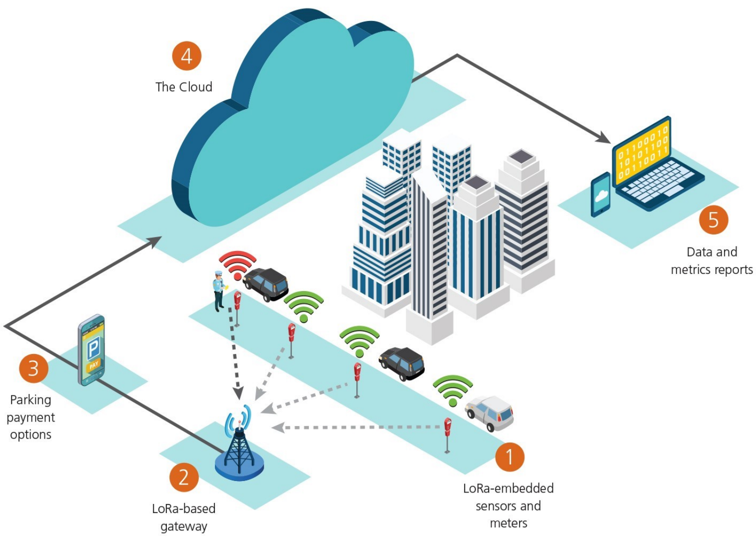
Projet personnel et professionnel (PPP)

Rapport de projet

Détermination d'une suite optimale d'investissements pour répondre à une demande d'abonnés à leur raccordement sur un central téléphonique.



Réalisé par :
TCHAH Oussama
LAHSINI Mohamed

Encadré par :
Mr. EN-NOUAARY Abdeslam

Filière SUD - Cloud & IOT

Table des matières

Première partie

Rapport

0.1 Introduction

Dans le cadre du projet personnel et professionnel (PPP) et dans le but de consolider nos connaissances en terme de programmation et conception et analyse des graphes, nous avons choisi de traiter un problème d'optimisation qui consiste à déterminer une suite optimale d'investissements pour répondre à une demande d'abonnés à leur raccordement sur un central téléphonique. Le but est donc de mettre en épreuve nos acquis en matière de théorie des graphes à savoir : **l'algorithme de Moore-Djikstra** ainsi que **les chemins optimaux**, afin de pourvoir lister les différentes stratégies optimales d'investissement pour chacun des environnements que nous allons proposer. En outre, ce projet sera également une occasion de fortifier notre esprit d'analyse et ce en faisant plusieurs simulations interprétations pour en tirer des conclusions pertinentes.

0.2 Objectif

L'objectif du projet est donc, d'une part, de modéliser le problème sous forme d'un graphe, d'appliquer l'algorithme de Moore-Djikstra pour déterminer le chemin optimal qui représente la stratégie optimale que nous cherchons, et d'autre part, de faire des simulations en agissant sur plusieurs paramètres du problème et de les interpréter.

0.3 Formulation et modélisation

0.3.1 Formulation du problème

On considère le pb d'un service public devant satisfaire une certaine demande de ses usagers (fonction croissante du tps). Soit $f(t)$ sur [0,T], le nombre d'abonnés demandant leur raccordement sur un central téléphonique à l'instant t. Il va falloir donc investir de nouveaux équipements. On suppose que l'on s'engage à chaque instant à ce que le central ait une capacité totale supérieure ou égale à la totalité de la demande, donc, dès que les matériels existants se saturent, il faut faire des extensions. Pour réaliser ces extensions, on dispose d'un matériel unique dont chaque module a une capacité C connue (il est toujours possible de faire 2, 3, $\cdots$, k extensions simultanées) et un coût γ qui est une **fonction croissante du temps**. (rapporté à l'année 0).

Chaque fois qu'une extension est décidée, le coût de mise en chantier est une quantité fixe δ (indépendante du nombre de modules effectivement installés). Si par exemple, on décide d'investir p modules à l'instant t, le coût total de l'opération (rapporté à la l'instant 0) est :

$$\frac{\delta + p\gamma}{(1+\tau)^t}$$

où τ est le taux d'actualisation. Soit n le nombre total de modules devant être installés à la fin de période [0,T], n vérifie :

$$nC >= f(T) > (n-1)C$$

.

0.3.2 Modélisation du problème

On construit le graphe G suivant :
On prendra comme ensemble des sommets x_0, x_1, $\cdots$, x_n où xi correspond à l'installation de i

modules. Pour j > i, il existera un arc (xi, xj) de coût $\frac{\delta+(j-i)\gamma}{(1+\tau)^{t_j}}$ où $t_j = f^{-1}(jC)$. Alors, la détermination d'une politique optimale d'investissements revient à déterminer un chemin de valeur minimale entre x_0 et x_n.

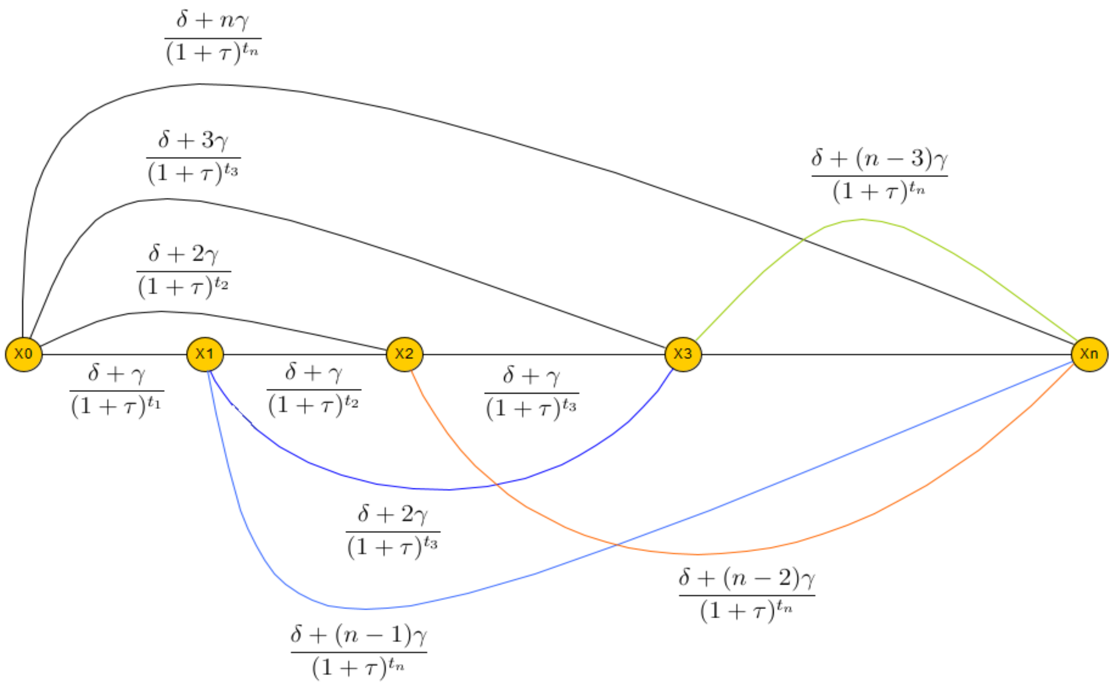


FIGURE 1 – **Graphe**

0.4 Implémentation

Afin de trouver le chemin optimal représentant la meilleur stratégie d'investissement à suivre, nous avons utilisé l'algorithme de Moore-Djikstra :
Etape 1 : (Initialisation) W = {1} , π(1) = 0, π(i) = c(1, i) si i $\in \Gamma_1$ et π(i) = 1 sinon, où Γ_j = {k $\in$ X/(j, k) $\in$ U}.
Etape 2 : Sélectionner j $\in \bar{W}$ tel que, π(j) = {minπ(i) : i $\in \bar{W}$} Faire : W $\leftarrow$ W $\cup$ j Si W = X, Stop ; sinon aller à l'étape 3.
Etape 3 : Faire pour tout i $\in \Gamma_j$ et i $\in \bar{W}$; $\pi(i) \leftarrow$ min{$\pi(i)$, π(j) + c(j, i)} et retourner à l'étape 2.

Les figures suivantes représentent l'implémentation de l'algorithme de Moore-Djikstra ainsi que de notre problème (définition du graphe et des variables du problème) en utilisant le langage de programmation **PYTHON**.(*)
(***Voir annexe, figure 2 et 3**)

0.5 Simulation

Dans cette partie, nous allons agir sur le paramètre γ qui représente le coût des modules à installer et qui est une fonction croissante du temps. Considérons la fonction suivante : $\gamma(i) = \alpha + ti$ avec i le pas, nous allons tracer la courbe de la fonction γ en fonction du pas i en choisissant, dans un premier lieu, la fonction f suivante : $f_1(x) = x^3 + 3x$ qui est une fonction polynômiale. Puis, pour les mêmes variations du pas et pour les mêmes valeurs de C et δ fixées dans la première simulation,

nous avons choisi une autre fonction f exponentielle $f_2(x) = e^x$.(*) Rappelons que la fonction f détermine le nombre d'abonnés demandant leur raccordement sur un central téléphonique lors d'une période et qui est aussi une fonction croissante du temps.
(***Voir annexe, figure 4 et 5**)
Nous remarquons que tout changement dans la valeur de gamma γ implique un changement au niveau du chemin optimal retourné par notre programme, ce qui signifie que le coût des modules est une donnée fondamentale de ce problème qui a un impact direct sur la stratégie d'investissement à suivre comme l'indiquent les figure dans l'annexe. En outre, nous remarquons aussi que le choix de la fonction f affecte aussi, pour une même valeur du pas i, le chemin optimal retourné par notre programme. Ainsi, la recherche d'une stratégie d'investissement repose essentiellement sur une étude de marché afin de connaître et d'anticiper l'évolution des prix des modules au cours des années à venir ainsi que le nombre d'abonnés demandant leur raccordement sur un central téléphonique.
N.B : Les fichiers générés au niveau de la partie simulation sont déposés dans le repository "PPP" sur **gitHub**.

0.6 Conclusion

Le problème de raccordement des abonnées à un central téléphonique est un problème qui ne cesse d'évoluer vu la grande évolution démographique mondiale, l'augmentation importante de la consommation des téléphones ainsi que l'évolution des outils et appareils de raccordement et des technologies de télécommunication. Le problème dépend aussi de la zone géographique à couvrir (quartier, ville, département, ...), du matériel utilisé (type des modules, câblage utilisé, ...) et donc fait partie des problèmes à plusieurs variantes qui nécessitent une large étude statistique du territoire, de la population et des technologies afin de pouvoir choisir la zone exacte à couvrir, l'équipementier adéquat à choisir ainsi que la stratégie optimale d'investissement à mener. Ce projet nous a été d'une grande importance, puisque c'était une opportunité de mettre en pratique les différentes notions étudiées notamment en théorie des graphes, en optimisation, en algorithmique et en programmation. En outre, ce projet nous a donné l'occasion de renforcer notre esprit d'analyse, de modélisation et de conception ; sans oublier notre capacité de gestion et de travail en groupe via le logiciel **gitHub** ainsi que d'autres logiciels de chat, surtout dans ces conditions par lesquelles nous passons dernièrement.

0.7 Remerciements

Avant de finir, nous tenons à remercier infiniment notre cher professeur et encadrant monsieur EN-NOUAARY Abdeslam pour tous les efforts fournis, son implication et tout le temps qu'il nous a accordé, non seulement durant la période du projet mais aussi tout au long de cette année universitaire. Nous remercions aussi monsieur RAHHALI Khalid de nous avoir généreusement aidé à réaliser ce travail.

Deuxième partie

Annexe

0.8 Annexe

Cette partie contient les différentes figures : graphes, implémentations, etc... :

Moore-djikstra.py > djikstra

```python
def djikstra(sommet, liaison, start):
    chemin = []
    # Etape 1:
    ######################################
    s = start
    resul = {}
    W = [s]
    pi = {s:0}
    fct = lambda c: pi[c]
    for i in sommet:
        if i!=s:
            if (s,i) in liaison:
                pi[i] = liaison[(s,i)]
                resul[i] = s
            else :
                pi[i] = 10000000000000000000
    # Main loop
    while len(W)!=len(sommet):
        # Etape 2 :
        #####################################
        T = sorted(pi, key = fct)
        nn = T[0]
        for i in range(len(T)):
            if T[i] not in W:
                if i<len(T)-1:
                    nn = T[i+1]
                    if nn != T[i]:
                        W.append(T[i])
                        s = T[i]
                        break
                if i==len(T)-1:
                    W.append(T[i])
                    s = T[i]
                    break
        # Etape 3 :
        #####################################
        for i in sommet:
            if i not in W:
                if (s,i) in liaison:
                    k = min(pi[i], pi[s]+liaison[(s,i)])
                    if k!=pi[i]:
                        resul[i] = s
                    pi[i] = k
    a = len(sommet_simulation)-1
    chemin.append(a)
    while a!= start:
        prefix = resul[a]
        a = prefix
        chemin.append(a)
```

Figure 2 – **Implémentation de l'algorithme de Moore-Djikstra en langage PYTHON (suite)**

```python
    chemin.reverse()
    print(chemin)
    return W
if __name__ == "__main__":
    #simulation:
    ba = 10
    T = 3
    C = 1
    gamma = []
    delta = 10
    taux = 0.05
    n =1
    ############## calcule de C:
    maxf = f(T)
    s_simulation = 0
    while n*C<maxf:

        n = n+1
    ###########################

    ############# fonction gamma
    for b in range(n+1):
        gamma.append(ba)
        ba = ba+1
    #############################
    sommet_simulation = [i for i in range (0,n+1)]
    liaison_simulation  = {}
    t = [0]
    for j in range(1,len(sommet_simulation)):
        resulttt = fp(C*j,0.1)
        t.append(resulttt)
    for i in range (len(sommet_simulation)):
        for j in range (i+1, len(sommet_simulation)):
            b = (1+taux)**t[j]
            a = (delta + (j-i)*gamma[j])
            liaison_simulation[(i,j)] = a/b
    djikstra(sommet_simulation,liaison_simulation,s_simulation)
```

FIGURE 3 – **Définition du graphe et des différents variables du problème en langage PYTHON (suite)**

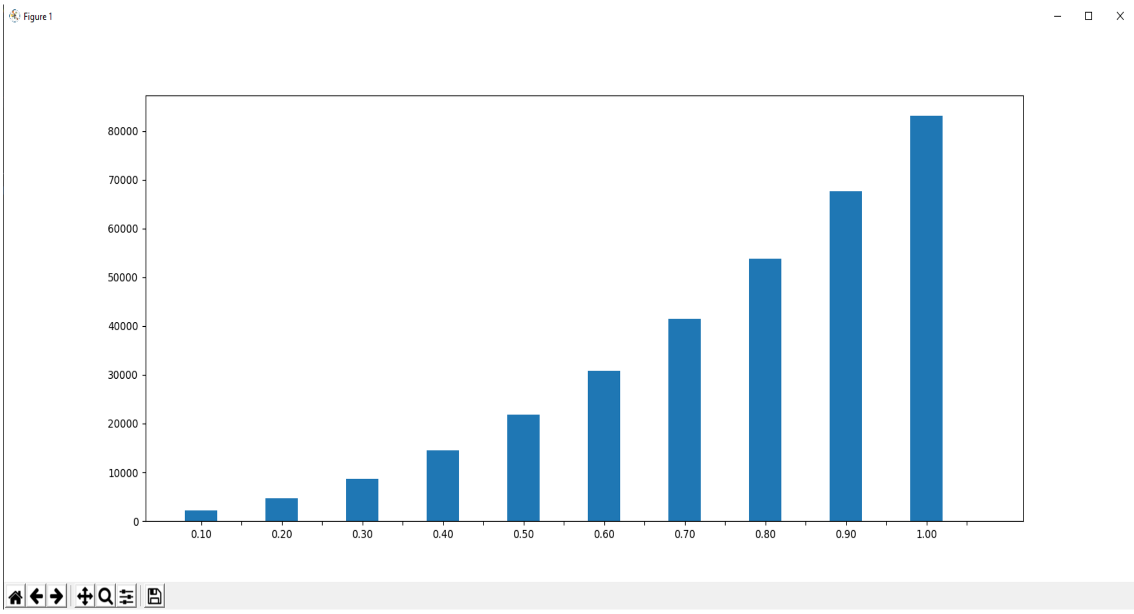


FIGURE 4 – **Tracé de la courbe de la fonction γ en fonction du pas i, avec f_1 une fonction polynômiale, en utilisant matplotlib**

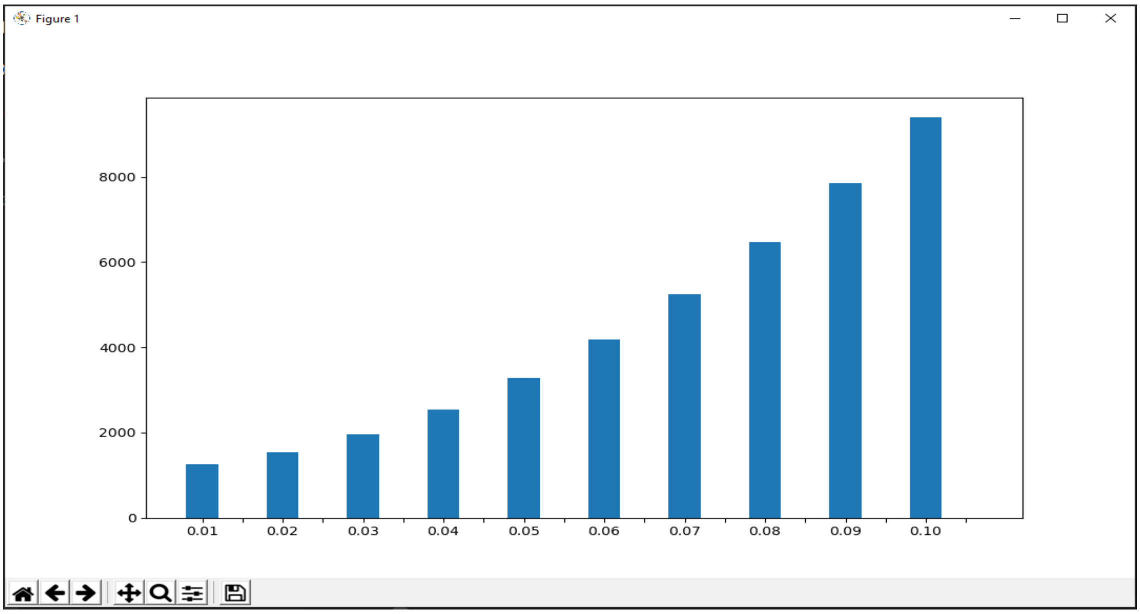


FIGURE 5 – **Tracé de la courbe de la fonction γ en fonction du pas i, avec f_2 une fonction exponentielle, en utilisant matplotlib**