

Université Lille
Licence mention informatique S5
Méthodes Linéaires (ML)

RAPPORT DU DEVOIR MAISON DE ML
Les transistors d'Ornicar

SKOCZYLAS Nestor
FRET Gaëlle

20 novembre 2021

Ce rapport traitera de l'ensemble des questions figurants dans cet exercice, les unes après les autres. Pour chacune d'elle, les choix de modélisations, les hypothèses et les réponses apportées par les modèles, seront expliqués.

Le sujet expose les transistors électroniques, ici c'est un fabricant nommé Ornicar. Les transistors sont divisés en deux types : PNP et NPN. Chaque transistor possède deux caractéristiques principales, le gain hfe (coefficient d'amplification) et vbe (tension base émetteur). De plus, la valeur de hfe réside entre 0 et MAXHFE (avec MAXHFE = 600) tandis que celle de vbe réside entre 0 et MAXVBE (avec MAXVBE = 1 Volt).

Le but de cet exercice est de minimiser la dispersion d'un ensemble de transistors. La formule pour calculer cette dispersion est fournie dans le sujet :

$$D(\{T1, \dots, Tn\}) = \max(dh/MAXHFE, dv/MAXVBE)$$

où $dh = \max(h_i) - \min(h_i)$ et $dv = \max(v_i) - \min(v_i)$.

Question 1. RESOLUE

1.1 Hypothèse

Objectif : On veut calculer la dispersion d'un ensemble de transistors.

Hypothèse : On suppose alors que le résultat qu'on aura sera proche de 1 afin de la minimiser au maximum sur la suite des questions. Par conséquent, 0 est la valeur vers laquelle on veut tendre au maximum.

1.2 Modélisation

Ensemble :

- TRANS
ensemble de tous les transistors

Paramètres (ce que l'on connaît) :

- type
type de transistors (nnp ou pnp)
- MAXHFE
maximum que peut atteindre hfe
- MAXVBE
maximum que peut atteindre vbe
- hfe
coefficient d'amplification d'un transistor, doit être compris entre 0 et MAXHFE
- vbe
tension base émetteur, doit être compris entre 0 et MAXVBE

Variables (ce que l'on veut calculer) :

- disp
la dispersion, ce que l'on veut obtenir

- dh
maximum de h_i soustrait au minimum de h_i
- dv
maximum de v_i soustrait au minimum de v_i
- maxhi
maximum de hfe
- minhi
minimum de hfe
- maxvi
maximum de vbe
- minvi
minimum de vbe

Contraintes à respecter :

- hfe d'un transistor
on doit sélectionner le plus grand et le plus petit hfe dans les transistors de l'ensemble
- vbe d'un transistor
on doit sélectionner le plus grand et le plus petit vbe dans les transistors de l'ensemble
- Calcul de dh
on soustrait le max et min de hfe
- Calcul de dv
on soustrait le max et min de vbe
- Calcul de la dispersion
la dispersion dépend du maximum entre maxdh et maxvh. Ces derniers valant respectivement dh/MAXHFE et dv/MAXVBE

1.3 Réalisation en AMPL

1.3.1 Ensemble, paramètres et variables

```
set TRANS;
```

L'ensemble *TRANS* est l'ensemble des transistors, utile pour calculer la dispersion.

```
param type {TRANS} symbolic;
param MAXHFE >= 0;
param MAXVBE >= 0;
param hfe {TRANS} > 0;
param vbe {TRANS} > 0;
```

Le paramètre *type* permet d'identifier le type de transistor, soit pnp soit npn. *MAXHFE* et *MAXVBE* sont les maxima des *hfe* et *vbe* des transistors, ils sont donc dépendants de l'ensemble *TRANS*.

```
var disp >= 0;
var dh >= 0;
var dv >= 0;
var maxhi >= 0;
var minhi >= 0;
var maxvi >= 0;
var minvi >= 0;
```

1.3.2 Contraintes

Contrainte 1 : dh et dv seront le résultat de la soustraction du maximum et du minimum du hfe et du vbe de transistors. On veut ainsi modéliser une des parties de la formule de la dispersion.

```
subject to dhfe : dh = maxhi - minhi;
```

```
subject to dvbe : dv = maxvi - minvi;
```

Contrainte 2 : On sélectionne le hfe maximum et minimum dans l'ensemble des transistors (TRANS).

```
subject to maxhfei {t in TRANS} : maxhi >= hfe[t];
```

```
subject to minhfei {t in TRANS} : minhi <= hfe[t];
```

Contrainte 3 : On sélectionne le vbe maximum et minimum dans l'ensemble des transistors (TRANS).

```
subject to maxvbei {t in TRANS} : maxvi >= vbe[t];
```

```
subject to minvbei {t in TRANS} : minvi <= vbe[t];
```

Contrainte 4 : On calcule la dispersion en deux étapes : on calcule la première partie de la formules de la dispersion ($dh/MAXHFE$), puis la deuxième ($dv/MAXVBE$). Pour ne garder que le maximum entre les deux.

```
subject to maxdh : disp = dh / MAXHFE;
```

```
subject to maxdv : disp = dv / MAXVBE;
```

1.3.3 Objectif

```
minimize dispersion : disp;
```

Car le but de l'exercice est de minimiser la dispersion.

1.3.4 Données

```
param MAXHFE := 600;
```

```
param MAXVBE := 1.0;
```

```
param : TRANS : hfe vbe type =
```

```
T1 369.713399 0.505002 npn
```

```
T2 172.445927 0.527908 pnp
```

```
T3 388.349744 0.678436 pnp
```

```
T4 345.246256 0.506357 pnp
```

```
T5 159.318987 0.601071 pnp
```

```
...
```

1.1.5 Résultat

```
MINOS 5.51 : optimal solution found.  
4 iterations, objective 0.6519189267
```

```
disp = 0.651919
```

La dispersion de l'ensemble des transistors est de 0.651919. L'hypothèse précédemment énoncé s'avère correcte, car la dispersion tend plutôt vers 1.

Question 2. RESOLUE

2.1 Hypothèse

Objectif: On veut calculer une meilleure dispersion (donc une plus faible que celle de la question 1). On s'autorise à exclure jusqu'à 4 transistors.

Hypothèse: On sait que les caractéristiques des transistors doivent être les plus proches possibles afin d'avoir une meilleure dispersion, alors on peut supposer que ce sont ceux ayant des valeurs extrêmes qui seront exclus. On suppose aussi que le nombre de transistors exclus sera de 4 car plus retire ceux qui ont les valeurs les plus éloignées, plus la dispersion tendra vers 0.

2.2 Modélisation

Paramètre ajouté au modèle de la Q1:

- E
nombre maximum de transistors à exclure

Variable ajouté au modèle de la Q1:

- $est_conserve$
Permet de savoir si un transistor est exclu ou conservé dans l'ensemble

Contraintes ajoutées/modifiées au modèle de la Q1:

- hfe et vbe
Les calculs sont modifiés pour prendre en compte le fait que le transistor est exclu ou non de l'ensemble (et donc par extension du calcul de la dispersion)
- $est_conserve$
total des transistors conservés dans l'ensemble des transistors

2.3 Réalisation en AMPL

2.3.1 Paramètre et variables

```
param E >= 0;
```

Le paramètre E correspond aux 4 transistors à exclure de l'ensemble *TRANS*.

```
var est_conserve {TRANS} binary;
```

La variable *est_conserve* est un tableau binaire (0 ou 1). Pour chaque transistor il y aura soit :

- 0 si on enlève le transistor
- 1 si on garde le transistor

Cette variable dépend donc de l'ensemble TRANS.

2.3.2 Contraintes

Contrainte 1 : contrainte sur les maxima de hfe et vbe. On multiplie hfe/vbe par la variable est-conservé comme ça :

- si le transistor est conservé, la variable vaudra 1 et on stockera ainsi la valeur de hfe/vbe dans le max.
- sinon *est_conserve* vaudra 0, donc la valeur stockée dans le max sera de 0 (ce n'est donc pas ce maximum qui sera conservé pour les calculs de la dispersion).

```
subject to maxhfei {t in TRANS} : maxhi >= hfe[t] * est_conserve[t];
```

```
subject to maxvbei {t in TRANS} : maxvi >= vbe[t] * est_conserve[t];
```

Contrainte 2 : contrainte sur les minima de hfe et vbe. La première partie du calcul est identique à la contrainte précédente. On ajoute à cette première partie l'addition qui suit, $(1 - \text{est_conserve}[t]) * \text{MAXHFE}$ (ou MAXVBE), ainsi :

- Si *est_conserve* vaut 1, l'addition vaudra 0 et on aura ainsi $1 * \text{hfe} + 0 = \text{hfe}$.
- Dans le cas contraire l'addition vaudra *maxhfe* ce qui donnera : $0 * \text{hfe} + \text{maxhfe} = \text{maxhfe}$. Et si on conserve le maximum dans le minimum, cette valeur ne sera pas conservée pour la suite des calculs.

```
subject to minhfei {t in TRANS} : minhi <= est_conserve[t] * hfe[t]
+ (1 - est_conserve[t]) * MAXHFE;
```

```
subject to minvbei {t in TRANS} : minvi <= est_conserve[t] * vbe[t]
+ (1 - est_conserve[t]) * MAXVBE;
```

2.3.3 Objectif

```
minimize dispersion : disp;
```

L'objectif ne change pas entre la Q1 et la Q2.

2.3.4 Données

```
param MAXHFE := 600;
param MAXVBE := 1.0;
param E := 4; <- le nombre de transistors à exclure
```

```
param : TRANS : hfe vbe type =
T1 369.713399 0.505002 npn
T2 172.445927 0.527908 pnp
T3 388.349744 0.678436 pnp
```

```
T4 345.246256 0.506357 pnp
T5 159.318987 0.601071 pnp
...
```

On ajoute dans le fichier les commandes qui permettent de lancer notre modèle et d'afficher le résultat pour ne pas avoir à les écrire dans la console à chaque fois :

```
solve;
display disp;
display est_conserve ;
```

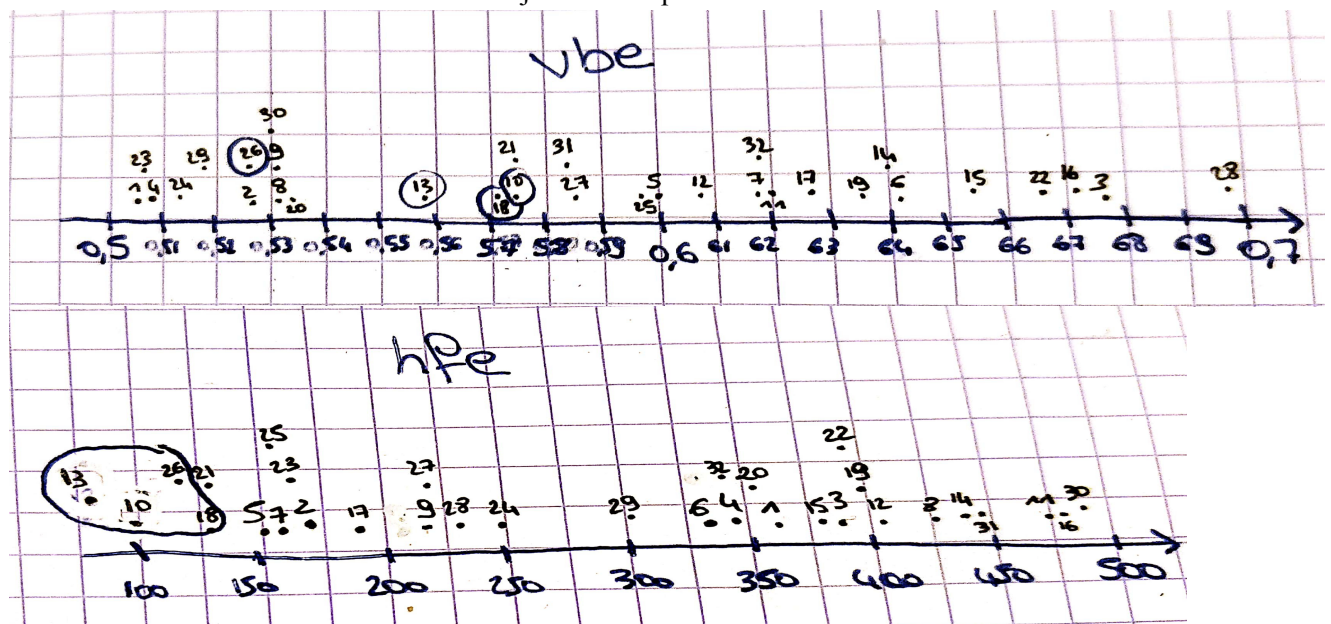
2.3.5 Résultat

```
Gurobi 8.1.0: optimal solution; objective 0.5814170133
50 simplex iterations
1 branch-and-cut nodes
plus 3 simplex iterations for intbasis
disp = 0.581417
```

```
est_conserve [*] :=
```

```
  T1 1    T13 0    T17 1    T20 1    T24 1    T28 1    T31 1    T6 1
T10 0    T14 1    T18 0    T21 1    T25 1    T29 1    T32 1    T7 1
T11 1    T15 1    T19 1    T22 1    T26 0    T3 1    T4 1    T8 1
T12 1    T16 1    T2 1    T23 1    T27 1    T30 1    T5 1    T9 1
;
```

La dispersion de l'ensemble des transistors conservés est de 0.581417. On s'aperçoit donc que la dispersion est plus faible que sur la première version. Notre hypothèse sur le fait que seuls les transistors avec des valeurs extrêmes soient exclus est juste. Nous pouvons le vérifier ainsi :



Schémas : Chaque point représente la valeur du h_{fe}/v_{be} de chaque transistor (la numérotation de chaque transistor correspond à celle des datas). Les transistors entourés sont ceux exclus.

De plus nous pouvons voir que 4 transistors ont bien été exclus. Notre modèle a fonctionné.

Question 3. RESOLUE

3.1 Critère

Sachant que la dispersion est encore trop grande, on découpe le stock de transistors en P paquets de transistors, où P vaut 3, ayant chacun le même nombre de transistors à 1 près. Le critère permettant d'avoir la plus petite dispersion possible pour chaque paquet est de départager les transistors pour que ceux aillant les caractéristiques les plus proches (autrement dit les valeurs de h_{fe} et v_{be}) soient dans le même paquet. Ainsi les 3 paquets contiendront précisément deux fois onze transistors et une fois dix transistors. De plus, on crée un nouvel ensemble nommé *PAQUETS* qui permettra d'affecter les transistors de l'ensemble *TRANS* dans plusieurs paquets. Ce nouvel ensemble comprendra un nombre de paquets allant de 1 à P (c'est à dire 3 paquets au total). Toutes les variables et les contraintes utilisées précédemment seront liées à l'ensemble des paquets. En plus, les différents paquets ne peuvent avoir une différence de nombre de transistors au sein d'eux supérieurs à 1. Nous souhaitons obtenir à la fin la dispersion moyenne de chaque paquet.

Question 4. RESOLUE

4.1 Hypothèse

Objectif : On veut calculer la plus petite dispersion possible grâce aux critères précédemment énoncés.

Hypothèse : On suppose que la dispersion moyenne des 3 paquets sera nettement plus faible (qu'elle tendra vers 0) et que si on additionne les dispersions de chaque paquet, la dispersion sera alors plus faible que celle de la question 2 ou au pire égale.

4.2 Modélisation

Ensemble ajouté au modèle de la Q2 :

- P
ensemble des paquets

Paramètre modifié au modèle de la Q2 :

- E devient P
nombre de paquets de transistors à créer

Variables modifiées et ajoutées au modèle de la Q2 :

- $disp$, dh , dv , max_{hi} , min_{hin} , max_{vin} , min_{vi}
Dépendent maintenant de l'ensemble de paquets car on souhaite minimiser la dispersion des paquets
- $est_conserve$ devient $est_affecte$
Pour savoir à quel paquet chaque transistor est affecté. Elle dépend de l'ensemble des paquets en plus de l'ensemble des transistors
- $moyenne_disp$

dispersion moyenne des 3 paquets

Contraintes à respecter modifiées et ajoutées au modèle de la Q2 :

- On ajoute la nouvelle dépendance des variables à l'ensemble PAQUETS dans les anciennes contraintes
- Dans les calculs des min et des max pour hfe et vbe, est_conserve devient est_affecte
- moyenne dispersion
moyenne de la dispersion d'un paquet
- est_affecte
chaque transistor doit être affecté à un et un seul paquet
- nombre de transistors par paquets
les paquets ne doivent pas avoir une différence de nombre de transistors supérieur à 1

4.3 Réalisation en AMPL

4.3.1 Ensemble, paramètre et variables

```
set PAQUETS = 1..P;
```

L'ensemble *PAQUETS* est l'ensemble des paquets contenant les transistors. Il y a un nombre *P* de paquets, avec *P* qui vaut 3.

```
param P >= 0;
```

Initialisation du paramètre *P* correspondant aux 3 paquets dans lesquels les transistors seront affectés.

```
var disp {PAQUETS} >= 0;  
var dh {PAQUETS} >= 0;  
var dv {PAQUETS} >= 0;  
var maxhi {PAQUETS} >= 0;  
var minhi {PAQUETS} >= 0;  
var maxvi {PAQUETS} >= 0;  
var minvi {PAQUETS} >= 0;
```

Les variables dépendent maintenant de l'ensemble des paquets car nous ne souhaitons plus la dispersion d'un unique ensemble de transistors mais une dispersion en fonction de paquets de transistors.

```
var est_affecte {TRANS, PAQUETS} binary;
```

La variable *est_affecte* est un tableau binaire (0 ou 1) : 1 si le transistor est affecté à un paquet et 0 sinon. Elle doit donc dépendre du transistor mais aussi du paquet.

```
var moyenne_disp >= 0;
```

La variable *moyenne_disp* correspond à la moyenne de la dispersion des 3 paquets.

4.3.2 Contraintes

Contrainte 1 : Les anciennes contraintes sont modifiées afin que les variables dépendent de l'ensemble PAQUET. De plus, on remplace `est_conserve` par `est_affecte` :

```
subject to dhfe {p in PAQUETS} : dh[p] = maxhi[p] - minhi[p];
subject to dvbe {p in PAQUETS} : dv[p] = maxvi[p] - minvi[p];

subject to maxhfei {t in TRANS, p in PAQUETS} : maxhi[p] >= hfe[t] *
est_affecte[t,p];
subject to minhfei {t in TRANS, p in PAQUETS} : minhi[p] <=
est_affecte[t,p] * hfe[t] + (1 - est_affecte[t,p]) * MAXHFE;

subject to maxvbei {t in TRANS, p in PAQUETS} : maxvi[p] >= vbe[t] *
est_affecte[t,p];
subject to minvbei {t in TRANS, p in PAQUETS} : minvi[p] <=
est_affecte[t,p] * vbe[t] + (1 - est_affecte[t,p]) * MAXVBE;

subject to maxdh {p in PAQUETS} : disp[p] = dh[p] / MAXHFE;
subject to maxdv {p in PAQUETS} : disp[p] = dv[p] / MAXVBE;
```

Contrainte 2 : la moyenne de la dispersion des 3 paquets dans lesquels les transistors ont été affectés (somme des dispersions de tous les paquets / nombre de paquets).

```
subject to moyenne : sum {p in PAQUETS} disp[p] / P = moyenne_disp;
```

Contrainte 3 : un transistor, doit être affecté dans un paquet au minimum et au maximum. C'est à dire que si on additionne les affectations d'un transistor, on doit trouver 1 (0+0+1 ou 0+1+0 ou 1+0+0 avec le premier chiffre de l'addition, l'affectation au paquet 1, le deuxième celle du paquet 2, et le troisième celle du paquet 3).

```
subject to affectation_trans {t in TRANS} : sum {p in PAQUETS}
est_affecte[t,p] = 1;
```

Contrainte 4 : pour tout paquets, la différence entre deux paquets ne peut être supérieur à un. C'est à dire que si on prend deux groupes différents (quels que soient les deux groupes) et qu'on compte leur nombre de transistors, la différence d'effectif doit être inférieur ou égale à 1.

```
subject to effectif_paquet {p1 in PAQUETS, p2 in PAQUETS} : (sum {t1
in TRANS} est_affecte[t1, p1]) - (sum {t2 in TRANS} est_affecte[t2,
p2]) <= 1;
```

4.3.3 Objectif

```
minimize dispersion : moyenne_disp;
```

La dispersion est maintenant la moyenne des 3 dispersions des paquets.

On souhaite que la minimisation de la dispersion se fasse sur les paquets et non sur un unique ensemble de transistors.

4.3.4 Données

```
param MAXHFE := 600;
param MAXVBE := 1.0;
param P := 3; <- le nombre de paquets à créer

param : TRANS : hfe vbe type =
T1 369.713399 0.505002 npn
T2 172.445927 0.527908 pnp
T3 388.349744 0.678436 pnp
T4 345.246256 0.506357 pnp
T5 159.318987 0.601071 pnp
...

solve;
display est_affecte; <- notre tableau binaire pour connaître l'affectation de chaque transistor
display dispersion; <- la dispersion moyenne des trois paquets
display disp; <- la dispersion réelle des trois paquets
```

4.3.5 Résultat

```
Gurobi 8.1.0: optimal solution; objective 0.1939414667
7133 simplex iterations
207 branch-and-cut nodes
plus 19 simplex iterations for intbasis
est_affecte [*,*]
:      1      2      3      :=
T1      1      0      0
T10     0      1      0
T11     0      0      1
T12     0      0      1
T13     0      1      0
T14     1      0      0
T15     0      0      1
T16     0      0      1
T17     0      1      0
T18     0      1      0
T19     0      0      1
```

T2	0	1	0
T20	1	0	0
T21	0	1	0
T22	0	0	1
T23	0	1	0
T24	1	0	0
T25	0	1	0
T26	0	1	0
T27	0	0	1
T28	1	0	0
T29	1	0	0
T3	0	0	1
T30	0	0	1
T31	0	0	1
T32	1	0	0
T4	1	0	0
T5	0	1	0
T6	1	0	0
T7	0	1	0
T8	0	0	1
T9	1	0	0

;

dispersion = 0.193941

On observe que la dispersion est beaucoup plus faible que les deux précédentes. De plus si on additionne les 3 dispersions, la dispersion totale est bien inférieure à la dispersion de la Q2. Ensuite nous pouvons voir que chaque transistor est affecté à exactement un groupe et que les effectifs des groupes sont bien de 11, 11 et 10 transistors. Nos hypothèses sont donc correctes. En affichant la dispersion moyenne nous pouvons avoir une idée plus claire sur la minimisation de notre dispersion. En affichant les dispersions réelles nous pouvons vérifier que la moyenne est correcte. En conclusion notre dispersion tend bien vers 0, c'est un succès.

Question 5. NON RESOLUE

5.1 Hypothèse

Objectif : Nous devons maintenant créer les meilleures paires de transistors possible (c'est à dire dont la dispersion est la plus faible).

Hypothèse : Afin de créer les meilleures paires possibles il faut que les critères suivants soient le plus proche possible : la valeur de hfe, la valeur de vbe, le type de transistor (pnp ou npn). Ainsi on pourra avoir des paire de transistors les plus identiques possible. En respectant toutes ces contraintes on peut supposer que la valeur de dispersion de chaque paire sera quasi nulle.

5.2 Modélisation

Description du modèle :

Nous allons créer deux ensembles correspondant aux types de transistors possibles : npn et pnp. Dans chaque ensemble séparément nous allons pouvoir comparer les valeurs de hfe et vbe afin de pouvoir faire les paires.

Ensembles ajoutés/modifiés au modèle de la Q4 :

- NPN
l'ensemble des transistors de type npn
- PNP
l'ensemble des transistors de type pnp
- PAIRE
l'ensemble PAQUETS est renommé PAIRES pour une meilleure compréhension

Paramètres ajoutés/modifiés au modèle de la Q4 :

- On modifie P
Le nombre de paquets à créer qui vaudra : nombre de transistors / 2

Variables ajoutés/modifiés au modèle de la Q4 :

- Suppression de est_affecte
- binome_npn
Pour savoir dans quel binome se trouve les transistors de l'ensemble npn
- binome_pnp
Pour savoir dans quel binome se trouve les transistors de l'ensemble pnp
- disp
Dépend des ensembles NPN et PNP en plus de l'ensemble PAIRE
- les variables doivent dépendre des ensembles NPN ou PNP (en fonction du transistor) en plus de l'ensemble PAIRE

Contraintes à respecter modifiées et ajoutées au modèle de la Q4 :

- Modification de l'effectif par paquet
la différence d'effectif entre chaque paquet doit être de 0
- Les contraintes précédentes doivent être modifiées en incluant les nouveaux ensembles en fonction des variables utilisées
- Les contraintes doivent être modifiées pour prendre en compte binome_npn ou binome_pnp et non plus est_affecte

5.3 Réalisation en AMPL

5.3.1 Ensemble, paramètre et variables

```
set NPN within TRANS := {t in TRANS : type[t] = "npn"}  
set PNP within TRANS := {t in TRANS : type[t] = "pnp"}
```

Les deux nouveaux ensembles dans lesquels les transistors seront mis en fonction de leur type (pnp ou npn).

```
var binome_npn {TRANS, NPN} binary;
var binome_pnp {TRANS, PNP} binary;
```

Création d'un tableau binaire pour chaque ensemble : à chaque colonne on trouvera un 0 ou un 1 en fonction de si le transistor appartient ou non à ce binôme.

5.3.2 Contraintes

Contrainte 1 :

```
subject to effectif_paquet {p1 in PAQUETS, p2 in PAQUETS} :
    (sum {t1 in TRANS} est_affecte[t1, p1]) - (sum {t2 in TRANS}
est_affecte[t2, p2]) = 0;
```

On modifie l'effectif des paquets : comme il doit y avoir deux transistor par paquet et que notre ensemble initial de transistor en contient un nombre pair, la différence d'effectif entre chaque groupe doit être de zéro.

Attention : comme vous pouvez le voir nous retrouvons le binary « est_affecte » et non les nouveaux binary « binome_npn » et « binome_pnp » car nous n'avons pas réussi à trouver comment faire (cf la conclusion expliquant nos difficultés).

5.3.3 Objectif

```
minimize dispersion : moyenne_disp;
```

L'objectif ne change pas de la question précédente.

5.3.4 Données

```
data;
param MAXHFE := 600;
param MAXVBE := 1.0;
param P := card(TRANS)/2; <- le nombre de binômes à créer

param : TRANS : hfe vbe type =
T1 369.713399 0.505002 npn
T2 172.445927 0.527908 pnp
T3 388.349744 0.678436 pnp
T4 345.246256 0.506357 pnp
T5 159.318987 0.601071 pnp
...

solve;
display binome_npn; <- on affiche le tableau montrant les binomes dans l'ensemble NPN
display binome_pnp; <- on affiche le tableau montrant les binomes dans l'ensemble PNP
```

```
display dispersion; <- on affiche la dispersion moyenne des binômes
display disp; <- on affiche la dispersion réelle pour vérifier nos resultats
```

5.3.5 Résultat

X

CONCLUSION :

Nous avons rencontré les difficultés suivantes lors de notre devoir maison :

- Dans la question 2 : l'utilisation du binary (est-conserve) ne nous a pas semblé évidente. Après avoir longuement réfléchi nous nous sommes appuyés sur l'exemple des étudiants et des créneaux horaires dans le pdf du CM pour trouver la réponse. Ensuite, comprendre comment faire les contraintes avec « est_conserve » pour récupérer les hfe et vbe minimaux $(\text{est_conserve}[t] * \text{hfe}[t] + (1 - \text{est_conserve}[t]) * \text{MAXHFE})$ a été assez compliqué. En effet, le fait de mettre 0 dans le max et le max dans le min pour que les valeurs ne soient pas conservées dans le cas où la transistor n'était pas conservé nous a pris du temps et beaucoup d'essais. Après réflexion nous aurions pu faire des calculs plus simples si le transistor exclu prenait la valeur 1 et celui conservé la valeur 0.
- Dans la question 4 : Dans un premier temps nous n'avions pas pensé à faire dépendre toutes les variables de l'ensemble PAQUETS. Nous cherchions des contraintes supplémentaires à écrire pour relier la dispersion aux paquets. Ensuite la difficulté a été de trouver comment exprimer les contraintes sur la taille des paquets (chaque paquet doit contenir le même nombre de transistors à 1 près). Notre première idée a été de faire les contraintes suivante :
« subject to effectif_paquet {p in PAQUETS} : $\sum \{t \text{ in TRANS} \mid \text{est-affecte}[t,g]\} \geq (\text{card}(\text{TRANS})/P)$ »
«subject to effectif_paquet {p in PAQUETS} : $\sum \{t \text{ in TRANS} \mid \text{est-affecte}[t,g]\} \leq (\text{card}(\text{TRANS})/P+1)$ ».
Mais malgré les essais de modification de syntaxe nous n'avons pas réussi à la faire fonctionner correctement et sommes donc parti sur un autre raisonnement pour arriver à notre contrainte actuelle.
- Dans la question 5 : Nous avons créé deux ensembles correspondants au type de chaque transistor. Le problème rencontré est le suivant : comment faire pour que les variables dépendent SOIT du premier ensemble SOIT du deuxième en fonction de son type ? De même pour les contraintes en utilisant binome_npn OU binome_pnp. Une solution serait de créer deux fois chaque variable et contrainte. Chaque exemplaire serait attaché à un ensemble et on pourrait travailler avec un binary à la fois. Malheureusement nous avons un nombre limité de variables et contraintes à créer et en faisant ainsi nous dépassons largement ce seuil. Nous n'avons donc pas pu résoudre cette question. Nous avons ainsi fait des modifications de code dans le fichier de la question 5 sans pour autant toucher aux affectations des variables et sans remplacer le binary de la question 4 par les nouveaux.