Chapter 3
Efficient and Fair Resource Trading Management

Abstract In this chapter, we investigate the resource trading problem in a utility and cloud computing setting where multiple tenants communicate in a Peer-to-Peer (P2P) fashion. Enabling resource trading in cloud unleashes the untapped cloud resources, thus presents a flexible solution for managing resource allocation. However, finding an efficient and fair resource allocation is challenging mainly due to the heterogeneity of resource valuations. Our work first develops a utility-oriented model to support resource negotiation and trading. Based on this model, we adopt a multiagent-based technique that allows a group of autonomous tenants to reach an efficient and fair resource allocation. Further, we add budget limitation to each tenant and propose a directed hypergraph model to facilitate resource trading amongst heterogeneous tenants. We develop a directed hypergraph model to facilitate trading decision making, and design a class of heuristic-based distributed resource trading protocols in favor of different performance metrics.

The rest of the chapter is organized as follows. We first present an overview of the proposed research in Sect. 3.1. We then summarize the related work in Sect. 3.2. In Sect. 3.3, we describe the problem setting and quantify the objectives of the resource trading problem. In Sect. 3.4, we introduce a multiagent-based technique to achieve optimal resource trading efficiency and fairness. Section 3.5 further investigates allocation strategies with limited budget. We propose a novel directed hypergraph model and develop a series of distributed resource trading protocols based on heuristic approaches. Finally, Sect. 3.6 shows simulation results and analyzes their implications.

H. Zhao and X. Li, *Resource Management in Utility and Cloud Computing*, SpringerBriefs in Computer Science, DOI 10.1007/978-1-4614-8970-2_3,

3.1 Overview

Nowadays, the utility and cloud computing model is mostly vendor driven, with users having no control over the data or the technology supported by the cloud. Such a vendor-driven model, although convenient to use, brings many issues to light, e.g., failure of monocultures, tradeoff between convenience and control, and concerns about environmental impact [5]. To address these issues, researchers have proposed an alternative model that provides a collaborative resource sharing platform that forms a community-based cloud computing environment [16, 18, 24]. Different from the centralized vendor model, this community-based cloud leverages under-utilized networked private resources for infrastructure support. Tenants within the same community cloud typically share common security and compliance concerns, and may delegate management to some trusted third-party organization.

Similar to the centralized vendor-driven model, the community-based model offers computation and storage resources as metered services. Therefore, the design goal of the shared cloud resource platform should not only focus on the quality of computing service, but should equally address the economic aspect such that tenants receive cost-effective cloud service provisioning. While managing resource allocation is relatively straightforward in the centralized vendor-driven model (e.g., Amazon®'s on-demand and spot instance pricing), it is particularly challenging due to the heterogeneity in the multitenancy environment. In a community cloud, we are facing a free market where tenants are only incentivized to accept profitable resource exchange. As a result, a well designed multitenancy resource trading protocol is highly desirable to effectively regulate the management of resource allocation.

In this chapter, we study the distributed resource trading problem in a community-based utility and cloud computing environment, and propose a set of multitenancy resource trading protocols to jointly optimize resource allocation efficiency and fairness. Specifically, better efficiency refers to the increased aggregate valuations of all the tenants, and better fairness is interpreted as reduced envy between every pairwise combination of tenants. Our solution follows a market-oriented design principle, and uses a directed hypergraph model to integrate these two seemingly conflicting design objectives into one unified resource trading framework. It directly extends the work of Chevaleyre [10], and further addresses the challenge of budget limited resource trading. With systematic analysis of the resource trading market, a set of heuristic-based distributed resource trading protocols are developed and evaluated.

The comprehensive study presented in this chapter has broad utility in the growing world of "everything-as-a-service". It characterizes the extent to which independent and self-interested tenants interact with each other. Our analysis shows that incentive preserving resource exchanges tend to benefit the system, both from a global view of the overall service efficiency and from a local view of the improved service quality valuation. Moreover, the proposed resource trading approaches are complementary to the vendor-driven cloud computing. For example, consider user

Alice rents a virtual machine from Amazon® with reserved instance pricing. After Alice finishes her job and before the lease expires, Alice might "sublease" this virtual machine to user Bob in order to partially compensate for her resource rental cost.

3.2 Related Work

The study described in this chapter presents distributed protocol design to jointly optimize resource trading efficiency and fairness. As the organization of distributed resource evolves towards a more hierarchical architecture [20], distributed algorithms designed for solving combinatorial multi-criteria optimization problems become more attractive. Common optimization techniques include machine learning [26], evolutionary algorithms [13], swarm intelligence [25], and socialeconomy approaches [21, 27]. All these approaches share a common flavor that involves interacting entities evolving towards the optimal solution (by following certain learning or negotiation rules). Our proposed approach falls into the category of socialeconomy approaches. They are built based on the observation that resource management in distributed systems shares common features with commodity allocations driven by market power in the economic study. It is widely adopted to create a computational economy for grid computing [1, 6] and the emerging cloud computing [7, 30]. In an early study, Wolski et al. [32] presented two different market strategies for controlling resource allocation, namely commodities markets and auction. The commodities markets strategy treats disparate resources as interchangeable commodities, while auction requires orchestration from a centralized auctioneer for collecting bids and determining winners. Our proposed resource trading framework is designed for a community cloud environment, and belongs to the commodities market category. In particular, we propose a P2P resource trading market for managing cloud resource allocation. Example research related to this notion includes [12, 31]. In [12], a P2P data replication system was proposed to improve fault-tolerance of digital collections in library. In [31], the authors proposed a multiple currency economy that any peer can issue its own currency. Different from their design, peers directly exchange resources in our distributed resource trading design.

In this chapter, two economic metrics are used to quantify the quality of an allocation: efficiency in terms of overall social welfare, and fairness in terms of envy-freeness. The metric of efficiency is important to characterize the achievable system performance, and was studied in a number of publications [3, 17, 34]. Meanwhile, the metric of fairness highlights individual's utility such that each individual achieves the maximum contentment of its allocated share [14]. Compared to efficiency, the envy-free fairness has generally received far less attentions. A related work targeting grid computing is found in [28]. Using game theory, the authors tackled a multicriteria optimization problem with the aid of axiomatic theory of equity. The authors concluded that for fair and feasible scheduling on global scale computational grid, a strong community control is required. The research conducted in this chapter approaches the multicriteria optimization problem from a different angle, and further investigates how to balance the two metrics amongst budget-aware distributed tenants.

Our proposed protocols utilize a directed hypergraph model. A hypergraph is an extension of the graph concept that one edge (called a hyperedge) can connect an arbitrary set of vertices rather than two. A hypergraph model is flexible and informative to use in algorithm design as it generalizes the graph. For that reason, it becomes attractive to improve algorithm performance in various research domains, e.g., page reputation computation for search engines [2], cellular mobile communication [29] and memory management [19]. For large-scale scientific computing, Çatalyürek and Aykanat [9] proposed a multilevel partitioning approach for mapping repeated sparse matrix-vector computations to multicomputers using hypergraph. Their approach significantly reduces communication overheads while achieving drastically improved mapping results. In their hypergraph model, hyperedges represent affinity among subsets of the data, and the weights reflect the strength of this affinity. We model the resource trading problem in a similar manner that aims to optimize the aggregate weights of the directed hypergraph model.

3.3 A Distributed Resource Trading Framework

This section presents the design overview of a distributed resource trading framework for the community cloud. In Sect. 3.3.1, we depict the resource trading system model. In Sect. 3.3.2, we clarify the problem assumptions, define the goals for resource trading, and formulate the problem.

3.3.1 System Model

Consider a scenario where a number of highly autonomous tenants connected in a P2P manner, each holding a set of indivisible resources. A resource is an abstraction of hardware bundle or software service, e.g., Virtual Machine (VM), computational time, etc. These resources form a publicly accessible resource pool, and they are completely allocated to all the tenants initially, as described in Fig. 3.1. All tenants form a collaborative community with common purposes and concerns. The underlying P2P communication infrastructure ensures that every tenant is able to talk to every other tenant within the same community (they may not communicate directly, but there is at least one communication path between every pairwise tenants on the topology). For this study, we do not consider dynamic tenants join and leave. We also assume that the distributed system is reliable. Any resource can be assigned to any tenant, incurring certain benefit and cost that may vary depending on the specific resource-tenant assignment. Each tenant can be involved in any number of

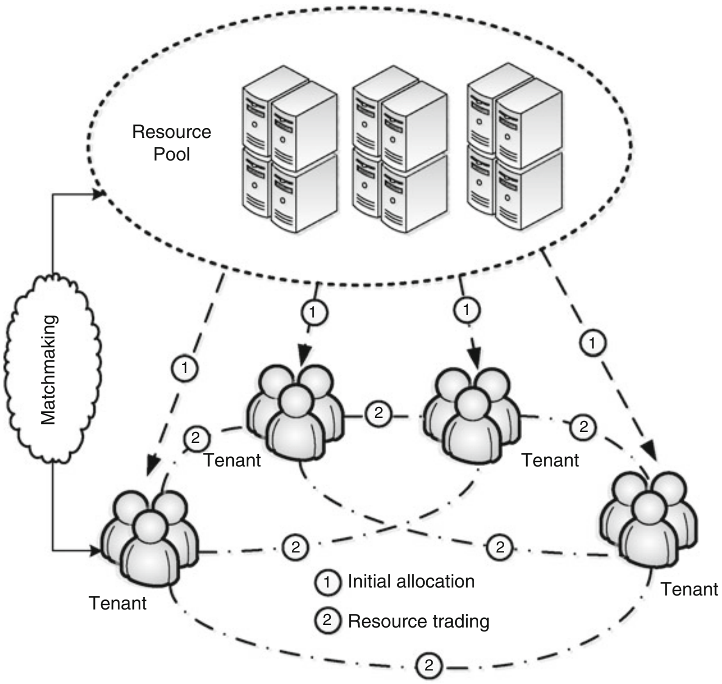

Fig. 3.1 Multitenancy resource trading: system model

resource trading activities, following the specific tenant negotiation protocol. The distributed resource trading results in a remapping of resources to tenants. We call each instance of such a resource remapping matchmaking. Tenants are incentivized to purchase under-utilized resources from the tenants who currently hold them. As a result, the system evolves towards better resource utilization in the long run.

Formally, let $\mathbb{P} = \{p_1, \ldots, p_n\}$ be the finite set of tenants, and let $\mathbb{R} = \{r_1, \ldots, r_m\}$ be the finite set of indivisible resources. Typically we have $|\mathbb{R}| > |\mathbb{P}|$. This, however, is not necessarily always the case, i.e., some tenants may obtain empty allocation. A matchmaking is defined as a mapping $\mathscr{A}\colon \mathbb{P} \to 2^{\mathbb{R}}$. More specifically, we have the following definition:

Definition 3.1 (Matchmaking). A matchmaking $A = \{A_1, A_2, \ldots, A_n\}$ is a mapping $\mathscr{A}\colon \mathbb{P} \to 2^{\mathbb{R}}$ satisfying: $A_i \bigcap A_j = \emptyset$, and $\bigcup A_i = A$.

The condition of $\bigcup A_i = A$ ensures that the final matchmaking result is a complete allocation.

3.3.2 Problem Statement

For each tenant, we assume a private valuation model, indicating that tenants are mutually blind to each other and evaluate individual allocation independently. The **valuation** of p_i is defined by the valuation function $V_i(\cdot)$, $V_i(\emptyset) = 0$ and $V_i(A_i) \geq V_i(A_i^*)$ for all $A_i \supseteq A_i^*$. Moreover, we assume the valuation function is modular, i.e., $V_i(A_i \cup A_j) = V_i(A_i) + V_i(A_j) - V_i(A_i \cap A_j)$ for all $A_i, A_j \subseteq A$.

Our first goal for distributed resource trading protocol design is to achieve optimal matchmaking efficiency such that the **social welfare**, i.e., $\omega = \sum_{i=1}^{n} V_i(A_i)$, is maximized.

Definition 3.2 (Efficiency). Let Γ be the set of all possible matchmaking results, an efficient matchmaking is an allocation $A = \{A_1, A_2, \ldots, A_n\}$ that maximizes the social welfare: $\omega_{max} = \max_{A \in \Gamma} \sum_{p_i \in \mathbb{P}} V_i(A_i)$.

The efficiency criterion reflects the overall system performance. For example, suppose there are two resources, one with 2 cores + 1G memory and the other one with 1 core + 2G memory, also assuming user Alice has a CPU-bound job and user Bob has a memory-bound job. Therefore, Alice has higher valuation for the first resource while Bob prefers the second resource. By assigning the first resource to Alice and the second to Bob, the aggregate valuation is maximized, and the system features best job turnaround time.

We define a *resource-bundle* as a collection of one or more resources held by any tenant p_i, i.e., a resource-bundle is a non-empty subset of A_i. We define a **Deal** as the basic event in the multitenancy resource trading framework. A deal represents the process of resource-bundle transfer from one tenant to another. In order to acquire resources from another tenant, certain amount of compensation is necessary to complete the deal. A **Payment Function** $\varphi_{i,j}$ defines this compensation amount p_i pays to p_j. If $\varphi_{i,j}$ is negative, then p_i receives money from p_j. Each tenant keeps a record of its payment history. Formally, we define p_i's **Balance** as the summation of its withdrawals and deposits in all deals p_i is involved in: $\theta_i = \sum \varphi_i$. All tenants are utility-driven that seek to make profit at each deal. Formally, suppose after a deal, the allocation of p_i becomes $\tilde{A}_i$, a deal must be a **Rational Deal (RD)** if and only if $V_i(\tilde{A}_i) - V_i(A_i) \geq \varphi_{i,j}$ for all $p_i \in \mathbb{P}$. Note that the requirement of rational deal applies to both tenants involved in the deal, thus is a bilateral constraint. The **Utility** of p_i is given as $U_i(A_i) = V_i(A_i) - \theta_i$.

The second goal of our protocol design is to promote fairness within the system. By associating the valuation and payment function, fairness denotes to envy-free [4] amongst all tenants, indicating that no tenant would get better off by swapping its allocation with another peer though a rational deal. Specifically, the definition of a fair allocation is given as follows.

Definition 3.3 (Fairness). Let Γ be the set of all possible matchmakings, a matchmaking result is characterized as fair iff there exists $A = \{A_1, A_2, \ldots, A_n\} \in \Gamma$ such that: (a) $\forall p_i, p_j \in \mathbb{P}$, p_i and p_j has direct connection; and (b) $V_i(A_i) - \theta_i \geq V_i(A_j) - \theta_j$.

The fairness criterion is in line with the envy-free definition given out in [10] that takes transferable utility into account. The authors proved that a **E**fficient and **E**nvy-**F**ree (**EEF**) state always exists. Here, we further extend their result by adding topology constraint to the fairness definition. Our definition limits envy-free states to neighboring tenants. This is justifiable as the underlying communication topology might not be a fully connected network. In addition, a common practice in distributed systems is to employ a budget transfer mechanism to enforce incentives for community control [22]. For example, in P2P and social networks, some form of digital cash, or numerical reputations representing trust relationships may be used for rewarding and punishing certain actions. We formally define budget constraint as follows.

Definition 3.4 (Budget). Budget b_i^t expresses maximum amount p_i is able to offer after t deals. Let b_i^0 be the initial budget initially, we have:

$$b_i^t = b_i^0 - \theta_i^t$$

Given any initial allocation, the goal of this study is to investigate to what extent efficiency and fairness can be achieved in the multitenancy resource trading framework described above, and to design resource trading protocols to guide tenant interactions evolving towards system-wide efficiency and fairness. We analyze situations with and without the budget limitation. From now on, we label the scenario with budget constraint as **budget-aware**, and refer to the later scenario as **budget-unaware**.

3.4 Budget-Unaware Resource Trading Protocol

In this section, we develop a resource trading protocol without the presence of budget constraint. Our protocol design is based on the multiagent-based resource allocation optimization framework presented in [10].

3.4.1 Preliminaries

By following certain payment rules, we will show that the resource trading protocol is capable of reaching topology-wide efficiency as well as envy-free fairness upon convergence. A topology-wide efficient allocation is an allocation such that for every tenant, the allocation for the sub-topology consisting of that tenant and its direct neighbors is efficient, i.e., the matchmaking achieves maximum social welfare on the sub-topology. We introduce topology-wide efficiency because for a partially connected communication topology, a globally efficient matchmaking is not guaranteed unless the order of resource trading is carefully planned. An example is given out in [33], Sect. 3.4.1.

In the resource trading framework, each tenant completes transactions with neighbors using only rational deals (RD), and obtains or loses resource bundle accordingly. An RD indicates that the transaction is beneficial for pushing resources to tenants who value them more. In fact, **ANY** sequence of RD executions will achieve efficiency with regard to the underlying communication topology. This is due to the following observations: (1) RD increases social welfare according to its definition; and (2) if no more RD is possible, then the matchmaking must reach the maximum possible social welfare. Given modular valuation function, we have the following proposition.

Proposition 1 (Convergence to Efficiency [15]). *Any sequence of RD involving any number of resource exchanges will eventually yield to topology-wide efficiency.*

The reasoning behind Proposition 1 is fairly simple. Each RD results in remapping of resources to tenants with higher interests. When no RD is possible with respect to the communication topology, the system converges to a topology-wide efficient state. Another implication is that the final state is independent of the execution order of RDs. Now suppose after an execution of an RD, the current allocation becomes $\tilde{A}$. Since the deal is bilaterally beneficial to both tenants involved in the deal, we calculate the payment range with the following equations.

$$\begin{aligned} V_i(\tilde{A}_i) - V_i(A_i) &\geq \varphi_{i,j} \\ V_j(\tilde{A}_j) - V_j(A_j) &\geq -\varphi_{i,j} \end{aligned} \tag{3.1}$$

By solving this equation, the result of the payment function $\varphi_{i,j}$ falls into the range of $[V_j(A_j) - V_j(\tilde{A}_j), V_i(\tilde{A}_i) - V_i(A_i)]$, i.e., the rational payment range.

3.4.2 A Multiagent Based Optimization Approach for Resource Trading

This section introduces the theoretical foundation of our multitenancy resource trading protocol design. It is mainly based on the theoretical framework developed by Chevaleyre et al. [10, 11] for multiagent systems. One central conclusion is that resource allocation efficiency and fairness can be simultaneously achieved in a multiagent negotiation framework. In order to achieve this state, a proper payment function was selected to deal with the increased social surplus $\omega(\tilde{A}) - \omega(A)$ after each deal. In particular, a payment function called **Globally Uniform Payment Function (GUPF)** was proposed. Suppose A and $\tilde{A}$ are allocations before and after an RD execution, respectively, the GUPF is defined as follows.

$$\textbf{GUPF: } \varphi_i = [V_i(\tilde{A}_i) - V_i(A_i)] - \frac{[\omega(\tilde{A}) - \omega(A)]}{n} \tag{3.2}$$

Equation (3.2) is labeled as globally uniform because this payment is imposed on all tenants. For tenants who do not involved in the deal, $V_i(\tilde{A}_i) - V_i(A_i)$ equals to zero, so each of them receives an equal share of the social surplus created by the trading activity. Note that GUPF is within the bound of rational payment (3.1). In addition to GUPF, a one-off payment amount at initial is introduced. The initial payment amount, called **initial equitability payment**, is defined as: $\varphi^0 = V_i(A_i^0) - \frac{\omega(A^0)}{n}$. The main purpose for this payment function is to "level the playing field". The next two theorems show that imposing initial equitability payment and GUPF for resource trading leads to efficient and fair matchmaking. The following theorem shows that individual utility is invariant after every RD.

Theorem 3.1. *If each tenant pays initial equitability payments at start and pays GUPF after each RD executes, then all tenants share the same utility: $U_i(A_i) = \frac{\omega(A)}{n}$ after each RD.*

With this invariant, we prove the following theorem. Note that our version is slightly different from that presented in [10], as we target at topology-wide efficiency and use a more strict assumption of modular domain.

Theorem 3.2 (Convergence to Efficiency and Fairness [10]). *When all valuations are modular and budget limitation is not a concern, paying initial equitability payment at start and GUPF after each RD for every $p_i \in \mathbb{P}$ will converge to a matchmaking state that achieves both topology-wide efficiency and envy-free fairness.*

More details about these two theorems and the implementation of the protocol are described in the Sect. 3.4 of [33].

3.5 Budget-Aware Resource Trading Protocol

3.5.1 Modeling Resource Trading Using a Directed Hypergraph

When budget constraint is imposed, the convergence to the optimal matchmaking state might not exist. In this section, we develop a directed hypergraph model for community-based cloud resource trading. A hypergraph is a generalization of the 2D graph that an edge can connect a set of vertices. If the hypergraph is directional, an edge (a.k.a. a hyperarc) connects a hypernode (head) with a set of hypernodes (tail set). The motivation behind the directed hypergraph model lies in its implication for one-to-many relationship. A 2D graph merely models connectivity among tenants, but cannot represent task allocation and envy relationship among them. A directed hypergraph is more informative, succinctly capturing the scenario that a resource is held by some tenant, but inspires more interest from some other tenants each holding a set of resources.

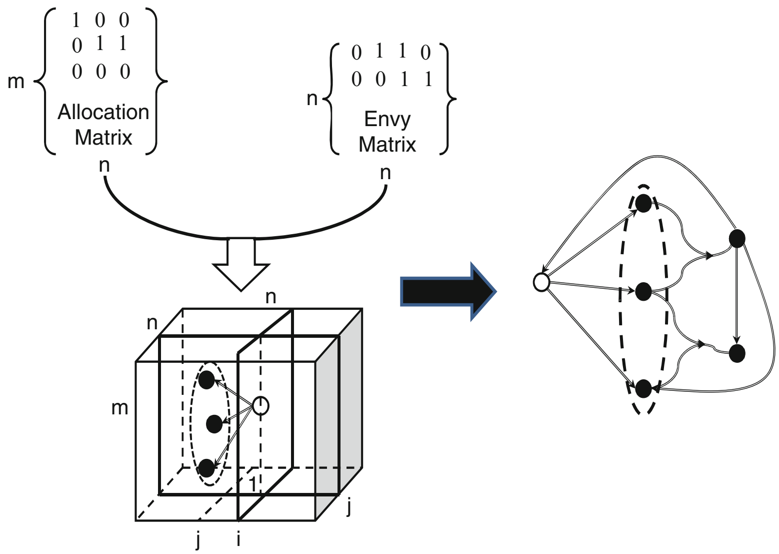

Fig. 3.2 A directed hypergraph model. The proposed directed hypergraph model derives from an $m \times n \times n$ hyperspace. A hypernode is a point mapping allocation and envy relationship on the hyperspace. A hyperarc connects a hypernode $v \in V$ with a set of other hypernodes belonging to a common tenant and has envy relationship with the v's host tenant. An example directed hypergraph is shown on the *right side*

We propose two matrices to build up a hyperspace. The first matrix is an Allocation Matrix (AM). It is an $m \times n$ matrix that takes binary values, representing current resource matchmaking state for all tenants. Each entry $\alpha_{i,j}$ in AM is defined as follows.

$$\alpha_{i,j} = \begin{cases} 1 & \text{tenant } j \text{ holds resource } i \\ 0 & \text{otherwise} \end{cases}$$

The second matrix is an Envy Matrix (EM) representing current matchmaking unfairness (or envy relationship). Suppose we have two tenants, Alice and Bob. Bob is said to envy Alice when Bob has higher valuation for some resource currently allocated to Alice. Again, we use binary values to represent the envy relationships. Formally, An Envy Matrix is a $n \times n$ matrix defined as follows.

$$\varepsilon_{i,j} = \begin{cases} 1 & p_i \text{ is envies } p_j \\ 0 & \text{otherwise} \end{cases}$$

Combining the allocation matrix and the envy matrix, we are ready to unifying allocation and envy relationships into one directed hypergraph model. We first create a three-dimensional space, $m \times n \times n$, as shown in the left side of Fig. 3.2. A directed hypergraph $H = (V, E)$ is composed of a finite non-empty set V of hypernodes and a finite non-empty set E of hyperarcs. Using the coordinates of the hyperspace, we define the hypernode as follows. A hypernode v (3.3) is a three-tuple (x, y, z), where $x \in \mathbb{R}$ represents the resource, $y \in \mathbb{R}$ represents the tenant currently holding x, and z is some tenant has envy relationship with y, i.e., $z \in \mathbb{P}, \varepsilon_{y,z} = 1$. A hyperarc e (3.4) is a pair $< T, h >$, where $T \subseteq V$ is the tail of e and $h \in V \setminus T$ is its head. The tail set T includes those hypernodes whose host tenants involved in envy relationships with the host of the head.

Hypernode: A hypernode is a three-tuple:

$$
\begin{aligned}
v = (x, y, z) \in V \\
\text{s.t. } x \in \mathbb{R} \\
y \in \mathbb{P}, \text{and } \alpha_{x,y} = 1 \\
z \in \mathbb{P}, \text{and } \varepsilon_{y,z} = 1
\end{aligned}
\tag{3.3}
$$

Hyperarc: A hyperarc $e \in E$ is an ordered pair $< T, h >$ iff:

$$
\begin{aligned}
e =< T, h >\in E \\
\text{s.t. } h = (x_1, y_1, z_1) \in V \\
v = (x_2, y_2, z_2) \in T \subseteq V \\
y_2 = z_1
\end{aligned}
\tag{3.4}
$$

Each tenant can establish a local view of the directed hypergraph. The hyperarcs imply potential transactions to be negotiated. In a distributed environment, when one transaction is accomplished using an RD, resource allocation changes which might affect other resource trading activities. Building a directed hypergraph is thus helpful to evaluate the quality of trading selections. For example, there are many applications of the optimal structures in the proposed directed hypergraph model, such as optimal spanning hypertree and optimal edge cover. Readers can find more information in Sect. 3.5.2 of [33].

3.5.2 Protocol Design

When proposing for resource trade, a tenant rationally calculates its payment amount. When the budget limitation b_i is imposed on $p_i \in \mathbb{P}$, the rational payment amount $\varphi_{i,j}$ for trade proposal is in the range of:

$$\varphi_{i,j} \in [V_j(A_j) - V_j(\tilde{A}_j), \min\{V_i(\tilde{A}_i) - V_i(A_i), b_i\}]. \tag{3.5}$$

Protocol 1: (a) V-BaMRT (b) E-BaMRT (c) P-BaMRT

```
begin
    for p_i ∈ ℙ do
        Establishes local view with neighboring peers;
        // --Trade Proposal--
        while p_i has at least one envious neighbor do
            a) Sorts potential transactions based on envy degree;
            b) Selects p_j with the highest envy degree drop;
            c) Selects payment within the range defined by Equation 3.5;
            if p_j accepts offer then
                • Make payments;
                • Removes p_j from its envy list
        // --Offer Selection--
        while conflicting offers arrival do
            Selects offer with;
                { (a) highest social welfare gain, or
                { (b) largest envy degree decrease, or
                { (c) highest transaction profits
            Accepts offer;
            Receives payments and updates local view;
```

According to analysis in Sect. 3.4.2, resource allocation in the community cloud evolves towards efficient and fair state when tenants pay initial equitability φ^0 and GUPF in BuMRT. However, when budget limitation presents, tenants do not always abide by these routine payments. Therefore, we are interested in investigating the transition of resource allocation states, when tenants pay different amounts as long as the amounts fall in the range of (3.5). In this section, we propose a series of heuristic-based BaMRTs. These protocols confine the trading activities of each tenant to neighboring peers, allowing them to conduct local negotiations. However, they are different with each other in terms of trading selection criterion. The complete description of the proposed BaMRT protocols are illustrated in Protocol 1.

Tenants delegate trading controls to trading agents who perform two basic operations periodically: proposing trade and selecting offer. When proposing a trade, the agent simply selects the neighboring peer who he envies most as the trading partner. In order to quantify the matchmaking unfairness between pairwise trading partners, we use the following equation to define the envy degree on a particular hyperarc.

$$\sigma_{i,j} = \max\{U_i(\tilde{A}_i, \tilde{\theta}_i) - U_i(A_i, \theta_i), 0\}$$

The trading agent may select any payment amount within the rational range. A tenant can set up a predefined payment policy for the trading agent. For example, a conservative policy results in resource acquisition with low cost, while an aggressive policy helps funding peer tenants to conduct further trades, and might benefit more in return. We will evaluate different payment policies in the performance evaluation section. When multiple offers arrive, each trading agent needs to carefully evaluate trading decisions with a local view of the directed hypergraph model. This is especially important when offers conflict with each other since the resource can only be granted to one neighbor. In our design, each trading agent employs a hill climbing technique to negotiate resource trading with neighboring peers. The hill climbing algorithm is fast and effective in finding a local optimal matchmaking. The local optimal offer selection decision must be rational, as the payment amounts conforming to RD increase the overall social welfare (Proposition 1). In other words, if a trade occurs, the allocation efficiency is reinforced, and the corresponding envy relationship between the trading parties is eliminated.

We propose three versions of BaMRT in favor of different trading selection criterion. Each of them follows different paths to reach the local minimum. The first version labeled as **Valuation oriented BaMRT (V-BaMRT)**, let trading agents select trades with the highest social welfare gain. In the second version, each agent selects the neighboring peer who he envies most as the trading partner. We label this version of BaMRT as **Envy oriented BaMRT (E-BaMRT)**. Finally, we propose **Profit oriented BaMRT (P-BaMRT)**, in which agents select offers that will bring in the highest transaction profits (defined as the difference of payment and gained valuation). These protocols work similarly to BuMRT except that they do not require message broadcasting to redistribute social wealth within the community.

3.6 Performance Evaluation

In this section we investigate the performance of the proposed protocols through three different sets of simulations. First, we implement BuMRT and validate its achievable efficiency and fairness. In the second set of simulations, three versions of BaMRT presented in Protocol 1 are compared in various norms. Finally, we evaluate the performance impact of different payment selection policy and initial budget settings for BaMRT.

3.6.1 Simulation Settings

We instantiate the matchmaking framework to a generalized distributed computing environment, and implement the resource trading protocols using SimGrid [8]. The core scheduling and communication functions are implemented using the application-level simulation interfaces provided by the MSG module of SimGrid.

A community cloud platform with 20 computational nodes (tenants) is simulated. We also creates 800 synthetic task units (resources). To create a heterogeneous platform, we assign different computational and networking settings to the computational nodes. As such the same task unit presents different values to different nodes. In SimGrid, this information is encapsulated in separate XML files. Node i's satisfaction of its current allocation is quantified by a concave valuation function $V_i(\cdot)$, where $V_i(x)$ defines the utility of node i obtaining x tasks [23]. The concavity assumption indicates that the marginal valuation diminishes when the allocation increases. Specifically, we use the following concave function to represent valuation,

$$V_i(x) = c \cdot x^r,$$

where the constant coefficient c is set to 10.0, and r is randomly generated in the range of $(0.2, 0.6)$.

We primarily use four metrics to evaluate the performance of the proposed protocols. First, we use *social welfare* to quantify the allocation efficiency. Second, in order to validate fairness, the *total envy degree* amongst all nodes is recorded after each transaction. In addition, two nodes that envy each other form an envious pair. The total number of envious pairs is also counted throughout the negotiation process. Finally, we measure *system profit* as an indication of system's side utility. For each transaction, the profit earned is the difference of buyer's valuation and the associated payment amount. The system profit is thus defined as the cumulative profit earned in all transactions.

3.6.2 Evaluation of BuMRT

In the first set of simulations, nodes negotiate with each other using BuMRT until convergence is reached. The results are plotted in Fig. 3.3. At the start of each simulation, 800 task units are randomly mapped to 20 nodes. We generate three topology profiles representing different network configurations. The first topology profile (labeled as "fully connected") describes a fully connected mesh network, and the rest profiles describe two relatively sparse network topologies. The fully connected topology has a total node degree of $20 \times 19 = 380$. The node degrees of the other two profiles are normalized relative to the fully connected profile. We use these normalized values, 0.45 and 0.72, to represent the connectivity of both profiles. In order to validate efficiency, we also implement a self-adaptive auction algorithm [34] that achieves maximum social welfare when tasks are allocated. This result, labeled as "optimal" in Fig. 3.3a, defines the global optimal social welfare. From Fig. 3.3a, we observe that in all topology profiles, the overall social welfare increases all the time and converges after around 24 transactions. In addition, for the fully connected network, the final allocation achieves the maximum social welfare when converges. Figure 3.3b, c show that all simulations converge to fair state where all envy relations are eliminated. Note that after each

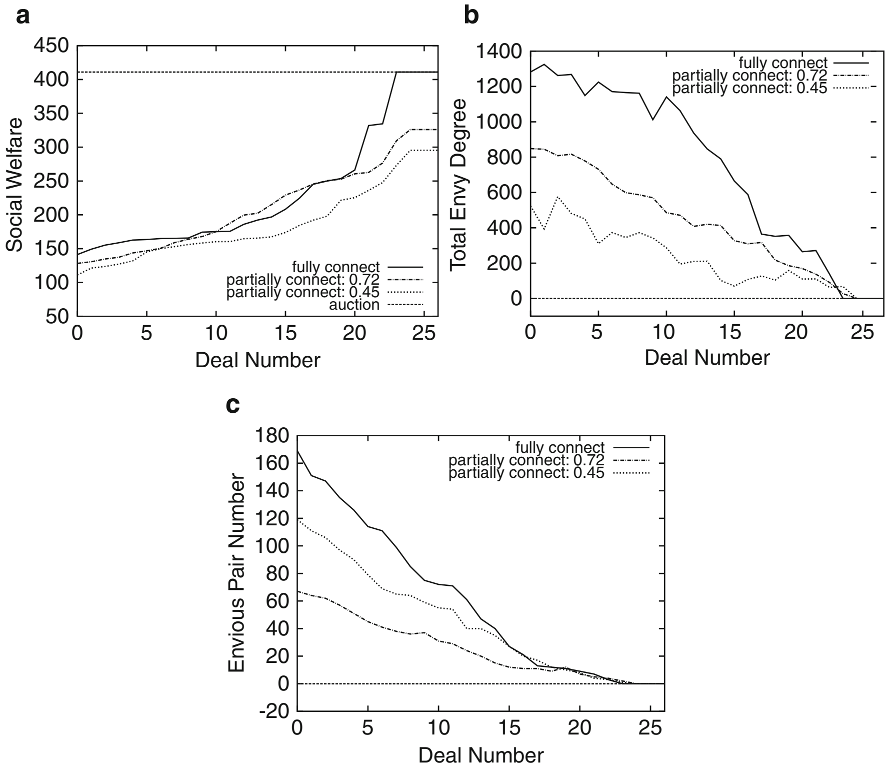

Fig. 3.3 Performance evaluation for budget-unaware case. (**a**) Measurement of efficiency; (**b**) measurement of fairness: envy degree; (**c**) measurement of fairness: envious pair

transaction, both envy degree and envious pair number do not necessarily decrease. This can be explained as follows: although the overall unfairness will be eliminated eventually, each single transaction only eliminates envy between the two trading partners, but may create envy relationship between other pairs. Another interesting observation for Fig. 3.3 is that the initial matchmaking unfairness is closely related to the network connection degree. This is because envy relation is more likely to present if more nodes are connected. Moreover, more connected network also means more opportunities for tasks to be assigned to nodes who value them more. Therefore, the achievable local efficiency is more likely to increase as the network becomes more connected.

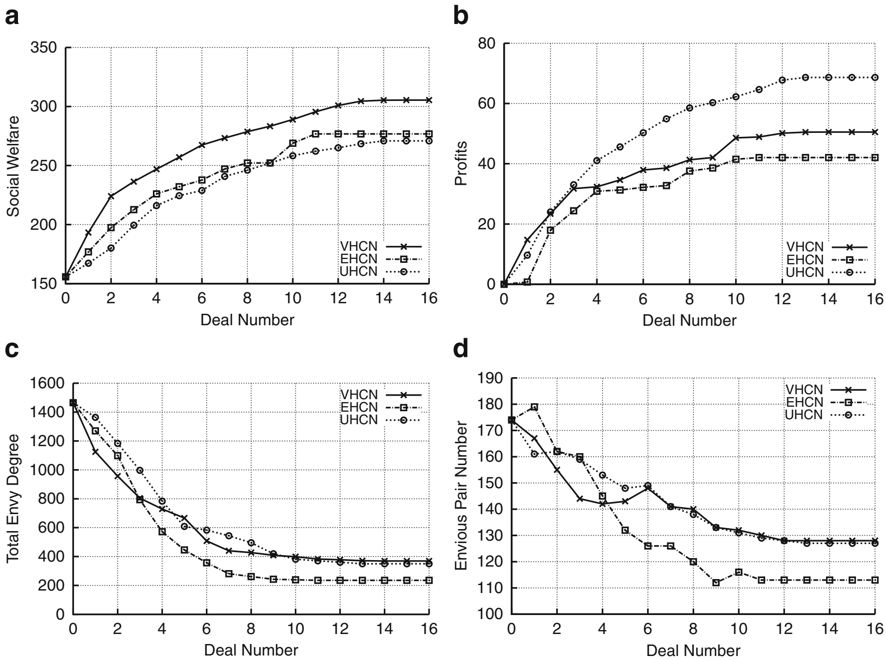

Fig. 3.4 Performance evaluation for budget-aware case. (**a**) Efficiency improvement; (**b**) profits gain; (**c**) fairness improvement: envy degree; (**d**) fairness improvement: envious pair

3.6.3 Evaluation of BaMRT

Next, we add budget limitation to each node and compare the performance of different versions of BaMRT presented in Sect. 3.5. The node and the task unit number are set to be 20 and 800 respectively. Based on the analysis of the average transaction payment range, we assign each node an initial budget of 100. When a transaction is completed, the node who makes payment will deduct the corresponding amount from its balance. Conversely, its trade partner will add the same amount to its balance. For fair comparison, all simulations are conducted using the same setting (valuation functions and initial allocation). All simulations use a same fully-connected network. The comparison results are exhibited in Fig. 3.4. From these results, we draw the conclusion that the performance of each protocol is primarily influenced by the offer selection strategy. In V-BaMRT, the offer brings the most social welfare growth is selected. Therefore in Fig. 3.4a we observe that V-BaMRT leads to the highest local efficiency when converged. Similarly, Fig. 3.4c, d show that E-BuMRT performs better in promoting fairness. And not surprisingly, the overall profits gain is in favor of U-BuMRT, as shown in Fig. 3.4b.

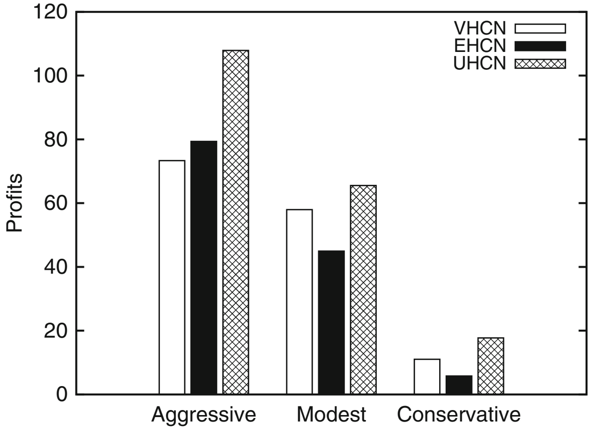

Fig. 3.5 Impact of different payment selection strategies for budget-aware case

3.6.4 Sensitivity Analysis

In this section, we investigate the impact of different payment selection strategies and initial budget settings. The configuration parameters are kept the same as in Sect. 3.6.3.

As analyzed in Sect. 3.5.2, each tenant can set up arbitrary payment policy for the trading agent. A conservative policy results in resource acquisition with low cost, while an aggressive policy helps funding other tenants to conduct more trading activities. Which policy gives better result depends on the offer selection strategies and initial budget distribution. We modify the simulation code to let each node select payment amount within the allowed range deterministically. Specifically, let the payment selection range be $(low, high)$, we devise three deterministic payment selection strategies for evaluation:

- Aggressive: $payment = low + 0.75 \times (high - low)$
- Modest: $payment = low + 0.5 \times (high - low)$
- Conservative: $payment = low + 0.25 \times (high - low)$

We compare the aggregate profits of the system in Fig. 3.5. Each value is the average result of 20 simulation runs. The result suggests that more aggressive bidding behavior will result in higher system profits at convergence. This can be explained that if all nodes offer higher at each deal, more nodes will get funded that lead to more transactions. As a result, the micro-economy of the small computing community is boosted.

Finally, we alter the initial budget assignment and measure its impact to the system envy degree. Taking initial budget of 100 to be the base case (marked as "1X"), the startup fund for each node is altered from 0.5 to 2 times of 100. Again we average the result of 20 simulation runs. The comparison is visualized in Fig. 3.6. We observe that for the case of abundant initial fund assignment, the convergence value is close to that achieved by BuMRT. When the initial budget reaches 200,

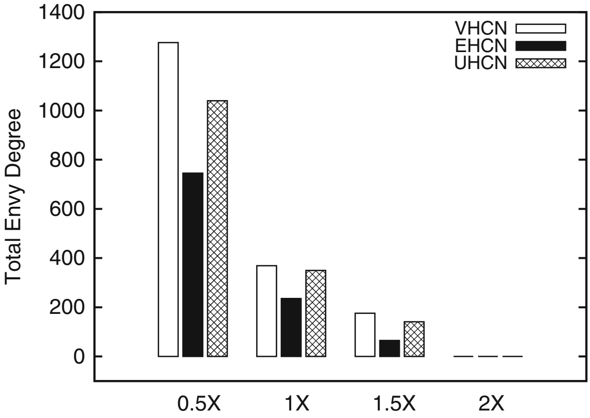

Fig. 3.6 Impact of initial budget assignment for budget-aware case

all protocols converge to an envy degree of 0 as if there are no budget constraint. On the contrary, for a poorly funded computing community, the trading activities are more likely to freeze due to lack of budget, resulting in potential longer convergence time and higher degree of unfairness.

References

1. Abramson, D., Buyya, R., Giddy, J.: A computational economy for grid computing and its implementation in the nimrod-g resource broker. Future Gener. Comput. Syst. **18**, 1061–1074 (2002)
2. Berlt, K., Moura, E.S.D., Carvalho, A.L.D.C., Cristo, M., Ziviani, N., Couto, T.: A hypergraph model for computing page reputation on web collections. In: Brazilian Symposium on Databases, pp. 35–49 (2007)
3. Boche, H., Naik, S., Alpcan, T.: Characterization of non-manipulable and pareto optimal resource allocation strategies for interference coupled wireless systems. In: Proceedings of the 29th conference on Information communications (INFOCOM 2010), pp. 2417–2425 (2010)
4. Brams, S.J., Taylor, A.D.: Fair division: from cake-cutting to dispute resolution. Cambridge University Press (1996)
5. Briscoe, G., Marinos, A.: Digital ecosystems in the clouds: Towards community cloud computing. In: 3rd IEEE International Conference on Digital Ecosystems and Technologies (DEST'09)., pp. 103–108 (2009)
6. Buyya, R., Abramson, D., Venugopal, S.: The grid economy. Proceedings of the IEEE **93**(3), 698–714 (2005)
7. Buyya, R., Pandey, S., Vecchiola, C.: Cloudbus toolkit for market-oriented cloud computing. In: Proceedings of the 1st International Conference on Cloud Computing (CloudCom'09), pp. 24–44 (2009)
8. Casanova, H., Legrand, A., Quinson, M.: SimGrid: a Generic Framework for Large-Scale Distributed Experiments. In: 10th IEEE International Conference on Computer Modeling and Simulation (UKSIM 2008) (2008)
9. Çatalyürek, U., Aykanat, C.: Hypergraph-partitioning-based decomposition for parallel sparse-matrix vector multiplication. IEEE Trans. Parallel Distrib. Syst. **10**(7), 673–693 (1999)

10. Chevaleyre, Y., Endriss, U., Estivie, S., Maudet, N.: Reaching envy-free states in distributed negotiation settings. In: Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI 2007), pp. 1239–1244 (2007)
11. Chevaleyre, Y., Endriss, U., Maudet, N.: Allocating goods on a graph to eliminate envy. In: Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI 2007), pp. 700–705 (2007)
12. Cooper, B., Garcia-Molina, H.: Peer-to-peer resource trading in a reliable distributed system. In: the First International Workshop on Peer-to-Peer Systems (IPTPS 2001), pp. 319–327 (2002)
13. di Costanzo, A., de Assuncao, M.D., Buyya, R.: Harnessing cloud technologies for a virtualized distributed computing infrastructure. IEEE Internet Computing **13**, 24–33 (2009)
14. Edmonds, J., Pruhs, K.: Balanced allocations of cake. In: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), pp. 623–634 (2006)
15. Estivie, S., Chevaleyre, Y., Endriss, U., Maudet, N.: How equitable is rational negotiation? In: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems (AAMAS 2006), pp. 866–873 (2006)
16. Eucalyptus Community Cloud. http://www.eucalyptus.com/eucalyptus-cloud/community-cloud
17. Gradwell, P.: Distributed combinatorial resource scheduling. In: 1st International Workshop on Smart Grid Technologies (2005)
18. Federal Community Cloud. http://www-304.ibm.com/industries/publicsector/us/en/contentemplate1/!!/xmlid=207581
19. Krishnamoorthy, S., Çatalyürek, U., Nieplocha, J., Rountev, A., Sadayappan, P.: Hypergraph partitioning for automatic memory hierarchy management. In: Proceedings of the 2006 ACM/IEEE conference on Supercomputing (SC 2006) (2006)
20. Kwok, Y.K., Hwang, K., Song, S.: Selfish grids: Game-theoretic modeling and nas/psa benchmark evaluation. IEEE Trans. Parallel Distrib. Syst. **18**, 621–636 (2007)
21. Lai, K., Rasmusson, L., Adar, E., Zhang, L., Huberman, B.A.: Tycoon: An implementation of a distributed, market-based resource allocation system. Multiagent Grid Syst. **1**, 169–182 (2005)
22. Laoutaris, N., Poplawski, L.J., Rajaraman, R., Sundaram, R., Teng, S.H.: Bounded budget connection (bbc) games or how to make friends and influence people, on a budget. In: Proceedings of the twenty-seventh ACM symposium on Principles of distributed computing (PODC 2008), pp. 165–174 (2008)
23. Ma, R.T., Chiu, D.M., Lui, J.C., Misra, V., Rubenstein, D.: On resource management for cloud users: A generalized kelly mechanism approach. Tech. rep., Electrical Engineering (2010)
24. Marinos, A., Briscoe, G.: Community cloud computing. CoRR **abs/0907.2485** (2009)
25. Pandey, S., Wu, L., Guru, S.M., Buyya, R.: A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In: Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA 2010), pp. 400–407 (2010)
26. Perez, J., Germain-Renaud, C., Kégl, B., Loomis, C.: Multi-objective reinforcement learning for responsive grids. Journal of Grid Computing **8**, 473–492 (2010)
27. Regev, O., Nisan, N.: The POPCORN market. online markets for computational resources. Decision Support Systems **28**(1–2), 177–189 (2000)
28. Rzadca, K., Trystram, D., Wierzbicki, A.: Fair game-theoretic resource management in dedicated grids. In: Proceedings of the Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2007), pp. 343–350 (2007)
29. Sarkar, S., Sivarajan, K.: Hypergraph models for cellular mobile communication systems. IEEE Transactions on Vehicular Technology **47**(2), 460–471 (1998)
30. Song, B., Hassan, M., Huh, E.N.: A novel cloud market infrastructure for trading service. In: Proceedings of the 2009 International Conference on Computational Science and Its Applications (ICCSA 2009), pp. 44–50 (2009)
31. Turner, D.A., Ross, K.W.: A Lightweight Currency Paradigm for the P2P Resource Market. In: Seventh International Conference on Electronic Commerce Research (2004)

32. Wolski, R., Plank, J.S., Brevik, J., Bryan, T.: G-commerce: Market formulations controlling resource allocation on the computational grid. In: Proceedings of the 15th International Parallel & Distributed Processing Symposium (IPDPS 2001), pp. 46–53 (2001)
33. Zhao, H.: Exploring Cost-Effective Resource Management Strategies in the Age of Utility Computing. Ph.D. thesis, University of Florida, Gainesville, FL, USA (2013)
34. Zhao, H., Li, X.: Efficient grid task-bundle allocation using bargaining based self-adaptive auction. In: Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2009), vol. 0, pp. 4–11 (2009)