

## 4.1. Introduction

Ce chapitre est consacré à la réalisation et l’implémentation de différentes fonctionnalités de l’application. Il consiste à traduire la conception du système que nous avons effectué dans le chapitre précédent en code source exécutable. Donc nous expliquerons notre approche pour résoudre le problème de conception et gestion des indisponibilités des sites dans la chaîne logistique. Nous avons choisi d’Utiliser les algorithmes génétiques comme métaheuristique pour résoudre notre problème qui consiste à localiser les DCs et l’affectation des détaillants au DCs et la sélection des fournisseurs, Ensuite la gestion des indisponibilités des DCs.

Dans ce chapitre nous allons présenter l’environnement de développement et les différents outils utilisés pour réaliser notre application, puis expliquer son fonctionnement en présentant quelques interfaces illustratives.

## 4.2. Présentation de l’environnement et des outils de développement.

### 4.2.1. Environnement matériel :

Notre application va être réalisée sur une machine qui comporte les caractéristiques suivant :

- ↳ Marque : Dell.
- ↳ Modèle : INSPIRON 15.
- ↳ Processeur : Intel® Core™ i5-3337U CPU @ 1.80 GHz 1.80 GHz
- ↳ Mémoire installée (RAM): 4.00 Go.
- ↳ Type du système : Système d’exploitation 64bits.
- ↳ Système d’exploitation : Windows 10 édition professionnel.

### 4.2.2 Environnement de développement :

#### 4.2.2.1 Java NetBeans

Pour mettre en œuvre notre application, nous avons choisi comme langage de programmation le langage Java NetBeans 8.1.

**NetBeans** est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License).

En tirant avantage de cette trousse à outils gratuite, basée sur des standards, les développeurs peuvent concevoir des applications complexes plus rapidement, avec une plus grande assurance de robustesse et de concevoir des applications qui résisteront à l'épreuve du temps.



**Figure 4.1** icone de NetBeans

#### 4.2.2.2 StarUML :

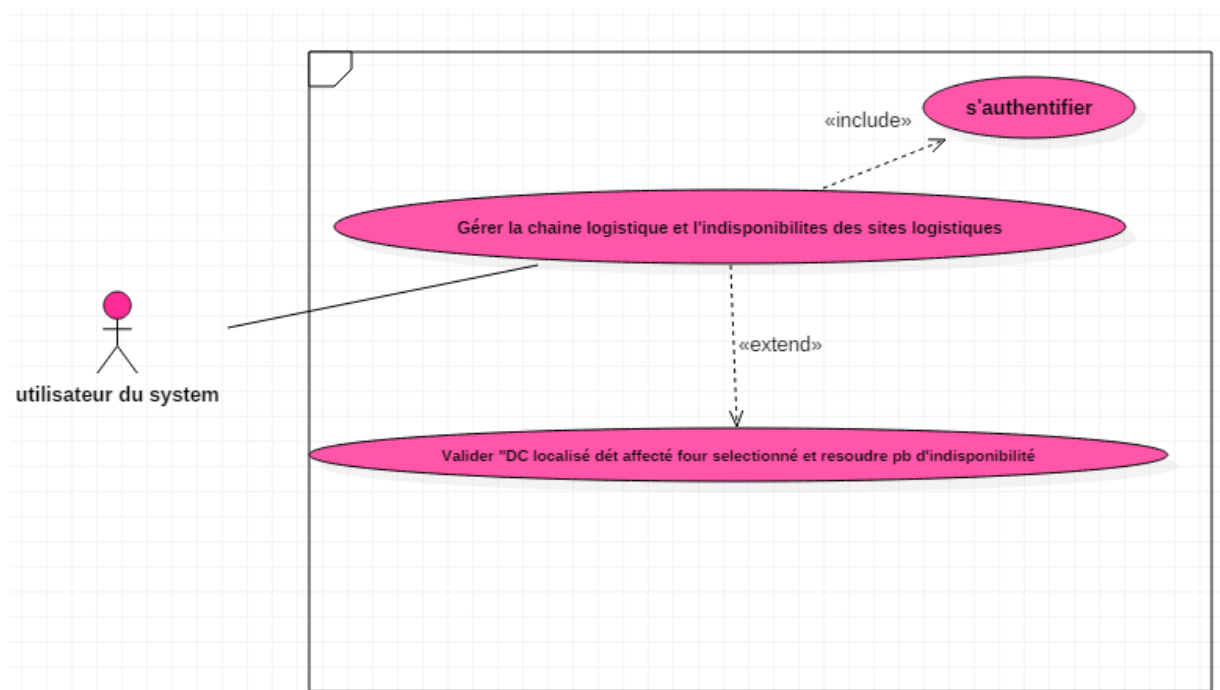


**Figure 4.2** icone de starUml

StarUML est un logiciel de modélisation UML, cédé comme open source par son éditeur, à la fin de son exploitation commerciale,

StarUML gère la plupart des diagrammes spécifiés dans la norme UML 2.0.

### 4.3. Le diagramme de cas d'utilisation



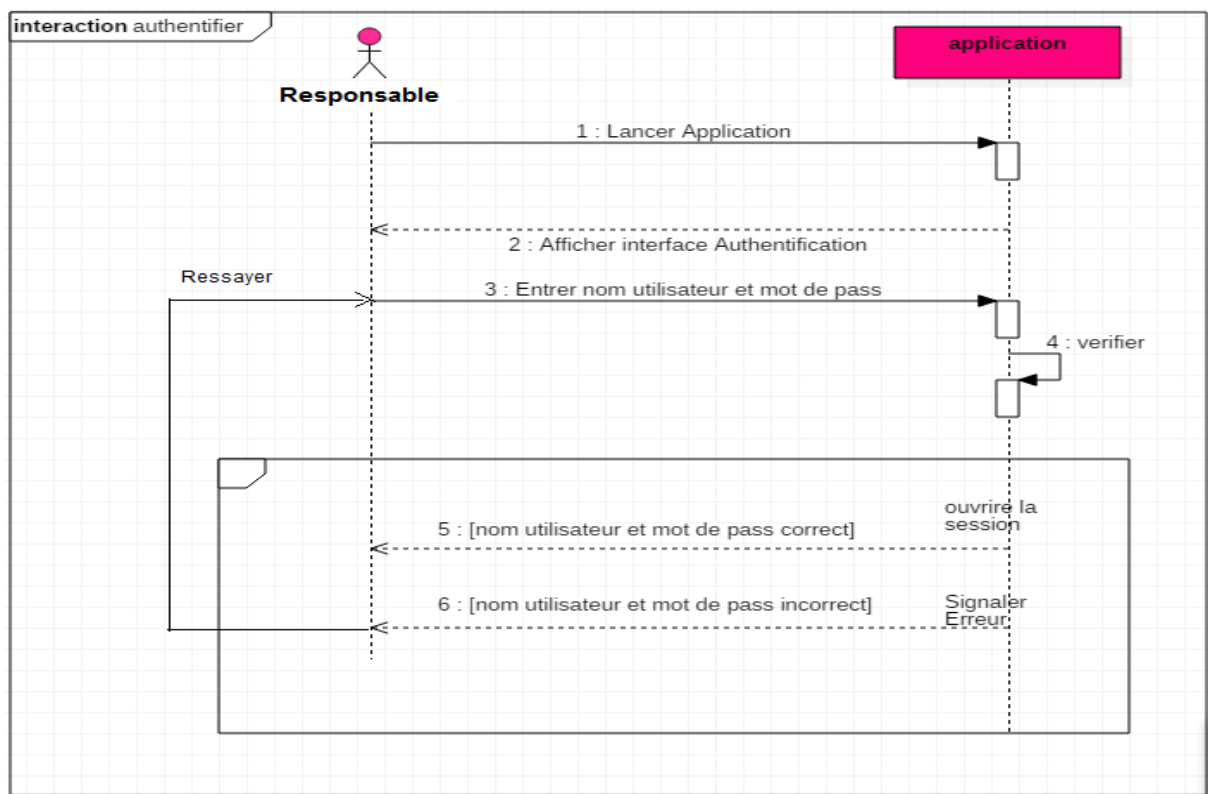
**Figure 4.3** diagramme de cas d'utilisation

#### 4.4. Les diagrammes de séquence de notre système

Le diagramme de séquence représente la succession chronologique des opérations réalisées par un acteur .Il indique les objets que l'acteur va manipuler et les opérations qui font passer d'un objet à l'autre.

- Cas d'utilisation « Authentifier »

Pour accéder au système, l'utilisateur doit s'authentifier en saisissant son login et mot de passe. Ensuite, le système passe à l'authentification des informations saisies si l'utilisateur est reconnu, alors le système charge le menu principal. S'il n'est pas reconnu, un message doit être renvoyé comme nous allons voir dans la figure suivante :

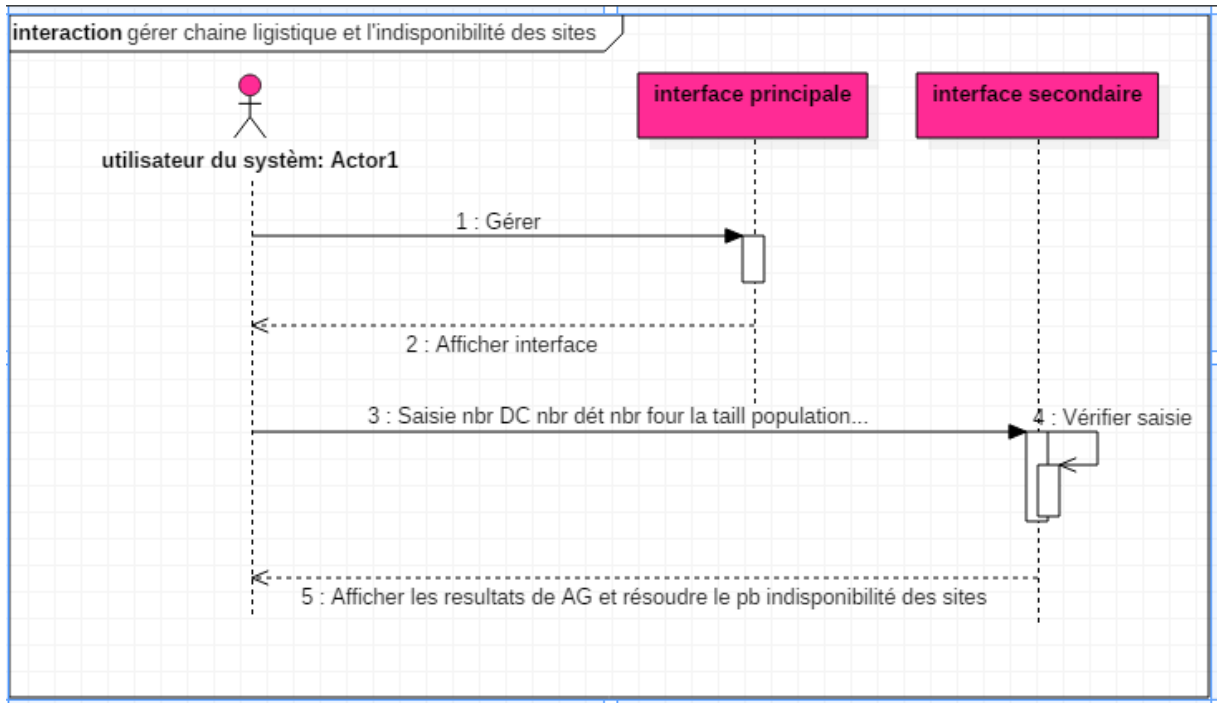


**Figure 4.4** Diagramme de séquence « authentification »

- Cas d'utilisation « Gérer la structure générale de la chaîne logistique et l'indisponibilité des sites logistiques »

Après s'authentifier, l'utilisateur fait le choix Gérer la structure générale de la chaîne logistique et l'indisponibilité des sites dans le menu principal, puis le système lance le

chargement d'interface secondaire concerné, l'utilisateur saisie le nombre des DCs et le nombre des fournisseurs et détaillants et la taille de population, nombre d'itération puis valider, Ensuite le système affiche les résultats obtenues par l'AG, comme nous allons voir dans la figure suivante :



**Figure 4.5** Diagramme de séquence « Gérer la structure générale de la chaîne logistique et l'indisponibilité des sites logistiques »

## 4.5. L'application des étapes d'algorithme génétique

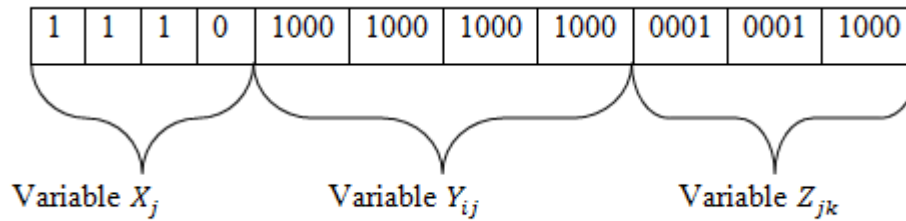
### 4.5.1. Localisation des DCs et l'affectation des détaillants et choix de fournisseurs (étape E1) :

Nous utilisons un algorithme génétique pour déterminer la structure optimale de la chaîne étudiée. Dans notre cas, une solution candidate est composée de valeurs binaires 0 ou 1, où chaque chromosome est constitué de trois parties. La première partie représente la localisation des DCs, la seconde partie l'allocation des détaillants aux DCs et la troisième partie représente l'affectation des DCs aux fournisseurs. Pour chaque solution candidate nous calculons sa fitness pour obtenir le coût total généré.

#### ➤ Représentation de solution :

Dans notre approche génétique, nous représentons la chaîne logistique sous forme d'un chromosome, ce dernier est composé de trois parties : la première  $X_j$  est pour représenter

les DC, la deuxième  $Y_{ij}$  pour les DET et la troisième  $Z_{jk}$  pour les fournisseurs par exemple un chromosome correspondant à un problème avec 4 détaillants (nous avons 4 DCs candidats situés dans les mêmes régions que les détaillants) et 3 fournisseurs potentiels est illustré à travers la figure 4.6



**Figure 4.6** Représentation du chromosome

Les différentes étapes de l'algorithme génétique utilisé se présentent comme suit :

- Etape 1. Générer la population initiale P de taille N aléatoirement, dans cette étape nous avons utilisé une fonction aléatoirement pour générer N chromosomes qui représentent la population initiale.
- Etape 2. Evaluer toutes les solutions dans P : c'est-à-dire calculer pour chaque chromosome sa fitness.
- Etape 3. Sélectionner les deux parents : Elle consiste à tirer au hasard deux solutions de la population P.
- Etape 4. Générer deux solutions enfants par le croisement des deux solutions parents avec une probabilité  $P_c$ .
- Etape 5. Exécuter l'opérateur de mutation avec une probabilité  $P_m$  pour chaque solution enfant.
- Etape 6. Exécuter une fonction de correction pour les solutions infaisables qui ne respectent pas les contraintes du modèle proposés c'est à dire si le chromosome n'accepte pas les contraintes donc appliquer la fonction de correction.
- Etape 7. Ajouter les deux solutions enfants dans la population suivante G.
- Etape 8. Répéter les étapes de 3 à 7 pour obtenir N solutions enfants dans G.
- Etape 9. Répéter les étapes de 2 à 8 jusqu'à la satisfaction des conditions d'arrêt.

➤ . **Fonction fitness:**

Le calcul de fitness pour minimiser le coût cette dernière donnée par la formule suivante :  $(Fon)J^* = Min_{X,Y,Z} J(X,Y,Z)$  (4.1)

Où  $J(X,Y,Z)$  est défini par :

$$J(X,Y,Z) = \sum_{j \in I} f_j X_j + \sum_{j \in I} \sum_{i \in I} \mu_i d_{ij} Y_{ij} + \sum_{j \in I} \sum_{i \in I} \sum_{k \in K} \mu_i a_{jk} Y_{ij} Z_{jk} + \sum_{j \in I} \sum_{k \in K} \sqrt{2h_j F_{jk} \sum_{i \in I} \mu_i Y_{ij} Z_{jk}} + \sum_{j \in I} \sum_{k \in K} Z_{jk} h_j \sqrt{L_{jk} \sum_{i \in I} \sigma_i^2 Y_{ij} + \lambda_{jk}^2 D_j^2 Z_{jk}} \quad (4.2)$$

Avec les contraintes suivantes:

$$\sum_{j \in I} Y_{ij} = 1 \quad \forall i \in I \quad (4.3)$$

$$\sum_{k \in K} Z_{jk} = X_j \quad \forall i \in I \quad (4.4)$$

$$Y_{ij} < X_j \quad \forall i, j \in I \quad (4.5)$$

$$X_j, Y_{ij}, Z_{jk} \in \{0,1\} \quad \forall i, j \in I \quad \forall k \in K \quad (4.6)$$

La fonction objectif (1) représente la somme des coûts de localisation, coûts de livraison, coûts d'approvisionnement, coûts de stockage et de commande ainsi que les coûts de maintien des stocks de sécurité. La contrainte (4.3) exige que chaque détaillant soit servi par un et un seul DC localisé. La contrainte (4.4) assure que l'approvisionnement de chaque DC ouvert se fait par un et un seul fournisseur. La contrainte (4.5) assure que les détaillants sont servis que par les DCs ouverts. La nature binaire des différentes variables de décision est exprimée par la contrainte (4.6).

• **Notations et variables utilisées**

Pour la formulation mathématique du problème étudié, nous utilisons les variables et les notations suivantes :

➤ **Notations**

$I$ : Ensemble des zone de demandes (détaillants) indexés par  $i$ ;

$K$ : Ensemble des fournisseurs indexés par  $k$ ;

$DC_j$ : Centre de distribution localisé dans la zone de demande  $j$ ;

$\mu_i$  : Demande globale générée par le détaillant  $i$ ;

$D_j$ : Demande moyenne journalière du  $DC_j$ ;

$\sigma_i^2$  : Variance de la demande journalière générée par le détaillant  $i$  ;

$f_j$  : Coût fixe de localisation du  $DC_j$  ;

$d_{ij}$  : Coût de livraison unitaire du  $DC_j$  vers le détaillant  $i$  ;

$h_j$  : Coût de stockage annuel (par unité de produit) dans le  $DC_j$  ;

$F_{jk}$  : Coût fixe de commande (inclus coût fixe de transport) placée par le  $DC_j$  auprès du fournisseur  $k$  ;

$\alpha_{jk}$  : Coût unitaire d'approvisionnement (prix d'achat et de transport) du  $DC_j$  auprès du fournisseur  $k$ ;

$L_{jk}$  : Délai moyen d'approvisionnement en jours du  $DC_j$  auprès du fournisseur  $k$ ;

$\lambda_{jk}^2$  : Variance du délai d'approvisionnement du  $DC_j$  auprès du fournisseur  $k$ ;

$\alpha$  : Niveau de service dans les centres de distribution;

$Z_\alpha$  : Coefficient de sécurité tel que  $P(Z < Z_\alpha)$ ;

➤ **Variables de décision**

$X_j = \{1 \text{ si } DC_j \text{ est localisé}, 0 \text{ sinon} \}$

$Y_{ij} = \{1 \text{ si le détaillant } i \text{ est servi par le } DC_j, 0 \text{ sinon} \}$

$Z_{jk} = \{1 \text{ si le fournisseur est sélectionné pour approvisionner le } DC_j, 0 \text{ sinon} \}$

#### 4.5.2. Gestion des indisponibilités des DCs (étape E2)

Pour la deuxième étape E2, un ou plusieurs DCs peuvent devenir indisponibles. Dans cette étape pour chaque indisponibilité d'un DC, une nouvelle allocation (ou réallocation) est effectuée pour les détaillants concernés par l'indisponibilité, ceci revient à exécuter le même AG de la première période mais en considérant que les DCs opérationnels et les détaillants affectés par l'indisponibilité.

- Le coût de gestion des indisponibilités de l'étape E2 est égal à :

$$CGI(E2) = C(E1) - C(E2) \quad (4.7)$$

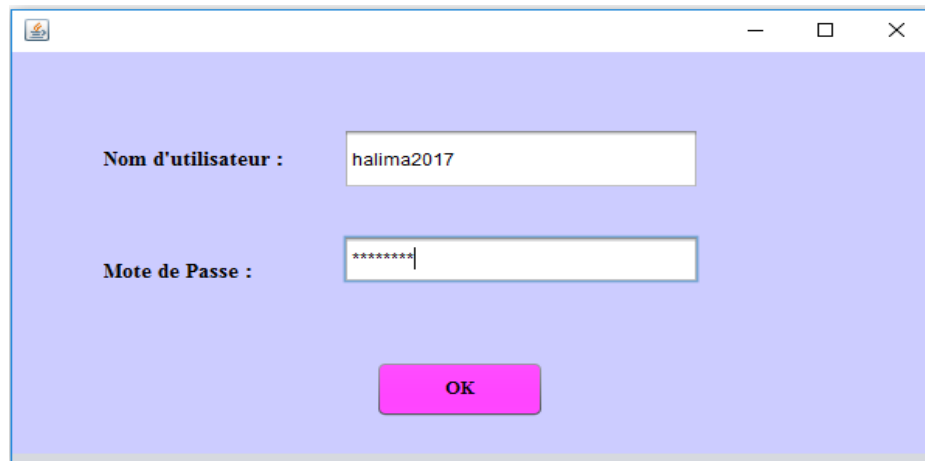
Où :  $C(E1)$  : Coût global généré durant l'étape E1 ;

$C(E2)$ : Coût global généré durant l'étape E2;

$CGI(E2)$ : Coût de gestion des indisponibilités durant l'étape E2 ;

#### 4.6. Les interfaces d'application :

##### 4.6.1. L'interface d'Authentification

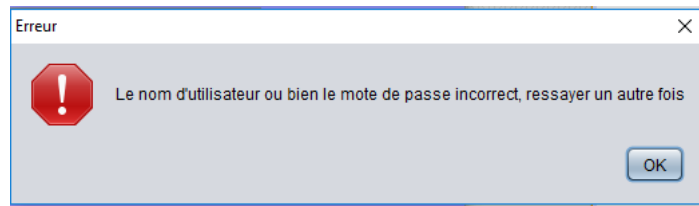


The image shows a software window titled 'Page d'Authentification'. It has a light blue background. On the left, there are two labels: 'Nom d'utilisateur :' and 'Mote de Passe :'. To the right of 'Nom d'utilisateur :' is a text input field containing the text 'halima2017'. To the right of 'Mote de Passe :' is a password input field showing seven asterisks '\*\*\*\*\*'. Below these fields is a pink rectangular button with the text 'OK' in black.

**Figure 4.7** Page d'Authentification

Saisie le nom d'utilisateur et le mot de passe puis valider Si les deux corrects le système lance le chargement de l'interface d'accueil, sinon affiche un message d'erreur Comme illustré dans la figure suivante :





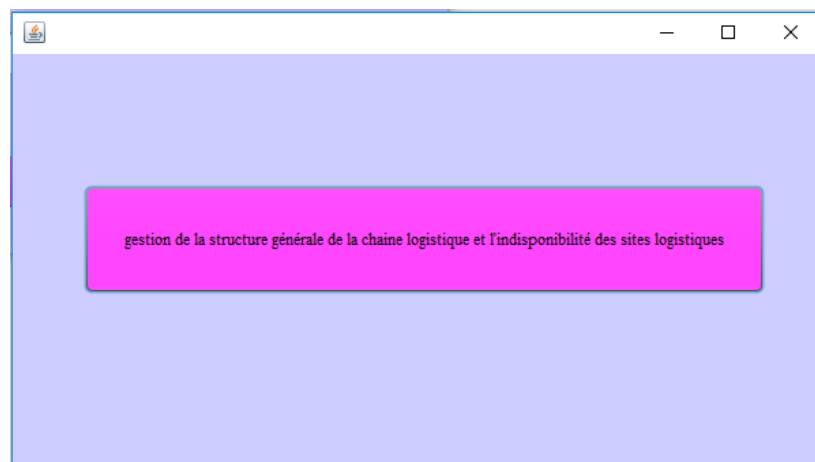
**Figure 4.8** Message d'erreur de l'Authentification.

#### **4.6.2. L'interface d'accueil**

La page accueil de notre application contient deux(02) boutons sont :

- gestion de la structure générale de la chaine logistique
- gestion de l'indisponibilité dans la chaine logistique

Comme illustré dans la figure suivante



**Figure 4.9** Page d'accueil

Si on click sur le bouton gestion de la structure générale de la chaine logistique et l'indisponibilité des sites logistiques sera paraitre une fenêtre secondaire.

### 4.6.3. L'interface secondaire



**Figure 4.10** interface secondaire

Ensuit valider donc le résultat affiché DCs localiser et Les fournisseurs sélectionnés et résoudre le problème de l'indisponibilité des sites logistiques, alors afficher aussi comme suite :

Premièrement afficher le benchmark :

```
***** Affichage de benchmark *****
vecteur f :
f[0]=9767
f[1]=9142
f[2]=6743
f[3]=6364
u :
u[0]=1470
u[1]=1556
u[2]=473
u[3]=379
d :
d[0][0]=1
d[0][1]=3
d[0][2]=2
d[0][3]=1
d[1][0]=2
d[1][1]=2
d[1][2]=1
d[1][3]=4
d[2][0]=2
d[2][1]=3
d[2][2]=2
d[2][3]=1
d[3][0]=1
d[3][1]=2
d[3][2]=4
d[3][3]=3
h :
h[0]=25
h[1]=25
h[2]=25
h[3]=25
F
F[0][0]=2
F[0][1]=15
```

**Figure 4.11** L'affichage de benchmark



--	--	--	--	--	--

Où :

- Nombre des détaillants (Dét) : Nous avons considéré des problèmes avec 10, 15 et 20 détaillants.
- Nombre des fournisseurs (Fou) : Nous avons considéré des problèmes constitués de 4, 7 et 10 fournisseurs.
- DCI = Nombre de DC localisés.                      - Fs = Nombre de fournisseur sélectionnés.

#### **4.8. Conclusion**

Dans ce chapitre, nous avons présenté et expliqué notre vision pour résoudre le problème de l'indisponibilité des sites dans la chaîne logistique par l'utilisation des algorithmes génétiques, notre objectif est de trouver premièrement la structure générale de la chaîne logistique par la localisation des DCs et l'affectation des détaillants aux DCs et la sélection des fournisseurs pour l'approvisionnement et deuxièmement la gestion de l'indisponibilité des sites dans la chaîne logistique.