

Documentation Service Bureau

SB

M P S X - M I P

CODE DE PROGRAMMATION LINEAIRE AVEC  
VARIABLES BORNEES ET VARIABLES ENTIERES

Ce manuel d'utilisation du système MPSX-MIP  
est une adaptation française des brochures  
américaines :

Introduction to MPSX	H20-0849
Control Language - User's Manual	H20-0932
Linear and Separable Programming description Manual	H20-0968
Report Generator Program description Manual	H20-0560
Readcomm Program description Manual	H20-0960
MPSX-MIP Program description Manual	H20-0908

Il ne prétend pas être exhaustif et les rempla-  
cer, aussi l'utilisateur devra t'il s'y reporter en  
cas de nécessité.

Il présente par ailleurs quelques particularités  
propres au SERVICE BUREAU.

BIBLIOTHEQUE DU CERIST

03 005 135

151 79

TABLE DES MATIERES

---

	Page
INTRODUCTION .....	1
1ère PARTIE : COMPILATION - EXECUTION PROGRAMME DE CONTROLE...	15
Chapitre I : COMPILATION - EXECUTION.....	17
Chapitre II : PROGRAMME DE CONTROLE - SYNTAXE.....	20
Chapitre III : PROGRAMME DE CONTROLE - INSTRUCTIONS.....	26
Instructions d'identification.....	26
PROGRAM.....	26
PEND.....	27
DC.....	27
Appel de procédures.....	27
Instructions de transfert.....	28
GOTO.....	28
EXEC.....	29
STEP.....	29
CONTINUE.....	29
IF.....	30
TALLY.....	30
EXIT.....	30
Instructions d'assignation.....	30
égalité.....	31
MOVE.....	31
TITLE.....	32
MVADR.....	32
MVIND.....	33
MVIND2.....	34
Macro-Instructions.....	34
Chapitre IV : INTERRUPTIONS.....	37
par fréquence.....	37
des procédures.....	38
INITIALZ.....	39

BIBLIOTHEQUE DU CERIST

	Page
Chapitre V : PROCEDURES GENERALES.....	40
TIME.....	40
WRITE = TYPE.....	41
STATUS.....	42
BCD.....	43
READ.....	44
COMMUN.....	45
NEGDJ.....	46
 2ème PARTIE : LA PROGRAMMATION LINEAIRE.....	 47
Chapitre I : DEFINITION DU MODELE-FORMAT DES DONNEES...	49
Généralités.....	49
Données de CONVIRT.....	51
Données de REVISE.....	53
Données d'INPUTPF.....	61
 Chapitre II : LES LISTES DE SÉLECTION.....	 63
 Chapitre III: PROCEDURES D'ENTREE.....	 65
Généralités.....	65
CONVERT.....	69
REVISE.....	74
INPUTPF.....	78
COPY.....	79
SETUP.....	81
FLAGS.....	85
FIXVAR.....	88
PROTECT.....	91
 Chapitre IV : PROCEDURES ITERATIVES.....	 93
Généralités.....	93
PRIMAL.....	99
DUAL.....	102
OPTIMIZE.....	103
PARARHS.....	108
PARAOBJ.....	109
PARARIM.....	110
PARACOL.....	111
PARAROW.....	113
 Chapitre V : PROCEDURES DE SORTIE.....	 115
Généralités.....	115
SOLUTION.....	118
RANGE.....	125
CHECK.....	133
TRANCOL.....	135
TRANROW.....	139
TRACE.....	141
PICTURE.....	144
BCDOUT.....	148
PROBLEMS.....	152

	Page
Chapitre VI ; PROCEDURES DE DEPART, SAUVEGARDE ET REPRISE	155
Généralités.....	155
CRASH.....	157
SAVE.....	159
RESTORE.....	161
SAVERMS.....	163
PUNCH.....	165
INSERT.....	168
BASIC.....	170
BASICX.....	171
NEWRES.....	173
Chapitre VII : PROCEDURES DE REDUCTION.....	175
REDUCE.....	175
RECUCEM.....	178
RECREATE.....	180
CREATEM.....	182
Chapitre VIII : PROCEDURES DIVERSES.....	185
INVERT - RETRIEVE.....	185
SELECT.....	187
Chapitre IX : FONCTIONS SEPARABLES.....	191
Description générale.....	191
Insertion des variables spéciales dans le modèle.....	197
3ème PARTIE ; EDATEUR DE RAPPORTS.....	199
Chapitre I : GENERALITES.....	201
Chapitre II : LANGAGE EXTERIEUR.....	204
Chapitre III: PROCEDURES.....	227
ANALYZE.....	227
SETREP.....	230
REPORT.....	233
<u>4ème PARTIE ; VARIABLES MIXTES.....</u>	<u>235</u>
Chapitre I : GENERALITES.....	237
Chapitre II : PRESENTATION DES DONNEES.....	241
Chapitre III : OPTIMIX.....	243
Chapitre IV : PROCEDURES & MACRO.....	247
INIMIX.....	247
MIXSTART.....	249
MIXFLOW.....	253
MIXSTATS.....	259

	Page
MIXSAVE.....	261
MIXFIX.....	262
<u>5ème PARTIE : COMMUNICATIONS AVEC DES PROGRAMMES EXTERIEURS</u>	265
Chapitre I : FORMAT DE COMMUNICATIONS.....	267
Généralités.....	267
Liaison avec le système MPSX.....	268
Sortie en format de communication.....	271
READCOMM.....	281
Chapitre II : FORMAT STANDARD.....	287
Généralités.....	287
Sortie en format standard.....	288
APPENDICES.....	289
APPENDICE A : DJLTRN.....	289
APPENDICE B : FICHIERS.....	297
ASSIGN.....	302
CLOSEF.....	303
APPENDICE C : TOLERANCES.....	305
APPENDICE D : REGION DE COMMUNICATION.....	307
INDEX ALPHABETIQUE DES PROCEDURES.....	317

## INTRODUCTION

### 1 - Description

MPSX-MIP (Mathematical Programming System Extended - Mixed Integer Programming) est un Programme Général destiné à la Programmation Mathématique. Il est utilisable sous le Système d'Exploitation 360 (OS 360) sur les ordinateurs de la série 360, modèles 30 à 85, et ceux de la série 370, équipés d'une mémoire de 128.000 octets au minimum.

La dimension des problèmes que MPSX peut résoudre et la rapidité avec laquelle il fournit la solution sont évidemment liées à la machine employée (environ 300 lignes sur la configuration minimum et jusqu'à 8000 lignes sur une machine à 512.000 octets de mémoire). Mais l'utilisation du même code de calcul sur toutes les machines assure la compatibilité parfaite des données et des résultats : ce qui permet de passer sans problème d'un ordinateur à l'autre.

MPSX est un Système ouvert dans lequel les modules de Programmation Mathématique peuvent s'ajouter au fur et à mesure de leur développement, élargissant son champ d'application.

sont actuellement disponibles :

#### Un module de Programmation Linéaire

utilisant la méthode révisée du simplexe, inverse factorisée, avec variables bornées et double limitation de l'activité des lignes. Ce module traite également les fonctions séparables (linéarisables par tronçons).

#### Un module de Programmation Linéaire Mixte

traitant des variables continues et jusqu'à 4095 variables entières. Ce module utilise une méthode de recherche par arborescence "Branch and Bound".

#### Un module d'Edition de Rapport

permettant à l'aide d'un langage simple accessible au non spécialiste de la Programmation Linéaire d'éditer des résultats sous la forme désirée.

#### Un module de Communication avec l'extérieur

permettant à des programmes écrits en FORTRAN par l'utilisateur d'accéder aux résultats de MPSX.

Le langage MARVEL, vaste système conçu pour la manipulation et l'édition de données brutes et de résultats, n'est pas décrit dans cette brochure. L'utilisateur intéressé par ce système pourra se référer à la brochure en langue anglaise (H20 - 0505).

## 2 - Les problèmes traités par MPSX

La Programmation Linéaire est une technique de recherche opérationnelle permettant de trouver une solution optimale à un problème lorsque celui-ci peut être modélisé sous forme de relations linéaires entre des variables.

Sa forme canonique est :

Trouver les valeurs de  $x_j$  qui minimisent (ou maximisent)

$$\sum_j c_j x_j$$

En vérifiant simultanément toutes les contraintes

$$\sum_j a_{ij} x_j = b_i$$

La Programmation Linéaire Séparable est une technique de programmation mathématique qui permet de traiter des problèmes contenant des contraintes non linéaires (mais séparables, c'est-à-dire polygonales) en utilisant les méthodes générales de la programmation linéaire.

La Programmation Linéaire Mixte est un cas particulier de la programmation linéaire dans lequel certaines des variables sont astreintes à la condition supplémentaire de ne prendre que des valeurs entières.

Traiter l'un de ces problèmes, c'est donc :

- a) Formuler le modèle mathématique du problème, c'est-à-dire établir les relations entre variables ainsi que la fonction à optimiser.
- b) Décrire la matrice des coefficients  $c_j$ ,  $a_{ij}$  et  $b_i$  de ces relations linéaires dans un format pouvant être lu par le programme MPSX. Ceci est réalisé soit en décrivant directement la matrice dans le format requis, soit en utilisant le système de génération de matrices MPSMG (MPSX Matrix Generator) disponible en Service Bureau.
- c) Optimiser le problème en utilisant au mieux toutes les ressources du langage de contrôle et des procédures du système MPSX.
- d) Eventuellement faire des analyses postoptimales et des paramètres sur la solution trouvée à l'aide des possibilités offertes par MPSX.
- e) Eventuellement présenter les résultats sous forme de rapports conçus par l'utilisateur au moyen du programme d'édition de rapport en clair MPSRG (MPSX Report Generator).

### 3 - Aperçu de la méthode mathématique utilisée

MPSX utilise comme algorithme de résolution la méthode révisée du simplexe avec inverse factorisée, variables bornées et double limitation de l'activité des lignes. La méthode du simplexe repose sur le fait que s'il existe  $m$  contraintes (ou lignes) de la matrice linéairement indépendantes, il existe alors un ensemble de  $m$  colonnes (variables ou vecteurs) qui seront aussi linéairement indépendantes. Un second membre (RHS) peut alors s'exprimer en fonction de ces  $m$  variables qui forment une base du problème.

La méthode du simplexe explore successivement une série de bases, en échangeant à chaque étape ou itération une variable de la base avec une variable hors base jusqu'à ce qu'une solution (solution de base acceptable) vérifiant toutes les contraintes soit trouvée. La méthode du simplexe parcourt ensuite un ensemble de solutions de base acceptables de façon à en trouver une qui minimise ou maximise la fonction économique. Cette solution est alors appelée solution optimale.

Les problèmes de programmation linéaire n'ont pas forcément une solution optimale. Si dans un problème il n'existe pas de solutions vérifiant simultanément toutes les contraintes, on dit que le problème est inacceptable (INFEASIBLE). S'il existe une solution acceptable qui donne à la fonction économique une valeur infinie la solution est dite illimitée (UNBOUNDED).

### 4 - Vue d'ensemble sur le système MPSX-MIP

#### 4.1 Programme de contrôle et Procédures

Le Système MPSX se compose d'un langage de contrôle et d'une bibliothèque de programmes comprenant le COMPILER, l'EXECUTOR et un ensemble de Procédures.

Le programme de contrôle, écrit par l'utilisateur en langage MPSX, permet de mettre en oeuvre ces procédures en choisissant la stratégie de résolution la mieux adaptée à son problème. C'est une suite d'instructions qui permettent d'effectuer :

- l'appel de procédures
- des mouvements de données et définition de constantes
- des calculs arithmétiques et logiques
- des branchements
- la définition de macro-instructions.

Dans une première étape de travail (JOB STEP) ce programme de contrôle est compilé par le COMPILER en un programme résultant directement exécutable par l'EXECUTOR

Dans une deuxième étape de travail, l'EXECUTOR contrôle l'exécution du programme de contrôle, c'est-à-dire appelle les procédures, effectue les mouvements de données, calcule les expressions arithmétiques et logiques...

Cet enchaînement est représenté sur la figure 1.

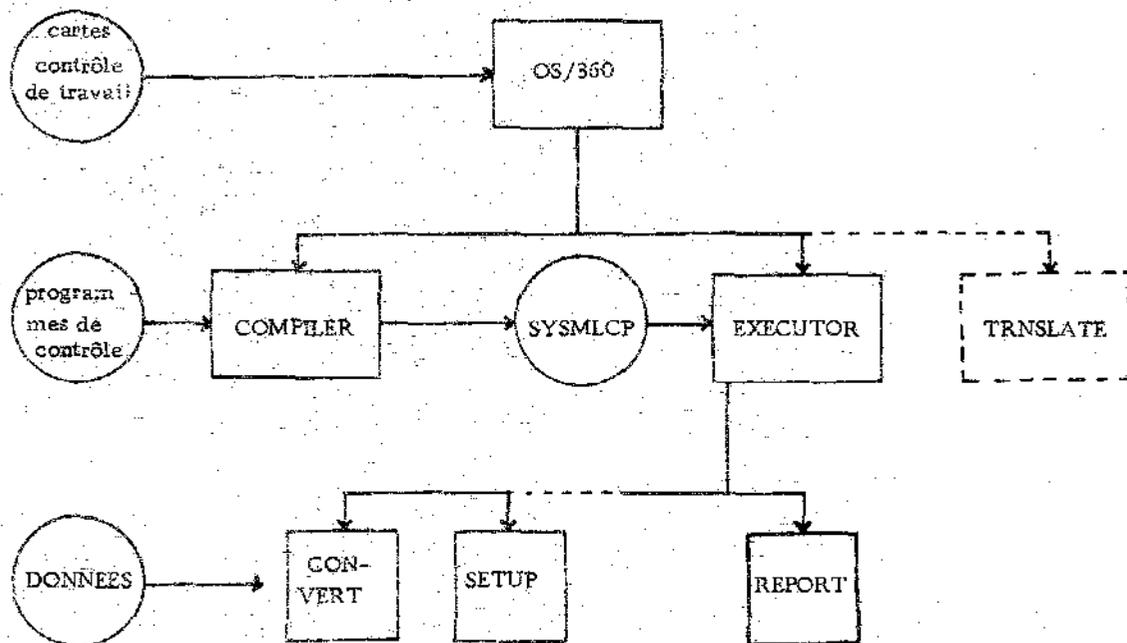


Figure 1

## 4.2 Région de Communication

Pour communiquer entre elles et avec le programme de contrôle, les procédures utilisent une zone de la mémoire principale appelée Région de Communication (CR). Cette zone est divisée en un certain nombre de cellules ayant chacune un nom symbolique préétabli par le système MPSX.

En se référant à ces noms dans le programme de contrôle, on peut ainsi accéder au contenu de ces cellules. Tous les noms des cellules de la CR commencent par la lettre X. La liste des cellules auxquelles a accès l'utilisateur est donnée dans l'Appendice D.

## 4.3 Demande

Normalement, à la fin de son exécution, une procédure rend le contrôle à l'EXECUTOR qui prend alors en considération l'instruction suivante du programme de contrôle.

La demande est un moyen qui permet d'interrompre le déroulement logique d'un programme de contrôle pour se transférer ailleurs dans ce programme quand un événement déterminé se produit. A chaque type de demande correspond une cellule de la Région de Communication dans laquelle on placera l'adresse du point auquel on désire se débrancher quand l'évènement correspondant à cette demande se produit.

## 4.4 Résumé des instructions

### 4.4.1. Instructions de délimitation de programme

PROGRAM : Cette instruction non exécutable indique le début du programme de Contrôle.

PEND : Cette instruction non exécutable indique la fin du programme de contrôle.

### 4.4.2. Instruction d'appel aux Procédures

Elles sont composées du nom de la procédure et des paramètres que l'on désire passer. Une vue d'ensemble de ces procédures est donnée dans le paragraphe suivant.

Leur description détaillée se trouve dans chacune des parties de cette brochure consacrées aux différents modules du système MPSX.

#### 4.4.3. Instructions de Mouvement de données

- MOVE      Mouvement de caractères d'une zone de mémoire à une autre.
- MVADR     Mouvement d'une adresse du programme de contrôle dans une zone déterminée de la mémoire.
- MVIND     Mouvement dans une zone dont l'adresse est donnée de façon indirecte.
- MVIND2    Mouvement d'une zone dont l'adresse est donnée indirectement dans une autre zone dont l'adresse est également donnée de façon indirecte.

NB. Les zones de mémoire peuvent être soit des cellules de la Région de Communication, soit des zones définies par l'Utilisateur.

#### 4.4.4. Instructions arithmétiques

Elles sont constituées par un ensemble de variables et de constantes réunies entre elles par les opérateurs arithmétiques

= - + \* / \*\*

#### 4.4.5. Instructions logiques

Elles sont constituées par un ensemble de variables logiques réunies par des opérateurs logiques (AND, OR, NOT...) et/ou par des variables arithmétiques réunies par des opérateurs de comparaison (GT, LT, EQ... signifiant plus grand que, plus petit que, égal à...).

#### 4.4.6. Instructions de Transfert

- GOTO      Permet un débranchement incondtionnel à un point donné du programme
- IF        Permet un débranchement conditionné par le résultat d'une opération logique.
- EXEC      Permet un débranchement incondtionnel à un point donné du programme avec sauvegarde de l'adresse de cette instruction, ce qui permet de revenir en ce point du programme.
- CONTINUE Permet en fin de séquence de rendre la main au dernier EXEC ou à l'instruction qui a provoqué la dernière demande.

STEP	Permet en fin de séquence de rendre la main à l'instruction qui suit le dernier EXEC ou à l'instruction qui suit l'instruction qui a provoqué la dernière demande.
TALLY	Permet de faire un nombre défini de boucles, elle effectue un branchement à un point du programme N+1 fois N étant spécifié dans l'instruction.
EXIT	Permet de rendre la main à l'Operating System en fin d'exécution du programme de contrôle.

#### 4.4.7. Instructions de définition de constante

DC	(Define Constant) permet d'associer à un nom une zone mémoire et de lui donner une valeur initiale.
TITLE	Permet de fournir un titre qui sera imprimé en tête de chaque page à l'exécution.

#### 4.4.8. Instructions de définition de Macro-Instructions

Une macro-instruction est un ensemble d'instructions groupées sous un nom auquel l'utilisateur pourra faire référence dans son programme quand il voudra exécuter la série d'instructions

MACRO	Indique le début de définition d'une macro-instruction
MEND	Indique la fin de la définition

Entre ces deux instructions se trouveront :

- une carte en tête donnant le nom de la macro et lui spécifiant éventuellement une liste de paramètres.
- l'ensemble des instructions constituant la macro-instruction.

L'emploi des macro-instructions permet d'éviter l'écriture de séries d'instructions.

#### 4.5 Macro instruction système

Une macro-instruction système est une macro-instruction que l'utilisateur n'a pas besoin de définir dans le programme de contrôle car sa définition est stockée dans la bibliothèque du système MPSX.

Le système MPSX fournit une macro-instruction d'initialisation : INITIALZ.

Cette macro-instruction effectue toutes les initialisations standard relatives aux demandes, tolérances, et fréquences d'interruption.

Le système MPSX fournit également d'autres macro-instructions système. Leur description est faite à la suite des procédures auxquelles elles se rapportent.

#### 4.6 Résumé des procédures et macro-instructions

On peut regrouper les procédures et macro-instructions selon leurs fonctions :

- . d'entrée et de mise à jour des fichiers.
- . d'optimisation
- . de sortie
- . d'analyses post optimale
- . d'arrêt et de reprise de travail
- . de départ
- . de Commandes
- . d'utilisation de fichiers

##### 4.6.1. Procédures d'entrée et de mise à jour des fichiers

CONVERT lit des données en BCD (Décimal Codé Binaire) ou en format de communication (données générées par MARVEL) et les ré-écrit sur le fichier PROBFIL selon un format interne condensé.

Un même PROBFIL peut contenir plusieurs problèmes, identifiés par leur nom. En écrivant chaque problème, CONVERT fait également un contrôle d'erreurs sur les données et calcule des statistiques destinées à la procédure SETUP.

REVISE génère un nouveau problème sur le PROBFIL à partir d'un ancien problème et de données de révision. L'ancien problème peut être soit sur le PROBFIL, soit sur un autre fichier : OLDPROBFIL.

COPY	Permet de mettre à jour le PROBFILÉ en sélectionnant et copiant des problèmes de l'OLDPFILE sur le PROBFILÉ.
SETUP	sélectionne un problème du PROBFILÉ, analyse les statistiques du modèle (Nombre de lignes, colonnes, élément non nuls), les types d'unités d'entrée-sortie, adapte une base initiale, alloue dynamiquement la mémoire nécessaire à la résolution du problème et crée les fichiers de travail destinés aux procédures de calcul.
SAVERHS	Ajoute un second membre (RHS) au PROBFILÉ, à la matrice de travail ou aux deux.
FLAGS	Sélectionne un sous ensemble des données de la matrice originale d'un problème du PROBFILÉ et crée les fichiers de travail destinés aux procédures de calcul pour restreindre temporairement le problème.
DJLTRN	produit les données de codes de programmation linéaires IBM autres que MPSX ou MPS en format MPSX. Cette procédure devra être effectuée dans une étape de travail séparée et n'exclut pas le CONVERT.
INPUTPF	accepte en entrée : <ul style="list-style-type: none"> <li>- la matrice de sortie du générateur de matrice MPSMG</li> <li>- des données en format MPSX non triées</li> <li>- des données en format libre.</li> </ul> génère en sortie un fichier décrivant la matrice selon le format requis par la procédure CONVERT.
PROTECT	ordonne les problèmes sur le PROBFILÉ de façon à ce qu'un problème déterminé soit placé en dernier.
ANALYZE	compile le programme de génération de rapport et analyse, si nécessaire, les données du rapport.
SETREP	complète la compilation d'un générateur de rapport en résolvant les références du problème résolu.

#### 4.6.2. Procédures et macro-instruction d'optimisation

Ces procédures optimisent le modèle mathématique à partir des données créées sur les fichiers de travail par la procédure SETUP.

Les multiples options associées à ces procédures permettent à l'utilisateur un grand choix d'algorithme dans la recherche de la solution.

Les options standard de la macro-instruction système INITIALZ sont généralement suffisantes.

PRIMAL	c'est la principale procédure d'optimisation. Elle recherche la solution optimale (si elle existe) avec un algorithme primal composite. En fin de procédure la solution peut être optimale, inacceptable (INFEASIBLE) ou illimitée (UNBOUNDED).
DUAL	obtient une solution primale acceptable par un algorithme dual. Cette procédure ne recherche pas l'optimalité elle devra donc être suivie par un appel de PRIMAL.  PRIMAL et DUAL nécessitent une base de départ complète, celle-ci est fournie soit par restauration ou insertion d'une base de départ, soit par les procédures de démarrage CHERCHE et CRASH, soit par SETUP.
INVERT	calcule l'inverse de la base courante. La réduction qui en résulte augmente la vitesse et la précision de la résolution dans les itérations suivantes.
RETRIEVE	fait si possible rentrer dans la base les vecteurs sortis par INVERT à cause des singularités de la matrice.
OPTIMIZE	est une macro-instruction du système MPSX qui fournit un algorithme de résolution évolué tout en simplifiant l'écriture du programme de contrôle.
MIXFLOW	effectue la recherche de solutions entières dans un problème à variables mixtes.
OPTIMIX	est une macro-instruction du système MPSX qui simplifie l'emploi de MIP en développant une stratégie standard de résolution des problèmes à variables mixtes.

#### 4.6.3. Procédures de sortie

Ces procédures permettent de sortir les solutions trouvées.

SOLUTION	édite la solution (niveau d'activité, coût réduit, coûts originaux). Elle peut être appelée à tout instant en cours ou en fin d'optimisation ou de paramétrisation.
BCDOUT	fournit un fichier en BCD (décimal code binaire) à partir d'un problème du PRODFILE (c'est le contraire de CONVERT).
PICTURE	représente la matrice de travail sous forme de tableau en indiquant l'existence et l'ordre de grandeur de chacun des éléments.

TRANCOL	transforme les variables hors base de la matrice pour montrer l'effet sur la solution courante de la variation de l'une d'entre elles. L'information est donnée colonne par colonne.
TRANROW	donne la même information que TRANCOL, mais ligne par ligne TRANROW peut également donner l'inverse explicite de la base courante.
TRACE	donne un rapport sur les vecteurs provoquant l'inacceptabilité d'une solution.
CHECK	imprime pour chaque ligne la valeur de l'erreur sur cette ligne.
STATUS	imprime l'état du système (Mémoire, unité d'entrée /sortie etc...).
REPORT	provoque l'impression d'un rapport précédemment créé.
MIXSTATS	cette procédure de MIP imprime un résumé de l'état de la recherche de solutions entières.

#### 4.6.4. Procédures d'analyses post-optimales

Ces procédures permettent une fois que la solution optimale a été trouvée, d'analyser la plage de validité de cette solution ou variation en fonction de modifications du second membre, de la fonction économique ou de lignes ou colonnes de la matrice.

RANGE	donne : - l'effet sur la solution optimale de la variation d'un coefficient de coût.  - le coût entraîné par la modification de l'activité optimale d'une ligne ou d'une colonne et la plage d'activité pour laquelle ce coût est valide.
PARAOBJ	assure une paramétrisation de la fonction économique.
PARARHS	assure une paramétrisation du second membre.
PARARIM	assure une paramétrisation simultanée de la fonction économique et du second membre.
PARACOL	assure une paramétrisation sur une colonne déterminée de la matrice.
PARAROW	assure une paramétrisation sur une ligne déterminée de la matrice.

#### 4.6.5. Procédures d'arrêt et de reprise de travail

Ces procédures permettent de sauvegarder et de retrouver la base courante et certaines informations relatives à un problème donné.

SAVE	permet la sauvegarde sur le PROFILE de la base courante du problème et d'un ensemble de cellules de la région de communication (CR). L'identification se fait par le nom du problème et par un nom de sauvegarde.
RESTORE	Restaure une base et les cellules de la région de communication sauvegardées par SAVE.
PUNCH	perfore la base courante du problème en format image carte.
INSERT	Permet d'insérer ou de modifier une base à l'aide d'un fichier en image cartes produit par PUNCH ou fourni par l'utilisateur.
MIXSAVE	cette procédure de MIP sauvegarde l'état de prospection atteint pour permettre de le restaurer dans un passage ultérieur.
MIXSTART	cette procédure de MIP initialise la recherche de solutions entières ou restaure une recherche précédemment sauvegardée.

#### 4.6.6. Procédures et macro-instruction de départ

Ces procédures permettent de démarrer un problème pour calculer une meilleure base de départ pour PRIMAL que celle fournie par SETUP lorsque INSERT ou RESTORE ne sont pas utilisés.

CHERCHE	calcule une base départ à partir de la base constituée par les vecteurs logiques.
CRASH	fournit une bonne base de départ en introduisant des vecteurs structuraux dans la base et en essayant de réduire le nombre d'inacceptabilités de la solution.
BASICX	permet de trouver une solution basique à un problème dont on connaît une solution non basique. Elle est utile en particulier pour optimiser une matrice ayant une structure par blocs dont on connaît les solutions optimales correspondant à chaque bloc pris séparément.
BASIC	est une macro-instruction du système MP5X qui fournit une solution de base à un problème construit sur différents sous-problèmes.

#### 4.6.7. Procédures de contrôle

Ces procédures indiquent l'état du système à un instant donné et permettent de contrôler l'avancement du traitement.

SELECT	rend accessible au programme de contrôle des éléments du problème tels que : niveaux d'activités, coûts, valeurs du second membre, activités duales et coûts marginaux de variables sélectionnées.
PROBLEMS	imprime et rend accessible au programme de contrôle les noms des problèmes qui sont sur le PROFILE et/ou l'OLDPROFILE.
WRITE	imprime la valeur de variables définies par l'utilisateur dans le programme de contrôle.
TIME	calcule et rend accessible au programme de contrôle le temps écoulé.
READ	permet de lire des données pour initialiser des cellules de la région de Communication.
BCD	convertit une variable entière du programme de contrôle en une constante caractère.
FREECORE	libère la mémoire pour les programmes de l'utilisateur.
TYPE	imprime sur la console opérateur la valeur de variables définies par l'utilisateur.
NEGDJ	calcule le nombre de DJ négatifs.
COMMON	imprime des messages sur la console opérateur et attend une réponse.
MIXFIX	fixe les variables entières à leurs activités entières permettant ainsi de faire une analyse post optimale d'une solution entière donnée.

#### 4.6.8. Procédures et macro-instructions de réduction

REDUCE	essaye de réduire la dimension d'un problème en éliminant lignes et colonnes redondantes, en libérant des colonnes dont les bornes ne peuvent être atteintes, en remplaçant des contraintes par des bornes sur des variables.
RECREATE	recrée le problème original à partir du problème réduit par REDUCE de façon à pouvoir faire des analyses post-optimales ou éditer des rapports.

- REDUCEM est une macro-instruction du système MPSX qui enchaîne automatiquement l'appel des procédures nécessaires à la réduction d'un problème
- CREATEM est une macro-instruction du système MPSX qui enchaîne automatiquement l'appel des procédures nécessaires pour recréer le problème original.

#### 4.6.9. Procédures d'utilisation des fichiers

Ces procédures permettent d'identifier et de manipuler à l'aide du programme de contrôle les fichiers qui ne sont pas définis par le système.

- ASSIGN Associe un nom de fichier permettant d'identifier un fichier dans le programme de contrôle, à un DDNAME défini dans les cartes de contrôle OS.
- CLOSEF Effectue la fermeture d'un fichier déterminé.
- PREPIN Ouvre un fichier d'entrée pour READCOMM
- PREPOUT Ouvre un fichier de sortie pour READCOMM