

Article

A Game Theoretic Approach for D2D Assisted Uncoded Caching in IoT Networks

Jiajie Ren ^{1,*}  and Chang Guo ² ¹ School of Artificial Intelligence and Information Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China² College of Information, Mechanical, and Electrical Engineering, Shanghai Normal University, Shanghai 201418, China; guochang@shnu.edu.cn

* Correspondence: happyren@zust.edu.cn

Abstract

Content caching and exchange through device-to-device (D2D) communications can offload data from the centralized base station and improve the quality of users' experience. However, existing studies often overlook the selfish nature of user equipment (UE) and the heterogeneity of content preferences, which limits their practical applicability. In this paper, we propose a novel incentive-driven uncoded caching framework modeled as a Stackelberg game between a base station (BS) and cache-enabled UEs. The BS acts as the leader by determining the unit incentive reward, while UEs jointly optimize their caching strategies as followers. The particular challenge in our formulation is that the uncoded caching decisions make the UEs' total utility maximization problem into a non-convex integer programming problem. To address this, we map the UEs' total utility maximization problem into a potential sub-game and design a potential game-based distributed caching (PGDC) algorithm that guarantees convergence to the optimal joint caching strategy. Building on this, we further develop a dynamic iterative algorithm to derive the Stackelberg equilibrium by jointly optimizing the BS's cost and the total utility of UEs. The simulation results confirm the existence of the Stackelberg Equilibrium and demonstrate that the proposed PGDC algorithm significantly outperforms benchmark caching schemes.

Keywords: D2D assisted IoT caching networks; incentive mechanism; Stackelberg game; potential game



Academic Editor: Jose I. Moreno Novella

Received: 19 August 2025

Revised: 8 September 2025

Accepted: 15 September 2025

Published: 18 September 2025

Citation: Ren, J.; Guo, C. A Game Theoretic Approach for D2D Assisted Uncoded Caching in IoT Networks. *Future Internet* **2025**, *17*, 423. <https://doi.org/10.3390/fi17090423>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the tremendous advances of Internet-of-Things (IoT) communication technologies and the ever-increasing use of smart IoT devices, wireless data traffic has exhibited explosive growth in the past few years, which puts a heavy burden on the existing IoT networks. To cope with this issue, academia and industry are beginning to explore effective approaches, such as, massive MIMO [1], coordinated multipoint transmission [2], and reconfigurable intelligent surfaces [3], to improve the IoT network transmission capacity. In addition, a lot of popular content in the IoT environment is frequently requested by various users, incurring repetitive transmissions and large-scale congestion during peak traffic time [4]. Device-to-device (D2D) communications with caching techniques can be considered as a promising solution [5] to relieve the IoT network traffic burden and improve users' experience (QoE).

With D2D-based caching, user equipment (UE) can cache contents with individual preferences and share content with their neighbor UEs to reduce the content retrieval

delay. D2D-based caching can also help mitigate the traffic load on the base station (BS) by enabling direct content sharing between UEs. UE's willingness to share content is a critical issue in D2D networks. This is because helping the BS with caching and sharing content will inevitably incur additional costs, such as content maintenance costs and transmission delay costs. Thus, it is necessary to develop an incentive mechanism to motivate UEs to participate in content sharing via D2D communications. Recent studies have leveraged game theory to investigate the incentive mechanism and caching optimization problem in the IoT networks [6–9]. In [6,7], the authors investigate the uncoded caching optimization problem, where the content is either fully cached or not cached at all. Thus, the caching decision is the integer variable. In order to derive the closed-form Stackelberg Equilibrium (SE), the utility functions in [6,7] are modeled as the quadratic function or the logarithm function. However, the modeled utility functions are highly dependent on preset parameters and may not accurately reflect users' true benefits. Hence, the question of how to solve the non-convex integer caching optimization in game-theoretic framework is intractable and imperative. Additionally, we find that the content caching and request probabilities in the game models of the existing works [8,9] are usually assumed to be homogeneous, i.e., all UEs have the same request probability for one particular content, which is not practical in the real world, where each UE normally has different interest and preference for network content.

In this paper, we consider a D2D-assisted IoT caching network including a base station and a set of cache-enabled UEs with the content request probability following a heterogeneous distribution. We formulate the economic interactions between BS and UEs as a one-leader-one-follower Stackelberg game in which the BS acts as the leader and all UEs jointly act as the follower. In the proposed game, the BS initially determines the unit incentive reward submitting to the cache-enabled UEs. Then, based on the given reward, UEs utilize the local cooperative caching mechanism to proactively cache appropriate content, with the aim of maximizing their total utility. Meanwhile, BS aims to keep the incentive reward as small as possible to minimize its own cost. In order to find the SE of the proposed Stackelberg game, we first formulate the total utility optimization problem (i.e., the follower's problem), which is constrained by the binary caching decision variables of all UEs. Thus, the follower's problem is a non-convex integer optimization problem and it is difficult to find the closed-form solution. Fortunately, we can map the total utility optimization into a sub-game and prove that this sub-game with UE's local cooperative utility function is an exact potential game, in which at least one Nash Equilibrium (NE) exists. Building on this, we propose a potential game-based distributed caching (PGDC) algorithm to acquire the optimal joint caching placement strategy of all UEs. Although the proposed PGDC algorithm adopts a Gibbs-like update rule for caching strategy selection, it is specifically tailored for the uncoded caching problem in D2D-assisted IoT networks. The novelty lies in the integration of context-specific utility function design, potential game mapping, and distributed learning to address a challenging non-convex integer optimization problem. Then, the achieved optimal joint caching strategy can be regarded as the follower's best response to the proposed Stackelberg game, we design a dynamic iterative algorithm based on the best response to solve the BS's cost optimization problem (i.e., the leader's problem) and identify the optimal incentive reward. Thus, the Stackelberg Equilibrium of the proposed game is derived.

The contributions of this paper are summarized as follows:

- Considering the content preferences and share willingness of UEs, we propose an uncoded caching incentive mechanism that motivates UEs to share cached contents, assisting the BS in alleviating content delivery burdens. The profit interactions between BS and UEs are modeled as a Stackelberg game to jointly optimize the cost of BS and UEs' total utility.

- In order to derive the SE of the proposed game, we insightfully formulate the non-convex total utility optimization problem for all UEs as a potential sub-game, and prove this sub-game involves at least one NE. We further demonstrate that the optimal joint caching strategy must be a NE of this sub-game. Subsequently, we design a PGDC algorithm to find the optimal solution to the sub-game for a given incentive reward, and theoretically analyze the convergence of the proposed PGDC algorithm.
- Leveraging the optimal joint caching placement strategy as the follower's best response to the Stackelberg game, we design a dynamic iterative algorithm to efficiently solve the BS's cost optimization problem. This algorithm further enables us to determine the SE of the proposed game, which allows BS to achieve its optimal cost and UEs to obtain their maximum total utility, respectively.
- Through extensive simulation experiments, we evaluate the convergence of the proposed PGDC algorithm and validate the existence of Stackelberg Equilibrium. We compare the performance of several benchmark caching strategies with our proposed caching solution. The simulation results demonstrate the significant superiority of the PGDC algorithm.

2. Related Work

Caching in IoT networks can significantly improve the network performance, which has attracted many researchers' attention to investigating the optimal caching policies. In [10], the authors studied the impact of bursty traffic and random availability of caching helpers on the weighted throughput and the average delay in caching helper networks by adopting the most popular caching (MPC) policy. However, MPC may not be the optimal policy in the heterogeneous content request probability model. Zhang et al. [11] proposed a cooperative caching architecture of the unmanned aerial vehicle and user terminals, and solved the caching placement by the many-to-many swap-matching algorithm. In [12], Cao et al. took account of the user's mobility when creating their caching strategy. They developed a mobility-aware joint routing and caching strategy and designed a greedy caching algorithm to minimize the cost of the content delivery. In general, a greedy caching policy cannot acquire the global optimal solution. In [13], the authors investigated the mobility-aware content caching and updating optimization problem based on a user-interest prediction model in the IoT networks.

However, the aforementioned research ignored the selfish nature and unwillingness to share exhibited by UEs in real life. Thus, it is essential to design an incentive mechanism that stimulates UEs to be willing to share content and that benefits the two-side profits of BS and UEs. In [5], the authors proposed a green incentive mechanism to encourage mobile D2D users to cache and share content with the aim of energy saving. In [14], the authors utilized the Nash bargaining solution approach to design a novel incentive mechanism in order to encourage users to cache and share content. In [15], the authors proposed a monetary incentive-based mechanism to boost user participation, thereby facilitating successful D2D communication and achieving the high cache hit probability. Furthermore, combined with incentive mechanism, game theory [16] plays a vital role in the mutually influential economic strategies between BS and UEs in IoT caching networks, which is also an useful mathematical model to design the incentive mechanism and analyze the bilateral conflicts of economic issues.

There are a large number of works focusing on the caching issues from the game theoretic perspective. Su et al. [17] investigated the interaction between UEs and social groups in mobile social caching networks using the Stackelberg game and obtained the optimal caching strategy by the backward induction method. A joint video rental pricing and caching optimization problem is studied in [18] by considering the non-cooperative

and cooperative caching of BSs under the framework of Stackelberg game. Jiang et al. [19] proposed a joint mobile edge caching and peer content-sharing framework, and derived the the game equilibrium under a generic usage-based pricing scheme. Moreover, we find that studies [17–19] all adopt the homogeneous content request probability, which is impractical in real life. In [20], Zheng et al. solved the large-scale uncoded caching problem in mobile edge networks using the game-based alternating direction method of multipliers. Jiang et al. [21] leveraged the crowdsourcing technique to study the edge caching system, where content providers shared a proportion of profit with UEs as an incentive reward and formulated a two-stage Stackelberg game to optimize the revenue of content providers and UEs. In [22], a blockchain incentive scheme was proposed and the Stackelberg game models using different pricing methods were formulated so as to divide uniform pricing and discriminatory pricing. In [23], in order to develop the multi-bitrate video caching strategy, the authors transformed the caching problem into the knapsack problem and applied the dominant strategy from game theory.

Unlike existing works, we propose an uncoded caching incentive mechanism for UEs with consideration of UEs' heterogeneous content preference and share willingness. We formulate the the economic interactions between BS and UEs as a Stackelberg game to jointly optimize the cost of BS and the total utility of UEs. Since the UEs' total utility optimization problem is a non-convex integer programming problem, we map this thorny problem into a potential sub-game, which is not investigated in most of the existing uncoded caching games.

3. System Models

3.1. Network Model

As shown in Figure 1, we consider a typical D2D-assisted IoT caching network comprising a base station (BS) and a set of cache-enabled user equipments (UEs). For clarity, let BS be indexed 0 and let $\mathcal{N} = \{1, 2, \dots, N\}$ denote the set of UEs. Each UE $n \in \mathcal{N}$ has limited caching capacity c . We assume that each UE is able to request and share content directly with its neighbor UEs via D2D links. Let $J_n \subseteq \mathcal{N}$ denote the set of neighbors of UE n , i.e.,

$$J_n = \{j \in \mathcal{N} | l_{n,j} \leq r_n\}, \quad (1)$$

where $l_{n,j}$ is the distance between UE j and UE n , r_n denotes the transmission radius of UE n .

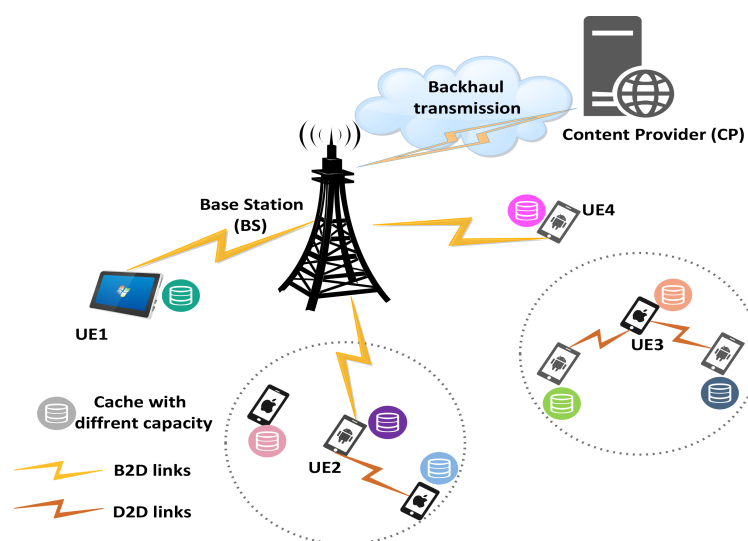


Figure 1. A typical D2D-assisted IoT caching network model.

3.2. Content Request and Cache Placement Model

There are F contents in the IoT network, which are indexed by the set $\mathcal{F} = \{1, 2, \dots, F\}$. We assume each type of content has the same normalized size and is denoted by s . A number of prior studies adopted the homogeneous content request probability [17–19], which means all users in the networks have the same content request probability. In fact, UEs have different preferences about all kinds of content. Hence, we adopt a heterogeneous content request probability. That is, the request probability for the same content differs among the various UEs. Let $[f]_n, f \in \mathcal{F}$ denote the rank of content f according to UE n 's preference. For example, if $[f]_n = 1$, that means content f is UE n 's favourite content. If $[f]_n = F$, that means content f is UE n 's least favourite content. We assume that each UE requests the content independently and let P_{nf} denote the request probability of content f by UE n . We consider the assumption from [24], the content request probability is modeled by the Zipf distribution (i.e., the content popularity distribution) (Since we consider individual preference, here, the content popularity distribution is only for each UE, which is the same for the individual content request distribution, and not for all UEs throughout the whole network.). The probability of requesting content f by UE n is expressed as

$$P_{nf} = \frac{[f]_n^{-\alpha}}{\sum_{f=1}^F 1/[f]_n^{\alpha}}, \forall n \in \mathcal{N}, \forall f \in \mathcal{F}, \quad (2)$$

where $\alpha > 0$ determines the skewness of the content popularity distribution for each UE. When Zipf exponent α is large, it indicates that the request of each UE focuses on the first few popular types of content, which means the content with a small ranked index has a large request probability. In addition, we assume that the content request popularity of each UE remains stable and unchanged during the long period (e.g., a week).

During the off-peak period, each UE decides whether to cache the entire content (i.e., uncoded caching) according to their preference. We define $x_{nf}, \forall n \in \mathcal{N}, \forall f \in \mathcal{F}$ as the binary decision variable to determine whether the content f is cached by UE n or not. If $x_{nf} = 1$, the content f is cached by UE n ; otherwise, $x_{nf} = 0$. Due to the limited caching storage capacity, the total number of contents cached by UE n should be satisfied by the following constraint:

$$\sum_{f=1}^F x_{nf} \leq c, \forall n \in \mathcal{N}. \quad (3)$$

3.3. Communication Model

In order to avoid transmission interference among UEs, we assume UEs can request and transmit content using orthogonal frequency division multiple access (OFDMA) [25]. The transmission rate of UE n from its neighbor j can be computed as

$$T_{n,j} = B_{n,j} \log\left(1 + \frac{P_j |h_{n,j}|^{\gamma}}{\sigma^2}\right), \quad (4)$$

where $B_{n,j}$ denotes the communication bandwidth between UE n and its neighbor j , P_j denotes the transmission power of UE j , $|h_{n,j}|$ denotes the channel gain between UE n and UE j , γ denotes the path loss exponent of the transmission links, and σ^2 denotes the power of the additive white Gaussian noise. Similarly, the transmission rate of UE n from BS can be computed as

$$T_{n,0} = B_{n,0} \log\left(1 + \frac{P_0 |h_{n,0}|^{\gamma}}{\sigma^2}\right), \quad (5)$$

where $B_{n,0}$ denotes the communication bandwidth between UE n and BS, P_0 denotes the transmission power of BS, and $|h_{n,0}|$ denotes the channel gain between UE n and BS.

The content transmission delay between UE n and j and between UE n and BS can be calculated by $D_{n,j} = s/T_{n,j}$ and $D_{n,0} = s/T_{n,0}$, respectively. The main notations and descriptions are listed in Table 1.

Table 1. Notations and descriptions.

Notation	Description
$\mathcal{N}$	the set of UEs
J_n	the set of neighbors of UE n
$\mathcal{F}$	the set of contents
$[f]_n$	the rank of content f according to UE n 's preference
P_{nf}	the request probability of content f of UE n
x_{nf}	caching decision of UE n
c	caching capacity
σ^2	noise power
$D_{n,j}$	transmission delay between UE n and j
$D_{n,0}$	transmission delay between UE n and BS
r	unit incentive reward
$[n]_j$	the delay rank of UE n to its neighbor UE j
$(k)_n$	index of the neighbor UE with the k th lowest delay to UE n
η_0	unit transmission delay cost between UEs
η_1	unit transmission delay cost between UE and BS
η_2	unit transmission energy cost between UE and BS
β	learning parameter

4. Stackelberg Game Formulation

In this section, we first design the utility function of UEs and the cost function of BS, respectively, based on the incentive mechanism. Due to the high cost of a single UE directly requesting content from BS, the proposed local cooperative caching mechanism among UEs enables the sharing of cached content between UEs and their neighbors. This not only provides the incentive reward for content sharers, but also reduces the cost of content requesters requesting content from the base station. Hence, the utility of UEs under the local cooperative caching mechanism is significantly higher than that under the non-cooperative caching mechanism. Then, we formulate the benefit interactions between UEs and BS as a single-leader-one-follower Stackelberg game to jointly optimize the incentive reward and caching placement strategy. In this game model, the leader (BS) has the priority to take action first (i.e., determining a unit reward), then the follower (all UEs) makes joint caching decisions to maximize their total utility according to the unit reward given by BS. Finally, the Stackelberg game can be decomposed into two sub-problems: the total utility maximization problem for all UEs and the cost minimization problem of BS.

4.1. Utility of the UE

The utility of UE n mainly consists of two parts: the incentive rewards paid by the base station and the delay costs incurred by requesting content. Firstly, we establish a revenue model based on the obtained incentives. Since J_n denotes the neighbour set of UE n , we sort $j \in J_n$ in an increasing order of the transmission delay to UE n and let $(k)_n$ denote the index of the neighbor UE with the k th lowest transmission delay to UE n . We have $(0)_n = n$, which means the 0th lowest transmission delay to UE n is itself. Moreover, we define $[n]_j$ as the transmission delay rank of UE n to its neighbor UE j . Note that $[j]_j = 0$; that is, UE j requesting the content from its own cache has the minimum delay. When UE n 's neighbor j submits a content request, UE j will first check its own local cache. If the requested content can not be found in UE j 's local cache, then UE j will submit a request to its neighbor UEs according to the sorted delay $(k)_j$. Let r denote the unit reward paid to

UE n by the BS when the UE n caches the requested content of its neighbor j and shares the cached content with neighbor requester j . Note that the incentive reward could be of any form, such as direct monetary incentives or virtual currency, and can be implemented by smart contracts. Thus, the total incentive revenue $R(n)$ of the UE n from the incentive reward can be formulated as

$$R(n) = \sum_f \sum_{j \in J_n} P_{jf} \prod_{k=0}^{[n]_j-1} (1 - x_{(k)jf}) x_{nf} r. \quad (6)$$

In Equation (6), if requester j 's neighbor UEs who have a lower transmission delay than UE n do not cache j 's requested content (i.e., $x_{(k)jf} = 0, \forall k < [n]_j$) and UE n caches the requested one (i.e., $x_{nf} = 1$), then we have $\prod_{k=0}^{[n]_j-1} (1 - x_{(k)jf}) x_{nf} = 1$. Only in this case can UE n obtain the incentive reward from BS. From Equation (6), we find that the revenue of UE n not only depends on its own caching decision x_{nf} but that it also depends on requester j 's neighbors' caching decisions.

Next, we consider the cost of UE n . We model the operation cost of UE n as the content transmission delay cost, which is similar to the assumption in [26]. UE n can acquire the requested content in three ways according to the local cooperative caching mechanism. Hence, the transmission delay has three different conditions and we define the transmission delay between UE n and its k th neighbor according to the sorted delay as $D_{n,(k)_n}$. Firstly, if the requested content in its local cache, then we assume that UE n satisfies its request with no delay cost, i.e., $D_{n,(0)_n} = 0$. Secondly, if its local cache does not have the requested content but its neighbors have a copy, then UE n will request the content from its neighbors based on the sorted delay. Lastly, neither UE n 's local cache nor all its neighbors cache the copy; UE n has to request the content from BS. The delay between UE n and BS is denoted by $D_{n,0}$. In particular, to emphasize that the incentive mechanism facilitates cooperation among UEs, we assume that the transmission distance between UEs is much shorter than that to the BS. Consequently, the delay incurred when UEs request contents from the BS via congested backhaul links is typically larger than the transmission delay between UEs. Thus, we state that $D_{n,(k)_n} < D_{n,0}, 0 \leq k \leq |J_n|$. Nevertheless, in advanced 5G/6G networks with extremely optimized base stations and high-capacity backhaul, the UE–BS transmission delay may be smaller than that of certain D2D links. Incorporating such cases into the model is an interesting development and will be discussed in our future work. The total cost $C(n)$ of the UE n can be formulated as

$$C(n) = \sum_f P_{nf} \left[\eta_0 \sum_{j=1}^{|J_n|} \prod_{k=0}^{j-1} (1 - x_{(k)nf}) x_{(j)nf} D_{n,(j)_n} + \eta_1 \prod_{k=0}^{|J_n|} (1 - x_{(k)nf}) D_{n,0} \right], \quad (7)$$

where η_0 is the unit transmission delay cost between UEs and η_1 is the unit transmission delay cost between UE and BS.

In Equation (7), if UE n finds the requested content f in its local cache, i.e., $x_{(0)nf} = 1$, then two items in the square brackets are both equal to zero, which means UE n can obtain the requested content from its local cache without any delay cost. Otherwise, the transmission delay cost of UE n becomes nonzero. If the requested content can be found in the j' th neighbor of UE n according to the sorted delay, i.e., $x_{(j')nf} = 1$ and $x_{(k)nf} = 0, \forall k < j'$, then we have $\prod_{k=0}^{j'-1} (1 - x_{(k)nf}) x_{(j')nf} = 1$, which means the first term is equal to 1 while the second term in the square brackets is equal to zero. If no requested content can be found in either UE n 's local cache or UE n 's neighbours, i.e., $x_{(k)nf} = 0, \forall 0 \leq k \leq |J_n|$, then

$\prod_{k=0}^{|J_n|} (1 - x_{(k)nf}) = 1$, which means the first term is equal to zero while the second term in the square brackets is equal to 1.

According to the above analysis, we consider that the caching action strategy set of UE n is denoted by $x_n = x_{n1} \otimes \dots \otimes x_{nf} \dots \otimes x_{nF}$, $x_{nf} \in \{0, 1\}$ for $f \in \mathcal{F}$, where symbol $\otimes$ is a Cartesian product. Let $x_{J_n}, \forall n \in \mathcal{N}$ represent the caching action strategy set of n 's neighbor UEs. Then, the utility function $U_n(x_n, x_{J_n})$ for UE n can be defined as the difference between the incentive revenue received from BS and the UE's content transmission delay cost. Hence, the utility $U_n(x_n, x_{J_n})$ of the UE n can be expressed as

$$\begin{aligned} U_n(x_n, x_{J_n}) &= R(n) - C(n) \\ &= \sum_f \sum_{j \in J_n} P_{jnf} r \prod_{k=0}^{[n]_j-1} (1 - x_{(k)jf}) x_{nf} - \sum_f P_{nf} \left[\eta_0 \sum_{j=1}^{|J_n|} \prod_{k=0}^{j-1} (1 - x_{(k)nf}) x_{(j)nf} D_{n,(j)n} + \right. \\ &\quad \left. \eta_1 \prod_{k=0}^{|J_n|} (1 - x_{(k)nf}) D_{n,0} \right]. \end{aligned} \quad (8)$$

4.2. Cost of the BS

We consider that the total cost of BS is composed of two parts: the incentive rewards paid to caching UEs and the content transmission energy cost. In order to reduce the transmission burden of BS and lower the delay of content requests at UEs, BS offers incentive rewards to caching UEs to promote content transmission and sharing. Let r denote the unit incentive reward paid to UEs who have shared cached contents to their neighbors. Due to the limited caching capacity of UEs, if the requested content cannot be found in the requester UE n 's own cache or its neighbors' caches, the requester UE n has to request the content from the BS, which incurs BS transmission energy cost. We assume that the transmission energy between BS and the requester UE n is $E_{n,0} = P_0 \cdot D_{n,0}$. Therefore, the total cost C_{BS} of BS can be formulated as

$$C_{BS} = \sum_n \sum_f (P_{nf} x_{(0)nf} r \sum_{j \in J_n} P_{jnf} \prod_{k=0}^{[n]_j-1} (1 - x_{(k)jf})) + \eta_2 \sum_n \sum_f P_{nf} \prod_{k=0}^{|J_n|} (1 - x_{(k)nf}) E_{n,0}, \quad (9)$$

where η_2 is the unit transmission energy cost between UE and BS. In Equation (9), if the requested content f is cached by UE n , i.e., $x_{(0)nf} = 1$, then we can easily find that the second term is equal to zero. In this case, when UE n 's neighbor j requests a desired content f according to the sorted delay $(k)_j$, we have two sub-cases. One is $\prod_{k=0}^{[n]_j-1} (1 - x_{(k)jf}) = 1$, which means UE j 's first $[n]_j - 1$ neighbors all fail to cache the content f and UE n has the copy. Then, UE n will deliver the content to UE j and UE n will receive the incentive reward paid by BS, which incurs the incentive cost of BS. The other sub-case involves the existence of $k' \leq [n]_j - 1$, such that $x_{(k')jf} = 1$, which means one of UE j 's first $[n]_j - 1$ neighbors has cached the content f and UE j can obtain the requested content from UE k' . The BS neither pays the incentive reward to the UE n nor incurs the transmission energy cost. If UE n does not cache the content f , i.e., $x_{(0)nf} = 0$, UE n wants to request the content f . In this case, we also have two sub-cases. The first sub-case is that one of UE n 's neighbors has cached the copy $x_{(k)nf} = 1, \exists 1 \leq k \leq |J_n|$, and the BS neither pays the incentive reward to the UE n nor incurs the transmission energy cost. The second sub-case is that no requested content can be found from UE n 's neighbors, i.e., $x_{(k)nf} = 0, \forall 1 \leq k \leq |J_n|$; thus, UE n can only acquire the requested content from BS. Thus, the first term in Equation (9) equals zero, and BS will generate the content transmission energy cost.

4.3. Stackelberg Game

In the proposed Stackelberg game, BS first determines the unit reward as the incentive for UEs. If the announced reward by the BS is higher, UEs are quite willing to cache the requested content of their neighbors and share the content via D2D communications. However, BS will accordingly incur a higher incentive reward cost. If the announced reward is lower, UEs may prefer caching the content according to their own preferences and may not be willing to share the content, which will incur a higher transmission energy cost for BS. Hence, BS needs to carefully determine the optimal unit reward to minimize its own cost. The cost optimization problem of BS can be formulated as the following Problem 1.

Problem 1.

$$\min_r C_{BS} \quad (10)$$

$$s.t. \quad r \geq 0, \quad (11)$$

$$x_{nf} \in \{0, 1\}, \quad \forall n \in \mathcal{N}, \forall f \in \mathcal{F}, \quad (12)$$

where C_{BS} is the cost function of BS given in Equation (9), r is the unit incentive reward determined by BS, and $x_{nf} \in \{0, 1\}$ is the caching decision of UE n , $\forall n \in \mathcal{N}$.

Given the reward paid by BS, all UEs try to find the optimal joint caching strategy to maximize the total utility according to the local cooperative caching mechanism. Therefore, the total utility optimization problem can be formulated as the following Problem 2.

Problem 2.

$$\max_{x_n} U_{UE} = \sum_{n=1}^N U_n(x_n, x_{J_n}) \quad (13)$$

$$s.t. \quad \sum_{f=1}^F x_{nf} \leq c, \quad \forall n \in \mathcal{N}, \quad (14)$$

$$x_{nf} \in \{0, 1\}, \quad \forall n \in \mathcal{N}, \forall f \in \mathcal{F}, \quad (15)$$

where U_{UE} is the total utility of all UEs and $U_n(x_n, x_{J_n})$ is the utility function of the UE n given in Equation (8). The objective of Equation (13) is subject to the maximum caching storage capacity constraint and the caching decision of UE n , $\forall n \in \mathcal{N}$.

Problems 1 and 2 formulate the Stackelberg game between BS and all UEs. Each player in the game is a selfish and rational individual and tries to achieve the maximum utility/the minimum cost. Therefore, the purpose of the Stackelberg game is to find the Stackelberg Equilibrium (SE), from which neither the BS nor the UEs have the incentive to deviate. Let $\mathbf{X} = x_1 \otimes x_2 \otimes \dots \otimes x_N$ denote the joint caching strategy of all UEs. We have the definition of the SE for the proposed Stackelberg game in the following.

Definition 1. Let r be the feasible solution to the cost optimization problem of BS and let $\mathbf{X}$ denote a feasible joint caching strategy in response to the total utility optimization problem of UEs. Then, $(r_{SE}, \mathbf{X}_{SE})$ is a Stackelberg Equilibrium (SE) for the proposed Stackelberg game if any $(r, \mathbf{X})$ in the feasible region satisfies the following conditions:

$$C_{BS}(r_{SE}, \mathbf{X}_{SE}) \leq C_{BS}(r, \mathbf{X}_{SE}), \quad (16)$$

$$U_{UE}(r_{SE}, \mathbf{X}_{SE}) \geq U_{UE}(r_{SE}, \mathbf{X}). \quad (17)$$

5. Stackelberg Game Solution

In this section, we derive the SE of the proposed Stackelberg game by using the backward induction method [27]. That is, to find an SE of the Stackelberg game, it is first necessary to determine the optimal joint caching strategy of UEs in Problem 2, which can be regarded as the best response for a given incentive reward r . Then, the optimal incentive reward can be acquired by solving the cost optimization problem of BS according to the obtained optimal joint caching strategy. However, in the UEs' total utility maximization problem, the caching decision $x_{nf}, \forall n \in \mathcal{N}, \forall f \in \mathcal{F}$ is a binary variable, which incurs the UEs' total utility sub-problem; this is a non-convex integer programming problem. Hence, it is difficult to obtain the optimal joint caching strategy in the closed form. Using a method different from that used to solve the analytical solution of the best response in the existing game models, we formulate the total utility maximization problem of UEs as a sub-game and prove that the sub-game is an exact potential game. Then, we develop a potential game-based distributed caching algorithm to obtain the optimal joint caching strategy. In the BS's cost-optimal sub-problem, we design a dynamic iterative algorithm to find the optimal incentive reward based on the optimal joint caching strategy.

5.1. UEs' Potential Sub-Game

In this subsection, we first prove that the sub-game of all UEs is an exact potential game utilizing local caching information exchange among UEs and that it possesses at least Nash Equilibrium (NE). However, in many cases, NE is not necessarily the optimal solution (i.e., the optimal joint caching strategy). Similarly, the optimal solution may not be NE. Here, we will prove that the total utility function of all UEs satisfies the monotone and submodular. Then, we will theoretically derive the bound of the total utility under the condition of the NE according to the submodular property. Moreover, we can prove that the optimal solution of the total utility optimization problem must be a NE.

According to the cooperative caching mechanism, if UE n changes its current caching strategy, then it will not only affect its own utility but also its neighbors' utility. Thus, we define a local cooperative utility function $U_n^{lc}(x_n, x_{J_n})$ of UE n as follows:

$$U_n^{lc}(x_n, x_{J_n}) = U_n(x_n, x_{J_n}) + \sum_{j \in J_n} U_j(x_j, x_{J_j}). \quad (18)$$

In Equation (18), the local cooperative utility is composed of two parts. The first term is UE n 's own utility and the second part is the sum of neighbors' utilities of UE n .

From Equation (18), it can be seen that the local cooperative utility improvement problem depends on the caching decision of UE n and its neighbor UEs. Thus, we consider the caching decision as the action strategy in a game and formulate the local cooperative utility improvement problem as a caching action game $\mathcal{G}_c = [\mathcal{N}, \{x_n\}_{n \in \mathcal{N}}, \{U_n^{lc}\}_{n \in \mathcal{N}}]$, where $\mathcal{N} = \{1, 2, \dots, N\}$ is the set of players (UEs), $\{x_n\}$ is the caching action strategy of player n , and $\{U_n^{lc}\}$ is the local cooperative utility function of player n . Below, we provide proof that $\mathcal{G}_c$ is an exact potential game.

Definition 2 ([28]). A caching action game $\mathcal{G}_c$ is called an exact potential game if there exists a potential function Φ such that for every $n \in \mathcal{N}$, the following holds:

$$\Phi(x'_n, \mathbf{X}_{-n}) - \Phi(x_n, \mathbf{X}_{-n}) = U_n^{lc}(x'_n, x_{J_n}) - U_n^{lc}(x_n, x_{J_n}), \quad (19)$$

where $\mathbf{X}_{-n} = x_1 \dots \otimes x_{n-1} \otimes x_{n+1} \dots \otimes x_N$ denotes the caching action strategy of all the players except player n .

Theorem 1. $\mathcal{G}_c$ is an exact potential game and has at least a Nash Equilibrium (NE).

Proof. First, we consider the total utility U_{UE} of all UEs as the potential function; that is,

$$\Phi(x_n, \mathbf{X}_{-n}) = U_{UE} = \sum_{n=1}^N U_n(x_n, x_{J_n}). \quad (20)$$

Suppose that a randomly selected UE n changes its caching strategy x_n into x'_n , while other UEs' caching action strategies remain unchanged. Then, the change in the local cooperative utility of UE n is given by

$$U_n^{lc}(x'_n, x_{J_n}) - U_n^{lc}(x_n, x_{J_n}) = U_n(x'_n, x_{J_n}) - U_n(x_n, x_{J_n}) + \sum_{j \in J_n} (U_j(x_j, x'_{j_j}) - U_j(x_j, x_{j_j})), \quad (21)$$

where $U_j(x_j, x'_{j_j})$ denotes UE j 's new utility function after its neighbor UE n changes the caching strategy. On the other hand, the change in the potential function due to the caching strategy change of UE n is

$$\begin{aligned} \Phi(x'_n, \mathbf{X}_{-n}) - \Phi(x_n, \mathbf{X}_{-n}) &= U_n(x'_n, x_{J_n}) - U_n(x_n, x_{J_n}) + \sum_{j \in J_n} (U_j(x_j, x'_{j_j}) - U_j(x_j, x_{j_j})) \\ &\quad + \sum_{i \in \mathcal{N} \setminus J_n, i \neq n} (U_i(x_i, x'_{i_i}) - U_i(x_i, x_{i_i})), \end{aligned} \quad (22)$$

where $\mathcal{N} \setminus J_n$ means UE n 's neighbor set J_n is not included in set $\mathcal{N}$. Since UE n 's caching decision only affects its neighbor UEs, we have

$$U_i(x_i, x'_{i_i}) - U_i(x_i, x_{i_i}) = 0, \forall i \in \mathcal{N} \setminus J_n, i \neq n. \quad (23)$$

Thus, we have

$$\Phi(x'_n, \mathbf{X}_{-n}) - \Phi(x_n, \mathbf{X}_{-n}) = U_n^{lc}(x'_n, x_{J_n}) - U_n^{lc}(x_n, x_{J_n}). \quad (24)$$

From Equation (24), we find that the change in UE n 's local cooperative caching utility caused by UE n 's unilateral caching action strategy change is equal to the change of the potential function; that is, the change in the total utility U_{UE} of all UEs. Hence, the caching action game $\mathcal{G}_c$ is an exact potential game. According to [28], it has been proved that each potential game has at least a Nash Equilibrium. $\square$

From Theorem 1, we know that $\mathcal{G}_c$ has at least one Nash Equilibrium. Next, we take advantage of submodular property to theoretically prove that the total utility under the NE condition can achieve at least $\frac{1}{1+\delta}$ optimal, where $0 \leq \delta \leq 1$ is a constant. Meanwhile, we can prove that the optimal caching strategy must be a NE. In the subsequent analysis, we first provide the definition of the submodular function and then prove that the total utility function of all UEs is submodular. Then, we prove the bound for the total utility U_{UE} of all UEs under the Nash Equilibrium condition.

Definition 3 (Submodular functions [29]). Let Q be a finite ground set, a set of function $U: 2^Q \rightarrow \mathbb{R}$ is submodular if sets X, Y satisfy $X \subseteq Y \subseteq Q$, $f \in Q - Y$ then

$$U(X \cup \{f\}) - U(X) \geq U(Y \cup \{f\}) - U(Y). \quad (25)$$

The submodular function has the similar property with the diminishing marginal returns, that is, the marginal gain of adding an element to the larger set will decrease.

Define a finite ground set $Q = \{Q_{nf} | n \in \mathcal{N}, f \in \mathcal{F}\}$, where Q_{nf} denotes that the content f may be placed in the cache of UE n . The ground set can be divided into N disjoint subsets $Q_1, Q_2, \dots, Q_N$, where $Q_n = \{Q_{n1}, \dots, Q_{nF}\}$ denotes the all content placement configurations in the UE n . The joint caching strategy $\mathbf{X}$ is a subset of Q such that $Q_{nf} \in \mathbf{X}$ if and only if $Q_{nf} = 1$. Thus, for a feasible joint caching strategy $\mathbf{X}$, the caching capacity constraint of UE n can be defined as $|\mathbf{X} \cap Q_n| \leq c, \forall n \in \mathcal{N}$.

Lemma 1. The total utility U_{UE} given in Equation (13) is monotone and submodular.

Proof. We consider two caching placement sets X and Y which satisfy that $X \subset Y \subset Q$. For any randomly selected UE n , let x_n and x_{j_n} denote UE n 's caching placement strategy and the caching placement strategies of UE n 's neighbors, respectively. We assume that $(x_n \cup x_{j_n}) \subset X \subset Y \subset Q$. Considering adding one element $Q_{af'} \in Q - Y, a \in \mathcal{N}$ into caching placement sets X and Y , we have the following two cases:

Case 1: UE n chooses to add $Q_{nf'}$ to its own caching. For the convenience of analysis, we assume UE n acquires $Q_{nf'}$ from its i th neighbor $(i)_n$. (If we consider that UE n acquires $Q_{nf'}$ from BS, we have similar results.) That is, $Q_{nf'} = x_{nf'} = 1$. Then, according to the caching placement X , the gain of UE n 's utility is

$$U_n(X \cup Q_{nf'}) - U_n(X) = \sum_{j \in J_n} P_{jf'} \prod_{k=0}^{[n]_j-1} (1 - x_{(k)jf'})r + P_{nf'} D_{n,(i)_n} \eta_0 > 0. \quad (26)$$

According to the caching placement Y , the gain of UE n 's utility is

$$U_n(Y \cup Q_{nf'}) - U_n(Y) = \sum_{j \in J_n} P_{jf'} \prod_{k=0}^{[n]_j-1} (1 - x_{(k)jf'})r + P_{nf'} D_{n,(i)_n} \eta_0. \quad (27)$$

From above analysis, we find that $U_n(X \cup \{Q_{nf'}\}) - U_n(X) = U_n(Y \cup \{Q_{nf'}\}) - U_n(Y)$, which satisfies the property of submodular function. And it can be seen that the utility gain of adding an element in this case is always monotone increase.

Case 2: UE n 's j th neighbor chooses to add f' to its own caching ($Q_{(j)nf'} = 1$). In this case, we have following two sub-cases:

2-1: Based on the caching placement Y , UE n can acquire the content f' from its neighbor $(i)_n$ with $i < j$, i.e., $Q_{(i)nf'} \in Y$. Then, the gain of UE n 's utility is $U_n(Y \cup Q_{(j)nf'}) - U_n(Y) = 0$. Based on the caching placement X , UE n can obtain the content f' from its neighbor $(i')_n$ with $i' \geq i$. If $i' < j$, then the gain of UE n 's utility is zero. Otherwise $i' > j$, the gain of UE n 's utility is $U_n(X \cup Q_{(j)nf'}) - U_n(X) = \eta_0 P_{nf'} (D_{n,(i')_n} - D_{n,(j)_n}) > 0$. In this sub-case, we have $U_n(X \cup Q_{(j)nf'}) - U_n(X) \geq U_n(Y \cup Q_{(j)nf'}) - U_n(Y)$.

2-2: Based on the caching placement Y , UE n can obtain the content f' from its neighbor $(i)_n$ with $i > j$, and the gain of UE n 's utility is $U_n(Y \cup Q_{(j)nf'}) - U_n(Y) = \eta_0 P_{nf'} (D_{n,(i)_n} - D_{n,(j)_n})$. Based on the caching placement X , UE n can obtain the content f' from its neighbor $(i')_n$ with $i' \geq i$. The gain of UE n 's utility is $U_n(X \cup Q_{(j)nf'}) - U_n(X) = \eta_0 P_{nf'} (D_{n,(i')_n} - D_{n,(j)_n})$. Then, we have

$$U_n(X \cup Q_{(j)nf'}) - U_n(X) - (U_n(Y \cup Q_{(j)nf'}) - U_n(Y)) = \eta_0 P_{nf'} (D_{n,(i')_n} - D_{n,(i)_n}) \geq 0. \quad (28)$$

In this case, we have $U_n(X \cup \{Q_{(j)nf'}\}) - U_n(X) \geq U_n(Y \cup \{Q_{(j)nf'}\}) - U_n(Y)$, which satisfies the property of submodular function.

Hence, the utility function $U_n(x_n, x_{j_n})$ of UE n is monotone and submodular. The submodular function is closed, i.e., the submodular function remains a submodular function

after a nonnegative linear combination [30]. Hence, the total utility U_{UE} of all UEs is also a submodular function. $\square$

From Lemma 1, we conclude that the total utility U_{UE} is a submodular function. We have the following theorem to theoretically prove that the total utility U_{UE} under the NE condition is at least $1/(1+\delta)$, $0 \leq \delta \leq 1$ of the optimal total utility (lower bound) and when the optimal caching strategy is the Nash equilibrium, we can acquire the optimal total utility (upper bound).

Theorem 2. Let $\mathbf{X}_{opt}$ and $\mathbf{X}_{NE}$ denote the optimal joint caching strategy and the Nash equilibrium joint caching strategy, respectively. Then, there exists $0 \leq \delta \leq 1$ such that the total utility U_{UE} of all UEs under the Nash equilibrium satisfies

$$\frac{1}{1+\delta} U_{UE}(\mathbf{X}_{opt}) \leq U_{UE}(\mathbf{X}_{NE}) \leq U_{UE}(\mathbf{X}_{opt}), \quad (29)$$

where $U_{UE}(\mathbf{X}_{opt})$ is optimal total utility for all UEs.

Proof. From Lemma 1, we know that the total utility U_{UE} of all UEs is a monotone increasing submodular function. Then, according to [31], we have the lower bound of total utility under the condition of NE. For any $0 \leq \delta \leq 1$, we have

$$U_{NE}(\mathbf{X}_{NE}) \geq \frac{1}{1+\delta} U_{UE}(\mathbf{X}_{opt}). \quad (30)$$

Next, we prove the upper bound of total utility under the condition of NE. We assume that $U_{UE}(\mathbf{X}_{opt})$ is the global optimal total utility, i.e., $U_{UE}(\mathbf{X}_{opt}) \geq U_{UE}(\mathbf{X})$, $\mathbf{X}$ is any feasible joint caching strategy, and assume the optimal caching strategy $\mathbf{X}_{opt} = \{x_1, x_2, \dots, x_N\}$ is not the NE. Then, there must exist one UE n that can increase its local cooperative utility by changing its caching strategy x_n to x'_n , i.e., $U_n^c(x'_n, \mathbf{x}_{-n}) > U_n^c(x_n, \mathbf{x}_{-n})$. According to the definition of the potential game, we have

$$\Phi(x'_n, \mathbf{x}_{-n}) > \Phi(x_n, \mathbf{x}_{-n}). \quad (31)$$

Since $\Phi(x_n, \mathbf{x}_{-n}) = U_{UE}(\mathbf{X}_{opt})$, that is contradicted by the assumption that $U_{UE}(\mathbf{X}_{opt})$ is the global optimal total utility. Hence, the optimal joint caching strategy must be the NE and $U_{UE}(\mathbf{X}_{opt}) \geq U_{UE}(\mathbf{X}_{NE})$. $\square$

5.2. PGDC Algorithm

In this subsection, we develop a potential game-based distributed caching (PGDC) algorithm to directly maximize the local cooperative utility and further optimize the total utility of all UEs. The proposed algorithm is a distributed caching scheme that only requires local communications between UEs without BS control. According to Theorem 1, when the local cooperative utility reaches the maximum, the total utility of all UEs is also maximized. Meanwhile, we can acquire the optimal joint caching strategy. From Theorem 2, we know that the obtained optimal joint caching strategy must be a NE of the sub-game. Therefore, in the proposed Stackelberg game, we can first solve the follower's total utility optimization problem based on Algorithm 1 and obtain the optimal joint caching strategy, which can be regarded as the best response for the given incentive reward r . Then, we can solve the BS's cost optimization problem based on the best response in the next subsection.

In order to solve the UEs sub-game, we develop the potential game-based distributed caching algorithm as shown in Algorithm 1. In this algorithm, we assume that each UE can communicate and exchange caching information with its neighbors. The main idea

of the proposed algorithm can be described as follows: First, we initialize the caching placement strategy randomly for each UE. In each iteration, a player (UE) is randomly selected to explore its new caching strategy (note that the content in the explored caching strategy has not really been cached yet). Then, the selected UE exchanges the explored caching information with its neighbors and calculates its new local cooperative utility. Furthermore, the probability that the selected UE updates its new caching strategy follows the Gibbs distribution [32], which is coupled with the calculated local cooperative utility of the selected UE as shown in Equation (18). When the local cooperative utility of the new explored caching strategy is larger than that of the current caching strategy, the Gibbs distribution will guarantee that the selected UE chooses the new explored caching strategy with high probability and, accordingly, the local cooperative utility of the selected UE will be improved. The proposed algorithm will end when the maximum iteration is reached or we achieve the maximum local cooperative utility.

Algorithm 1 Potential game-based distributed caching algorithm (PGDC)

Initialization: At time $t = 0$, each UE n , $\forall n \in \mathcal{N}$ caches c contents randomly. Set maximum iteration $t_{\max}$.

Loop for $t = 1, 2, 3, \dots, t_{\max}$

- 1: **Selecting UE:** UE n is selected randomly with equal probability. Then, each UE n calculates the local cooperative utility $U_n^{lc}(x_n, x_{J_n})$ by exchanging its current caching information with its neighbors.
- 2: **Exploring new caching strategy:** Each selected UE n chooses a trial caching strategy x'_n independently, i.e., UE n explores c new content randomly and calculates its new local cooperative utility by exchanging its new caching information with its neighbors, denoted as $U_n^{lc}(x'_n, x_{J_n})$.
- 3: **Updating caching probability:** The selected UE n chooses the trial content caching strategy in the next iteration or retains the current content caching strategy according to the following probability:

$$\begin{aligned} Pr[x_n(t+1) = x'_n] &= \frac{\exp(\beta U_n^{lc}(x'_n, x_{J_n}))}{\varphi}, \\ Pr[x_n(t+1) = x_n] &= \frac{\exp(\beta U_n^{lc}(x_n, x_{J_n}))}{\varphi}, \end{aligned} \quad (32)$$

$\varphi = \exp(\beta U_n^{lc}(x_n, x_{J_n})) + \exp(\beta U_n^{lc}(x'_n, x_{J_n}))$ and β is a learning parameter. All other UEs keep their caching strategies unchanged, i.e., $x_{-n}(t+1) = x_{-n}(t)$.

- 4: $t = t + 1$, if $t < t_{\max}$, then return to step 1, else return the optimal joint caching strategy.

End Loop

In the following, we will prove the proposed algorithm is convergent and can reach stationary distribution by leveraging the balanced equation.

Theorem 3. In the potential game-based distributed caching algorithm, the unique stationary distribution $\pi(X)$ of any joint caching strategy $X \in \mathbf{X}$ is given as follows:

$$\pi(X) = \frac{\exp(\beta \Phi(X))}{\sum_{X \in \mathbf{X}} \exp(\beta \Phi(X))}, \quad (33)$$

where $\mathbf{X}$ represents all the possible feasible joint caching strategies and $\Phi(\cdot)$ is the potential function given by (20).

Proof. According to [33], let $X(t) = \{x_1(t), \dots, x_N(t)\}$ denote the joint caching strategy of all UEs at the t -th iteration. According to the Algorithm 1, the caching state $X(t)$ is determined by state $X(t-1)$, $X(t)$ is a discrete-time Markov process. Moreover, the caching

states of all UEs is limited and any two of the caching states can be arbitrarily transformed. It is easily concluded that $X(t)$ is irreducible and aperiodic, which means $X(t)$ has a unique stationary distribution. Thus, $X(t)$ should satisfy the following balanced equation:

$$\pi(X)P_{XY} = \pi(Y)P_{YX}, \quad (34)$$

where $X, Y \in \mathbf{X}$ are two different joint caching strategies and P_{XY} denotes the transition probability from caching state X to state Y .

From Algorithm 1, the probability that UE n selected to explore the caching strategy is $1/N$ and the probability that the caching strategy x_n of UE n changing to x'_n is $\frac{\exp(\beta U_n^{lc}(x'_n, x_{J_n}))}{\varphi}$. Hence, at the t -th iteration, we have

$$\pi(X)P_{XY} = \frac{\exp(\beta\Phi(X))}{\sum_{\chi \in \mathbf{X}} \exp(\beta\Phi(\chi))} \times \left(\frac{1}{N}\right) \times \left(\frac{\exp(\beta U_n^{lc}(x'_n, x_{J_n}))}{\varphi}\right). \quad (35)$$

Let τ be denoted as

$$\tau = \frac{1}{N\varphi[\sum_{\chi \in \mathbf{X}} \exp(\beta\Phi(\chi))]} \quad (36)$$

Then, we have

$$\pi(X)P_{XY} = \tau \exp(\beta\Phi(X) + \beta U_n^{lc}(x'_n, x_{J_n})). \quad (37)$$

Based on the symmetry property, we can derive the following equation,

$$\pi(Y)P_{YX} = \tau \exp(\beta\Phi(Y) + \beta U_n^{lc}(x_n, x_{J_n})). \quad (38)$$

According to the balanced equation and the definition of potential game that $\Phi(X) - \Phi(Y) = U_n^{lc}(x_n, x_{J_n}) - U_n^{lc}(x'_n, x_{J_n})$, we have

$$\sum_{X \in \mathbf{X}} \pi(X)P_{XY} = \sum_{X \in \mathbf{X}} \pi(Y)P_{YX} = \pi(Y) \sum_{X \in \mathbf{X}} P_{YX} = \pi(Y). \quad (39)$$

Therefore, we prove the potential game-based distributed caching algorithm has a unique stationary distribution. $\square$

The proposed PGDC algorithm can be readily applied in D2D-assisted IoT networks for practical deployment. For instance, in a smart community environment, UEs (e.g., smartphones, mobile terminals) can leverage the PGDC algorithm to dynamically update their cached content based on both their own demand and their neighbors' preferences. This not only reduces latency and offloads the central base station but also aligns with the incentive mechanisms provided by the base station. Since the algorithm only requires local information exchange between neighbors and does not rely on global network knowledge, it is suitable for large-scale privacy-sensitive applications. Therefore, PGDC provides a practical solution that bridges theoretical game-theoretic modeling with implementable caching strategies in real IoT networks.

5.3. BS's Cost Optimization Sub-Problem

In this subsection, we design a dynamic iterative algorithm according to the optimal joint caching strategy to solve the BS's cost optimization problem and seek the optimal incentive reward r_{opt} . After obtaining the r_{opt} and the corresponding $\mathbf{X}_{opt}$, we then acquire the SE of the proposed Stackelberg game. The designed dynamic iterative algorithm includes the interactions between BS and UEs, as shown in Algorithm 2. In Algorithm 2, BS as the leader first sets the reward $r = 0$ and starts the Stackelberg game. The follower

(all UEs) executes the PGDC algorithm to obtain the corresponding optimal joint caching strategy for a given incentive reward r , and computes the corresponding optimal total utility U_{UE} according to Equation (13). Then, BS calculates its own cost based on the obtained optimal joint caching strategy and increases the incentive reward r by a very small increment $\Delta_r > 0$. The above interactions are repeated iteratively.

Algorithm 2 Dynamic iterative algorithm

```

1: function BS-game ( $\mathbf{X}, r$ )
2:   BS: set  $r = 0$ 
3:   UEs-game ( $\mathbf{X}, r$ )
4:   calculate BS cost according to Equation (9)
5:    $r \leftarrow r + \Delta_r$ 
6: until  $C_{BS}$  reaches the minimum costs, the corresponding incentive reward denoted by
    $r_{opt}$  and the corresponding joint caching strategy denoted by  $\mathbf{X}_{opt}$ 
7: return the optimal reward  $r_{opt}$  and the optimal joint caching strategy  $\mathbf{X}_{opt}$ 
8: End function
9: function UEs-game ( $\mathbf{X}, r$ )
10:  repeat PGDC algorithm until it reaches the maximum iteration or achieves the
    maximum local cooperative utility
11: return the optimal joint caching strategy  $\mathbf{X}_r$  and the maximum total utility  $U_{UE}$  accord-
    ing to Equation (13) for a given  $r$ 
12: End function
  
```

Convergence analysis: At the beginning of iterations, the cost of BS will decrease as the reward r increases. This is because UEs are willing to cache and share content based on their neighbors' preferences when they receive the incentive reward. However, the decreasing trend of the cost will not last very long. When the cost of the BS minimizes at a certain incentive reward, it will slowly increase after this certain point due to the limited caching capacity of UEs. That is, UEs' requests that cannot be satisfied by their neighbors will be submitted to the BS, which increases the BS transmission energy cost. The certain incentive reward makes the cost of the BS reach the minimum value denoted by r_{opt} and the corresponding optimal joint caching strategy is denoted by $\mathbf{X}_{opt}$, which contains the SE point of the Stackelberg game. That is, $(r_{opt}, \mathbf{X}_{opt})$ is the SE of the proposed Stackelberg game.

5.4. Complexity Analysis

Next, we analyze the complexity of the PGDC algorithm and the dynamic iterative algorithm. In each iteration of the PGDC algorithm, the computation complexity of utility for the selected UE is determined by the number of neighbors and the caching capacity. The upper bounds of the number of neighbors and the caching capacity are N and F , respectively. Thus, the computation complexity of each iteration is $O(NF)$, and the overall computational complexity of the PGDC algorithm is $O(t_{max}NF)$. The dynamic iterative algorithm mainly comprises two steps: calculating the cost of BS and executing the PGDC algorithm. The complexity of calculating the cost is $O(N^2F)$, and the complexity of the PGDC algorithm is $O(t_{max}NF)$. Therefore, given that the dynamic iterative algorithm runs for $\frac{r_{max}}{\Delta_r}$ iterations, its overall computational complexity is $O(\frac{r_{max}}{\Delta_r} NF(N + t_{max}))$.

6. Performance Evaluation

In this section, we show the numerical results to validate the existence of SE in the proposed Stackelberg game. Moreover, we compare the performance of the PGDC algorithm with other benchmark caching schemes under different simulation parameters.

6.1. Experimental Setups

We consider a D2D assisted IoT caching network with a base station and 50 UEs, where the transmission radius of BS is 500 m and the maximum communication distance of UEs is 50 m. There are 40 pieces of content in the network, each with a size of 100 MB. The location distribution of BS and UEs in the simulation environment is shown in Figure 2. Each UE has no more than four neighbors and every UE has heterogeneous content request probability based on its own preference. In the simulations, we set the ranks of contents for each UE randomly to model different preferences of UEs. We further compare the PGDC algorithm in Algorithm 1 with three other benchmark caching schemes: (1) the most popular caching (MPC) scheme [10], in which each UE caches the c most favourite pieces of content according to their preference for a given reward r ; (2) the greedy caching (GC) scheme [12], in which each UE caches c pieces of content based on its own maximum utility $U_n(x_n, x_{j_n})$ for a given reward r , without considering its local cooperative utility $U_n^{lc}(x_n, x_{j_n})$; and (3) the random caching (RC) scheme, in which each UE caches c pieces of content randomly for a given reward r . In addition, the primary simulation parameters are listed in Table 2.

Table 2. Simulation parameters.

Parameter	Value
Number of UEs (N)	50
Number of contents (F)	40
Size of content (S)	100 MB
Transmission power of BS (P_0)	30 w
Transmission power of UE (P_n)	0.5 w
Bandwidth for D2D links ($B_{n,j}$)	30 MHz
Bandwidth for BS-UE links ($B_{n,0}$)	10 MHz
Unit transmission delay cost (η_0)	0.05
Unit transmission delay cost (η_1)	0.1
Unit transmission energy cost (η_2)	0.1
Noise power (σ^2)	−174 dBm/Hz
Pathloss exponent (γ)	3
Channel gain ($ h_{n,j} $)	10^{-2}
Unit transmission delay cost (η_0)	0.05
Learning parameter (β)	50

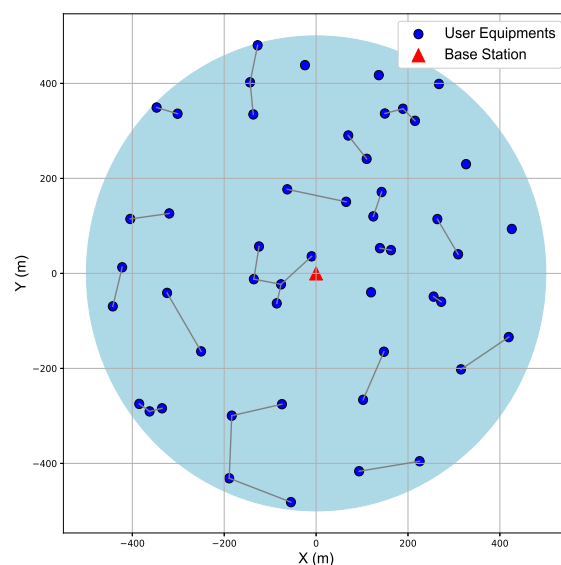


Figure 2. The location distribution of BS and UEs in the simulation environment.

6.2. Numerical Results

We first evaluate the convergence of the PGDC algorithm in Algorithm 1 with different caching capacities when we set the incentive reward $r = 1$. In Figure 3, we find that the number of iterations needed for the total utility of all UEs to reach the stable state under different caching capacity is no more than 1500, which indicates that the designed PGDC algorithm can converge quickly to a stable cache state. Moreover, we can further conclude that the larger caching capacity is, the greater the total utility will be. This is because a larger caching capacity helps UEs share more content under the given incentive reward, while reducing the delay cost of requesting content from the base station.

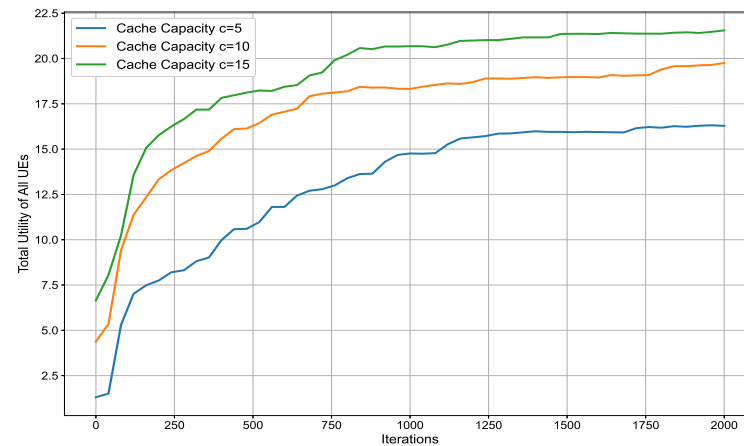


Figure 3. The convergence performance of the PGDC algorithm with different caching capacity.

Figure 4 shows the impact of incentive reward r on the total utility U_{UE} for all UEs and the cost C_{BS} of BS and illustrates the existence of the Stackelberg Equilibrium of the proposed game. It can be observed that the total utility increases monotonically with the given reward r . That is because a larger r encourages UEs to be more willing to cache content based on their neighbors' preferences, which leads to larger incentive revenue. Furthermore, we find that the cost of BS decreases with the increase in reward r at the beginning. However, when the reward r gradually increases to a point (i.e., r_{opt}), the cost of BS will begin to increase. This is because UEs will take into account neighbors' preferences with regard to the incentive reward, which enables the requested content to be directly obtained from neighbor UEs and reduces the request from the base station, thereby lowering the transmission delay cost of the base station. In addition, the initial incentive reward offered by the base station is significantly less than the unit transmission energy cost. Therefore, the total cost of the BS decreases in the early stage. As the incentive rewards gradually increase, the incentive expenses for BS start to rise. Additionally, due to the limited caching capacity of UEs, when the preferences of their neighbors cannot be satisfied, some requested contents will inevitably not be retrieved in the local cache and the neighbor caches. These requests will be sent to the BS, thereby increasing the transmission energy expenses of the BS. In Figure 4, when we gradually increase the incentive reward, we can find that the optimal incentive reward r_{opt} is 0.12, which corresponds to the minimum cost of BS. This point, together with its corresponding optimal joint caching strategy, constitutes the equilibrium point we aim to identify in the Stackelberg game.

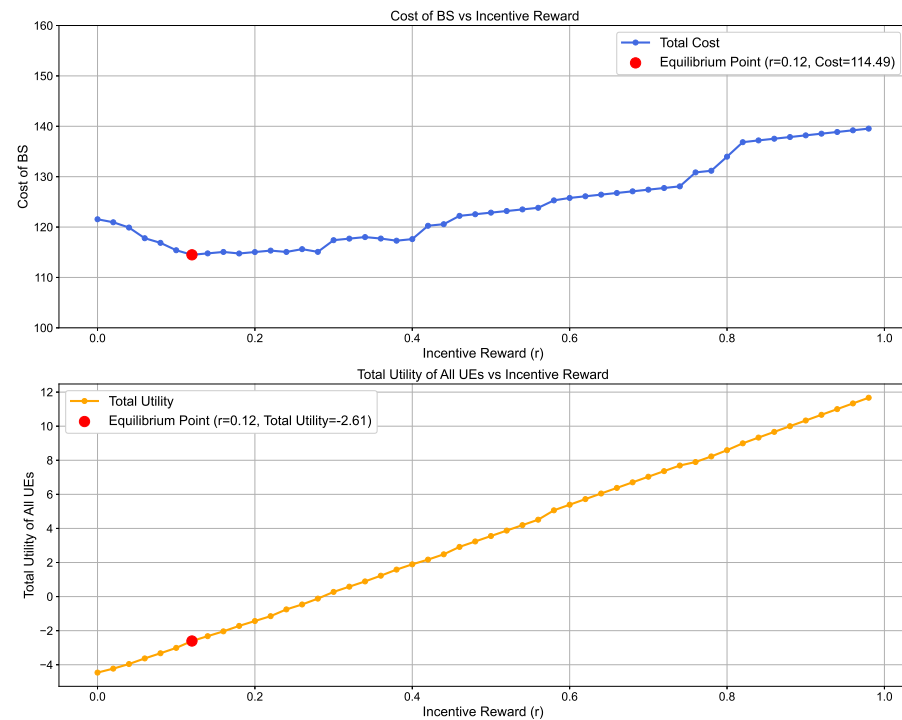


Figure 4. The illustration of the existence of Stackelberg Equilibrium under different incentive rewards.

Figure 5 illustrates the impact of Zipf exponent α on the total utility U_{UE} for the proposed PGDC algorithm and the other three caching schemes when we set the incentive reward $r = 1$ and each UE's capacity is five. It can be seen that the total utility in PGDC algorithm, MPC scheme and GC scheme increases respectively when α increases from 0.8 to 1.6. The total utility of the RC scheme shows some fluctuations when α increases. When α is larger, there are fewer popular pieces of content in the network, which means that the top-ranked content for each UE has a larger request probability. Hence, once the popular content has been cached, resulting in larger request demands, the total utility in the PGDC algorithm, MPC scheme, and GC scheme will increase accordingly. In RC scheme, the popular content may not be cached when α increases, which leads to fluctuations in the total utility. Moreover, we find that our proposed PGDC algorithm achieves the highest total utility compared with the other three caching schemes, indicating that it can yield additional utility gains. Since each UE has heterogeneous caching preference, in the MPC scheme, UEs only cache their own favourite c content, and do not consider their neighbors' preferences. Hence, the total utility of the MPC scheme is smaller than that of the proposed PGDC algorithm.

Figure 6 shows the impact of caching capacity on the total utility for the proposed PGDC algorithm and the other three caching schemes when we set the incentive reward $r = 1$ and Zipf exponent $\alpha = 1.2$. We find that the total utility of the proposed PGDC algorithm is the highest compared with other benchmark caching schemes. When the caching capacity is five, quantitative analysis shows the PGDC algorithm achieves a total utility that is 25.1% higher than the GC scheme, and 86.1% higher than the MPC scheme. Moreover, it can be observed that the total utility increases with the caching capacity. This can be explained based on the fact that when the caching capacity becomes larger, each UE is able to cache more content in its local cache, which can satisfy its own needs or the requests of neighbors. Accordingly, a larger caching capacity enables UEs to either gain more incentive benefits or reduce delay costs, thereby contributing to an increase in total utility.

Figure 7 compares the total utility under different amounts of content when the incentive reward r is 1, Zipf exponent α is 1.2 and the caching capacity is 5. When the amount of content in the network increases, the total utility of UEs shows a decreasing trend in all four caching algorithms. The main reason for this change is that the caching capacity is limited and the requests of UEs also become more diverse as the amount of content increases. When the limited local caching capacity and the content in the neighbor caches cannot meet the diverse demands, UEs will request content from the BS, which increases the corresponding transmission delay cost and thereby reduces the total utility.

Figure 8 illustrates the impact of different numbers of neighbors on the total utility in all four caching algorithms. (In order to highlight the impact of the number of neighbors on the total utility, the location distribution of BS and UEs shown in Figure 2 is no longer adopted. In each comparison, it is assumed that the number of neighbors for each UE is the same. For example, when we set the number of neighbors to 2, all UEs have two neighbors in all caching algorithms.) It can be observed that the total utility improves significantly when the number of neighbors increases from 1 to 4. Furthermore, the proposed PGDC algorithm exhibits the highest growth rate in this scenario. This is because increasing the number of neighbors can enhance the chances of content sharing or content requests, which increases the incentive benefits or reduces the transmission delay cost accordingly. Hence, the total utility presents a notable growth when the number of neighbors increases.

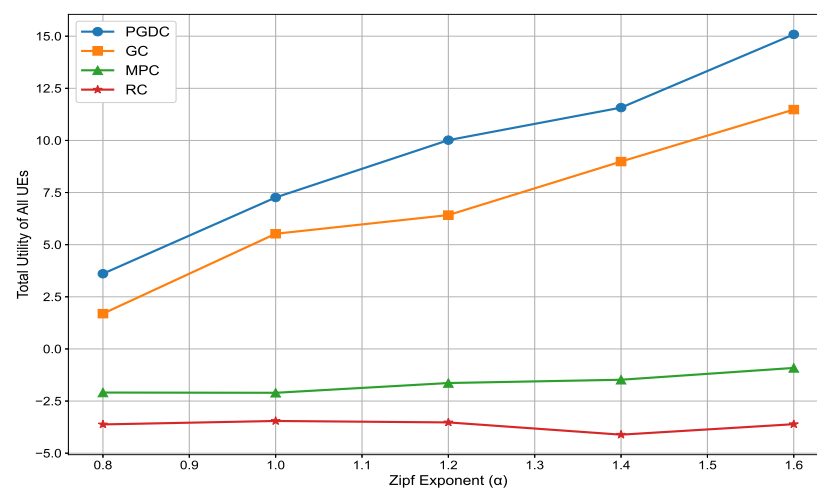


Figure 5. The total utility of all UEs with different Zipf exponents.

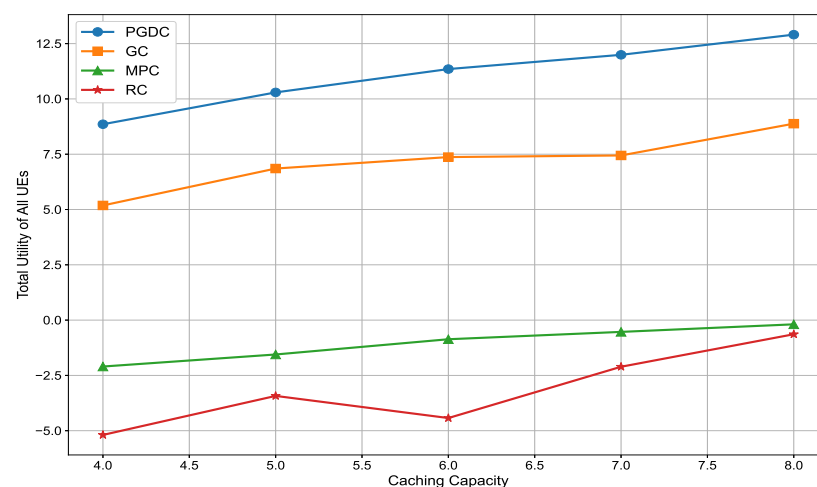


Figure 6. The total utility of all UEs with different caching capacity.

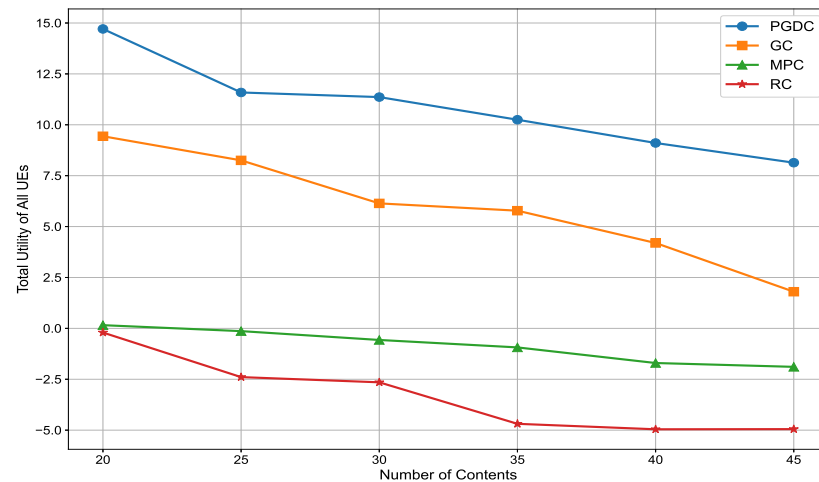


Figure 7. The total utility of all UEs with different amounts of content.

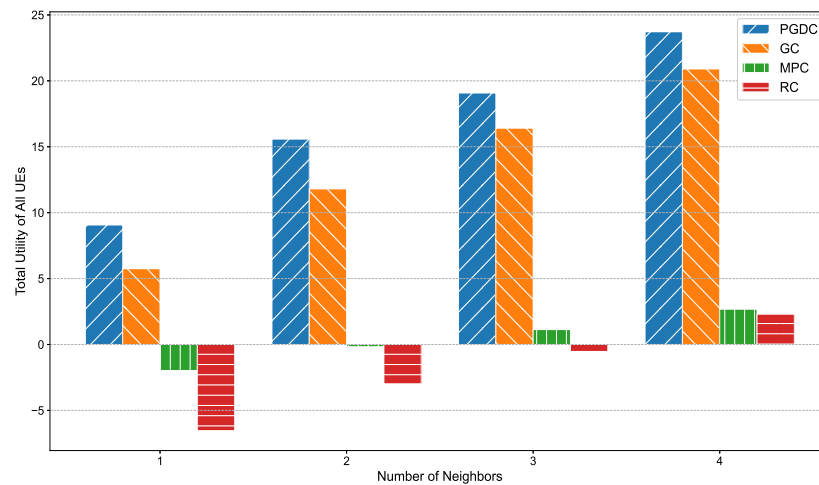


Figure 8. The total utility of all UEs with different numbers of neighbors.

Finally, we conduct experiments to validate the performance gap between the Nash Equilibrium (NE) solutions derived via the PGDC algorithm and the global optimal solutions—a gap for which a theoretical bound is provided in Theorem 2. Since finding the true global optimum $U_{UE}(X_{opt})$ for larger networks is computationally prohibitive (NP-hard), we instead use a small-scale network setup with a limited number of UEs $N = 3$ and content $F = 5$. Table 3 compares the ratio $\frac{U_{UE}(X_{NE})}{U_{UE}(X_{opt})}$ under different caching capacities when the incentive reward r is 1, Zipf exponent α is 1.2. We find that all ratios $\frac{U_{UE}(X_{NE})}{U_{UE}(X_{opt})}$ for different NE strategies fall within the range $[\frac{1}{1+\delta}, 1]$, and increasing iterations of the PGDC algorithm can improve the ratio. When the number of iterations is large enough, we find the ratio is equal to 1, which fully demonstrates that the PGDC algorithm is capable of finding the optimal joint caching strategy. Furthermore, when capacity is equal to 4, the optimal total utility decreases compared to the other two scenarios. This is because the content cached by each UE can basically meet its own needs, and the mutual requests between UEs are also reduced accordingly. This leads to a decrease in the incentive earnings, thereby causing a reduction in the total utility.

Table 3. Different capacity vs. iterations.

Capacity	$U_{UE}(X_{opt})$	10 Iterations		25 Iterations		50 Iterations	
		$U_{UE}(X_{NE})$	Ratio	$U_{UE}(X_{NE})$	Ratio	$U_{UE}(X_{NE})$	Ratio
2	2.43	1.78	0.73	2.37	0.98	2.43	1.0
3	2.05	1.59	0.77	2.05	1.0	2.05	1.0
4	1.43	1.43	1.0	1.43	1.0	1.43	1.0

Table 4 compares the ratio $\frac{U_{UE}(X_{NE})}{U_{UE}(X_{opt})}$ under different Zipf exponents when the incentive reward r is 1, and the caching capacity is 2. We also find that, for different NE strategies, all ratios $\frac{U_{UE}(X_{NE})}{U_{UE}(X_{opt})}$ lie within the range $[\frac{1}{1+\delta}, 1]$. Similarly, increasing iterations of the PGDC algorithm can improve the ratio. When the number of iterations is sufficiently large, the PGDC algorithm can find the optimal joint caching strategy.

Table 4. Different Zipf exponents vs. iterations.

Zipf Exponents	$U_{UE}(X_{opt})$	10 Iterations		25 Iterations		50 Iterations	
		$U_{UE}(X_{NE})$	Ratio	$U_{UE}(X_{NE})$	Ratio	$U_{UE}(X_{NE})$	Ratio
0.8	2.23	1.58	0.71	2.18	0.97	2.23	1.0
1	2.34	1.66	0.71	2.28	0.98	2.34	1.0
1.2	2.43	1.78	0.73	2.37	0.98	2.43	1.0

7. Conclusions

In this paper, we investigated a joint caching incentive and caching placement optimization problem in a D2D-assisted IoT caching network which consists of one BS and several cache-enabled UEs. We first introduced an uncoded caching incentive mechanism with the consideration of users' different content preferences. Then, we formulated the economic interactions between BS and UEs as a one-leader–one-follower Stackelberg game. In the proposed game, the BS initially determines the unit incentive reward to all UEs to encourage content sharing between UEs. Based on the given reward, UEs make the joint caching decision to maximize their total utility. In order to find an SE of the proposed Stackelberg game, we proposed a PGDC algorithm to achieve the maximum total utility of UEs and to obtain the optimal joint caching strategy as the follower's best response. Furthermore, we proved the convergence of the PGDC algorithm. Based on the optimal joint caching strategy, we developed a dynamic iterative algorithm to solve the BS's cost optimization problem. Finally, we presented the numerical results to validate the existence of SE in the proposed Stackelberg game. We also compared the performance of the proposed PGDC algorithm with other benchmark caching schemes. The simulation results demonstrate a significant performance advantage for our approach. For instance, under typical network settings with caching capacity of $c = 5$, the PGDC algorithm yields a total utility gain of 25.1% over GC algorithm and 86.3% over MPC algorithm. This highlights the effectiveness of our PGDC algorithm in achieving substantial extra utility gain for the D2D-assisted IoT caching network.

Author Contributions: System model design, algorithm proposal, and simulation conduction, J.R. Manuscript writing, editing, and review: J.R. and C.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China under Grant NO. 62201350; Scientific Research Fund of Zhejiang Provincial Education Department under Grant NO. Y202454938; Natural Science Foundation of Zhejiang University of Science and Technology under Grant NO. 2023QN115.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

IoT	Internet-of-Things
BS	Base station
UE	User equipment
D2D	Device-to-device
OFDMA	Orthogonal frequency division multiple access
PGDC	Potential game based distributed caching
SE	Stackelberg Equilibrium
NE	Nash Equilibrium
MPC	Most popular caching
GC	Greedy caching
RC	Random caching

References

1. Yang, Y.; Wang, C.-X.; Huang, J.; Thompson, J. Characteristics and channel capacity studies of a novel 6G non-stationary massive MIMO channel model considering mutual coupling. *IEEE J. Sel. Areas Commun.* **2024**, *42*, 1519–1533. [\[CrossRef\]](#)
2. Zhang, J.; Yan, S.; Peng, M.; Ouyang, Q. Coordinated multi-point enabled ISAC under asynchronous errors: Performance analysis and waveform-beamforming optimization. *IEEE Trans. Veh. Technol.* **2025**, *74*, 12189–12205. [\[CrossRef\]](#)
3. Mu, X.; Xu, J.; Wang, Z.; Al-Dhahir, N. Simultaneously transmitting and reflecting surfaces for ubiquitous next-generation multiple access in 6G and beyond. *Proc. IEEE* **2024**, *112*, 1346–1371. [\[CrossRef\]](#)
4. Liu, T.; Li, J.; Shu, F.; Guan, H.; Yan, S.; Jayakody, D.N.K. On the incentive mechanisms for commercial edge caching in 5G wireless networks. *IEEE Wireless Commun.* **2018**, *25*, 72–78. [\[CrossRef\]](#)
5. Zheng, Q.; Shan, H.; Hou, F.; Shi, Z.; Zhang, Z. Incentive mechanism design for green mobile D2D caching networks. *IEEE Trans. Green Commun. Netw.* **2022**, *6*, 484–499. [\[CrossRef\]](#)
6. Cheng, Y.; Zhang, J.; Yang, L.; Zhu, C.; Zhu, H. Joint multioperator virtual network sharing and caching in energy harvesting-aided environmental Internet of Things. *IEEE Internet Things J.* **2020**, *7*, 7689–7701. [\[CrossRef\]](#)
7. Xu, Q.; Su, Z.; Ni, J. Incentivizing secure edge caching for scalable coded videos in heterogeneous networks. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 2480–2492. [\[CrossRef\]](#)
8. Chan, Y.W.; Shih, C.M.; Chien, F.T. Enhancing content caching in D2D networks: A Stackelberg game approach with wireless energy transfer and social awareness. *IEEE Access* **2025**, *13*, 26765–26781. [\[CrossRef\]](#)
9. Zhu, Y.; Zhu, Q. Joint optimization algorithm for UAV-assisted caching and charging based on wireless energy harvesting. *Appl. Sci.* **2025**, *15*, 3908. [\[CrossRef\]](#)
10. Pappas, N.; Chen, Z.; Dimitriou, I. Throughput and delay analysis of wireless caching helper systems with random availability. *IEEE Access* **2018**, *6*, 9667–9678. [\[CrossRef\]](#)
11. Zhang, T.; Wang, Y.; Yi, W.; Liu, Y.; Nallanathan, A. Joint optimization of caching placement and trajectory for UAV-D2D networks. *IEEE Trans. Commun.* **2022**, *70*, 5514–5527. [\[CrossRef\]](#)
12. Cao, Y.; Maghsudi, S.; Ohtsuki, T.; Quek, T.Q.S. Mobility-aware routing and caching in small cell networks using federated learning. *IEEE Trans. Commun.* **2023**, *72*, 815–829. [\[CrossRef\]](#)
13. Zhao, X.; Zhu, Q. Mobility-aware and interest-predicted caching strategy based on IoT data freshness in D2D networks. *IEEE Internet Things J.* **2021**, *8*, 6024–6038. [\[CrossRef\]](#)
14. Kumar, S.; Misra, S. Joint content sharing and incentive mechanism for cache-enabled device-to-device networks. *IEEE Trans. Veh. Technol.* **2021**, *70*, 4993–5002. [\[CrossRef\]](#)

15. Khan, K.S.; Naeem, A.; Jamalipour, A. Incentive-based caching and communication in a clustered D2D network. *IEEE Internet Things J.* **2022**, *9*, 3313–3320. [[CrossRef](#)]
16. Han, Z.; Niyato, D.; Saad, W.; Basar, T.; Hjørungnes, A. *Game Theory in Wireless and Communication Networks: Theory, Models and Applications*; Cambridge University Press: Cambridge, UK, 2012.
17. Su, Z.; Xu, Q.; Yang, Q.; Hou, F. Edge caching for layered video contents in mobile social networks. *IEEE Trans. Multimedia* **2017**, *19*, 2210–2221. [[CrossRef](#)]
18. Zou, J.; Li, C.; Zhai, C.; Xiong, H.; Steinbach, E. Joint pricing and cache Placement for video caching: A game theoretic approach. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 1566–1583. [[CrossRef](#)]
19. Jiang, C.; Gao, L.; Luo, J.; Zhou, P.; Li, J. A game-theoretic analysis of joint mobile edge caching and peer content sharing. *IEEE Trans. Netw. Sci. Eng.* **2023**, *10*, 1445–1461. [[CrossRef](#)]
20. Zheng, Z.; Song, L.; Han, Z.; Li, G.Y.; Poor, H.V. A Stackelberg game approach to proactive caching in large-scale mobile edge networks. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 5198–5211. [[CrossRef](#)]
21. Jiang, C.; Gao, L.; Luo, J.; Gong, S. Crowdsourcing for mobile edge caching: A game-theoretic analysis. In Proceedings of the IEEE International Conference on Communications, Shanghai, China, 20–24 May 2019.
22. Yang, Y.; Liu, Z.; Liu, Z.; Chan, K.Y.; Xie, Y.; Guan, X. Joint optimization of edge computing resource pricing and wireless caching for blockchain-driven networks. *IEEE Trans. Veh. Technol.* **2022**, *71*, 6661–6670. [[CrossRef](#)]
23. Guo, M.; Zhang, D.; Xing, W.; Shao, X.; Liu, Z.; Zhang, Y. Optimal multi-bitrate video caching and processing in edge computing: A Stackelberg game approach. *IEEE Internet Things J.* **2025**, *12*, 25059–25076. [[CrossRef](#)]
24. Breslau, L.; Cao, P.; Fan, L.; Phillips, G.; Shenker, S. Web caching and Zipf-like distributions: Evidence and implications. In Proceedings of the IEEE International Conference on Computer Communications, New York, NY, USA, 21–25 March 1999; pp. 126–134.
25. Sun, H.; Zhou, Y.; Zhang, H.; Ale, L.; Dai, H.; Zhang, N. Joint Optimization of Caching, Computing and Trajectory Planning in Aerial Mobile Edge Computing Networks: A MADDPG Approach. *IEEE Internet Things J.* **2024**, *11*, 40996–41007. [[CrossRef](#)]
26. Poularakis, K.; Iosifidis, G.; Argyriou, A.; Tassioulas, L. Video delivery over heterogeneous cellular networks: Optimizing cost and performance. In Proceedings of the IEEE International Conference on Computer Communications, Toronto, ON, Canada, 27 April–2 May 2014; pp. 1078–1086.
27. Fudenberg, D.; Tirole, J. *Game Theory*; MIT Press: Cambridge, MA, USA, 1991.
28. Monderer, D.; Shapley, L.S. Potential games. *Games Econ. Behav.* **1996**, *14*, 124–143. [[CrossRef](#)]
29. Shanmugam, K.; Golrezaei, N.; Dimakis, A.; Molisch, A.; Caire, G. Femtocaching: Wireless content delivery through distributed caching helpers. *IEEE Trans. Inf. Theory* **2013**, *59*, 8402–8413. [[CrossRef](#)]
30. Wu, H.; Shang, H. Potential game for dynamic task allocation in multi-agent system. *ISA Trans.* **2020**, *102*, 208–220. [[CrossRef](#)]
31. Vetta, A. Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions. In Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science, Vancouver, BC, Canada, 19 November 2002; pp. 416–425.
32. Sutton, R.; Barto, A. *Reinforcement Learning: An Introduction*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2018.
33. Young, H.P. *Individual Strategy and Social Structure*; Princeton University Press: Princeton, NJ, USA, 1998.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.