Available online at www.sciencedirect.com
SCIENCE DIRECT®

J. Parallel Distrib. Comput. 65 (2005) 1366 – 1383

Journal of Parallel and Distributed Computing
www.elsevier.com/locate/jpdc

A pricing strategy for job allocation in mobile grids using a non-cooperative bargaining theory framework

Preetam Ghosh*, Nirmalya Roy, Sajal K. Das, Kalyan Basu

Center for Research in Wireless Mobility and Networking (CReWMaN), Department of Computer Science and Engineering, The University of Texas at Arlington, Arlington, TX 76019-0015, USA

Received 5 January 2004; received in revised form 5 November 2004; accepted 11 May 2005
Available online 11 July 2005

Abstract

Due to their inherent limitations in computational and battery power, storage and available bandwidth, mobile devices have not yet been widely integrated into grid computing platforms. However, millions of laptops, PDAs and other portable devices remain unused most of the time, and this huge repository of resources can be potentially utilized, leading to what is called a mobile grid environment. In this paper, we propose a game theoretic pricing strategy for efficient job allocation in mobile grids. By drawing upon the Nash bargaining solution, we show how to derive a unified framework for addressing such issues as network efficiency, fairness, utility maximization, and pricing. In particular, we characterize a two-player, non-cooperative, alternating-offer bargaining game between the Wireless Access Point Server and the mobile devices to determine a fair pricing strategy which is then used to effectively allocate jobs to the mobile devices with a goal to maximize the revenue for the grid users. Simulation results show that the proposed job allocation strategy is comparable to other task allocation schemes in terms of the overall system response time.

Keywords: Grid computing; Mobile devices; Bargaining theory; Non-cooperative games; Load balancing

1. Introduction

Grid computing provides a distributed computing infrastructure for solving large-scale advanced scientific and engineering problems through sharing of resources, usually over high-speed communication networks [8,9]. Computational grids typically have a conglomeration of various resources with different owners at geographically different sites. Several Grid systems including Globus [7] have addressed many of these issues with the exception of resource trading and quality of service (QoS)-based scheduling. The GRACE [2] architecture leverages existing technologies such as Globus, and provides new services that are essential for resource trading and aggregation, depending on their availability, capability, cost, and users' QoS requirements. An important issue of such grid computing systems is the efficient assignment of jobs and utilization of resources of unused devices, commonly referred to as the *load balancing* or *job scheduling* problem. This problem is often formulated in the context of a *system model*, an abstraction of the underlying resources, that provides information to the job allocator regarding the availability and properties of resources at any point in time. The job allocator then allocates jobs to the available resources and attempts to optimize specified performance metrics, such as time deadline or revenue maximization.

Given that millions of laptops, PDAs and other portable devices remain unused most of the time, the grid architecture is recently extended in [21] leading to what is called a *mobile grid* environment. The goal is to potentially utilize the huge repository of resources of mobile devices to provide

* Corresponding author. Fax.: +1 817 272 3784.
E-mail addresses: ghosh@cse.uta.edu (P. Ghosh), niroy@cse.uta.edu (N. Roy), das@cse.uta.edu (S.K. Das), basu@cse.uta.edu (K. Basu)
URLs: http://crewman.uta.edu/∼preetam (P. Ghosh), http://crewman.uta.edu/∼nirmalya, http://crewman.uta.edu/∼das, http://crewman.uta.edu/∼basu.

0743-7315/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jpdc.2005.05.013

a seamless source of computational power and storage capacity. However, this concept offers significant challenges mainly due to the inherent limitations in processing, memory, battery power and wireless communications capabilities of mobile devices. In a mobile grid, a more important performance metric is system *throughput* where the resources are *distributively owned*. In this environment, a resource owner has the right to define a very sophisticated *usage policy*, e.g., a job can run on a mobile device only if it generates a certain minimum revenue. Distributed ownership requires a scheduling paradigm that can operate in an environment where *resource owners* (i.e., mobile devices) and *resource users* (i.e., wireless access points servers) dynamically define their own policies and models.

Job scheduling in mobile grid computing thus demands for a decentralized algorithm with a robust system model. We also need to consider an *economic pricing model* that will govern the cost benefits of mobile device owners to allow complex computational jobs to be performed at those devices. Due to the conflict of interest between the players, namely the mobile device and the wireless access point server (WAPS), this pricing model can be more realistically formulated using a *non-cooperative bargaining theory* [20] framework.

Although Game Theoretic approaches have been proposed to develop economic models for resource management and scheduling in grid computing [3], they suffer from precise lack of formulation in the sense that the actual mapping of the problem into a game between two players has not been shown, nor are stated analytical modeling and results. Also, mobile grid computing is a completely new paradigm for which only a very crude economic model has been specified in [21]. We envision that potentially there are many mobile devices distributed in the network, which will be competing to share the jobs originated by the grid community. There arise several challenging issues such as:

(1) efficient job allocation to different mobile devices taking into account various performance requirements;
(2) handling fairness in pricing the job allocation;
(3) the ability to implement the allocation scheme in a distributed manner with minimum communication overheads;
(4) maximizing the network efficiency, i.e., minimizing the response time.

1.1. Related works

In mobile grid environments, the integration of wireless mobile devices to exploit the available processing power introduces new challenges. A proxy-based clustered architecture for mobile grids is proposed in [21]. The pricing and job scheduling policies in mobile grids need to manage resources and application execution depending on the requirements of resource consumers (i.e., WAP Servers) and resource owners (i.e., mobile devices). They also need to continuously adapt to changes in the availability of resources. This introduces a number of challenging issues that need to be addressed; namely, site autonomy, resource allocation or co-allocation, online control and so on. Several grid systems including Globus [7] have addressed many of these issues with the exception of resource trading and QoS-based scheduling. The GRACE framework [2] particularly addresses these two later issues by leveraging existing technologies such as Globus and providing new services that are essential for resource trading and aggregation, depending on their availability, capability, cost, and user QoS requirements. It develops a generic distributed game theoretic architectural framework and strategies for resource trading using different economic models.

Some scheduling mechanisms based on game theoretic negotiation, deployed in existing grid computing systems, is shown in Table 1. However, none of these games attempts to capture the competitiveness among the mobile devices, nor do they aim at maximizing the grid community's revenue. Cooperative game theory has been used to obtain a Nash bargaining framework to address issues like network efficiency, fairness and revenue maximization for bandwidth allocation and pricing in broadband networks [28]. Direct application of a cooperative bargaining theory solution [11] and an optimal scheme based on the overall system response time for load balancing [25] do not consider the pricing constraints of a mobile device. In this paper, we first propose a pricing model and subsequently address the issue of dynamic job allocation such that the grid community's revenue is maximized and also the overall expected job execution time is minimized.

1.2. Our contributions

The main contributions of this paper are two-fold. First, we propose a game theoretic framework to implement the pricing model. The two players, namely the WAP Server (acting on behalf of the grid community) and the mobile device, play an incomplete information alternating-offer, non-cooperative bargaining game [4,6,17] to decide upon the *price per unit resource* charged by that mobile device. The dynamics of interaction is shown in Fig. 1. The concept of incomplete information ensures that the two players have no idea of each other's reserved valuations, i.e., the maximum offered price for WAP Server (acting as the *buyer* of resources) and minimum expected price for mobile device (acting as the *seller* of resources). Assuming there are n mobile devices under a single WAP Server, the WAP Server has to play n such games with the corresponding devices to form the price per unit resource vector, p_i. In particular, by drawing upon the Nash bargaining framework from non-cooperative game theory, the pricing strategy is guaranteed to be fair. Furthermore, we make this pricing scheme stable, so that there would be no incentives for the grid community or the mobile devices to deviate from the mutual

Table 1
Grid resource scheduling based on game theoretic approach

Grid scheduling systems	Game theoretic model	Role on resource management
Spawn [26] and Popcorn [18]	Auction Model	Spawn supports execution of a hierarchy of variable size processes depending upon the resource cost. An application of Popcorn need to specify a budget for processing each of its modules.
Nimrod-G [2]	Bargaining Model, Posted Price Model	Supports budget and deadline constrained scheduling algorithm. The resource assignment depends on cost, power, availability and QoS requirement of users.
Mungi [12], MOSIX [1] and Nimrod-G	Commodity Market Model	Mungi allows the object to get some storage area, against certain rent from them.
Rexec and Anemone [5]	Bid-based Proportional Resource Sharing	In this case resource assignment occurs based on the value of Utility function.
SETI@Home, Condor [22] and MojoNation [16]	Community, Coalition and Bartering	MojoNation is a content-sharing community network. Here contributors can earn revenue by sharing storage.
Mariposa [23]	Tender/Contract-Net Model	Mariposa behaves like Popcorn. It supports budget-based processing and storage management.
This Paper	Non-Cooperative Bargaining Model	Here load balancing occurs based on the pricing strategy, maximizing the revenue of the grid community and minimizing the job execution time.

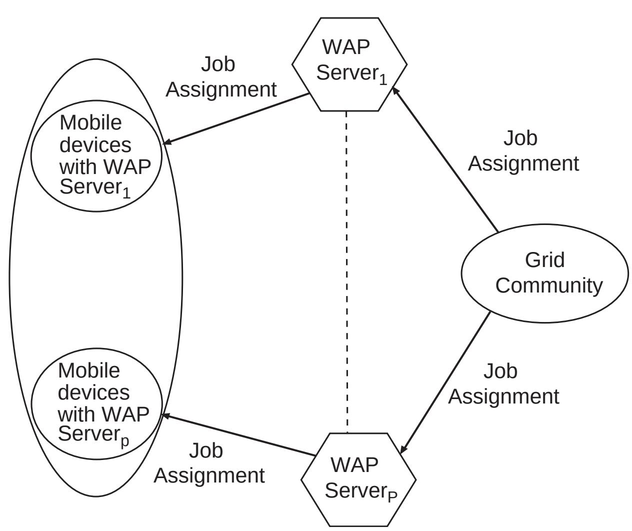


Fig. 1. Dynamics of different mobile user groups with different WAPS.

agreement. Our bargaining protocol is simple, reduces computational and communication costs, and also avoids using a central matchmaker that may otherwise be a bottleneck in the system.

The second important contribution is the workload allocation scheme based on the derived pricing model. We formulate the job scheduling as a *constrained minimization* problem that will maximize the revenue for the grid community. We also introduce a new algorithm to allocate jobs to the different mobile devices. Our bargaining scheme is shown to work well based on the market dynamics. The job allocation strategy yields an overall execution time comparable to other load balancing schemes. A preliminary version of this work appeared in [10].

The rest of the paper is organized as follows. Sections 2 describes an architectural overview of our mobile grid computing system. The problem formulation and motivation behind our bargaining game are discussed in Section 3. The bargaining protocol is presented in Section 4, while Section 5 highlights the job allocation scheme. Section 6 presents the simulation results and Section 7 concludes the paper.

2. Mobile grid computing architecture

Fig. 2 illustrate an architecture for mobile grid computing. It is based on a wireless cellular network in which each cell consists of a number of mobile devices along with one wireless access point (WAP). Each such cell is called a basic service set (BSS) according to the IEEE 802.11 based wireless LAN nomenclature [13]. The WAP inside each BSS is connected through an Intranet. The WAP Server acts as a job allocator as well as a negotiator during each bargaining session on behalf of the grid community. Multiple BSSs are connected together to form an extended service set (ESS). A mobile device can change its location for bargaining from one BSS to another BSS in which case it will negotiate with the corresponding WAP Server of the new BSS, i.e., it can negotiate with any WAP Server under the same ESS. Now these WAP Servers can be interconnected with or without wires with an edge router which accepts the job from a grid controller (GC) and also returns the computational results to the GC. The GC is a dedicated node or server of the grid community and acts as the logical component to interconnect the BSSs. This GC provides distributed services to allow for the roaming of mobile devices between BSSs. We assume that there is a job scheduler in the GC that will assign jobs on behalf of the grid community to different WAP Servers under it according to their capacity through an edge router. The WAP Server in turn subdivides the job among different mobile devices according to their resource constraints.

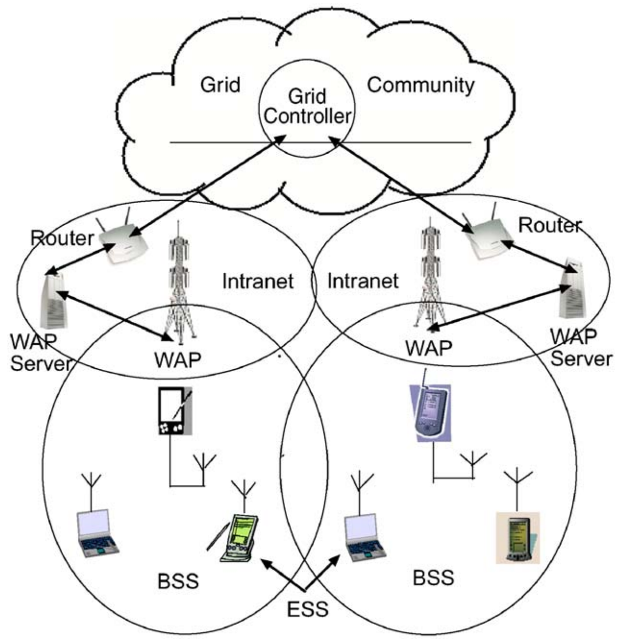


Fig. 2. System architecture of mobile grid.

It starts an alternating offer bargaining game according to the rules of our proposed bargaining protocol to fix the price vector and then allocates jobs optimally to maximize the GC's revenue. On completion of the allocated jobs, the mobile devices will send back the results to the WAP Server, which in turn is returned to the GC via the edge router.

3. Game notation for pricing

In a mobile grid environment, the WAP Server tries to acquire some available resources from a large number of mobile devices. Let us analyze the competitive mobile grid resource management scenario. Assume there are a total P WAP Servers and a total of Q mobile devices. At time t, let the number of devices under Server W_i be denoted as $n(t)_i$, where $1 \leqslant i \leqslant P$, such that $\sum_{i=1}^{P} n(t)_i = Q$. Fig. 3 depicts a snapshot of the system where each mobile device is only associated with one WAP Server at a time. If we can model this one-to-one relationship between a particular mobile device and its current WAP Server as a bargaining game, BG_j, for $1 \leqslant j \leqslant Q$, at any instant and manage their relationship properly with the game output, then the competitive scenario can be considered as multiple instances of this one-to-one game between the two players (e.g.,WAP Server vs. mobile device).

Both the players will try to maximize their utility functions defined later and hence the game reduces to the simple case of dividing the difference between the maximum buying price offered by the grid community and the minimum selling price expected by the mobile users, called the reserved valuations [20] according to game theory conventions. A complete information cooperative bargaining theory

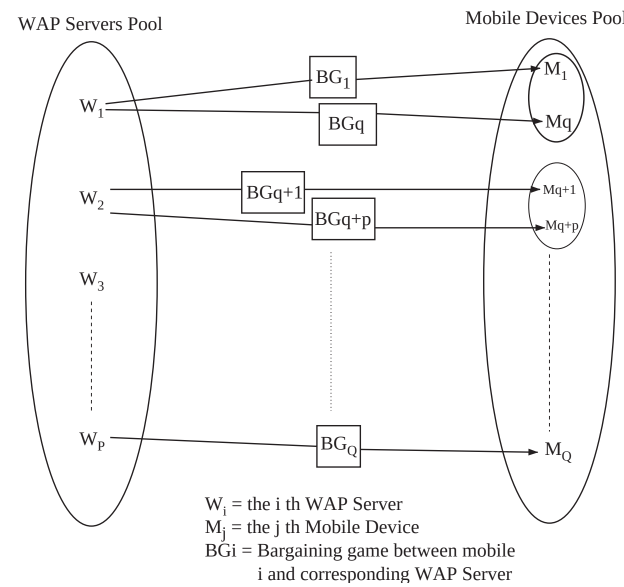


Fig. 3. Mapping of multiple mobile user pool under different WAPs.

solution requires both players to know these reserved valuations beforehand, which is not realistic, particularly for a grid environment. Therefore, we need to model the game using incomplete information alternating-offers bargaining theory.

4. The bargaining protocol

Let us denote the resulting game as $\Gamma(q)$ where q is the probability that the negotiation will break down in any period. The strategy of both WAP Server and mobile device in $\Gamma(q)$ is defined exactly as for a bargaining game of alternating offers [20]. Let (λ, γ) be a pair of strategies that leads to the outcome (x^t, t) in a bargaining game of alternating offers. Let x denotes the outcome for the first bargainer, y the outcome for the second bargainer and t is the associated time deadline with that outcome. Thus (λ, γ) leads to agreement (x^t, t) with probability $(1-q)^t$ and to break-down (BD) with probability $1-(1-q)^t$. The algorithm, shown in Fig. 4, describes the basic procedure of alternating-offer bargaining.

Following Nash [19], the term "bargaining" is used to refer to a situation in which:

(1) There is a conflict of interest on which the agreement to conclude,
(2) A WAP Server and mobile devices under it have the power to conclude a mutually beneficial agreement, and
(3) Any agreement cannot be imposed on either the WAP Server or mobile devices without their approval.

The bargaining procedure is as follows. Bargainer A starts the negotiation by sending a proposal to bargainer B. Now B can either accept or reject it. If the offer is accepted, then the bargaining ends and the agreement is implemented. If the

```
Initialize t = 0 ;
WAP Server proposes an offer (x^0)
if x^0 ≥ (Mobile device's standard
price with highest expected surplus)
   then Mobile device accepts the offer;
        outcome=(x^0,0);
   else Increment t;
Mobile device counter proposes an offer y^1;
if y^1 ≤ (WAP Server's standard price
with highest expected surplus)
   then WAP Server accept;
        outcome=(y^1,1);
   else Increment t;
WAP Server proposes an offer (x^2)
if x^2 ≥ (Mobile device's standard
price with highest expected surplus)
   then Mobile device accepts the offer;
        outcome=(x^2,2);
   else Increment t;
Mobile device counter proposes an offer y^3;
if y^3 ≤ (WAP Server's standard
price with highest expected surplus)
   then WAP Server accept;
        outcome=(y^3,3);
   else continues;
```

Fig. 4. Procedure of an alternating offer.

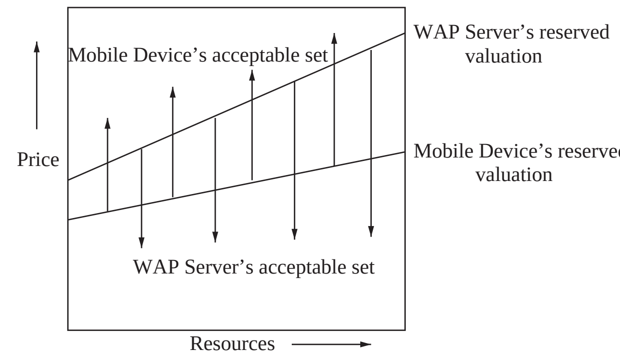


Fig. 5. Bargaining solution space.

offer is rejected, then B must send back a counter-proposal to specify his preferences to A. Now A will evaluate the proposal and choose either to accept or reject it. This process continues until an agreement is reached. In order to produce a solution space for bargaining, the individual valuations of a mobile device (seller of resources) and the WAP Server (buyer of resources) should overlap. The solution space as shown in Fig. 5 represents the bargaining of two variables: price and resources like CPU cycles, power consumption, memory bandwidth, storage capacity, etc. The *reserved valuation* of a mobile device is a straight line representing minimum selling price at different resources. So, the mobile devices' acceptable set is all the points above this straight line such that the higher points are strictly preferred. The upward slope of this line signifies that the minimum price increases as the resources utilization increases. So, the mobile device's offered price always lies above this line. Conversely, the WAP Server's reserved valuation is its maximum buying price at different resources, which is represented in Fig. 5 by the upper straight line. The WAP Server's acceptable set is all the points below this straight line, in which the lower points are strictly preferred. Note is that each player wants to offer a price that is farther from its reserved valuation. In other words, the solution space consists of the points in between these two straight lines, i.e., the intersection of these two acceptable sets. Consequently, the result of bargaining will fall in this solution space where both the WAP Server and the mobile device can make a surplus and try to reach a mutually beneficial agreement. Following the approach in [27], we characterize the game by the three rules described below.

Rule 1. Both players choose an alternative at every step that earns them the highest expected surplus, such as the maximum possible profit.

From WAP Server's point of view,

$$\begin{aligned} \textit{Expected utility} = E[\textit{Surplus}] &= (\textit{reserved valuation of } w \\ &\quad - \textit{standard price}) \times \textit{probability} \\ &\quad \times (\textit{standard price}), \end{aligned}$$

where $E[X]$ denotes the expected value of X, w stands for WAP Server, and probability (standard price) is the probability that the standard price will be accepted by the mobile device as predicted by the WAP Server. The standard price represents the different offered prices used by the WAP Server to compute its expected surplus.

On the other hand, from the mobile device's point of view,

$$\begin{aligned} \textit{Expected utility} = E[\textit{Surplus}] = (&\textit{standard price} \\ &- \textit{reserved valuation of } m) \\ &\times \textit{probability } (\textit{standard price}), \end{aligned}$$

where m stands for the mobile device and probability (standard price) is the probability that the standard price will be accepted by the WAP Server as predicted by the mobile device. Here the standard price represents the different requested prices used by the mobile device to compute its expected surplus.

Rule 2. If an offer is rejected, then the WAP Server and mobile device will reduce the probability (standard price) based on the available resources, bandwidth and time deadline. This reduction monotonically decreases as the alternatives come closer to their reserved valuations. In other words, it is more likely to be accepted by the opponent.

Rule 3. The demands of both WAP Server and mobile device are decreased over time. For this we have considered a negative exponential function over time, say $e^{-z_i t}$, where $z_i > 0$ for $i = w$ or m, denotes the discount rates of both the negotiators. This assumption helps to converge the negotiation scheme faster. Considering this rule, for WAP

Table 2
WAP Server's computation for making decision

Offered price ($)	Probability	Expected surplus ($)
40	0.10	06
60	0.40	16
80	0.70	14
90	0.90	09
100	1.00	00

Table 3
Mobile device's computation for making decision

Offered price ($)	Probability	Expected surplus ($)
60	1.00	00
70	0.90	09
80	0.70	14
90	0.40	12
110	0.10	05

Server:

$$\textit{Expected utility} = E[\textit{Surplus}] = (\textit{reserved valuation of } w - \textit{standard price}) \times \textit{probability} \times (\textit{standard price}) \times e^{-z_w t}$$

and for mobile device:

$$\textit{Expected utility} = E[\textit{Surplus}] = (\textit{standard price} - \textit{reserved valuation of } m) \times \textit{probability } (\textit{standard price}) \times e^{-z_m t}.$$

An Example: Let us consider the following example. Assume that the WAP server's reserved valuation for a fixed resource is \$100 (maximum buying price) and the mobile device's reserved valuation for the same resource is \$60 (minimum selling price). Let the mobile device make an initial offer of \$110. The WAP Server finds that it does not maximize its surplus and hence rejects the offer and performs the calculations as shown in Table 2.

The WAP Server's counter-offer depends upon its expected surplus. It asks for \$60, which maximizes its expected surplus. Now its the mobile device's turn. Its acceptance or rejection depends upon the calculations in Table 3.

Since the WAP Server's offer of \$60 does not match the mobile device's corresponding offered price with the highest expected surplus, it is rejected. The mobile device makes a counter-offer and asks for \$80. Then the WAP Server, on finding its offer was rejected, modifies his prediction probability scheme by certain rules, such as probability of all standard price is decreased by 0.30 (say). Now the new action at WAP Server's end depends upon the expected surplus in Table 4.

Table 4
WAP Server's computation using modified probability for making decision

Offered price ($)	Updated probability	Expected surplus ($)
40	$0.10 - 0.30 = 0.00$	00
60	$0.40 - 0.30 = 0.10$	04
80	$0.70 - 0.30 = 0.40$	08
90	$0.90 - 0.30 = 0.60$	06
100	$1.00 - 0.30 = 0.70$	00

In this case the mobile device's offered price matches with WAP Server's standard price corresponding to the highest expected surplus. So, they ultimately reach an agreement that lies in the solution space given by the intersection of the two acceptable sets as shown in Fig. 4.

4.1. Attributes of WAP Server and mobile device

The real-life parameters affecting the game are modeled by the following attributes. Here we have used x to represent both the WAP Server (w) and Mobile device (m) depending upon the context (Table 5).

Reserved valuation (R_x): R_w is the maximum possible price that the grid community (i.e., WAP Server) is willing to pay to get the work done. R_m is the minimum possible price that the mobile device m expects for doing the work.

Market price (M_x): This value is calculated based on the statistics. The WAP Server and mobile device will maintain a history keeping track of the recent bargaining games that they have participated in. This history will help determine the "Market Value" of the resources the grid community is seeking for. So, it depends on the kind of jobs that was distributed by the WAP Server before and executed by the mobile device. Also important are the characteristics of jobs, such as time deadline, data storage required (that affects the bandwidth), CPU cycles required, etc.

Probability $(p_x^{M_x})$ *that bargainers will accept* M_x: The parameter $P_w^{M_w}$ is the WAP Server's perceived probability that there are mobile devices willing to do the job at the market price M_w. This can also be statistically calculated by maintaining the history of the previous bargaining games. The WAP Server just needs to categorize the mobile devices by the offers they accepted in previous games and then $P_w^{M_w}$ can be calculated by counting how many of the mobile devices have their accepted prices greater than or equal to M_w. Similarly, $P_m^{M_m}$ defines the mobile devices's perceived probability that the WAP Server will accept the market price M_m.

Offered price: This is basically given by maximizing the utility function of counter-offering, defined as utility (Counter offer by *bargainer*), by the negotiator. The standard price corresponding to the highest utility gives the offered price.

Opponent's offered price (O_x): It is the most recent requested price by the opponent.

Table 5
Different parameters of WAP Server and mobile device

Parameters	Meaning of the symbols
R_x	Reserved valuation
M_x	Market price
$p_x^{M_x}$	Probability that Bargainer will accept M_x, i.e., market price
O_x	Offered price
$p_x^{O_x}(acc)$	Probability predicted by the one bargainer that other bargainer will accept offered Price O_x
O_{x_y}	Expected counter offered price of one bargainer predicted by the opponent
$p_x^{O_x}(rco)$	Probability that bargainer will reject O_x and counteroffer
$p_x^{O_x}(rbd)$	Probability that bargainer will reject O_x and breakdown
$e^{-z_x t}$	Discount factor
ϑ	Resource constraints $\vartheta \in \{0, 1\}$

Probability ($p_x^{O_x}(acc)$) *that bargainers will accept* O_x: This is the negotiator's perceived probability that the opponent will accept its offer and is calculated by the profile tree approach discussed later.

Probability that bargainer will reject O_x *and break-down* ($P_x^{O_x}(rbd)$): This signifies the negotiator's perceived probability that the opponent will reject his offer and the game will break down.

Probability that bargainer will reject O_x *and counter-offer* ($P_x^{O_x}(rco)$): This signifies the negotiator's perceived probability that the opponent will reject his offer but the game will not break down. It is given by the following expression: $P_x^{O_x}(rco) = (1 - P_x^{O_x}(acc) - P_x^{O_x}(rbd))$ where $x \in \{w, m\}$.

Discount factor ($e^{-z_x t}$): This signifies the penalty afflicted on the utility function of the negotiator as the bargaining continues. The idea is to reduce the effective surplus of the negotiator in course of the bargaining game, and thus there is more incentive in completing the game early.

Expected counter-offered price (O_{x_y}) *of bargainer* x *as predicted by opponent* $y\ \forall x, y \in \{w, m\}$: This is the counter-offered price of the opponent as predicted by the negotiator and is determined by an intelligent guess of the opponent's reserved valuation.

Resource constraint (ϑ): It behaves like a step function. If the resource availability of any one of the negotiators does not fulfill the minimum requirement, then $\vartheta = 0$, otherwise $\vartheta = 1$. That is, if a mobile device does not have sufficient resources to offer, then it sets $\vartheta = 0$ and breaks down from the game.

4.2. Formal model of the pricing strategy

The various utility functions governing our bargaining game is presented in this section.

(1) *From WAP Server's perspective*: In this case, all the probabilities are predicted by the WAP Server as a belief of the mobile device's next action based on the computational state of the game as depicted in Fig. 6.

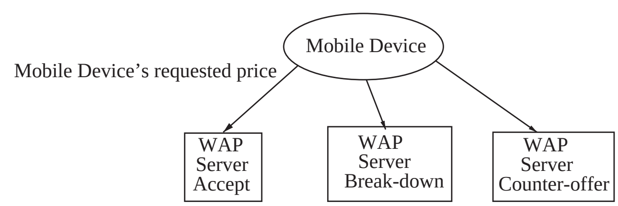


Fig. 6. Various actions of WAP Server against mobile device's offer.

1. If the WAP Server accepts the current-offer and arrives at an agreement, then its expected utility is simply

$$\begin{aligned} &\textit{Utility (Acceptance by w)} \\ &\quad = \textit{Value of (Surplus)} = ([\textit{reserved valuation of w} \\ &\qquad - \textit{offered price by m}) + (\textit{market price of w} \\ &\qquad - \textit{offered price by m})] \times \vartheta \\ &\quad = [(R_w - O_m) + (M_w - O_m)] \times \vartheta. \end{aligned}$$

The term $(M_w - O_m)$ indicates the penalty from the market if the offered price by m is accepted by the bargaining game.

2. If the WAP Server rejects the offer from mobile device and breaks down from the game, then its expected utility is

$$\begin{aligned} &\textit{Utility (Break-down by w)} \\ &\quad = [(\textit{reserved valuation of w} \\ &\qquad - \textit{market price of w}) \\ &\qquad \times \textit{probability (market price will be} \\ &\qquad \textit{accepted by mobile user)}] \times e^{-z_w t} \\ &\quad = (R_w - M_w) \times p_w^{M_w} \times e^{-z_w t}. \end{aligned}$$

As all the terms (except the discount factor) are constants for each bargaining session, *Utility* (*Break-down by w*) is reduced with time and becomes zero when $t = deadline$, which implies that the game converges. Also, we do not multiply the utility function with the resource

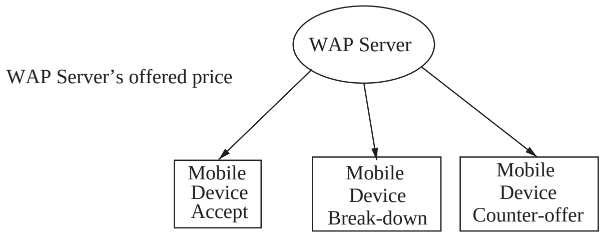


Fig. 7. Various actions of mobile device's against WAP Server's offer.

constraint, because its value is 1 when the resource requirements are met, and if ϑ goes to zero, the game should breakdown, i.e., *Utility* (*Acceptance by w*) and *Utility* (*Counter offer by w*) should become zero, which makes *Utility* (*Break-down by w*) the greatest, and the game breaks-down.

3. Now, the WAP Server's counter-offer depends upon the following factors: (i) Revenue obtained if mobile device accepts the current-offer of w, (ii) Revenue obtained if mobile device rejects the offer of w and breaks down from the game and (iii) Revenue obtained if mobile device rejects the counter offer of w and proposes another offer from its end.

$$
\begin{aligned}
&\textit{Utility (Counter-offer by w)}\\
&\quad = [[(\textit{reserved valuation of w}\\
&\qquad - \textit{current offered price by w})\\
&\qquad + (\textit{market price of w} - \textit{offered price by w})]\\
&\qquad \times \textit{probability (a mobile user will accept)}\\
&\qquad + \textit{Utility (Breakdown by w)}\\
&\qquad \times \textit{probability (a mobile user will reject and}\\
&\qquad \textit{Break-down)} + (\textit{reserved valuation of w}\\
&\qquad - \textit{expected counter-offered price of}\\
&\qquad \textit{m predicted by w}) \times \textit{probability}\\
&\qquad (\textit{a mobile user will reject and counter-offer})]\\
&\qquad \times e^{-z_w t} \times \vartheta\\
&\quad = [[(R_w - O_w) + (M_w - O_w)] \times p_w^{O_w}(acc)\\
&\qquad + [\textit{Utility (Break-down by w)} \times p_w^{O_w}(rbd)]\\
&\qquad + [(R_w - O_{m_w}) \times p_w^{O_w}(rco)]] \times e^{-z_w t} \times \vartheta.
\end{aligned}
$$

Here also, when $t = deadline$, *Utility (Counter-offer by w)* becomes zero making the game converge. Also, it should be noted that the predicted probabilities are made use of in this expression (and not in the other two), to normalize the three utility functions as we have $p_w^{O_w}(acc) + p_w^{O_w}(rbd) + p_w^{O_w}(rco) = 1$.

(2) *From mobile device's perspective*: We use the same types of utilities as in the previous case, however, all the probabilities are now predicted by mobile device as a belief of WAP Server's next action based on the state of the game as shown in Fig. 7.

$$
\begin{aligned}
&\textit{Utility (Acceptance by m)}\\
&\quad = [(O_w - R_m) + (O_w - M_m)] \times \vartheta,\\
&\textit{Utility (Break-down by m)}\\
&\quad = [(M_m - R_m) \times p_m^{M_m}] \times e^{-z_m t},\\
&\textit{Utility (Counter offer by m)}\\
&\quad = [[(O_m - R_m) + (O_m - M_m)]\\
&\qquad \times p_m^{O_m}(acc) + [\textit{Utility (Break-down by m)}\\
&\qquad \times p_m^{O_m}(rbd)] + [(O_{w_m} - R_m)\\
&\qquad \times p_m^{O_m}(rco)]] \times e^{-z_m t} \times \vartheta.
\end{aligned}
$$

Proposition 1. *The asymmetric reduction of perceived probability* (*different reduction rates of perceived probability corresponding to the offered prices*) *with time helps to accelerate the convergence within a given time deadline.*

Proof. In each bargaining session, the offered prices and the reserved valuation of both the negotiators are always fixed. Now, offered prices depend upon expected surplus which in turn depends upon the perceived probability of acceptance. So, as we decrease the perceived probability monotonically with time, the expected surplus also decreases monotonically. This motivates the bargainers to offer prices which are closer to their reserved valuation. □

Proposition 2. *If both the negotiators are rational, then they always come to an agreement* (*i.e. the game converges*) *as bargaining time* $t \to \infty$.

Proof. Both the negotiators offer prices which are closest to their opponents' reserved valuation at the start of each Bargaining Session. However, with rejections from the opponent they realize that their perceived probabilities of acceptance by the opponent is lower than what they believed. So, they offer a new price closer to their own reserved valuations following *Proposition* 1. Thus, the bargaining converges. Now, if we forcibly make $z_i = \infty$, (for $i = w, m$), when $t =$ deadline, the bargaining game is guaranteed to converge within the specified deadline. □

4.3. Model of interdependent attributes

In case of incomplete information bargaining, the bargainers do not reveal their computational strategies. Here each proposal and response provide some information about future predictions and decisions. They act as signals that help the bargainers to update their beliefs about what the other has computed. Each of them uses this information along with their own computations to determine its future computing actions, proposals and responses. Both the WAP Server and mobile device can directly observe all proposals and responses, but their computational actions are private. A bargainer can only try to guess this information from his opponent's states like acceptance (*acc*), rejection with counter-offer (*rco*), or rejection with breakdown (*rbd*).

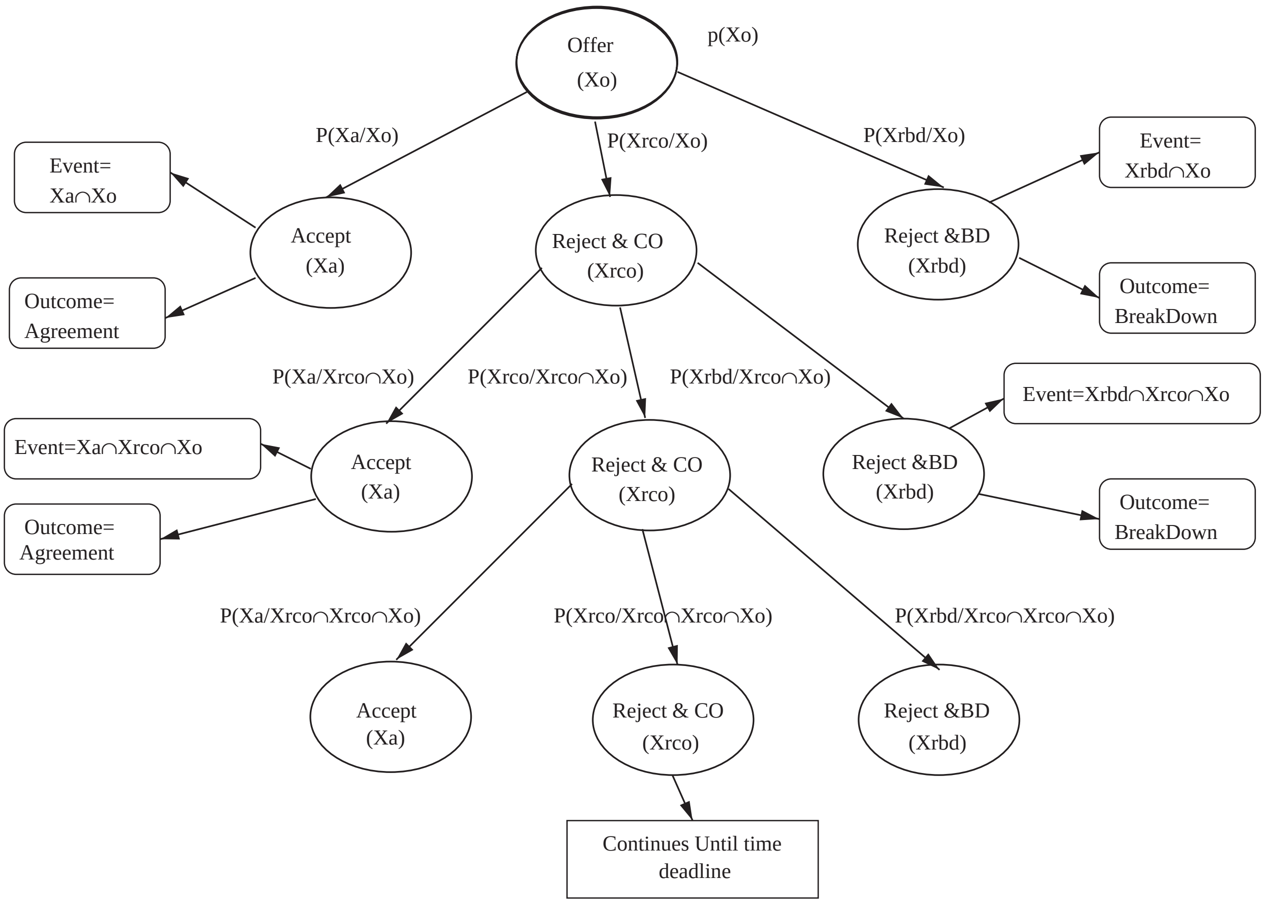


Fig. 8. A profile-based tree structure of alternating offers.

In our scenario, there are three types of perceived probabilities for each of the bargainers: perceived probability that an offer will be accepted by the opponent, an offer will be rejected by the opponent which in turn makes a counter offer or the opponent moves out of the game. The values of these three parameters depend upon the state of computation of the responder. Let the computational state at time t at the ith WAP Server w_i and jth mobile user m_j end are, respectively, $C_{w_i}(t)$ and $C_{m_j}(t)$. Let $A_i(t)$ be assumption of bargainer i at time t, which is a probability distribution over the set of states of computation that his opponent j may be in at time t. Let $a_i(C_{m_j}(t))$ be the probability that bargainer i believes that bargainer j is in his computational state $C_{m_j}(t)$. Now this bargainer updates its perceived probability depending on its own computational state and from his opponent's responses [15].

In Fig. 8 we represent a profile (history)-based tree structure of alternating offers with nodes representing some of the actions of WAP Server and mobile devices. Each branch represents the corresponding perceived conditional probability of each action. The pair of reservation values (R_i) are drawn from a continuous joint distribution function, $F_i(r_i)\ \forall\ i = 1, 2$, with a positive density f_i defined on $[0, 1] \times [0, 1]$. These distribution functions are common knowledge to the system. Each branch probability is calculated by a prediction of the opponent's next possible action. The bargainer i is able to compute the conditional probability $P[C_{m_j}(t-1) \rightarrow C_{m_j}(t)/C_{w_i}(t)]$ of bargainer j by traversing from the state $C_{m_j}(t-1)$ to $C_{m_j}(t)$ given its own computation state $C_{w_i}(t)$ at time t by reviewing the tree structure. So,

$$a_i(C_{m_j}(t)) = \sum_{C_{m_j}(t-1)} P[C_{m_j}(t-1) \rightarrow C_{m_j}(t)/C_{w_i}(t)] \times a_i[C_{m_j}(t-1)]. \quad (1)$$

Similarly, the bargainer j (i.e., the mobile device) computes the predicted probabilities as follows:

$$a_j(C_{w_i}(t)) = \sum_{C_{w_i}(t-1)} P[C_{w_i}(t-1) \rightarrow C_{w_i}(t)/C_{m_j}(t)] \times a_i[C_{w_i}(t-1)]. \quad (2)$$

We can observe that whenever a counter-offer comes from the opponent, the current player knows the opponent's state of computation (i.e., reject and counter-offer). Thus, we know that $C_{m_j}(t-1)$ = Reject & CO. And correspondingly, we can write Eq. (1) as

$$a_i(C_{m_j}(t)) = P[C_{m_j}(t-1) \to C_{m_j}(t)/C_{w_i}(t)] \times a_i[C_{m_j}(t-1)] \tag{3}$$

and from Eq. (2) we get,

$$a_j(C_{w_i}(t)) = P[C_{w_i}(t-1) \to C_{w_i}(t)/C_{m_j}(t)] \times a_j[C_{w_i}(t-1)]. \tag{4}$$

5. Job allocation scheme

We consider a single class job distributed system consisting of n mobile devices. The WAP Server knows the price per unit resource p_i and processing rate μ_i of the ith mobile device. If different grid users assign jobs to the same WAP Server, then the WAP will have to maintain price vectors for each of the grid users. This is because, different grid users will have different reserved valuations and hence the bargaining game will give different results each time. Also, the bargaining game is played offline, i.e., the WAP server is assumed to have an idea of the reserved valuations of different grid users, and it will have its price vectors ready before the job allocation is done. Here, we will be allocating jobs coming from a particular grid user. We assume that the mobile devices deal with two resources, namely idle CPU cycles and buffer size (required for queueing up jobs) with equal weight. In other words, p_i = price per unit CPU cycle = price per unit buffer size of the ith mobile device.

Modeling each mobile device as an M/M/1 queue, the expected execution time [25] over all jobs executed by the system is given by $\sum_{i=1}^{n} \frac{\beta_i}{\Phi(\mu_i-\beta_i)}$ where, μ_i is the average processing rate and β_i is the average job arrival rate at mobile device i and Φ is the total job arrival rate at the WAP Server. The execution time at every node comprises a queueing delay and an actual processing delay. We assume a constant k_i which maps the execution time to the amount of resources (both CPU cycles and buffer size) consumed at node i. Thus, the price to get β_i amount of work performed at the ith node is given by $\frac{k_i p_i \beta_i}{\Phi(\mu_i-\beta_i)}$ and the overall cost of the system is given by

$$\mathcal{C}(\beta_i) = \sum_{i=1}^{n} \frac{k_i p_i \beta_i}{\Phi(\mu_i - \beta_i)}. \tag{5}$$

Now, our objective is to find an efficient job allocation scheme $\{\beta_1, \beta_2, \ldots, \beta_n\}$ of the n mobile devices which will optimize the revenue of the grid community, by minimizing $\mathcal{C}(\beta_i)$. Eq. (5) can be written as

$$\mathcal{C}(\beta_i) = -\sum_{i=1}^{n} \frac{k_i p_i}{\Phi} + \sum_{i=1}^{n} \frac{k_i p_i \mu_i}{\Phi(\mu_i - \beta_i)}. \tag{6}$$

Therefore, $\mathcal{C}(\beta_i)$ is minimized if $\sum_{i=1}^{n} \frac{k_i p_i \mu_i}{\Phi(\mu_i-\beta_i)}$ is minimized. The job allocation should obey the following conditions:

$$\textit{Positivity}: \beta_i \geqslant 0, \quad i = 1, \ldots, n, \tag{7}$$

$$\textit{Conservation}: \sum\nolimits_{i=1}^{n} \beta_i = \Phi, \tag{8}$$

$$\textit{Stability}: \beta_i < \mu_i, \quad i = 1, \ldots, n. \tag{9}$$

Definition 1. The optimization problem is defined as

$$\text{Minimize } \mathcal{F}(\beta_i) = \sum_{i=1}^{n} \frac{k_i p_i \mu_i}{\Phi(\mu_i - \beta_i)}$$

subject to the following constraints: $\sum_{i=1}^{n} \beta_i = \Phi$, $\mu_i > \beta_i$ and $\beta_i \geqslant 0$.

The second constraint ensures that no individual grid node is saturated.

Theorem 1. *The objective function $\mathcal{F}(\beta_i)$ is minimized if*

$$\beta_i = \mu_i - \sqrt{k_i p_i \mu_i}\, \frac{\sum_{j=1}^{n} \mu_j - \Phi}{\sum_{j=1}^{n} \sqrt{k_j p_j \mu_j}} \tag{10}$$

subject to the constraints $\sum_{i=1}^{n} \beta_i = \Phi$ and $\mu_i > \beta_i$. The minimum value of $\mathcal{F}(\beta_i)$ is given by

$$\frac{\left(\sum_{i=1}^{n} \sqrt{k_i p_i \mu_i}\right)^2}{\sum_{i=1}^{n} \mu_i - \Phi}.$$

Proof. This is a non-linear programming problem which is solved by using Lagrange multiplier theorem as shown below. Let $a \leqslant 0, \eta_i \leqslant 0$, for $i = 1, \ldots, n$ be the Lagrange multipliers. The corresponding Lagrangian function for the problem is given by

$$L(\beta_i, a, \eta_i) = \sum_{i=1}^{n} \frac{k_i p_i \mu_i}{\Phi(\mu_i - \beta_i)} + a\left(\sum_{i=1}^{n} \beta_i - \Phi\right) + \sum_{i=1}^{n} \eta_i(\beta_i - \mu_i).$$

The optimal solution satisfies the Kuhn–Tucker conditions,

$$\frac{\delta L}{\delta \beta_i} = \frac{k_i p_i \mu_i}{\Phi(\mu_i - \beta_i)^2} + a + \eta_i = 0, \quad \text{for } i = 1, \ldots, n, \tag{11}$$

$$\frac{\delta L}{\delta a} = \sum \beta_i - \Phi = 0, \tag{12}$$

$$\mu_i - \beta_i \geqslant 0, \quad \eta_i(\mu_i - \beta_i) = 0, \quad \eta_i \leqslant 0, \quad \text{for } i = 1, \ldots, n. \tag{13}$$

Since $\mu_i - \beta_i > 0$, Eq. (13) gives $\eta_i = 0$. Thus, Eq. (11) reduces to

$$\frac{k_i p_i \mu_i}{\Phi(\mu_i - \beta_i)^2} + a = 0 \quad \text{for } i = 1, \ldots, n \tag{14}$$

$$\Rightarrow \sqrt{-a\Phi} = \frac{\sum \sqrt{k_i p_i \mu_i}}{\sum \mu_i - \Phi}, \tag{15}$$

Eqs. (14) and (15) give

$$\beta_i = \mu_i - \frac{\sqrt{k_i p_i \mu_i}(\sum_{j=1}^{n} \mu_j - \Phi)}{\sum_{j=1}^{n} \sqrt{k_j p_j \mu_j}}. \qquad \square$$

If we also introduce the condition that β_i should be non-negative, *Theorem* 1 cannot guarantee a solution. Note that β_i becomes negative when $\sqrt{\mu_i} < \sqrt{k_i p_i} \frac{\sum_{j=1}^{n} \mu_j - \Phi}{\sum_{j=1}^{n} \sqrt{k_j p_j \mu_j}}$, signifying the mobile device i is too slow to carry out the job allocated to it. So, we set $\beta_i = 0$ implying that no job is allocated to the ith mobile device. We then eliminate the ith mobile device from consideration and recompute the workload allocation for the other $(n-1)$ devices. This process is continued iteratively until a feasible solution is found. Our algorithm closely follows the COOP algorithm [11] and has $O(nlogn)$ running time. It is described below:

PRIMAL Algorithm. **Pri**ce-based opti**mal** workload allocation scheme.

Input: The average processing rates of mobiles: $\{\mu_1, \mu_2, \ldots, \mu_n\}$.

Total job arrival rate Φ.

The price per unit resource vector: $\{p_1, p_2, \ldots, p_n\}$.

The constants vector: $\{k_1, k_2, \ldots, k_n\}$.

Output: The optimal job allocation$\{\beta_1, \beta_2, \ldots, \beta_n\}$.

1. Sort the mobile devices in decreasing order of $\frac{\mu_1}{k_1 p_1} \geqslant \frac{\mu_2}{k_2 p_2} \geqslant \cdots \geqslant \frac{\mu_n}{k_n p_n}$;
2. c$\leftarrow \frac{\sum_{i=1}^{n} \mu_i - \Phi}{\sum_{i=1}^{n} \sqrt{k_i p_i \mu_i}}$;
3. while $\left(c > \frac{\sqrt{\mu_n}}{\sqrt{k_n p_n}}\right)$ do

   $\beta_n \leftarrow 0$;
   $n \leftarrow n - 1$;
   $c \leftarrow \frac{\sum_{i=1}^{n} \mu_i - \Phi}{\sum_{i=1}^{n} \sqrt{k_i p_i \mu_i}}$;
4. for $i = 1, \ldots, n$ do
   $\beta_i \leftarrow \mu_i - c\sqrt{k_i p_i \mu_i}$;

The validity of this algorithm is proved by the following theorem:

Theorem 2. *If* $\sqrt{\mu_i} < \sqrt{k_i p_i} \frac{\sum_{j=1}^{n} \mu_j - \Phi}{\sum_{j=1}^{n} \sqrt{k_j p_j \mu_j}}$, *for i* $1 \leqslant i \leqslant n$, *then* $\mathcal{F}(\beta_i)$ *is minimized by setting* $\beta_i = 0$, *subject to the additional constraint* $\beta_i \geqslant 0$ *in addition to the two constraints stated in Theorem* 1.

Proof. Let $\frac{\mu_1}{k_1 p_1} \geqslant \frac{\mu_2}{k_2 p_2} \geqslant \cdots \geqslant \frac{\mu_m}{k_m p_m}$. If $\sqrt{\mu_m} < \sqrt{k_m p_m} \frac{\sum_{j=1}^{m} \mu_j - \Phi}{\sum_{j=1}^{m} k_j p_j \mu_j}$, the objective function is minimized when $\beta_m = 0$ subject to the additional condition $\beta_i \geqslant 0$. Proceeding in the above manner, we have

$$\frac{\delta L}{\delta \beta_i} = \frac{k_i p_i \mu_i}{\Phi(\mu_i - \beta_i)^2} + a + \eta_i = 0, \quad i = 1, \ldots, m-1, \tag{16}$$

$$\frac{\delta L}{\delta \beta_m} = \frac{k_m p_m \mu_m}{\Phi(\mu_m - \beta_m)^2} + a + \eta_m - \gamma_m = 0, \tag{17}$$

$$\frac{\delta L}{\delta a} = \sum \beta_i - \Phi = 0, \tag{18}$$

$$\mu_i - \beta_i \geqslant 0, \quad \eta_i(\mu_i - \beta_i) = 0, \quad \eta_i \leqslant 0, \quad \text{for } i = 1, \ldots, m, \tag{19}$$

$$\beta_m \geqslant 0, \quad \gamma_m \beta_m = 0, \quad \gamma_m \leqslant 0. \tag{20}$$

Since $\mu_i > \beta_i$, Eq. (19) gives $\eta_i = 0$ for $i = 1, \ldots, m$.

Thus, Eq. (16) reduces to

$$\frac{k_i p_i \mu_i}{\Phi(\mu_i - \beta_i)^2} + a = 0, \quad \text{for } i = 1, \ldots, m-1. \tag{21}$$

Eq. (17) reduces to

$$\frac{k_m p_m \mu_m}{\Phi(\mu_m - \beta_m)^2} + a - \gamma_m = 0. \tag{22}$$

Now, we have two situations:

Case 1: $\beta_m > 0$. Eq. (20) gives $\gamma_m = 0$, and from Theorem 1 we can infer

$$\beta_i = \mu_i - \frac{\sqrt{k_i p_i \mu_i}(\sum_{j=1}^{m} \mu_j - \Phi)}{\sum_{j=1}^{m} \sqrt{k_j p_j \mu_j}} \quad \text{for } i = 1, \ldots, m. \tag{23}$$

Case 2: $\beta_m = 0$. Eq. (20) gives $\gamma_m < 0$. Thus Eq. (22) reduces to

$$\frac{k_m p_m \mu_m}{\Phi(\mu_m - \beta_m)^2} + a = \gamma_m < 0. \tag{24}$$

Putting, $\beta_m = 0$ in Eq. (24) we get, $a < -\frac{k_m p_m}{\Phi \mu_m}$. Again, Eq. (24) gives

$$\sqrt{-(a\Phi)}(\mu_m - \beta_m) > k_m p_m \mu_m.$$

Taking summation on both sides, we get,

$$\sum k_j p_j \mu_j < \sqrt{-(a\Phi)}\left[\sum \mu_j - \Phi\right]. \tag{25}$$

Substituting the value of 'a' from Eq. (25), we get

$$\sqrt{\mu_m} < \frac{\sqrt{k_m p_m}\left[\sum_{j=1}^{m} \mu_j - \Phi\right]}{\sum_{j=1}^{m} k_j p_j \mu_j}. \qquad \square \tag{26}$$

In other words, the job allocation $\{\beta_1, \beta_2, \ldots, \beta_n\}$ given by PRIMAL is an optimal solution for the minimization problem stated in *Theorem* 1. We have assumed that the bandwidth of the wireless channel is always greater than that required to transfer the amount of job allocated to the corresponding mobile device. If, however, the bandwidth goes below the minimum requirement, the WAP Server will make $p_i = \infty$ for the corresponding mobile device i which automatically ensures that $\beta_i = 0$.

6. Performance evaluation

We developed a simulation platform to evaluate the performance of our Bargaining protocol with the proposed utility functions.

6.1. Assumptions

1. The WAP Server and mobile device do not know the other's reserved valuations at the start of the bargaining game.

2. Each player draws its reserved valuation independently using a random number generator and uses the same to guess the opponent's reserved valuation. The initial offered price of any negotiator is determined from the guessed reserved valuation in the following way. If the WAP Server starts the game, then $I_{O_w} = \alpha \times$ *minimum (guess* R_m, M_w) where, I_{O_w} = Initial offered price from WAP Server end; $guess\,R_m$ = WAP Server's guess of reserved valuation of Mobile device; M_w = Market Price of the resource known to WAP Server; α = a constant (we have assumed $\alpha = 0.5$). Thus, α is a measure of the amount of profit that the WAP wants to make. If the mobile device starts the game, then $I_{O_m} = \beta \times$ *maximum (guess* R_w, M_m) where, I_{O_m} = Initial offered price from mobile device, *guess* R_w = Mobile device's guess of reserved valuation of WAP Server; M_m = Market Price of the resource known to mobile device; β = a constant (we have assumed $\beta = 1.5$). Similarly, β is a measure of the amount of profit that the mobile device owner expects. The offered prices of the WAP Server follow a monotonically increasing function and those for the mobile devices follow a monotonically decreasing function. To keep the offered prices always in between the interval $(0, 1)$, we have considered the parameters of our negative exponential function in such a way that the array of offered prices of WAP Server is given by

$$O_w[i] = O_w[0] + (R_w - O_w[0]) \times (1 - \exp^{-(i\times 6.9/\delta)}),$$

where δ = total number of offered prices; $O_w[0]$ = initial offered price of WAP Server; $O_w[i]$ = the ith offered price of server. Similarly, for the mobile device we have

$$O_m[i] = R_m + (O_w[0] - R_m) \times \exp^{-(i\times 6.9/\delta)}.$$

3. The bargaining is based upon price per unit resource. Each bargainer makes an offer maximizing his revenue.

4. A WAP Server (buyer) is not allowed to make an offer which is less than his previous offered price, and a mobile device (seller) is not allowed to make an offer which is greater than his previous offered price within one bargaining session.

5. The predicted probability calculations have been simplified for our simulation purpose. We started with the same exponential distributions for the predicted probability of acceptance of the opponent as for the offered prices. Thus,

$$P_w[i] = P_w[0] + (1 - P_w[0]) \times (1 - \exp^{-i\times 6.9}),$$

$$P_m[i] = P_m[0] + (1 - P_m[0]) \times (1 - \exp^{-i\times 6.9}).$$

The predicted probabilities of break-down have been initially assumed to be (1—predicted probability of acceptance for that offered price). Thus, the predicted probabilities for counter-offer are initially 0 for all the possible prices to be offered.

At every step, the predicted probabilities of acceptance are updated as

$$P_w[i] = P_w[i] \times P_w[i] \times (1 - \exp^{-6.9\times\delta/t}),$$

where t = current time (measured in the current number of alternating offers). We assume that during every counter-offer, the probability of acceptance by the opponent decreases. The amount of this decrease, however increases exponentially with t. Thus, $P_{w_t}[i] = P_{w_{t-1}}[i] \times (1 - \exp^{-6.9\times\delta/t})$ and the actual predicted probability at time t, is given by

$$P_{w_t}[i] = P_{w_{t-1}}[i] \times P_{w_t}[i] \times (1 - \exp^{-6.9\times\delta/t})$$
$$\Rightarrow P_w[i] = P_w[i] \times P_w[i] \times (1 - \exp^{-6.9\times\delta/t}).$$

Similarly, for the mobile device we have,

$$P_m[i] = P_m[i] \times P_m[i] \times (1 - \exp^{-6.9\times\delta/t}).$$

The predicted probability of break-down vector is kept the same during the simulation, and the predicted probability of counter-offer is calculated by the expression: $1 - P_x^{O_x}(acc) - P_x^{O_x}(rbd)$. The results have been generated with $z_i = 0.5$,

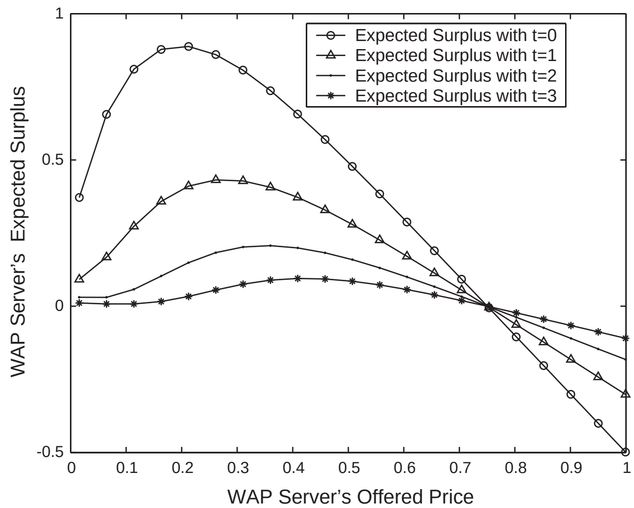


Fig. 9. Expected revenue of WAP Server vs. offered prices.

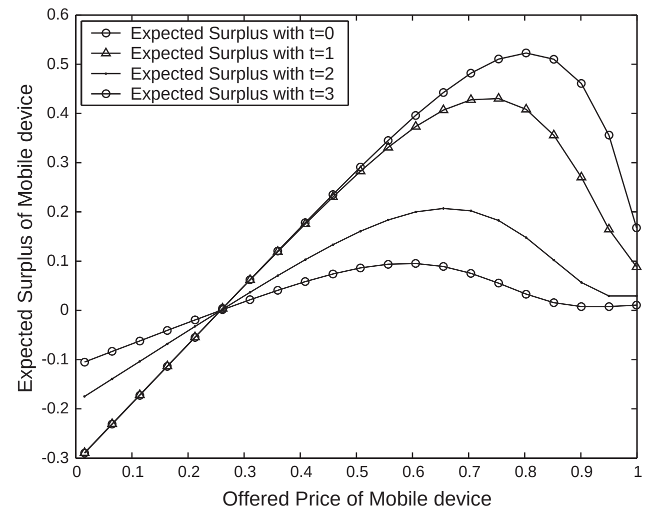


Fig. 11. Expected revenue of mobile device vs. offered prices.

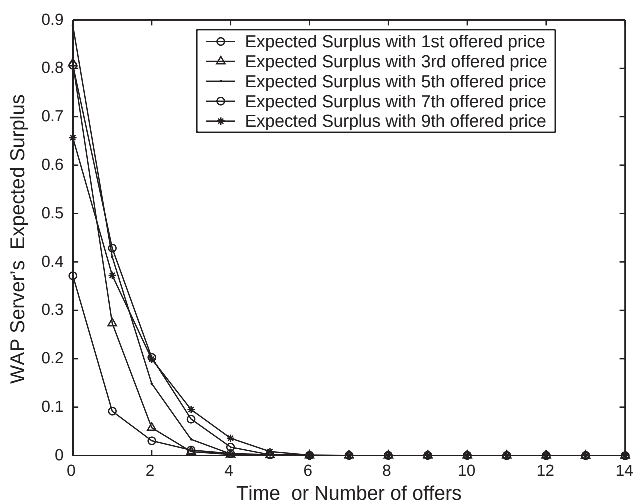


Fig. 10. Expected share of WAP Server vs. time with some fixed offered price.

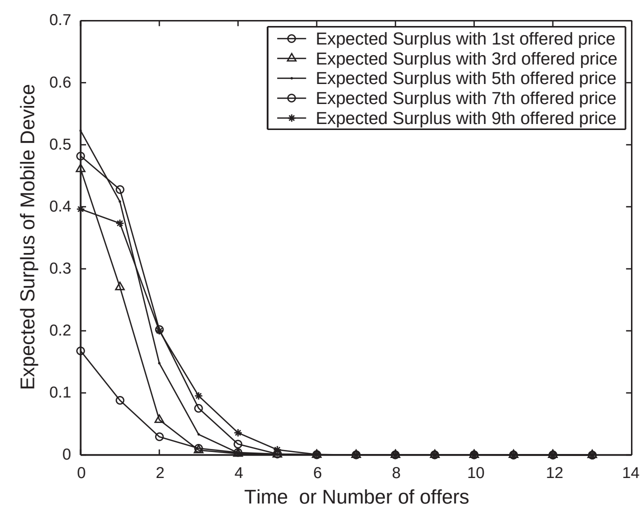


Fig. 12. Expected share of mobile device vs. time intervals with a fixed offered price.

when $t < deadline$ and $z_i = \infty$, when $t = deadline$, for $i = w, m$.

6.2. Simulation results

The goals of this evaluation are to measure the performance of our Bargaining Protocol based on our utility functions. We can observe that the nature of the plots for mobile devices and WAP Servers are the same. This is because we have assumed the same system parameters (i.e., z_i, M_x and $p_x^{M_x}$) for the two players. Thus the plots does not quite show the performance of the bargaining protocol with different values of these parameters. But, they depict the correctness of the utility functions and show the dependency of the expected surplus of the players on the system parameters.

Fig. 9 describes the expected revenue of WAP Server against each of its offered prices with time. We observe that the expected revenue is maximized in the middle of the first half of its offered prices. With increasing t, the revenue gradually decreases which is due to the nature of the utility functions, as the discount factor of the WAP Server accelerates the negotiation by decreasing the individual revenue with time. Fig. 10 shows the effect of the revenue with a fixed offered price against time. Again in this case, the revenue decreases with increasing t, and is maximum for the fifth offered price at $t = 0$. Fig. 11 presents the revenue earned by mobile devices against some of its offered prices. It can be observed from this graph that the revenue is maximized at the second half of its offered prices. Since the offered price of the mobile devices follow a monotonically decreasing function, its revenue decreases as the prices reach the reserved valuation. The revenue of the mobile device is much more during the initial phase of the bargaining. Fig. 12 represents the variation of the revenue of mobile device against time during the negotiation process. It gains the maximum

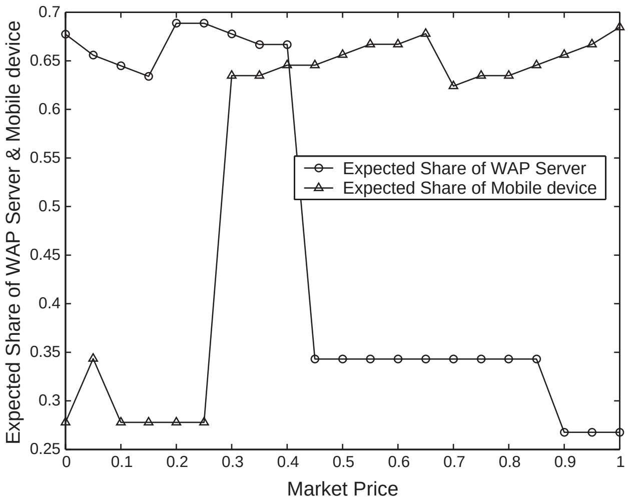


Fig. 13. Expected share of WAP Server and mobile device vs. market price.

revenue during the initial phase of the bargaining. Fig. 13 shows that as the market price increases in case of WAP Server, the expected share of the revenue also comes down. If the market price is low, the WAP Server does not have to offer high prices for getting the work done by mobile devices as there might be other clients in the market who can do the same work with less price. On the other hand, for a mobile device, if the market price of doing the job is high, it would not be offering lower prices during the bargaining session which in turn produces less revenue for the device. Because it knows that very few mobile devices are available for performing that particular computational task below a certain level of market price. But as the market price goes down, it has to offer lower prices for coming up with an agreement with the WAP Server which reduces its effective share of the revenue.

With this pricing strategy in place, we now simulate a heterogeneous grid system of 16 mobile devices under one WAP server with four different processing rates. The alternating offer bargaining game has already decided the price vector, p_i, for our job allocation algorithm. Also, we consider the case where no breakdown has occurred. Otherwise, for every breakdown, the corresponding mobile device will not be considered for job allocation. Table 6 gives the system configuration for our simulation environment. The first row gives the relative processing rate of the four types of mobile devices, i.e., it signifies how fast the mobile device is when compared to the slowest type. The second row gives the number of mobile devices belonging to each type. We arbitrarily assume the processing rates (third row) of the mobile device types, because it will not affect our results. The last row gives the values for k_i, the constant which maps the execution time at mobile device i to the amount of resources consumed at i. It can be seen that a faster mobile device has a higher k_i value because it will expect more price to perform the same amount of work as other slower devices

Table 6
System configuration

Relative processing rate	1	2	5	10
Number of mobile devices	6	5	3	2
Processing rate (jobs/sec)	0.013	0.026	0.065	0.13
k_i	1	2	3	4

and thus the "effective resources" consumed in that case is higher.

For comparison purposes, we have implemented the following two load-balancing schemes in addition to the proposed PRIMAL algorithm:

COOP scheme [11]: The cooperative load balancing (COOP) scheme is used to minimize the expected execution time of jobs from the mobile device perspective. The load allocation at each device is determined by minimizing the expected execution time at each mobile device and can be obtained by solving the following linear optimization problem:

$$\min_{\beta_i} \left\{ \frac{1}{\mu_i - \beta_i} \right\}, \quad \text{for } i = 1, \ldots, n$$

subject to the constraints (7–9). The minimization problem has been subsequently mapped to a cooperative bargaining theory problem and solved by obtaining the corresponding Nash bargaining solution (NBS).

OPTIM scheme [25]: This optimal static load balancing scheme is applied to minimize the overall expected execution time of jobs from a system, i.e., the grid user perspective.

Here, the load allocation is also obtained by solving a linear optimization problem given below:

$$\min_{\beta_i} \sum_{i=1}^{n} \frac{\beta_i}{\mu_i - \beta_i}, \quad \text{for } i = 1, \ldots, n$$

subject to the same constraints (7–9).

Fig. 14 shows the plots for total price that the grid user has to pay against system utilization. System utilization (ρ) is defined as the ratio of the total arrival rate to aggregate processing rate of the system: $\rho = \frac{\Phi}{\sum_{i=1}^{n} \mu_i}$. The price increases with increasing ρ (ranging from 10% to 90%) because the mobile devices get overloaded resulting in higher overall expected response time and subsequently higher cost for the grid user. Fig. 14 shows three curves corresponding to random, strictly decreasing and strictly increasing price vector (the mobile devices are initially numbered in decreasing order of their processing rates). The random price vector is the one obtained by our pricing strategy and the curve for the total price of a grid user for random price vector lies between that for the ascending and descending price vector cases. This can be explained by the fact that if the faster devices charge lesser prices (price vector in ascending order), then they will get the bulk of the work resulting in lesser overall response time and subsequently lesser total price for

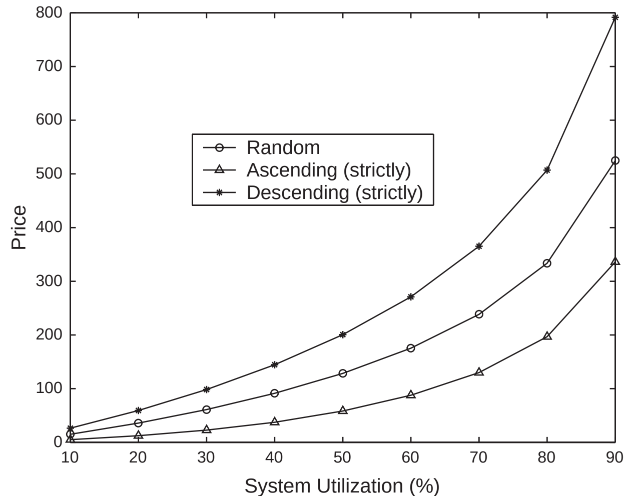


Fig. 14. Price vs. system utilization.

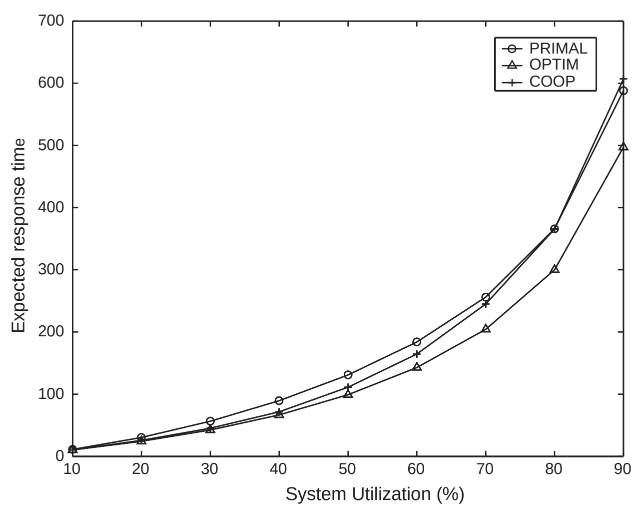


Fig. 15. Response time vs. system utilization.

the grid user. Similarly, if the faster devices charge more (price vector in descending order), then they will get lesser jobs resulting in greater total response time and subsequently greater price for the grid user.

Fig. 15 plots the overall expected response time against system utilization. We observe that PRIMAL performs equally well for smaller ρ when compared to COOP and OPTIM. With higher values of ρ, OPTIM gives a better performance and PRIMAL catches up with COOP when ρ goes beyond 70%.

Another important performance metric is the *fairness index* [14] which is defined by

$$I(\mathcal{C}) = \frac{\left[\sum_{i=1}^{n} \mathcal{C}_i\right]^2}{n\sum_{i=1}^{n} \mathcal{C}_i^2},$$

where $\mathcal{C}_i$ denotes the price that the WAP Server has to pay to the ith mobile device for the jobs it executes. Obviously, if every $I(\mathcal{C}_i) = 1$, the load is evenly balanced and the system is 100% fair. With increasing variability in $\mathcal{C}_i$, the index

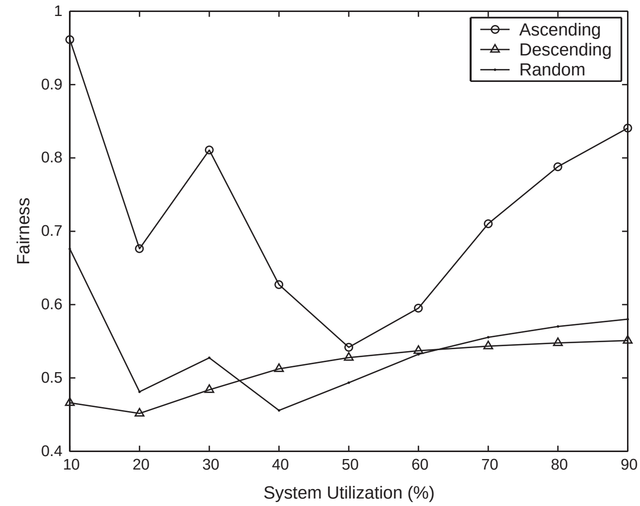


Fig. 16. Index of fairness vs. system utilization.

$I(\mathcal{C})$ decreases and the unfairness in the system increases. Fig 16 shows the change in fairness index for PRIMAL with increasing load in the system. Because the objective of PRIMAL is to reduce the overall cost for the jobs, it is an unfair scheme and the fairness index falls to as low as 0.48 for a descending price vector. Also, we can observe that when the price vector is in ascending order implying that the faster devices charge less, the fairness index varies between 0.8 and 0.9 at high load. This is because, more jobs are allocated to the faster devices such that the job processing time and the corresponding price charged by every device are quite close to each other.

We also study the effect of heterogeneity in the system on the total price that the WAP has to pay and the fairness index. Heterogeneity can be attributed to differences in processor speed, memory, I/O and bandwidth (which we have not considered in this work) of the individual devices. We will quantify heterogeneity by the *speed skewness* [24] defined by the ratio of maximum to minimum processing rates of the devices. We will use 16 devices in our simulation with 10 slower and six faster devices. The slower devices have a relative processing rate of 1, and that for the faster devices is varied from 1 (minimum heterogeneity) to 20 (maximum heterogeneity). Also, for simplicity, we assume k_i the same as the relative processing rate for the devices. The system utilization is kept constant at 60%.

Fig. 17 plots the overall response time of the jobs with increasing speed skewness. As the relative processing speed of the faster devices increases, the total response time goes on decreasing for all the three schemes as expected. OPTIM again performs better than the other two, but PRIMAL catches up with COOP when the relative processing speed is greater than 8. Fig. 18 plots the changes in total price charged by the mobile devices with increasing speed skewness for three different price vectors (as discussed before). Again, the ascending price vector performs better than random and descending price vectors. An interesting

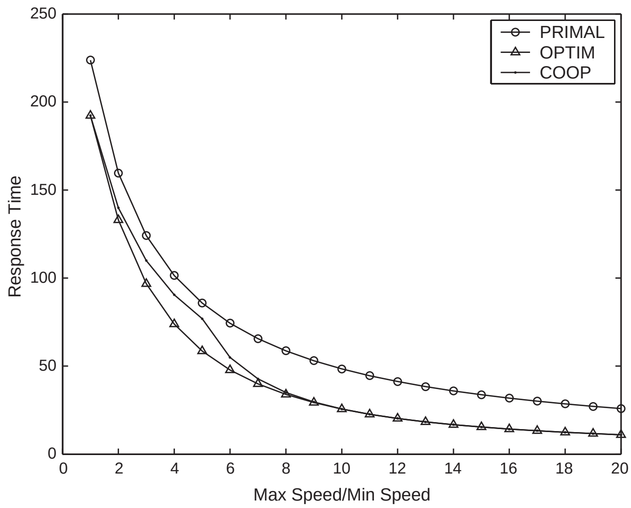


Fig. 17. Response time vs. speed skewness.

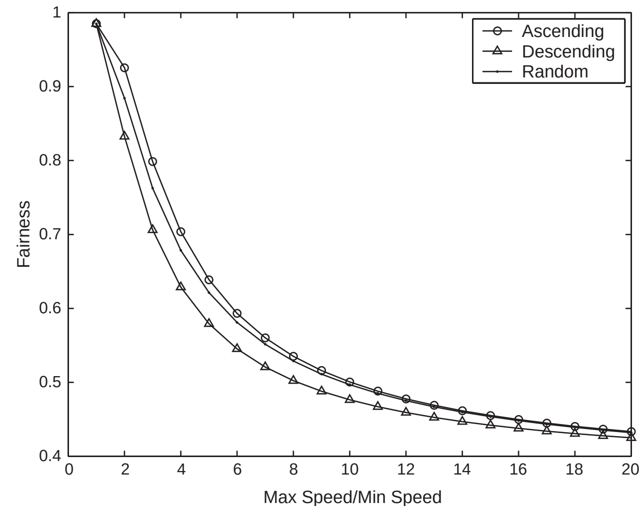


Fig. 19. Fairness vs. speed skewness.

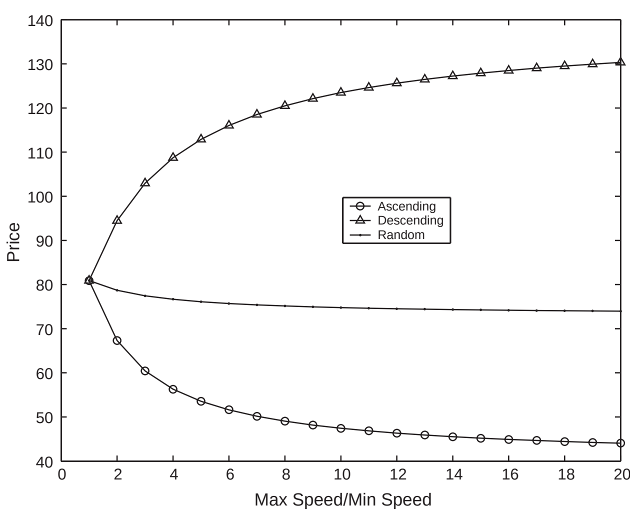


Fig. 18. Price vs. speed skewness.

point to note is that with a descending price vector, the total price increases for smaller values of the relative processing rate. This is because, the faster the devices, the lesser are the jobs allocated to them, resulting in an increase in the overall response time and a corresponding increase in price. Fig. 19 plots the fairness of PRIMAL with increasing speed skewness. Increasing heterogeneity in the system decreases the fairness to as low as 0.45 for a decreasing price vector at relative processing speed of 20. The ascending price vector still performs better than the other two, but the difference decreases with increasing skewness.

7. Conclusion

With the increasing demand for internet-connected wireless mobile devices and for the resource hungry computational grid, it is natural to propose efficient techniques to harness the processing power of millions of wireless devices. Till now various approaches like SETI@Home, Legion, GRACE, Globus and commercial ventures such as Parabon, Entropia have been proposed in normal grid environment. But no work has been done in mobile grid computing. We have designed an economic model and algorithmic framework so that resource hungry computational grids can buy and the mobile device can sell their computing power. Because of the inherent limitations in storage capacity and bandwidth availability of wireless devices, we have to motivate the skeptical device owners to contribute their mobile devices during off period. This potential restriction leads to the design of our pricing model in such a way that it maximizes the utility of both the grid community and all mobile users depending upon their respective strategies. In this paper, we have considered only two player games (one WAP server and one mobile device) for the time being. Multiplayer games will be more challenging due to the interaction between the mobile users under one WAP server. We plan to extend our work to model a $(n+1)$-player game between the WAP Server and the n mobile devices within its range to give a better approximation of the interactions between the players. Finally, the job allocation strategy can also be improved by considering a complex cost function that characterizes the mobile grid scenario in a better fashion.

Acknowledgments

This work was supported by NASA Ames Research Center under Cooperative Agreement Number NCC 2-5395 and NSF MRI grant EIA-0216500.

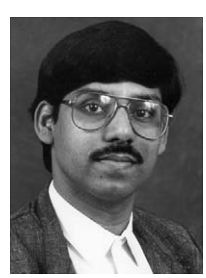
References

[1] Y. Amir, B. Awerbuch, A. Barak, S. Borgstrom, A. Keren, An opportunity cost approach for job assignment in a scalable computing cluster, IEEE Trans. Parallel Distrib. Systems 11 (7) (2000) 760–768.

[2] R. Buyya, Economic-based distributed resource management and scheduling for grid computing, Ph.D. Thesis, Monash University, Australia, April 2002, http://www.buyya.com/thesis/thesis.pdf
[3] R. Buyya, D. Abramson, J. Giddy, H. Stockinger, Economic models for resource management and scheduling in grid computing, Special Issue on Grid Computing Environments, The Journal of Concurrency and Computation: Practice and Experience (CCPE), Wiley Press, May 2002.
[4] K. Chatterjee, L. Samuelson, Bargaining with two-sided incomplete information: An infinite horizon model with alternating offers, Rev. Econ. Stud. 54 (1987) 175–192.
[5] B. Chun, Market-based cluster resource management, Ph.D. Dissertation, The University of California at Berkeley, USA, October 2001.
[6] J. Copic, C. Ponsati, Implementation by Decent Mechanisms, Department of Economics.
[7] K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith, S. Tuecke, A resource management architecture for metacomputing systems, Proceedings of the Fourth International Workshop on Job Scheduling Strategies for Parallel Processing, 30 March, 1998, Orlando, Florida, USA.
[8] I. Foster, C. Kesselman, The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufman, San Francisco, 1999.
[9] I. Foster, C. Kesselman, S. Tueke, The anatomy of the grid: Enabling scalable virtual organizations, Internat. J. Supercomput. Appl. (2001).
[10] P. Ghosh, N. Roy, K. Basu, S.K. Das, A game theory based pricing strategy for job allocation in mobile grids, Proceedings of the 18th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2004), Santa Fe, New Mexico.
[11] D. Grosu, A.T. Chronopoulos, M.Y. Leung, Load balancing in distributed systems: An approach using cooperative games, Proceedings of the 16th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2002), Ft Lauderdale, Florida, pp. 501–510.
[12] G. Heiser, F. Lam, S. Russell, Resource management in the Mungi single-address-space operating system, Proceedings of Australasian Computer Science Conference, 4–6 February, 1998, Perth, Australia, Springer, Singapore, 1998.
[13] IEEE 802.11 Technical Report, available at http://www.ieee802.org/11
[14] R.K. Jain, D.M. Chiu, W.R. Hawe, A quantitative measure of fairness and discrimination for resource allocation in shared computer system, Technical Report DEC-TR-301, Digital Equipment Corporation, Eastern Research Lab, 1984.
[15] K. Larson, T. Sandhlom, An alternating offers bargaining model for computationally limited agents, AAMAS'02, Bologna, Italy.
[16] Mojo-Nation, http://www.mojonation.net, June 2001.
[17] C. Mudgal, J. Vassileva, Bilateral negotiation with incomplete and uncertain information: A decision-theoretic approach using a model of the opponent, CIA 2000, pp. 107–118.
[18] N. Nisan, S. London, O. Regev, N. Camiel, Globally distributed computation over the internet: The POPCORN project, International Conference on Distributed Computing Systems (ICDCS'98), 26–29 May, 1998, Amsterdam, The Netherlands, IEEE CS Press, USA, 1998.
[19] M.J. Osborne, A. Rubinstein, Bargaining and Markets, Academic Press, Inc., New York, 1990.
[20] G. Owen, Game Theory, Second ed., Academic Press, New York, 1982.
[21] T. Phan, L. Huang, Chris Dulan, Challenge: Integrating mobile wireless devices into the computational grid, ACM MOBICOM'02, September 2002, Atlanta, USA.
[22] SETI@home homepage seti@home.ssl.berkeley.edu/
[23] M. Stonebraker, R. Devine, M. Kornacker, W. Litwin, A. Pfeffer, A. Sah, C. Staelin, An economic paradigm for query processing and data migration in Mariposa, Proceedings of the Third International Conference on Parallel and Distributed Information Systems, Austin, TX, USA, 28–30 September, 1994. Los Alamitos, CA, USA.
[24] X. Tang, S.T. Chanson, Optimizing static job scheduling in a network of heterogeneous computers, in: Proceedings of the International Conference on Parallel Processing, August 2000, pp. 373–382.
[25] A.N. Tantawi, D. Towsley, Optimal static load balancing in distributed computer systems, J. ACM 32(2) (April 1995), 373–382, 445–465.
[26] C. Waldspurger, T. Hogg, B. Huberman, J. Kephart, W. Stornetta, Spawn: A distributed computational economy, IEEE Trans. Software Eng. 18 (2) (February 1992) 103–117.
[27] P. Winoto, G. McCalla, J. Vassileva, An extended alternating-offers bargaining protocol for automated negotiation in multi-agent systems, Proceedings of the 10th International Conference on Cooperative Information Systems, (CoopIS'2002), Irvine, CA, November 2002, Lecture Notes in Computer Science, vol. 2519, Springer, Berlin, pp. 179–194.
[28] H. Yaiche, R.R. Mazumdar, C. Rosenberg, A game theoretic framework for bandwidth allocation and pricing in broadband networks, IEEE/ACM Trans. Networking 8 (5) (October 2000) 667–678.

Preetam Ghosh is currently a Ph.D. candidate in the Department of Computer Science and Engineering at the University of Texas at Arlington. He received his B.E. degree in 2000 from Jadavpur University, India, and his M.S. degree in 2004 from the University of Texas at Arlington, both in Computer Science and Engineering. His research interest includes resource management and load balancing issues in Mobile grids, scheduling in optical networks and QoS guarantees for multi-player online networked games.

Nirmalya Roy is currently a Ph.D. student in the CReWMaN Lab of Computer Science and Engineering Department, University of Texas at Arlington. He received his M.S. degree in 2004 from the University of Texas at Arlington and B.E. degree in 2001 from Jadavpur University, Calcutta, both in Computer Science and Engineering. His research interests include resource management in mobile and ubiquitous grid computing environment and Quality of Service (QoS) assurance in next generation ubiquitous grid infrastructure using game theoretic approach. He is a student member of IEEE and recipient of TxTEC Scholarship.



Sajal K. Das received B.Tech. degree in 1983 from Calcutta University, M.S. degree in 1984 from Indian Institute of Science, Bangalore, and Ph.D. degree in 1988 from the University of Central Florida, Orlando, all in Computer Science. Currently he is a Professor of Computer Science and Engineering and also the Founding Director of the Center for Research in Wireless Mobility and Networking (CReWMaN) at the University of Texas at Arlington (UTA). Prior to 1999, he was a professor of Computer Science at the University of North Texas (UNT), Denton where he founded the Center for Research in Wireless Computing (CReW) in 1997. Dr. Das is a recipient of the UNT Student Association's Honor Professor Award in 1991 and 1997 for best teaching and scholarly research; UNT's Developing Scholars Award in 1996 for outstanding research; and UTA's Outstanding Senior Faculty Research Award in Computer Science in 2001. He has visited numerous universities and research organizations worldwide for collaborative research and invited seminar talks. His research interests include resource and mobility management in wireless networks, mobile and pervasive computing, wireless multimedia and QoS provisioning, sensor networks, mobile Internet architectures and protocols, distributed processing and grid computing. He has published over 200 research papers in these areas and holds four US patents in wireless mobile networks. He received the Best Paper Awards in ACM MobiCom'99, ICOIN-16, ACM MSWiM 2000, and ACM/IEEE PADS'97. Dr. Das serves on the Editorial Boards of *Computer Networks*, *Journal of Parallel and Distributed Computing*, *Parallel Processing*, *Letters*, *Journal of Parallel Algorithms and Applications*, *IEEE Transactions on Mobile Computing and ACM/Kluwer Wireless Networks*. He served as General Chair of IEEE MASCOTS-2002, ACM WoWMoM 2000-02; General Vice Chair of IEEE PerCom-2003, ACM MobiCom-2000 and HiPC 2000-01; Program Chair of IWDC-2002, WoWMoM 1998-99; TPC Vice Chair of ICPADS-2002; and as TPC member of numerous IEEE and ACM conferences. He is a member of IEEE TCPP and TCCC Executive Committees and Advisory Boards of several cutting-edge companies.

Kalyan Basu received the B.E. degree in Electrical Engineering from Calcutta University, India, and the Master of Computer Technology degree from the Indian Institute of Technology, Delhi. He has been the managing director of CReWMaN Lab and a faculty member of the Computer Science and Engineering Department at the University of Texas at Arlington since January 2002. He has more than 30 years of telecommunication industry experience of which the last 27 years is on telecommunication research. Since 1980, he has worked at Nortel Telecommunications Research Division. He moved to BNR, Richardson, Texas, in 1989 to set up the wireless system engineering group and became the director of Wireless System Engineering in 1995. He left Nortel in early 2001 to become the vice president of system engineering for Yotta Networks, Plano, Texas, and lead the architecture team to design the Petabit Photonic Switch. He has received multiple US and international patents and published more than 50 papers in various international/national conferences. He is a member of the IEEE Computer Society.