

PROJET "PLANIFICATION DES COMMANDES"

MODSIM C : INTRODUCTION À L'OPTIMISATION DISCRETE"

LOUIS DAGE & MOHAMED AYADI

IFIE MI PROMO 2022

Projet « Planification de commandes » par Louis Dage et Mohamed Ayadi

Dans ce projet la problématique est la suivante : Comment optimiser un portefeuille de 20 commandes tout en respectant une charge horaire imposé. Pour cela nous disposons des informations suivantes : le temps d'occupation, temps de pause ou le temps de réglage par exemple.

Dans un premier temps, nous utiliserons une fonction objectif dite simple qui prendra comme paramètre d'entrée le taux d'occupation et le taux performance. Puis dans une dernière partie, nous prendrons aussi en compte le taux de classe client, le taux de service et le taux financier.

Cet ensemble de critère est représenté dans la fonction objectif qu'il faut maximiser afin d'atteindre l'objectif souhaité.

SOMMAIRE

- Preparation du planning - Mise en place des boutons de commandes utiles
- Premier jet d'optimisation dit "A la main" avec une fonction objectif simple
- Optimisation "A la main" avec une fonction objectif complexe
- Optimisation avec EVOLVER de la fonction objectif simple
- Optimisation avec EVOLVER de la fonction objectif complexe
- Analyse et conclusion

Pour établir cette optimisation, nous allons dans un premier temps créer des boutons de commande grâce à un codage en VBA pour faciliter notre étude. Dans un second temps, nous allons optimiser manuellement le portefeuille de commandes pour une fonction simple et pour une fonction objectif complexe. Puis dans un dernier temps, nous optimiserons avec l'algorithme génétique EVOLVE.

PREPARATION DU PLANNING

MISE EN PLACE DES
COMMANDES UTILE À
LA MISE EN PLACE DU
PLANNING

PORTFOLIO des COMMANDES en attente

Cde	Charge [min.]	Classe Client	Date Cde.	CA [€]	Coût [€]	Placement	Ligne planning
1	67	3	19/09/2018	3 064,00	2 170,00	0	
2	4	3	19/09/2018	981,00	2 540,00	0	
3	57	2	19/09/2018	1 420,00	870,00	0	
4	58	2	19/09/2018	2 243,00	1 560,00	0	
5	58	3	20/09/2018	2 330,00	2 080,00	0	
6	8	2	20/09/2018	1 012,00	1 080,00	0	
7	29	3	21/09/2018	1 789,00	980,00	0	
8	26	3	21/09/2018	1 246,00	1 760,00	0	
9	68	3	21/09/2018	1 787,00	1 140,00	0	
10	56	3	21/09/2018	1 640,00	2 060,00	0	
11	10	3	23/09/2018	1 574,00	860,00	0	
12	46	3	23/09/2018	2 182,00	960,00	0	
13	42	3	22/09/2018	2 208,00	1 920,00	0	
14	36	2	22/09/2018	2 075,00	1 300,00	0	
15	44	3	22/09/2018	2 812,00	2 940,00	0	
16	19	3	22/09/2018	958,00	650,00	0	
17	61	2	22/09/2018	2 026,00	2 110,00	0	
18	79	3	23/09/2018	2 528,00	2 250,00	0	
19	63	2	23/09/2018	2 084,00	1 130,00	0	
20	9	3	23/09/2018	1 418,00	1 590,00	0	

COMMANDES PLANIFIEES

Cde	Charge [min.]	Classe Client	Date Cde.	CA [€]	Coût [€]	Délai / Cde [jours]	CA - Coût [€]
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							

NCP
824 37 500,00 27 740,00 0 Ordonnancement

charge totale capa-charge 640 Carnet de commande cumul 0 0

Placement Cde	Effacement Planning	Raz-co. Placement
---------------	---------------------	-------------------

Bouton coder avec VBA

Données	
Capacité disponible	470
à la date du	05/10/2018
Temps d'ouverture	600
Temps de pause	130
Temps de réglage	30
Temps requis	470
Temps de fermeture	440
Temps d'occupation	0

Indicateurs %

Taux d'occup.	Taux de performance	Taux service	Taux classe client	Taux financier	Ft. Obj.
Valeur 0,0%	0,0%	#DIV/0!	0,00%	0,00%	Valeur 0,7083 REDEV/01
Cible 70,0%	100,0%	0,33%	100,00%	100,00%	Cible 0,7083 1,0000

Dans cette partie on va mettre en place grâce à un codage VBA trois boutons :

- Un bouton placer commande qui permet de placer les commandes dans un certains ordre que l'on donne grâce à une numérotation sur le tableau de planification des commandes.
- Un bouton qui va effacer le planning que l'on aura crée précédemment avec le bouton placement commande.
- Le dernier bouton permettra d'effacer la colonne de placement qui permet d'ordonner le planning de commandes.

Sélectionner les commandes

POINTEAUX de COMMANDES en attente						
Id	Charge Client (date)	Date calc.	CA (€)	Coût (€)	Placement	Ligne planning
1	47	2	18/09/2018	0.064,00	0	
2	4	3	18/09/2018	0,012,00	0	
3	87	2	18/09/2018	1.930,00	0	
4	58	2	18/09/2018	2.742,00	1	
5	94	3	18/09/2018	2.830,00	2	
6	8	2	18/09/2018	0.050,00	3	
7	49	3	18/09/2018	1.798,00	0	
8	76	3	18/09/2018	1.746,00	0	
9	84	2	18/09/2018	2.787,00	1	4
10	95	3	18/09/2018	1.840,00	2	
11	38	2	18/09/2018	1.514,00	0	
12	48	1	18/09/2018	2.002,00	0	
13	42	3	18/09/2018	2.206,00	1	5
14	80	2	18/09/2018	2.075,00	0	
15	44	3	18/09/2018	2.812,00	1	6
16	28	1	18/09/2018	958,00	0	
17	41	2	18/09/2018	2.040,00	1	7
18	72	3	18/09/2018	2.228,00	0	
19	63	1	18/09/2018	2.094,00	0	
20	9	3	18/09/2018	1.936,00	0	

Saisie d'un = 1

Attribution d'une ligne planning par ordre croissant

```

Application.ScreenUpdating = False

Dim NoCell As Long
Dim NoLig As Long
Dim Appartient As Boolean
Dim compteur As Long
Dim i As Long

NoLig = Target.Row
NoCell = Target.Column

If NoCell = 5 And NoLig > 2 And NoLig < 22 Then
    Appartient = True
Else
    Appartient = False
End If

If Appartient = True Then
    compteur = 1
    For i = 2 To 22
        If Cells(i, NoLig).Value = 1 Then
            Cells(i, NoLig).Value = compteur
            compteur = compteur + 1
        ElseIf Cells(i, NoLig).Value = 0 Then
            Cells(i, NoLig).Value = Empty
        End If
    Next
End If

```

Application.ScreenUpdating = False

Pour faciliter l'ordonnancement des commandes on crée un programme qui permet de créer un ordre dans la colonne « ligne planning ». On place des « 1 » dans la colonne placement et à chaque fois que l'on remplit une des lignes avec « 1 », la ligne de planning se remplit avec le numéro correspondant à l'ordre dans lequel doit passer la commande.

Nous avons réalisé cette manœuvre avec un code VBA. Au début du planning définit les variables du programme. Puis on crée une boucle if pour vérifier que la valeur prise par notre code est bien dans la colonne « placement » grâce à la variable « appartient ».

Puis dans une deuxième boucle if on établit un compteur qui compte le nombre de « 1 » qu'il y'a déjà dans la colonne et qui en fonction de cela attribue à la commande un numéro correspondant à son ordre de passage dans le carnet de commande.

Placer les commandes

PORTEFEUILLE de COMMANDES en attente

Cde	Charge (min.)	Classe Client	Date Cde.	CA (€)	Coût (€)	Placement	Ligne planning
1	67	1	19/09/2018	3 064,00	1 170,00	1	1
2	4	3	19/09/2018	861,00	1 540,00	1	2
3	37	1	19/09/2018	1 420,00	870,00	0	
4	58	2	19/09/2018	2 743,00	1 580,00	1	3
5	58	3	20/09/2018	2 330,00	2 080,00	0	

↓

Placement Cde.

COMMANDES PLANIFIES

Cde	Charge (min.)	Classe Client	Date Cde.	CA (€)	Coût (€)	Délai / Cde (jours)	CA - Coût (€)
1	67	1	19/09/2018	3 064,00	1 170,00	16	1894
2	2	4	19/09/2018	861,00	1 540,00	16	-679
3	4	58	19/09/2018	2 743,00	1 580,00	16	1163
4						0	
5						0	

```
Dim compteur As Long
Dim i As Long
Dim Lig As Long
Dim NumLig As Long
```

Définition des variables

```
compteur = 1

For i = 2 To 22
    If Cells(i, 8).Value = 1 Then
        Cells(i, 9).Value = compteur
        compteur = compteur + 1
    ElseIf Cells(i, 8).Value = 0 Then
        Cells(i, 9).Value = ""
    End If
Next
```

Associe les cellules du portefeuille de commandes au tableau de Commandes planifiées

```
For NumLig = 3 To 22
    Lig = Cells(NumLig, 9)
    If Lig > 0 Then
        Cells(Lig + 2, 11) = Cells(NumLig, 2)
        Cells(Lig + 2, 12) = Cells(NumLig, 3)
        Cells(Lig + 2, 13) = Cells(NumLig, 4)
        Cells(Lig + 2, 14) = Cells(NumLig, 5)
        Cells(Lig + 2, 15) = Cells(NumLig, 6)
        Cells(Lig + 2, 16) = Cells(NumLig, 7)
    End If
Next NumLig
```

Associe les cellules du portefeuille de commandes au tableau de Commandes planifiées

```
End Sub
```

Maintenant que l'on a codé la colonne « placement » et « ligne planning » on crée un bouton VBA qui permet quand l'on appuie dessus de copier-coller les lignes sélectionnées dans l'ordre dans le carnet de commande.

Pour cela dans notre programme on préleve les lignes qui ont été sélectionnées et pour les lignes sans attribution d'ordre on ne colle rien.

Effacer le carnet de commandes

The diagram illustrates the process of clearing a table of planned orders. It shows two versions of a table titled "COMMANDES PLANIFIEES". The first table contains data with rows numbered 1 to 5. The second table is identical in structure but has all cells in the data range cleared, containing only zeros. A red arrow points from the first table to a button labeled "Effacement Planning". A callout box contains the following VBA code:

```
Sub Bouton_EffacementPlanning()
Application.ScreenUpdating = False
Range("K3:P22").ClearContents
End Sub
```

Effacement des cellules de la plage [K3:P22] des Commandes planifiées

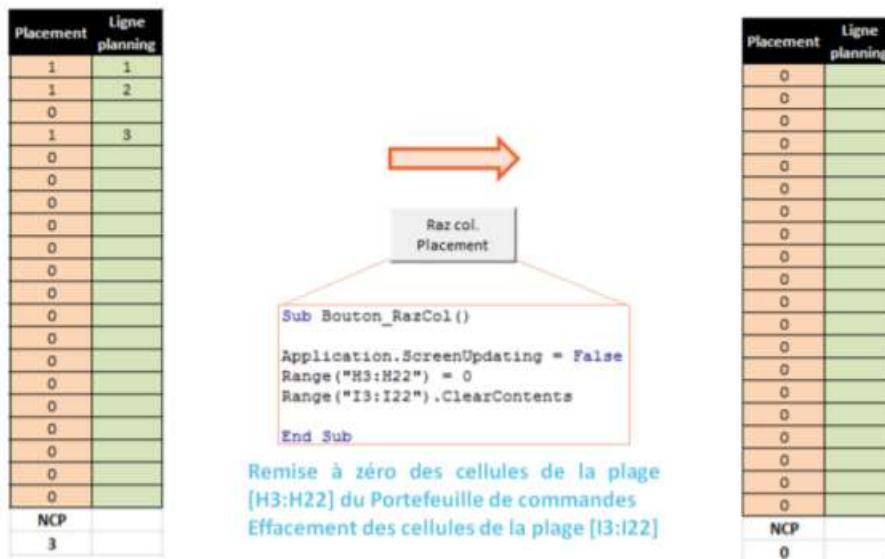
A lightbulb icon with the text ".ClearContents" is also present.

COMMANDES PLANIFIEES							
Cde	Charge (min.)	Classe Client	Date Cde.	CA (€)	Coût (€)	Délai / Cde (jours)	CA - Coût (€)
1	67	1	19/09/2018	9 064,00	1 170,00	16	1894
2	4	3	19/09/2018	861,00	1 540,00	16	-679
3	58	2	19/09/2018	2 743,00	1 580,00	16	1163
4							0
5							0

COMMANDES PLANIFIEES							
Cde	Charge (min.)	Classe Client	Date Cde.	CA (€)	Coût (€)	Délai / Cde (jours)	CA - Coût (€)
1							0
2							0
3							0
4							0
5							0

Ce bouton a simplement pour but d'effacer le carnet de commandes qui a été rempli au préalable avec le bouton « placer commande ». Pour cela, il suffit d'utiliser la fonction « clearContents » de VBA en sélectionnant l'étendue du tableau que l'on veut supprimer. On peut aussi utiliser une macro pour cela.

Remettre à zéro la colonne placement



Le but de ce bouton est comme le précédent de supprimer une partie du tableau mais cette fois il supprime les colonnes « Placement » et « Ligne planning ».

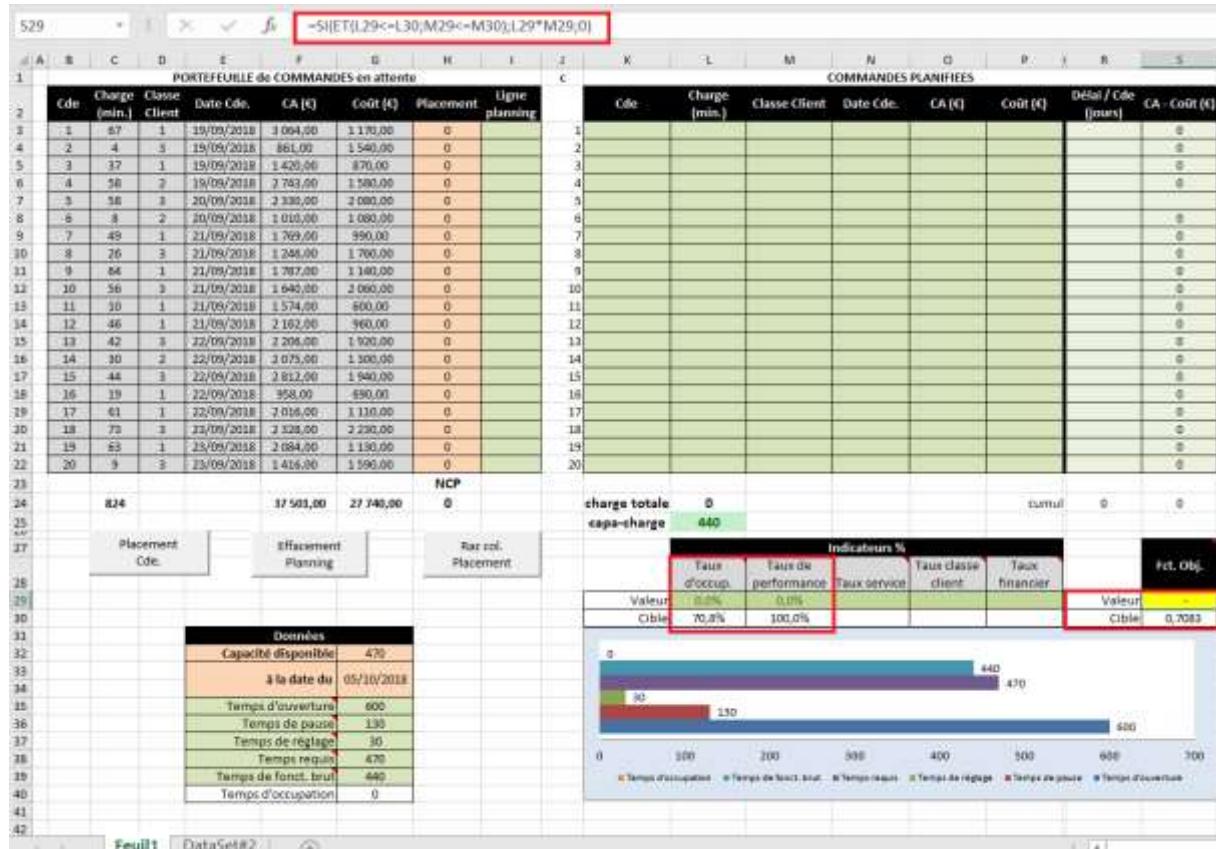
Ces boutons rendent l'expérience d'optimisation plus agréable puisqu'il permette d'accélérer le processus d'essai. Ainsi, on peut essayer, se tromper et recommencer rapidement. De plus, lors de l'utilisation de l'algorithme EVOLVER il s'avèrera pratique.



PREMIÈRE OPTIMISATION :
A LA MAIN

FONCTION OBJECTIF
SIMPLE

Critère d'optimisation



Pour l'instant, on va concentrer notre étude sur l'optimisation d'une fonction objectif simple qui prend en compte seulement deux indicateurs :

-Le taux d'occupation qui représente la charge totale divisé par le temps total d'ouverture. Un haut taux d'occupation signifie que le temps d'ouverture de l'entreprise est optimisé.

-Le taux de performance représente la charge totale divisé par le temps de fonction brut de l'entreprise. Un haut taux de performance signifie que le temps de machine de l'entreprise est optimisé.

On prend donc comme fonction objectif la multiplication de ces deux indicateurs si et seulement si leurs valeurs sont inférieures à la valeur cible. Car on ne peut pas avoir techniquement un taux d'occupation supérieur à 70,8% alors avoir une valeur au-dessus serait incohérent.

Conclusion

3 solutions	Charges ↗	Charges ↓	Prix
Pourquoi ?	Paramètre lié aux 2 indicateurs		Vision économique
Comment ?	Charges les plus ↗	Charges les plus ↓ (<46)	Rapport $\frac{CA - Coût}{Charge}$ ↗ Charge totale ≤ 305
N° commandes	1 – 10 – 14 – 17 – 18 – 19	2 – 3 – 6 – 8 – 11 – 12 – 13 – 14 – 15 – 16 – 20	1 – 3 – 7 – 11 – 12 – 14 – 15 – 16
Résultats	Taux d'occupation = 58,3% Taux de performance = 100,0% Fonction objectif = 0,5833	Taux d'occupation = 45,8% Taux de performance = 100,0% Fonction objectif = 0,4583	Taux d'occupation = 50,3% Taux de performance = 94,4,0% Fonction objectif = 0,4750

Voici un tableau récapitulatif avec 3 solutions d'optimisations :

- Une en priorisant les charges les plus élevées (>46) pour optimiser le taux d'occupation et le taux de performance. On peut observer que la fonction objective atteint 0.5833 ce qui paraît pas mal puisque le taux de performance est à son maximum et le taux d'occupation à 58.3%.
Cette solution priorise peu de commandes mais avec des charges assez hautes.
- La deuxième solution priorise un grand nombre de commandes mais avec des faibles charges horaires. Le taux de performance atteint alors 100% Mais on obtient un taux d'occupation des machines plus faibles et donc la fonction objectif est plus faible.
- Pour la dernière solution, on a essayé d'aborder le problème dans une vision plus économique de la chose. On a donc un taux de performance plus faible et donc une fonction objectif plus basse mais cela n'est pas surprenant puisque notre fonction objectif ne prends pas compte la taux financier des commandes.



PREMIÈRE OPTIMISATION : A LA MAIN

FONCTION OBJECTIF
COMPLÈXE

Introduction de nouveaux indicateurs

$$\checkmark \text{ TAUX D'OCCUPATION} = \frac{\text{Charge totale}}{\text{Temps d'ouverture}}$$

$$\checkmark \text{ TAUX DE PERFORMANCE} = \frac{\text{Charge totale}}{\text{Temps de fonctionnement brut}}$$

INDICATEURS DE DÉCISION

$$\checkmark \text{ TAUX DE SERVICE} = \frac{\text{Nombre de commandes}}{\text{Cumul Délai/Cde}}$$

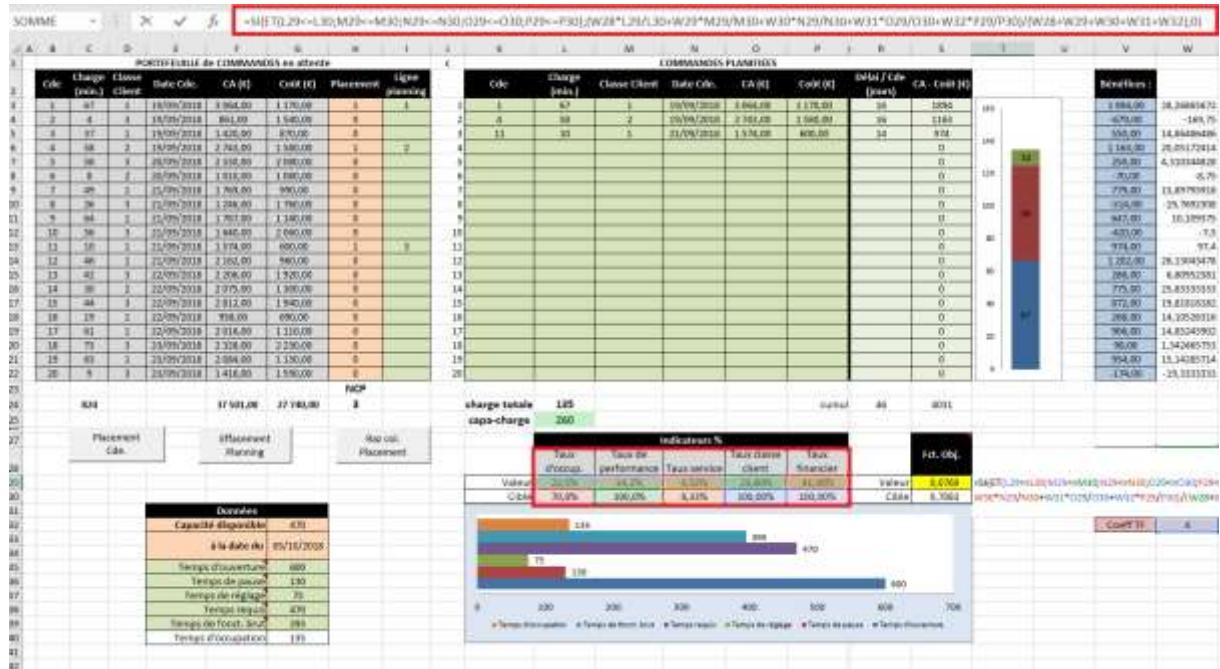
$$\checkmark \text{ TAUX DE CLASSE CLIENT} = 0,06 \times \text{Nombre de clients 1} + 0,116 \times \text{Nombre de clients 2} + 0,014 \times \text{Nombre de clients 3}$$

$$\checkmark \text{ TAUX FINANCIER} = \frac{\text{Cumul (CA-Coût)}}{\text{Cumul CA - Cumul Coût}}$$

On ajoute cette fois trois indicateurs qui semblent pertinents au vu de notre étude :

- Le taux de service qui traduit le nombre de commandes faites en fonction du délai de la commande. Globalement, cela permet d'indiquer qu'une commande passée est livrée assez rapidement en fonction de sa date de commande.
- Le taux de classe client traduit la classe auquel le client appartient. Un client de type 1 est un client privilégié. Un client de type 2 est un client à conquérir ce qui veut dire qu'il faut le privilégier pour qu'il reste un client à nous. Un client de type 3 est un client annexe : il n'est ni privilégié ni à privilégier.
- Le taux financier lui apporte une vision économique à la fonction objectif. Elle permet d'évaluer ce qu'apporte la commande passée à l'entreprise d'un point de vue financier.

Instauration d'une fonction objectif complexe



On implémente la nouvelle fonction objectif complexe qui prends en compte les trois nouveaux indicateurs. Nous avons augmenté l'importance du taux financier en implémentant des coefficients et en augmentant le coefficient du taux financier par rapport aux autres coefficients.

Optimisation manuelle avec la fonction objectif complexe

3 solutions	 Charges ↗	 Charges ↘	 Prix
Pourquoi ?	Paramètre lié aux 2 indicateurs		Vision économique
Comment ?	Charges les plus ↗	Charges les plus ↘ (<46)	Rapport $\frac{CA - Coût}{Charge}$ ↗ Charge totale ≤ 305
N° commandes	1 – 10 – 14 – 17 – 18 – 19	2 – 3 – 6 – 8 – 11 – 12 – 13 – 14 – 15 – 16 – 20	1 – 3 – 7 – 11 – 12 – 14 – 15 – 16
Résultats	Taux d'occupation = 58,3% Taux de performance = 100,0% Taux de service = 7,50% Taux de classe client = 32,40% Taux financier = 43,10% Fonction objectif = 0,6772	Taux d'occupation = 45,8% Taux de performance = 100,0% Taux de service = 7,19% Taux de classe client = 54,20% Taux financier = 35,75% Fonction objectif = 0,6166	Taux d'occupation = 50,3% Taux de performance = 94,4,0% Taux de service = 7,08% Taux de classe client = 49,00% Taux financier = 74,93% Fonction objectif = 0,7790

Pour optimiser manuellement notre fonction objectif, plus complexe, nous avons repris les 3 solutions présentées précédemment (mêmes numéros de commandes). La fonction objectif est la meilleure lorsque l'on optimise en fonction du prix : en effet cela s'explique par le fait qu'un fort coefficient pondère le taux financier.



DEUXIÈME OPTIMISATION : EVOLVER L'ALGORITHME GÉNÉTIQUE

FONCTION OBJECTIF
SIMPLE

Dans cette partie on souhaite troquer la méthode archaïque de chercher des solutions en faisant des essais « à la main » par l'utilisation d'un algorithme génétique.

Pour cela nous utiliserons Evolver qui nous permettra d'intégrer un algorithme génétique dans Excel.

L'objectif est de trouver de manière plus fiable la manière optimale de placer les commandes.

Pour rappel, la méthode dites « d'algorithmes génétiques » est une méthode reprenant les principes darwiniens. Elle se base sur la simulation de la sélection naturelle sur une population choisi en contrôlant les taux de mutations et de croisements afin d'établir une solution optimisée en un minimum de génération possible.

Les Paramètres que nous prendront donc en compte dans cette partie seront :

- La « **Population** » : représente l'ensemble des individus nécessaires mis en cause dans l'algorithme. Ici les individus qui composent notre population sont des placements de commandes.
- Le « **nombre d'essais** » : Représente le nombre de croisements qui seront effectués entre les individus, soit le nombre de génération que l'on aura eu à la fin de l'algorithme
- Le « **taux de croisement** » : représente la proportion d'informations que les parents vont transmettre à la descendance après croisement
- Le « **taux de mutation** » : représente la probabilité qu'une information possédée par aucun des deux parents n'apparaisse après un croisement.

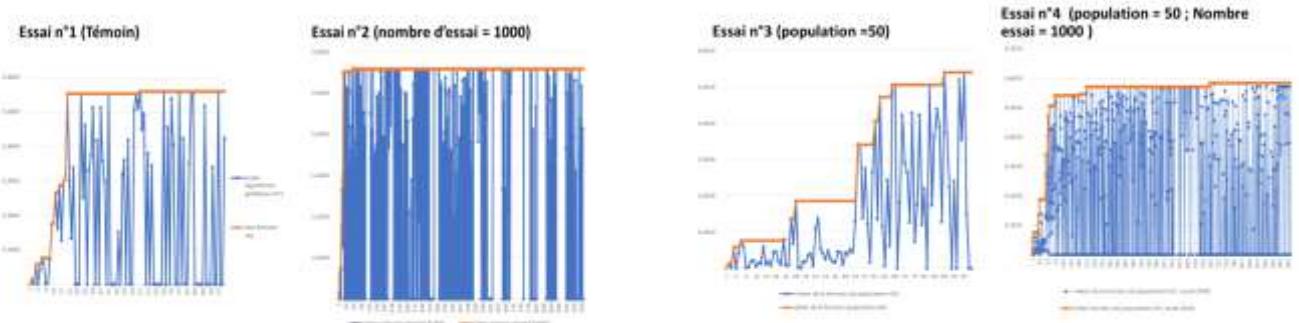
Optimisation avec evolver :

→ Avec une fonction simple pour commencer :

	Témoin (cas n°1)	Cas n°2	Cas n°3	Cas n°4	Cas n°5	Cas n°6	Cas n°7	Cas n°8
Nombre d'essais	100	1000	100	1000	100	100	100	100
Taille de la population	10	10	50	50	10	10	10	10
Taux de Croisement	0,5	0,5	0,5	0,5	0,9	0,25	0,5	0,5
Taux de Mutation	0,1	0,1	0,1	0,1	0,1	0,1	0,25	0,01

Ainsi nous avons effectué plusieurs tests afin d'identifier quels pouvaient être ceux qui étaient pertinents et surtout quels étaient ceux qui se révélaient être des facteurs clés de succès.

Influence du « Nombre d'essais » et de la taille de la « population » :



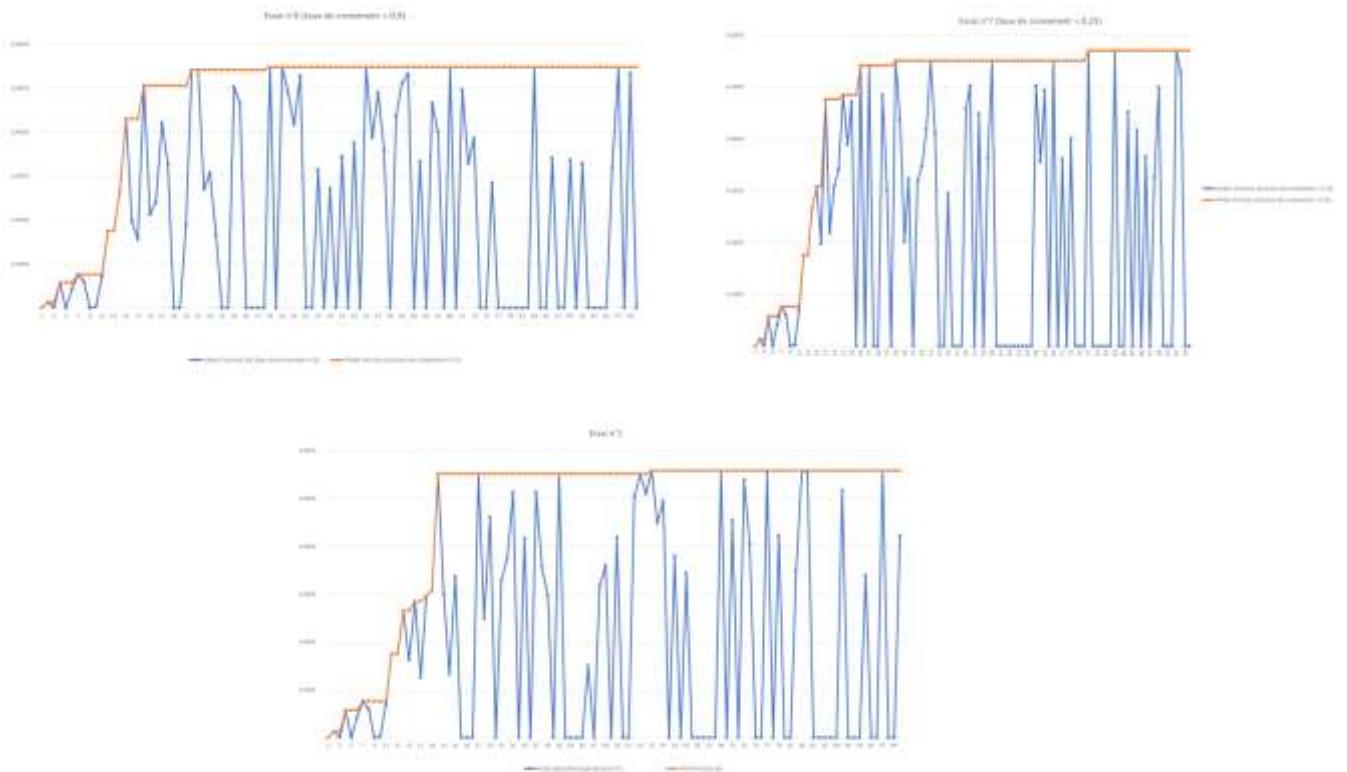
	Cas n°1	Cas n°2	Cas n°3	Cas n°4
Nombre d'essais	100	1000	100	1000
Population	10	10	50	50
Taux de Croisement	0.5	0.5	0.5	0.5
Taux de Mutation	0.1	0.1	0.1	0.1
Meilleur résultat de la fonction obj	0.5583	0.5583	0.5408	0.5833

Après avoir effectué ces tests on remarque dans un premier temps que les résultats sont tous plutôt similaire. On remarque également que les résultats des deux premiers tests sont exactement les mêmes, ce qui nous permet de conclure que l'augmentation du nombre d'essai seule, n'a pas un impact remarquable sur le résultat étant donné qu'on a multiplié le nombre d'essai par 10.

On remarque également que dans le cas n°3, on obtient un résultat moins bon que lors du premier test. Ceci peut s'expliquer par le fait que l'on a une population trop grande pour le nombre de croisement ce qui limite la diversité.

Ainsi, logiquement, lorsque l'on augmente les deux paramètres en même temps on obtient un meilleur résultat car on augmente à la fois le nombre de croisement et la diversité des individus qui se croisent.

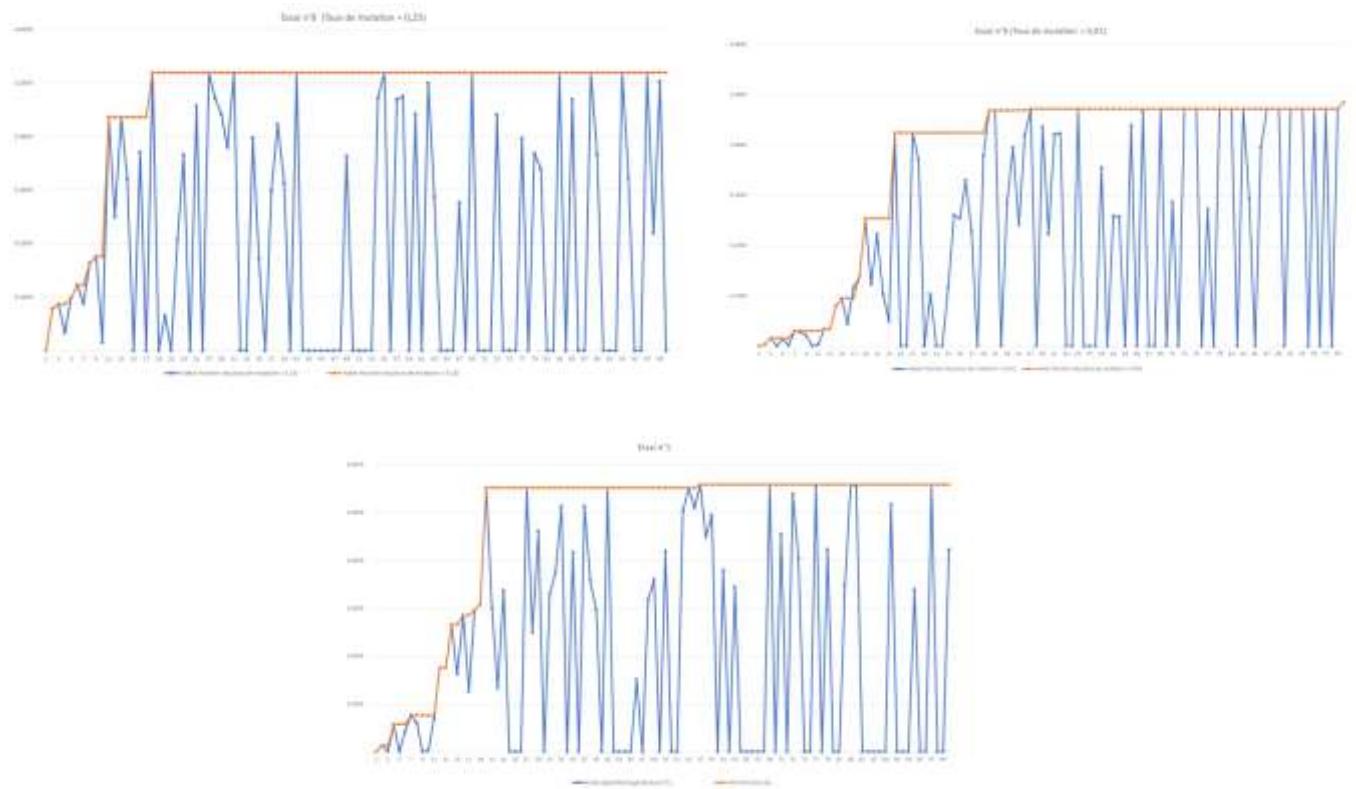
Influence du « taux de croisement » :



	Cas n°1	Cas n°5	Cas n°6
Nombre d'essais	100	100	100
Population	10	10	10
Taux de croisement	0.5	0.9	0.25
Taux de mutation	0.1	0.1	0.1
Max Fonction obj	0.5583	0.5484	0.5701

Dans les cas n°5 et n°6 notre étude s'est portée sur l'influence du **Taux de Croisement**. On remarque alors aisément que sans toucher à aucun autre des paramètres, on obtient un résultat plus faible dans le cas où l'on monte beaucoup le taux de croisement, ce qui s'explique par le fait qu'un trop grand brassage de l'information entraîne la non pérennisation de certains caractères voulus. A l'inverse si celui-ci n'est pas très élevé, cela semble permettre une meilleure transmission des caractères intéressants.

Influence du « taux de mutation » :



	Cas n°1	Cas n°7	Cas n°8
Nombre d'essais	100	100	100
Population	10	10	10
Taux de croisement	0.5	0.5	0.5
Taux de mutation	0.1	0.25	0.01
Max Fonction obj	0.5583	0.5190	0.4845

Enfin le dernier paramètres que nous avons étudié est le taux de mutation. On remarque aux résultats de la fonction obj que le changement seul de ce paramètre n'est pas pertinent puisque dans les deux cas on retrouve des résultats inférieurs aux résultats du cas témoin. Néanmoins cette étude nous révèle qu'un taux de mutation trop faible en comparaison avec le nombre d'essai est clairement contre-productif. Cela indique ainsi que la mutation est essentielle dans le processus d'évolution des individus.

Conclusion :

Ainsi chacun des paramètres se révèle être influant et une tendance se dégage alors pour pouvoir maximiser son résultat :

- Nombre d'essai élevé
- Population élevée
- Taux de croisement pas trop élevé (< 0.5)
- Taux de mutation pas trop élevé mais surtout pas trop faible en comparaison du nombre d'essais



DEUXIÈME OPTIMISATION : EVOLVER L'ALGORITHME GÉNÉTIQUE

FONCTION OBJECTIF
COMPLÈXE

Dans cette partie la fonction objectif devient un peu plus complexe et prend cette fois-ci en compte les trois nouveaux indicateurs définis plus tôt :

- Le taux de service
- Le taux de classe client
- Le taux financier

De plus tous les indicateurs de performances sont hiérarchisés en fonction de leur importance à travers l'attribution d'un coefficient chacun.

	Coeffs
Taux d'Occupation	2
Taux de Performance	2
Taux de Services	1
Taux de Classe Client	0.5
Taux Financier	3

Ces coefficients ont été choisis par nous en fonction de ce que nous estimions comme étant le plus important ou non.

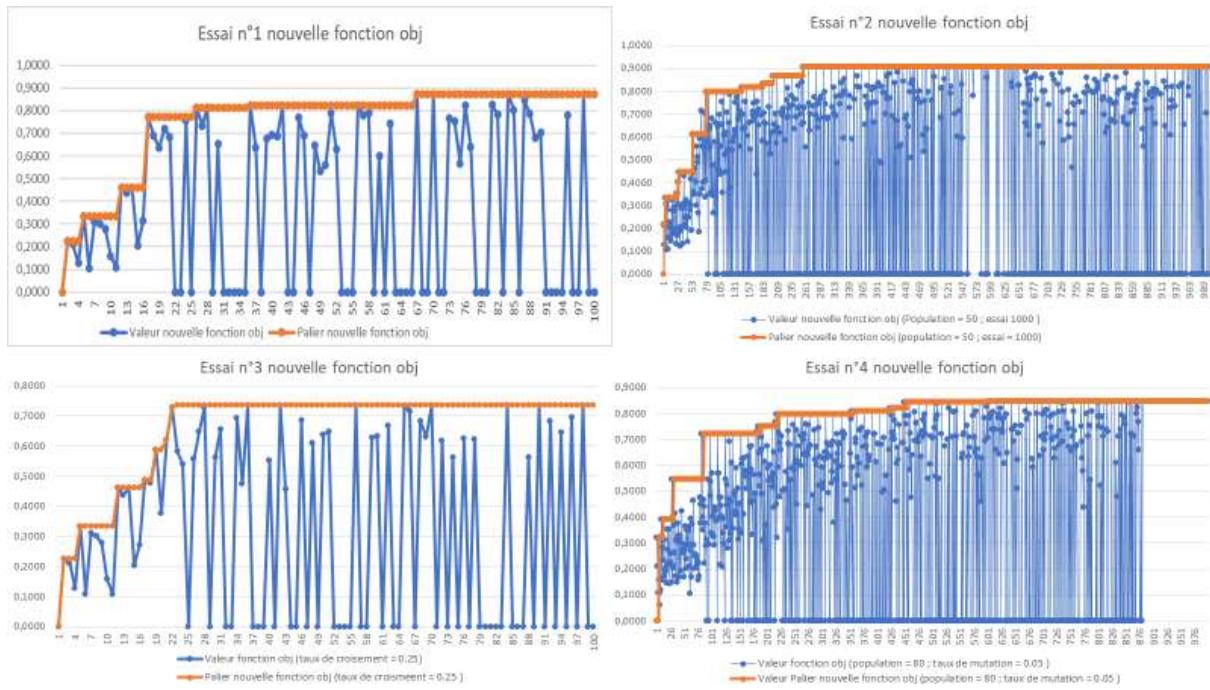
Enfin cette nouvelle fonction a pour valeur cible 1 et non plus 0.7083.

Optimisation avec evolver :

→ Avec une fonction complexe :

	Témoin (cas n°1)	Cas n°2	Cas n°3	Cas n°4
Nombre d'essais	100	1000	100	1000
Taille de la population	10	50	10	80
Taux de Croisement	0,5	0,5	0,25	0,5
Taux de Mutation	0,1	0,1	0,1	0,05

Pour cette partie nous avons choisi de faire plusieurs essais. Le premier essai est, de manière identique à la partie précédente, un test témoin permettant d'étudier l'influence des différents paramètres. Les cas suivants furent ensuite des tentatives d'optimisation du résultat en fonction des observations faites plus tôt.



Ainsi nous avons effectué ces quatre essais dont les résultats sont compilés ci-dessus et nous avons obtenu :

	Cas n°1	Cas n°2	Cas n°3	Cas n°4
Valeur de la nouvelle fonction obj	0.8723	0.9058	0.7372	0.8462

Les résultats obtenus sont ainsi bien meilleurs que ceux obtenus sur la précédente fonction obj, surtout dans le cas n°2.

Cependant ce n'est pas avec l'augmentation de la population ni un taux de mutation plus faible que l'on obtiendra un meilleur résultat qu'au cas n°2, comme en atteste le cas n°4.

De plus la baisse du taux de Croisement ne semble pas une solution viable non plus étant donné le cas n°2.

Par ailleurs l'augmentation conjointe du nombre d'essai et de la population n'est pas la solution pour avoir une meilleure solution puisque ce résultat est obtenu au 257^{ème} essai.

Ainsi notre hypothèse serait de d'augmenter légèrement les taux de Croisement et de Mutation.

ANALYSE ET CONCLUSION

Pour Conclure on peut dire que l'automatisation avec les macros vues dans les premières phases permets d'introduire à l'optimisation d'une solution.

Néanmoins cette solution a ses limites étant donnés les résultats obtenus avec Evolver. En effet l'optimisation manuelle relève du coup de chance est s'avère être donc très limité pour une utilisation récurrente du procédé.

Ainsi l'utilisation d'un algorithme génétique permet de s'approcher de manière plus sûre de la solution optimiser et permets de prendre en compte plusieurs indicateurs comme nous l'avons vu avec la deuxième fonction objectif.

Par ailleurs concernant le projet de placement commande, certaines limites apparaissent comme l'impossibilité de prévoir les temps de pause. En effet, si cela n'est pas gênant, cela le devient réellement sur plusieurs jours car on ne peut savoir si une tache est partageable sur plusieurs jours. Le fait de ne pas pouvoir prioriser les commandes est aussi un problème car les ordres de priorités de ces dernières sont les dates de commandes. Or il pourrait être avantageux d'influer également sur ce paramètre.