Hypergraph-based Task-Bundle Scheduling Towards Efficiency and Fairness in Heterogeneous Distributed Systems

Han Zhao, Xinxin Liu, Xiaolin Li
Scalable Software Systems Laboratory
Department of Computer Science
Oklahoma State University
Stillwater, OK 74078, USA
Email: {haz, xinxin, xiaolin}@cs.okstate.edu

***Abstract*—This paper investigates scheduling loosely coupled task-bundles in highly heterogeneous distributed systems. Two allocation quality metrics are used in pay-per-service distributed applications: efficiency in terms of social welfare, and fairness in terms of envy-freeness. The first contribution of this work is that we build a unified hypergraph scheduling model under which efficiency and fairness are compatible with each other. Second, in the scenario of budget-unawareness, we formulate a strategic algorithm design for distributed negotiations among autonomous self-interested computing peers and prove its convergence to complete local efficiency and envy-freeness. Third, we add budget limitation to the allocation problem and propose a class of hill-climbing heuristics in favor of different performance metrics. Finally we conduct extensive simulations to validate the performance of all the proposed algorithms. The results show that the decentralized hypergraph scheduling method is scalable, and yields desired allocation performance in various scenarios.**

***Keywords*-task scheduling; distributed systems; envy-free allocation; hypergraph**

I. Introduction

The ever-increasing demand for computing power motivates recent advances in multi-core technologies and high speed networking, rendering a variety of massively parallel and distributed computing platforms, such as p2p, grid, and emerging cloud computing in both academic and industrial communities. Although the computing power grows by leaps and bounds, the development of efficient scheduling mechanisms need to keep up with the pace. As in modern applications, autonomous and geographically distributed computers are not subject to central coordination, making resource and task scheduling decisions extremely complicated. In addition, multiple objectives for scheduling mechanism design pose more challenges. For example, independent task-bundle allocation problem can be evaluated in terms of efficiency and fairness respectively, but there lacks a unified model taking both criteria into account. We fill this gap by introducing a task-bundle allocation model based on directed hypergraph. The proposed model operates on highly autonomous computers in heterogeneous distributed environment, and strives to achieve both efficient and fair allocation within possible cost constraints.

The work presented in this paper is supported in part by National Science Foundation (grants OCI-0904938 and CNS-0709329).

In our previous work [1], we proposed efficient task-bundle allocation algorithm using self-adaptive auctions. To precisely capture the real time scenario and mitigate scheduling complexity we adopted a hierarchical parallel system model, which addresses enhanced site cooperation among different organizations. In such a model the system is no longer a flat orchestration of geographically scattered computers, but consists of schedulers at different administrative levels and highly dynamic individual computing peers. User requests are represented by independent tasks known as BoT (Bags-of-Tasks) [2], and are fulfilled by computing peers who contribute resources in exchange for either real or virtual monetary values. In that sense, although task allocation problem is NP-hard with no provable optimal solution exists, we circumvent the central scheduling overheads by formulating a distributed market-oriented framework where individual rational agents negotiate iteratively for desired results. In this paper we further jointly address two economically significant metrics for allocation: efficiency and fairness. The allocation efficiency is expressed as the overall valuations of the society, and fairness is interpreted as envy-freeness across all computing peers. We show how these two seemingly conflicting requirements are integrated using a directed hypergraph model. Moreover we investigate scheduling strategies for individual computing peer with and without budget constraint, and evaluate the scheduling performance through extensive simulations using SimGrid.

The *contribution* of this paper is four-fold: 1) we formulate a novel directed hypergraph model addressing both allocation efficiency and fairness in the market-oriented task-bundle allocation; 2) inspired by [3] and [4], we propose and analyze scheduling mechanisms for budget-free scenario and develop negotiation strategy for self-interested computers using the proposed hypergraph model; 3) we investigate negotiation strategies for individual computing peers on the presence of budget constraint in favor of efficiency, fairness and profits respectively; 4) we perform extensive simulations to verify the performance of the proposed allocation mechanisms.

The rest of the paper is organized as follows. Section II presents an overview of related work. In Section III we describe the target problem and formulate a unified hypergraph model for allocation mechanism design. In Section IV we develop a distributed negotiation strategy to achieve locally efficient and fair allocation under the directed hypergraph model. Section V further investigates allocation strategies with limited budget, and presents a series of negotiation strategies (HCN) based on hill climbing in favor of efficiency, fairness and profits respectively. Section VI exhibits extensive simulation results and finally, Section VII concludes the paper.

II. Related Work

In recent years, the impact of economic and game theory has been widely investigated in context of task and resource scheduling in parallel and distributed systems [5], [6], [7], [8]. This trend is largely due to the following observations: design similarity of market operating mechanisms and scheduling principles; and role similarity of rational economic individuals and egocentric heterogeneous computers. As a result game theoretic methods become popular for modeling non-cooperative, large-scale distributed environments. In a series of publications [9], [10], [11] Archer et al. highlighted and rectified drawbacks of the Vickrey-Clarke-Groves (VCG) mechanism [12], [13], [14], and proposed mechanism design approaches promoting honest strategy of rational agents. In [15] the authors studied how to measure the economic value of heterogeneous resources. For fair resource scheduling, Kim and Buyya [16] proposed a heuristic algorithm for job queuing model on hierarchical multi-organizational grids. The method is effective in reducing the total job waiting time. In [17], the authors presented a max-min fair resource sharing mechanism. Similar to fair bandwidth sharing in network research, when resource congestion happens, CPU rate assigned to the task is fairly reduced so that each user gets its share proportional to the user's weight. In short, existing research mainly focus on overall system performance such as average completion and makespan time, and relatively few authors address individual economic performance, which is equally important in scheduling [18]. For example, in the well-known cake cutting problem [19] resources are fairly allocated to self-interested agents such that everyone achieves the maximum contentment of its share, called envy-freeness. In this paper, we address the fair allocation issue extending previously developed socially efficient allocation framework [1]. A more related work proposed on similar system model is found in [20]. Using game theory, the authors conclude that for fair and feasible scheduling on global scale computational grid, a strong community control is required. However the concepts of scheduling efficiency and fairness are derived from axiomatic theory of equity. To the best of our knowledge, we are the first to integrate both allocation efficiency in terms of overall social welfare, and envy-freeness among selfish individuals into one unified scheduling model.

A hypergraph is a mathematical extension of conventional graph. It is different from a graph in that one edge (called a hyperedge) can connect an arbitrary set of vertices rather than only two vertices. Over the past few years, various hypergraph partitioning heuristics based on the Kernighan-Lin [21] method have been proposed. They are fast and effectively achieve maximum parallel efficiency while minimizing costs associated with cut-net, resulting in considerable applications of hypergraph model in multiple research domains such as wireless networks [22], [23], VLSI circuit design [24], memory management [25] and automated theorem [26]. For large scale scientific computing, Çatalyürek and Aykanat [27] pioneered in developing heuristic methods for mapping repeated sparse matrix-vector computations to multicomputers. Compared to graph models the hypergraph based method significantly reduces communication overheads while achieving drastically improved mapping results. A multilevel partitioning algorithm was further developed [28] and was applied to periodic dynamic workload balancing in adaptive scientific computations [29].

III. A Hypergraph-based Allocation Model

In this section, we present a novel directed hypergraph model for task-bundle allocation in heterogeneous distributed systems. We begin with basic notations and assumptions and describe the task-bundle allocation problem. For more details of system and application model, we refer readers to our previous work [1]. We then formulate a unified hypergraph model which integrates efficiency and fairness criteria using a three-dimensional task-peer matrix.

A. Preliminary Definitions

Let $\mathbb{P} = \{p_1...p_n\}$ denote a finite set of heterogeneous geographically distributed computing peers (peers for short) connected via network, and let $\mathbb{T} = \{t_1...t_m\}$ be a set of independent parallel tasks with identical attributes; all tasks have the same computational and communication size. Also we have $m > n$ for allocation. An allocation of tasks is defined as a mapping function $A(\mathbb{T}) \to \mathbb{P}$, more specifically:

Definition 1. Allocation: An allocation $A = \{A_1, A_2, ..., A_n\}$ is a mapping of tasks to peers $A(\mathbb{T}) \to \mathbb{P}$, which must satisfies: $A_i \bigcap A_j = \emptyset$, and $\bigcup A_i = A$.

We assume a private value model, indicating that each peer is mutually blind to each other and evaluates its task-bundle independently. The **valuation** of peer p_i for current allocation A_i is represented as $V_i(A_i)$, and follows the constraints of $V_i(\emptyset) = 0$ and $V_i(A_i) \geq V_i(A_i^*)$ for all $A_i \supseteq A_i^*$. Moreover, valuation is modular, that is: $V_i(A_i \cup A_j) = V_i(A_i) + V_i(A_j) - V_i(A_i \cap A_j)$ for all bundles $A_i, A_j \subseteq A$. The overall system valuation of all tasks over current allocation is called **social welfare**: $\omega = \sum_{i=1}^{n} V_i(A_i)$, which defines the first allocation metric:

Definition 2. Efficient Allocation: Let Γ be the set of all possible allocations, an efficient allocation is an allocation $A = \{A_1, A_2, ..., A_n\}$ which maximizes the social welfare: $\omega_{max} = \max_{A \in \Gamma} \sum_{p_i \in \mathbb{P}} V_i(A_i)$.

In order to acquire tasks from another peer, monetary compensation is involved in the exchange process. A payment function $\varphi_{i,j}$, represents such compensation amount p_i pays to p_j. If $\varphi_{i,j}$ is negative, then p_i receives money from p_j. Each peer keeps a record of its payment history, and p_i's payment balance is defined as the summation of its withdrawals and deposits at all deals: $\theta_i = \sum \varphi_i$. For each task migration between peers (we call it a deal), we assume all peers are self-interested, and the deal must be rational. Formally, suppose after task migration happens, allocation to p_i becomes $\tilde{A}_i$, a deal is a **Rational Deal (RD)** for p_i iff $V_i(\tilde{A}_i) - V_i(A_i) \geq \varphi_{i,j}$ for all $p_i \in \mathbb{P}$. Suppose after a series of RDs p_i obtains allocation A_i° with payment balance θ_i, the **utility** of p_i is given as $U_i(A_i^\circ) = V_i(A_i^\circ) - \theta_i$.

Now with all the notations and hypotheses, we define the second allocation criterion – fairness. In this paper fairness is interpreted as envy-freeness [30], indicating that no peer would get better off by exchanging its current allocation with another peer using an RD. Here we adopt the definition [3] taking both valuation and payment balance into consideration.

Definition 3. Fair Allocation: Let Γ be the set of all possible allocations, an allocation is called a fair allocation iff there exists $A = \{A_1, A_2, ..., A_n\} \in \Gamma$ such that: a) $\forall p_i, p_j \in \mathbb{P}, p_i$ and p_j has direct network connection; b) $U_i(A_i) \geq U_i(A_j)$.

In modern parallel systems consisting of geographically distributed autonomous sites, a budget transfer mechanism may be employed to enforce incentives for community control [31]. For example, in p2p and social network, some form of digital cash, or numerical reputations representing trust may be used for rewarding and punishing certain actions. In this work we analyze negotiation strategies with and without budget limitations. From now on we label the scenario without budget constraint **budget-unaware**, and refer to the later scenario as **budget-aware**.

B. Problem Statement

Given any initial allocation A^0 projecting $\mathbb{T}$ onto $\mathbb{P}$, the goal of this research is to develop a task exchange strategy for autonomous peers in order to achieve efficiency and fairness as much as possible. We first study the allocation strategy for the budget-unaware scenario. Next we add complexity to cope with budget constraint and investigate possible payment and offer selection policies for self-interested peers.

C. A Directed Hypergraph Model For Task-Bundle Allocation

To model the task-bundle allocation problem in hypergraph, multiple objectives should be unified in a way that vertices represent the allocated tasks associated with valuations on current allocation, and edges reflect envy relations across peers. Both objectives are characterized in matrices, named **Allocation Matrix (AM)**, and **Envy Matrix (EM)** respectively. The allocation matrix is an $m \times n$ matrix that represents current allocation situation. More specifically:

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

On the other hand, the envy matrix stands for envy relationship across all peers. According to definition 3 for fair allocation, peer p_i **envies** p_j only when (a) p_j is in p_i's neighboring set in physical topology, which means p_i and p_j can negotiate directly for task-bundle exchange, and (b) if p_i and p_j swaps allocation, p_i will get increased utility. At the beginning of each negotiation, each peer will check its neighboring peers and figures out its **envy set** Δ_i. The algorithm of envy set update for each peer is described in Algorithm 1.

Algorithm 1 Envy Set Update for p_i

```
for all p_i ∈ ℙ do
    /*scanning neighboring set*/
    for all p_j belongs to the neighboring set do
        /*Finds current task-bundle on p_j*/
        for all α_{k,j} = 1 do
            if V_i(t_k) > V_j(t_k) then
                • If no hyper-arc existed from p_i to
                  p_j, builds a hyper-arc from p_i to the
                  hyper-node whose T_id = k
                • Adds p_j to p_i's envy set if it is not
                  there
                • Adds t_k to p_i's wishing list
            end if
        end for
    end for
end for
```

Once the envy set update for each peer is done, the envy matrix is confirmed. It is defined to represent envy relationship across all peers. In particular we have:

$$\varepsilon_{i,j} = \begin{cases} 1 & p_j \text{ is in } p_i\text{'s envy set} \\ 0 & \text{otherwise} \end{cases}$$

Suppose the network topology together with envy are depicted as a envy graph. Figure 1 displays an example of transition from envy graph to envy matrix. Note that the envy representation is directional and according to the definition of EM the diagonal elements are 0.

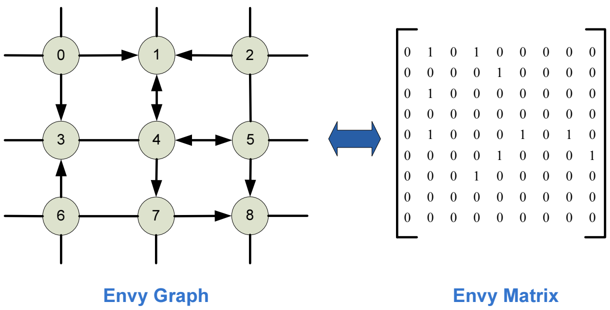


Figure 1. Transition from envy graph to envy matrix. In the envy graph, each circle stands for a computing peer and they are connected using a mesh structure. The directed arrow from one peer to another represents envy relation.

Each of the matrices defined above merely considers one aspect of allocation properties (current allocation valuation and unfairness respectively). Here we present a unified hypergraph model taking both aspects into account. A directed hypergraph $H = (X, E)$ is composed of a finite non-empty set X of hyper-nodes and a finite non-empty set $E \subseteq 2^X$ of hyper-arcs. A hyper-arc is directional, and unlike traditional graph model, a hyper-arc is able to link arbitrary number of hyper-nodes simultaneously. In the proposed directed hypergraph model, we build a three-dimensional matrix integrating both AM and EM. The information of task allocation and envy relation are mapped onto an $m \times n \times n$ matrix, as shown in Figure 2. Specifically, a hyper-node (Equation 1) is represented in a three-tuple $(T_{id}, P_{cur}, \Lambda)$. T_{id} represents the task ID, P_{cur} infers the peer ID currently holding task T_{id}, and Λ is P_{cur}'s envy set established in Algorithm 1. A hyper-arc is initiated by any peer who has a nonempty envy set, say p_i, and connects a set of hyper-nodes who are currently allocated to peers belonging to p_i's envy set Δ_i, and lower valued in P_{cur} than in P_i (Equation 2). If $p_j \in \Delta_i$, We say p_i envies p_j on hyper-node x_i iff $V_i(T_{id} \in x_i) > V_j(T_{id} \in x_i)$.

Hypernode definition: A hypernode is a three-tuple:

$$\begin{aligned} & x = (T_{id}, P_{cur}, \Lambda) \in X \\ \text{s.t. } & T_{id} \in \mathbb{T} \\ & P_{cur} \in \mathbb{P}, \text{and } \alpha_{i,P_{cur}} = 1 \\ & \Lambda = \{\varepsilon_{P_{cur},k} = 1 | 1 \leq k \leq n\} \end{aligned} \tag{1}$$

Hyper-arc definition: A hyper-arc $e \in E$ connects peer p_i to x, $p_i \to x$ iff:

$$\begin{aligned} & \{P_{cur} | P_{cur} \in x\} \in \{\Lambda | \Lambda \in x\} \\ & V_i(T_{id} \in x) > V_{cur}(T_{id} \in x) \end{aligned} \tag{2}$$

The hyper-arc definition implies possible business chances for p_i, which is helpful for building the negotiation algorithms later on. In the directed hypergraph model each hyper-node has an associated weight (valuation of current task-bundle), and envy relations reflected in interconnected hyper-arcs with other peers. The problem becomes finding a negotiation strategy for each peer to reallocate hyper-nodes, which eliminates degree of unfairness on hyper-arcs as many as possible while improving overall weights of the entire system.

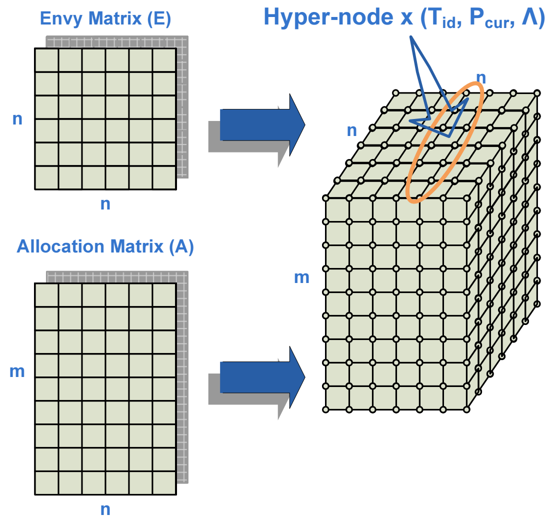


Figure 2. Hypergraph Allocation Model. The proposed directed hypergraph model derives from an $m \times n \times n$ matrix. A hypergraph vertex is represented by nonzero elements on the column of EM matrix (surface). Each processor launches a hyper-arc (not shown) pointing to the hyper-nodes whose corresponding task-bundle is more appreciated on that processor, but allocated to others.

IV. Allocation without Budget Constraint

In this section we develop a distributed negotiation strategy for the budget-unaware scenario. Figure 3 displays an example of task-bundle exchange between p_i and p_j. Suppose under current allocation A, p_j is in p_i's envy set. Therefore p_i will compensate for p_j with payment $\varphi_{i,j}$ in order to get task-bundle $\mathbb{T}^* \in \mathbb{T}$. In our task-bundle allocation settings, each peer negotiates with its neighbors using Rational Deals (RD), and requests task-bundle represented by hyper-nodes through hyper-arcs. Intuition on RD indicates that they beneficially contribute to push task-bundle to agents who value them more. Therefore, any sequence of RD will reach efficient allocation based on the following observations: 1) RD will increase system valuation according to the definition; and 2) if no more RD could happen, then the allocation must reach its maximum social welfare. Since we are interested in designing negotiation strategy within the directed hypergraph model, we would like to explore the influence of RD for hyper-node flow through hyper-arcs (task-bundle reallocation). Recall that in Section III we assume valuations are modular, which is necessary to get the following convergence result [32]:

Proposition 1. Convergence to efficiency: If valuations of each peer are modular, then any sequence of RD involving any number of task exchange will eventually yield to socially efficient allocation.

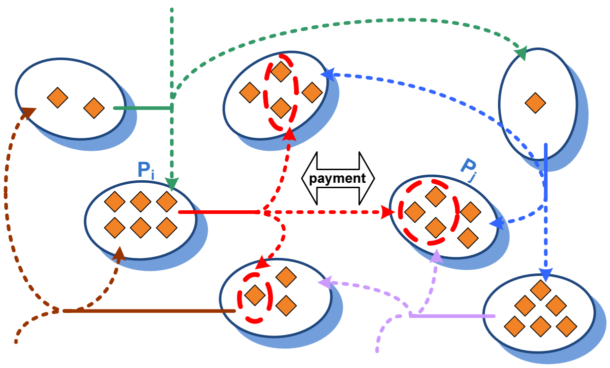


Figure 3. Example of task-bundle exchange between p_i and p_j. The ellipses represent computing peers while the diamonds represent tasks waiting to be allocated. Take p_i as an example, the red line going out of p_i stands for a hyper-arc. Therefore there are three peers in p_i's envy set including p_j. In p_j three tasks (hyper-nodes) have higher valuations in p_i and they are included in the red dashed circle. Hyper-nodes are moved from p_j to p_i and corresponding payments are paid by p_i.

Now suppose after the negotiation, allocation becomes $\tilde{A}$, we ask how much amount is appropriate to pay for each negotiation process as described in Figure 3 to get desired allocation. First note that payment should be rational, therefore the deal is bilaterally beneficial to both peers involved in the deal if:

$$\begin{aligned} V_i(\tilde{A}_i) - V_i(A_i) &\geq \varphi_{i,j} \\ V_j(\tilde{A}_j) - V_j(A_j) &\geq -\varphi_{i,j} \end{aligned} \tag{3}$$

Resolving Equation 3 makes payment function $\varphi_{i,j}$ falls into $[V_j(A_j) - V_j(\tilde{A}_j), V_i(\tilde{A}_i) - V_i(A_i)]$. Apparently a mutual beneficial deal at least does not decrease the overall social welfare ω.

A. Convergence Conditions

Chevaleyre et al. [3], [4] theoretically analyzed the conditions for convergence to efficiency and envy-freeness for multi-agent systems. One central conclusion is that efficiency and fairness are compatible in multi-agent negotiation framework. To support this conclusion, a proper payment function is selected to deal with the increased social surplus $\omega(\tilde{A}) - \omega(A)$ after each deal. In [3] a payment function called **Globally Uniform Payment Function (GUPF)** is proposed to facilitate the convergence. Suppose A and $\tilde{A}$ are allocations before and after the deal respectively, we have:

$$\textbf{GUPF: } \varphi_i = [V_i(\tilde{A}_i) - V_i(A_i)] - \frac{[\omega(\tilde{A}) - \omega(A)]}{n} \tag{4}$$

Equation 4 is defined as globally uniform because this payment is required for all peers. For peers not involved in the deal, $V_i(\tilde{A}_i) - V_i(A_i)$ equals to 0, so each of them receives an equal share of social surplus. Here we show that function GUPF falls into the derived bound of rational pay (Equation 3), thus GUPF and RD are compatible with each other.

Lemma 1. Suppose after one RD we select GUPF as the payment function for the deal between p_i and p_j, then GUPF must be rational for p_i and p_j.

Proof: It is obvious that social surplus at least does not decrease after the RD happens. Therefore $\omega(\tilde{A}) - \omega(A) \geq 0$. Next we prove that GUPF $\geq V_j(A_j) - V_j(\tilde{A}_j)$. According to the definition of social welfare we have:

$$\begin{aligned} \omega(\tilde{A}) - \omega(A) &= [\sum_{k \neq i,j} V_k + V_i(\tilde{A}_i) + V_j(\tilde{A}_j)] \\ &\quad - [\sum_{k \neq i,j} V_k + V_i(A_i) + V_j(A_j)] \\ &= [V_i(\tilde{A}_i) - V_i(A_i)] - [V_j(A_j) - V_j(\tilde{A}_j)] \end{aligned}$$

As a result, the following inequality stands:

$$[V_i(\tilde{A}_i) - V_i(A_i)] - [V_j(A_j) - V_j(\tilde{A}_j)] \geq \frac{\omega(\tilde{A}) - \omega(A)}{n}$$

Rearrange this inequality gives GUPF $\geq V_j(A_j) - V_j(\tilde{A}_j)$. Finally we have:

$$\text{GUPF} \in [V_j(A_j) - V_j(\tilde{A}_j), V_i(\tilde{A}_i) - V_i(A_i)].$$

■

In addition, to ensure that the final allocation can reach envy-freeness with RDs, an one-off payment called **initial equitability payment** is introduced [4] before all negotiations happen. Each agent pays the amount of its current bundle valuation for initial allocation, and receives an equal share of social welfare. The initial equitability payment for p_i is defined as: $\varphi^0 = V_i(A_i^0) - \frac{\omega(A^0)}{n}$. The authors conclude that suppose all valuations are modular and initial equitability payments have been paid by all peers, paying with GUPF at each negotiation process with any sequence will eventually lead to efficient and fair allocation [4]. However, in the multi-agent system one agent is able to negotiate with another without topology constraint. Here we are interested in how these conditions will influence our negotiation strategy design under the directed hypergraph model. The design and analysis are presented in the later parts of this section.

B. Negotiation Strategy Design

We propose a decentralized algorithm, in which negotiation process is conducted independently by each peer. After initial payments have been made at the start of the whole process, each peer performs two activities during the negotiation: making offer and selecting offer. For peer p_i, it makes offer by traversing each of its hyper-arcs and requests hyper-nodes representing envious task-bundle connected by that hyper-arc; at the same time it may receive multiple offers from its neighbors. With unlimited budget, at each time of negotiation each peer randomly picks an offer and makes announcement of the deal. After that transaction realized each peer calculates corresponding GUPF and updates the balance.

For better algorithm evaluation, we adopt the concept of envy degree defined in [4] to represent unfairness numerically:

Definition 4. Envy Degree: Envy Degree $\sigma_{i,j}$ characterizes extent of envy between p_i and p_j, specifically:

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

And the overall unfairness is the sum of envy degrees from all the hyper-arcs. Note that negotiation between p_i and p_j eliminates envy between them, but may create intensive envy with other peers, causing *oscillation* of the overall envy degree. Finally we define the termination condition as no more offers are available in the system. The complete procedure is shown in Algorithm 2.

Algorithm 2 Budget-unaware Negotiation Strategy

Establishes initial hypergraph
for all $p_i \in \mathbb{P}$ **do**
  Makes initial equitability payment φ^0
end for
while available offer number > 0 **do**
  for all $p_i \in \mathbb{P}$ **do**
    ——Offer Making Strategy——
    while p_i's envy set $\Delta_i \neq \emptyset$ **do**
      a) Randomly picks p_j from its hyper-arc
      b) Accesses hyper-node set through hyper-arc
      c) Makes offer
      if p_j accepts offer **then**
        • Finalizes hyper-nodes move
        • Removes p_j from hyper-arc
        • Reevaluates envy set Δ_i using Algorithm 1 and updates hypergraph
      end if
    end while
    ——Offer Selection Strategy——
    while offers on the table **do**
      a) Randomly selects offer and notifies its sender
      b) Finalizes hyper-nodes move
      c) Notifies all peers to pay GUPF
    end while
  end for
end while

C. Algorithm Analysis

We analyze utility variation of each peer after one deal happens. By using Algorithm 2 we have the following lemma for individual utility:

Lemma 2. If each peer pays initial equitablitiy payments at start and uses GUPF at each negotiation, then all peers have the same utility: $U_i(A_i) = \frac{\omega(A)}{n}$ at each allocation A.

Proof Sketch: First we show that balance θ_i of p_i remains invariant after each deal happens [4]. Apparently this is true for the base case, where equitability payment is used. Next suppose at allocation A, p_i's balance θ_i equals to $V_i(A_i) - \frac{\omega(A)}{n}$, then after allocation changes to $\tilde{A}_i$ with GUPF, adding GUPF: $[(V_i(\tilde{A}_i) - V_i(A_i)) - (\frac{\omega(\tilde{A})}{n} - \frac{\omega(A)}{n})]$ to p_i's balance at A leads to $\tilde{\theta}_i = V_i(\tilde{A}) - \frac{\omega(\tilde{A})}{n}$.

Now with this invariant, for each peer p_i, at allocation A its utility is computed as:

$$\begin{aligned} U_i(A_i) &= V_i(A_i) - \theta_i \\ &= V_i(A_i) - (V_i(A_i) - \frac{\omega(A)}{n}) \\ &= \frac{\omega(A)}{n} \end{aligned}$$

That is, after each deal every peer gets the utility of equal share of social welfare. ■

In economics and AI community, the concept of envy-freeness is commonly defined as free of valuation imbalances for all individual agents. That is, two peers connected via network are free of envy iff $V_i(A_i) \geq V_i(A_j)$ at final allocation. Now with Lemma 2 we show that defining fairness using accumulated utility (Definition 3) is actually compatible with this valuation-based fairness.

Theorem 1. *The valuation-based envy-freeness definition is compatible with its utility counterpart in Algorithm 2 (Definition 3), that is:* $(V_j(A_j) \geq V_i(A_j)) \simeq (U_i(A_i) \geq U_i(A_j))$.

proof: From Lemma 2 we know that balance of p_i at allocation A always equals to $V_i(A_i) - \frac{\omega(A)}{n}$. Therefore in budget-unaware case by taking balances into consideration, we actually have:

$$\begin{gathered} U_i(A_i) \geq U_i(A_j) \\ \Leftrightarrow \\ V_i(A_i) - (V_i(A_i) - \frac{\omega(A)}{n}) \geq V_i(A_j) - (V_j(A_j) - \frac{\omega(A)}{n}) \\ \Leftrightarrow \\ (V_j(A_j) \geq V_i(A_j)) \simeq (U_i(A_i) \geq U_i(A_j)) \end{gathered}$$

■

Next we answer the question of whether the maximum social welfare is attained after Algorithm 2 terminates. Unfortunately the globally efficient allocation does not necessarily exist with respect to partially connected system and we prove this by example:

Theorem 2. *For a partially connected distributed system, a globally efficient allocation may not be achieved after execution of Algorithm 2.*

Proof: We consider a system with three peers and one task initially assigned to p_1, and peers have different valuations on that task, as shown in Figure 4. Network links exist between $\{p_1, p_2\}$ and $\{p_1, p_3\}$. According to the procedure of envy set update described in Algorithm 1 both p_2 and p_3 envies p_1 on that task. However, in Algorithm 2, the offer arriving order is unpredictable and no restrictions are placed on offer selection order, p_1 may accept offer from p_3 and sends that task to p_3. As no link exists between p_2 and p_3, the allocation is final. This apparently contradicts the globally efficient allocation, where the task should be dispatched to p_2.

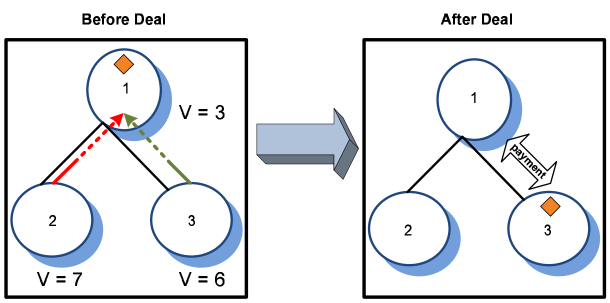


Figure 4. Proof to theorem 2: solid lines represent links and colored dashed lines represent hyper-arcs for envy

■

Although Algorithm 2 using hypergraph scheduling does not necessarily yield to globally efficient allocation, it does have the effect of pushing tasks all the way to its local center peer who appreciates them most. For final allocation achieved by Algorithm 2 we have the following definition defining its efficiency:

Definition 5. Complete Locally Efficient Allocation: A complete locally efficient allocation is an allocation such that for every peer, the allocation for the sub-topology consisting of that peer and its neighbors is efficient.

At last, we show that by using GUPF and initial equitability payment proposed in [4], our algorithm converges to the complete locally efficient allocation (Definition 5) and fairness (Definition 3) at the same time upon termination.

Theorem 3. *Upon termination, Algorithm 2 for budget-unaware scenario achieves complete locally efficient allocation and fairness at the same time.*

Proof: First we prove that when terminates, the result allocation $\dot{A}$ is complete locally efficient. Suppose this conclusion is not correct, then according to Algorithm 1 there must be at least one valid offer not being handled, which contradicts the fact that the algorithm has been terminated. Next we show that this locally efficient allocation is also fair. Recall that two peers can be involved in envy relation only if there exists a link between them. Now suppose p_i and p_j be any two agents connected by network. According to Definition 5 transferring p_j's allocation to p_i will not increase $\omega(\dot{A})$:

$$V_i(\dot{A}_i) + V_j(\dot{A}_j) \geq V_i(\dot{A}_i \cup \dot{A}_j)$$

As valuations are modular, $V_i(\dot{A}_i \cup \dot{A}_j) = V_i(\dot{A}_i) + V_i(\dot{A}_j) - V_i(\dot{A}_i \cap \dot{A}_j)$. Since $\dot{A}_i \cap \dot{A}_j = \emptyset$, we have:

$$V_j(\dot{A}_j) \geq V_i(\dot{A}_j)$$
$$\Leftrightarrow$$
$$V_i(\dot{A}_i) - (V_i(\dot{A}_i) - \frac{\omega(\dot{A})}{n}) \geq V_i(\dot{A}_j) - (V_j(\dot{A}_j) - \frac{\omega(\dot{A})}{n})$$

From Lemma 2 we know that $V_i(\dot{A}_i) - \frac{\omega(\dot{A})}{n}$ equals to θ_i and $V_j(\dot{A}_j) - \frac{\omega(\dot{A})}{n}$ equals to θ_j at final allocation $\dot{A}$. Therefore we have $U_i(\dot{A}_i) \geq U_i(\dot{A}_j)$ for every connected pair of computing peers. Specially, for a complete connected distributed system the allocation after executing Algorithm 2 is guaranteed to be globally efficient and fair. ■

V. Budget-aware Negotiation Strategy

In this section we add complexity to the task-bundle allocation problem with preset budget constraint on all peers. Let b_i be p_i's budget at current allocation A and b_i^0 be the initial budget, we formally define budget constraint for p_i as follows:

Definition 6. Budget Constraint: Budget constraint b_i expresses maximum amount p_i is able to offer at allocation A. Specifically:

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

With the limitation of budget, p_i will decide its payment $\varphi_{i,j}$ for making offer in the range of:

$$\varphi_{i,j} \in [V_j(A_j) - V_j(\tilde{A}_j), \min\{V_i(\tilde{A}_i) - V_i(A_i), b_i\}] \quad (5)$$

As payment is not arbitrary for each deal, Algorithm 2 presented in the previous section no longer fits. For instance, for extreme case in which initial budget b_i^0 equals to zero, p_i is not capable of making any payment, thus requiring initial equitability payment φ^0 at beginning for all peers is impossible. In this section we modify Algorithm 2 by proposing a local search negotiation strategy using hill climbing. It is fast and easy to implement, and effective in finding local optimal allocation using the directed hypergraph model. The foundation of the **Hill Climbing Negotiation (HCN)** is built upon the observation that, any payment function following rational exchange does no harm to the overall social welfare. Therefore at each negotiation step if a deal happens using a rational payment function subject to budget constraint (Equation 5), it at least does not impair the social efficiency. As such HCN strives to find local minimum of total envy degree in the system while maintaining non-decreasing efficiency.

In the budget-aware negotiation process, each peer checks its envious task-bundle and finds out envy set following Algorithm 1. When multiple offers arrive, the peer evaluates each offer by tentatively moving the envious task-bundle to the offer provider, and calculates corresponding system envy degree change after the tentative deal. If the overall envy degree does decrease then the offer is valid and marked as one potential selection. When there are multiple valid offers available we propose three versions of HCN in favor of valuation, envy degree and profit respectively. Each of them represents different paths to reach the local minimal, hence maximum reachable fairness under budget constraint. In **Valuation oriented HCN (V-HCN)** p_i selects the deal with the highest social welfare gain; in **Envy oriented HCN (E-HCN)** p_i picks the deal reducing most overall envy degree; and in **Utility oriented HCN (U-HCN)** the offer with the highest side utility that equals to the difference of payment and exchanged task-bundle valuation is selected. The complete description of HCN is illustrated in Algorithm 3.

Algorithm 3 Budget-aware Hill Climbing Negotiation: (a) Valuation Oriented (b) Envy Oriented (c) Utility Oriented

```
Establishes initial hypergraph
while there exists at least one offer makes Σ_{i=1}^n σ_i(Ã) − Σ_{i=1}^n σ_i(A) < 0 do
    for all p_i ∈ ℙ do
        ———Offer Making Strategy———
        while p_i's envy set ≠ ∅ do
            1) Sorts potential deals on hyper-arc according to envy degree
            2) Selects p_j with highest σ_{i,j}
            3) Randomly selects payment within
               [V_j(A_j) − V_j(Ã_j), min{V_i(Ã_i) − V_i(A_i), b_i}]
            if p_j accepts offer then
                1) Finalize hyper-nodes move
                2) Reevaluates envy using Algorithm 1 and updates hypergraph
            end if
        end while
        ———Offer Selection Strategy———
        while offers on the table do
            Selects offer with
              { (a) highest social welfare gain, or
              { (b) largest envy degree decrease, or
              { (c) highest side utility increase
            /*Evaluates potential deal*/
            if Σ_{i=1}^n σ_i(Ã) < Σ_{i=1}^n σ_i(A) then
                Accepts offer
            else
                Back to loop beginning and reselects offer
            end if
        end while
    end for
end while
```

VI. Performance Evaluation

In this section we investigate the performance of aforementioned hypergraph scheduling strategies through three sets of simulations. First we implement Algorithm 2 for budget-unaware performance validation. In the second set of simulations we compare V-HCN, E-HCN and U-HCN in various performance metrics, and finally in the third set of simulation we analyze the effects of different bidding strategies and examine performance impact of initial budget settings.

A. Simulation Settings

All the scheduling algorithms are implemented using the SimGrid framework [33]. Specifically all the core functions of hypergraph scheduling are implemented using the application-level simulation interfaces in MSG module of SimGrid V3.2. To accommodate heterogeneity, we created computing platforms and network deployments in accordance with realistic resource distributions of heterogeneous desktop grid environment [1] and encapsulated the configuration information in separate *XML* files. For budget-unaware simulations we evaluated performance with system scale ranging from 50 to 80 computing peers, and for budget-aware simulations we used a small system with 20 peers. We also created multiple independent tasks with identical attributes of computational size S_{cp} and communication size S_{cm}, and at each run the number of tasks is set to be four times of the peer number. In experiments of [4] task valuations for each peer are simply randomly generated. Here we propose a strategy which limits the maximum valuation v_i^{max} of each peer p_i and uniformly distributes p_i's valuation for each single unit of task over $\{1, ..., v_i^{max}\}$. This method better differentiates the peers based on the heterogeneity nature of the system, as users would like to pay more for faster service per task, the same task presents more values to the more powerful peer. In addition it conforms to the modular valuation assumption. Suppose each task is associated with a fixed initial setup overhead time t^{setup}, and let p_i has incoming bandwidth bw_i (*B/s*) and computing power cp_i (*flop/s*), the maximum valuation v_i^{max} is defined as the reciprocal of the task handling time:

$$\begin{aligned} v_i^{max} &= \frac{1}{\hat{t_i}} \text{ in which:} \\ \hat{t_i} &= t^{setup} + t^{trans} + t^{exec} \\ &= t^{setup} + \frac{S_{cm}}{bw_i} + \frac{S_{cp}}{cp_i} \end{aligned} \tag{6}$$

Four performance metrics are used in this study. First we use *social welfare* to indicate the efficiency of the allocation. Next after each deal happens we calculate *total envy degree* and *envious peer number*, both of which capture the fairness of the scheduling result. Finally we measure *system profit* as an indication of side utility. The system profit is defined as the summation of profit gain of all deals.

B. Budget-unaware Simulation Analysis

In this set of simulations, peers negotiate with each other using payment function GUPF until reaching complete locally efficient allocation. The results are shown in Figure 5. First we run the test with 80 peers and 320 tasks and examined allocation efficiency in terms of overall social welfare and individual utility respectively. The simulation stopped when no more rational deal can be found. To better visualize the trend we logarithmically scale the results. The convergent point of each curve indicates the termination of the algorithm. Based on Figure 5(a) two observations are made. First both curves increase monotonically, as each deal is rational. Second, the two curves increase at the same rate. Both confirm the conclusions of Lemma 2 and Theorem 3. For the rest of the simulations we raised the system scale from 50 to 80 to check scheduling fairness. In Figure 5(b) and Figure 5(c) we can observe that, all negotiations are converged to fair state where no agent would envy others. In addition the directed hypergraph scheduling is scalable as deal number for convergence grows in accordance with the scale increase. Note that after each deal both the envy degree and the envious pair number do not necessarily decrease. This can be explained by the oscillation phenomenon mentioned in Section IV: although the system envy will be wiped out eventually, each single deal only eliminates envy between the agents participating in the deal, but may cause more envy in the whole system.

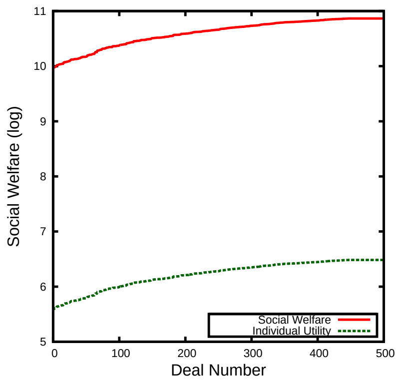
C. Budget-aware Simulation Analysis

Next we added budget limitation to each peer and compared the performance of different versions of HCN algorithm proposed in Section V. The system size is set to be 20. According to analysis of average deal payment range we give each peer an initial budget of 100. When the deal is accepted the peer who makes payment will deduct the corresponding amount from its budget and its deal partner will receive this amount as a bonus to its budget. To achieve a fair comparison all simulations were conducted with the same valuation distribution and initial topology. Moreover, the algorithm behaves in a stochastic manner as payment is randomly selected within rational and budget range at each deal. To avoid this effect we averaged the results of 10 runs for each algorithm. The comparison results are displayed in Figure 6. From these results we draw the conclusion that the final allocation performance is influenced by the offer selection strategy at each deal. In V-HCN the offer brings maximum social welfare increase is selected, therefore in Figure 6(a) we observe that V-HCN yields to highest local efficiency when converged. Similarly Figure 6(c) and Figure 6(d) shows that E-HCN performs better in eliminating system envy, while in Figure 6(b) the overall profits are in favor of U-HCN.

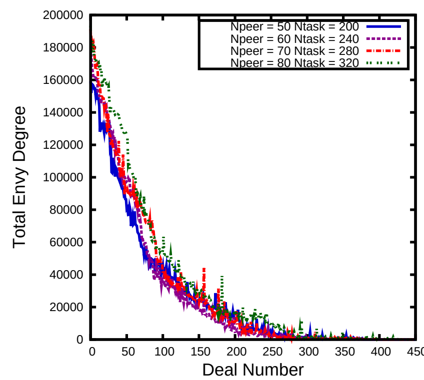
At last we investigate the impact of different payment strategies and initial budget settings. In the original U-HCN algorithm, payment is selected randomly subject to Equation 5 at each deal. Suppose at each deal the payment is selected within the range of $[lowest, highest]$, now instead of random selection, we deterministically select payment amount of fixed value within the range. Specifically we devised three bidding strategies for evaluation:

- aggressive bidding
  $payment = lowest + 0.75 \times (highest - lowest)$
- modest bidding
  $payment = lowest + 0.5 \times (highest - lowest)$
- conservative bidding
  $payment = lowest + 0.25 \times (highest - lowest)$

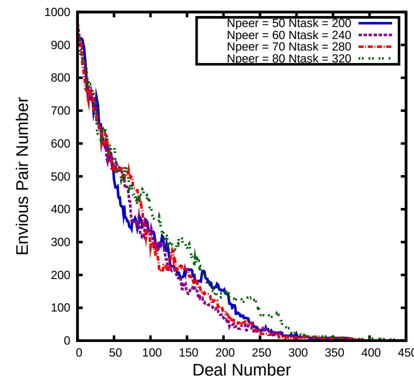
The results shown in Figure 7(a) suggest that more aggressive bidding behavior will result in higher system profits after each deal. This can be explained that if all peers are willing to offer higher at each bargain, the final successful deals will definitely bring more incomes, thus lead to higher overall profits. We also generated different initial budgets for the system and observed its impact. We selected E-HCN as the base algorithm and experimented with different amounts of initial budget assignment. Again the same valuation distribution and topology were created for fair comparison. Three scenarios representing *rich*, *middle class* and *poor* startup funds were simulated and the results are exhibited in Figure 7(b) and Figure 7(c). Clearly for the system with abundant initial funds, the convergence to the local minimum envy degree is close to the budget-unaware situation. On the contrary for the system with insufficient funds the offer making process is more likely to get stuck due to lack of budgets, resulting in longer convergence rate and higher local minimum envy.



(a) Efficiency improvement

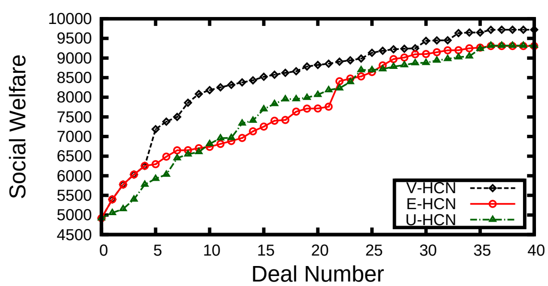


(b) System envy degree drop

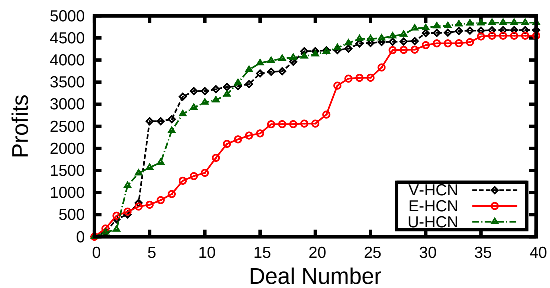


(c) Envious pair number drop

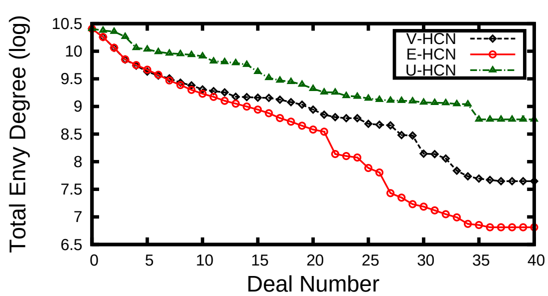
Figure 5. Budget-unaware: hypergraph scheduling performance metrics



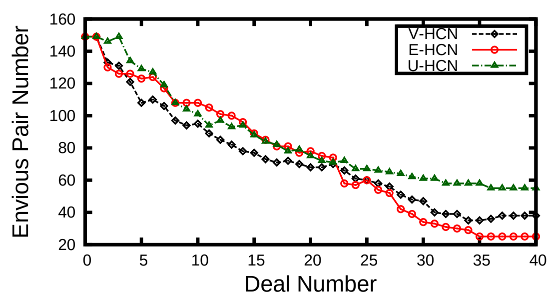
(a) Efficiency improvement



(b) Profits gain

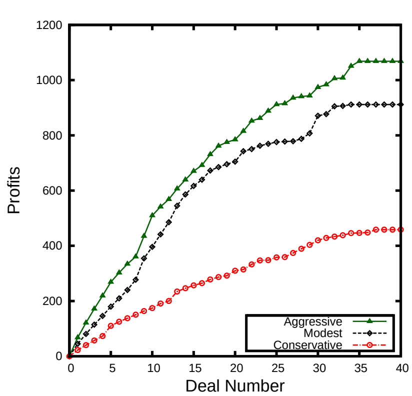


(c) System envy degree decrement

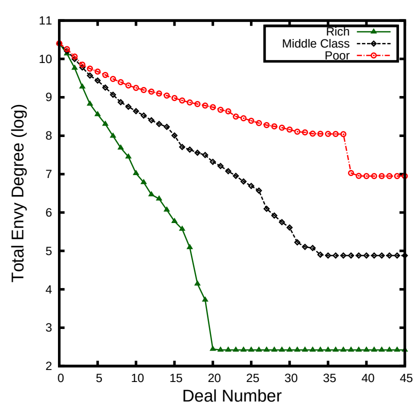


(d) Envious pair number decrement

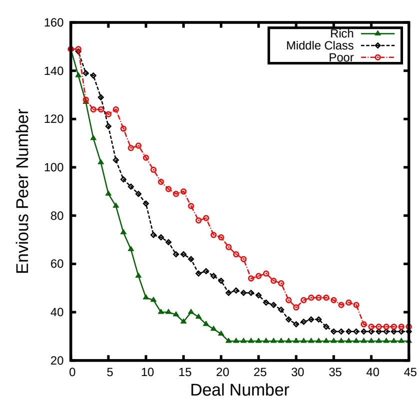
Figure 6. Budget-aware: performance comparison of hill climbing negotiation strategies (HCN)



(a) Impact of bidding strategy for modified U-HCN



(b) Impact of initial budget assignment for V-HCN: envy degree



(c) Impact of initial budget assignment for V-HCN: envious pair number

Figure 7. Budget-aware: Potential performance impact factor analysis

VII. Conclusion

In this paper we have studied the problem of BoT-style task-bundle allocation in highly heterogeneous distributed systems. The goal of this research is to achieve efficient and fair allocation in a decentralized manner. Towards this goal we proposed a directed hypergraph scheduling model, which integrates allocation efficiency and envy information in a three-dimensional matrix. Using the hypergraph model, we presented individual negotiation strategies with and without budget constraint. For the budget-unaware scenario the strategy is guaranteed to converge to complete locally efficient and fair allocation, while for the later case, we developed a hill-climbing heuristic strategy that is capable of achieving allocation close to maximum efficiency and fairness. Simulation results show that the proposed hypergraph scheduling methods perform well in a wide range of scenarios.

Acknowledgment

The authors would like to thank the anonymous referees for their suggestions, which have helped improve the quality and presentation of this paper.

References

[1] H. Zhao and X. Li, "Efficient grid task-bundle allocation using bargaining based self-adaptive auction," in *IEEE International Symposium on Cluster Computing and the Grid (CCGrid 09)*, vol. 0, 2009, pp. 4–11.

[2] A. Iosup, O. Sonmez, S. Anoep, and D. Epema, "The performance of bags-of-tasks in large-scale distributed systems," in *Proceedings of the 17th international symposium on High performance distributed computing (HPDC 08)*, 2008, pp. 97–108.

[3] Y. Chevaleyre, U. Endriss, and N. Maudet, "Allocating goods on a graph to eliminate envy," in *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI 07)*, July 2007, pp. 700–705.

[4] Y. Chevaleyre, U. Endriss, S. Estivie, and N. Maudet, "Reaching envy-free states in distributed negotiation settings," in *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI 07)*, January 2007, pp. 1239–1244.

[5] E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden, "The price of stability for network design with fair cost allocation," in *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS 04)*, 2004, pp. 295–304.

[6] R. Buyya, D. Abramson, and J. Giddy, "Nimrod/G: An architecture of a resource management and scheduling system in a global computational grid," in *Proceedings of the 4th International Conference on High Performance Computing in the Asia-Pacific Region*, vol. 1, 2000, pp. 283–289.

[7] R. Wolski, J. S. Plank, J. Brevik, and T. Bryan, "G-commerce: Market formulations controlling resource allocation on the computational grid," in *Proceedings of the 15th International Parallel & Distributed Processing Symposium (IPDPS 01)*, 2001, pp. 46–53.

[8] A. Das and D. Grosu, "Combinatorial auction-based protocols for resource allocation in grids," in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS 05) - Workshop 13*, 2005.

[9] A. Archer and E. Tardos, "Truthful mechanisms for one-parameter agents," in *FOCS '01: Proceedings of the 42nd IEEE symposium on Foundations of Computer Science*, 2001, p. 482.

[10] A. Archer, C. Papadimitriou, K. Talwar, and E. Tardos, "An approximate truthful mechanism for combinatorial auctions with single parameter agents," in *SODA '03: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, 2003, pp. 205–214.

[11] A. F. Archer, "Mechanisms for discrete optimization with rational agents," Ph.D. dissertation, Cornell University, Ithaca, NY, USA, 2004, adviser-Tardos, Éva.

[12] W. Vickrey, "Counterspeculation, auctions, and competitive sealed tenders," *The Journal of Finance*, vol. 16, no. 1, pp. 8–37, 1961.

[13] E. H. Clarke, "Multipart pricing of public goods," *Public Choice*, vol. 11, no. 1, 1971.

[14] T. Groves, "Incentives in teams," *Econometrica*, vol. 41, no. 4, pp. 617–31, 1973.

[15] L. He and T. R. Ioerger, "Task-oriented computational economic-based distributed resource allocation mechanisms for computational grids," in *Proceedings of the International Conference on Artificial Intelligence (IC-AI 04)*, 2004, pp. 462–468.

[16] K. H. Kim and R. Buyya, "Fair resource sharing in hierarchical virtual organizations for global grids," in *Proceedings of the 8th IEEE/ACM International Conference on Grid Computing (GRID 07)*, 2007, pp. 50–57.

[17] N. Doulamis, E. Varvarigos1, and T. Varvarigou, "Fair scheduling algorithms in grids," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 11, pp. 1630–1648, 2007.

[18] L. Amar, A. Mu'alem, and J. Stößer, "On the importance of migration for fairness in online grid markets," in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems (AAMAS 08)*, 2008, pp. 1299–1302.

[19] J. Edmonds and K. Pruhs, "Balanced allocations of cake," in *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 06)*, 2006, pp. 623–634.

[20] K. Rzadca, D. Trystram, and A. Wierzbicki, "Fair game-theoretic resource management in dedicated grids," in *Proceedings of the Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGRID 07)*, 2007, pp. 343–350.

[21] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *The Bell system technical journal*, pp. 291–307, 1970.

[22] Q. Li, G. Kim, and R. Negi, "Maximal scheduling in a hypergraph model for wireless networks," in *IEEE International Conference on Communications (ICC 08)*, May 2008, pp. 3853–3857.

[23] A. L. Rosenberg, "Path-robust multi-channel wireless networks," in *Proceedings of 23th International Parallel and Distributed Processing Symposium (IPDPS 09)*, 2009.

[24] C. J. Alpert, J.-H. Huang, and A. B. Kahng, "Multilevel circuit partitioning," *Design Automation Conference*, vol. 0, p. 530, 1997.

[25] S. Krishnamoorthy, U. Çatalyürek, J. Nieplocha, A. Rountev, and P. Sadayappan, "Hypergraph partitioning for automatic memory hierarchy management," in *Proceedings of the 2006 ACM/IEEE conference on Supercomputing (SC 06)*, 2006.

[26] E. Amir and S. McIlraith, "Partition-based logical reasoning for first-order and propositional theories," *Artif. Intell.*, vol. 162, no. 1-2, pp. 49–88, 2005.

[27] U. Çatalyürek and C. Aykanat, "Hypergraph-partitioning-based decomposition for parallel sparse-matrix vector multiplication," *IEEE Trans. Parallel Distrib. Syst.*, vol. 10, no. 7, pp. 673–693, 1999.

[28] K. D. Devine, E. Boman, R. T. Heaphy, R. H. Bisseling, and U. V. Çatalyürek, "Parallel hypergraph partitioning for scientific computing," in *Proceedings of 20th International Parallel and Distributed Processing Symposium (IPDPS 06)*, 2006.

[29] U. Çatalyürek, E. Boman, K. Devine, D. Bozdag, R. Heaphy, and L. Riesen, "Hypergraph-based dynamic load balancing for adaptive scientific computations," in *Proceedings of 21th International Parallel and Distributed Processing Symposium (IPDPS 07)*, March 2007, pp. 1–11.

[30] S. J. Brams and A. D. Taylor, *Fair division : from cake-cutting to dispute resolution*. Cambridge University Press, 1996.

[31] N. Laoutaris, L. J. Poplawski, R. Rajaraman, R. Sundaram, and S.-H. Teng, "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 08)*, 2008, pp. 165–174.

[32] S. Estivie, Y. Chevaleyre, U. Endriss, and N. Maudet, "How equitable is rational negotiation?" in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems (AAMAS 06)*, 2006, pp. 866–873.

[33] H. Casanova, A. Legrand, and M. Quinson, "SimGrid: a Generic Framework for Large-Scale Distributed Experiments," in *10th IEEE International Conference on Computer Modeling and Simulation*, 2008.