



# Proposition d'une nouvelle stratégie de l'incitation à la mise en cache dans les réseaux d'internet de véhicules

## 4.1 Introduction

Avec l'explosion du volume de données mobiles, il devient difficile de fournir un service de haute qualité aux utilisateurs mobiles. Par conséquent, la mise en cache en périphérie de réseau des contenus populaires a reçu une attention considérable car elle peut rapprocher le contenu des utilisateurs mobiles, pour diminuer les délais d'accès/téléchargement et réduire la charge sur les liaisons de réseau cœur [6]. De plus, vu l'hétérogénéité des équipements mobiles, la diversité des demandes des utilisateurs en termes de qualité de service, et les ressources de stockage limitées des terminaux mobiles, il est important d'améliorer l'efficacité de la gestion de contenu dans la mise en cache en périphérie de réseau. Pour cela, plusieurs chercheurs [2], [33], [43], [35] ont déjà étudié le problème de mise en cache d'une perspective commerciale en se concentrant sur la façon comment inciter et encourager les acteurs à participer dans l'amélioration du processus de la mise en cache.

Dans le même contexte, nous proposons dans ce chapitre une stratégie d'incitation à la mise en cache pour un réseau d'internet de véhicules, dans lequel plusieurs sociétés de transport de voyageurs en commun (Public Transport Company, PTCs) qui offrent un service de stockage payant (des caches embarqués sur des véhicules) aux fournisseurs de contenu (Content Provider, CP) à l'instar de Youtube et Netflix qui possèdent des contenus généralement de la vidéo et cherchent à les rapprocher de leurs utilisateurs finaux pour améliorer la qualité de service (Quality of Experience, QoE).

Le reste de ce chapitre est organisé comme suit. Dans les sections 4.2, nous décrivons le

modèle du système et nous modélisons la distribution de la popularité du contenu, ainsi que la popularité globale des CPs. Puis, nous discutons la motivation de la politique d'incitation à la mise en cache pour chacun des deux approches proposées : premièrement, nous formalisons la théorie des contrats pour le marché monopolistique puis le jeu de type Stackelberg pour le marché concurrentiel et nous cherchons dans chaque approche les paramètres optimaux de notre système. La section 4.3 conclut le chapitre.

## 4.2 Incitation à la mise en cache dans les réseaux de véhicules

Nous présentons dans cette section notre politique d'incitation à la mise en cache pour les réseaux d'Internet de véhicules. Nous commençons par présenter le modèle du système. Puis, nous modélisons la distribution de la popularité du contenu et la popularité globale des CPs. Enfin, nous détaillons notre politique d'incitation et la solution proposée.

### 4.2.1 Modèle du système

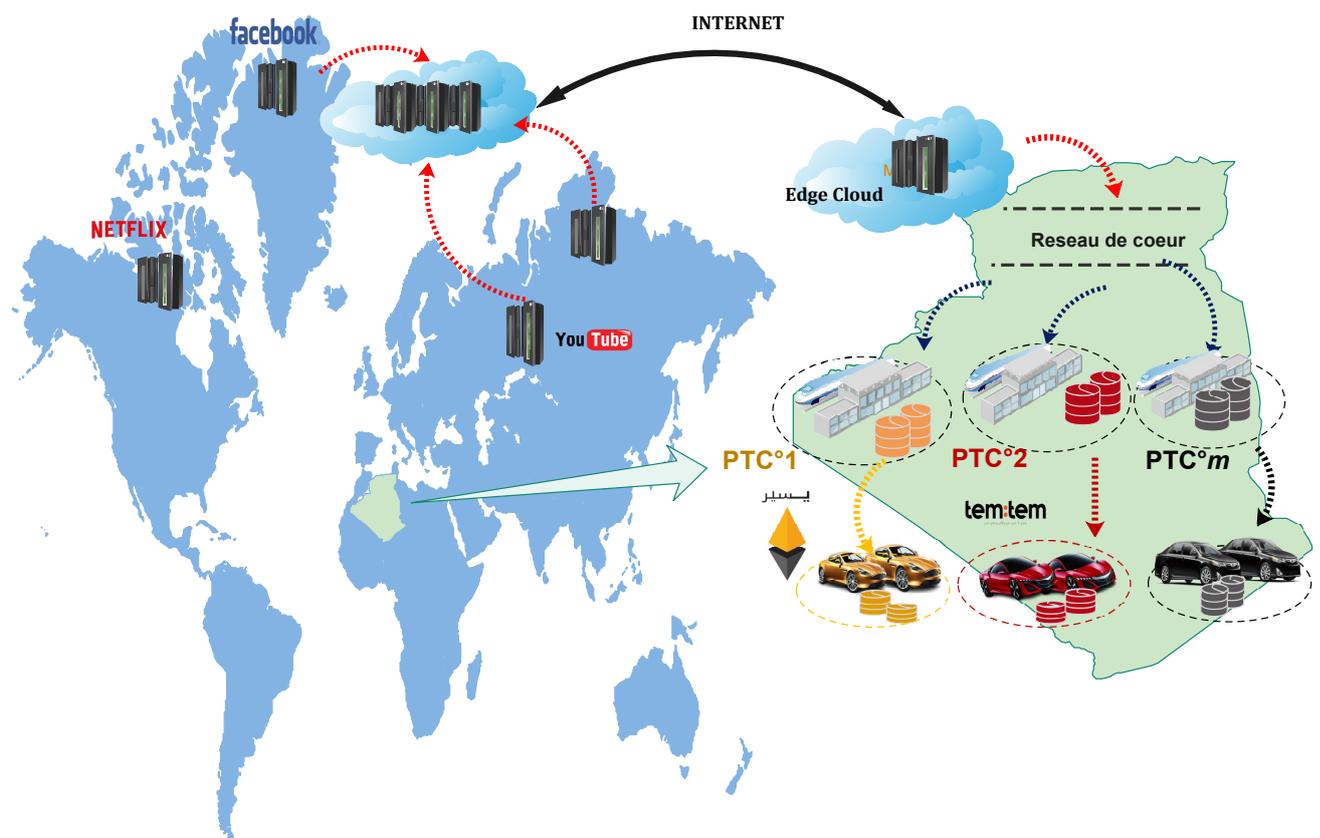


Figure 4.1: Modèle du système de la politique de l'incitation à la mise en cache dans les réseaux d'IoVs.

Comme il est illustré à la figure 4.1, nous considérons un système d'incitation à la mise en

cache dans les réseaux d'IoV. Notre scénario est composé de  $m$  sociétés de transports public des voyageurs (Public Transport Company, PTCs) et un ensemble de  $\mathcal{C}$  fournisseurs de contenu (Content Provider, CPs).  $G = \{PTC_1, PTC_2, \dots, PTC_m\}$  est l'ensemble des PTCs. Comme exemple des PTCs nous citons : les sociétés des bus comme **FlixBus**, les sociétés de Taxi ou de co-voiturage (**BlaBlaCar**) ou même les CTCs (Compagnie de Transport avec Chauffeur) comme **Yassir**, **Uber**, **Lyft**, **temtem**, etc. Chaque PTC  $i$  possède un ensemble de  $v_i$  véhicules géographiquement répartis sur l'ensemble de réseau autoroutier, tel que  $\mathcal{V}_i = \{v_{1,i}, v_{2,i}, \dots, v_{v_i,i}\}$ . Chaque véhicule  $v_{v_i,i}$  de PTC  $i$  est équipé par un cache embarqué (un espace de stockage) et peut servir les autres véhicules dans son entourage en utilisant des liaisons véhicule à véhicule (V2V). Sans perte de généralité, nous supposons que tous les caches sur les véhicules ont la même taille de stockage  $T_c$ . D'autre part, les véhicules ont la même puissance de transmission  $P^t$ . Les véhicules communiquent les uns avec les autres en utilisant IEEE 802.11p et avec l'infrastructure fixe en utilisant 4G/5G. Les PTCs proposent le service de mise en cache aux  $\mathcal{C}$  fournisseurs de contenu (CPs). Notons par  $\mathcal{C} = \{CP_1, CP_2, \dots, CP_{|\mathcal{C}|}\}$  l'ensemble des CPs, qui sont prêts à mettre leurs fichiers en cache des véhicules afin d'améliorer la QoE de leurs utilisateurs. Ces derniers sont classés en différents types en fonction de leurs popularités, ces types sont inconnus au PTC. Comme exemple de CPs, nous pouvons cité : YouTube, Netflix, Spotify, Facebook, etc.

Chaque  $CP_c (0 < c \leq \mathcal{C})$  gère une bibliothèque de contenu local, notée  $L_c$ , composée d'un ensemble fini  $f < \infty$  de ses fichiers populaires  $\mathcal{F}_c$  tel que  $\mathcal{F}_c = \{\mathcal{F}_{1,c}, \mathcal{F}_{2,c}, \dots, \mathcal{F}_{f,c}\}$  [33], et tous les fichiers  $\mathcal{F}_c \in L_c$  sont supposés avoir la même unité de taille [2]. Cependant, la capacité de stockage des véhicules est limitée et pas tous les fichiers des CPs peuvent être stockés en cache avec suffisamment de copies. Les principales notations utilisées dans ce chapitre sont résumées dans le tableau 4.1.

Notation	Description
$c$	Le nombre de CPs
$m$	Le nombre de PTCs
$v$	Le nombre de véhicules
$f$	Le nombre de fichiers d'un CP
$\mathcal{C} = \{CP_1, CP_2, \dots, CP_c\}$	L'ensemble de CPs
$G = \{PTC_1, PTC_2, \dots, PTC_m\}$	L'ensemble des PTCs
$\mathcal{V}_i = \{v_{1,i}, v_{2,i}, \dots, v_{v,i}\}$	L'ensemble de véhicule de $PTC_i$
$\mathcal{F}_c = \{\mathcal{F}_{1,c}, \mathcal{F}_{2,c}, \dots, \mathcal{F}_{f,c}\}$	L'ensemble de fichiers de CPs
$p_{i,c}$	La popularité du $i$ - ème fichier
$Q$	La quantité des fichiers populaires qu'un CP veut mettre en cache.
$\theta_k$	La popularité du $k^{ieme}$ CP
$\pi_i$	Le prix initial annoncé par le PTC
$Q^t$	La quantité totale des fichiers de $CP_j$ caché chez le $PTC_i$
$Q_{ij}$	La quantité de contenus de $CP_j$ acceptés dans le cache de $PTC_i$
$P^t$	La puissance de transmission des fichiers d'un CP
$\mu$	L'unité de coût de transmission et la maintenance
$\eta_i$	Le coût d'énergie qu'il consomme pour gérer les accès aux fichiers stockés dans les caches de $PTC_i$
$T_c$	La taille de stockage des caches de véhicules
$S_i$	La taille globale de stockage d'un $PTC_i$

Table 4.1: Résumé des principales notations

#### 4.2.2 Modélisation de la popularité du contenu des CPs

La popularité du contenu est définie comme la distribution de probabilité des demandes des fichiers populaires de la bibliothèque  $\mathcal{F}_c$  par tous les abonnés du réseau (par exemple, les vidéos qui sont fréquemment demandés par les utilisateurs mobiles).

L'ensemble des fichiers est désigné par  $\mathcal{F}_c = \{\mathcal{F}_{1,c}, \mathcal{F}_{2,c}, \dots, \mathcal{F}_{f,c}\}$  et il est composé des fichiers  $\mathcal{F}$ . La popularité de ces fichiers est indiquée par un vecteur  $p_c = \{p_{1,c}, p_{2,c}, \dots, p_{f,c}\}$  qui peut être modélisée par la distribution Zipf comme dans les études précédentes [2] [43] [35]. La popularité du  $i$  - ème fichiers est donné par :

$$p_{i,c} = \frac{1/i^\gamma}{\sum_{j=1}^{\mathcal{F}} 1/j^\gamma} ; \quad i = 1 \dots \mathcal{F} \quad (6)$$

Où  $i < \mathcal{F}$  c'est le nombre d'éléments des fichiers dans la bibliothèque de  $CP_c$ , et l'exposant  $\gamma$  est une valeur positive caractérisant la popularité du contenu et il peut varier d'un CP à l'autre. C'est-à-dire un  $\gamma$  plus grand correspond à un contenu plus utilisé, donc les fichiers les plus populaires représentent la majorité des demandes. La distribution de Zipf trie le contenu par ordre décroissant de popularité (le contenu avec un plus petit  $i$  correspond à une plus grande popularité) [2].

Dans le même temps, les MUs ont des affiliations déséquilibrées en ce qui concerne les  $CP$ , *i.e.*, certains CPs ont plus d'utilisateurs mobiles que d'autres. Par exemple, la majorité des MUs peut avoir tendance à accéder à YouTube pour le streaming vidéo (Certains CP peuvent avoir plus d'utilisateurs mobiles que les autres) [48]. Les préférences des MUs envers les CPs est indiquée par  $\theta_k = \{\theta_1, \theta_2, \dots, \theta_C\}$  où  $\theta_k, k = 1, \dots, C$  représente la popularité total de  $CP_k$  (Type de CP).

La distribution des préférences  $\theta$  peut également être modélisée en calculant l'écart type de popularité des fichiers de CP qui peut être définie comme suite :

$$\theta_k = \sqrt{\frac{\sum_{f=1}^F \left( p_f - \frac{\sum_{f=1}^F p_f}{F} \right)^2}{F}} ; \quad k = 1 \dots C \quad (7)$$

Dans le reste de ce chapitre, nous décrirons nos deux politiques d'incitation à la mise en cache. Une première politique pour l'incitation dans un marché de mise en cache monopoliste et une deuxième politique dans un marché de l'incitation à la mise en cache compétitif.

### 4.2.3 Politique d'incitation dans un marché de mise en cache monopoliste

Dans cette politique, nous considérons un marché de mise en cache monopoliste composé d'un seul PTC et plusieurs CPs. Notre idée de base consiste à placer les contenus populaires des CPs dans les caches des véhicules qui sont distribués dans les endroits les plus proches possible des utilisateurs mobiles. Donc, les CPs ont besoin de cache des véhicules pour stocker leurs contenus, et le PTC aussi a besoin de remplir les caches de ses véhicules. Par conséquent, ces deux acteurs ont intérêt à collaborer entre eux pour un déploiement réussi de la mise en cache.

Nous proposons dans cette section une politique d'incitation à la mise en cache mobile dans les réseaux d'IoV avec des informations asymétriques. Dans laquelle les CPs rivalisent pour augmenter la QoE de leurs utilisateurs par rapport aux contenus stockés dans les espaces limités de stockage des véhicules. Cela, en plaçant les contenus populaires dans les endroits

plus proches possibles des utilisateurs mobiles afin de bénéficier d'un taux de transmission plus élevé et d'un délai d'accès plus rapide.

Par conséquent, Le PTC a pour objectif de déterminer une politique de prix qui motive les CPs à participer au système de mise en cache en offrant des prix de stockage plus attractifs. En effet, le CP récompense le PTC en fonction du nombre de contenus populaires qu'il accepte de mettre en cache en payant un prix monétaire pour chaque contenu mis en cache.

#### 4.2.3.1 Approche basée sur la théorie des contrats pour l'incitation à la mise en cache monopoliste dans les réseaux d'IoV

Sur la base du modèle d'incitation à la mise en cache mobile dans les réseaux IoVs présenté ci-dessus, nous considérons un scénario pratique avec des informations asymétriques au sein duquel les CPs peuvent être de différents types privés et ils sont inconnus du PTC.

Dans notre travail, nous ne pouvons pas utiliser le jeu de Stackelberg comme dans les travaux [2], [48], [35] et [85] par ce que ces études supposent généralement que les joueurs sont honnêtes et révèlent toutes leurs informations privées, ce qui n'est pas toujours une hypothèse valable en pratique. À cette fin, pour gérer le problème d'asymétrie de l'information, nous nous tournons vers la théorie des contrats dans notre système incitatif. Ces informations privées concernent la popularité et la volonté de mise en cache de chaque CP, nous formulons l'interaction entre le PTC et les CPs sur la base de la théorie des contrats qui permet au PTC de définir un contrat  $(Q_k, \pi_k)$  pour chaque CP de type  $\theta_k$  en présence d'informations asymétriques. Tel que  $\pi_k$  dénote la récompense monétaire qui est versée par le  $CP_k$ , et  $Q_k$  dénote l'espace de stockage alloué au  $CP_k$ .

##### 1) Type de fournisseur de contenu (CP)

Nous définissons le type du  $CP_i$  comme une représentation de sa popularité du contenu (la popularité globale des fichiers) et sa volonté de mise en cache. Nous supposons qu'il existe au total  $\mathcal{C}$  différents types et nous désignons l'ensemble des types par  $\theta = \{\theta_1, \theta_2, \dots, \theta_e\}$ .

Nous utilisons la popularité définie dans l'équation (7) pour représenter les types des CPs [42], [44] qui sont triés par ordre croissant comme suivant  $\theta_1 \leq \dots \leq \theta_k \leq \dots \leq \theta_e$ . Un type supérieur implique une plus grande popularité par rapport au CP d'un type inférieur. Veuillez noter que l'environnement est à information asymétrique où les valeurs exactes des types de CP sont des informations privées. En effet, les informations véridiques de popularité sont nécessaires au PTC pour définir les contrats qui optimisent le bénéfice des CPs et couvrir les coûts de la mise en cache. Ce coût comprend les frais de déploiement de mémoire de stockage et aussi pour rafraîchir les unités de stockage. Ici, notre objectif est de concevoir des contrats qui incitent les CPs à révéler les vraies valeurs de leurs types  $\theta$  au PTC. De ce fait, au lieu de fournir un contrat uniforme à tous les CPs, le PTC propose un ensemble de contrats selon le

type  $\theta$  de CP. Il est laissé à libre choix des CPs d'accepter ou de refuser tout type de contrats. Si le CP refuse de signer un contrat, nous supposons que le CP reçoit un contrat  $(Q_1, \pi_1)$  tel que le prix est nul  $\pi_1 = 0$ .

## 2) Modèle des utilités

Dans cette sous-section, nous présentons la fonction d'utilité des CPs et de PTC basée sur le contrat signé.

Le PTC qui possède le monopole de marché de mise en cache établit les contrats avec les entrées  $\{Q, \Pi\}$  qui sont les combinaisons de la quantité et de prix de ses ressources, à savoir les caches embarqués sur les véhicules. L'ensemble des prix est noté par  $\Pi = \{\pi_1, \pi_2, \dots, \pi_c\}$  et il est associé à l'ensemble  $Q = \{Q_1, Q_2, \dots, Q_c\}$  des espaces alloués à chaque type  $\theta_k$ . Les CPs sont libres de décider quelle quantité de stockage ils vont acheter.

### A) Utilité du PTC

La fonction d'utilité de PTC peut être définie comme l'ensemble des gains monétaires  $\pi_k$  qui sont facturées aux CPs en contrepartie de stockage de leurs fichiers moins les coûts  $c(Q_k)$  de la gestion des ressources allouées aux CPs, y compris le coût de stockage et de la transmission [42].

Le PTC peut connaître la probabilité qu'un CP appartienne au type-  $k$  malgré l'asymétrie de l'information, qui peut être indiquée par  $\varphi_k$  et elle doit satisfaire la condition  $\sum_{k=1}^c \varphi_k = 1$ . Dans ce travail, nous supposons que tous les types des CPs ont une probabilité égale à  $\varphi_k = 1/c$ . [86]

La fonction d'utilité globale de PTC peut être formulée comme suit :

$$u_{PTC} = \sum_{k=1}^c \varphi_k (\pi_k - c(Q_k)) \quad (8)$$

Le gain de PTC dans le problème de mise en cache pour chaque type de CP est représenté par le profit monétaire total  $\pi_k$  que le CP $_k$  doit payer en contrepartie de stockage de la quantité  $Q_k$  de ses fichiers populaires.

Le coût (les pertes) de PTC concernant l'opération de la mise en cache est représenté par le coût de stockage en cache. Il est défini, par le coût énergétique que le PTC consomme pour gérer les différents accès aux fichiers stockés dans les caches plus le coût de la transmission et de partage des fichiers de CP (pareil comme il a été proposé dans [44]). Par conséquent, le coût de la mise en cache de tous les fichiers de CP dans l'espace  $Q_k$  de PTC peut être défini par l'équation suivante :

$$c(Q_k) = (\mu * p^t * Q_k + \eta_k * \sigma_k) \quad (9)$$

Tel que :

$S$  : La capacité maximale en cache de tous les véhicules de PTC.

$Q_k$ : La quantité des fichiers de  $CP_k$ .

$P^t$ : La puissance de transmission des fichiers d'un CP.

$\mu$  : L'unité de coût de transmission et de la maintenance.

$\eta_k$  : Le coût en énergie que le PTC consomme pour gérer les accès de  $CP_k$  aux fichiers stockés dans les caches.

$\sigma_k$  : dénote l'écart type de la popularité des fichiers  $Q_k$  de  $CP_k$ , il est calculé comme suit :

$$\sigma_k = \sqrt{\frac{\sum_{q=1}^Q \left( p_q - \frac{\sum_{q=1}^Q p_q}{Q} \right)^2}{Q}} \quad (10)$$

à partir de (9), l'utilité de PTC peut être donnée par :

$$u_{PTC} = \sum_{k=1}^c \varphi_k \left( \pi_k - (\mu * P^t * Q_k + \eta_k * \sigma_k) \right) \quad q \neq k \quad (11)$$

## B) Utilité du $CP_k$

La fonction d'utilité des  $CP_k$  de type  $\theta_k$  qui décide de mettre en cache un ensemble de fichiers  $\mathcal{F}_k$  dans les caches embarqués sur les véhicules de PTC sur la base de contrat  $(Q_k, \pi_k)$  est représentée comme dans [86] par les récompenses reçues  $r_k(\theta_k, Q_k)$  moins le coût de stockage  $\pi_k$ .

$$u_{CP}(k) = r(\theta_k, Q_k) - \pi_k \quad (12)$$

Le gain de  $CP_k$ , noté par  $r_k(\theta_k, Q_k)$ , représente la satisfaction de ses utilisateurs qui est modélisée par la popularité qui peut être atteinte pour tous les fichiers de CP (la popularité totale de  $CP_k$ ). Ce gain est modélisé via une fonction croissante par rapport à la quantité des fichiers stockés de  $CP_k$  et leur popularité. Il est calculé par la fonction suivante :

$$r(\theta_k, Q_k) = \theta_k \log(1 + Q_k) \quad (13)$$

À partir de (13) et comme dans [44], [87] et [86], nous pouvons remarquer que  $r(\theta_k, Q_k)$  est une fonction concave strictement croissante puisque  $r(0) = 0$ ,  $r(\theta_k, Q_k)' > 0$  et  $r(\theta_k, Q_k)'' < 0$ .

$r(\theta_k, Q_k)$  augmente rapidement jusqu'à un certain seuil lorsque la quantité et la popularité des fichiers mis en cache augmentent, puis augmentent plus lentement lorsque les fichiers stockés devient redondants.

Le coût de CP est défini par le prix facturé  $\pi_k$  par le PTC en contrepartie de stockage de  $Q_k$ . Donc la fonction d'utilité de  $CP_k$  peut être désormais donnée par :

$$u_{CP}(k) = \theta_k \log(1 + Q_k) - \pi_k \quad (14)$$

Étant donné la fonction d'utilité dans (14), le CP choisira l'entrée qui maximisera son propre gain.

### 3) Solution de contrat

Dans cette sous-section, nous cherchons à formuler le contrat qui aide à réaliser un système de mise en cache efficace. Premièrement, nous décrivons les contraintes nécessaires qui peuvent garantir la faisabilité du contrat. Ensuite, nous formulons le problème de maximisation d'utilité de système basé sur les contraintes des faisabilités. Enfin nous cherchons à simplifier et calculer les valeurs optimales  $(Q_k^*, \pi_k^*)$ .

#### A) Contraintes de faisabilité d'un contrat

Le PTC doit s'assurer que le coût des frais de stockages est au moins couvert par le prix facturé aux CPs [33].

Pour inciter les CPs à collaborer avec le PTC dans le système de mise en cache, le contrat qu'un CP a sélectionné doit respecter les contraintes de faisabilité qui sont la rationalité individuelle et compatibilité de l'incitation pour tous les types de CPs [88].

**Définition 4.1. Rationalité individuelle (Individual Rationality, IR) :** IR est définie dans la théorie des contrats comme une propriété selon laquelle chaque CP est considéré comme rationnel. Autrement dit, le contrat qu'un CP choisit devrait garantir que son utilité soit toujours positive par rapport à son type [33], [44]. IR peut-être exprimé comme suit :

$$u_{CP}(k) = r(\theta_k, Q_k) - \pi_k \geq 0 \quad , \quad \forall k \quad (15)$$

Les contrats proposés par le PTC doivent apporter des utilités non négatives aux CPs, ce qui motive le CP à participer activement au processus de mise en cache.

**Définition 4.2. Compatibilité incitative (Incentive Compatibility, IC) :** le IC signifie qu'un CP ne peut pas gagner une utilité en acceptant une entrée de contrat qui n'est pas conçu

pour son type [44]. Plus précisément, un contrat satisfait la contrainte de compatibilité incitative si chaque CP de type  $\theta_k$  préfère révéler son type réel  $\theta_k$  plutôt que d'un autre type  $\theta_{k'}$  [48]. IC peut être écrit comme suit :

$$r(\theta_k, Q_k) - \pi_k \geq r(\theta_k, Q_{k'}) - \pi_{k'}, \quad \forall k \neq k' \quad (16)$$

## B) Problème d'optimisation des contrats

Au-delà des contraintes IR et IC et comme dans les travaux [42], [46] et [87] le PTC conçoit plusieurs contrats et chaque CP choisit la bonne entrée à acheter (le bon contrat qui optimise son utilité). L'objectif de PTC est de maximiser son utilité  $u_{PTC}(\theta_k)$  en offrant les entrées de contrat optimale  $(Q_k^*, \pi_k^*)$ ,  $\forall k = 1 \dots C$ . Donc la conception du contrat optimal peut être formulée comme un problème de maximisation de l'utilité du PTC, comme suit :

$$\begin{aligned} (Q_k^*, \pi_k^*) &= \arg \max_{Q_k, \pi_k} \sum_{k=1}^c \varphi_k \left( \pi_k - \left( (\mu * p^t * Q_k + \eta_k * \sigma_k) \right) \right) \\ \text{S.C. IR(15):} & \theta_k \log(1 + Q_k) - \pi_k \geq 0, \forall k \\ \text{IC(16):} & \theta_k \log(1 + Q_k) - \pi_k \geq \theta_k \log(1 + Q_{k'}) - \pi_{k'}, \forall k \\ & \sum_{k=1}^e Q_k \leq 1, Q_k \geq 0, \forall k \end{aligned} \quad (17)$$

## C) Résolution de contrat optimal

La principale difficulté pour résoudre le problème décrit à l'équation (17) est qu'il existe un grand nombre de contraintes IR et IC ( $k$  pour IR et  $k \times (k - 1)$  Contraintes IC). Par conséquent, pour rendre (17) plus simple, nous simplifions premièrement les contraintes IR et IC, puis nous résolvons le problème.

### i. Simplification de contrat

Afin de résoudre le problème dans (17), nous devons simplifier les contraintes IR et IC avant de les résoudre. Nous simplifions ces contraintes en suivant la méthode standard décrite dans [88]. Premièrement, nous exprimons les contraintes IR par le lemme 1.

**lemme1 :** pour la solution optimale, étant donné que l'IC est satisfaite, la contrainte IR pour le type le plus bas est une liaison, c'est-à-dire :

$$r(\theta_1, Q_1) - \pi_1 = 0 \quad (18)$$

**preuve1 :** on a par définition des types de CPs:  $\theta_1 \leq \dots \leq \theta_k \leq \dots \leq \theta_e$ . Aussi, d'après les contraintes d'IC dans (17), nous avons  $r(\theta_k, Q_k) - \pi_k \geq r(\theta_k, Q_1) - \pi_1 \geq r(\theta_1, Q_1) - \pi_1$ . Tant

que  $r_k(\theta_k, Q_k)$  augmente strictement avec  $\theta_k$  et si la contrainte d'IR de CP de type 1 est satisfaite alors :  $r(\theta_1, Q_1) - \pi_1 > 0$ . De plus, du point de vue de PTC, afin de maximiser son profit, le PTC souhaite augmenter le prix autant que possible. En fixant  $r(\theta_1, Q_1) - \pi_1 = 0$ , le PTC obtient le bénéfice maximal. Par conséquent, la condition dans (17) sera vérifiée lorsque le contrat est optimal.

Les contraintes IR peuvent être réduites par le lemme 1, qui indique que le prix  $\pi_1$  pour le type le plus bas  $\theta_1$  doit être égal à l'évaluation de la quantité  $Q_1$ , c'est-à-dire, les gains de type  $\theta_1$  le plus faible ne réalisent aucun bénéfice, tandis que les bénéfices des autres CPs sont supérieurs à ceux du type  $\theta_1$  [42], [88].

Ensuite, nous prouverons que la contrainte IC : peut être réduite dans les lemmes suivants (2,3,4 et 5) :

**lemme2 :** pour tout contrat  $(Q_k, \pi_k)$  de type  $\theta_k$  :  $Q_k > Q_{k'}$  si et seulement si  $\theta_k > \theta_{k'}$  et  $Q_k = Q_{k'}$  si et seulement si  $\theta_k = \theta_{k'}$ .

**preuve2 :** selon la contrainte d'IC, nous avons :

$$r(\theta_k, Q_k) - \pi_k \geq r(\theta_k, Q_{k'}) - \pi_{k'}, \quad \forall k \neq k' \quad (19)$$

et

$$r(\theta_{k'}, Q_{k'}) - \pi_{k'} \geq r(\theta_{k'}, Q_k) - \pi_k, \quad \forall k \neq k' \quad (20)$$

En additionnant les équations ci-dessus, nous aurons :

$$r(\theta_k, Q_k) + r(\theta_{k'}, Q_{k'}) \geq r(\theta_k, Q_{k'}) + r(\theta_{k'}, Q_k) \quad (21)$$

En remplaçant (13) par (21), et après quelques manipulations, nous obtenons :

$$(\theta_k - \theta_{k'}) (\log(1 + Q_k)) \geq (\theta_k - \theta_{k'}) (\log(1 + Q_{k'})) \quad (22)$$

1) SI  $\theta_k > \theta_{k'}$  Alors  $Q_k > Q_{k'}$  : on a  $Q_k > Q_{k'}$ , cela implique  $\theta_k - \theta_{k'} > 0$ . En divisant les deux côtés de (22) par  $(\theta_k - \theta_{k'})$  il reste  $\log(1 + Q_k) \geq \log(1 + Q_{k'})$ . De la définition de la fonction logarithme en mathématiques, nous savons que  $\log(1 + Q_k)$  augmente strictement en fonction de  $Q_k$ . Comme  $\log(1 + Q_k) > \log(1 + Q_{k'})$  est vrai, nous pouvons déduire que  $Q_k > Q_{k'}$ .

2) SI  $Q_k > Q_{k'}$  Alors  $\theta_k > \theta_{k'}$  Similaire au premier cas, nous commençons par la contrainte (22). Comme  $Q_k > Q_{k'} > 0$  et  $\log(1 + Q_k)$  augmente strictement avec  $Q_k$ , nous pouvons déduire que  $\log(1 + Q_k) \geq \log(1 + Q_{k'})$ . Par conséquent, on obtient  $\theta_k - \theta_{k'} > 0$ . Ainsi, nous avons prouvé que  $\theta_k > \theta_{k'}$  si et seulement si  $Q_k > Q_{k'}$ .

En utilisant le même processus, nous pouvons facilement prouver que  $\theta_k = \theta_{k'}$  si et seulement si  $Q_k = Q_{k'}$ . Cela implique que la quantité  $Q_k$  augmente de façon monotone

avec le type  $\theta_k$  lorsque le contrat satisfait aux contraintes d'IC. Comme conclusion, nous pouvons dire qu'un CP de type élevé devrait recevoir plus d'espace qu'un CP de type faible. Si deux CPs reçoivent le même espace de mise en cache, ils doivent appartenir au même type, et vice-versa [42], [88]. Donc: Si  $\theta_1 \leq \dots \leq \theta_k \leq \dots \leq \theta_c$ . Alors:

$$0 \leq Q_1 \leq \dots \leq Q_k \leq \dots \leq Q_c \quad (23)$$

**lemme3 :** si le contrat satisfait aux contraintes d'IC, la condition suivante est vraie: Compte tenu de la quantité  $Q_k > Q_{k'}$  si et seulement si le prix satisfait  $\pi_k > \pi_{k'}$ .

**preuve3 :** si le contrat satisfait les contraintes d'IC indiquées en (16), c'est-à-dire :  $r(\theta_k, Q_{k'}) - \pi_{k'} \geq r(\theta_k, Q_k) - \pi_k$ ,  $\forall k \neq k'$  qui peut être réécrit comme :

$$r(\theta_{k'}, Q_{k'}) - r(\theta_k, Q_k) \geq \pi_{k'} - \pi_k, \quad k \neq k'.$$

Puisque  $r(\theta_k, Q_k)$  est une fonction croissante de  $Q_k$  et si  $Q_k > Q_{k'}$  alors :

$$r(\theta_{k'}, Q_{k'}) - r(\theta_k, Q_k) < 0 ; \text{ cela implique que } \pi_{k'} - \pi_k < 0 ; \text{ Donc : } \pi_k > \pi_{k'}.$$

Le lemme 3 présente une propriété importante pour un contrat, c'est-à-dire qu'une quantité supérieure correspond à un prix plus élevé et vice-versa.

D'après le lemme 2 et le lemme 3, nous concluons que : « dans un contrat réalisable, un CP<sub>k</sub> d'un type supérieur  $\theta_k$  peut acheter un espace de caches  $Q_k$  plus grand , ce qui l'aide à un QoE plus élevées » [42].

En combinant les lemmes 2 et 3, nous constatons que les conditions nécessaires aux contraintes d'IC, compte tenu des types  $\theta_1 \leq \dots \leq \theta_k \leq \dots \leq \theta_c$  sont  $0 \leq Q_1 \leq \dots \leq Q_k \leq \dots \leq Q_c$  e  $0 \leq \pi_1 \leq \dots \leq \pi_k \leq \dots \leq \pi_c$ . Par conséquent nous pouvons facilement déduire ce qui suit :

$$0 \leq u_{CP}(1) \leq \dots \leq u_{CP}(k) \leq \dots \leq u_{CP}(C) \quad (24)$$

Les CPs de type supérieur reçoivent plus d'utilité que les CPs de type inférieur. Si un CP sélectionne le contrat conçu pour un type élevé, même s'il reçoit plus d'espace de cache, le profit qui peut gagner ne peut pas compenser le coût de paiement versé au PTC. De plus, si un CP sélectionne un contrat destiné à un type faible, bien que moins de paiement au PTC, le moins d'espace de cache qu'il obtient. Le CP peut recevoir l'utilité maximale si et seulement s'il sélectionne le contrat qui convient le mieux à sa préférence. Ainsi, nous pouvons garantir que le contrat est sincèrement révélé.

D'après [86] et pareil comme dans [42] et [85], les contraintes IC contiennent deux types de contraintes. La contrainte entre le type  $\theta_i$  et le type  $\theta_j$ ,  $j \in \{1, \dots, i-1\}$  est appelée *Downward Incentive Constraints (DIC)*. En particulier, la contrainte entre le type  $\theta_i$ , et le type  $\theta_{i-1}$ , est appelée *Local Downward Incentive Constraints (LDIC)*. Au même temps, les contraintes entre les systèmes de type  $\theta_i$  et de type  $\theta_j$ ,  $j \in \{i+1, \dots, C\}$  est appelé *Up Ward Incentive Constraints (UIC)*, et la contrainte entre le type  $\theta_i$  et le type  $\theta_{i+1}$  est appelée *Local Up Ward Incentive Constraints (LUIC)*.

Tout d'abord, nous montrons que les contraintes DICs peuvent être réduites [88] :

**lemme4 . (Réduire les contraintes LDIC (*Local Downward Incentive Constraints*)) :** si les contraintes LDIC sont satisfaites pour un type  $\theta_k$  de CP , c'est-à-dire:

$$r(\theta_k, Q_k) - \pi_k \geq r(\theta_k, Q_{k-1}) - \pi_{k-1}, \forall k = \{1..C\} \quad (25)$$

Alors les contraintes d'IC seront valables pour tout  $k' \leq k$ , donc :

$$r(\theta_k, Q_k) - \pi_k \geq r(\theta_k, Q_{k'}) - \pi_{k'} \quad (26)$$

**preuve4 :** nous avons deux LDIC comme suit:

$$r(\theta_k, Q_k) - \pi_k \geq r(\theta_k, Q_{k-1}) - \pi_{k-1} \quad (27)$$

$$r(\theta_{k-1}, Q_{k-1}) - \pi_{k-1} \geq r(\theta_{k-1}, Q_{k-2}) - \pi_{k-2} \quad (28)$$

En remplaçant (13) par (28), et après quelques manipulations, nous obtenons :

$$(\theta_{k-1} (\log(1 + Q_{k-1})) - \pi_{k-1}) \geq (\theta_{k-1} (\log(1 + Q_{k-2})) - \pi_{k-2}) \quad (29)$$

Nous avons  $\theta_k > \theta_{k-1}$  , donc l'inégalité dans (29) devient :

$$\theta_k (\log(1 + Q_{k-1}) - \log(1 + Q_{k-2})) \geq \theta_{k-1} (\log(1 + Q_{k-1}) - \log(1 + Q_{k-2})) \geq \pi_{k-1} - \pi_{k-2} \quad (30)$$

De plus, (25) est équivalent à :

$$\theta_{k-1} (\log(1 + Q_{k-1}) - \log(1 + Q_{k-2})) \geq \pi_{k-1} - \pi_{k-2} \quad (31)$$

En additionnant (30) et (31), nous obtenons:

$$\theta_k (\log(1 + Q_k) - \pi_k) \geq \theta_k \log(1 + Q_{k-2}) - \pi_{k-2} \equiv r(\theta_k, Q_k) - \pi_k \geq r(\theta_k, Q_{k-2}) - \pi_{k-2} \quad (32)$$

Par conséquent, si pour le CP de type k-1, le LDIC détient la contrainte d'incitation par rapport au type k-2 . Ce processus peut être étendu vers le bas du type k-2 à 1. Ce qui prouve toutes les contraintes LDICs. Compte tenu de la sélection aléatoire de  $\theta$ , nous avons terminé la preuve [46].

Ensuite, nous montrons que toute les contraintes UIC peuvent être réduites [88] :

**lemme5 (Réduire les contraintes LUIC (*Local Up Ward Incentive Constraints*)):** si les contraintes LUICs sont satisfaites pour tous les types  $\theta_k$  des CPs , c'est-à-dire,

$$r(\theta_k, Q_k) - \pi_k \geq r(\theta_k, Q_{k+1}) - \pi_{k+1}, \forall k = \{1 \dots \mathcal{C}\} \quad (33)$$

Alors les contraintes d'IC seront valables pour tout  $k < k'$  , c'est-à-dire,

$$r(\theta_k, Q_k) - \pi_k \geq r(\theta_k, Q_{k'}) - \pi_{k'} \quad (34)$$

**preuve5.** la preuve est la même que lemme 4.

**lemme6:** (pour réduire les contraintes d'IC): pour maximiser les bénéfices du PTC, les contraintes d'IC peuvent être remplacées par :

$$r(\theta_k, Q_k) - \pi_k = r(\theta_k, Q_{k-1}) - \pi_{k-1} \quad (35)$$

**lemme7:** soit  $\{Q_k, \pi_k\}$  un contrat réalisable, alors l'unique prix optimal satisfait :

$$\pi_1^* = r(\theta_1, Q_1) = \theta_1 \log(1 + Q_1) \quad (36)$$

$$\pi_k^* = \pi_{k-1}^* + \theta_k (\log(1 + Q_k) - \log(1 + Q_{k-1})) \quad (37)$$

**preuve [46]:** compte tenu de la quantité de mise en cache est fixe, l'utilité de PTC donc est décidée par  $\sum_{i=1}^{\mathcal{C}} \varphi_i \pi_i$ , Supposons qu'il existe un autre paiement possible  $\{\pi'_k, \forall k\}$ , qui est considéré comme meilleure solution que  $\{\pi_k^*, \forall k\}$  dans (37) , Ainsi, il y a au moins un prix  $\pi'_k > \pi_k^*$  pour un type  $\theta_k$ .

1) Si  $k = 1$ : alors  $\pi'_1 > \pi_1^*$  et  $\pi_1^* = \theta_1 \log(1 + Q_1)$  Par conséquent :  $\pi'_1 > \theta_1 \log(1 + Q_1)$  qui ne respecte pas les contraintes IR pour le type  $\theta_1$ .

2) Si  $k > 1$  :  $\{\pi'_k, \forall k\}$  doivent satisfaire le LDIC (26) alors :

$$\theta_k \log(1 + Q_k) - \pi'_k \geq \theta_k \log(1 + Q_{k-1}) - \pi'_{k-1} \text{ ce qui implique :}$$

$$\pi'_k \leq \pi'_{k-1} + \theta_k (\log(1 + Q_k) - \log(1 + Q_{k-1})) .$$

En remplaçant  $\pi_k^* = \pi_{k-1}^* + \theta_k (\log(1 + Q_k) - \log(1 + Q_{k-1}))$  dans cette expression de comparaison  $\pi'_k \leq \pi'_{k-1} + \theta_k (\log(1 + Q_k) - \log(1 + Q_{k-1}))$  nous obtenons :  $\pi'_{k-1} \geq \pi_{k-1}^*$  qui implique par induction  $\pi'_1 \geq \pi_1^*$  : qui ne respecte pas les contraintes IR pour le type  $\theta_1$ . Avec la démonstration par l'absurde, nous obtenons (37).

Après la simplification de (37), nous pouvons conclure que :

$$\pi_k^* = \theta_1 \log(1 + Q_1) + \sum_{i=2}^k w_i \quad , \quad \forall k = \{1, 2, \dots, \mathcal{C}\} \quad (38)$$

$$\text{Où : } W_i = \begin{cases} 0 & \text{si } i = 1 \\ \theta_i (\log(1 + Q_i) - \log(1 + Q_{i-1})) & \text{si } i = 2 \dots k \end{cases}$$

## ii. Optimalité du contrat

Compte tenu des contraintes de faisabilité d'un contrat, nous formulerons ci-après le problème d'optimisation du système. En utilisant les lemmes dans la sous-section précédente, le problème d'optimisation dans (17) peut être réécrit comme :

$$\begin{aligned}
 (Q_k^*, \pi_k^*) &= \operatorname{argmax}_{Q_k, \pi_k} \sum_{k=1}^c \varphi_k (\theta_1 \log(1 + Q_1) + \sum_{i=1}^k w_i - c(Q_k)) \\
 \text{s.c : } &0 \leq Q_1 \leq \dots \leq Q_k, \leq \dots \leq Q_c \\
 \text{IR(18) : } &r(\theta_1, Q_1) - \pi_1 = 0 \\
 \text{IC(35) : } &r(\theta_k, Q_k) - \pi_k = r(\theta_k, Q_{k-1}) - \pi_{k-1} \\
 &\sum_{k=1}^c Q_k \leq 1, Q_k \geq 0, \forall k
 \end{aligned} \tag{39}$$

Où :

$$W_i = \begin{cases} 0 & \text{si } i = 1 \\ \theta_i (\log(1 + Q_i) - \log(1 + Q_{i-1})) & \text{si } i = 2 \dots k \end{cases} \tag{40}$$

Afin de résoudre le problème (39), une approche standard consiste à ignorer la condition de monotonie (23) dans un premier temps, puis de vérifier si la solution remplit cette condition [46], [87]. Après avoir supprimé la condition de monotonie, on peut réécrire le problème d'optimisation en (39) comme suite:

$$\begin{aligned}
 (Q_k^*, \pi_k^*) &= \operatorname{argmax}_{Q_k, \pi_k} \sum_{k=1}^c u_{ptc}(k) \\
 \text{s.c : } &0 \leq Q_1 \leq \dots \leq Q_k \leq \dots \leq Q_c \\
 \text{IR(18) : } &r(\theta_1, Q_1) - \pi_1 = 0 \\
 \text{IC(35) : } &r(\theta_k, Q_k) - \pi_k = r(\theta_k, Q_{k-1}) - \pi_{k-1} \\
 &\sum_{k=1}^c Q_k \leq 1, Q_k \geq 0, \forall k
 \end{aligned} \tag{41}$$

Tel que:

$$u_{PTC}(k) = \varphi_k \theta_k \log(1 + Q_k) + \omega_k \left( \sum_{i=k+1}^c \varphi_i \right) - \varphi_k c(Q_k) \tag{42}$$

Et:  $\omega_k = (\theta_k - \theta_{k+1}) \log(1 + Q_k), \forall k = \{1, 2, \dots, C-1\}$  et  $\omega_k = 0$  si  $k = C$

Et :  $\varphi_k = \frac{1}{C}, \forall k$

Et (42) peut être réécrit comme suit :

$$u_{PTC}(k) = \varphi_k (\theta_k \log(1 + Q_k) + (C - k)(\theta_k - \theta_{k+1}) \log(1 + Q_k) - c(Q_k)) \tag{43}$$

En simplifiant l'expression d' $u_{ptc}(k)$ :

$$u_{PTC}(k) = \varphi_k [\theta_k (C - k + 1) \log(1 + Q_k) + ((k - C) \theta_{k+1} \log(1 + Q_k)) - c(Q_k)] \tag{44}$$

Nous calculons les dérivées partielles par rapport à  $Q_k$  Tel que  $k = 1, 2, \dots, C$  :

$$\frac{du_{PTC}(k)}{dQ_k} = \left( \varphi_k \left( ((C-k+1)\theta_k \log(1+Q_k)) + ((k-C)\theta_{k+1} \log(1+Q_k)) - (\mu * P^t * Q_k + \eta_k * \sigma_k) \right) \right)' \quad (45)$$

$$= \varphi_k \left( \frac{\theta_k(C-k+1)}{(1+Q_k)} + \frac{\theta_{k+1}(k-C)}{(1+Q_k)} - (\mu * P^t) \right) \quad (46)$$

Nous calculons la deuxième dérivée de l'utilité de PTC par rapport à  $Q_k$ :

$$\frac{d^2 u_{PTC}(k)}{d^2 Q_k} = \left( \varphi_k \left( \frac{\theta_k(C-k+1)}{(1+Q_k)} + \frac{\theta_{k+1}(k-C)}{(1+Q_k)} - (\mu * P^t) \right) \right)'' \quad (47)$$

$$= \varphi_k \left( \frac{-(n-k+1)}{(1+Q_k)^2} \theta_k + \frac{-(k-C)}{(1+Q_k)^2} \theta_{k+1} \right) \quad (48)$$

$$= \varphi_k \left( -\frac{(c-k)\theta_k}{(1+Q_k)^2} - \frac{\theta_k}{(1+Q_k)^2} + \frac{(C-k)\theta_{k+1}}{(1+Q_k)^2} \right) \quad (49)$$

$$= \varphi_k \left( -\frac{(c-k)\theta_k}{(1+Q_k)^2} - \frac{\theta_k}{(1+Q_k)^2} + \frac{(C-k)\theta_{k+1}}{(1+Q_k)^2} \right) \quad (50)$$

Par simplification de l'expression

$$\frac{d^2 u_{PTC}(k)}{d^2 Q_k} = \varphi_k \left( -\left( \frac{(C-k)\theta_k}{(1+Q_k)^2} + \frac{\theta_k}{(1+Q_k)^2} \right) + \frac{(C-k)\theta_{k+1}}{(1+Q_k)^2} \right) \quad (51)$$

$$\rightarrow \frac{d^2 u_{PTC}(k)}{d^2 Q_k} > 0$$

**Preuve :** on a  $(1+Q_k)^2 > 0$  et tantque  $S > Q_k$  Alors :  $(S-Q_k)^3 > 0$

D'après les lemmes de la sous-section précédente :  $\theta_{k+1} > \theta_k > 0$  et  $n \geq k, \forall k$

Donc :

$$\frac{\theta_{k+1}}{(1+Q_k)^2} > \frac{-\theta_k}{(1+Q_k)^2} \quad (52)$$

En multipliant (52) par  $(C-k)$  ; nous obtenons :

$$\frac{(c-k)\theta_{k+1}}{(1+Q_k)^2} > \frac{-(C-k)\theta_k}{(1+Q_k)^2} \quad (53)$$

En soustrayant  $\left(\frac{\theta_k}{(1+Q_k)^2}\right)$  des deux côtés de l'expression , nous obtenons :

$$\frac{(C-k)\theta_{k+1}}{(1+Q_k)^2} - \frac{\theta_k}{(1+Q_k)^2} > -\frac{(C-k)\theta_k}{(1+Q_k)^2} - \frac{\theta_k}{(1+Q_k)^2} \quad (54)$$

Nous concluons que :

$$\frac{(C-k)\theta_{k+1}}{(1+Q_k)^2} > -\frac{(C-k)\theta_k}{(1+Q_k)^2} - \frac{\theta_k}{(1+Q_k)^2} \quad (55)$$

Donc :

$$\varphi_k \left( -\left( \frac{(C-k)\theta_k}{(1+Q_k)^2} + \frac{\theta_k}{(1+Q_k)^2} \right) + \frac{(C-k)\theta_{k+1}}{(1+Q_k)^2} \right) > 0 \quad (56)$$

Alors  $:u_{ptc}(k) > 0$ . On'a  $u_{ptc}(k) > 0$  donc la fonction d'utilité de PTC est une fonction convexe par rapport à  $Q_k$ ,  $\forall k$ . Par conséquent le problème d'optimisation dans (41), est un problème d'optimisation convexe standard, qui peut être résolu par la méthode du multiplicateur lagrangien [3], [46] et [87]. Nous construisons la fonction de Lagrange comme suit :

$$L = \sum_{k=1}^c u_{pTC}(k) + \left( \partial^1 \sum_{k=1}^c Q_k - 1 \right) + \sum_{k=1}^c (\partial_k^2 Q_k) \quad (57)$$

Où  $\partial^1$  et  $\partial_k^2$  sont les multiplicateurs de Lagrange. La valeur optimale de la quantité des fichiers populaires ( $Q_k^*$ ) est obtenue en appliquant la condition de Karush-Kuhn-Tucker. Pour cela, il suffit de mettre la première dérivée de  $L = 0$ . Ici,  $Q_k^*, k = \{1, 2, \dots, C\}$  sont des solutions de :

$$\begin{cases} u'_{pTC}(k) + \partial^1 + \partial_k^2 = 0, \forall k = \{1, 2, \dots, C\} \\ \partial^1 (\sum_{k=1}^c Q_k - 1) = 0 \\ \partial_k^2 Q_k = 0, \forall k = \{1, 2, \dots, C\} \end{cases} \quad (58)$$

Où  $u'_{pTC}(k)$  est la dérivée de premier ordre de  $u_{pTC}(k)$  par rapport à  $Q_k$ . En outre, nous devons vérifier si ces solutions sont réalisables. Si ( $Q_k^*$ ) satisfait à la condition de monotonie (24) et les lemme(2) et (3), elle peut être considéré comme notre quantité optimale  $Q_k^* = \overline{Q}_k^*, \forall k = \{1, 2, \dots, C\}$ . Sinon, nous devons faire quelques ajustements basés sur la proposition suivante [46], [89].

**Proposition 1 :** soit  $X_1(x)$  et  $X_2(x)$  des fonctions concaves sur  $x$ . Si  $\overline{x}_1 \geq \overline{x}_2$  ou  $\overline{x}_1 = \operatorname{argmax}_{x_1} X_1(x)$  et  $\overline{x}_2 = \operatorname{argmax}_{x_2} X_2(x)$  alors  $\overline{x}_1 = \overline{x}_2$  où :

$$\{\overline{x}_1, \overline{x}_2\} = \operatorname{argmax}_{x_1, x_2} \sum_{i=1}^2 X_i(x_i) \quad , \quad \text{s.c. } x_1 \leq x_2 \quad (59)$$

**Preuve :** la preuve de la proposition 1 est donnée dans [89].

La proposition 1 peut être étendue à une forme plus générale : si  $\bar{x}_1 \geq \bar{x}_2 \geq x_3 \geq \dots \geq \bar{x}_c$   
Où  $\bar{x}_k = \operatorname{argmax}_{x_1, x_2} \sum_{k=1}^c X_k(x_k)$  alors  $\bar{x}_1 = \bar{x}_2 = \dots = \bar{x}_e$  où :

$$\{\bar{x}_k\} = \operatorname{argmax}_{x_k} \sum_{k=1}^e X_k(x_k) \quad , \quad \text{s. c. } x_1 \leq x_2 \leq \dots \leq x_c \quad (60)$$

On note une sous-séquence de  $\{\bar{Q}_k^*\}$ , disons  $\{\bar{Q}_i^*, \bar{Q}_{i+1}^*, \dots, \bar{Q}_j^*\}$  comme sous-séquence ir-réalisable, Si  $\bar{Q}_i^* \geq \bar{Q}_{i+1}^* \geq \dots \geq \bar{Q}_j^*$ , par exemple : dans une allocation de quantité  $\{\bar{Q}_k^*\} = \{1, 3, 3, 2, 6, 5\}$  il y a deux sous-séquences possibles, c'est-à-dire  $\{\bar{Q}_1^*, \bar{Q}_2^*, \bar{Q}_3^*\}$  et  $\{\bar{Q}_5^*, \bar{Q}_6^*\}$  Selon la proposition 1, les valeurs ajustées satisfont  $Q_i^* = Q_{i+1}^* = \dots = Q_j^*$ . De plus, devrait maintenir les contraintes d'appartenance à un type, c'est-à-dire :

$$\varphi_i Q_i^* + \varphi_{i+1} Q_{i+1}^* + \dots + \varphi_j Q_j^* = \varphi_i \bar{Q}_i^* + \varphi_{i+1} \bar{Q}_{i+1}^* + \dots + \varphi_j \bar{Q}_j^* \quad (61)$$

Donc :

$$Q_i^* = Q_{i+1}^* = \dots = Q_j^* = \frac{\varphi_i \bar{Q}_i^* + \varphi_{i+1} \bar{Q}_{i+1}^* + \varphi_j \bar{Q}_j^*}{\varphi_i + \varphi_{i+1} + \dots + \varphi_j} \quad (62)$$

En substituant la quantité optimale  $Q_i^*$  en (38) nous obtenons le prix optimal correspondant  $\pi_k^*$  comme suit :

$$\pi_k^* = \theta_1 \log(1 + Q_1^*) + \sum_{i=1}^k W_i^* \quad , \quad \forall k = \{1, 2, \dots, n\} \quad (63)$$

Pour :  $k = \{1, 2, \dots, n\}$

$$W_i^* = \begin{cases} 0 & \text{si } i = 1 \\ \theta_i (\log(1 + Q_i^*) - \log(1 + Q_{i-1}^*)) & \text{si } i = 2 \dots k \end{cases} \quad (64)$$

Dans ce qui suit, nous donnons dans l'algorithme 1 l'ensemble des étapes (algorithme) qui permet le calcul de contrat optimal .

---

#### Algorithme 1 : Algorithme de calcul de contrat optimal

---

**Entrée :**  $\mathcal{C}, \varphi_k, \theta = \{\theta_1, \theta_2, \dots, \theta_e\}$

**Sortie :**  $Q, \pi$

**Début**

1. **Pour chaque type  $\{k = 1 \dots C\}$  Faire**

a) Calculer :

$$u_{\text{PTC}}(k) = \varphi_k (\theta_k \log(1 + Q_k) + (C - k)(\theta_k - \theta_{k+1}) \log(1 + Q_k) - c(Q_k)) \quad (65)$$

b) Calculer :

$$Q_i^* = \begin{cases} u'_{\text{ptc}}(k) + \partial^1 + \partial_k^2 = 0 & , \forall = \{1, 2, \dots, c\} \\ \partial^1 \left( \sum_{k=1}^c Q_k - 1 \right) = 0 \\ \partial_k^2 Q_k = 0 & , \forall = \{1, 2, \dots, C\} \end{cases} \quad (66)$$

2. **Fin Pour.**

3. **Tant que  $Q_i^*$  n'est pas faisable**

a) Trouver une sous-séquence infaisable  $\{\overline{Q}_i^*, \overline{Q}_{i+1}^*, \dots, \overline{Q}_j^*\}$

b) Calculer  $Q_i^* = Q_{i+1}^* = \dots = Q_j^* = \frac{\varphi_i \overline{Q}_i^* + \varphi_{i+1} \overline{Q}_{i+1}^* + \varphi_j \overline{Q}_j^*}{\varphi_i + \varphi_{i+1} + \dots + \varphi_j}$

4. **Fin Tant que.**

5. **Pour  $k = 1 \dots C$  Faire**

6. Calculer  $W_i = \begin{cases} 0 & \text{si } i = 1 \\ \theta_i (\log(1 + Q_i) - \log(1 + Q_{i-1})) & \text{si } i = 2 \dots C \end{cases}$

7. Définir :  $\pi_k^* = \theta_1 \log(1 + Q_1) + \sum_{i=1}^k W_i$

8. **Fin Pour.**

**Fin**

---

#### 4.2.4 Politique d'incitation dans un marché de mise en cache compétitif

Dans cette politique, nous considérons un marché de mise en cache compétitif composé de plusieurs PTCs et plusieurs CPs. Notre idée consiste à placer les contenus populaires des CPs dans les caches embarqués sur les véhicules des PTCs qui sont déployés dans les endroits plus proches des utilisateurs mobiles. Dans notre politique, les CPs ont besoin des caches embarqués sur les véhicules des PTCs pour stocker leurs contenus, et les PTCs eux aussi ont besoin des CPs et plus particulièrement de leurs contenus pour remplir leurs caches. Par conséquent,

ces deux acteurs ont intérêt à collaborer les uns avec les autres pour un déploiement plus efficace de mise en cache.

Nous considérons dans cette sous-section, une politique d'incitation à la mise en cache compétitive dans les réseaux IoVs avec des informations symétriques. Dans notre politique, les PTCs encouragent les CPs à venir stocker leurs contenus populaires sur leurs caches embarqués sur les véhicules mobiles. Les PTCs utilisent le prix comme un moyen d'encouragement des CPs. À cause de la taille limitée de l'ensemble de l'espace de stockage en cache offert par les PTCs, les CPs vont entrer en compétition sur cette ressource limitée. Aussi, à cause de nombre limité des fichiers populaires des CPs, les PTCs vont aussi entrer en compétition pour stocker le maximum des fichiers populaires des CPs.

Les PTCs visent à gagner plus de profit monétaire, tant dis que les CPs visent à améliorer la QoE de leurs utilisateurs. Ce scénario se déroule avec la certitude que les CPs sont honnêtes lors de la transmission/déclaration de leurs informations de contenu aux PTCs, les valeurs exactes de leurs popularités des fichiers par exemple.

#### 4.2.4.1 Approche basée sur un jeu de type Stackelberg pour l'incitation à la mise en cache compétitive dans les réseaux d'IoV

Afin de modéliser notre politique d'incitation à la mise en cache, nous utilisons un jeu de type Stackelberg. Ce dernier est un jeu séquentiel qui permet de modéliser l'interaction compétitive entre deux types d'acteurs rationnels, à savoir les leaders et les suiveurs.

Dans notre jeu, les PTCs agissent d'abord en tant que leaders en annonçant leurs stratégies qui consistent en un doublet : le prix initial de mise en cache ( $\pi_i, i = \overline{1, m}$ ) et la taille globale de stockage disponible ( $S_i, i = \overline{1, m}$ ). En allouant leurs caches embarqués sur les véhicules aux CPs, les PTCs visent à maximiser leurs profits monétaires de l'opération de mise en cache. Les CPs réagissent après comme des suiveurs en répondant par leurs stratégies qui consistent à déterminer les quantités optimales de leurs contenus populaires ( $Q_{i,j}^*$ ) qu'ils souhaitent stocker en cache. En mettant en cache le maximum de leurs contenus les plus populaires dans les véhicules ayant le taux d'accès le plus élevé, chaque CP vise à maximiser la satisfaction de ses utilisateurs via l'amélioration de la QoE. Enfin, chaque PTC, après avoir reçu toutes les réponses des CPs, optimisera sa propre stratégie tout en prenant en considération les stratégies des autres PTCs. Cela, par le calcul du prix optimal de la mise en cache qui sera annoncé ensuite pour tous les CPs.

Ainsi, sur la base du scénario décrit ci-dessus et à cause de la limitation des ressources des CPs et des PTCs, on peut distinguer deux sous-jeux non-coopératifs. Un premier jeu non-coopératif qui modélise la compétition entre les suiveurs (CPs) sur la quantité limitée de l'espace de stockage des PTCs. Un deuxième jeu non-coopératif pour modéliser le conflit entre les leaders (PTCs) sur le nombre limité des fichiers populaires des CPs proposés pour le stockage en cache. Dans ce qui suit, nous fournissons les fonctions des utilités des leaders et

des suiveurs, respectivement. La fonction d'utilité est exprimée comme une fonction générale, telle que  $Utilité = Gain - Coût$ .

### 1) Utilité du leader (PTC)

Dans cette sous-section, nous décrirons mathématiquement les facteurs qui peuvent influencer le gain, le coût et par la suite l'utilité de chaque leader, tout en respectant les stratégies discutées précédemment. Afin de les noter sous forme de formules mathématiques.

#### a. Gain

Le gain de chaque leader ( $PTC_i$ ) dépend principalement des profits monétaires obtenus en stockant le contenu populaire des CPs. Nous formalisons le gain du  $PTC_i$ , Notée par  $R_{PTC_i}$ , comme une fonction monotone croissante en fonction de la quantité de contenus stockés, noté  $Q_{i,j}$ , dans ses caches embarqués sur les véhicules et du prix unitaire de stockage, noté  $\pi_i$ , d'un seul fichier de CP.  $R_{PTC_i}$  est donné par :

$$R_{PTC_i} = \sum_{j=1}^n \pi_i \times Q_{i,j} \quad (67)$$

Tel que la valeur du prix initiale de  $\pi_{i,j}$  est calculée en utilisant une fonction sigmoïde croissante. Cette fonction permet de définir des niveaux de prix suivant l'écart type de la popularité des fichiers  $\sigma_{i,j}$  de  $CP_j$  dont les valeurs varient dans l'intervalle  $[0, 1]$ .  $\pi_{i,j}$  est donné par :

$$\pi_{i,j} = \frac{1}{1 - e^{-\sigma_{i,j}}} \quad (68)$$

Où  $\sigma_{i,j}$  dénote l'écart type de la popularité des fichiers de  $CP_j$  par rapport aux utilisateurs de  $PTC_i$ .  $\sigma_{i,j}$  est calculé comme suit :

$$\sigma_{i,j} = \sqrt{\frac{\sum_{k=1}^{Q^t} \left( p_{i,j,k} - \frac{\sum_{k=1}^{Q^t} p_{i,j,k}}{Q^t} \right)^2}{Q^t}} \quad (69)$$

Tel que :  $Q^t$  dénote la quantité totale des fichiers de  $CP_j$  stocké sur les caches embarqués sur les véhicules de  $PTC_i$ , et  $p_{i,j,k}$ : la popularité des fichiers stockés en cache.

### b. Coût

Pour chaque  $PTC_i$  le coût de la mise en cache des quantités  $Q_{i,j}$  de contenu dans les caches, noté  $C_{PTC_i}$ , pour tout  $CP_j$  ( $j = \overrightarrow{1, n}$ ) est égale au coût en termes d'énergie de transmission ou de partage des fichiers avec le voisinage des utilisateurs de CP plus le coût énergétique qu'il consomme pour gérer les différents accès aux fichiers stockés en cache [44].  $C_{PTC_i}$  est donné par :

$$C_{PTC_i} = \mu \sum_{j=1}^n P^t Q_{i,j} + \eta_i \sum_{j=1}^n \sigma_{i,j} \quad (70)$$

Où :

$\mu$  : Dénote l'unité de coût de transmission et de maintenance.

$P^t$  : Dénote la puissance de transmission d'un fichier de CP.

$\sigma_{i,j}$  : Dénote l'écart type de popularité du contenu de  $CP_j$  connu par  $PTC_i$ .

$\eta_i$  : Dénote le coût d'énergie que consomme le  $PTC_i$  pour gérer les accès aux fichiers stockés dans ses caches.

### c. Fonction d'utilité

A partir de 67 et 70, la fonction d'utilité du leader  $PTC_i$ , notée par  $U_{PTC_i}$ , peut être exprimée comme une fonction croissante en termes du gain et décroissante par rapport au coût.  $U_{PTC_i}$  est donnée par :

$$U_{PTC_i} = R_{PTC_i} - C_{PTC_i} = \sum_{j=1}^n \pi_i Q_{i,j} - \sum_{j=1}^n (\mu P^t Q_{i,j} + \eta_i \sigma_{i,j}) \quad (71)$$

## 2) Utilité du suiveur (CP)

L'utilité de CP peut être exprimée comme une fonction générale en termes du gain et du coût de CP.

### a. Gain

Chaque  $CP_j$  vise à améliorer la satisfaction de ses utilisateurs via la maximisation de la quantité de ses fichiers populaires stockés sur l'ensemble des caches sur les véhicules des PTC  $i$  avec la plus grande popularité. Le gain de  $CP_j$ , noté par  $R_{CP_j}$ , est exprimé en termes de la quantité de contenu stocké sur les caches des véhicules et sa popularité, comme ci-dessous :

$$R_{CP_j} = \sum_{i=1}^m (\sigma_{i,j} - \tau_{i,j}) \times \log(1 + Q_{i,j}) \quad (72)$$

Tel que  $\tau_{i,j}$  : dénote un seuil de popularité de la mise en cache, donné par le  $CP_j$  et attribué comme une condition nécessaire que l'endroit où ses fichiers sont placés doivent garantir un taux d'accès, i.e., le désir de stocker un fichier dans le cache sur un véhicules avec un taux d'accès  $\geq \tau_{i,j}$ , cette condition doit être respectée par le  $PTC_i$ . Ce seuil est calculé à l'aide d'une fonction sigmoïde définie en fonction de l'écart-type de popularité des fichiers  $\sigma_{i,j}$ , comme suite :

$$\tau_{i,j} = \frac{1}{1 - e^{-\sigma_{i,j}}} \quad (73)$$

### b. Coût

Le coût principal du suiveur, noté  $C_{CP_j}$ , est égale aux profits monétaires payés par le  $CP_j$  en contrepartie de stockage de la quantité de son contenu populaire au niveau de chaque  $PTC_i$  avec  $i = \overline{1, m}$ .  $C_{CP_j}$  est calculé par la fonction suivante :

$$C_{CP_j} = \sum_{i=1}^m \pi_i Q_{i,j} \quad (74)$$

Où :  $\pi_i$  c'est le prix unitaire proposé par  $PTC_i$ ,  $Q_{i,j}$  dénote la quantité de contenus populaires de  $CP_j$  stockés par le  $PTC_i$  et  $m$  dénote le nombre des sociétés de transport PTCs.

### c. Fonction d'utilité

A partir de 72 et 74, la fonction d'utilité de  $CP_j$ , notée par  $U_{CP_j}$ , peut être exprimée comme une fonction croissante en termes de gain (satisfaction) et décroissante en termes de coût monétaire de mise en cache.  $U_{CP_j}$  est donnée comme suit :

$$U_{CP_j} = R_{CP_j} - C_{CP_j} = \sum_{i=1}^m ((\sigma_{i,j} - \tau_{i,j}) * \log(1 + Q_{i,j}) - \pi_i Q_{i,j}) \quad (75)$$

Les joueurs doivent garantir que leurs utilités soient toujours positives, en s'assurant toujours que le coût reste toujours inférieur par rapport au gain.

Dans ce qui suit, nous détaillons les deux sous-jeux non coopératifs entre les leaders et après entre les suiveurs.

### 3) Sous-jeu non-coopératif des leaders

En raison de nombre de fichiers populaires limité des CPs, les leaders entrent en concurrence sur cette ressource afin d'optimiser leurs utilités. Cette compétition est modélisée à l'aide d'un sous-jeu non coopératif, défini comme suit.

**Définition 1.** le sous-jeu non-coopératif entre les leaders est défini par  $J_1 = (T, S, U)$  :

- $T = \{1, \dots, M\}$ : représente l'ensemble fini et non vide des joueurs, *i.e.*, les sociétés de transport en commun des PTCs.
- $S = \{s_1, \dots, s_m\}$ : représente l'ensemble fini et non vide des stratégies de PTCs où  $s_i$ , tel que  $i = \overline{1, m}$ , est la stratégie de leader  $i$ , exprimée en fonction de la taille globale de stockage et le prix initial de la mise en cache ( $S_i, \pi_i$ ).
- $U = \{U_{PTC_1}, \dots, U_{PTC_m}\}$  dénote l'ensemble fini des fonctions d'utilité des leaders. La fonction d'utilité du joueur  $i$  est définie dans l'équation (71).

### 4) Sous-jeu non-coopératif des suiveurs

Le deuxième sous-jeu non-coopératif est formulé pour modéliser la concurrence entre les suiveurs (CPs) sur l'espace de stockage disponible des PTCs pour la mise en cache.

**Définition 2.**  $J_2 = (C, S, U)$  est un jeu non-coopérative entre les suiveurs tel que :

- $C = \{1, \dots, N\}$ : représente l'ensemble fini et non vide des suiveurs, *i.e.*, les fournisseurs de contenu CPs.
- $S = \{s_1, \dots, s_n\}$ : représente l'ensemble fini et non vide des stratégies de CPs où  $s_j$  (tel que  $i = \overline{1, m}$ ) est la stratégie de suiveur  $j$ , qui est exprimée en termes de la quantité qui souhaite stockée en mémoire cache embarquée sur les véhicules des PTCs.
- $U = \{U_{CP_1}, \dots, U_{CP_m}\}$  dénote l'ensemble fini des fonctions d'utilité des suiveurs. La fonction d'utilité du suiveur  $j$  est définie dans l'équation (75).

Dans ce sous-jeu, la situation d'équilibre pour tous les CPs n'aura lieu que si aucun CP n'espère améliorer son utilité en modifiant les quantités de son contenus/fichiers stockés dans les caches des PTCs.

L'équilibre de Nash (NE) représente la solution optimale des jeux non-coopératifs avec information complète et des stratégies pures. Un équilibre de Nash correspond à une situation d'équilibre où aucun joueur n'a intérêt à dévier unilatéralement de cette situation c'est-à-dire aucun PTC ne souhaite améliorer ces bénéfices en stockant un nombre différent de fichier ou en modifiant le prix de mise en cache.

Dans la section suivante, nous cherchons à prouver l'existence de l'équilibre de Stackelberg et après le calcul des stratégies optimales des joueurs.

### 5) Equilibre de Stackelberg

Notre jeu de type Stackelberg est composé de deux sous-jeux non-coopératif : le premier est entre les leaders et le second entre les suiveurs, chacun des deux jeux est un problème d'optimisation. Les leaders (PTCs) visent à remplir au maximum leurs caches embarqués sur les véhicules pour maximiser leurs gains monétaires, comme ils cherchent également à réduire les coûts de gestion du trafic de données. Les suiveurs (CPs) visent à maximiser la satisfaction de leurs utilisateurs via l'amélioration de la QoE, le temps de téléchargement/accès, etc. Cela, peut se faire en stockant en cache le maximum des contenus les plus populaires, tout en respectant un seuil de popularité, autrement dit, vise à stocker dans les caches des véhicules ayant le taux d'accès le plus élevé, en essayant de rationaliser le coût de mise en cache.

Le but principal de notre jeu est de trouver l'équilibre de Stackelberg (ES) et les valeurs optimales des stratégies des joueurs. Par définition l'équilibre fait référence à une situation mutuellement satisfaisante pour tous les joueurs. Pour laquelle, aucun joueur, n'a la motivation à dévier unilatéralement en espérant améliorer son utilité [41] et [53]. En d'autres termes, ni les leaders (PTCs) ni les suiveurs (CPs) n'ont intérêt à s'écarter du point d'équilibre de Stackelberg. Pour être plus précis, L'ES est exactement le point que les leaders et les suiveurs souhaitent atteindre [33].

Nous utilisons l'induction par rebours pour la solution de notre jeu [33] [41] [50]. Pour cela, nous commençons par chercher la meilleure réponse des suiveurs dans l'équilibre de Nash de sous jeu non-coopératif. Ensuite, nous cherchons la solution optimale des leaders dans l'équilibre de Stackelberg.

#### a) Stratégies optimales des suiveurs

Considérons la stratégie annoncée par chaque leader, représentée par le doublet  $(S_i, \pi_i)$ , les suiveurs vont entrer en compétition entre eux pour l'espace limitée annoncée pour la mise en cache. Chaque  $CP_j$  commencera par optimiser sa stratégie qui consiste à déterminer la quantité optimale des fichiers populaires qu'il va stocker dans les caches de chaque  $PTC_i$ , noté par  $Q_{i,j}^*$ , en fonction du prix annoncé par chaque  $PTC_i, i.e., \pi_i$ . Cette quantité  $Q_{i,j}^*$  est obtenue en résolvant le problème d'optimisation suivant :

$$\begin{aligned} Q_{i,j} &= \operatorname{argmax}_{Q_{ij}} U_{CP_j}(Q_{i,j}, \pi_i) \\ &\text{s.t } 0 < Q_{i,j} \leq S_i \\ &\quad \pi_i > 0 \end{aligned} \tag{76}$$

Tel que  $U_{CP_j}$  est défini dans l'équation (75).

Pour résoudre le problème d'optimisation du suiveur dans (76), nous étudions la meilleure réponse de  $CP_j$ . Pour ce but, nous vérifions tout simplement si ce problème est une optimisation concave en procédant par le calcul des dérivées partielles du premier et de second ordre de la fonction d'utilité  $U_{CP_j}$  pour le suiveur  $j$  par rapport à  $Q_{i,j}$ .

### i. Calcul de la dérivée de premier ordre

$$\frac{dU_{CP_j}}{dQ_{ij}} = \left( \sum_{i=1}^m ((\sigma_{ij} - \tau_{ij}) * \log(1 - Q_{ij}) - \pi_i Q_{ij}) \right)' \quad (77)$$

Application des formules de différentielles :

$$\frac{dU_{CP_j}}{dQ_{ij}} = \sum_{i=1}^m \frac{d}{dQ_{ij}} ((\sigma_{ij} - \tau_{ij}) * \log(1 - Q_{ij}) - \pi_i Q_{ij}) \quad (78)$$

On peut déduire pour  $Q_{i,j}$  tel que  $i = \overline{1, m}$  :

$$\begin{aligned} \frac{dU_{CP_j}}{dQ_{1,j}} &= \left( \frac{(\sigma_{1,j} - \tau_{1,j})}{(1 + Q_{1,j})} - \pi_1 + 0 + \dots + 0 \right) \\ \frac{dU_{CP_j}}{dQ_{2,j}} &= \left( 0 + \frac{(\sigma_{2,j} - \tau_{2,j})}{(1 + Q_{2,j})} - \pi_2 + \dots + 0 \right) \\ \frac{dU_{CP_j}}{dQ_{m,j}} &= \left( 0 + 0 + \dots + \frac{(\sigma_{m,j} - \tau_{m,j})}{(1 + Q_{m,j})} - \pi_m \right) \end{aligned} \quad (79)$$

Simplification finale de l'expression pour obtenir la dérivée générale :

$$\frac{dU_{CP_j}}{dQ_{ij}} = \sum_{i=1}^m \frac{(\sigma_{ij} - \tau_{ij})}{(1 + Q_{ij})} - \pi_i \quad (80)$$

### ii. Calcul de la dérivée de second ordre

$$\frac{d^2U_{CP_j}}{dQ_{ij}} = \left( \sum_{i=1}^m ((\sigma_{ij} - \tau_{ij}) * \log(1 - Q_{ij}) - \pi_i Q_{ij}) \right)'' \quad (81)$$

$$\frac{d^2 U_{CP_j}}{dQ_{ij}} = \left( \frac{dU_{CP_j}}{dQ_{ij}} \right)' = \left( \sum_{i=1}^m \frac{(\sigma_{ij} - \tau_{ij})}{(1 + Q_{ij})} - \pi_i \right)' \quad (82)$$

$$\frac{d^2 U_{CP_j}}{dQ_{ij}} = \sum_{i=1}^m -\frac{(\sigma_{ij} - \tau_{ij})}{(1 + Q_{ij})^2} \quad (83)$$

Par observation, il est clair que la valeur de deuxième dérivée est négative  $\left( -\frac{(\sigma_{ij} - \tau_{ij})}{(1 + Q_{ij})^2} < 0 \right)$  donc par suit la valeur de la somme reste  $< 0$ ), la chose qui détermine que la fonction d'utilité  $U_{CP_j}$  est concave par rapport à la quantité  $Q_{i,j}$  et que le système admit au moins une solution optimale. Ainsi, au moins une quantité maximale globale existe et elle peut être obtenue par la condition d'optimisation de la contrainte de Karush-Kuhn-Tucker (KKT condition). Par conséquent, la meilleure décision du suiveur  $j$  peut être facilement obtenue en mettant la première dérivée = 0 et résoudre l'équation correspondante :

$$\frac{dU_{CP_j}}{dQ_{i,j}} = \sum_{i=1}^m \left( \frac{(\sigma_{i,j} - \tau_{ij})}{(1 + Q_{i,j})} - \pi_i \right) = 0 \quad (84)$$

$$\frac{(\sigma_{i,j} - \tau_{ij})}{(1 + Q_{i,j})} - \pi_i = 0 \quad (85)$$

$$1 + Q_{i,j} = \frac{(\sigma_{i,j} - \tau_{ij})}{\pi_i} \quad (86)$$

$$Q_{i,j}^* = \frac{\sigma_{i,j} - \tau_{i,j} - \pi_i}{\pi_i} \quad (87)$$

tel que  $\pi_i \neq 0, 0 < \sigma_{ij} \leq 1$

Donc  $Q_{i,j}^* = \frac{\sigma_{i,j} - \tau_{i,j} - \pi_i}{\pi_i}$  c'est la quantité optimale des fichiers populaires d'un  $CP_j$  qui vont être stockés par le  $PTC_i$

## b) Stratégies optimales des leaders

Compte tenu des meilleures réponses stratégiques des CPs, et après la réception de toutes les décisions optimales de la part des CPs, chacun des leaders  $PTC_i$  va optimiser sa stratégie (le prix optimal de stockage) par résoudre le problème d'optimisation suivant :

$$\begin{aligned} \pi_i^* &= \operatorname{argmax}_{\pi_i} U_{PTC_i}(\pi_i) \\ \text{s.t. } \pi_i &> 0 \end{aligned} \quad (88)$$

On a

$$U_{PTC_i} = \sum_{j=1}^n \pi_i Q_{ij} - \left( \mu \sum_{j=1}^n P^t Q_{ij} + \eta_i \sum_{j=1}^n \sigma_{ij} \right) \quad (89)$$

Pour résoudre le problème d'optimisation des leaders dans (88), on commence par remplacer la quantité  $Q_{i,j}$  par  $Q_{i,j}^*$ , on obtient donc la formule suivante :

$$U_{PTC_i} = \sum_{j=1}^n \pi_i \frac{\sigma_{i,j} - \tau_{i,j} - \pi_i}{\pi_i} - \left( \mu \sum_{j=1}^n P^t \frac{\sigma_{i,j} - \tau_{i,j} - \pi_i}{\pi_i} + \eta_i \sum_{j=1}^n \sigma_{i,j} \right) \quad (90)$$

$$U_{PTC_i} = \sum_{j=1}^n \sigma_{i,j} - \tau_{i,j} - \pi_i - \left( \mu \sum_{j=1}^n P^t \frac{\sigma_{i,j} - \tau_{i,j} - \pi_i}{\pi_i} + \eta_i \sum_{j=1}^n \sigma_{i,j} \right) \quad (91)$$

Ensuite, nous calculons la première et la deuxième dérivée de la fonction d'utilité  $U_{PTC_i}$  pour prouver que la fonction est convexe et que l'équilibre du jeu existe.

#### i. Calcul de la dérivée de premier ordre

$$\frac{dU_{PTC_i}}{d\pi_i} = \left( \sum_{j=1}^n \sigma_{i,j} - \tau_{i,j} - \pi_i - \mu \sum_{j=1}^n P^t \frac{\sigma_{i,j} - \tau_{i,j} - \pi_i}{\pi_i} - \eta_i \sum_{j=1}^n \sigma_{i,j} \right)' \quad (92)$$

Application des formules différentielles :

$$\frac{dU_{PTC_i}}{d\pi_i} = \frac{d\left(\sum_{j=1}^n \sigma_{i,j} - \tau_{i,j} - \pi_i\right)}{d\pi_i} - \frac{d\left(\mu \sum_{j=1}^n P^t \frac{\sigma_{i,j} - \tau_{i,j} - \pi_i}{\pi_i}\right)}{d\pi_i} - \frac{d\left(\eta_i \sum_{j=1}^n \sigma_{i,j}\right)}{d\pi_i} \quad (93)$$

On obtient :

$$\frac{dU_{PTC_i}}{d\pi_i} = -1 + \mu P^t \times \sum_{j=1}^n \left( \frac{\sigma_{i,j} - \tau_{i,j}}{\pi_i^2} \right) \quad (94)$$

#### ii. Calcul de la dérivée de second ordre

$$\frac{d^2 U_{PTC}}{d\pi_i} = \left( \frac{dU_{PTC}}{d\pi_i} \right)' = \left( -1 + \mu P^t \times \sum_{j=1}^n \left( \frac{\sigma_{i,j} - \tau_{i,j}}{\pi_i^2} \right) \right)' \quad (95)$$

On obtient :

$$\frac{d^2 U_{PTC_i}}{d\pi_i} = -2\mu P^t \times \sum_{j=1}^n \left( \frac{\sigma_{i,j} - \tau_{i,j}}{\pi_i^3} \right) \quad (96)$$

Il est clair que la valeur de deuxième dérivée est négative (nous établissons que la valeur de la somme est positive  $\sum_{j=1}^n \left( \frac{\sigma_{i,j} - \tau_{i,j}}{\pi_i^3} \right) > 0$  du fait que les valeurs de  $\sigma_{i,j} - \tau_{i,j}$  et  $\pi_i$  sont toujours positives (par définition des concepts de base), il en va de même pour les constantes  $\mu, P^t$  sont strictement positives. Par conséquent, la dérivée prend le signe de multiplicateur (-2)), ce qui détermine que la fonction d'utilité  $U_{PTC_i}$  est concave par rapport au prix de stockage  $\pi_i$  et que le système admit au moins une solution optimale. Cette solution peut être obtenue par la condition d'optimisation de la contrainte de Karush-Kuhn-Tucker, en fixant la première dérivée = 0 et en résolvant l'équation correspondante :

$$\frac{dU_{PTC_i}}{d\pi_i} = -1 + \mu P^t \times \sum_{j=1}^n \left( \frac{\sigma_{i,j} - \tau_{i,j}}{(\pi_i)^2} \right) = 0 \quad (97)$$

$$\sum_{j=1}^n \left( \frac{\sigma_{i,j} - \tau_{i,j}}{(\pi_i)^2} \right) = \frac{1}{\mu P^t} \quad (98)$$

$$\frac{1}{(\pi_i)^2} \sum_{j=1}^n (\sigma_{i,j} - \tau_{i,j}) = \frac{1}{\mu P^t} \quad (99)$$

$$\pi_i^* = \pm \sqrt{\mu P^t \sum_{j=1}^n (\sigma_{i,j} - \tau_{i,j})} \quad (100)$$

Parmi les deux solutions, nous prenons la valeur positive de  $\pi_i^*$  car elle vérifie la contrainte (88), nous avons donc :

$$\pi_i^* = \sqrt{\mu P^t \sum_{j=1}^n (\sigma_{i,j} - \tau_{i,j})} \quad (101)$$

Dans ce qui suit, nous proposons un algorithme pour calculer la valeur du prix dans l'équilibre de Stackelberg pour chaque  $PTC_i \xrightarrow{i=1,m}$

### iii. Algorithme de calculer l'équilibre de Stackelberg

---

**Algorithme 2 :** Algorithme itératif pour trouver la stratégie optimale du leader

---

#### Initialisation

- Soit  $\mathcal{I}$  l'ensemble des  $PTC_i$  tel que  $i = \overline{1, m}$ .
- $\pi_i(t_0)$  : la valeur initiale du prix annoncé par le  $PTC_i$  au début de jeu, instant  $t = 0$ , calculer en utilisant l'équation (68).
- $U_{PTC_i}(t_0)$  : La valeur initiale de l'utilité de  $U_{PTC_i}$ , calculer en utilisant l'équation (71).
- Fixer la valeur de  $\xi$  (niveau de la précision) à une valeur initiale (e.g.,  $5 \times 10^{-7}$ )
- $\pi^{ES} = \{\emptyset\}$  représente le vecteur de rang  $m$  des prix optimaux pour les PTCs

#### Fin Initialisation

#### Début

1. **Pour** chaque  $PTC_i$  **faire**
2. Calculer  $\pi_i^*(t_1)$  à l'instant  $t_1 = t_0 + 1$ , en utilisant l'équation (101)
3. Calculer  $U_{PTC_i}(t_1)$  à l'instant  $t_1 = t_0 + 1$ , en utilisant l'équation (71), le prix  $\pi_i^*(t_1)$  calculé dans l'étape précédente 2 et la quantité optimal (87)
4. **Tanque** ( $U_{PTC_i}(t_1) - U_{PTC_i}(t_0) > \xi$ ) **faire**
  - a)  $t_0 \leftarrow t_1$
  - b)  $U_{PTC_i}(t_0) \leftarrow U_{PTC_i}(t_1)$
  - c) Calculer  $\pi_i^*(t_1)$  au nouvel instant  $t_1 = t_0 + 1$ , en utilisant l'équation (101)
  - d) Calculer la nouvelle valeur  $U_{PTC_i}(t_1)$  en utilisant l'équation (71), la quantité optimale (87) et le prix calculé dans l'étape {4.c}
5. **Fin Tanque**
6.  $\pi^{ES} = \pi^{ES} \cup \pi_i^*(t_1)$
7. **Fin Pour**

**Fin**

---

### 4.3 Conclusion

Dans ce chapitre, nous avons proposé deux mécanismes d'incitation à la mise en cache dans les réseaux d'IoV : le premier est considéré comme un marché monopoliste joué entre un PTC et plusieurs CPs, le deuxième est un marché compétitif avec plusieurs PTCs et plusieurs CPs. Pour les deux scénarios, chaque PTC vise à remplir au maximum ses caches embarqués sur les véhicules afin de maximiser son gain monétaire, tout en incitant les CPs à venir stocker leurs contenus populaires dans ces caches. D'autre part, chaque CP vise à améliorer la satisfaction de ses utilisateurs en améliorant leurs QoE.

Nous avons formulé la première politique d'incitation à l'aide de la théorie des contrats. La deuxième politique compétitive en tant qu'un jeu de type Stackelberg avec multiples leaders et multiples suiveurs. Nous avons aussi décrit les fonctions d'utilité de chaque acteur de mise en cache. Après, nous avons basé sur la condition d'optimisation contrainte de Karush-Kuhn-Tucker pour calculer les valeurs optimales des joueurs. Enfin, nous avons terminé chacun des deux propositions par un algorithme : le premier pour définir le contrat optimal, le deuxième a but de trouver l'équilibre de Stackelberg.

Dans le chapitre suivant, nous allons implémenter les deux stratégies d'incitation à la mise en cache afin d'étudier leurs performances respectives.