du box qui le contient. Or, bien souvent, le module extérieur n'est pas isolé dans son box, ce qui induit une certaine imprécision à ce niveau.

Une approche différente a été proposée dans [BHANDARI,HIRSCH&SIEWIOREK 88] pour optimiser la connectique au niveau global. Il s'agit d'effectuer le bipartitionnement des modules de chaque noeud en ne considérant que les connexions à l'intérieur du noeud, donc sans utiliser de modules fictifs. La connectique globale est ensuite optimisée en effectuant des échanges de boxes pour les segments issus d'un même bipartitionnement.

Cette technique ne permet pas de remédier à toutes les mauvaises configurations qui peuvent apparaître lors du bipartitionnement effectué sans la prise en compte des connexions vers l'extérieur du noeud.

La figure suivante en donne un exemple :



L'échange des boxes entre les 2 segments ne permet pas de supprimer les connexions externes de la coupe.

De plus, pour que l'échange des boxes soit directement possible entre deux noeuds frères, il faudrait que leurs dimensions soient égales. Ceci n'est généralement pas le cas et implique un réajustement des boxes pour permettre l'échange.

Utilisée seule, cette technique n'est pas aussi performante que l'ajout de noeuds fictifs lors du bipartitionnement. Néanmoins, malgré le temps important qu'elle nécessite, elle peut être utilisée en complément de l'emploi de noeuds fictifs.

Lorsque le noeud à traiter ne contient plus que deux modules, la bipartition choisie est systématiquement celle où chaque segment contient un module car c'est la seule qui permet de vérifier les contraintes du problème de bipartitionnement. L'affectation de chaque module à un segment se fait en tenant compte des connexions des modules vers l'extérieur du noeud.

Supposons que la ligne de coupe soit horizontale, on évalue pour chaque module le score suivant :

$$\text{score}(m) = NC_N - NC_S$$

avec

NC_N = Nombre de connexions de m dont tous les modules extérieurs sont situés au Nord du box.

NC_S = Nombre de connexions de m dont tous les modules extérieurs sont situés au Sud du box.

Le module affecté dans le segment Nord est alors celui de score le plus élevé. On procède de façon similaire quand la ligne de coupe est verticale.

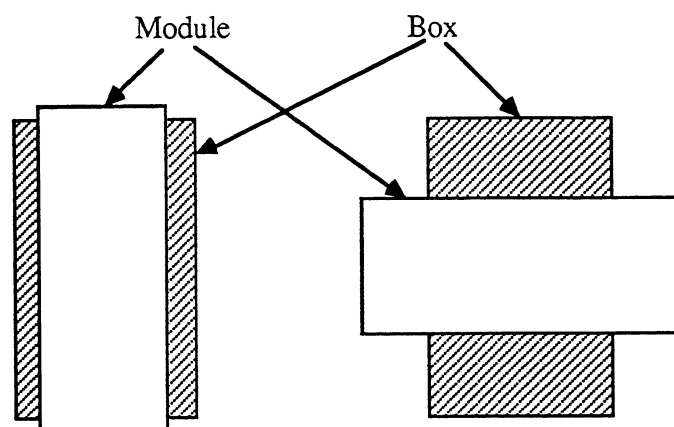
II.3 Orientation des blocs et ajustement des positions

Lorsque chacune des feuilles de l'arbre de placement ne contient plus qu'un seul module, le placement relatif des modules les uns par rapport aux autres est entièrement déterminé et le processus de bipartitionnement récursif des feuilles s'arrête.

Chaque feuille est alors composée d'un module et de son emplacement sur le circuit.

A ce stade du placement, l'orientation des modules qui n'a pas été prise en compte jusque-là peut être déterminée. Pour chaque module elle est choisie en fonction de deux critères :

- Le module doit déborder le moins possible de son emplacement. En effet les modules et les boxes étant rarement de forme carrée, certaines orientations permettent un meilleur ajustement. Il est préférable de déformer le moins possible le box car cela se répercute sur tout le placement et peut entraîner une augmentation de la surface totale du circuit.



Du point de vue du débordement, l'orientation d'un module se ramène à deux cas. Ici l'orientation verticale est meilleure que l'orientation horizontale.

- La longueur totale des connexions, du module aux autres modules connectés,

doit être minimisée.

On procède à un traitement séquentiel des modules en faisant l'hypothèse que l'orientation choisie pour un module dépend faiblement de l'orientation des autres modules. Pour cela la longueur de chaque connexion du module est évaluée en fixant ses points d'attachement sur les autres modules au centre de leur box.

Pour chacune des huit orientations d'un module (4 rotations x 2 symétries) on évalue un coût qui tient compte des deux critères précédents. On choisit alors l'orientation de meilleur coût.

Le coût d'une orientation est calculé de la façon suivante :

Le module doit être centré au milieu de son box.

$$\text{Coût (orientation)} = \alpha \cdot \text{surface de débordement du module par rapport au box} + \beta \cdot \text{longueur totale des connexions du module}$$

où α et β sont des constantes réelles.

La longueur d'une connexion est estimée par la somme des distances euclidiennes entre le point d'attachement de la connexion sur le bloc et chacun des centres des modules connectés. On peut pondérer les termes de la somme, de façon à favoriser la minimisation des distances avec les connecteurs proches. En effet, l'orientation du module ne modifie le trajet des connexions que dans son voisinage immédiat. Dans ce cas, un poids plus fort sera attribué aux connecteurs du voisinage.

L'orientation choisie est celle dont le coût est le plus faible.

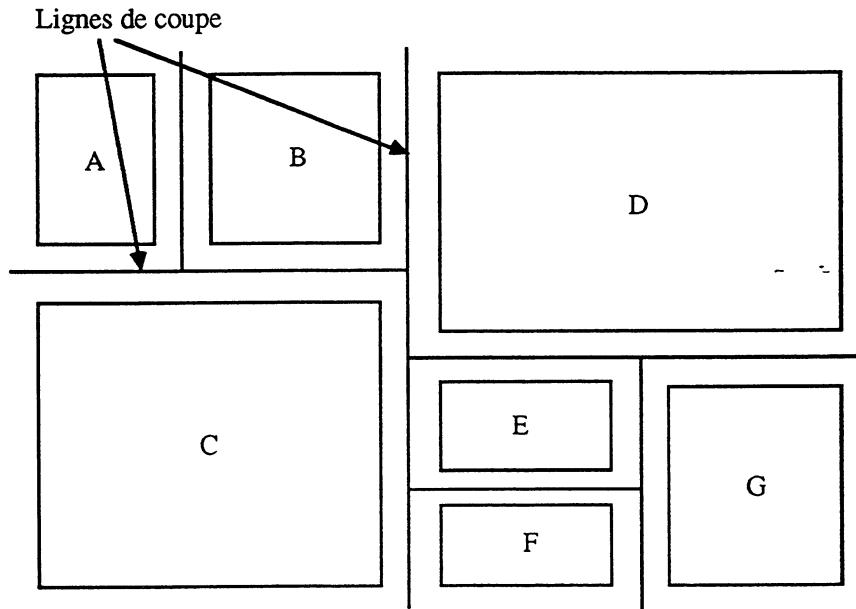
Une fois l'orientation des modules accomplie, la position de chacun d'entre eux doit être ajustée afin d'obtenir le placement final. Ceci s'effectue par un réajustement des boxes, visant à supprimer les débordements éventuels et surtout à laisser entre les blocs l'écart nécessaire au routage des connexions.

Le réajustement des boxes doit se faire à tous les niveaux de l'arbre car la modification d'un noeud se propage jusqu'à la racine.

Dans un premier temps, le box de chaque feuille est ramené aux dimensions exactes du module que celle-ci contient. Ensuite, le réajustement des noeuds internes de l'arbre se fait de manière ascendante en tenant compte de la place nécessaire au routage des connexions. Chaque noeud, dont les deux fils ont été traités, est traité à son tour, et ainsi de suite jusqu'à la racine.

Le routage des connexions s'effectue dans les espaces libres laissés entre les modules et appelés "canaux". Chaque module doit être entièrement entouré de canaux, le contour du circuit constituant un canal pour les modules périphériques.

On s'aperçoit que dans un placement issu de coupes successives, chaque ligne de coupe correspond à l'emplacement d'un canal.

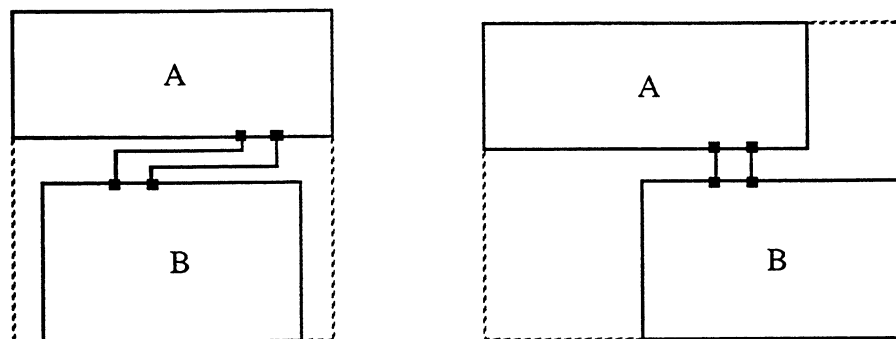


Placement issu de coupes successives

Ainsi, lors du réajustement d'un noeud interne de l'arbre, la ligne de coupe de ce noeud sera matérialisée par un canal entre ses deux fils.

A l'issue de la phase de réajustement, le placement obtenu doit satisfaire les contraintes de routage tout en minimisant la surface totale du circuit et la longueur totale de connexions.

Lors du réajustement d'un noeud interne, ces deux objectifs sont bien souvent antagonistes, comme le montre la figure suivante.

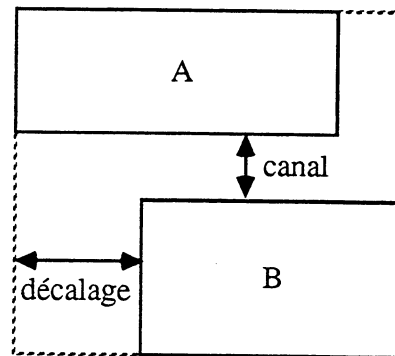


Minimisation de la surface et minimisation de la longueur de connexions

A.T.Sherman [SHERMAN 89] propose un compromis consistant à minimiser une fonction tenant compte de ces deux objectifs.

Pour cela on introduit deux quantités, le canal et le décalage, à l'aide desquelles le box du noeud est entièrement défini.

On suppose dans la suite que le noeud traité est l'assemblage de deux modules et qu'il admet une ligne de coupe horizontale. La figure suivante illustre le placement des deux modules du noeud à partir des valeurs du canal et du décalage.



Positions relatives des modules exprimées en fonction du canal et du décalage

Le canal représente la largeur nécessaire entre les deux modules pour le routage des connexions, tandis que le décalage désigne la distance entre les côtés gauches des modules, pour aligner le mieux possible les connecteurs. Le décalage peut être négatif ; un décalage positif signifie que B se déplace à l'Est de A.

La fonction objective à minimiser représente la surface du box. Celle-ci dépend des deux paramètres canal et décalage. L'intérêt de la modélisation consiste à exprimer la valeur du canal comme une fonction du décalage. Ainsi si x représente une valeur du décalage on a :

$$\text{Surface-box}(x) = \text{Largeur-box}(x) \cdot (y_A + \text{canal}(x) + y_B)$$

où y_A et y_B désignent les hauteurs des modules A et B, et $\text{Largeur-box}(x)$ est la largeur du box, qui dépend également du décalage.

La valeur du canal est définie par :

$$\text{canal}(x) = \text{Largeur-piste} \cdot \text{Nombre-de-pistes}(x)$$

avec

Largeur-piste = constante représentant la largeur occupée par une piste

$$\text{et } \text{Nombre-de-pistes}(x) = \frac{1,5 \cdot \text{Long-conn-hori}(x)}{\max(x_A, x_B)} + \text{Pistes-supp}$$

Long-conn-hori(x) est une estimation de la longueur horizontale de connexions pour réaliser toutes les connexions communes entre les segments A et B ; elle est fonction du décalage. Pistes-supp est une constante représentant le nombre de pistes supplémentaires pour réaliser les autres connexions du canal et (x_A, x_B) désignent les largeurs des modules A et B.

Nous devons maintenant estimer la longueur horizontale des connexions entre A et B. On ne prend en compte que les connexions bi-terminales dont les connecteurs sont situés le long du canal qui sépare A et B.

Soient a_1, \dots, a_n et b_1, \dots, b_n les positions horizontales des connecteurs des n connexions communes à A et B, respectivement sur A et B, et exprimées relativement au coin bas et gauche de chaque module. On suppose que chaque connexion ne touche qu'une fois chacun des modules.

Alors pour une valeur x du décalage, la longueur horizontale de la connexion i ($1 \leq i \leq n$) est donnée par :

$$l = |(b_i + x) - a_i| = |a_i - b_i - x| = |x_i^* - x|$$

où $x_i^* = a_i - b_i$ représente la valeur du décalage qui aligne les connecteurs de la connexion i .

On en déduit que :

$$\text{Long-conn-hori}(x) = \sum_{i=1}^n |x_i^* - x|$$

Cette fonction est une fonction convexe qui prend son minimum pour la valeur x_G^* , le barycentre des x_i^* .

Il en résulte que Surface-box(x) est une fonction convexe du décalage, dont on cherche le minimum dans l'intervalle $[x_{\min}, x_{\max}]$ où :

$$x_{\min} = \min\{0, \min\{x_1^*, \dots, x_n^*\}\}$$

$$x_{\max} = \max\{x_D, \max\{x_1^*, \dots, x_n^*\}\}$$

avec $x_D =$ valeur du décalage pour aligner les bords droits des modules.

Il existe un cas dans lequel Surface-box(x) prend son minimum en x_G^* :

Supposons que l'on ait $x_D > 0$, alors si $x_G^* \in [0, x_D]$, Long-conn-hori(x) prend son minimum dans l'intervalle où Largeur-box(x) est constant et de

valeur minimum. De ce fait, x_G^* est la valeur du décalage qui minimise $\text{Surface-box}(x)$. Il en est de même si $x_D < 0$ et $x_G^* \in [x_D, 0]$.

Dans le cas contraire, on peut trouver le minimum de $\text{Surface-box}(x)$ par une méthode itérative de type Newton.

L'utilisation de cette méthode pour ajuster les positions relatives de deux modules est intéressante lorsque les deux modules partagent beaucoup de connexions bi-terminales dont les connecteurs sont situés le long du canal. Si un connecteur est situé sur un côté opposé, la prise en compte de la connexion n'est pas exacte car sa minimisation ne correspond pas à l'alignement de ses terminaisons, étant donné qu'elle doit contourner un module.

De plus, si la connexion est adjacente à d'autres modules que les deux modules du noeud, alors elle doit sortir du canal et donc il peut s'avérer inutile d'aligner les connecteurs.

Pour cette raison, nous nous contentons d'aligner le mieux possible les connecteurs, en conservant la valeur de Largeur-box minimum. Ceci est réalisé en prenant une valeur du décalage comprise dans $[0, x_D]$ si $x_D > 0$, ou dans $[x_D, 0]$ sinon, qui soit la plus proche possible de x_G^* quand celui-ci n'appartient pas à l'intervalle.

Dans le cas d'un noeud dont les fils A et B sont des ensembles de plusieurs modules, A.T.Sherman généralise la méthode précédente. Pour cela il considère chaque connexion entre A et B comme deux intervalles, chacun contenant les projections des connecteurs d'un des segments sur l'horizontale. Les valeurs du décalage pour minimiser une connexion donnée sont alors celles pour lesquelles les deux intervalles se chevauchent.

Comme dans le cas précédent, cette généralisation n'est utilisable que pour les connexions internes au noeud (ie ne contenant pas de modules à l'extérieur) et dont les connecteurs sont situés le long du canal. Le rapport du nombre de ces connexions sur le nombre de connexions dans la coupe est bien inférieur au cas précédent, et ne justifie pas l'utilisation de la généralisation.

Pour cette raison, le traitement des noeuds dont les fils sont des ensembles de plusieurs modules, est réalisé en prenant un décalage nul et en calculant la valeur du canal avec:

$$\text{Nombre-de-pistes} = \text{Nbre de connexions dans la coupe} + \text{Pistes-supp}$$

Le même traitement peut être effectué sur les noeuds pères des modules, dans le but d'augmenter la rapidité du programme de placement.

A chaque réajustement d'un noeud, les coordonnées de tous les modules qu'il contient sont recalculées par rapport au repère de son box. Ainsi lorsqu'on arrive au traitement de la racine, les coordonnées des modules sont données dans le repère lié au circuit.

A l'issue du réajustement des positions, le placement des modules est terminé et le routage des connexions peut être effectué. Dans un premier temps la phase globale du routage détermine pour chaque connexion la séquence de canaux à emprunter, puis le tracé des connexions est effectué dans chaque canal lors de la phase de routage fin. La phase globale du routage doit tenir compte de la capacité de chaque canal en nombre de connexions traçables. Il se peut alors que la largeur d'un canal n'ait pas été estimée suffisamment large lors du placement. Dans ce cas le réajustement des boxes doit être repris au niveau de la ligne de coupe correspondante dans l'arbre binaire de placement, et les noeuds ancêtres doivent être réajustés.

Contrairement à la méthode de placement par Analyse Factorielle décrite dans la première partie de cette thèse, le placement par bipartitionnements successifs permet de prendre en compte des contraintes telles que les plots d'entrées-sorties ou les blocs fixes. Il est également plus aisé d'attribuer un ratio adéquat aux blocs dits "déformables". Ce sont ces différents points que nous allons étudier dans la suite.

II.4 Prise en compte des plots d'entrées-sorties du circuit

Chaque circuit doit nécessairement recevoir une alimentation électrique externe et communiquer avec l'extérieur. Ces liaisons sont établies par le biais de connexions entre les blocs du circuit et les plots d'entrées-sorties. Ces derniers forment une couronne sur la périphérie du circuit.

Si l'attribution des connexions sur les plots d'entrées-sorties n'est pas spécifiée, alors on ne tient pas compte de ces liaisons durant le placement. Il suffit ensuite de déterminer l'affectation plot-connexion qui permet de minimiser la longueur de routage.

Cependant, il est fréquent que l'attribution des connexions sur les plots d'E/S

soit imposée par le boîtier qui recevra le circuit, ou par la carte sur laquelle il sera implanté.

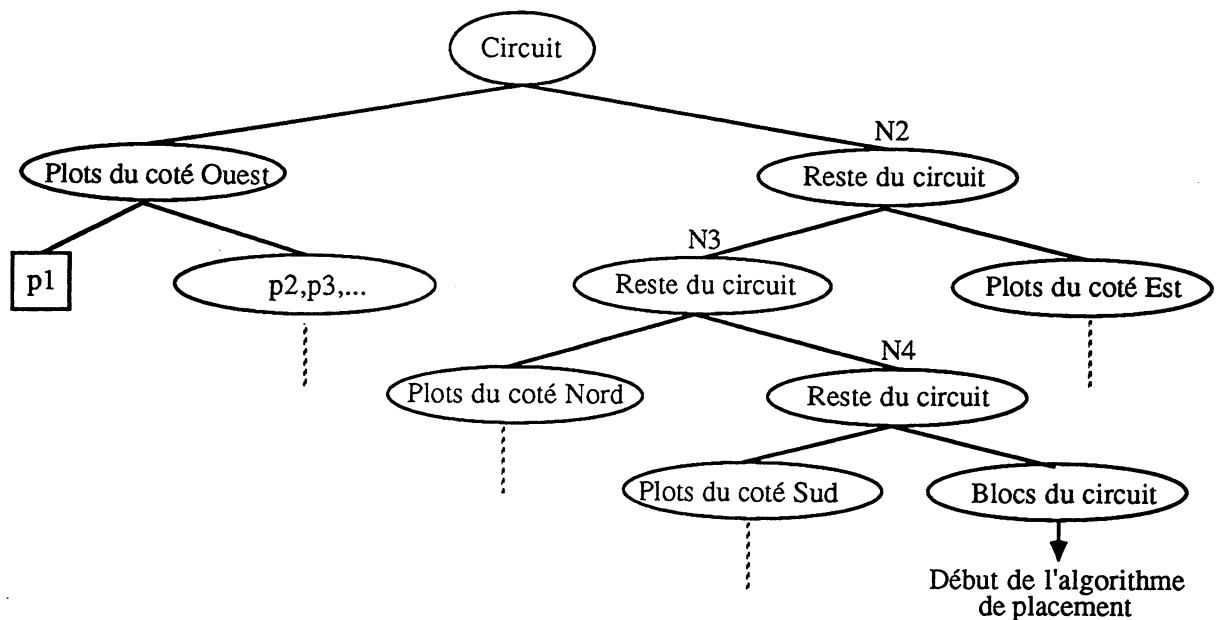
Dans ce cas, ces connexions doivent être prises en compte durant le placement, au même titre que les connexions internes du circuit, afin d'en minimiser la longueur.

Les plots d'E/S ont la particularité d'être fixes, ils ne peuvent donc pas être traités comme les autres blocs.

Nous effectuons donc un pré-traitement des plots, basé sur le fait qu'ils sont placés sur la périphérie du circuit. En effet, cette distinction permet dès le début de les isoler, au moyen de lignes de coupe, dans la partie haute de l'arbre binaire de placement :

Pour chacun des quatre côtés, une ligne de coupe sépare les plots correspondants du reste du circuit, puis récursivement dans chacun de ces segments, des lignes de coupe isolent chaque plot dans un box. Evidemment durant cette phase, les lignes de coupe ne sont pas déterminées de la même façon que durant le placement effectif. Ici le critère d'équilibrage n'est pas pris en compte puisque l'objectif à atteindre est uniquement d'isoler chaque plot dans un segment.

Quand chaque plot est isolé dans un box, l'algorithme de placement est appliqué sur le noeud ne contenant plus que les blocs internes du circuit. La figure suivante illustre alors l'allure de l'arbre binaire de placement.



Arbre binaire de placement contenant les plots d'E/S.

Durant tout l'algorithme de placement, la connectique avec les plots est prise en compte en les considérant comme modules fictifs lorsque cela est nécessaire. Le fait d'isoler chaque plot dans un box avant le début de l'algorithme de placement permet de connaître exactement sa position afin de l'introduire éventuellement comme module fictif dans l'ensemble à bipartitionner. Dans ce cas le plot est toujours introduit dans le bon segment.

Ce procédé assure une migration rapide des modules fortement connectés aux plots, vers la périphérie du circuit.

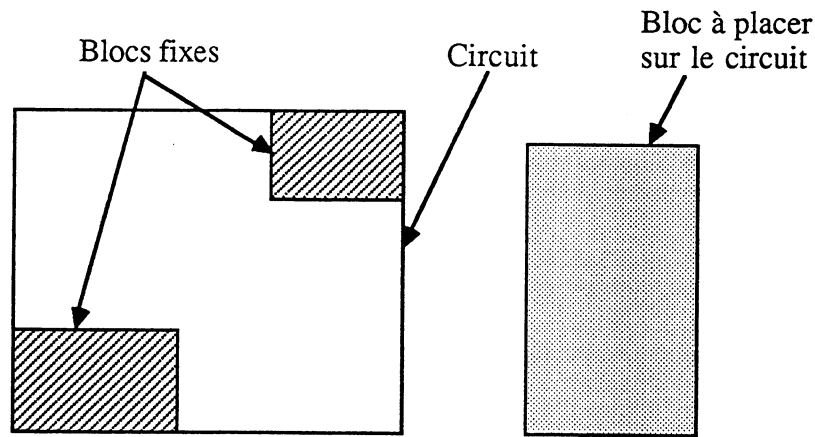
Lors du réajustement des positions des modules, les positions individuelles des plots ne doivent pas être modifiées car elles obéissent à des règles de conception très strictes qui sont fonction de la technologie utilisée. Pour cela on n'effectuera pas de réajustement à l'intérieur des noeuds contenant les plots de chacun des côtés; ces noeuds seront considérés comme déjà ajustés. Ainsi, seul l'ajustement des noeuds N2, N3, N4 et de la racine sera effectué en plus de celui du noeud contenant les modules du circuit.

II.5 Prise en compte de blocs fixes sur le circuit

Parmi les contraintes de faisabilité du placement, il peut exister des contraintes sur l'emplacement de certains blocs. Cela peut être le cas notamment pour un bloc dont la longueur des connexions avec les plots d'E/S doit être absolument minimisée. Dans ce cas on peut vouloir imposer au bloc la position en bordure des plots concernés. Un placement qui ne respecterait pas cette contrainte serait alors considéré comme non réalisable.

L'introduction d'une contrainte de bloc fixe implique que la surface hôte du circuit soit déterminée, afin de localiser la position du bloc concerné.

La prise en compte de telles contraintes est délicate, car elles peuvent réduire considérablement l'ensemble des dispositions possibles pour les blocs mobiles. Il faut alors que l'algorithme de placement soit apte à trouver une de ces configurations. On peut même imaginer que le respect de ces contraintes n'autorise aucune configuration des modules, comme dans l'exemple page suivante.



Disposition des blocs fixes n'autorisant aucun placement réalisable.

Lorsque les contraintes de blocs fixes consistent à n'imposer que des positions en bordure des plots d'E/S il est possible d'utiliser l'algorithme tel que nous l'avons défini. Il faut pour cela créer de nombreuses connexions entre les blocs fixes et les plots voisins, dans le but de forcer une migration des blocs fixes vers leur position. Ceci nécessite évidemment de prendre en compte les connexions avec les plots d'E/S durant le placement.

Dans le cas de positions quelconques, nous proposons ici une approche qui permet généralement d'obtenir un placement respectant les contraintes, lorsque celles-ci ne sont pas trop restrictives quant aux possibilités de placement des autres modules.

Cette approche repose essentiellement sur le fait que, durant le bipartitionnement d'un ensemble de modules, il est possible de figer des modules dans les segments. Ainsi, à tous les niveaux de l'arbre binaire il suffit de maintenir chaque bloc fixe dans le segment qui contient son emplacement.

Lors du traitement d'un noeud, la position de la ligne de coupe est déterminée après le bipartitionnement des modules, en fonction de la taille des segments (voir paragraphe II.1.2). Dans notre cas nous avons besoin de cette position avant le bipartitionnement, afin d'affecter chaque module fixe dans le bon segment.

Le but du bipartitionnement étant de créer deux segments de taille équivalente, on fait l'hypothèse que la ligne de coupe sera située au milieu du box.

Le bipartitionnement est ensuite réalisé en figeant chaque module fixe du noeud dans le segment qui contient son emplacement.

Si un module fixe chevauche la ligne de coupe, on doit choisir quel est le

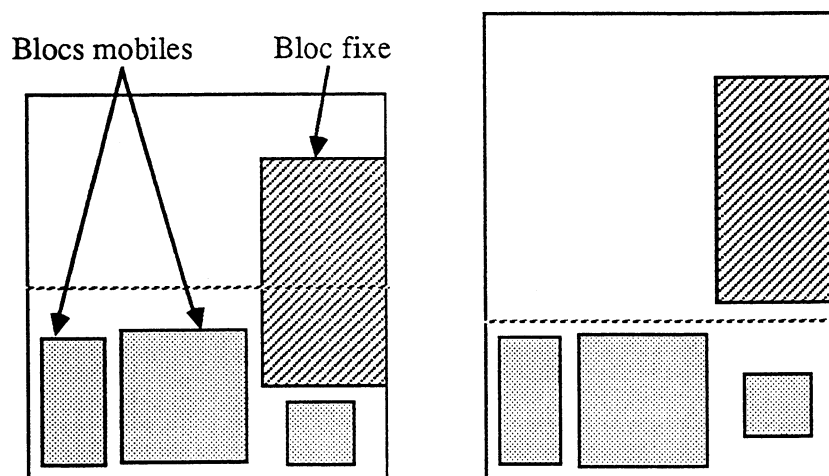
segment qui l'accueillera. Il paraît logique d'affecter le module fixe au segment qui en contient la plus grande partie.

Cependant cette alternative présente des inconvénients :

- D'une part, le nombre de bipartitions différentes qui respectent la contrainte d'équilibrage est réduit car la surface du module fixe est entièrement affectée à un segment et ceci diminue les possibilités d'affectation de modules libres à ce segment.

- D'autre part, l'arbre binaire nous conduit à obtenir un placement issu de coupes successives. Ainsi, lorsque durant le réajustement des boxes on atteindra celui du noeud concerné, la situation sera la même que si l'on avait déplacé la ligne de coupe jusqu'à ce qu'elle ne superpose plus le bloc. Il en résulte une perte importante de surface car durant le partitionnement des modules du noeud, le critère d'équilibrage ne correspond pas à ce que sera le rapport final des surfaces des boxes.

La figure suivante illustre une telle situation.



Le bloc fixe étant affecté entièrement au segment du haut durant le bipartitionnement, les autres modules doivent obligatoirement être situés dans celui du bas. Ceci entraîne une perte de surface lors du réajustement.

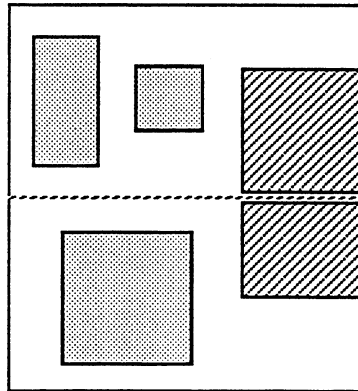
Des pertes de surface du même type peuvent survenir également dans le cas de blocs mobiles, mais plus rarement car ces derniers peuvent être orientés de façon à minimiser la surface du box.

Par conséquent, lorsqu'un bloc fixe chevauche la ligne de coupe, il est divisé en deux sous-modules fixes selon cette ligne, et chaque partie est figée dans le segment qui la contient. Cela permet une meilleure adéquation entre les surfaces des boxes et des segments et autorise une plus grande liberté de déplacement des modules, en respect du critère d'équilibrage.

Chacun des deux nouveaux modules fixes hérite des connexions situées sur son

contour.

L'exemple précédent peut alors donner :



Chacune des parties du bloc fixe est affectée à un segment, ce qui autorise plus d'échanges pour les blocs mobiles.

Le bipartitionnement des modules est ensuite effectué. Le rapport de taille des deux segments obtenus n'est généralement pas le même que celui des deux boxes correspondants. Dans ce cas la position de la ligne de coupe peut être ajustée si elle ne modifie pas la répartition des modules fixes dans les segments. Ce réajustement est utile car les dimensions des boxes permettent ensuite de guider l'orientation des modules pour l'optimisation de la surface du circuit. Si un module fixe chevauche la ligne de coupe, ce réajustement n'est pas effectué.

Nous n'avons pas trouvé de stratégie qui permette de déterminer une position de la ligne de coupe avant bipartitionnement, qui soit correcte après, à moins d'imposer une tolérance nulle, ce qui évidemment nuit à l'optimisation de la coupe.

Néanmoins, il est possible de réitérer le processus avec une nouvelle position de la ligne de coupe, qui respecte le rapport des segments. Ceci dans l'espoir d'obtenir deux nouveaux segments dont le rapport des tailles soit le même que celui des boxes. Pour cela on peut diminuer, entre chaque passe, la valeur de la tolérance, mais la convergence n'est pas assurée.

La direction de la ligne de coupe est choisie comme précédemment, avec, lorsque c'est possible, une minimisation du nombre de modules fixes coupés. Cela permet par la suite de faciliter la reconstitution des modules fixes initiaux.

Les noeuds de l'arbre sont ainsi traités jusqu'à ce que chaque feuille ne contienne qu'un module. On procède alors à l'orientation des modules mobiles

et au réajustement des positions. A ce stade, les boxes des blocs fixes correspondent exactement aux emplacements imposés sur le circuit, et chacun des blocs fixes initiaux a pu générer plusieurs nouveaux blocs.

Le réajustement tel qu'il est présenté dans la partie II.3 nécessite une structure d'arbre binaire. Pour cette raison, il n'est pas possible de reconstituer chaque bloc fixe initial à partir des blocs qu'il a généré, car cela n'aboutirait probablement pas à un placement pour lequel il existe un arbre binaire.

Deux solutions sont alors possibles :

- Effectuer le réajustement sur l'arbre binaire tel qu'on l'a obtenu :

Cela engendre une perte de cohésion des blocs fixes formant un bloc initial. On doit donc ensuite reformer chaque bloc fixe initial et, si c'est nécessaire, effectuer une décompaction au voisinage de ces blocs.

- Modifier l'arbre obtenu en choisissant pour chaque bloc fixe initial un des sous-blocs générés, auquel on attribuera les dimensions d'origine et toutes les connexions. Les autres blocs seront alors considérés comme ayant une surface nulle. Le sous-bloc choisi est celui ayant la surface la plus importante.

On effectue ensuite le réajustement sur cet arbre.

Cet artifice permet de garder la cohésion des blocs fixes tout en conservant la structure d'arbre binaire. Néanmoins, on perd une partie des avantages apportés par la division des blocs fixes.

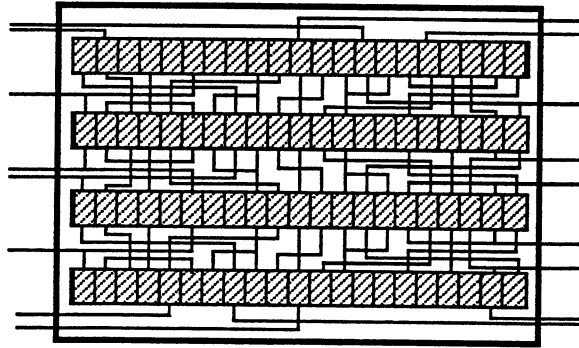
Quand les blocs fixes sont situés sur les bords du circuit, les possibilités qu'ils soient coupés sont restreintes ; cela est d'autant plus vrai qu'ils sont dans les angles. Dans ces conditions l'ajustement des positions ne pose pas de problème et les contraintes de blocs fixes sont bien prises en compte.

II.6 Prise en compte des blocs déformables

Parmi les différents blocs du circuit, peuvent se trouver des blocs dits "déformables" dont ni les dimensions, ni les positions des connecteurs ne sont connues. Ce sont généralement des blocs de logique combinatoire constitués de cellules standards, dont on estime la surface en fonction du nombre de cellules. Le placement et le routage des cellules standards au sein du bloc constituent un vaste problème que nous n'aborderons pas ici. Il nous suffit de savoir qu'il se caractérise par les deux points suivants :

- Les cellules standards sont disposées en plusieurs rangées situées les unes au-dessus des autres.
- Le routage s'effectue dans les canaux laissés entre les rangées.

La figure suivante donne l'allure générale d'un bloc constitué de cellules standards.



Bloc déformable constitué de cellules standards.

On voit donc que, suivant le nombre de rangées choisi pour disposer les cellules standards, le rapport de forme du bloc varie. Evidemment, pour un bloc donné le nombre de rangées doit être compris dans une fourchette de valeurs acceptable. Ainsi, le bloc ne peut prendre qu'un nombre fini de facteurs de forme, celui-ci étant défini par le rapport de la hauteur du bloc sur sa largeur.

Nous supposerons dans la suite que le facteur de forme peut varier de façon continue entre la valeur minimale et la valeur maximale acceptable.

Généralement les entrées-sorties du bloc sont situées sur les côtés du bloc perpendiculaires aux rangées.

L'objectif de la prise en compte des blocs déformables est de déterminer pour chaque bloc le rapport de forme qui minimise la surface du circuit, et lorsqu'ils sont indéterminés, les côtés d'attachement des ports d'entrées-sorties du bloc.

II.6.1 Ajustement du rapport de forme des blocs

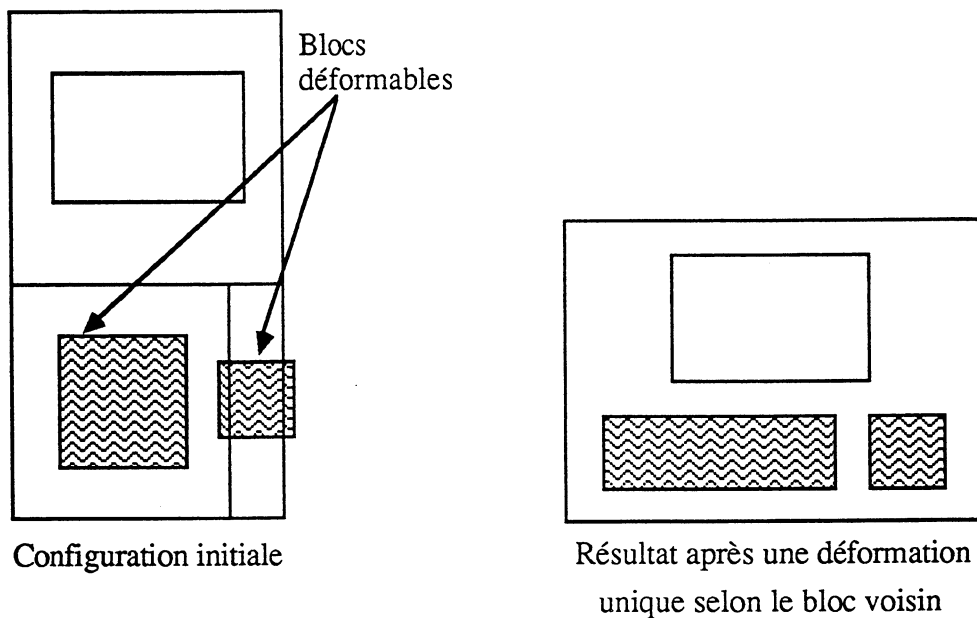
Lorsque la construction de l'arbre binaire de placement se termine, on remarque que sur chaque feuille, la surface du box et celle du bloc associé sont du même ordre de grandeur. En revanche, leurs dimensions ne sont pas forcément identiques. Tout débordement du box contribuera à une augmentation de la taille du circuit. D'où l'idée d'effectuer une première déformation des blocs mous avant la phase de réajustement, en leur faisant prendre le même rapport de forme que leur box. Ceci doit être fait dans la limite des valeurs acceptables pour le rapport de forme.

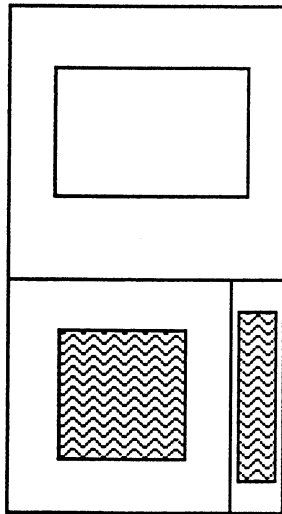
C'est à l'issue de la phase de réajustement, que va être déterminée la surface du circuit. Durant cette phase les dimensions du box de chaque noeud sont

réajustées de façon à englober les boxes de ses deux noeuds fils. Nous avons vu dans le paragraphe II.3 que les positions relatives de ces deux boxes sont fixées par les valeurs du canal et du décalage.

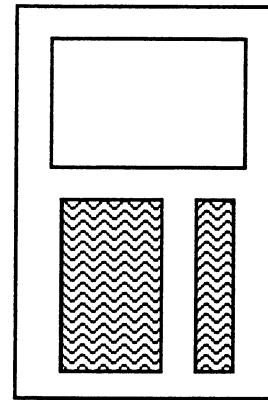
Lors du réajustement d'un noeud dont un des fils est un bloc déformable, la valeur du décalage est nulle car on ne connaît pas les positions des connecteurs sur ce bloc. La dimension du box dans le sens de la ligne de coupe est alors donnée par la plus grande dimension des fils dans ce sens, ce qui entraîne une perte de surface dans le cas où ces dimensions ne sont pas identiques. Pour éviter cela on déforme le bloc mou jusqu'à ce que ces dimensions soient les mêmes. Dans le cas où le rapport de forme ne le permet pas, on prend celui qui minimise la perte de surface.

Il peut sembler que cette deuxième transformation rende inutile la première, mais ce n'est pas le cas quand les deux fils du noeud à réajuster sont des blocs déformables. La figure suivante illustre ce qui se passe si l'on n'effectue que la seconde transformation.





Déformation de chaque module
selon la taille du box



Résultat après une seconde
déformation selon le bloc
voisin

On voit qu'il est préférable d'effectuer en premier lieu une déformation des blocs selon leur box, car ceux-ci reflètent un agencement parfait du circuit. De ce fait, plus on se rapproche des formes des boxes et plus on s'oriente vers une minimisation de la surface du circuit.

II.6.2 Côtés d'attachement des connexions

Les ports d'entrées-sorties du bloc sont généralement situés sur les côtés du bloc perpendiculaires aux rangées. Or certaines valeurs du rapport de forme permettent de disposer les rangées dans les deux directions; dans ce cas le choix est guidé par la meilleure position des ports d'E/S du bloc relativement à la connectique extérieure.

Nous définissons le rapport de forme d'un bloc mou par :

$$R = \frac{H}{L}$$

où H et L désignent la hauteur et la largeur du bloc.

H est donc fonction du nombre de rangées de cellules standards, tandis que L correspond à la largeur d'une de ces rangées.

On dit que le rapport de forme est symétrique si ses valeurs extrêmes R_{\min} et R_{\max} vérifient :

$$R_{\min} = \frac{1}{a} \leq R \leq R_{\max} = a \quad \text{avec } a \in \mathbb{R}$$

Dans ce cas, toutes les valeurs de R permettent de disposer les rangées de cellules standards dans les deux directions : on peut intervertir les valeurs de H et L sans dépasser les valeurs extrêmes du rapport de forme.

Lorsque $R_{\min} < 1$ et $R_{\max} > 1$, il existe un intervalle dans lequel le rapport de

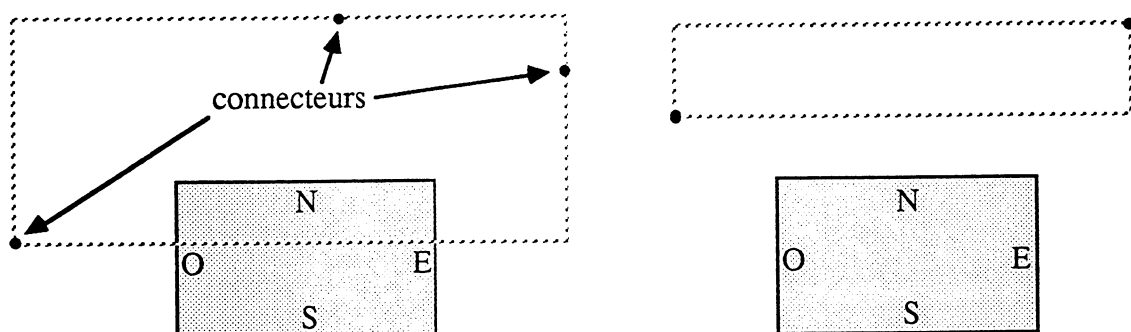
forme est symétrique. Si le rapport de forme choisi pour le bloc est situé dans cet intervalle, alors les côtés d'attachement des plots d'E/S sont indéterminés. On les détermine de façon à minimiser la connectique extérieure du bloc. Pour cela on détermine, pour chaque connexion, le côté d'attachement qui minimise la longueur estimée de son routage. On choisit ensuite comme côtés d'attachement de l'ensemble des connexions, les deux côtés opposés qui comptabilisent la majorité des côtés d'attachement calculés individuellement.

Il est fréquent que pour une connexion donnée, plusieurs côtés d'attachement minimisent la longueur estimée pour le routage. Dans ce cas la connexion sera comptabilisée dans le compte de chacun de ces côtés.

Considérons une connexion du bloc dont tous les connecteurs sont fixes, à l'exception de celui situé sur le bloc. Alors la longueur de routage de cette connexion est forcément supérieure à la longueur de routage nécessaire pour relier les connecteurs fixes. Cette dernière constitue donc une borne inférieure pour la longueur de la connexion et le côté d'attachement de la connexion sur le bloc courant doit être déterminé de façon à s'éloigner le moins possible de cette borne.

En choisissant un côté proche du tracé reliant les connecteurs fixes, on limite cet éloignement.

Il est raisonnable de supposer que le tracé reliant les connecteurs fixes est contenu dans le plus petit rectangle dont les côtés sont parallèles au circuit et qui englobe ces mêmes connecteurs. Ce rectangle se réduit d'ailleurs à un point quand la connexion est bi-terminale et à une droite quand elle est tri-terminale. On peut alors sélectionner comme côté d'attachement de la connexion sur le bloc, un des côtés les plus proches du rectangle. Si le rectangle recouvre partiellement ou totalement le bloc alors chacun des côtés dont une partie est recouverte peut être sélectionné.



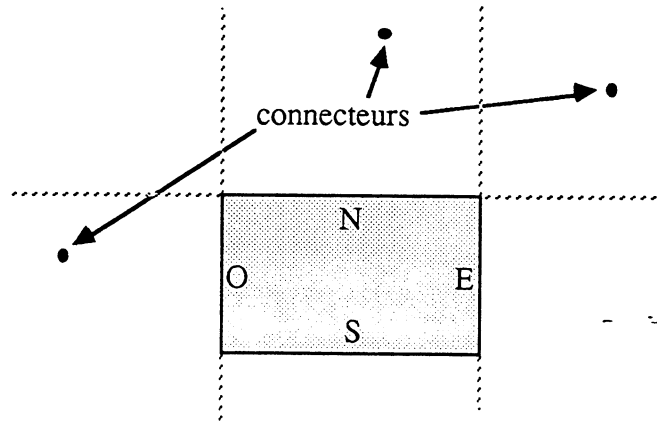
Dans les deux cas, les côtés Ouest, Nord et Est sont sélectionnés

Néanmoins, pour des raisons de simplicité, nous avons adopté une stratégie

approchante plutôt que de passer par le rectangle englobant.

Pour chaque côté du bloc, on considère l'ensemble des terminaux de la connexion situés dans le demi-plan du circuit, opposé au module et délimité par la droite prolongeant ce côté.

Si cet ensemble est non vide alors le côté peut être sélectionné comme côté d'attachement de la connexion.



Les côtés Ouest, Nord et Est sont sélectionnés

Il est fréquent d'obtenir plusieurs côtés qui satisfont cette condition, mais il n'est pas possible de les départager sans connaître la structure du réseau des canaux de routage.

Lorsque pour chaque connexion du bloc on a déterminé les meilleurs côtés d'attachement sur le bloc, il faut choisir quels seront les deux côtés opposés qui recevront effectivement l'ensemble de toutes les connexions. Chaque côté étant candidat à l'accueil d'un certain nombre de ces connexions, on choisit les deux qui en regroupent le plus grand ensemble.

Exemple :

Etant donné un bloc admettant 3 connexions. Si les côtés candidats à l'accueil sont:

connexion 1 : Ouest, Nord, Est.

connexion 2 : Ouest, Nord, Est.

connexion 3 : Sud.

Les côtés retenus seront Nord et Sud.

III. EXPÉRIMENTATION

Dans un premier temps, une comparaison des résultats entre le placement par bipartitionnements successifs et le placement par analyse du tableau des proximités est effectuée. Puis l'efficacité du traitement des nouvelles contraintes de plots ou de blocs fixes est illustrée sur un exemple.

III.1 Comparaison avec les résultats du placement par A.T.P

En ce qui concerne les circuits ayant une structure de maillage, on constate que lorsque le nombre de blocs est une puissance de 2 (16 blocs), le placement obtenu correspond exactement au maillage. Dans les autres cas, les résultats sont variables selon que la contrainte d'équilibrage permet ou non de retrouver une coupe compatible avec la structure du maillage. Il faut d'ailleurs noter que si, dans le cas du circuit 4×4, les blocs avaient des tailles très différentes les uns des autres, alors la configuration optimale ne serait sans doute pas retrouvée. En effet, il est peu probable que, dans ce cas, la contrainte d'équilibrage autorise les mêmes bipartitions.

L'ensemble des circuits utilisés lors du placement par A.T.P est repris ici. Le tableau ci-dessous présente les résultats obtenus avec le placement par bipartitionnements successifs en comparaison des meilleurs résultats obtenus avec le placement par A.T.P.

Les représentations graphiques des placements obtenus avec le placement par bipartitionnements successifs sont données en annexe.

Circuit	S_{bip}	L_{bip}	S_{atp}	L_{atp}
genc.1.1	135020016	3735692	211906112	6009952
genc.1.2	150422576	5326409	148476800	7491858
genc.1.3	143082464	4004480	118005600	4370413
genc.1.4	93436224	3359640	111163840	4644381
genc.1.5	57305672	1171044	58842464	1990362
genc.1.6	236150224	9415488	277388352	13859879
genc.1.7	4286824	44770	5541698	61538
genc.1.8	225375536	3968888	210318592	5860384
genc.1.9	131562072	3169266	174141312	5554510

S_{bip} et L_{bip} désignent la surface du circuit et la longueur estimée des connexions pour les placements issus de bipartitionnements successifs, tandis que S_{atp} et L_{atp} désignent la surface et la longueur estimée des connexions pour le meilleur résultat obtenu par la méthode d'analyse du tableau des proximités. La supériorité du placement par bipartitionnements successifs est incontestable puisque pour chaque circuit il fournit un meilleur placement en termes de longueur de connexion, et bien souvent en surface. On remarque que les placements obtenus ont des ratios proches de 1 et que la disposition des blocs

au sein du circuit est uniforme; ceci est dû au principe des coupes successives.

III.2 Traitement des blocs fixes et des plots d'E/S

Afin d'illustrer l'efficacité des algorithmes développés pour le traitement des blocs fixes et des plots d'entrées-sorties, nous avons utilisé le circuit régulier de 4x4 blocs.

Dans un premier temps, 8 plots d'E/S sont introduits autour du circuit à raison de deux par côté. Sur les côtés Ouest et Est, les plots sont connectés directement aux blocs du bord, tandis que sur les côtés Nord et Sud, les plots sont connectés aux blocs du bord opposé. Le résultat du placement est alors inversé par rapport à l'axe horizontal passant par le milieu du circuit, ce qui permet de retrouver une configuration optimale.

Chacun des quatre blocs centraux est ensuite déplacé et fixé dans le coin du circuit le plus proche. Ceci a pour conséquence de faire migrer les blocs qui se trouvaient dans les coins vers le centre du circuit, afin de fournir la nouvelle configuration optimale.

Les représentations graphiques des résultats obtenus sont données en annexe.

IV. CONCLUSION

Dans ce chapitre nous avons étudié une méthode classique de placement qui procède par bipartitionnements successifs de l'ensemble des modules. Cette méthode a été suggérée pour la première fois par M.A.Breuer en 1977; il s'agissait alors des principes généraux. Depuis, le placement par bipartitionnements successifs a été repris de nombreuses fois dans les publications et un grand nombre d'améliorations ont été apportées. Nous avons essayé, dans cette thèse, de présenter l'état de l'art actuel concernant cette méthode, en écartant toutefois les variations qui nous sont apparues inintéressantes.

Certains points spécifiques n'apparaissant pas, ou très peu, dans la littérature, nous avons développé et expérimenté quelques heuristiques. Elles concernent le traitement des blocs préfixés, des plots d'entrées-sorties et des blocs mous.

Suite à des discussions avec des concepteurs de circuits, ces évolutions nous étaient rapidement apparues indispensables afin de proposer un outil de placement qui réponde à leurs besoins.

Après l'étude du placement par analyse factorielle et la nécessité de développer un second outil de placement, le choix de la méthode des bipartitionnements successifs a été guidé en partie par la possibilité de développer de telles heuristiques.

Évidemment d'autres méthodes de placement peuvent permettre la prise en

compte de ces différents points. Certaines, comme l'utilisation du recuit simulé, ne nécessitent d'ailleurs aucune évolution particulière. Néanmoins notre choix s'est porté sur le placement par bipartitionnements successifs en raison de sa faible complexité, entraînant de rapides temps d'exécution, et de sa grande simplicité d'utilisation. A cela vient s'ajouter la bonne régularité des placements qu'il fournit, résultant du principe des coupes successives. La régularité du placement ne constitue pas en soi une contrainte à respecter mais elle est un atout important dans l'acceptation du résultat par un concepteur, le critère relatif à la longueur totale de connexions n'étant généralement pas appréciable de façon visuelle.

Les expérimentations effectuées sont venues confirmer la justesse de notre choix et les applications aux extensions que nous avons proposées ont validé les différentes heuristiques.

La méthode de placement par bipartitionnements successifs semble être arrivée à maturité et seule une amélioration du procédé de bipartitionnement d'un ensemble de modules pourrait éventuellement être appréciable. En effet celui-ci est toujours très dépendant de la bipartition initiale. D'autres méthodes de recherche de cette bipartition pourraient être envisagées. Nous pensons notamment à l'utilisation d'une technique d'analyse de données, comme la classification hiérarchique ascendante, qui peut aider à former des sous-ensembles fortement intra-connectés.

Enfin, signalons que le procédé de bipartitionnement d'un graphe peut être utilisé dans d'autres problèmes se posant lors de l'élaboration du placement, en particulier celui de la division des blocs de cellules standards, de façon à "combler les espaces vides" laissés dans le plan de masse [CHOTIN,BESSON &SAUCIER 91].

Conclusion générale

L'élaboration du plan de masse d'un circuit intégré constitue un problème important lors de la phase de conception. De la qualité du placement dépendent les performances et le coût du circuit. Depuis une vingtaine d'années, ce problème a suscité une quantité colossale de recherches et un nombre tout aussi important de publications aux inspirations les plus diverses. De la théorie des graphes à l'optimisation analytique, en passant par les techniques statistiques et la physique des matériaux, on ne compte plus le nombre d'applications qui ont été étudiées.

Pourtant, à l'heure actuelle, rares sont les circuits qui sont conçus à l'aide d'un outil de placement automatique. Les concepteurs ne sont pas convaincus de leur efficacité et on ne peut leur donner tort car, dans bien des cas, les solutions trouvées manuellement s'avèrent meilleures.

Paradoxalement, depuis quelques années, on note une baisse des publications et les chercheurs semblent se désintéresser du problème. Cela peut s'expliquer par la difficulté d'innovation dans un aussi vieux domaine, alors que de nouveaux problèmes, encore inexplorés, apparaissent continuellement du fait de l'évolution rapide des technologies de conception.

Dans cette thèse, nous avons essayé d'apporter une contribution à la résolution du problème de placement de circuits. Pour ceci, deux méthodes ont été étudiées.

La première est basée sur une technique statistique proche de l'analyse en composantes principales, l'analyse factorielle d'un tableau de distances, et recherche une minimisation de la longueur totale de connexions. Une analyse complète de cette méthode a été effectuée, et toutes les améliorations possibles ont été réalisées. Néanmoins, il ressort de la première partie de la thèse, que cette méthode peut difficilement fournir un placement final à elle seule, d'une part, parce qu'elle ne permet pas de prendre en compte les plots d'entrées-sorties du circuit ou les blocs pré-fixés, et d'autre part, parce qu'elle n'obéit à aucune contrainte de régularité, ce qui génère des circuits très désordonnés occupant une surface trop importante.

Dans la deuxième partie de la thèse, nous nous sommes intéressés à une méthode dérivée du bipartitionnement des sommets d'un graphe en deux sous-ensembles faiblement connectés. Après une étude de l'ensemble des méthodes existantes pour le problème du placement, celle-ci nous est apparue comme la plus apte à fournir, de façon continue, de bons résultats dans des délais

raisonnables.

Cette méthode, intitulée "placement par bipartitionnements successifs de l'ensemble des modules", a fait l'objet de nombreuses publications. Après en avoir exposé l'état de l'art, nous avons proposé des heuristiques concernant le traitement des blocs pré-fixés et des plots d'entrées-sorties qui ont été validées par l'expérimentation.

Les résultats obtenus par cette méthode de placement sont satisfaisants et surtout bien meilleurs que ceux fournis par l'analyse factorielle du tableau de distances.

Ceci nous amène à penser que le placement par bipartitionnements successifs constitue l'une des meilleures méthodes actuelles pouvant être appliquées au problème de la génération automatique de plan de masse de circuits intégrés.

Bibliographie

- [BARRA&Co. 87] "An application of data analysis methods and simulated annealing for automatic layout of circuits", J.R.Barra, M.Becker, E.F.Kouka et M.Tricot, Computer System Science and Engineering, Vol. 2, No. 1, p. 3-15, Janvier 1987.
- [BELLON 84] "Optimisation des methodes d'implantation automatisée sur les réseaux structurés", Alain Bellon, Thèse de 3eme cycle, Université de Grenoble, Décembre 1984.
- [BHANDARI,HIRSCH&SIEWIOREK 88] "The min-cut-shuffle: Toward a solution for the global effect problem of min-cut placement", 25th Design Automation Conference, 1988, p. 681-685.
- [BHASKER&SAHNI 86] "A linear algorithm to find a rectangular dual of a planar triangulated graph", 23rd Design Automation Conference, 1986, p. 108-114.
- [BOUROCHE&SAPORTA 80] "L'analyse des données", Presses Universitaires de France, coll. "Que sais-je ?", 1980, No.1854.
- [BREUER 77] " A class of min-cut algorithms", 14th Design Automation Conference, 1977, p. 284-290.
- [CHAISEMARTIN 86] "Contribution à la génération automatique de plans de masse", Thèse de 3eme cycle, Université de Grenoble, Novembre 1986.
- [CHAISEMARTIN&Co. 89] "Placement and channel routing by simulated annealing: some recent developments", P.Chaisemartin, G.Dreyfus, M.Fontet, E.Kouka, P.Loubières et P.Siarry, Computer System Science and Engineering, Vol. 4, No. 1, p. 35-48, Janvier 1989.
- [CHOTIN,MIGNOTTE&SAUCIER 90] "A Design Planner including a fast predictive floorplanning tool", Euro-Asic'90 International Conference, Juin 1990, p. 208-213.

- [CHOTIN,BESSON&SAUCIER 91] "Synthesis and partitioning of standard cells for floorplan optimization", Euro-Asic'91 International Conference, Mai 1991, p. 112-116.
- [DIDAY&Co. 83] "Eléments d'analyse des données", E.Diday, J.Lemaire, J.Pouget et F.Testu, Ed. Dunod, 1983.
- [FIDUCCIA&MATTHEYSES 82] "A linear-time heuristic for improving network partitions", 19th Design Automation Conference, 1982, p. 175-181.
- [FORBES 87] "Heuristic acceleration of force-directed placement" 24th Design Automation Conference, 1987, p. 735-740.
- [GAREY&JOHNSON 77] "The rectilinear steiner tree is NP-complete", SIAM Journal of applied mathematics, Vol. 32, No. 4, June 1977.
- [GAREY&JOHNSON 79] "Computer and intractability: a guide to the theory of N.P completeness" Freedman Edt., San Francisco, CA 1979.
- [GOLDBERG&BURSTEIN 83] "Heuristic improvement technique for bisection of VLSI Networks", IEEE International Conference on Computer Design, 1983, p. 122-125.
- [GOLUB&VAN LOAN 83] "Matrix computations", John Hopkins series in the mathematical sciences, The John Hopkins University Press, Baltimore and London, 1983.
- [GONDRAN&MINOUX 85] "Graphes et algorithmes", Collection de la direction des études et recherches d'Electricité de France, Editions Eyrolles, 1985.
- [HANAN,WOLF&AGULE 76] "A study of placement techniques", Journal of Design Automation and Fault Tolerant Computing, p. 28-61, 1976.
- [HERRIGEL &FICHTNER 89] "An analytic optimization technique for placement of macro-cells", 26th Design Automation Conference, 1989, p. 376-381.

- [JACKSON 89] "Directed graphs and the compaction of IC Designs", Contemporary Mathematics, Volume 89, 1989, p. 107-127.
- [KERNIGHAN&LIN 70] "An efficient heuristic procedure for partitioning graphs", Bell Sys. Tech. Journal, Vol. 49, February 1970, p. 291-308.
- [KIRKPATRICK,GELATT&VECCHI 83] "Optimization by simulated annealing", Science, vol. 220, No. 4598, p. 671-680, May 1983.
- [KLEINHANS,SIGL&JOHANNES 88] "GORDIAN: A new global optimization / rectangle dissection method for cell placement", IEEE International Conference on Computer-Aided Design, 1988, p. 506-509.
- [KLEINHANS,SIGL&JOHANNES 89] "Sea-of-gates placement by simultaneous quadratic programming combined with improved partitioning", VLSI'89 Conference, 1989, p. 445-454.
- [KOUKA&SAUCIER 87] "An application of exploratory data analysis techniques to floorplan design", 24th Design Automation Conference, 1987, p. 208-213.
- [KOUKA 89] "Un générateur automatique de plans de masse fondé sur des techniques statistiques d'analyse de données", Onde électrique, Mai-Juin 1989, p. 71-82.
- [KOUKA 89] "Génération automatique de plans de masse", Technique et Science Informatiques, Vol. 8, No. 6, 1989, p. 557-569.
- [KOZMINSKI&KINNEN 84] "An algorithm for finding a rectangular dual of a planar graph for use in area planning for VLSI integrated circuits", 21st Design Automation Conference, 1984, p. 655-656.
- [KRISHNAMURTHY 84] "An improved Min-Cut algorithm for partitioning VLSI networks", IEEE transactions on Computers, Vol. 33, No. 5, May 1984, p. 438-446.

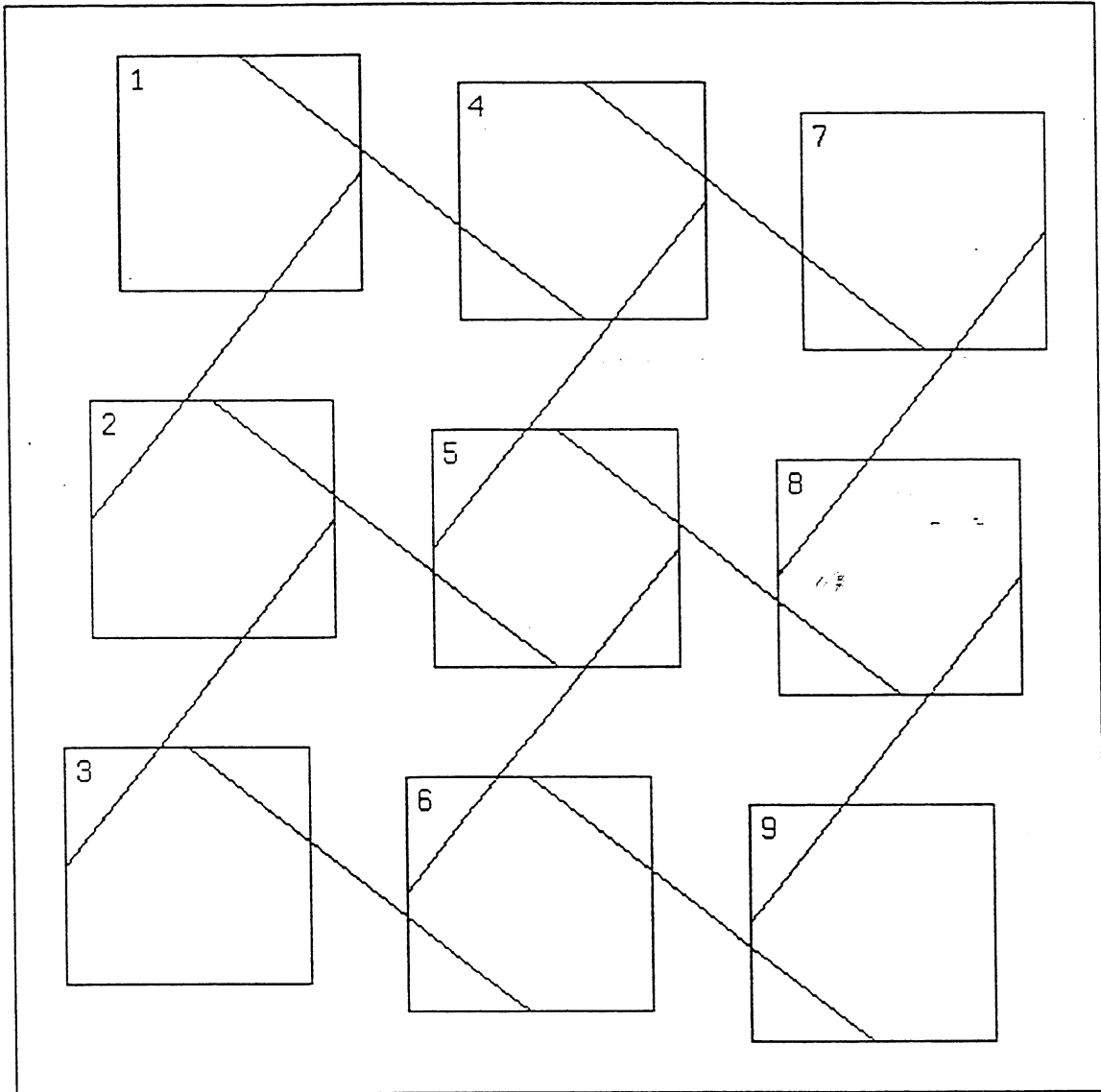
- [LAI&LEINWAND 88] "Algorithms for floorplan design via rectangular dualization", IEEE transactions on Computer-Aided Design, Vol. 7, No. 12, December 1988, p. 1278-1289.
- [LANDMAN&RUSSO 71] "On a pin versus block relationship for partitions of logic graphs", IEEE transactions on Computers, Vol. C-20, 1971, p. 1469.
- [LAPOTIN 85] "A global floorplanning approach for VLSI design", Research Report No. CMUCAD-85-69, SRC-SMU Center for Computer-Aided Design, Department of Electrical and Computer Engineering, Carnegie-Mellon University, Pittsburgh, PA 15213, USA.
- [LAPOTIN&DIRECTOR 86] "Mason: A global floorplanning approach for VLSI design", IEEE transactions on Computer-Aided Design, Vol. 5, 1986, p. 477-489.
- [LAUTHER 79] "A min-cut placement algorithm for general cell assemblies based on a graph representation", 16th Design Automation Conference, 1979, p. 1-10.
- [LAWLER 73] "Cutsets and partitions of hypergraphs", Networks, Vol. 3, 1973, p. 275-285.
- [LEINWAND&LAI 84] "An algorithm for building rectangular floorplans", 21st Design Automation Conference, 1984, p. 663-664.
- [LOKANATHAN&KINNEN 89] "Performance optimized floorplanning by graph planarization", 26th Design Automation Conference, 1989, p. 116-121.
- [NAHAR,SAHNI&SHRAGOWITZ 86] "Simulated annealing and combinatorial optimization", 23rd Design Automation Conference, 1986, p. 293-299.
- [NG,OLDFIELD&PITCHUMANI 87] "Improvements of a min-cut partition algorithm", IEEE International Conference on Computer-Aided Design, 1987, p. 470-473.

- [OREN 73] "Self-scaling variable metric algorithms without line search for unconstrained minimization", Math. Comp., 1973, vol. 27, No. 124.
- [OTTEN 82] "Automatic floorplan design", 19th Design Automation Conference, 1982, p. 261-267.
- [OTTEN&VAN GINNEKEN 84] "Floorplan design using simulated annealing", International Conference on Computer Design, November 1984, p. 96-98.
- [QUINN&BREUER 79] "A force directed component placement procedure for printed circuit boards", IEEE transactions on Circuits and Systems, Vol. 26, No.6, June 1979, p. 377-388.
- [RANKE&JOHANNES 89] "Macrocell placement by global optimization with uniform cell distribution", VLSI'89 Conference, 1989, p. 423-432.
- [ROSE,KLEBSCH&WOLF 90] "Temperature measurement and equilibrium dynamics of simulated annealing placements", IEEE transactions on Computer-Aided Design, Vol. 9, No. 3, March 1990, p. 253-259.
- [SAKAROVITCH 84] "Optimisation combinatoire: Programmation discrète", Méthodes mathématiques et algorithmiques, Hermann Enseignement des Sciences, 1984.
- [SHA&DUTTON 85] "An analytical algorithm for placement of arbitrarily sized rectangular blocks", 22nd Design Automation Conference, 1985, p. 602-608.
- [SHERMAN 89] "VLSI placement and routing: The PI project", Texts and monographs in computer science, Springer-Verlag, 175 Fifth Avenue, New York, NY 10010, USA.
- [SHIN,SANGIOVANNI-VINCENTELLI&SEQUIN 86] "Two dimensional compaction by zone refining", 23rd Design Automation Conference, 1986, p. 115-122.

- [SHIRAISHI&HIROSE 80] "Efficient placement and routing for masterslice LSI", 17th Design Automation Conference, 1980, p. 458-464.
- [SUTANTHAVIBUL,SHRAGOWITZ&ROSEN 90] "An analytical approach to floorplan design and optimization", 27th Design Automation Conference, 1990, p. 187-192.
- [TSITSIMIS 84] "Implantation automatique de circuits précaractérisés et prédiffusés", Thèse de 3eme cycle, Institut National Polytechnique de Grenoble, Octobre 1984.
- [YING&WONG 89] "An analytical approach to Floorplanning for hierarchical building blocks layout", IEEE transactions on-Computer-Aided Design, Vol. 8, No. 4, April 1989, p. 403-412.
- [ZIMMERMAN 88] "A new area and shape function estimation technique for VLSI layouts", 25th Design Automation Conference, 1988, p. 60-65.

Annexe I

Résultats du placement par ATP



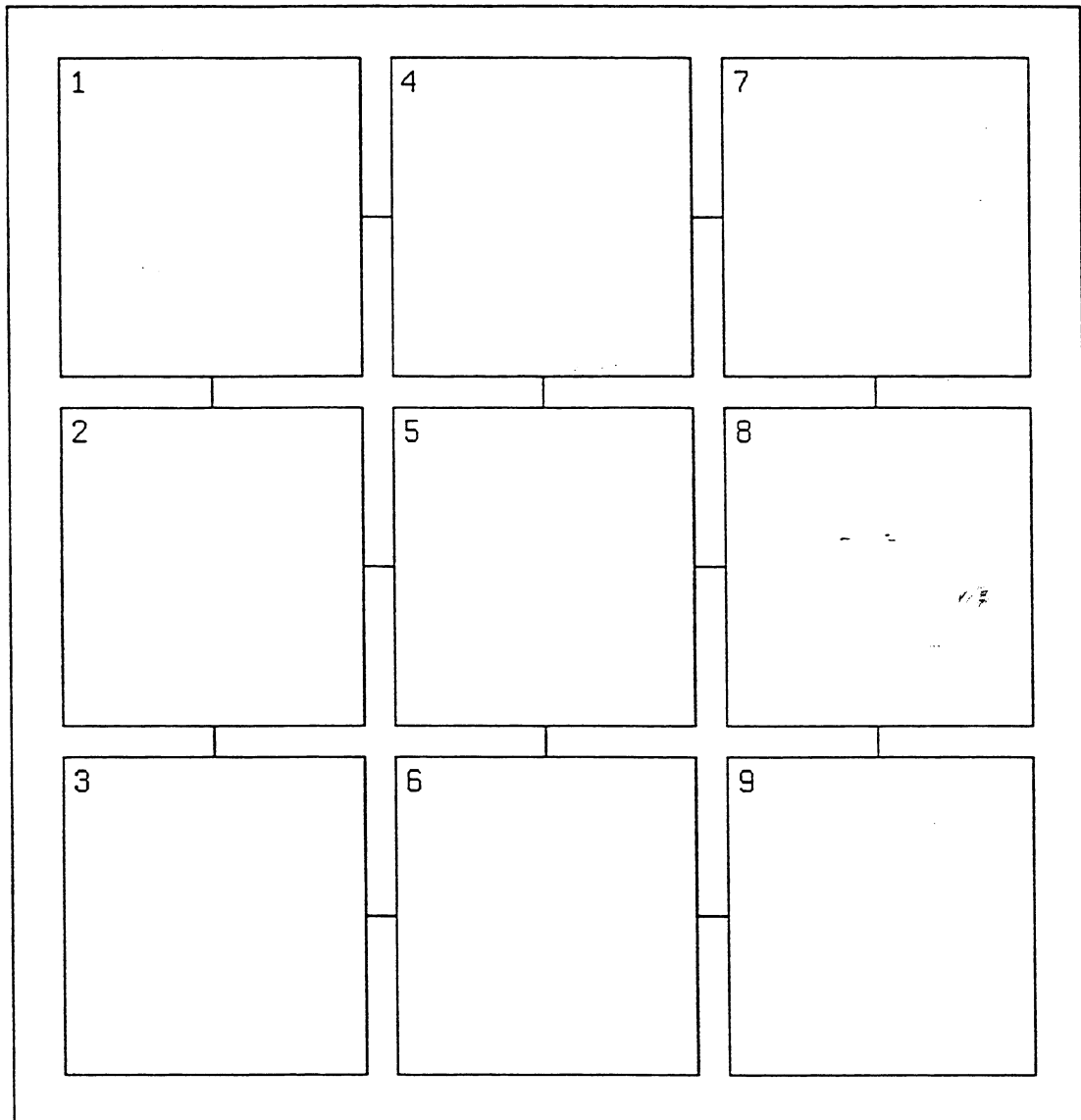
Circuit compose de 9 blocs et 12 connexions.

Projection du nuage sur le plan principal. Attention, pour visualisation les centres des blocs ont subis une homotethie de rapport 10540.

Qualite de representation du nuage sur les axes: 37% et 37%

Qualite de representation du nuage sur le plan: 74 %

Placement par analyse factorielle. Circuit: Dp9.ordonne



Placement obtenu apres decompaction et orientation des blocs.

Dimensions du circuit: 7456*7732

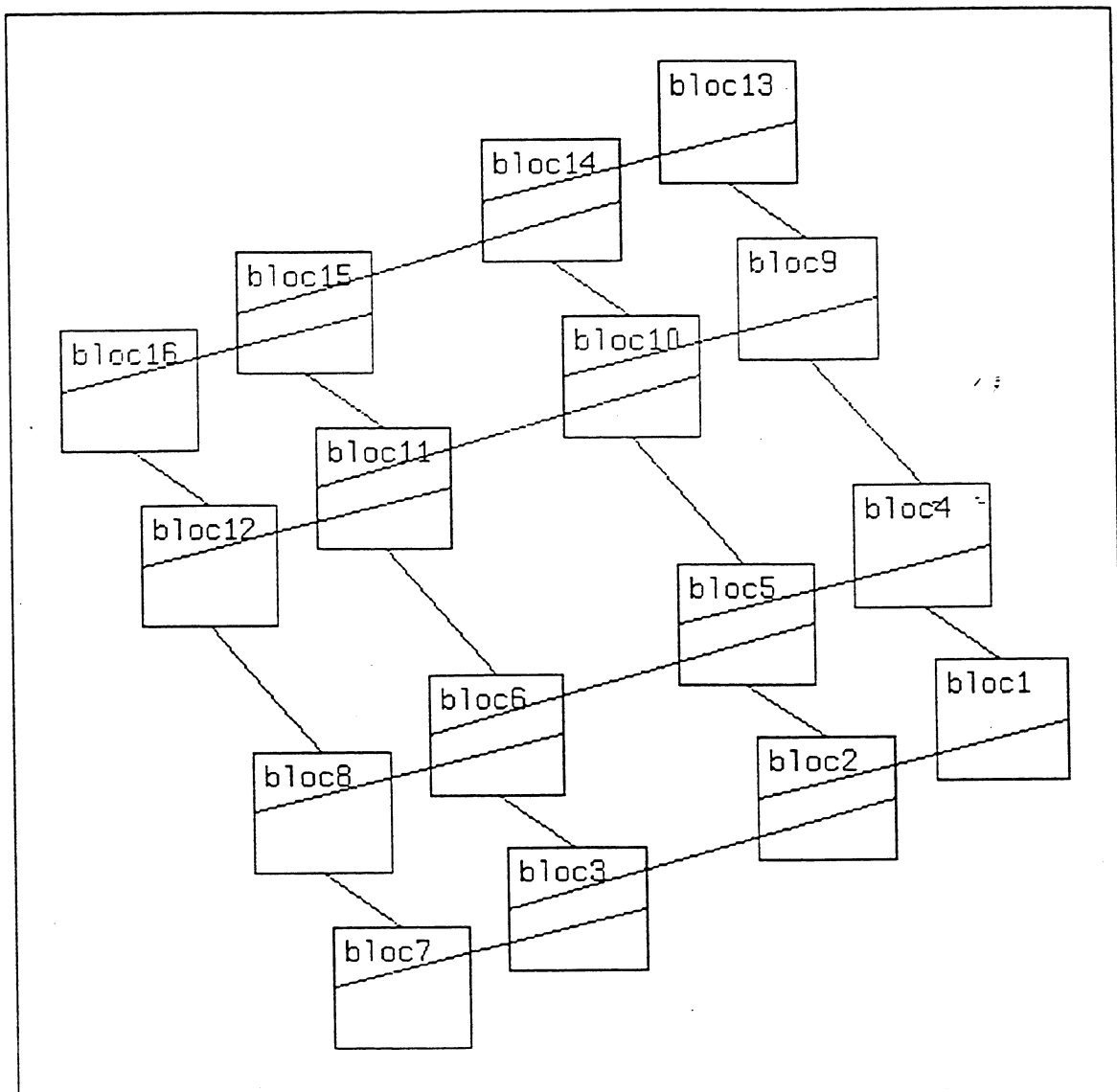
Surface du circuit: 57642236

Longueur estimee des connexions: 41437

Nombre total de blocs: 9

Nombre total de connexions: 12

Analyse factorielle, representation du nuage. Circuit: Dp16.desordonne



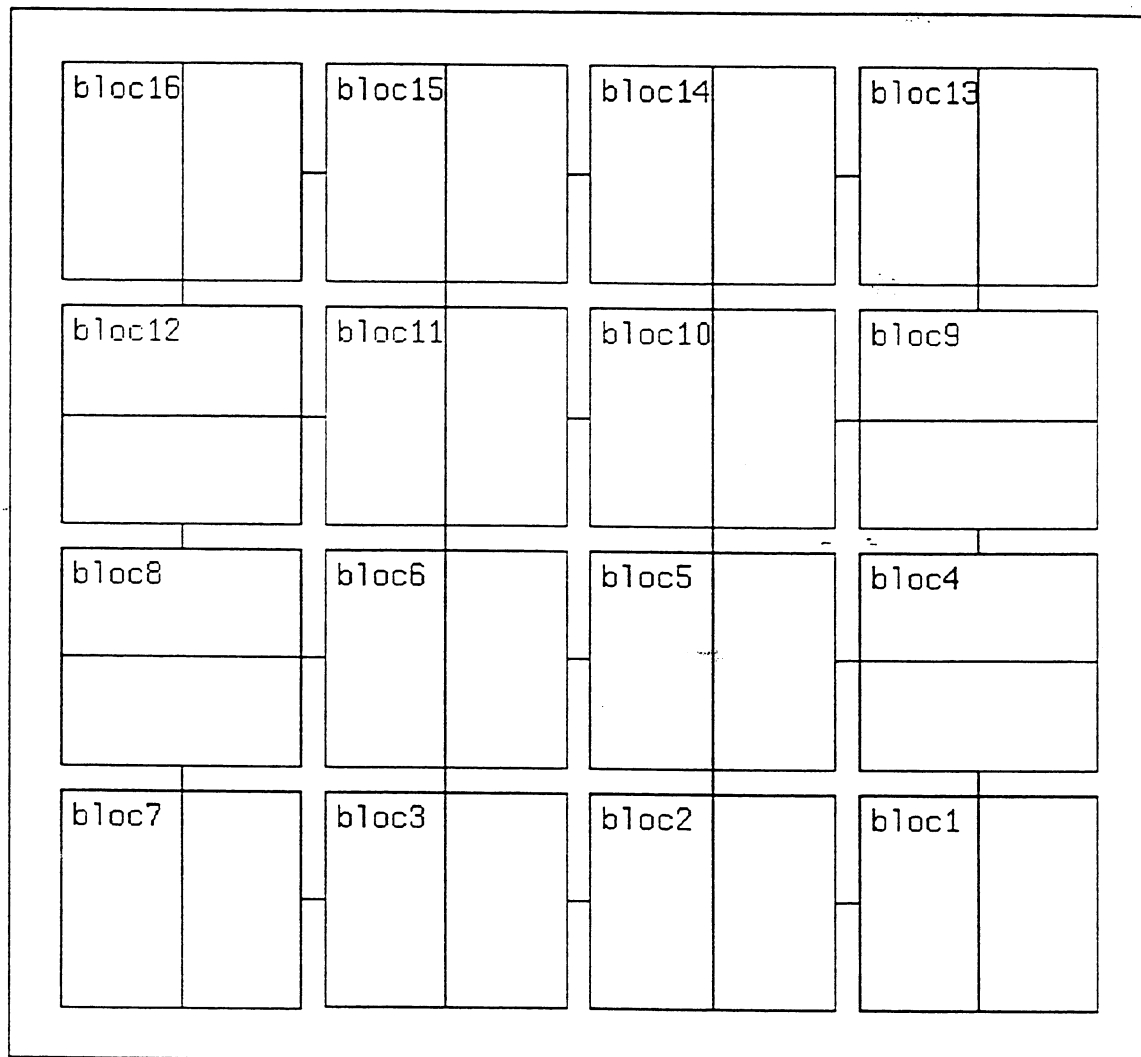
Circuit compose de 16 blocs et 24 connexions.

Projection du nuage sur le plan principal. Attention, pour visualisation les centres des blocs ont subis une homotethie de rapport 5081 .

Qualite de representation du nuage sur les axes: 34% et 34%

Qualite de representation du nuage sur le plan: 68%

Placement par analyse factorielle. Circuit: Dp16.desordonne



Placement obtenu apres decompaction et orientation des blocs.

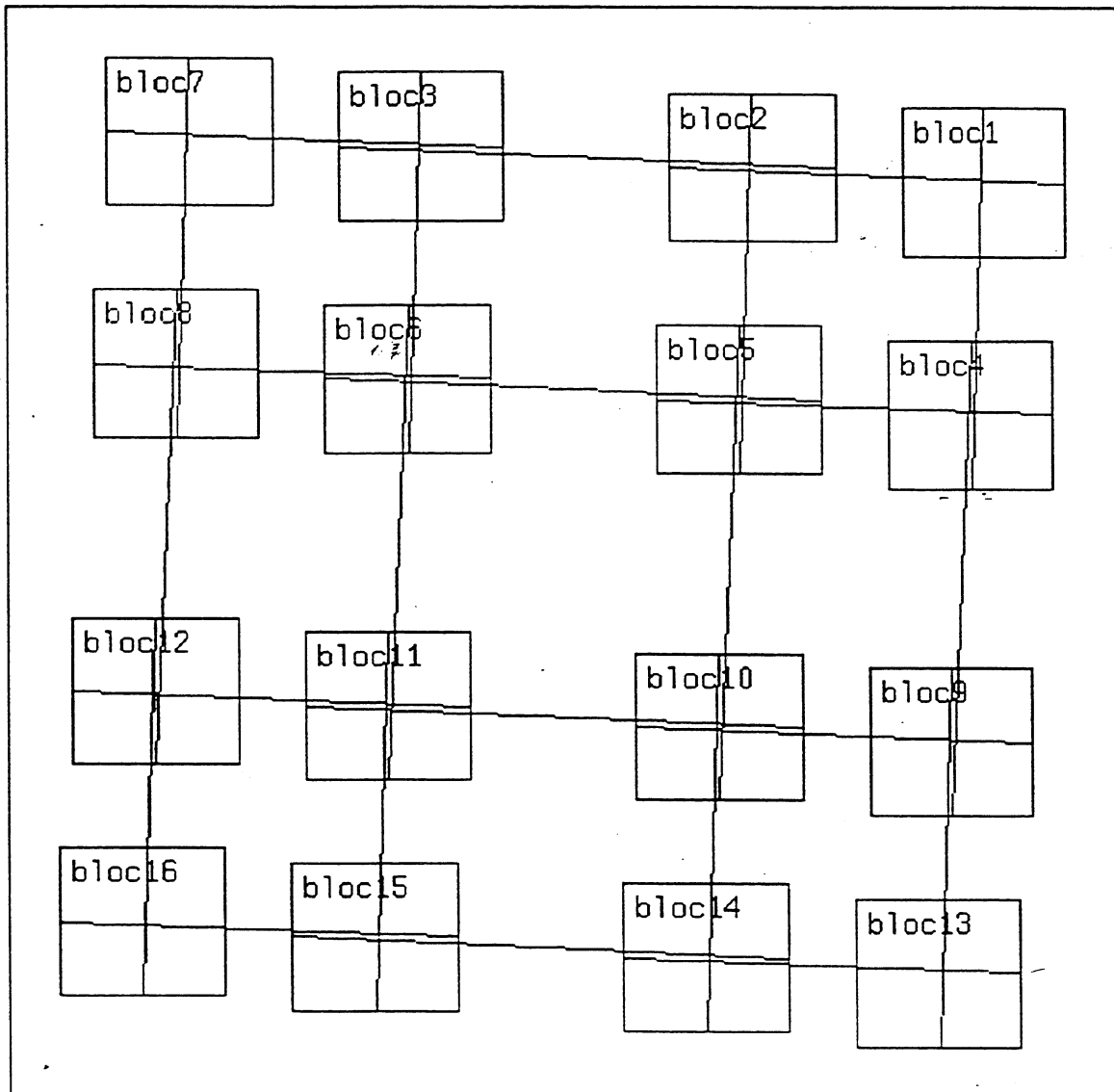
Dimensions du circuit: 4723*4319

Surface du circuit: 20395278

Longueur estimee des connexions: 166035

Nombre total de blocs: 16

Nombre total de connexions: 24



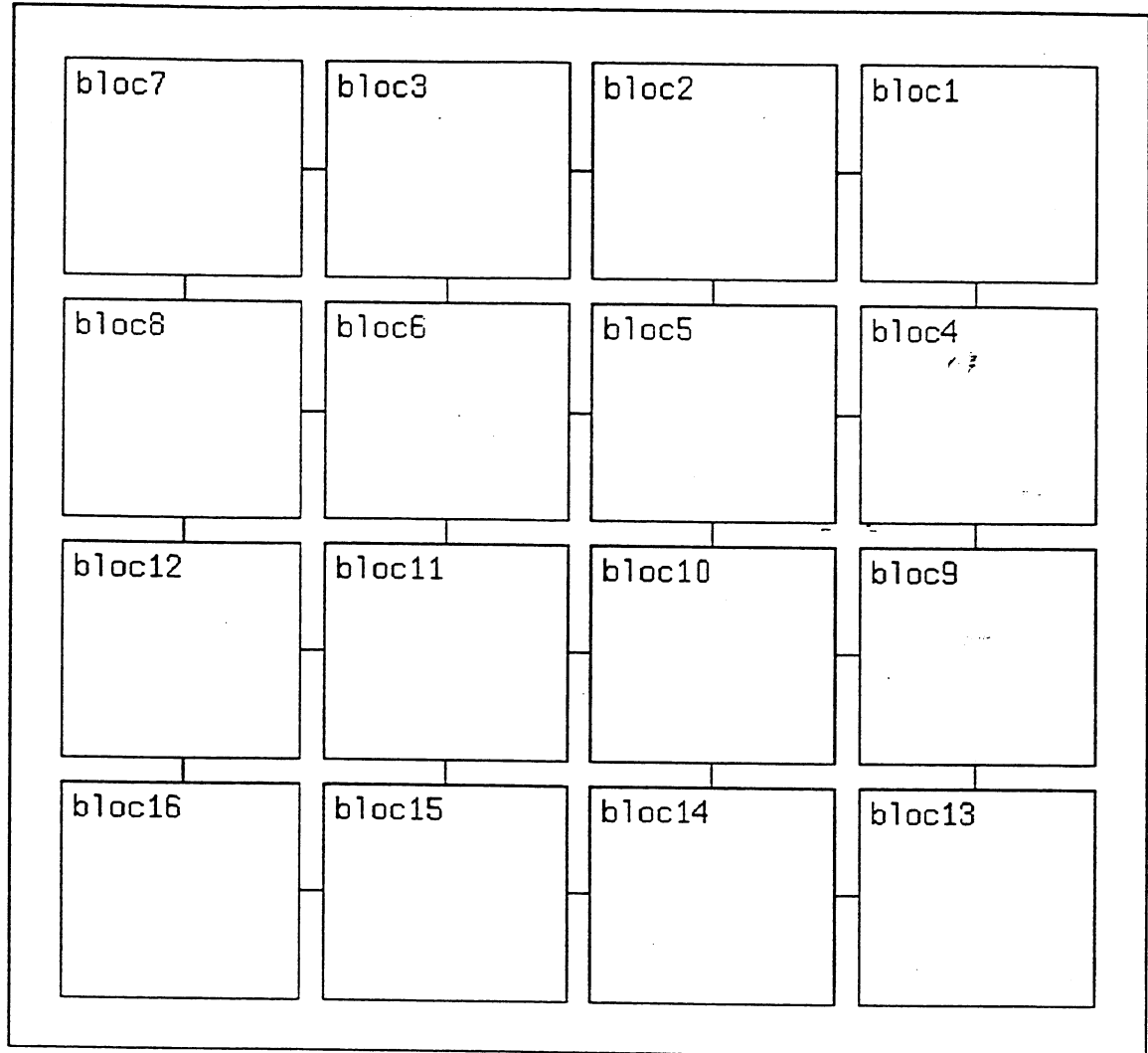
Circuit compose de 16 blocs et 24 connexions.

Projection du nuage sur le plan principal. Attention, pour visualisation les centres des blocs ont subis une homotethie de rapport 5081 .

Qualite de representation du nuage sur les axes: 34% et 34%

Qualite de representation du nuage sur le plan: 68%

Placement par analyse factorielle. Circuit: Dp16.ordonne



Placement obtenu apres decompaction et orientation des blocs.

Dimensions du circuit: 4722*4318

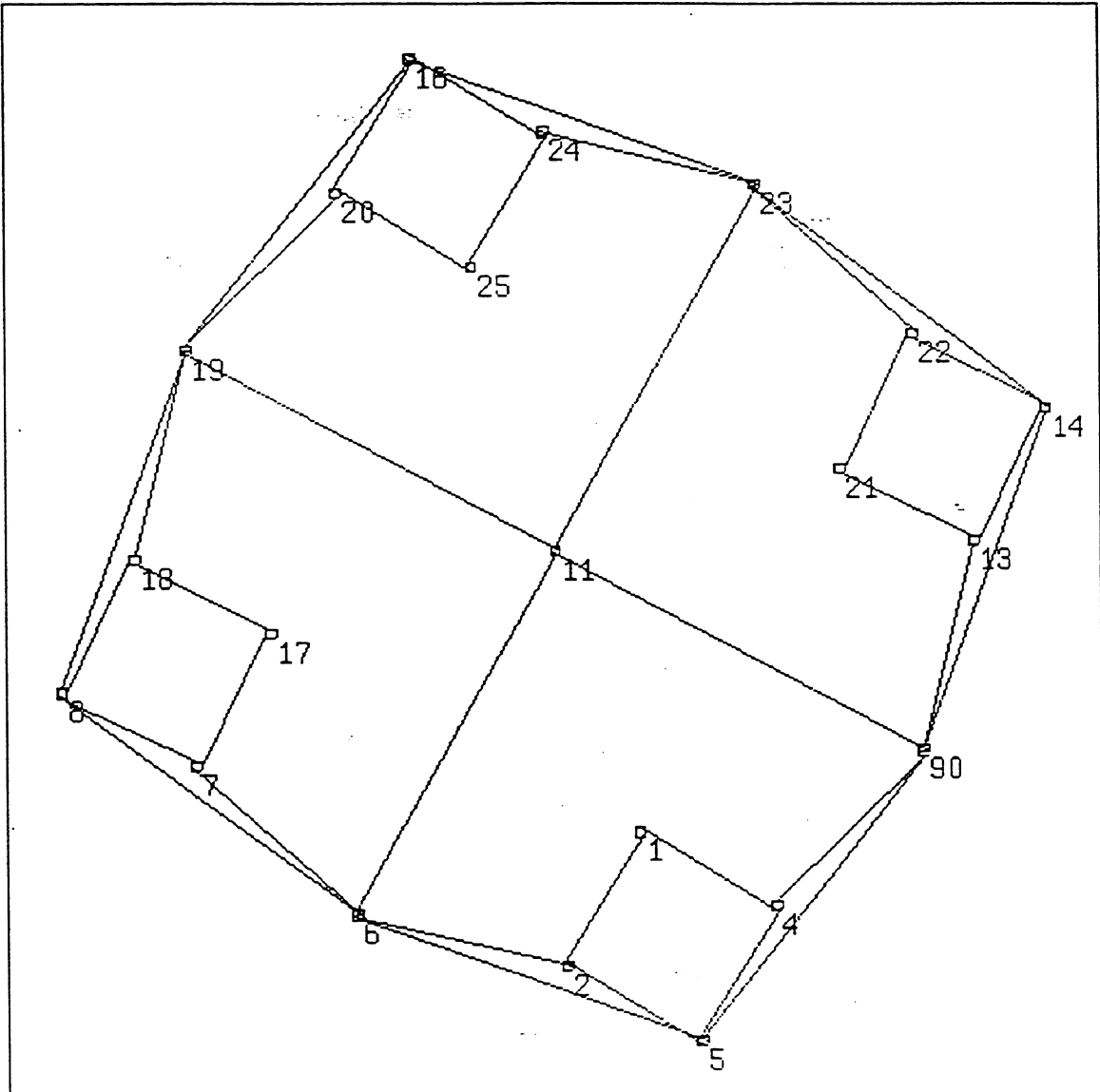
Surface du circuit: 20392846

Longueur estimee des connexions: 19639

Nombre total de blocs: 16

Nombre total de connexions: 24

Analyse factorielle, representation du nuage. Circuit: Dp25



Circuit compose de 25 blocs et 41 connexions.

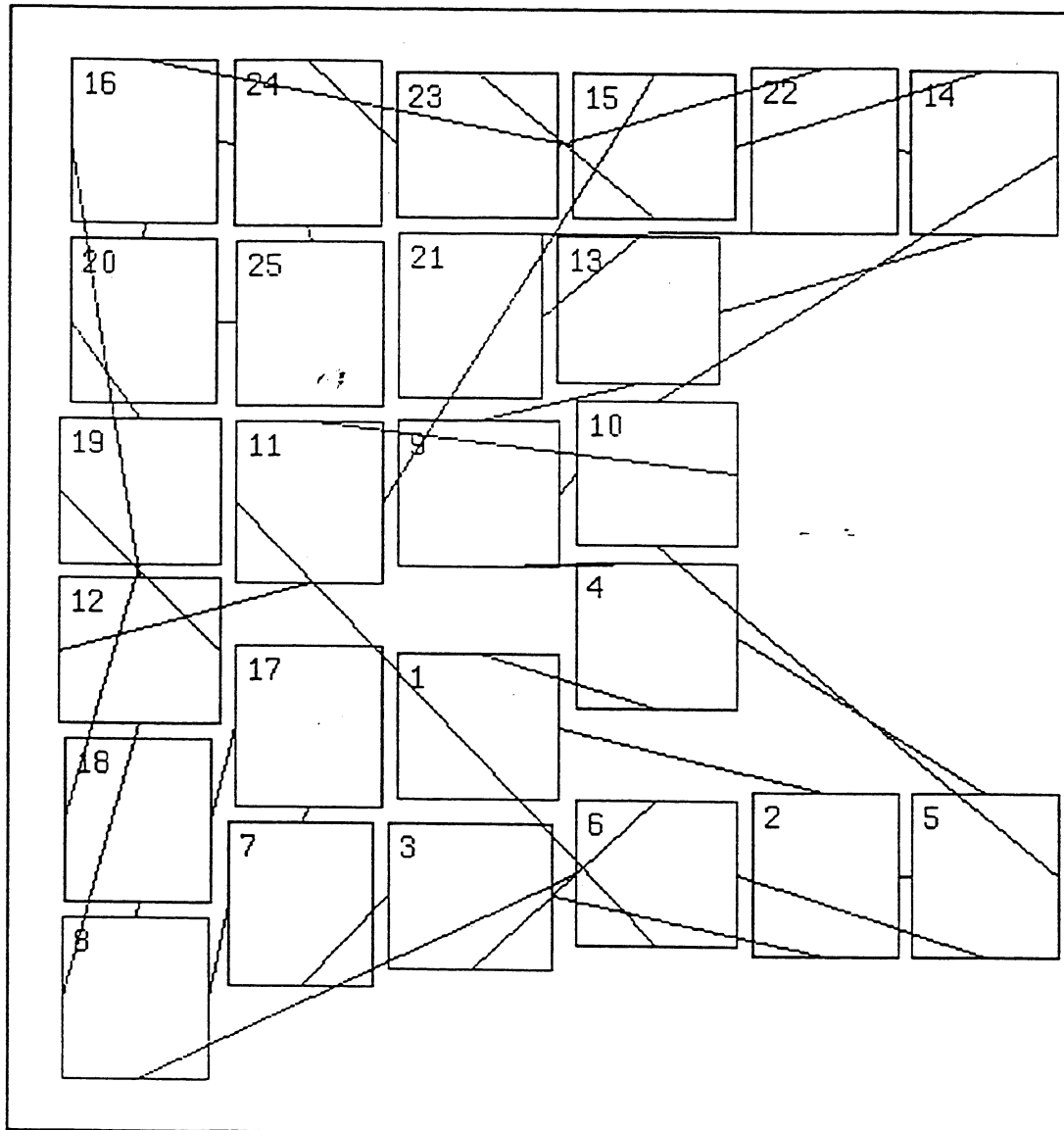
Projection du nuage sur le plan principal. Attention, pour visualisation les centres des blocs ont subi une homotethie de rapport ~~24777~~

Certains blocs se superposent.

Qualite de representation du nuage sur les axes: 10% et 10%

Qualite de representation du nuage sur le plan: 20%

Placement par analyse factorielle. Circuit: Dp25



Placement obtenu apres decompaction et orientation des blocs:

Dimensions du circuit: 6777*6886

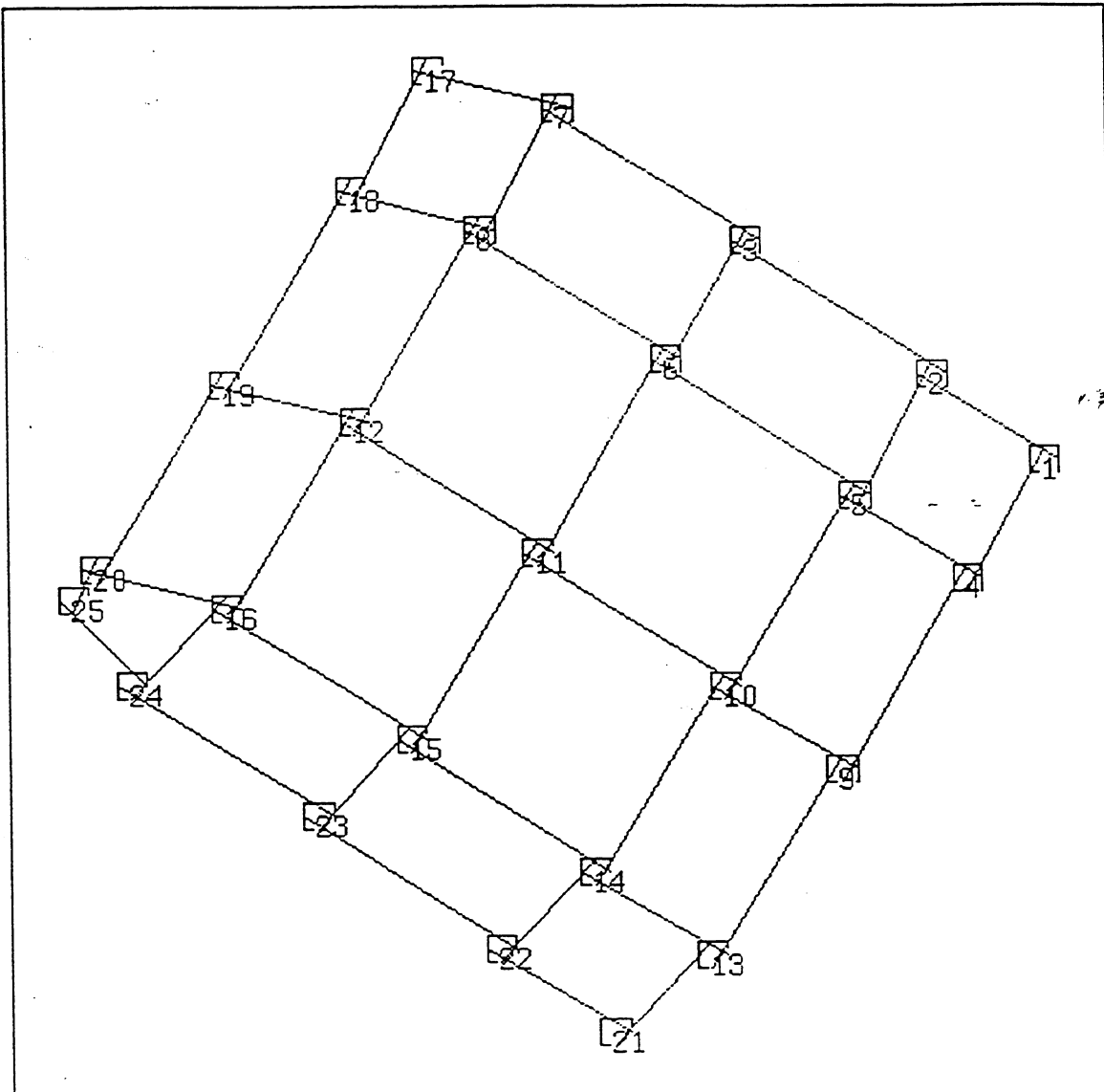
Surface du circuit: 46663212

Longueur estimee des connexions: 570687

Nombre total de blocs: 25

Nombre total de connexions: 41

Analyse factorielle, representation du nuage. Circuit: Dp25



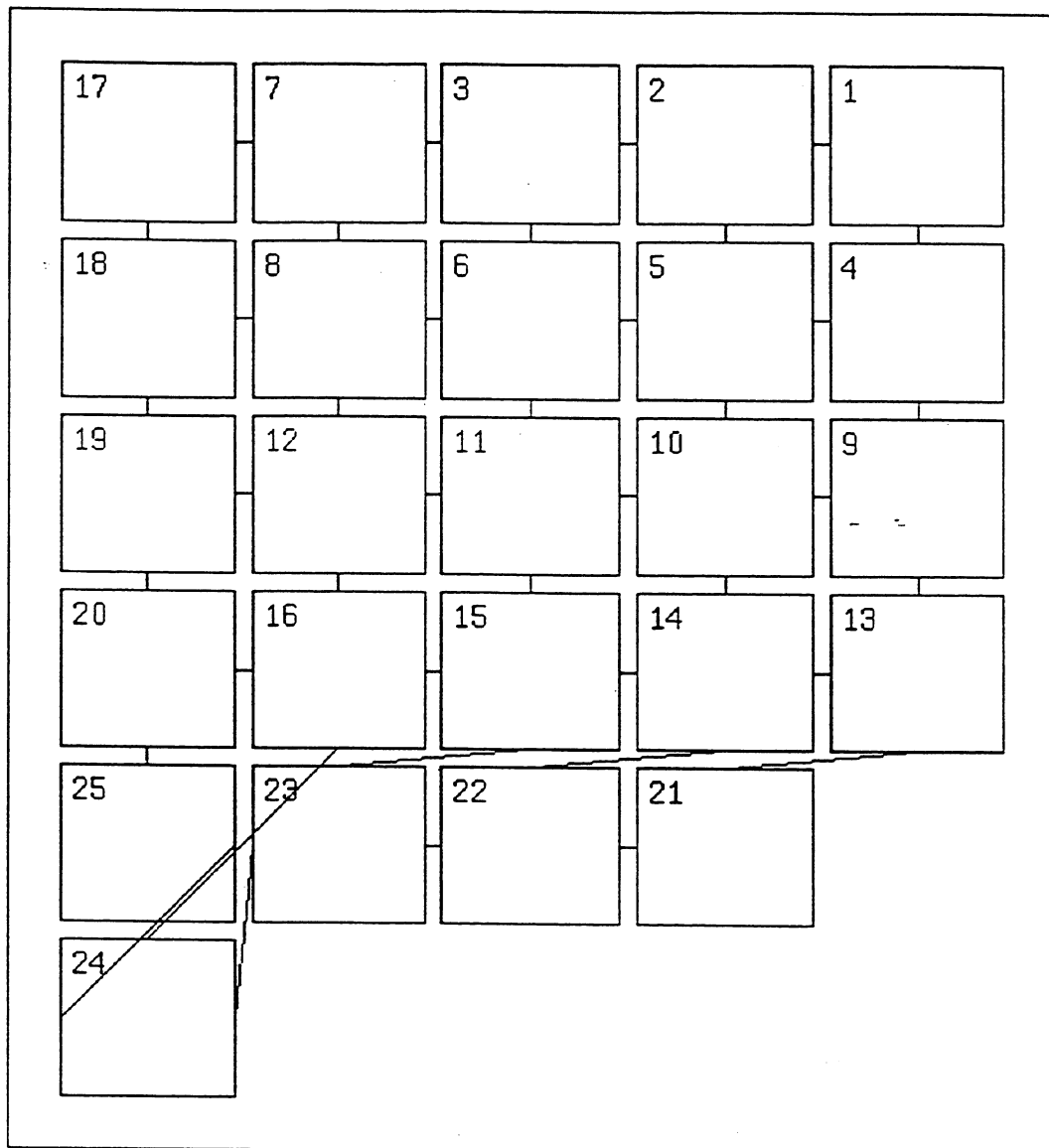
Circuit compose de 25 blocs et 41 connexions.

Projection du nuage sur le plan principal. Attention, pour visualisation les centres des blocs ont subis une homotethie de rapport 28410.

Qualite de representation du nuage sur les axes: 34% et 32%

Qualite de representation du nuage sur le plan: 66%

Placement par analyse factorielle. Circuit: Dp25



Placement obtenu apres decompaction et orientation des blocs.

Dimensions du circuit: 5980*6473

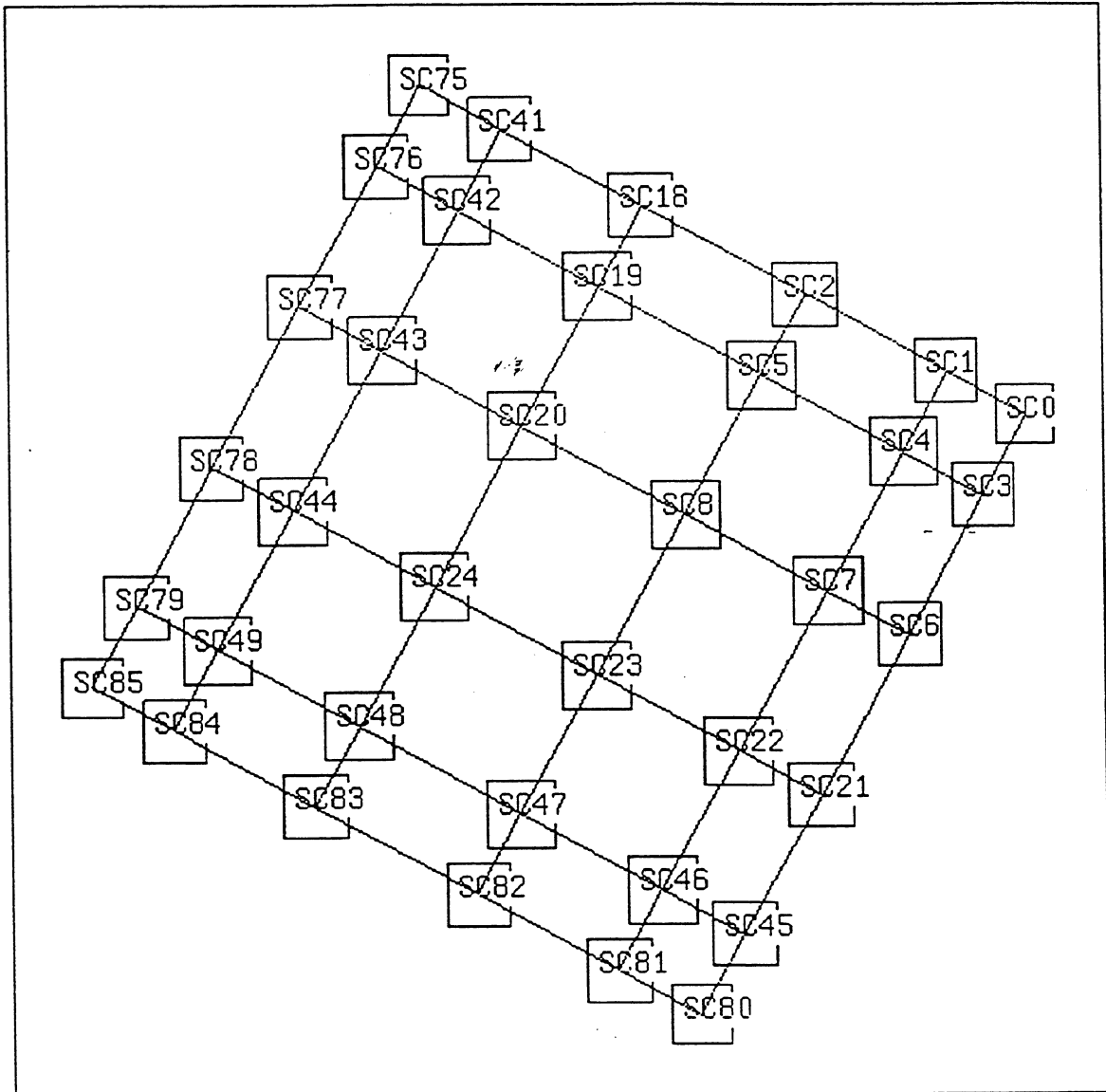
Surface du circuit: 38707340

Longueur estimée des connexions: 99835

Nombre total de blocs: 25

Nombre total de connexions: 41

Analyse factorielle, representation du nuage. Circuit: Dp36



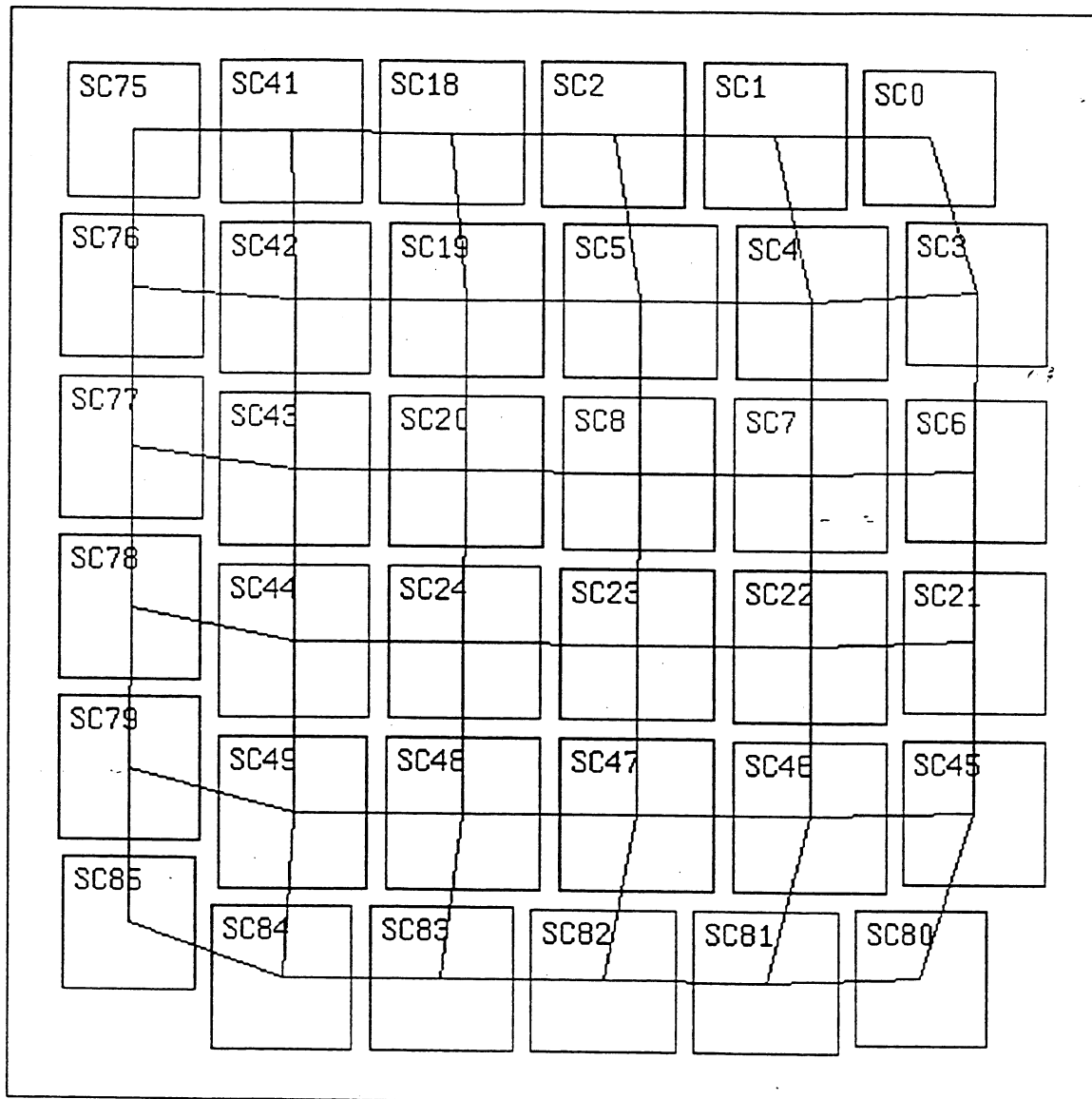
Circuit compose de 36 blocs et 60 connexions.

Projection du nuage sur le plan principal. Attention, pour visualisation les centres des blocs ont subis une homotethie de rapport 365 .

Qualite de representation du nuage sur les axes: 32% et 32%

Qualite de representation du nuage sur le plan: 64%

Placement par analyse factorielle. Circuit: Dp36



Placement obtenu apres decompaction et orientation des blocs.

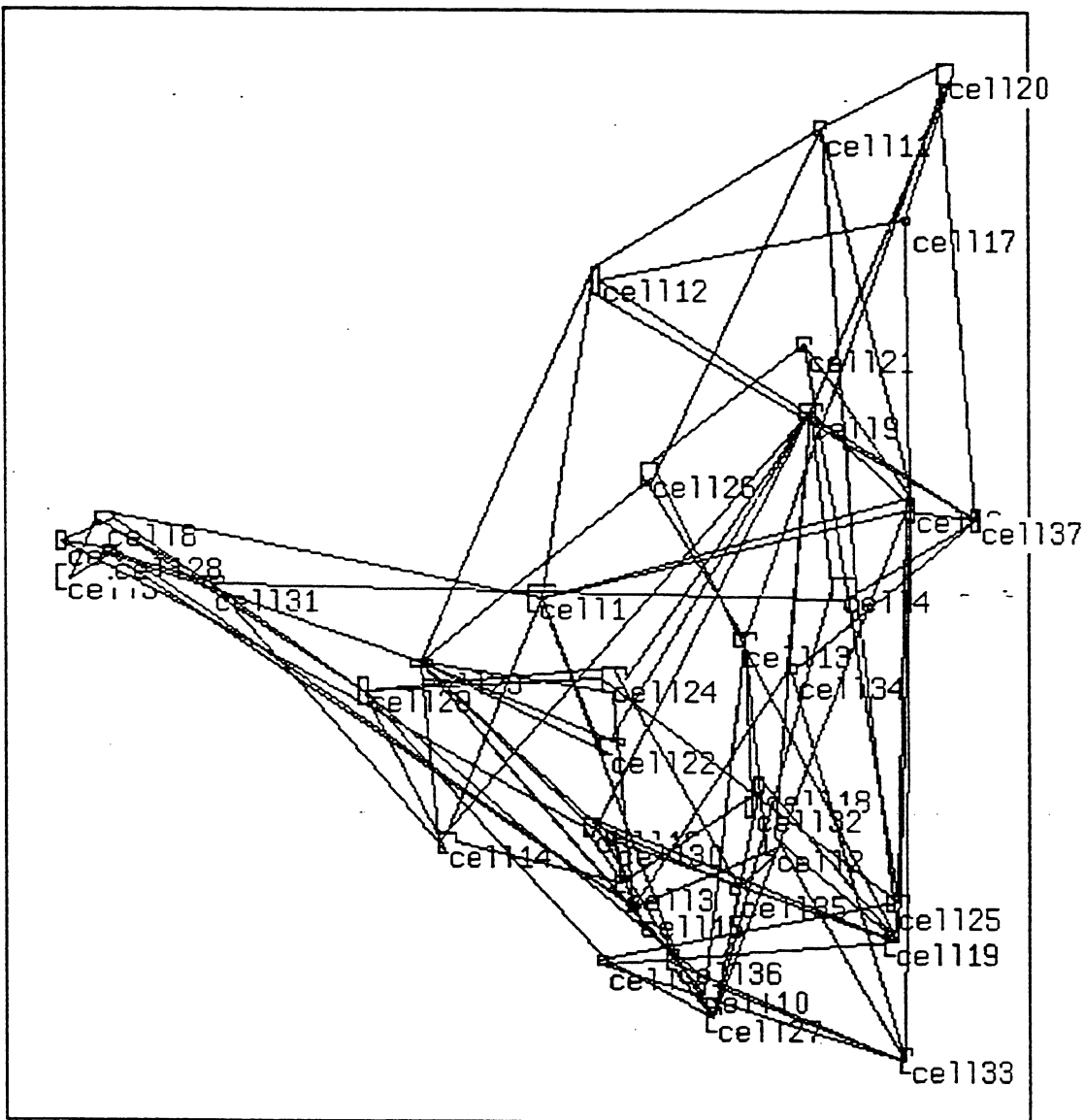
Dimensions du circuit: 204*204

Surface du circuit: 41750

Longueur estimee des connexions: 15752

Nombre total de blocs: 36

Nombre total de connexions: 60

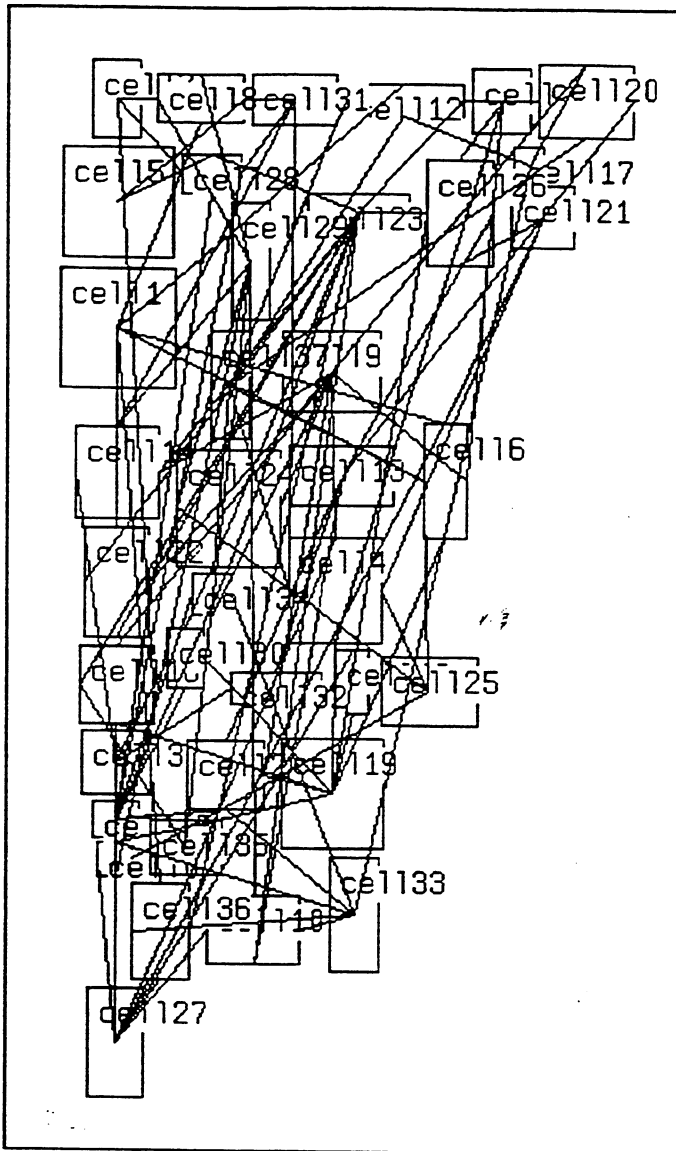


Circuit compose de 38 blocs et 103 connexions.

Projection du nuage sur le plan principal. Attention, pour visualisation les centres des blocs ont subis une homotethie de rapport 126644

Qualite de representation du nuage sur les axes: 8 % et 8 %

Qualite de representation du nuage sur le plan: 16 %



Placement obtenu apres decompaction et orientation des blocs.

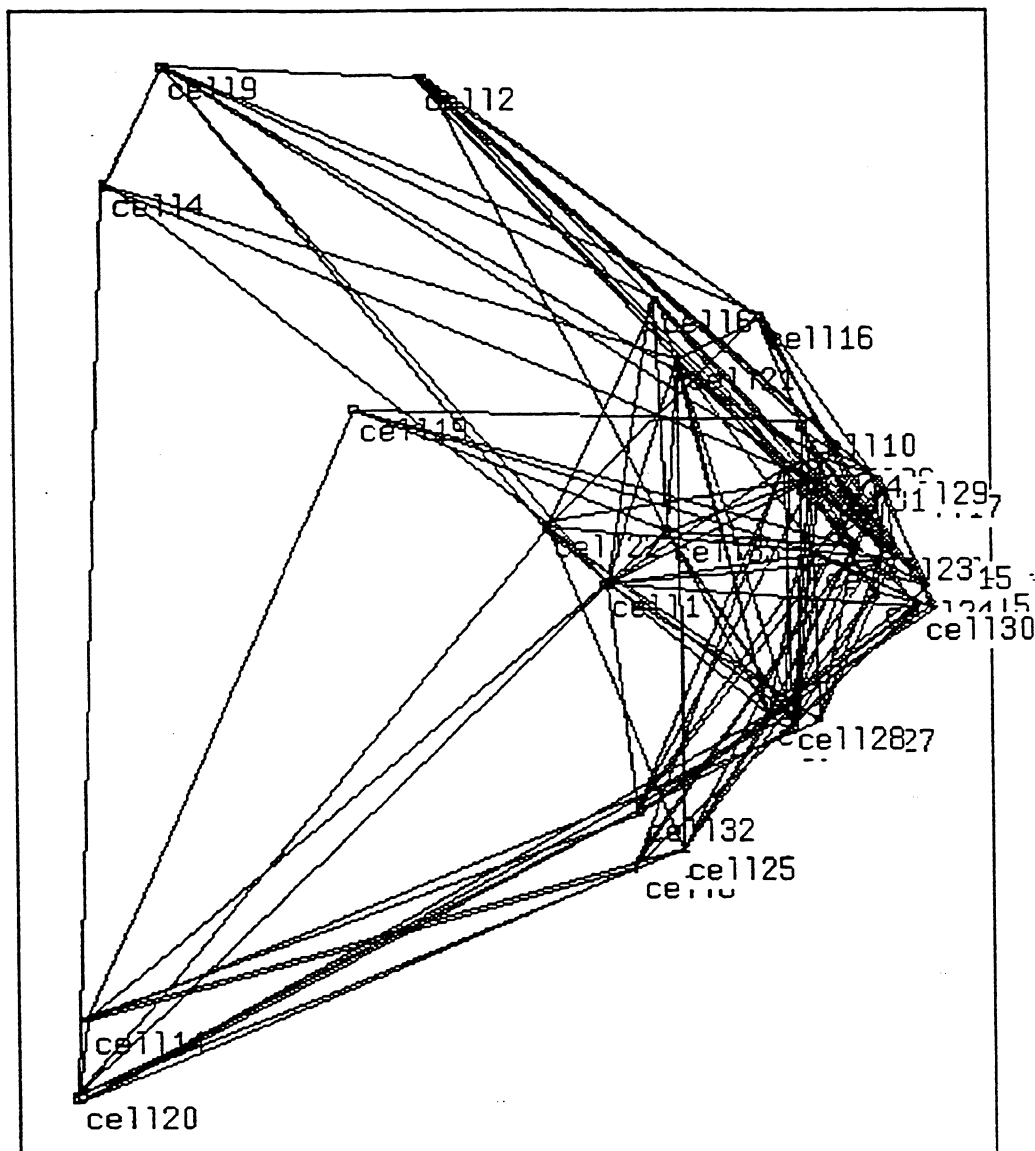
Dimensions du circuit: 11252*18832

Surface du circuit: 211906112

Longueur estimee des connexions: 6009952

Nombre total de blocs: 38

Nombre total de connexions: 103



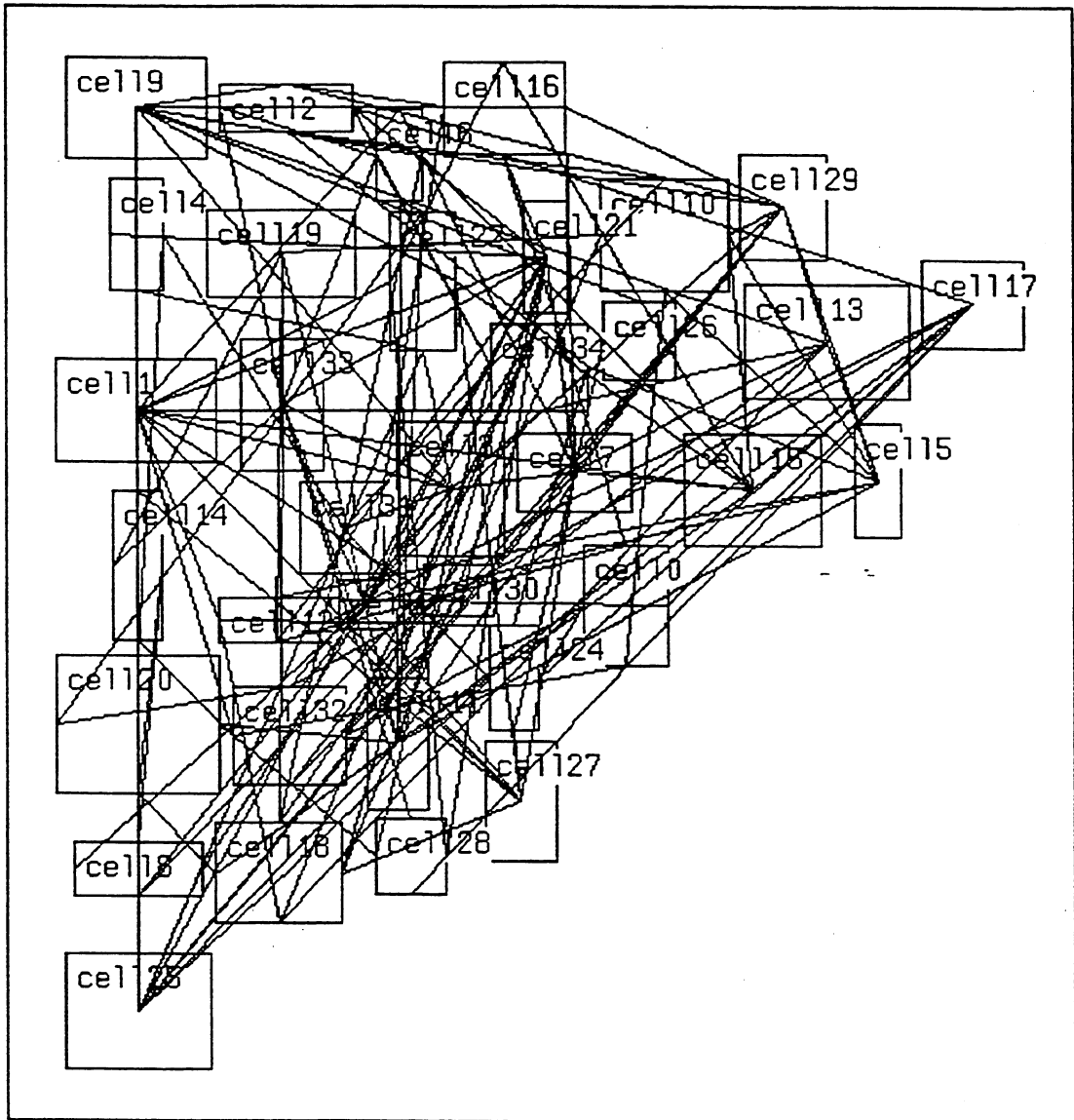
Circuit compose de 35 blocs et 161 connexions.

Projection du nuage sur le plan principal. Attention, pour visualisation les centres des blocs ont subis une homotethie de rapport 46157.

Certains blocs se superposent.

Qualite de representation du nuage sur les axes: 14% et 11%

Qualite de representation du nuage sur le plan: 25 %



Placement obtenu apres decompaction et orientation des blocs.

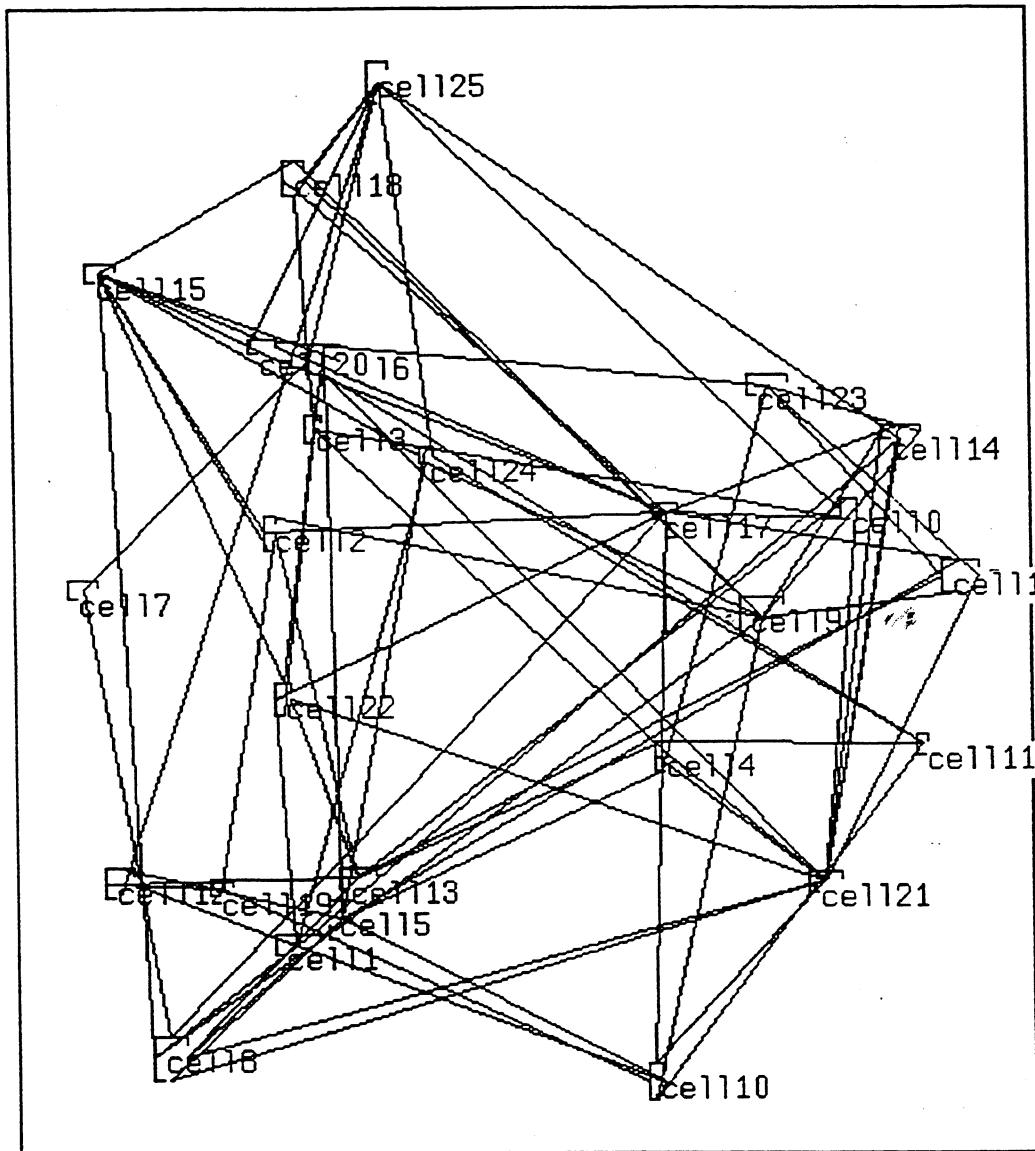
Dimensions du circuit: 13071*13481

Surface du circuit: 176214320

Longueur estimee des connexions: 8650483

Nombre total de blocs: 35

Nombre total de connexions: 161

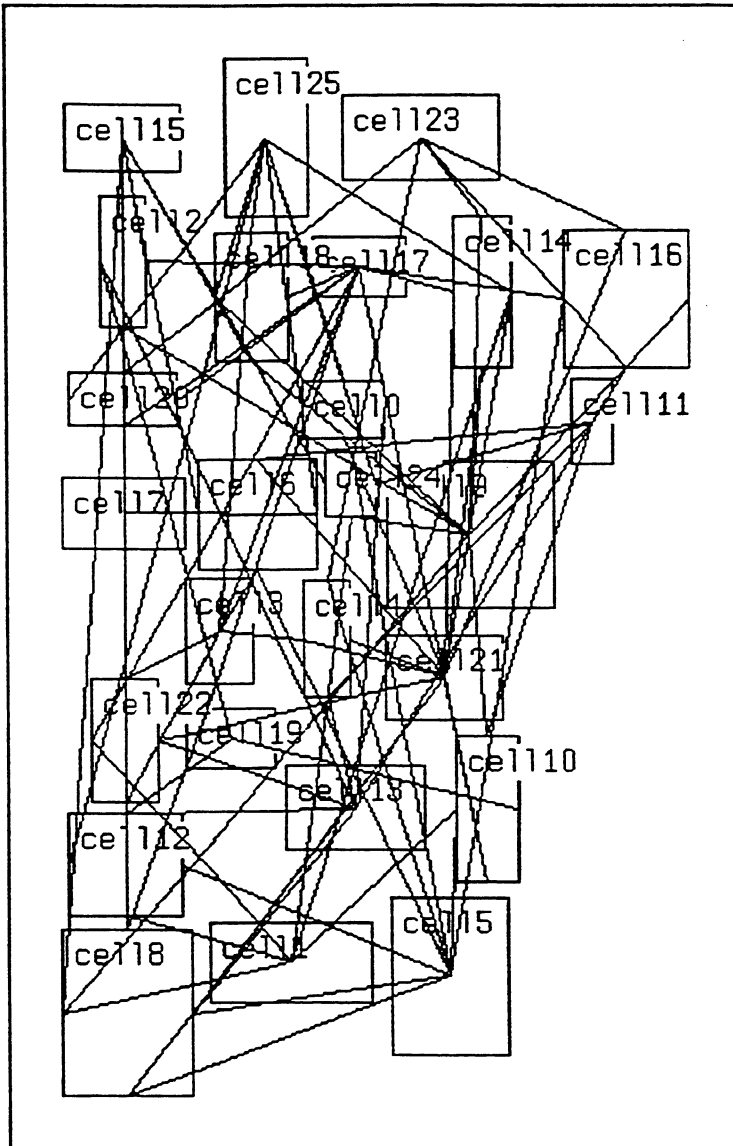


Circuit compose de 26 blocs et 93 connexions.

Projection du nuage sur le plan principal. Attention, pour visualisation les centres des blocs ont subis une homotethie de rapport 67215.

Qualite de representation du nuage sur les axes: 9 % et 8 %

Qualite de representation du nuage sur le plan: 17 %



Placement obtenu apres decompaction et orientation des blocs.

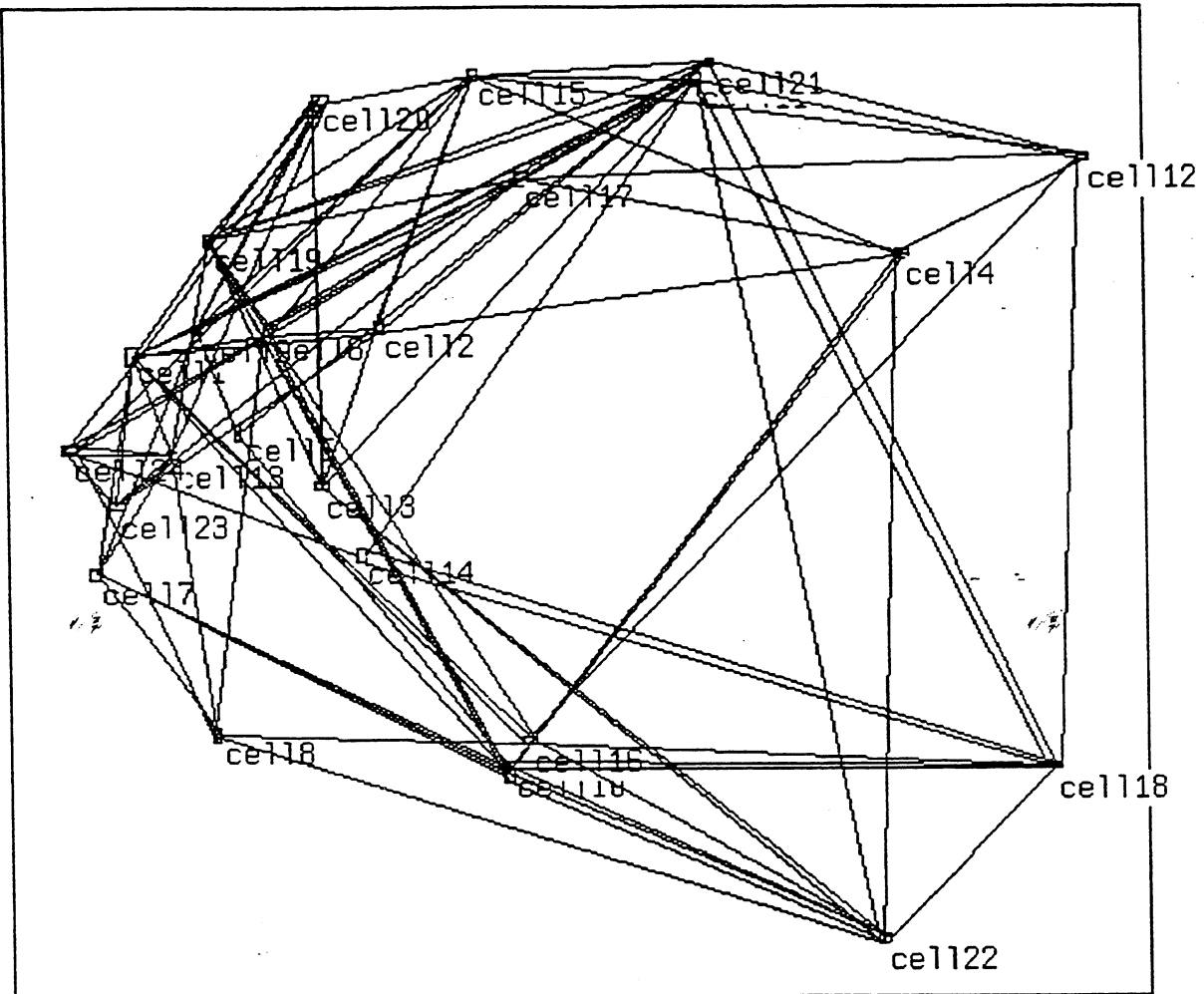
Dimensions du circuit: 8735*13509

Surface du circuit: 118005600

Longueur estimee des connexions: 4370413

Nombre total de blocs: 26

Nombre total de connexions: 93



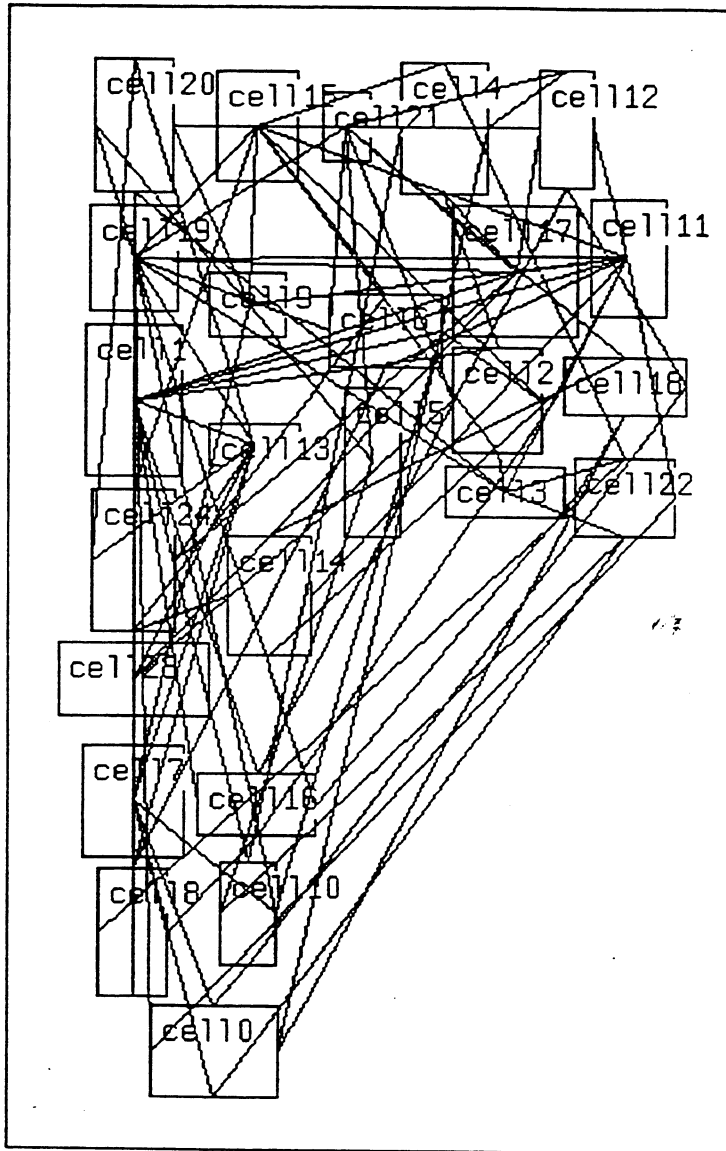
Circuit compose de 25 blocs et 102 connexions.

Projection du nuage sur le plan principal. Attention, pour visualisation les centres des blocs ont subis une homotethie de rapport 56799.

Certains blocs se superposent.

Qualite de representation du nuage sur les axes: 14% et 10%

Qualite de representation du nuage sur le plan: 24 %



Placement obtenu apres decompaction et orientation des blocs.

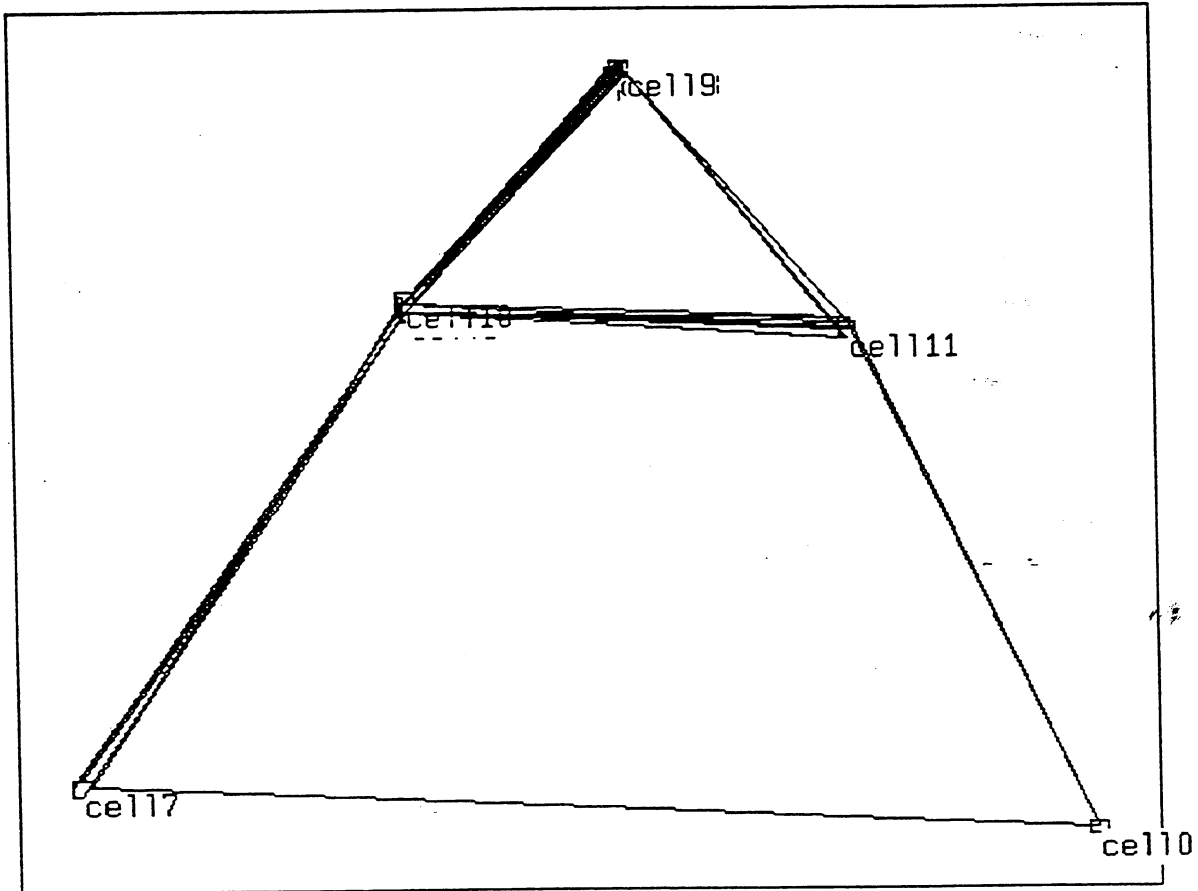
Dimensions du circuit: 9418*14651

Surface du circuit: 137992224

Longueur estimee des connexions: 5230919

Nombre total de blocs: 25

Nombre total de connexions: 102



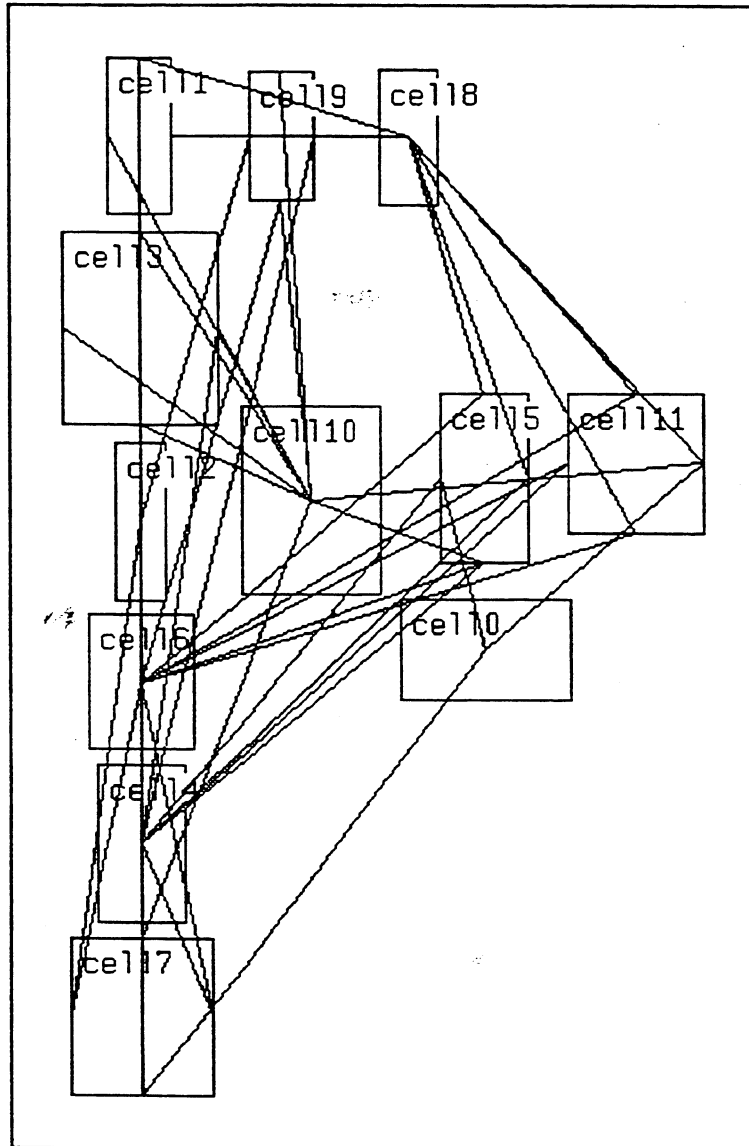
Circuit compose de 12 blocs et 51 connexions.

Projection du nuage sur le plan principal. Attention, pour visualisation les centres des blocs ont subis une homotethie de rapport 12775.

Certains blocs se superposent.

Qualite de representation du nuage sur les axes: 36% et 32%

Qualite de representation du nuage sur le plan: 68 %



Placement obtenu apres decompaction et orientation des blocs.

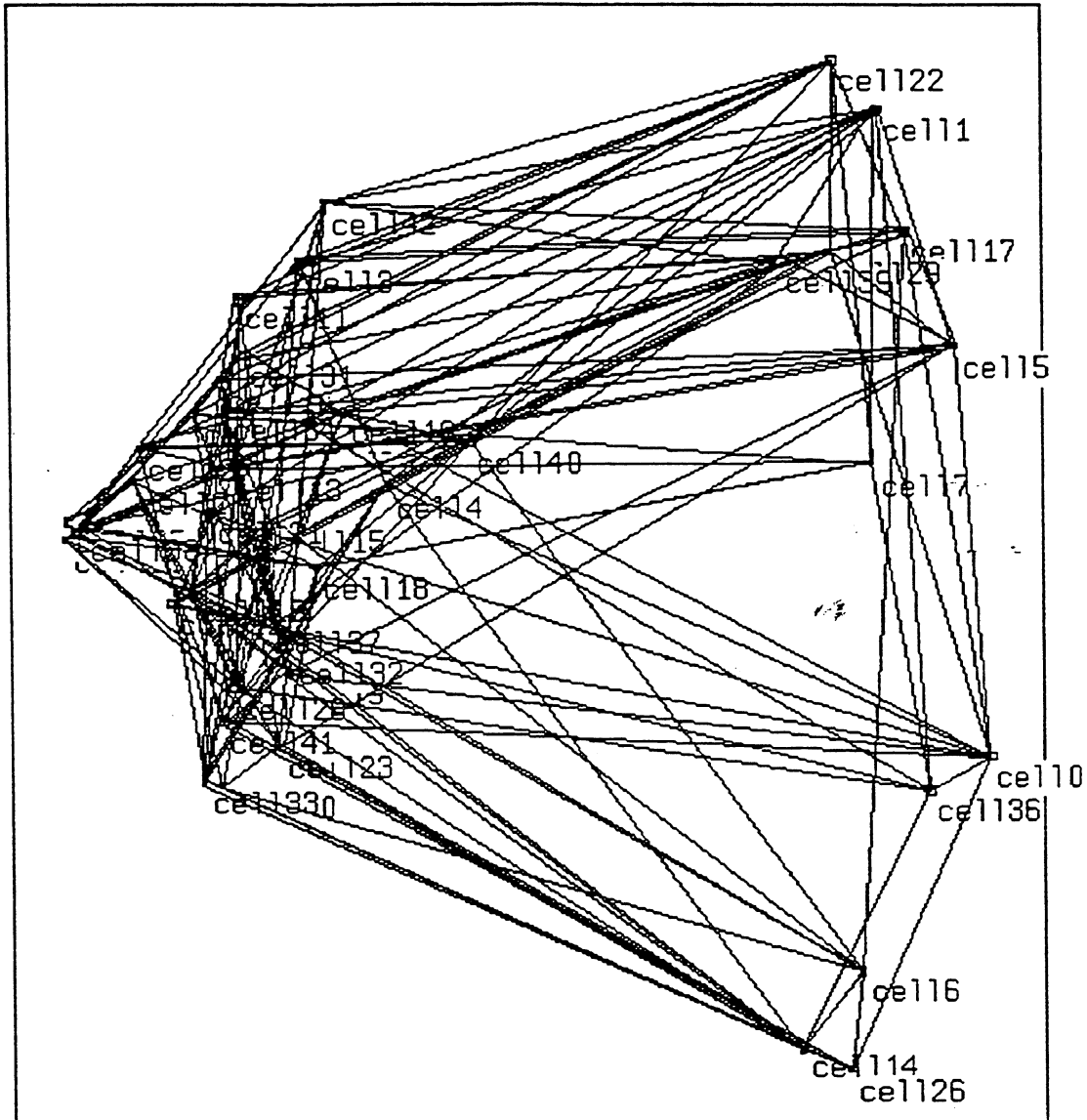
Dimensions du circuit: 7638*11533

Surface du circuit: 88092016

Longueur estimee des connexions: 2294832

Nombre total de blocs: 12

Nombre total de connexions: 51



Circuit compose de 43 blocs et 193 connexions.

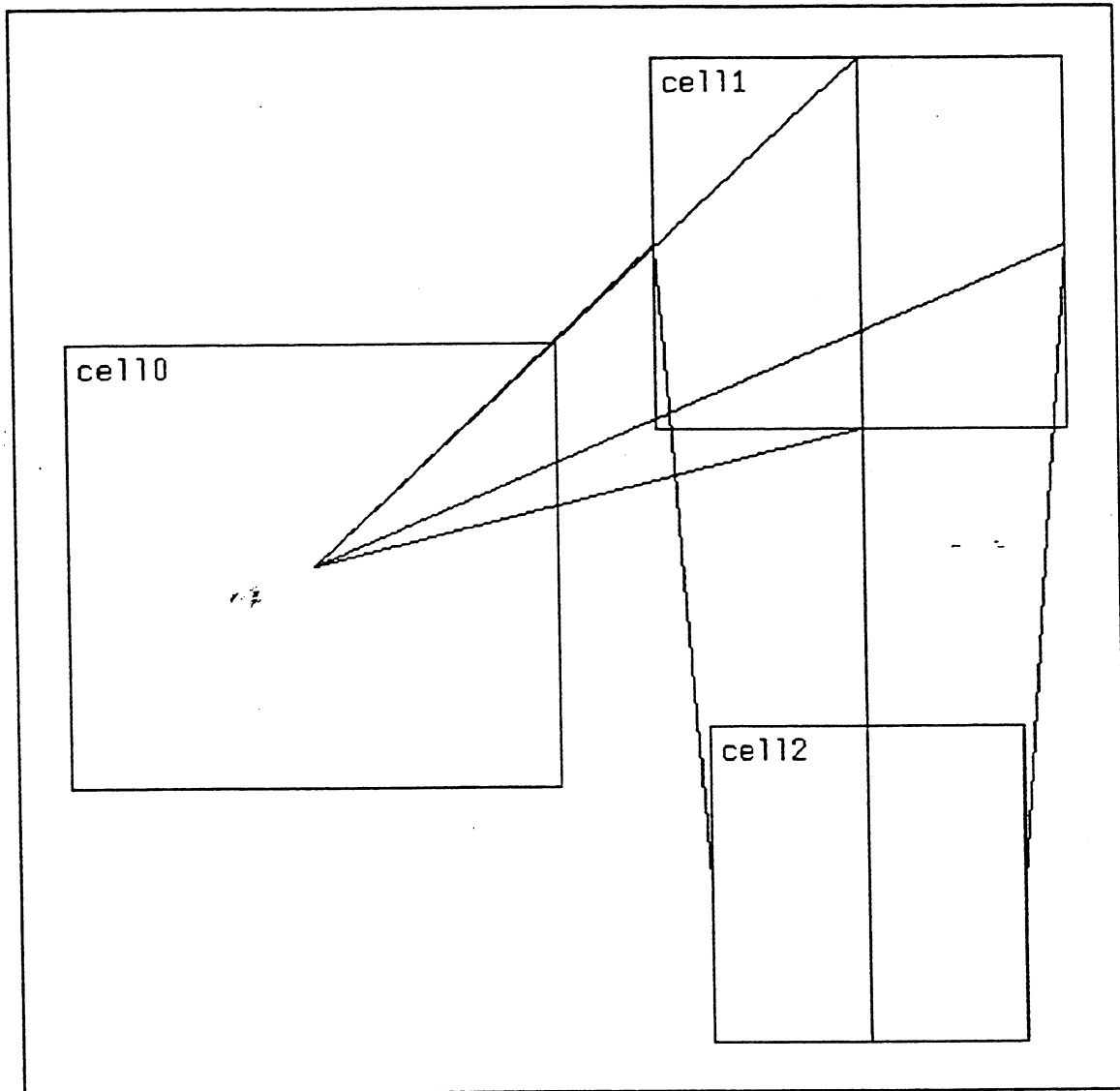
Projection du nuage sur le plan principal. Attention, pour visualisation

les centres des blocs ont subis une homotethie de rapport 115282

Certains blocs se superposent.

Qualite de representation du nuage sur les axes: 12% et 7 %

Qualite de representation du nuage sur le plan: 19 %

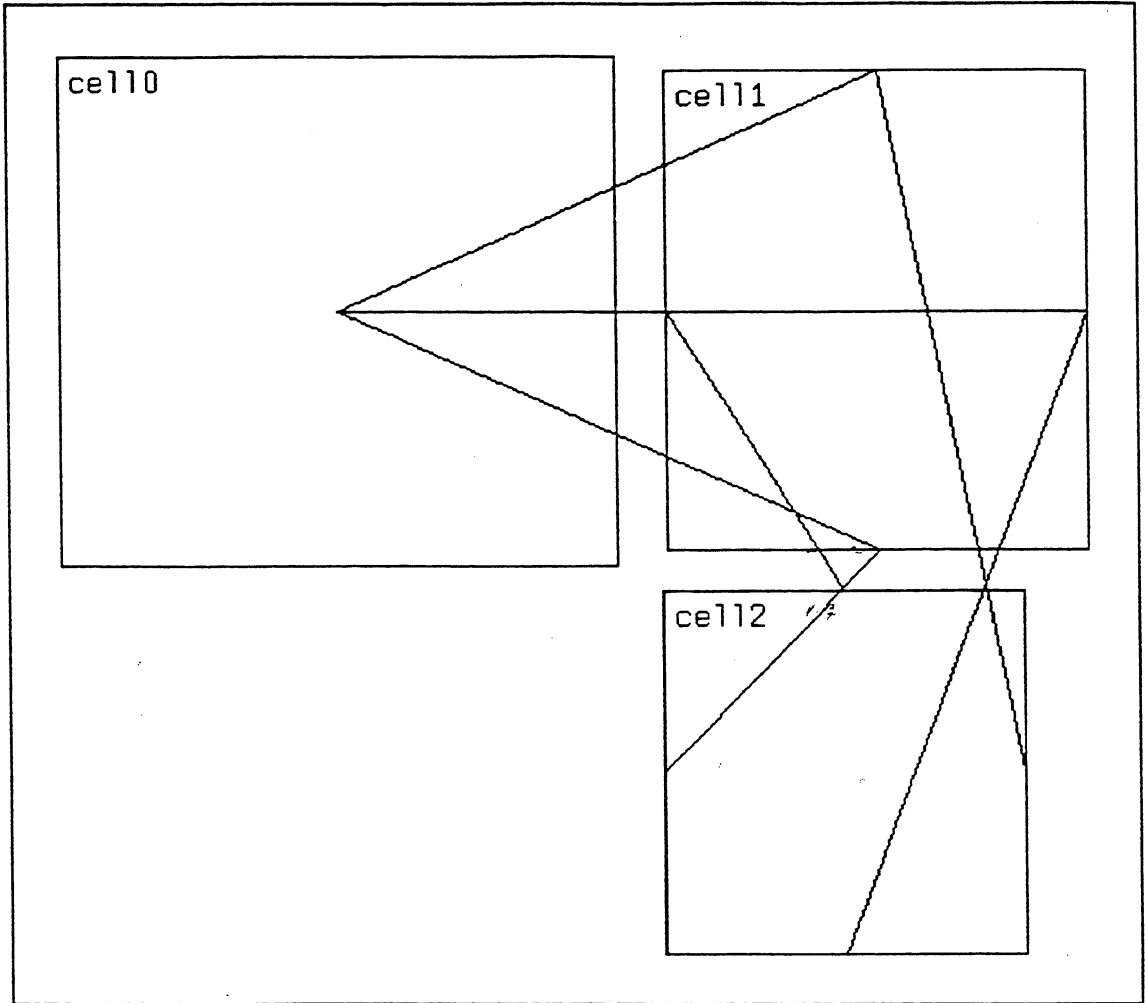


Circuit compose de 3 blocs et 4 connexions.

Projection du nuage sur le plan principal. Attention, pour visualisation les centres des blocs ont subis une homotethie de rapport 3660 .

Qualite de representation du nuage sur les axes: 50% et 50%

Qualite de representation du nuage sur le plan: 100%



Placement obtenu apres decompaction et orientation des blocs.

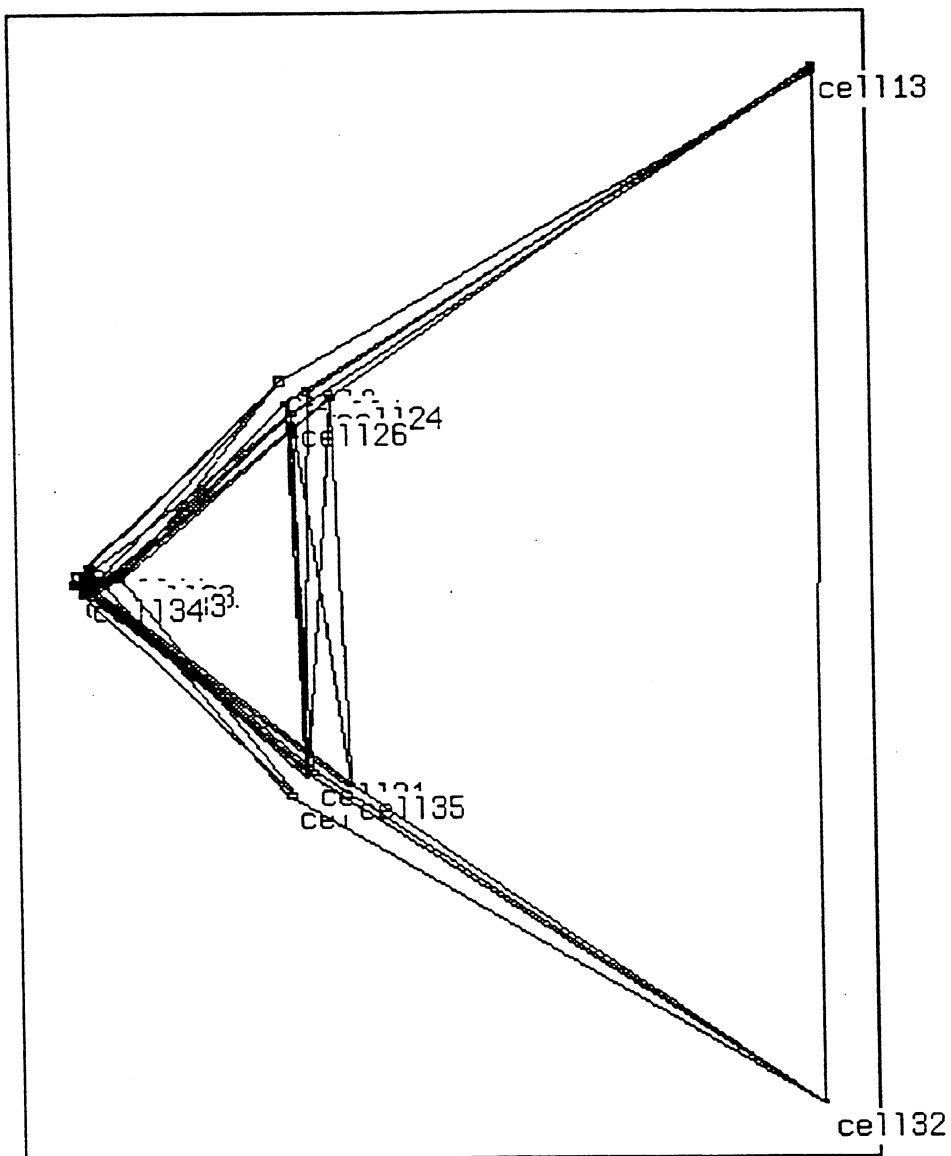
Dimensions du circuit: 2528*2211

Surface du circuit: 5590054

Longueur estimee des connexions: 63018

Nombre total de blocs: 3

Nombre total de connexions: 4



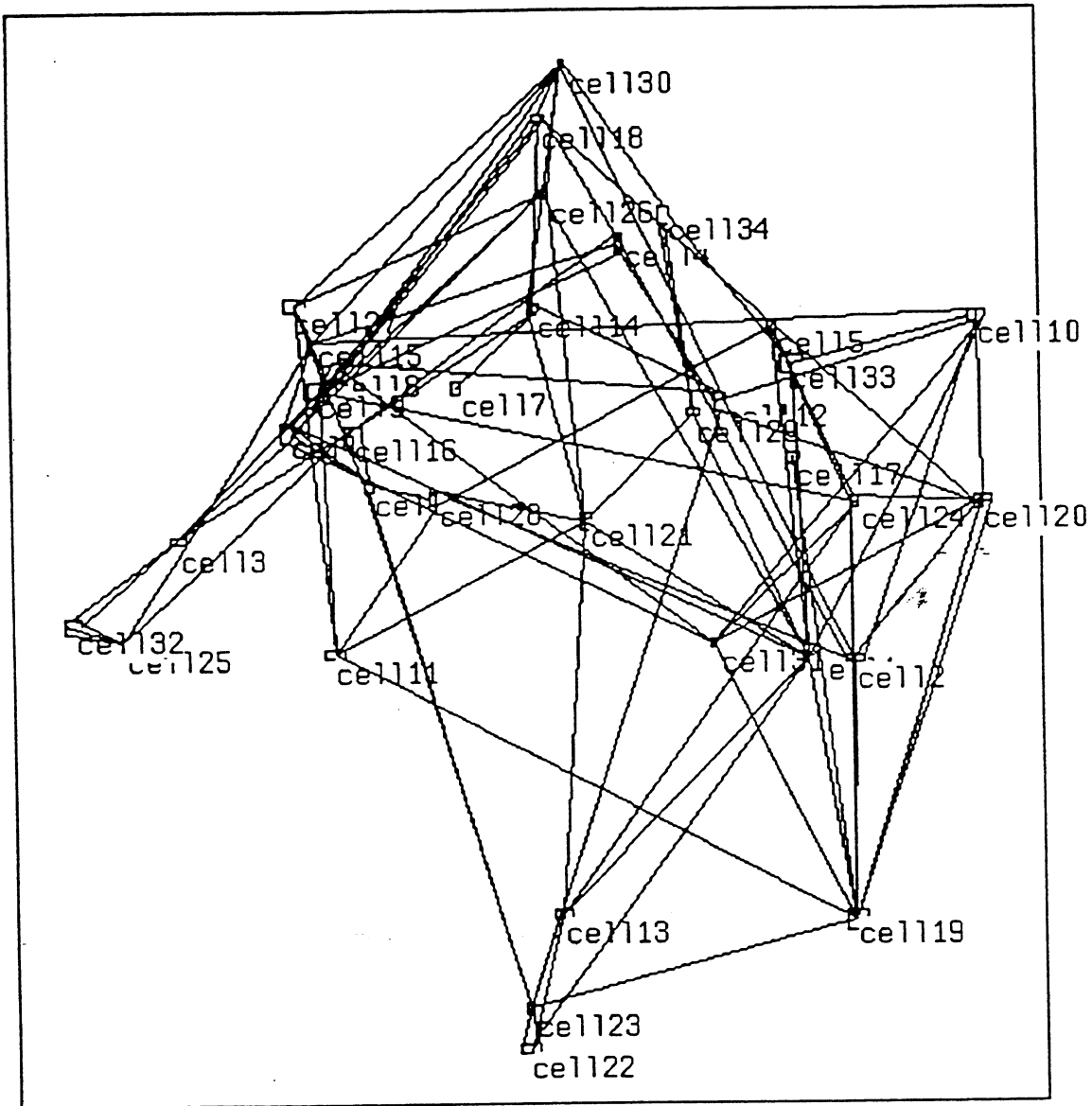
Circuit compose de 36 blocs et 98 connexions.

Projection du nuage sur le plan principal. Attention, pour visualisation les centres des blocs ont subis une homotethie de rapport 102876

Certains blocs se superposent.

Qualite de representation du nuage sur les axes: 25% et 19%

Qualite de representation du nuage sur le plan: 44 %

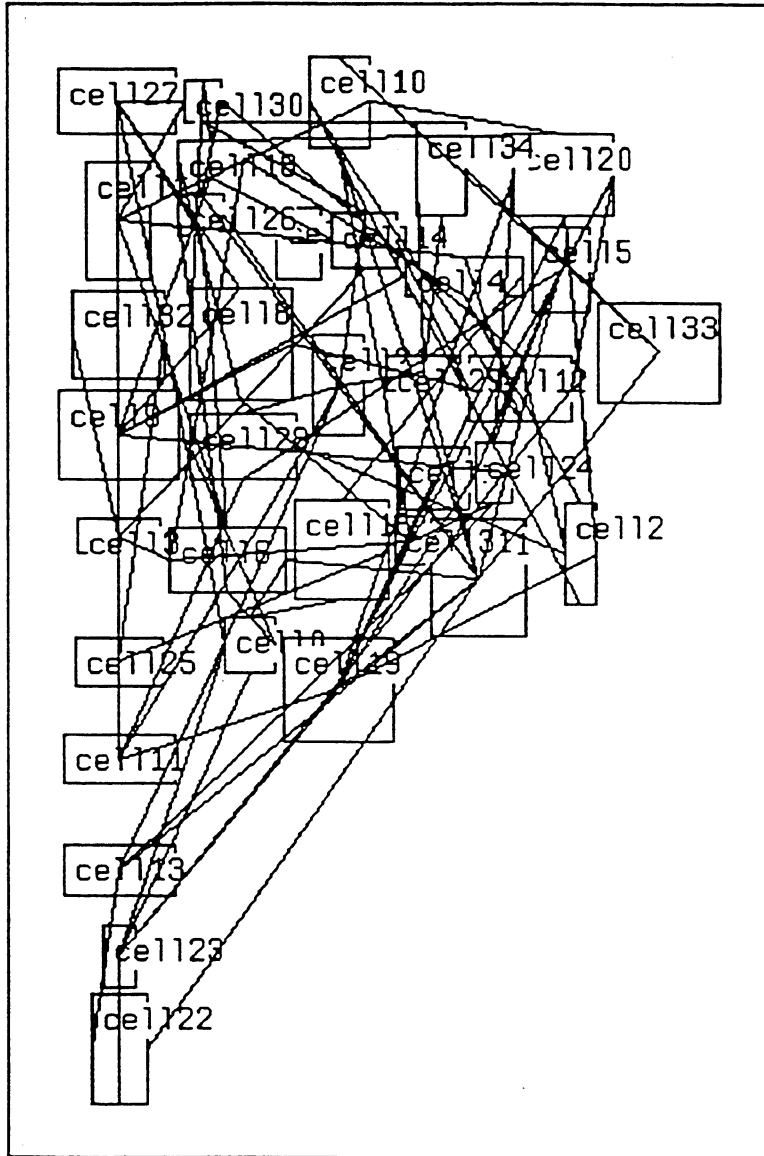


Circuit compose de 35 blocs et 104 connexions.

Projection du nuage sur le plan principal. Attention, pour visualisation les centres des blocs ont subis une homotethie de rapport 148546

Qualite de representation du nuage sur les axes: 9 % et 8 %

Qualite de representation du nuage sur le plan: 17 %



Placement obtenu apres decompaction et orientation des blocs.

Dimensions du circuit: 12647*18787

Surface du circuit: 237608784

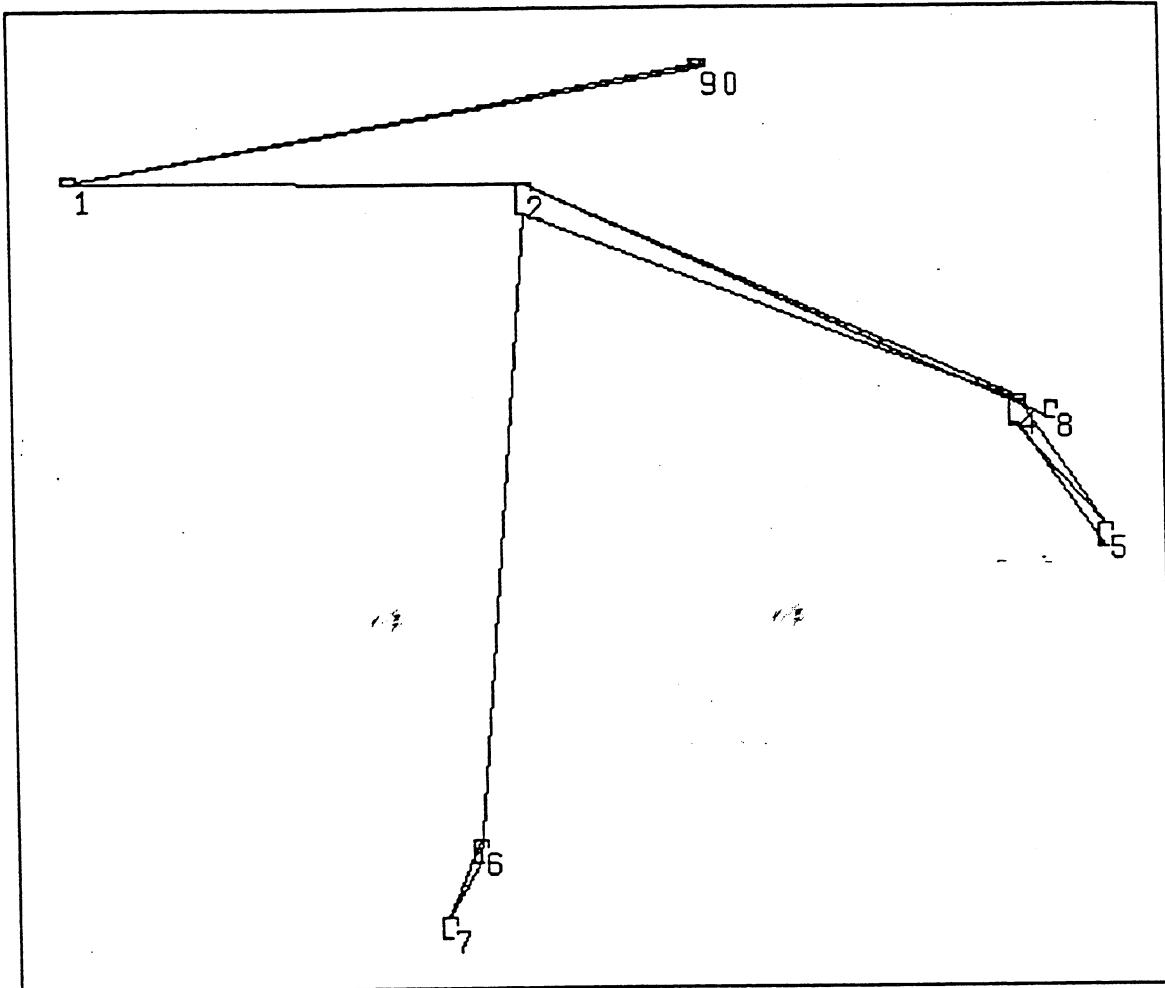
Longueur estimee des connexions: 5941700

Nombre total de blocs: 35

Nombre total de connexions: 104

Placement obtenu pour le circuit Xerox

(Benchmark MCNC)



Circuit compose de 10 blocs et 23 connexions.

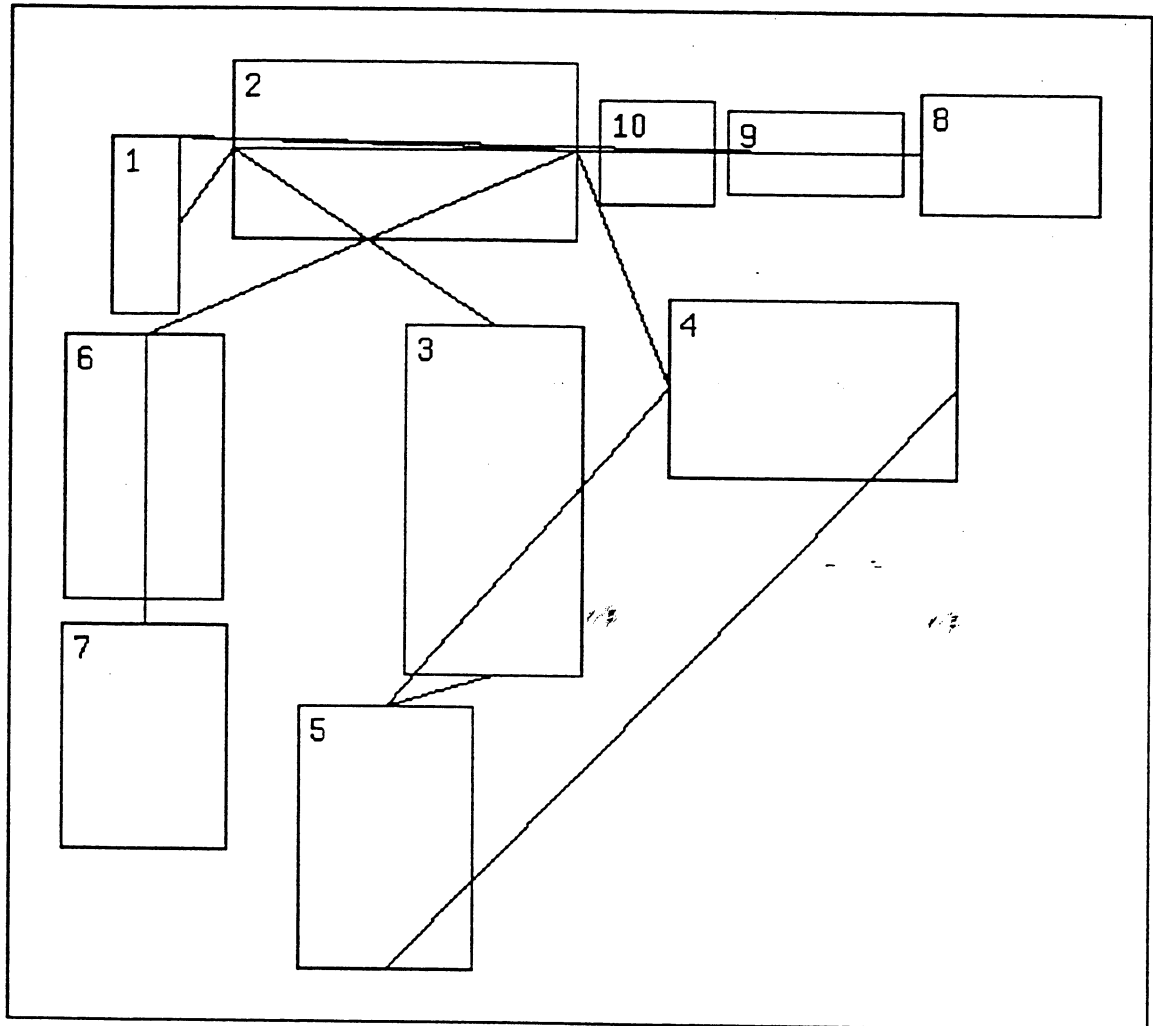
Projection du nuage sur le plan principal. Attention, pour visualisation les centres des blocs ont subis une homotethie de rapport 22508.

Certains blocs se superposent.

Qualite de representation du nuage sur les axes: 30% et 24%

Qualite de representation du nuage sur le plan: 54 %

Placement par analyse factorielle. Circuit: Xerox



Placement obtenu apres decompaction et orientation des blocs.

Dimensions du circuit: 10096*8897

Surface du circuit: 89823688

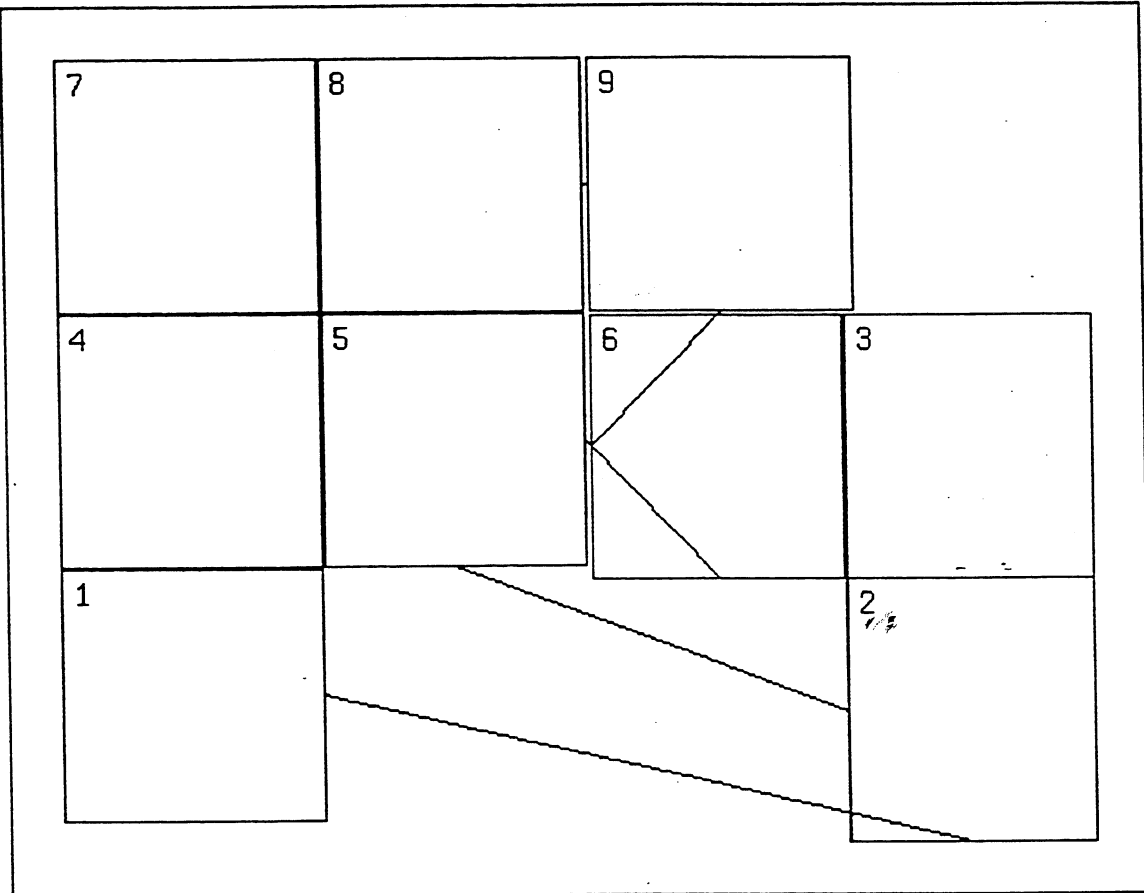
Longueur estimee des connexions: 454657

Nombre total de blocs: 10

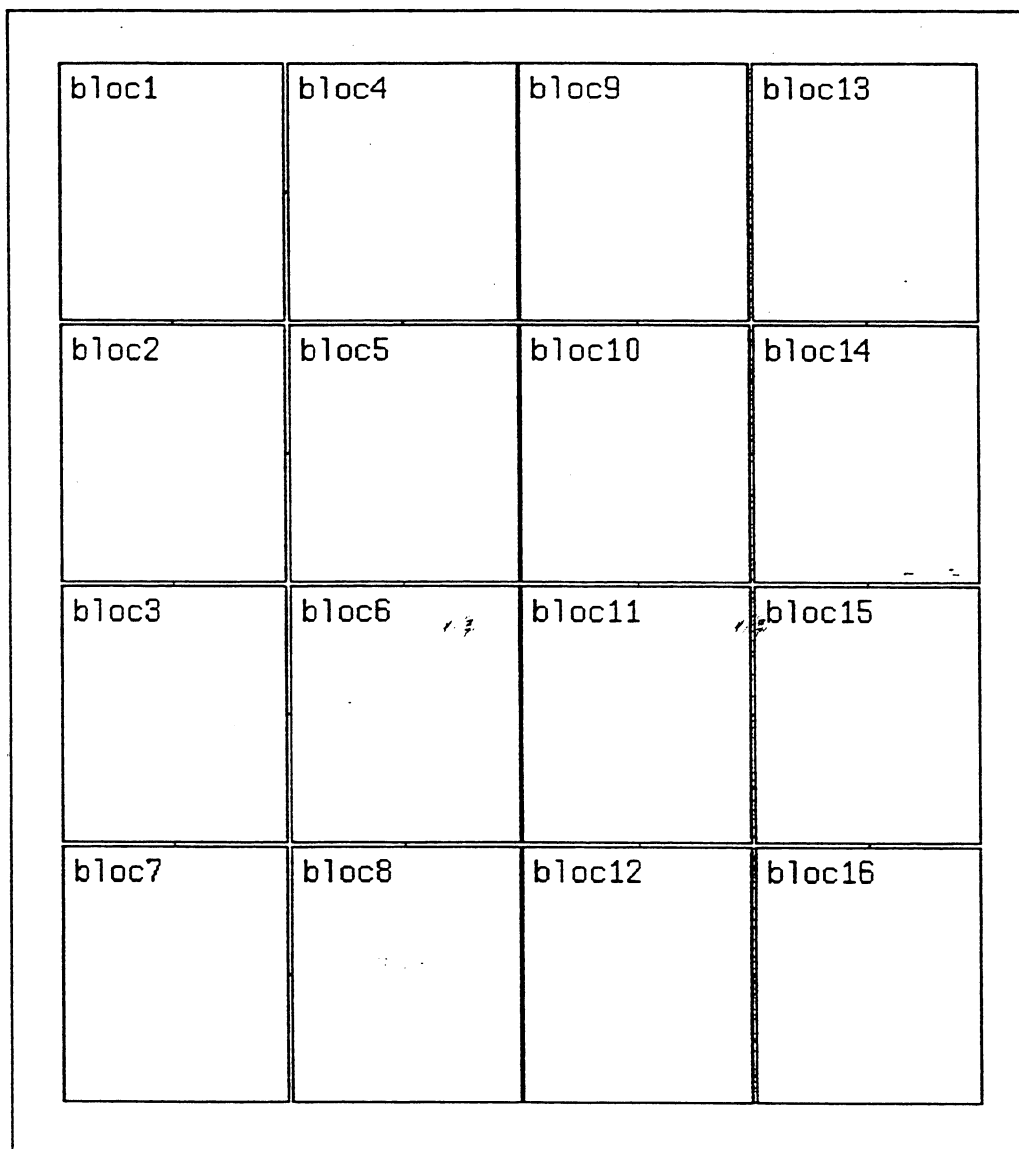
Nombre total de connexions: 23

Annexe II

Résultats du placement par bipartitionnements successifs

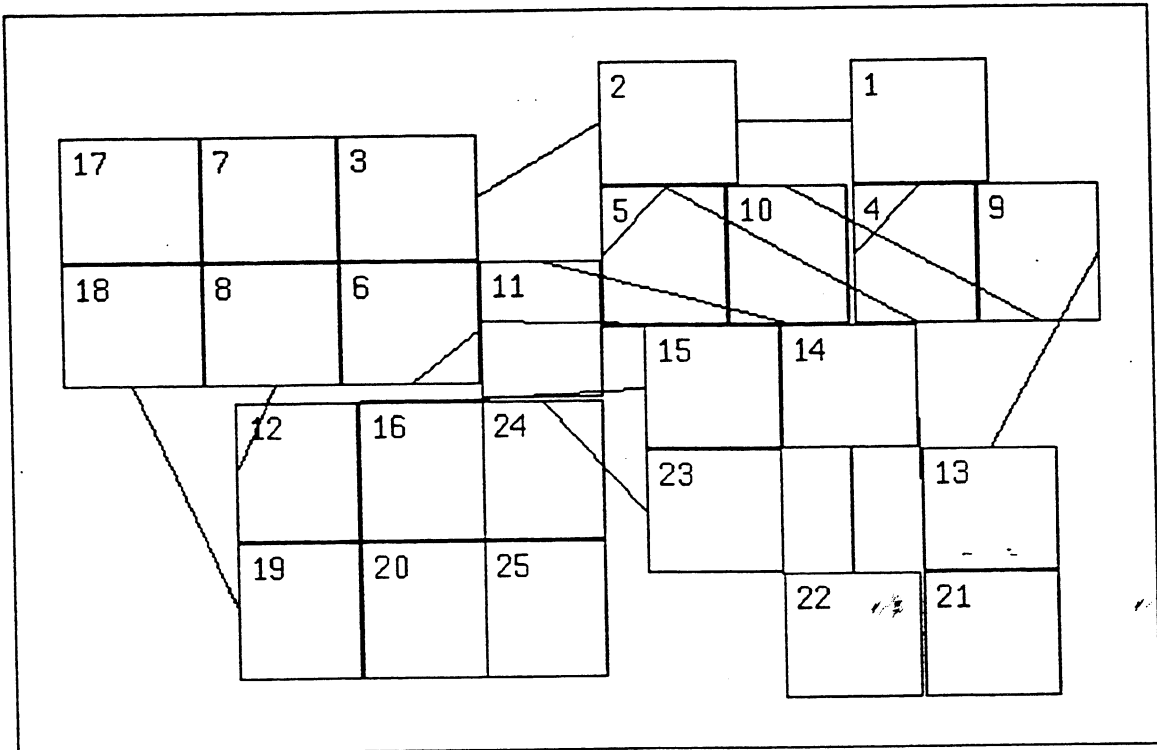


Dimensions du circuit:	9573*7424
Surface du circuit:	71075056
Longueur estimee des connexions:	253808
Nombre total de blocs:	9
Nombre total de connexions:	12



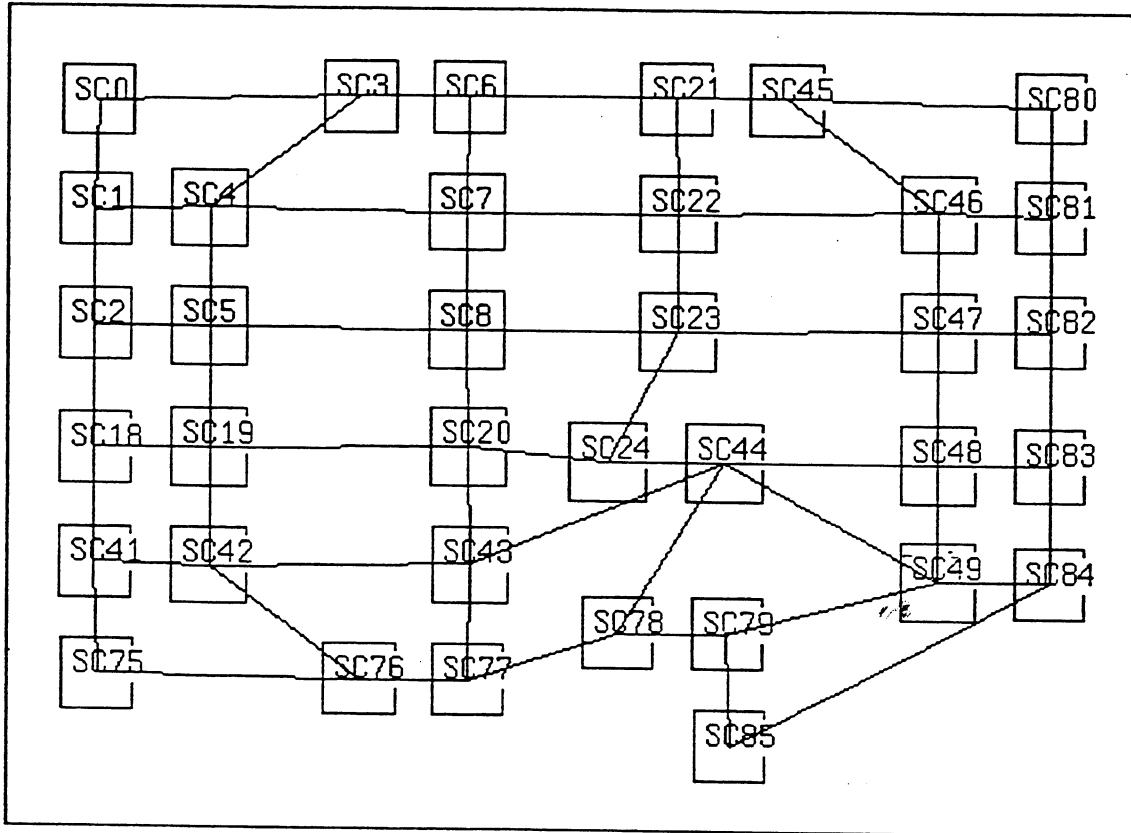
Dimensions du circuit: 4029*4445
Surface du circuit: 17909752
Longueur estimee des connexions: 2880
Nombre total de blocs: 16
Nombre total de connexions: 24

Placement par bipartitionnements successifs. Circuit: Dp25.res

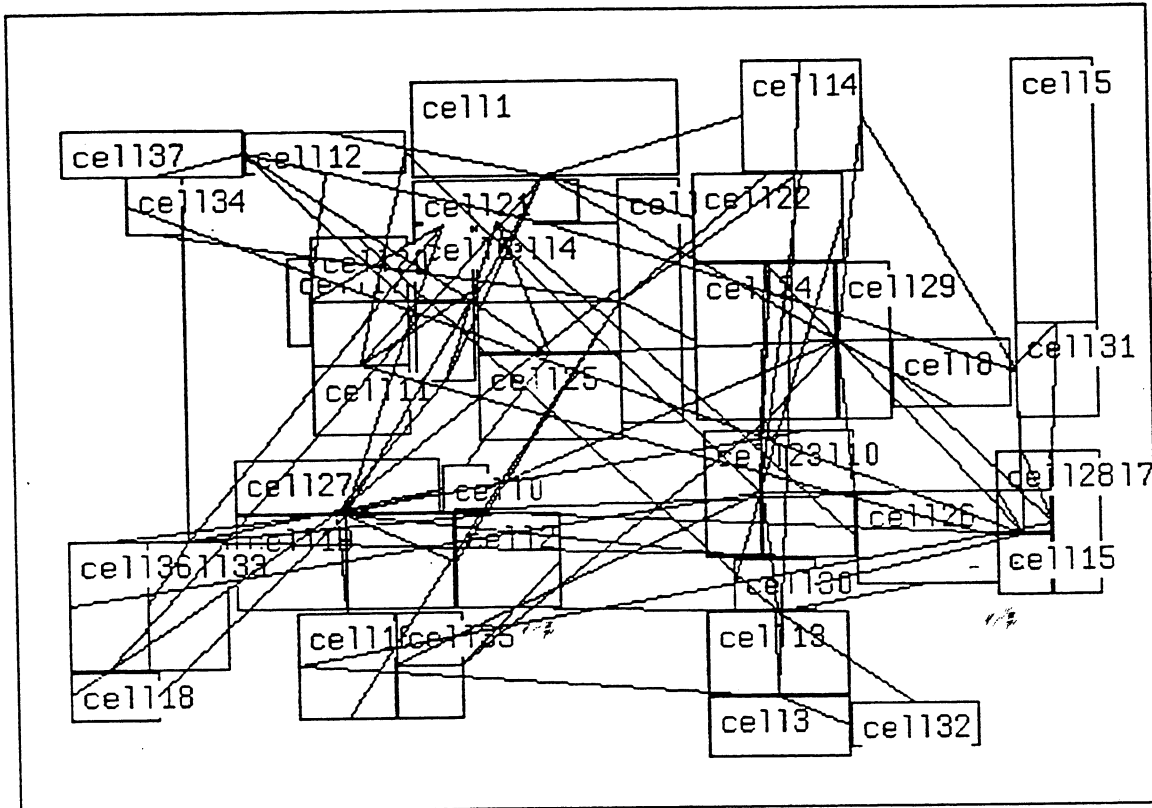


Dimensions du circuit:	8412*5422
Surface du circuit:	45605716
Longueur estimee des connexions:	240376
Nombre total de blocs:	25
Nombre total de connexions:	41

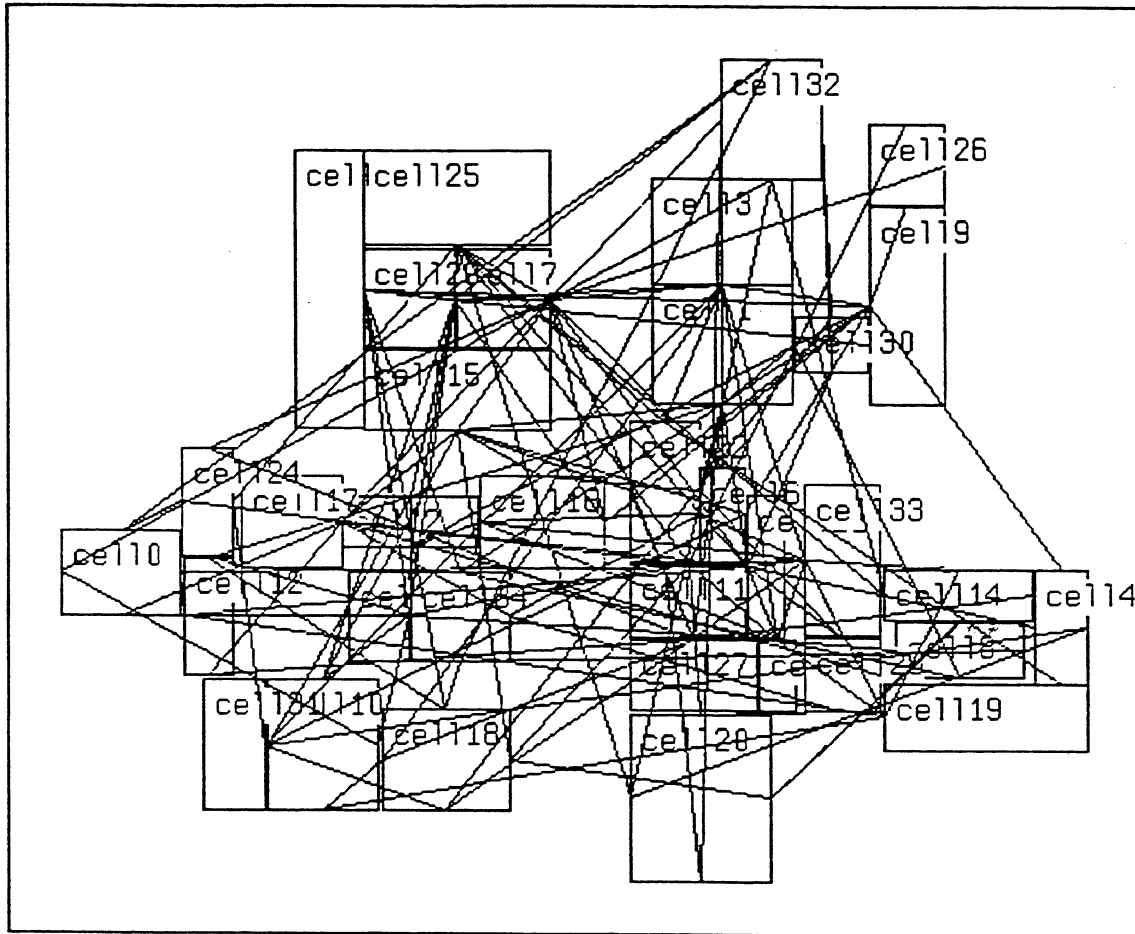
Placement par bipartitionnements successifs. Circuit: Dp36.res



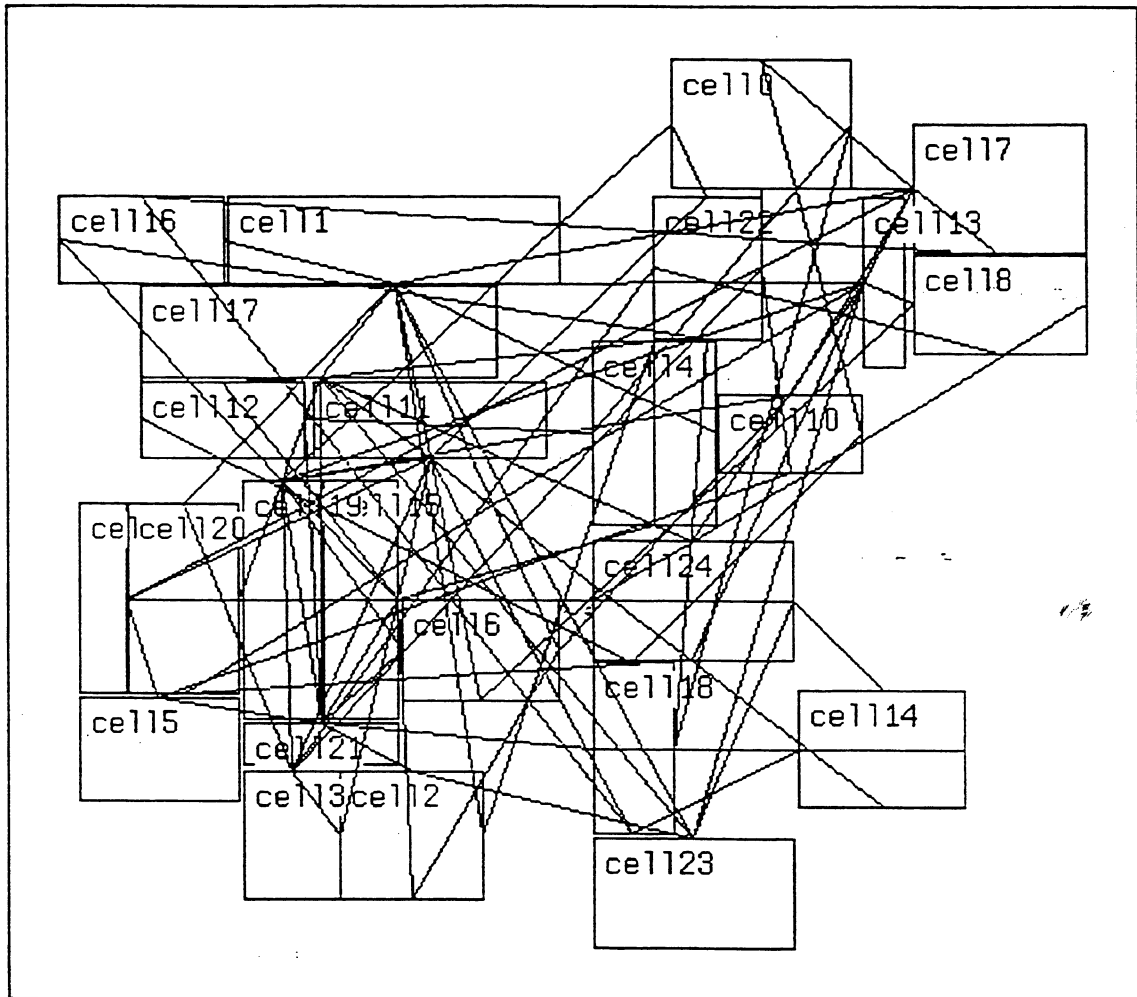
Dimensions du circuit:	429*308
Surface du circuit:	131918
Longueur estimee des connexions:	31992
Nombre total de blocs:	36
Nombre total de connexions:	60



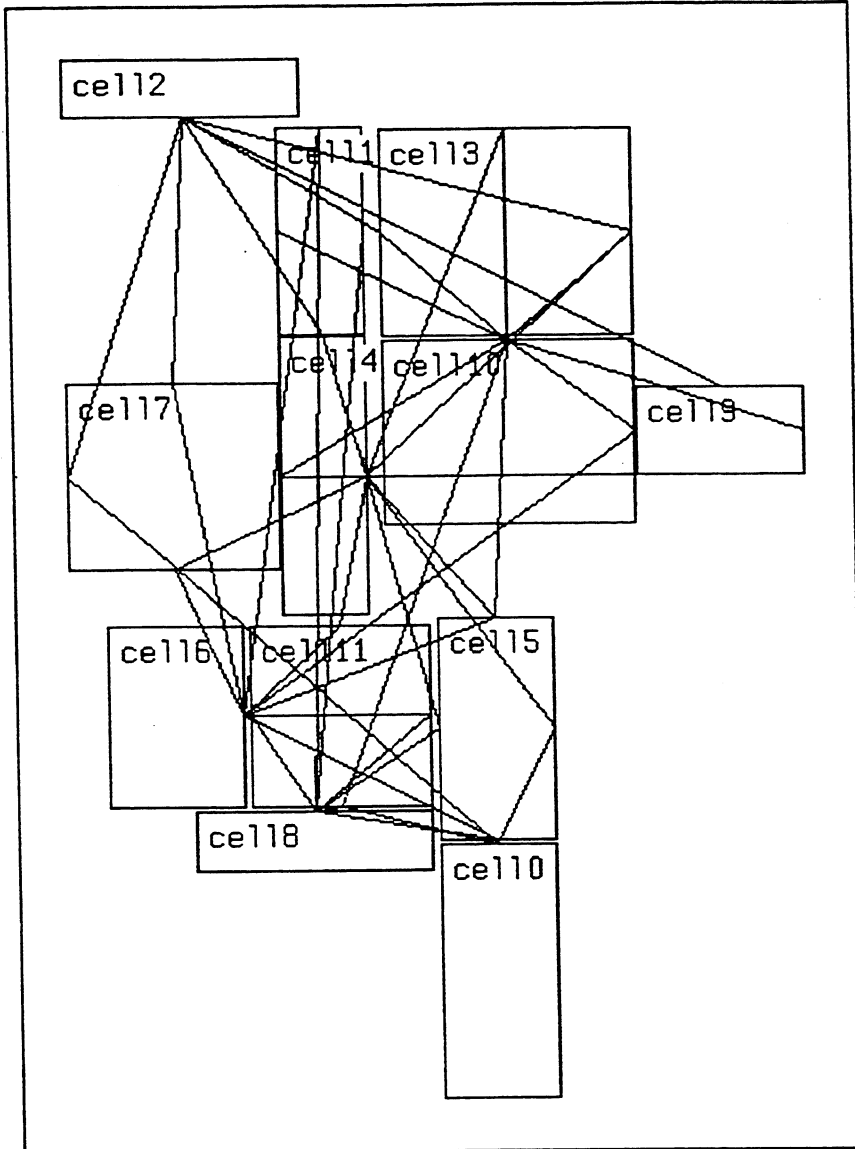
Dimensions du circuit:	13922*9699
Surface du circuit:	135020016
Longueur estimee des connexions:	3735692
Nombre total de blocs:	38
Nombre total de connexions:	103



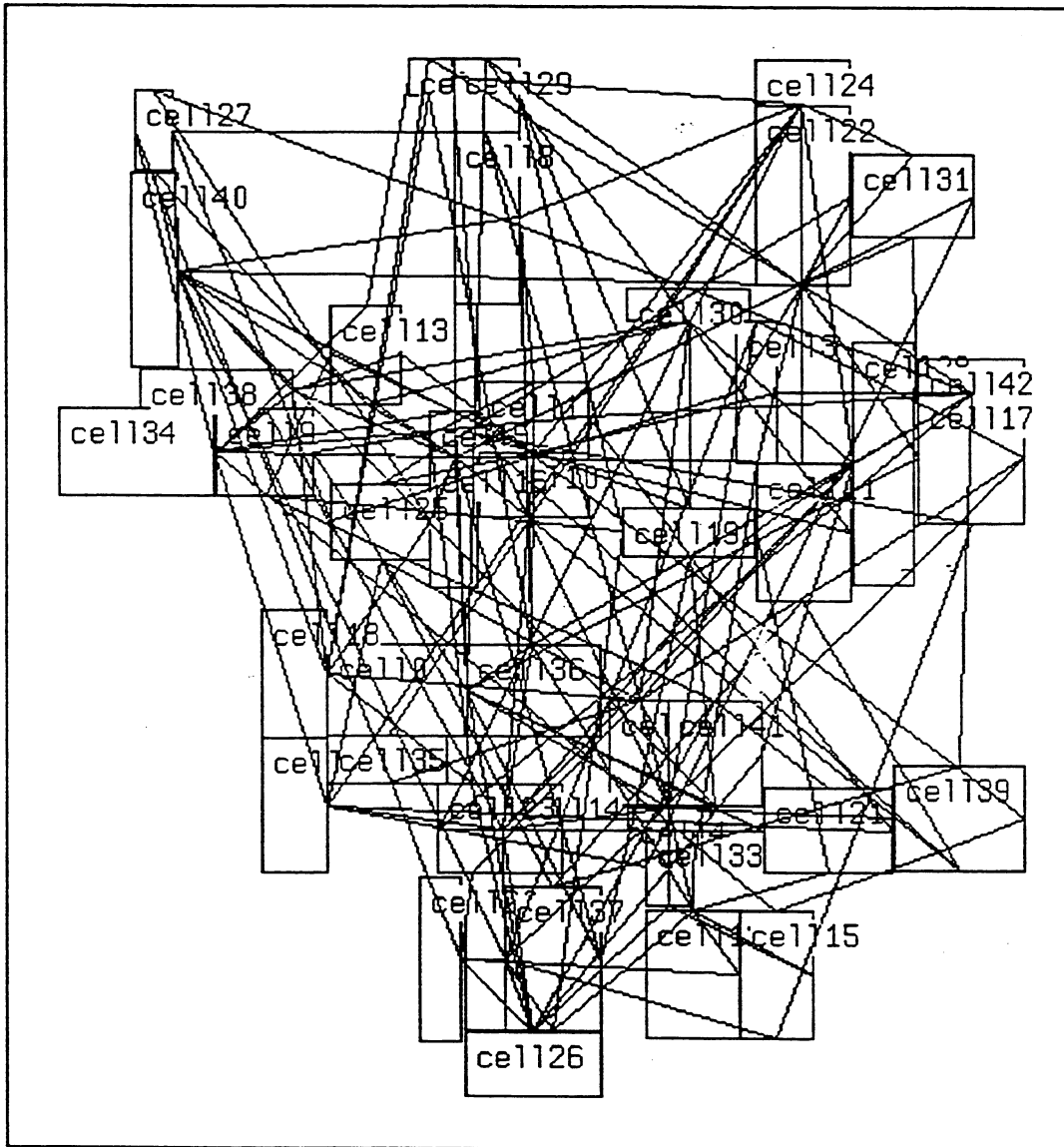
Dimensions du circuit:	13635*11032
Surface du circuit:	150422576
Longueur estimee des connexions:	5326409
Nombre total de blocs:	35
Nombre total de connexions:	161



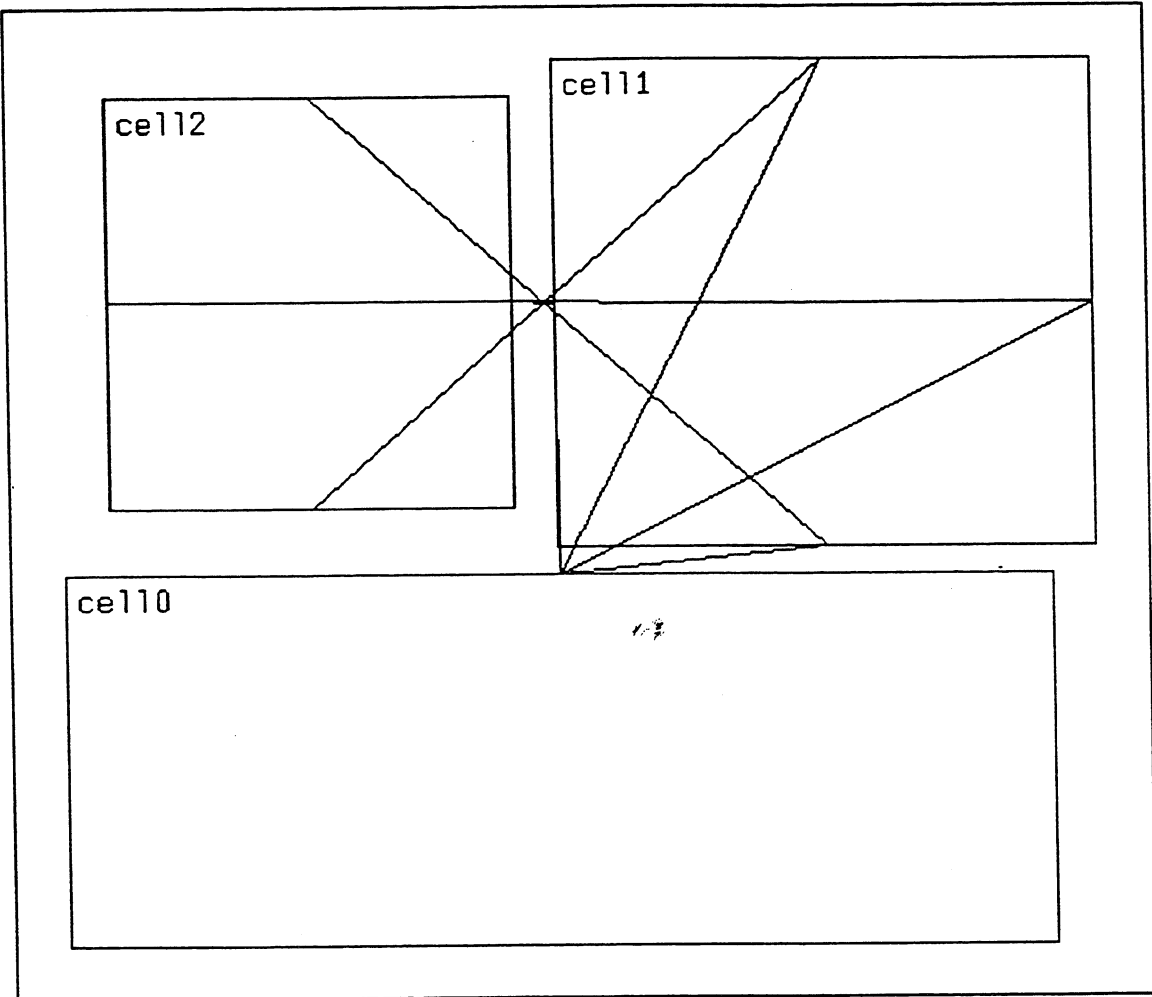
Dimensions du circuit:	10376*9005
Surface du circuit:	93436224
Longueur estimee des connexions:	3359640
Nombre total de blocs:	25
Nombre total de connexions:	102



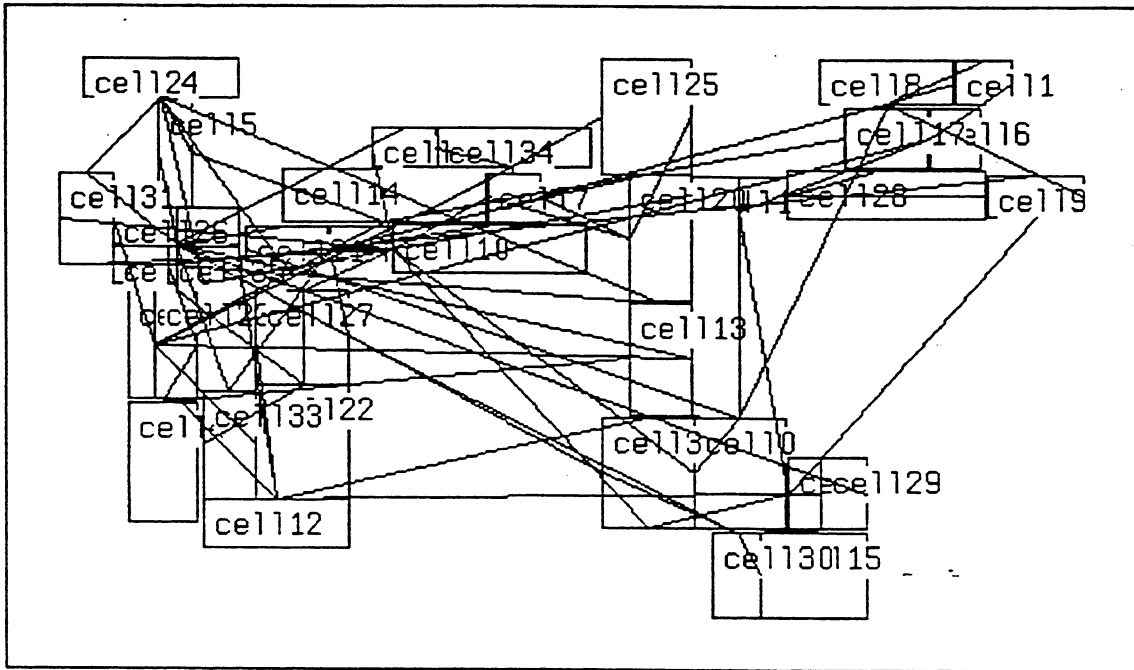
Dimensions du circuit:	6498*8819
Surface du circuit:	57305672
Longueur estimee des connexions:	1171044
Nombre total de blocs:	12
Nombre total de connexions:	51



Dimensions du circuit:	14976*15769
Surface du circuit:	236150224
Longueur estimee des connexions:	9415488
Nombre total de blocs:	43
Nombre total de connexions:	193



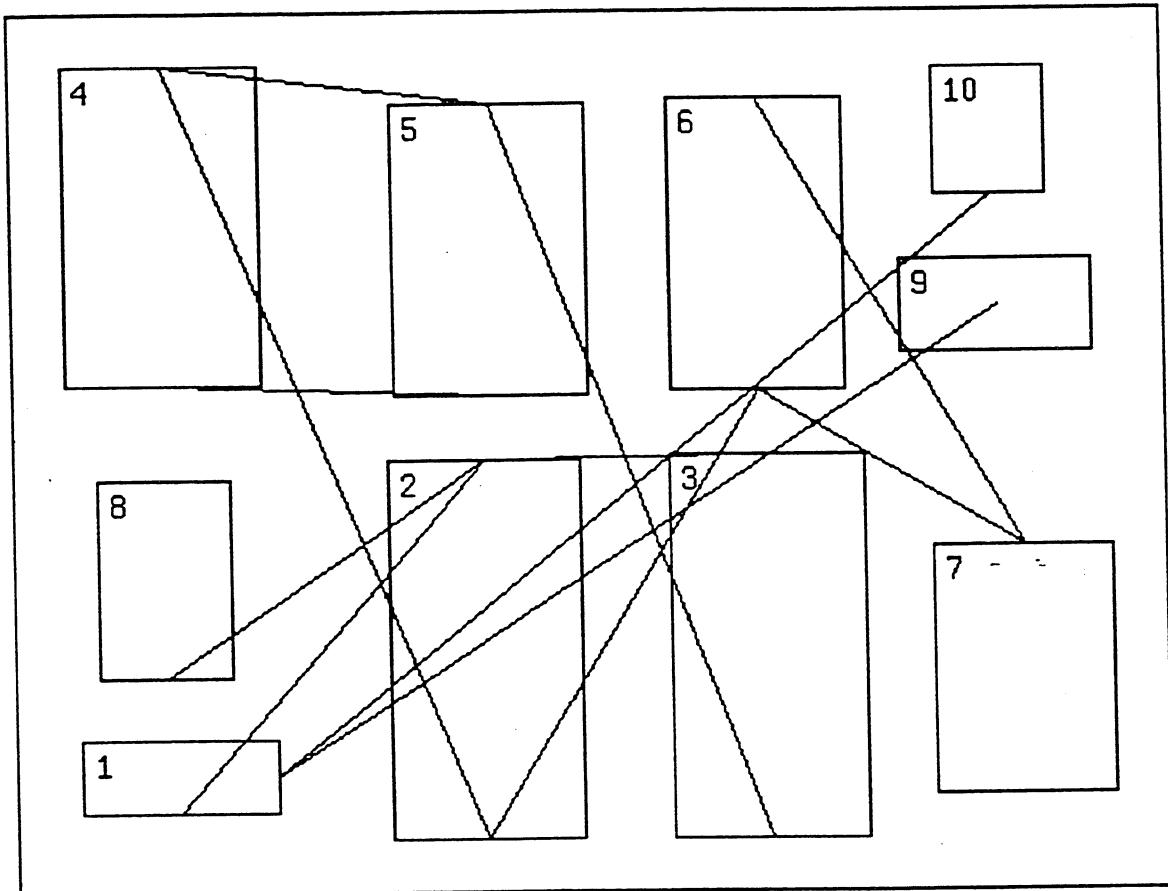
Dimensions du circuit:	2228*1924
Surface du circuit:	4286824
Longueur estimee des connexions:	44770
Nombre total de blocs:	3
Nombre total de connexions:	4



Dimensions du circuit:	19750*11411
Surface du circuit:	225375536
Longueur estimee des connexions:	3968888
Nombre total de blocs:	36
Nombre total de connexions:	98

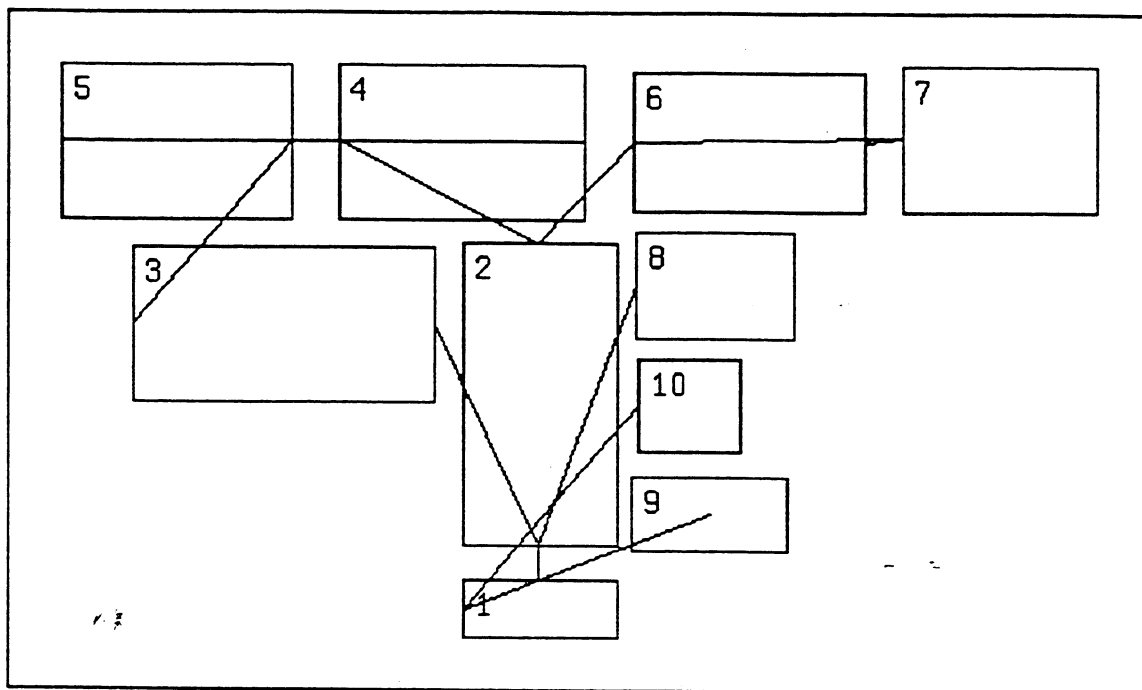
Placement obtenu pour le circuit Xerox

(Benchmark MCNC)



Dimensions du circuit:	9200*7028
Surface du circuit:	64664092
Longueur estimee des connexions:	567058
Nombre total de blocs:	10
Nombre total de connexions:	23

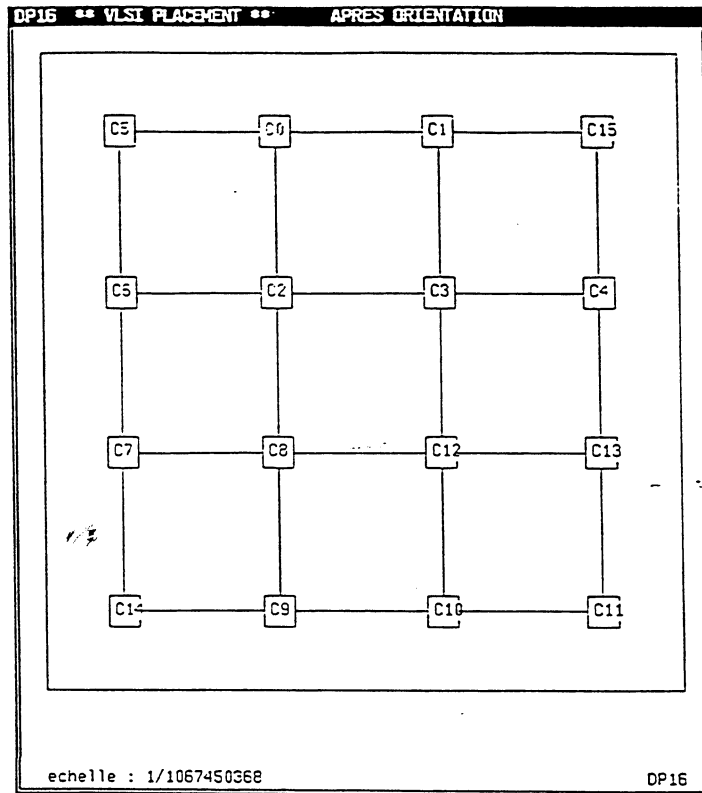
Placement par bipartitionnements successifs. Circuit: Xerox



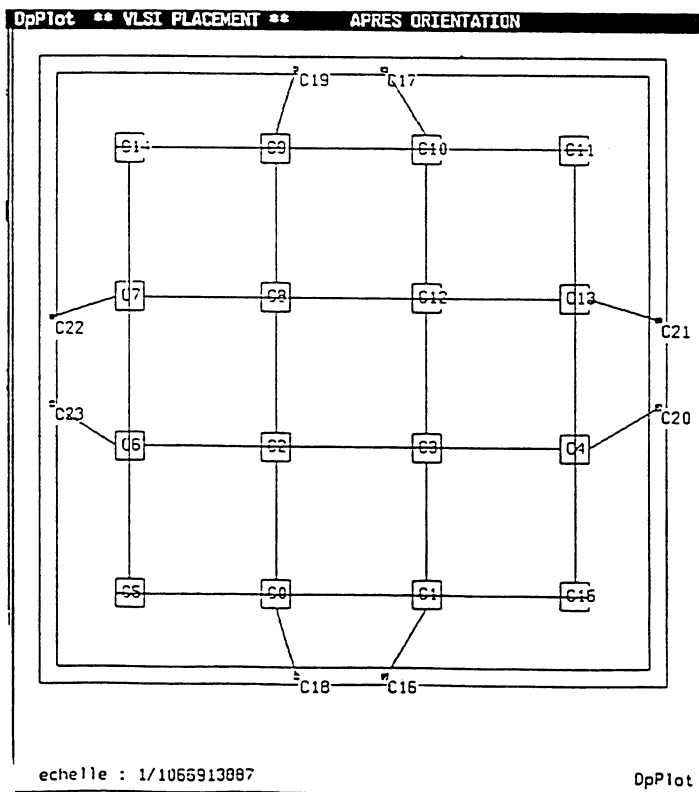
Dimensions du circuit:	11556*6796
Surface du circuit:	78529720
Longueur estimee des connexions:	293749
Nombre total de blocs:	10
Nombre total de connexions:	23

Traitement des plots d'E/S

Circuit 4x4 avant l'introduction des plots:



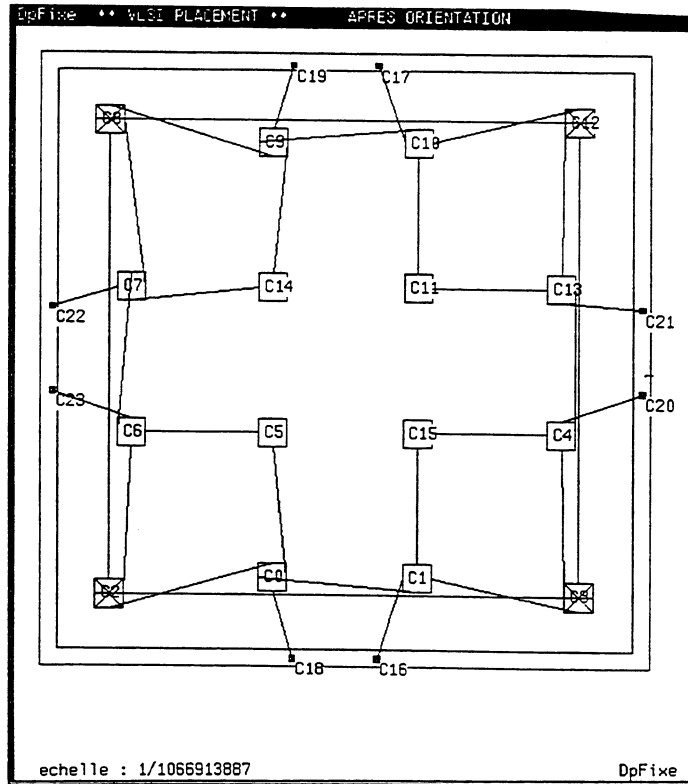
Résultat du placement après introduction des plots:



La configuration des blocs est inversée par rapport à l'axe horizontal passant par le milieu du circuit

Traitement des blocs fixes

Résultat du placement après que les blocs centraux aient été fixés dans les coins:



A U T O R I S A T I O N de S O U T E N A N C E

VU les dispositions de l'Arrêté du 30 mars 1992 relatif aux Etudes doctorales

VU les rapports de présentation de

- . Monsieur DREYFUS Gérard
- . Monsieur GERBER Roland

Monsieur Eric CHOTIN

est autorisé(e) à présenter une thèse en soutenance en vue de l'obtention du diplôme de DOCTEUR de L'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE, spécialité " Microélectronique "

Fait à Grenoble, le 6 novembre 1992
/ BV.

Pour le Président de l'INPG
et par délégation
le Directeur de l'Ecole Doctorale
J.L. LACOUME



Résumé :

Cette thèse présente l'étude et l'implantation de deux méthodes pour le placement automatique de circuits intégrés. Un circuit intégré peut être considéré comme un ensemble de blocs et une liste d'interconnexions entre ces blocs. Le problème du placement consiste à disposer les blocs sur la surface hôte en respectant diverses contraintes et en optimisant des critères comme la surface occupée et la longueur totale de connexions. Les méthodes présentées ici sont toutes les deux guidées par l'optimisation de la connectique. La première fait appel à une technique d'analyse de données, l'analyse d'un tableau de proximités. Dans un premier temps, des proximités sont définies entre les blocs de façon à refléter un agencement idéal en fonction de la connectique. L'utilisation de l'ATP permet alors d'obtenir une disposition planaire des blocs respectant au mieux les proximités qui ont été définies. L'analyse effectuée fait le point sur les diverses façons de définir les proximités entre les blocs, ainsi que sur les traitements ultérieurs destinés à l'obtention d'un placement réalisable. Les qualités et les limitations de cette approche sont ensuite discutées.

La seconde méthode est connue sous le nom de placement par bipartitionnements successifs. L'ensemble des blocs du circuit et la surface hôte sont ainsi bipartitionnés récursivement jusqu'à ce que l'emplacement de chaque bloc soit déterminé. A partir des algorithmes existants, des heuristiques ont été mises au point afin de permettre la prise en compte de contraintes supplémentaires comme le traitement des plots d'entrées-sorties ou des blocs pré-fixés. L'expérimentation a permis de valider ces heuristiques et de comparer les résultats du placement par bipartitionnements successifs à ceux fournis par la première méthode.

Mots clefs : placement de circuit, analyse de données, bipartitionnement de graphe.