Energy constrained resource allocation optimization for mobile grids

Li Chunlin*, Li Layuan

Department of Computer Science, Wuhan University of Technology, Wuhan 430063, PR China

A R T I C L E   I N F O

Article history:
Received 26 August 2008
Received in revised form 9 June 2009
Accepted 19 June 2009
Available online 24 June 2009

Keywords:
Mobile grid
Energy
Optimization

A B S T R A C T

A mobile grid incorporates mobile devices into Grid systems. But mobile devices at present have severe limitations in terms of processing, memory capabilities and energy. Minimizing the energy usage in mobile devices poses significant challenges in mobile grids. This paper presents energy constrained resource allocation optimization for mobile grids. The goal of the paper is not only to reduce energy consumption, but also to improve the application utility in a mobile grid environment with a limited energy charge, ensuring battery lifetime and the deadlines of the grid applications. The application utility not only depends on its allocated resources including computation and communication resources, but also on the consumed energy, this leads to a coupled utility model, where the utilities are functions of allocated resources and consumed energy. Energy constrained resources allocation optimization is formulated as a utility optimization problem, which can be decomposed into two subproblems, the interaction between the two sub-problems is controlled through the use of a pricing variable. The paper proposes a price-based distributed energy constrained resources allocation optimization algorithm. In the simulation, the performance evaluation of our energy constrained resources allocation optimization algorithm is conducted.

© 2009 Elsevier Inc. All rights reserved.

1. Introduction

Most of current grid environments do not take mobile devices into consideration. For the next generation Grids to be truly pervasive and mobile, we need to allow for the integration of mobile devices, and in order to leverage available resources and broaden the range of supplied services. Mobile devices are often resource limited: processing power is low, battery life is finite, and storage space is constrained. These restrictions slow application execution, and hinder operability. The use of mobile devices in grid environments may have two interaction aspects: devices are considered as the users of grid resources or as grid resources providers. Due to the limitation constraints on energy and processing capacity of mobile devices, their integration into the Grid as resource providers and not just consumers is very difficult [12,25,7,24,11,22,27,5,6,28,13,9]. There are three schemes for the integration of mobile devices into the Grid. First, mobile devices are merely interfaces to resources available in the Grid system, and do not contribute any services. Secondly, raw resources such as CPU, memory, and storage in mobile devices are used to complete the tasks in the grid environment. The second method just considers mobile devices as conventional resources to achieve goals. The third scheme is to exploit services in mobile devices to supporting mobile services in a mobile grid, this method is to enable mobile devices to contribute services. The third model is the one that provides the most complete integration where mobile devices can be both consumers and providers of services. Such an integration could open up possibilities in exploiting the mobile nature of these devices in a grid computing environment. Currently, research seldom focuses on mobile devices as resource providers in a mobile grid environment.

A mobile grid requires dynamic management of distributed resources, and such management needs to meet application quality requirements and prolong application lifetimes. The mobile grid application’s lifetime is determined by the available energy in the mobile devices. Mobile devices are battery-driven, and hence operate on an extremely frugal energy budget. So, battery energy limitation is the main challenge towards enabling persistent mobile grid computing. Energy minimization increases the time that mobile devices work properly without recharging or substituting batteries; energy efficiency is crucial to prolonging the lifetime of the mobile devices. But, current mobile grids lack an energy-efficient approach for exploiting mobile devices’ services.

This paper presents energy constrained resources allocation optimization for a mobile grid. The goal of the paper is not only to reduce energy consumption, but also to improve the application utility in a mobile grid environment with limited energy charge, ensuring the battery's lifetime and the deadlines of the grid applications. The application utility not only depends on its allocated resources including computation and communication resources, but also on the consumed energy, this leads to a coupled utility model,
where the utilities are functions of allocated resources and consumed energy. Energy constrained resources allocation optimization is formulated as a utility optimization problem, which is decomposed into two subproblems, the interaction between two sub-problems is controlled through the use of a pricing variable. The paper proposes a price-based distributed energy constrained resources allocation optimization algorithm. In the simulation, the performance evaluation of our energy constrained resources allocation optimization algorithm is conducted. To our knowledge, in the research area of energy-aware scheduling in mobile grid environments, this is the first solution that maximizes the grid system utility without exceeding the energy budget, execute budget and the deadline by using a pricing-based decomposition method.

The rest of the paper is structured as follows. Section 2 discusses the related works. Section 3 presents energy constrained resource allocation optimization in a mobile grid. Section 4 presents an energy constrained resources allocation optimization algorithm. In Section 5 the experiments are conducted and discussed. Section 6 gives the conclusions to the paper.

2. Related works

Mobile Grid computing is aimed at making Grid services available and accessible anytime anywhere from mobile devices, at the same time, ordinary grid users can exploit the limited resources of mobile devices. There are certain researches aiming to combining Grid environments with mobile devices [12,25,7,24,11,22,27,5,6,28,13,9], which incorporate mobile devices and terminals into the existing Grid systems. But mobile devices can consume services and share their resources since they are flexible, heterogeneous and energy limited. These limitations make incorporating mobile devices into a fixed Grid environment even more difficult.

Sameer Shiva et al. [25] focus on statically assigning resources in an ad hoc grid to an application composed of communicating subtasks. The goal of the allocation is to minimize the average percentage of energy consumed by the application to execute across the machines in the ad hoc grid, while meeting an application execution time constraint. Preetam Ghosh et al. [7] present a node mobility prediction framework based on a generic mobile Grid architecture. They formulate a cost effective job-scheduling scheme based on a predetermined pricing strategy at the wireless access point. Eunjeong Park et al. [24] propose to exploit a grid infrastructure to extend the battery life of mobile devices based on the contexts of mobile applications, devices, and grid systems. They present a framework, called SGA (selective grid access), to optimally utilize the limited resources of mobile devices and grids. SGA aims to reconfigure the placement of mobile applications on either the mobile device or grid nodes and to adjust the QoS level according to the current context of the residual energy and the resource availability of the grids. In [11], the concept of a decentralized job scheduler has been proposed which supports the integration of unstable mobile devices as computational grid resources. The concept has been applied to the demanding scenario of mobile ad-hoc grids. Scheduling decisions are delegated to the self-monitoring mobile worker peers that decide based on rules specified by policies. Otebolaku A.M. et al. [22] presented the current state of our work on mobile Grid clients’ adaptation mechanism and modeling. The overall goal of this adaptation mechanism is to reduce the response time between when a service request is made and when service is delivered to the service consumer. The model and adaptation mechanism would be able to considerably reduce the response time of the Grid service. V. Vetri Selvi et al. [27] presented a trace based mobility model in a mobile ad hoc grid to obtain the predictable position and stability time of a node in order to build a stable grid. They analyzed the performance of a mobile ad hoc grid both by using a theoretical model and by simulation. In the approach introduced by Chu et al. [5] for a mobile OGSLNET, each of the mobile devices runs its own Web server, and is, thus, a full-fledged participant in the grid. Kasula, Venkata Durga Kiran [6] proposes a layered system model to bridge the gap between the mobile and grid computing worlds. The model divides the complexities in mobile grid integration, among different components of different layers of the proposed model. The paper presents an efficient algorithm which addresses the problem of scheduling and disconnection. In paper [28], Sze-Wing Wong et al. presented the performance evaluation of Mobile Grid Services developed by using the MGS application-programming interface. The MGS API, constructed by combining an existing mobile agent system (JADE) and a generic grid system toolkit (Globus), is proposed to support the development of Mobile Grid Services (extended Grid services with mobility during execution). Konstantinos Katsaros et al. [13] investigated the fundamental issues arising in the path towards the realization of the Mobile Grid paradigm. They discussed various approaches in literature and pointed out the problems introduced by node mobility. They studied the performance of a hierarchical, campus-wide networking architecture based on appropriate traces.

Energy efficiency for high performance computing and communication system has recently become hot research area. Many works have been carried out on conserving energy, but few consider the energy in grid computing. Y. Huang et al. [9] present techniques for exploiting intermittently available resources in grid infrastructures to support QoS-based multimedia applications on mobile devices. They integrate power aware admission control, grid resource discovery, dynamic load-balancing and energy adaptation techniques to enable power deficient devices such as to run distributed multimedia applications. Ziliang Zong et al. [31] design energy efficient scheduling algorithms for parallel applications running on clusters, they propose a scheduling strategy called the energy efficient task duplication schedule, which can significantly conserve power by judiciously shrinking communication energy cost when allocating parallel tasks to heterogeneous computing nodes. Tarek A. AlEnawy et al. [3] propose to minimize the number of dynamic failures while remaining within the energy budget. They propose techniques to statically compute the speed of the CPU in order to meet the (m, k)-firm deadline constraints. Tao Xie et al. [29] address the issue of allocating tasks of parallel applications in heterogeneous embedded systems with the objective of energy-saving and latency-reducing. They proposed BEATA (Balanced Energy-Aware Task Allocation), a task allocation scheme considering both energy consumption and schedule length, is developed to solve the energy-latency dilemma. Kyong Hoon Kim et al. [8] provide power-aware scheduling algorithms for a bag of task applications with deadline constraints on DVS enabled cluster systems in order to minimize power consumption as well as to meet the deadlines specified by application users. Eunjeong Park et al. [23] designed an entire process of multimedia service composition for mobile computing. Their approach adapts the composition graph and the use of service routing for the context of mobile devices with the support of monitoring components. Chang-Qin Huang et al. [10] address power-Aware hierarchical scheduling in wireless Grids. The first level scheduler is responsible for mapping tasks among proxy-nodes and other fixed grid nodes; the second level scheduler conducts scheduling in each proxy-centric wireless domain. In the second-level scheduling algorithm, mobile node selection is targeted to minimize the energy consumed. Our work is different from [10] as we use utility optimization to solve energy constrained resources allocation problem in a mobile grid.

The works [16–20] mainly deal with resource allocation, QoS optimization in the computational grid and do not consider energy consumption for mobile grids. The methods and contributions of
this paper are different from above works. The goal of the paper is not only to reduce energy consumption, but also to improve the application utility in a mobile grid environment with limited energy charge, ensuring the battery lifetime and the deadlines of the grid applications. The work is different from other energy aware scheduling, which usually takes the consumed energy as constraints, our utility model regards consumed energy as one of the measures of the utility values, which indicates the tradeoff of application satisfaction and consumed energy.

3. Energy constrained resource allocation optimization for mobile grid

3.1. Mobile grid system

Fig. 1 illustrates a mobile grid system. It is based on a wireless network in which each cell consists of a number of mobile devices. Mobile devices residing in a cell of wireless networks are coordinated by a central entity that resides at the Access Point/Base Station, (BS) in order to perform a task. Mobile devices can be used as both resource consumers and resource providers. In a mobile grid, submitting jobs and receiving the results back is not straightforward, since power constraints and frequent disconnections are prevalent in wireless and mobile communications. The mobile grid proxy is used which acts as gateway to the grid. These mobile grid proxies undertake the role of the mediator between the mobile device and the grid system, and try to hide the instability of the wireless/mobile environment by acting on behalf of the mobile device. The devices provide the description of their capabilities and the degree of their availability to the BS. The BS is then responsible for decomposing incoming requests and scheduling the overall execution by providing specific tasks to each of the participating mobile devices. A request can either come from a mobile device or another grid client. BS can act as a mediator capable of hiding the heterogeneity of the participating devices from the requesting node, coordinating the overall execution of the submitted job and allowing the mobile device to appear to the rest of the network as an ordinary grid node. A grid node willing to provide service with resource capability and power is called a resource provider node (RPN) and the node which requests the service is called a user node (UN). The RPNs and UNs are the members of the grid. The nodes that are willing to share their resources specify the prices for their resources. The consumer node accepts a service based on the price, service time, etc. This leads to some negotiation between the user node (UN) and the resource provider node (RPN).

The paper formulates energy constrained resource allocation optimization in a mobile grid by adopting a computational economy framework. The proposed model consists of two types of agent: the grid resource agents that represent the economic interests of the underlying resources providers of the mobile grid, the grid user agents that represent the interests of grid user application using the grid to achieve goals. Interactions between the two agent types are mediated by means of market mechanisms. A market mechanism in economics is based on distributed self-determination, the variation of price reflects the supply and demand of resources, and market theory in economics provides a precise depiction of the efficiency of resource scheduling. Grid user agents are allowed to specify their requirements and preference parameters by a utility model. In our model, a utility function can be specified for each QoS dimension; we model each of these diverse requirements as a quality of service (QoS) dimension of a job. As a result, a market based mobile grid model inherently supports grid users with diverse requirements for the execution of their jobs. The utility values are calculated by the supplied utility function that can be formulated with the job parameters. The request is analyzed by the scheduler of grid market. Whenever a new grid user agent is created, it is first given an endowment of electronic cash to spend to complete its job. A job can be characterized by deadline, budget, data size and runtime requirements. The budget is the amount of money that the consumer promises to pay for the completion of the job. The grid market mechanism allows multiple grid resource agents and grid user agents to negotiate simultaneously, it uses price-directed approach to allocate appropriate grid resources. In this price-directed approach, an initial set of prices is announced to the grid user agent. Grid users can update their allocations based on the resource provider’s price policy, and iteratively approach an optimal solution. In each iteration, grid user agents individually determine their optimal allocation and communicate their results to the grid resource agents. Grid resource agents then update their prices and communicate the new prices to the user agents and the cycle repeats. Prices are then iteratively changed to accommodate the demands for resources until the total demand equals the total amount of resources available. We assume that when a grid user agent purchases a portion of resources owned by the resource agents, it is guaranteed that the user agent continues to receive the resource uninterrupted from the resource agent until its task is completed. Grid resource agents publish resource descriptions to the Grid Market. Resource providers compete actively for jobs from resource consumers and execute them for gaining profits. Every provider tries to maximize its profit based on its resource capability. We assume that the grid resource agents do not cooperate. Instead, they act non-cooperatively with the objective of maximizing their individual profits. The grid resource agents compete among each other to serve the grid user agents. The grid user agents do not collaborate either, and try to purchase as much grid resource as possible with the objective of maximizing their net benefit.

3.2. Mathematical formulation

The notations used in the following sections are listed in Table 1. In a mobile grid environment, the QoS of all applications running in mobile devices should be controlled, so that they do not exhaust the resources of the device on the mobile grid, including residual battery energy, memory and CPU capacity. It is important for the mobile grid system to manage energy consumption without compromising the system's performance. The paper considers energy constrained resource allocation optimization in mobile grid environment.

It is assumed that the mobile grid system consists of multiple grid sites that contain mobile nodes and ordinary fixed grid nodes. Mobile nodes consist of a collection of mobile devices $M$ connected by a wireless network. The set $M$ contains $n$ mobile devices, labeled as $m_1, m_2, \ldots, m_n$. Each site may contain multiple mobile nodes and computing nodes. The mobile devices in the grid system may have different resources such as networks, computing power and energy. A mobile device $m$ has an application set $A = \{A_1, A_2, \ldots, A_t\}$ and a resource set $R = \{R_1, R_2, R_n, \ldots\}, C_i$ is the available capacity...
of the resource $R$. The relation between the QoS and the resource consumption can be utilized to set dynamic QoS parameters. A mobile device estimates its energy consumption rate $e_{ri}$ for executing the application set $A = \{A_1, A_2, \ldots, A_i\}$, and the energy consumption constraint is $C_i$. For instance, the energy limitation of a mobile device imposes a constraint as follows:

$$e_{r_i} t_i \leq C_i.$$

where $t_i$ is the completion time of application $i$.

The processing power of a mobile device $m_i$ is measured by the average CPU speed. For any mobile device $m_i \in M$, there are grid jobs arriving at $m_i$. The jobs are assumed to be computationally intensive, mutually independent, and can be executed at any mobile device. As soon as a job arrives, it must be assigned to one mobile device for processing. When a job is completed, the executing mobile device will return the results to the originating mobile device or ordinary fixed grid node of the job. We use $J$ to denote the set of all jobs generated by grid application $i, J_i = \{j_1^i, j_2^i, \ldots, j_n^i\}$. Each grid job can be described as $J_i = (e_{r_i}, \{e_i\})$, in which $e_{r_i}$ stands for the time taken by the $i$-th grid application to complete the n-th job, $e_i$ stands for the energy dissipation of the n-th job. There are no dependencies among the jobs, so the submission order and completion order will not impact on the execution result. A user application set is represented as $A = \{A_1, A_2, \ldots, A_i\}$, for $1 \leq i \leq N$, grid application $A_i$ submits a job, together with parameters including: $T_i$, which is the deadline limit of job completion time, $B_i$, which is the expense budget limit for all jobs, and $E_i$, which is a limited energy budget for all jobs.

The energy consumption rate of each node in the system is measured by Joules per unit time. Let $e_{s_i}^n$ be the energy dissipation caused by grid application $i$'s nth job, $t_{r_i}^n$ is the execution time of job $n$ of grid application $i$ on the grid node. $e_r$ is the energy consumption rate of energy resource $i$. If the energy consumption is proportional to the execution time of job $n$, as is the case with battery energy. The energy dissipation of grid application $i$'s nth job can be written as below

$$e_{i}^n = e_r t_{r_i}^n.$$

We assume that the mobile grid has heterogeneous nodes with different system performance rates and network conditions. This means that the energy consumption of the mobile device can vary with the response time of the application and the network bandwidth. We denote $e_i^m$ as the consumed energy fraction of the energy $i$ (e.g. a battery) by grid application $i$. The total consumed energy of all grid applications $\sum_{i=1}^{I} e_i^m$ does not exceed the total capacity $C_0$ of energy $i$. Thus the following resource consumption constraint needs to be satisfied:

$$\sum_{i=1}^{I} e_i^m \leq C_0.$$

We define the energy consumption of each application $A_i$ as the sum of the energy consumed by $N$ grid jobs $\sum_{n=1}^{N} e_i^n$. The energy consumption of all grid jobs of each application $A_i$ should be less than the available resources of $e_i^m$ which is limited energy budget of grid user application $i$. For each grid application $A_i$, the consumed energy of all grid jobs of $A_i$ should satisfy

$$\sum_{n=1}^{N} e_i^n \leq e_i^m.$$

Now, we formulate the problem of energy constrained resource allocation optimization in mobile grid as a constraint optimization problem, the utility of the system $U_{syst}$ is defined as the sum of grid application utilities.

$$U_{syst} = \sum_{i=1}^{I} U_i(e_i, x_i^*, y_i^*)$$

where $e_i^m$ is the energy obtained by grid application $i$ from the energy $i$, $x_i^*$ is CPU allocation obtained by grid application $i$ from the computing resource provider $j$, $y_i^*$ is the bandwidth allocation obtained by grid application $i$ from the network resource provider $k$. The utility function for application $A_i$ depends on allocated resources $x_i^*$, $y_i^*$ and consumed energy $e_i^m$. The objective of energy constrained resource allocation optimization is to maximize the utility of the system $U_{syst}$ without exceeding the resource capacity, the energy budget, expense budget and the deadline. We formalize the problem using nonlinear optimization theory; the energy constrained resource allocation optimization in mobile grid can be formulated as follows.

$$\text{Max } U_{syst} = \sum_{i=1}^{I} U_i(e_i^m, x_i^*, y_i^*)$$

Subject to

$$\sum_{i=1}^{I} e_i^m \leq C_0, \quad C_j^r \geq \sum_{i=1}^{I} x_i^*, \quad C_{nk} \geq \sum_{i=1}^{I} y_i^*$$

$$B_i \geq \sum_{j=1}^{J_i} P_{e_i} + \sum_{j=1}^{J_i} P_{c_i} + \sum_{k=1}^{N} P_{n_k}$$

$$\sum_{n=1}^{N} e_i^n \leq e_i^m, \quad T_i \geq \sum_{n=1}^{N} t_{r_i}^n.$$

In the problem (3.1), the first type of constraint is related with different resource capacity. The QoS constraint implies that the aggregate network resource units $\sum_{i=1}^{N} y_i^*$ do not exceed the total capacity $C_{nk}$ of the network resource provider $k$, the aggregate consumed energy of all grid application $\sum_{i=1}^{I} e_i^m$ does not exceed the total $C_0$ of energy $i$, aggregate computing power $\sum_{i=1}^{I} x_i^*$ and does not exceed the total resource $C_j^r$ of the computing resource provider $j$. The second type of constraint is related with the grid application expense budget. A grid application needs to complete a sequence of jobs in a specified amount of time, $T_i$, while the payment overhead accrued cannot exceed $B_i$, which is the expense budget of grid application $i$. $P_{e_i}, P_{c_i}, P_{n_k}$ are the payments of the grid application $i$ to the energy storage provider $l$, computing resource provider $j$ and network resource provider $k$. The total payments of the grid application $i$ $\sum_{i=1}^{I} P_{e_i} + \sum_{i=1}^{I} P_{c_i} + \sum_{k=1}^{N} P_{n_k}$ does not exceed $B_i$. The total energy consumed by all jobs of grid application $i$ $\sum_{n=1}^{N} e_i^n$ cannot exceed the energy budget $e_i^m$ which is
the available energy obtained by grid application $i$ from the energy storage $l$.

We can apply the Lagrangian method to solve such a problem (3.1). The Lagrangian approach is used to solve the constrained optimization problems. Luh and Hoitomt [21] successfully adopted the Lagrangian approach by breaking the overall manufacturing problem into a series of sub-problems. This approach combines Lagrangian relaxation techniques with scheduling heuristics. Let us consider the Lagrangian form of the energy constrained resource allocation optimization problem:

$$L(e^l_i, x^l_i, y^l_i) = \sum_{i=1}^{I} U_i(e^l_i, x^l_i, y^l_i) - \lambda_i \left( \sum_{i=1}^{I} e^l_i - C_{e^l_i} \right) - \beta_i \left( \sum_{j=1}^{J} x^l_j - C_{x^l_j} \right) - \gamma_i \left( \sum_{j=1}^{J} Pe^{l_i}_j + \sum_{k=1}^{K} Pn^{l_i}_k - B_j \right) - \mu_i \left( \sum_{n=1}^{N} e^{n_i}_i - e^l_i \right) - \alpha_i \left( \sum_{n=1}^{N} t^{n_i}_n - T_i \right)$$

(3.2)

where $\lambda_i$, $\beta_i$, and $\psi_i$ are the Lagrange multipliers of grid application with their interpretation of energy price, computing resource capacity price, and network resource capacity price, respectively. Since the Lagrangian is separable, this maximization of Lagrangian over $(x^l_i, y^l_i, e^l_i)$ can be conducted in parallel at each application $A_i$. In problem (3.1), though the allocated resources $x^l_i, y^l_i$ and consumed energy $e^l_i$ are coupled in their constraints, respectively, they are separable. Given that the grid knows the utility functions $U$ of all the grid applications, this optimization problem can be mathematically tractable. However, in practice, it is not likely to know each application’s utility, and it is also infeasible for a mobile grid environment to compute and allocate resources in a centralized fashion. In order to derive a distributed algorithm to solve problem (3.1), we decompose the problem into the sub-problems.

In this paper, the maximization formulation of the grid system utility adopts a network utility maximization (NUM) framework [14] in which each application has an associated utility function. In [14], an optimization framework leads to a decomposition of the overall system problem into a separate problem for each user, in which the user chooses a charge per unit time that the user is willing to pay, and one for the network. The network’s optimization problem leads to two classes of algorithm, which may be interpreted in terms of either congestion indication feedback signals or explicit rates based on shadow prices. It was shown that a system optimum is achieved when users’ choices of charges and the network’s choice of allocated rates are in equilibrium.

The grid system utility denoted as the sum of grid application utility can be defined as follows (3.3):

$$U_{\text{system}} = \sum_{i=1}^{I} U_i(e^l_i, x^l_i, y^l_i) = \left( B_i - \sum_{j=1}^{J} Pe^{l_i}_j - \sum_{k=1}^{K} Pn^{l_i}_k \right) + \left( T_i - \sum_{n=1}^{N} t^{n_i}_n \right) + \sum_{j=1}^{J} \left( Pe^{l_i}_j \log e^l_i + Pe^{l_i}_j \log x^l_j + Pn^{l_i}_k \log y^l_k \right) + \left( e^l_i - \sum_{n=1}^{N} e^{n_i}_n \right).$$

(3.3)

Grid system utility functions are maximally optimized with specific constraints. In (3.3), $Pe^{l_i}_j \log e^l_i + Pe^{l_i}_j \log x^l_j + Pn^{l_i}_k \log y^l_k$ present the revenue of energy storage resource, computing power and network resource provider. We could have chosen any other form for the utility that increases with $x^l_j, y^l_k, e^l_i$. But we chose the log function because the benefit increases quickly from zero as the total allocated resource increases from zero and then increases slowly. Moreover, the log function is analytically convenient, increasing, strictly concave and continuously differentiable. The benefits of a grid resource provider are affected by payments of grid applications and allocated resources. It means that the revenue increases with increasing allocated resources and increasing payment.

The Lagrangian form of the problem (3.1) can be reformulated as follows (3.4):

$$L(e^l_i, x^l_i, y^l_i) = \left( B_i - \sum_{j=1}^{J} Pe^{l_i}_j - \sum_{k=1}^{K} Pn^{l_i}_k \right) + \left( T_i - \sum_{n=1}^{N} t^{n_i}_n \right) + \sum_{j=1}^{J} \left( Pe^{l_i}_j \log e^l_i + Pe^{l_i}_j \log x^l_j + Pn^{l_i}_k \log y^l_k \right) + \left( e^l_i - \sum_{n=1}^{N} e^{n_i}_n \right) - \lambda_i \left( \sum_{i=1}^{I} e^l_i - C_{e^l_i} \right) - \beta_i \left( \sum_{j=1}^{J} x^l_j - C_{x^l_j} \right) - \gamma_i \left( \sum_{j=1}^{J} Pe^{l_i}_j + \sum_{k=1}^{K} Pn^{l_i}_k - B_j \right) - \mu_i \left( \sum_{n=1}^{N} e^{n_i}_i - e^l_i \right) - \alpha_i \left( \sum_{n=1}^{N} t^{n_i}_n - T_i \right).$$

(3.4)

The system model presented by (3.1) is a nonlinear optimization problem with $N$ decision variables. Since the Lagrangian is separable, the maximization of the Lagrangian can be processed in parallel for grid user applications and grid resource providers respectively. From (3.4), the resource allocation $(e^l_i, x^l_i, y^l_i)$ solves problem (3.1) if and only if there exists a set of nonnegative shadow costs $(\lambda_i, \beta_i, \psi_i)$. Generally solving such a problem by typical algorithm such as steepest decent method and gradient projection method is of high computational complexity, which is very time costing and impractical for implementation. In order to reduce the computational complexity, we decompose the utility optimization problem (3.1) into two subproblems for grid user applications and grid resource providers so that the computational complexity is reduced. The shadow costs suggest a mechanism to distribute the resource optimization between the grid applications and the grid system. The problem (3.1) maximizes the utility of grid applications on the energy price, computing power capacity price, and network resource capacity price $\sum_{i=1}^{I} U_i(e^l_i, x^l_i, y^l_i)$ is the total utility of mobile grid system, $\beta_i \sum_{i=1}^{I} e^l_i$ is the computing power cost, $\lambda_i \sum_{i=1}^{I} e^l_i$ is the energy cost, $\psi_i \sum_{i=1}^{I} y^l_i$ is the network resource cost. By decomposing the Kuhn–Tucker conditions into separate roles of consumer and supplier at the grid market, the centralized problem (3.1) can be transformed into a distributed problem. A grid application’s payment is collected by the resource providers. The payments of grid applications paid to resource providers are the payments to resolve the optimality of resource allocation in the grid market. We decompose the problem into the following two subproblems (3.5) which is the grid user application QoS optimization problem and (3.6) which is the grid resource providers optimization problem, and seek a distributed solution where the grid resource provider does not need to know the utility functions of individual grid user application. Eqs. (3.5) and (3.6) derived from the distributed approach are identical to the optimal conditions given...
There exist applications as denoted by their resource conditions, while in Problem Sub 2, the grid resource provider adaptively allocates energy, CPU and bandwidth required by the grid application in the Problem Sub 1. The interaction between two sub-problems is controlled through the use of the price variable $\lambda_i$, $\beta_i$ and $\phi_i$, which is the energy price, computing power price, and network resource price charged from grid applications by grid energy resource, computing power and network resource. The interaction between two sub-problems also coordinates the grid application’s payment and the supply of grid resource providers.

Energy constrained resource allocation optimization problem involves variables from grid applications and resource providers. Lagrange relaxation and gradient optimization can be applied to decompose such an overall optimization problem into a sequence of two sub-problems, each only involving variables from the grid application and resource providers respectively. Interactions between the two sub-problems are through optimal price variables.

In Problem Sub 1, the grid application maximizes its satisfaction and gives the unique optimal payment to the resource provider under the energy budget, expense budget and the deadline constraint. The grid application optimization problem can be written as in Box II.

**Theorem 1.** There exist $Pe_i^*$, $Pc_i^*$, $Pn_j^*$ which are optimal payments of grid application $i$ paying for energy resource $l$, computing power $j$ and network resource $k$ to execute grid jobs under the completion time constraint.

The proof is in Appendix.

In problem Sub 2, different resource providers compute optimal resource allocation for maximizing the revenue of their own. Grid application $i$ submits the payment $Pe_i^*$ to the energy resource provider $i$, $Pn_k^*$ to network resource provider $k$ and $Pc_j^*$ to computing power provider $j$. $Pe_i^*$ presents the revenue obtained by the energy resource $l$ from grid application $i$. $Pc_j^*$ presents the revenue obtained by the computing power $j$ from grid application $i$. $Pn_k^*$ presents the revenue obtained by network resource $k$ from grid application $i$. The objective of resource providers is to maximize $Pe_i^* + Pn_k^* + Pn_k^* + Pn_k^*$. The grid resource providers cannot sell the resources to more grid applications than the total capacity.

In Problem Sub 1, the grid application adaptively adjusts its payments to computing power, network resource and energy based on the current resource conditions, while in Problem Sub 2, the grid resource provider adaptively allocates energy, CPU and bandwidth required by the grid application in the Problem Sub 1. The interaction between two sub-problems is controlled through the use of the price variable $\lambda_i$, $\beta_i$ and $\phi_i$, which is the energy price, computing power price, and network resource price charged from grid applications by grid energy resource, computing power and network resource. The interaction between two sub-problems also coordinates the grid application’s payment and the supply of grid resource providers.

Energy constrained resource allocation optimization problem involves variables from grid applications and resource providers. Lagrange relaxation and gradient optimization can be applied to decompose such an overall optimization problem into a sequence of two sub-problems, each only involving variables from the grid application and resource providers respectively. Interactions between the two sub-problems are through optimal price variables.

In Problem Sub 1, the grid application maximizes its satisfaction and gives the unique optimal payment to the resource provider under the energy budget, expense budget and the deadline constraint. The grid application optimization problem can be written as in Box II.

**Theorem 1.** There exist $Pe_i^*$, $Pc_j^*$, $Pn_k^*$ which are optimal payments of grid application $i$ paying for energy resource $l$, computing power $j$ and network resource $k$ to execute grid jobs under the completion time constraint.

The proof is in Appendix.

In problem Sub 2, different resource providers compute optimal resource allocation for maximizing the revenue of their own. Grid application $i$ submits the payment $Pe_i^*$ to the energy resource provider $i$, $Pn_k^*$ to network resource provider $k$ and $Pc_j^*$ to computing power provider $j$. $Pe_i^*$ presents the revenue obtained by the energy resource $l$ from grid application $i$. $Pc_j^*$ presents the revenue obtained by the computing power $j$ from grid application $i$. $Pn_k^*$ presents the revenue obtained by network resource $k$ from grid application $i$. The objective of resource providers is to maximize $Pe_i^* + Pn_k^* + Pn_k^* + Pn_k^*$. The grid resource providers cannot sell the resources to more grid applications than the total capacity.

In Problem Sub 1, the grid application adaptively adjusts its payments to computing power, network resource and energy based on the current resource conditions, while in Problem Sub 2, the grid resource provider adaptively allocates energy, CPU and bandwidth required by the grid application in the Problem Sub 1. The interaction between two sub-problems is controlled through the use of the price variable $\lambda_i$, $\beta_i$ and $\phi_i$, which is the energy price, computing power price, and network resource price charged from grid applications by grid energy resource, computing power and network resource. The interaction between two sub-problems also coordinates the grid application’s payment and the supply of grid resource providers.

Energy constrained resource allocation optimization problem involves variables from grid applications and resource providers. Lagrange relaxation and gradient optimization can be applied to decompose such an overall optimization problem into a sequence of two sub-problems, each only involving variables from the grid application and resource providers respectively. Interactions between the two sub-problems are through optimal price variables.

In Problem Sub 1, the grid application maximizes its satisfaction and gives the unique optimal payment to the resource provider under the energy budget, expense budget and the deadline constraint. The grid application optimization problem can be written as in Box II.

**Theorem 1.** There exist $Pe_i^*$, $Pc_j^*$, $Pn_k^*$ which are optimal payments of grid application $i$ paying for energy resource $l$, computing power $j$ and network resource $k$ to execute grid jobs under the completion time constraint.

The proof is in Appendix.

In problem Sub 2, different resource providers compute optimal resource allocation for maximizing the revenue of their own. Grid application $i$ submits the payment $Pe_i^*$ to the energy resource provider $i$, $Pn_k^*$ to network resource provider $k$ and $Pc_j^*$ to computing power provider $j$. $Pe_i^*$ presents the revenue obtained by the energy resource $l$ from grid application $i$. $Pc_j^*$ presents the revenue obtained by the computing power $j$ from grid application $i$. $Pn_k^*$ presents the revenue obtained by network resource $k$ from grid application $i$. The objective of resource providers is to maximize $Pe_i^* + Pn_k^* + Pn_k^* + Pn_k^*$. The grid resource providers cannot sell the resources to more grid applications than the total capacity.

In Problem Sub 1, the grid application adaptively adjusts its payments to computing power, network resource and energy based on the current resource conditions, while in Problem Sub 2, the grid resource provider adaptively allocates energy, CPU and bandwidth required by the grid application in the Problem Sub 1. The interaction between two sub-problems is controlled through the use of the price variable $\lambda_i$, $\beta_i$ and $\phi_i$, which is the energy price, computing power price, and network resource price charged from grid applications by grid energy resource, computing power and network resource. The interaction between two sub-problems also coordinates the grid application’s payment and the supply of grid resource providers.

Energy constrained resource allocation optimization problem involves variables from grid applications and resource providers. Lagrange relaxation and gradient optimization can be applied to decompose such an overall optimization problem into a sequence of two sub-problems, each only involving variables from the grid application and resource providers respectively. Interactions between the two sub-problems are through optimal price variables.

In Problem Sub 1, the grid application maximizes its satisfaction and gives the unique optimal payment to the resource provider under the energy budget, expense budget and the deadline constraint. The grid application optimization problem can be written as in Box II.

**Theorem 1.** There exist $Pe_i^*$, $Pc_j^*$, $Pn_k^*$ which are optimal payments of grid application $i$ paying for energy resource $l$, computing power $j$ and network resource $k$ to execute grid jobs under the completion time constraint.

The proof is in Appendix.

In problem Sub 2, different resource providers compute optimal resource allocation for maximizing the revenue of their own. Grid application $i$ submits the payment $Pe_i^*$ to the energy resource provider $i$, $Pn_k^*$ to network resource provider $k$ and $Pc_j^*$ to computing power provider $j$. $Pe_i^*$ presents the revenue obtained by the energy resource $l$ from grid application $i$. $Pc_j^*$ presents the revenue obtained by the computing power $j$ from grid application $i$. $Pn_k^*$ presents the revenue obtained by network resource $k$ from grid application $i$. The objective of resource providers is to maximize $Pe_i^* + Pn_k^* + Pn_k^* + Pn_k^*$. The grid resource providers cannot sell the resources to more grid applications than the total capacity.
Proof is in Appendix.

4. Energy constrained resource allocation optimization algorithm

The objective of the energy constrained resource allocation optimization algorithm is to maximize the utility of the grid system without exceeding the resource capacity, the energy budget, expense budget and the deadline. The proposed algorithm decomposes energy constrained resource allocation optimization problem into a sequence of two sub-problems via an iterative algorithm. In each iteration, in Problem Sub 2, the grid application computes the unique optimal payment to the resource provider under the energy budget, expense budget and the deadline constraint to maximize the grid application’s satisfaction. The grid application individually solves its energy resource price, computing power price and network resource price to complete its all jobs, adjusts its grid resource demand and notifies the grid resource provider about this change. In Problem Sub 2, different resource providers compute the optimal resource allocation for maximizing the revenue of their own. The grid resource provider updates its price according to optimal payments from grid application, and then sends the new prices to the grid applications and allocates the resource for grid application, and the cycle repeats. Fig. 2 is to show the activities of different parts of the Algorithm. The iterative algorithm that achieves energy constrained resource allocation optimization is described as follows.

Algorithm 1. Energy Constrained Resource Allocation Optimization Algorithm (ERAOA)

Routine 1

Input: new price $e_{i}^{(n+1)}$, $c_{j}^{(n)}$, $n_{k}^{(n)}$ from various resource providers

Step 1: Calculates optimal energy resource $Pe_{i}^{(n+1)}$ for energy resource $i$ to maximize the application’s utility $U_{app}(Pe_{i}, Pc_{j}, Pn_{k})$

$$Pe_{i}^{(n+1)} = \text{Max} \{ U_{app}(Pe_{i}, Pc_{j}, Pn_{k}) \}$$

Step 2: If $B_{i} \geq \sum_{j} Pc_{j}^{(n)} + \sum_{k} Pn_{k}^{(n)} + \sum_{i} Pe_{i}^{(n)}$

Then Return $Pe_{i}^{(n+1)}$ to energy resource $i$; Else Return Null;

Step 3: Calculates optimal computing power $Pc_{j}^{(n+1)}$ for computing power $j$ to maximze the application’s utility $U_{app}(Pe_{i}, Pc_{j}, Pn_{k})$

$$Pc_{j}^{(n+1)} = \text{Max} \{ U_{app}(Pe_{i}, Pc_{j}, Pn_{k}) \}$$

Step 4: If $B_{j} \geq \sum_{i} Pe_{i}^{(n)} + \sum_{k} Pn_{k}^{(n)} + \sum_{j} Pc_{j}^{(n)}$

Then Return $Pc_{j}^{(n+1)}$ to computing power $j$; Else Return Null;

Step 5: Calculates optimal network resource demand $Pn_{k}^{(n+1)}$ for network resource $k$ to maximize the application’s utility $U_{app}(Pe_{i}, Pc_{j}, Pn_{k})$

$$Pn_{k}^{(n+1)} = \text{Max} \{ U_{app}(Pe_{i}, Pc_{j}, Pn_{k}) \}$$

Step 6: If $B_{k} \geq \sum_{j} Pc_{j}^{(n)} + \sum_{i} Pe_{i}^{(n)} + \sum_{k} Pn_{k}^{(n)}$

Then Return $Pn_{k}^{(n+1)}$ to network resource $k$; Else Return Null;

Output: $Pe_{i}^{(n+1)}$ to energy resource $i$; $Pc_{j}^{(n+1)}$ to computing power $j$; $Pn_{k}^{(n+1)}$ to network resource $k$.

Routine 2

Input: optimal payments $Pe_{i}^{(n)}$, $Pc_{j}^{(n)}$, $Pn_{k}^{(n)}$ from grid application $i$
the JAVASIM network simulator (CPU and memory). The grid simulator is implemented on top of work state (latency and bandwidth), and the system loading state (battery power) state, then the monitoring. The proposed simulator considers mobile grid environment parameters such as energy constrained resource allocation optimization algorithm (ERAOA). Oursimulatorsupportsatopology of multiple LANs containing wired networks and wireless LANs, and bandwidth monitoring. The proposed simulator considers mobile grid environment parameters such as the battery (power) state, the network state (latency and bandwidth), and the system loading state (CPU and memory). The grid simulator is implemented on top of the JAVASIM network simulator [1]. In order to simulate the dynamics and heterogeneity of the Grid, all values of networks can be changed after topology generation. Network generator BRITE [2] generates the computer network topology. BRITE is a random network topology generator used to generate the simulation test bed. In the simulator, different agents are used namely resource provider agents, user agents and the grid scheduler agent which implements resource allocation and scheduling algorithm. The grid scheduler receives the task request, schedules the tasks to the host nodes, and then writes the scheduling records to the files for statistical analysis. The grid scheduler starts a listening thread that listens to the task requests. It receives the task requirements and puts them into the task queue. When the task queue is not empty, the grid scheduler starts the scheduling algorithm to find the right match. When resource agent updates its price, the resource agent forwards the price to user agents; the resource price is put in a packet. Whenever the new price packet passes to user agent, the user agent calculates the utility. According to the algorithm, if the price becomes higher than its maximum willingness to pay, the user agent does not buy the grid resource. The user agent can be informed the price for the next iteration by the next price packets.

We simulate a mobile grid environment with a 2 dimensional area of 500 m × 500 m to study the mobile device’s behavior. Each mobile device in the simulated environment has a maximal radio range of 100 m, and moves following a random-walking mobility model. Mobile devices dynamically enter and leave the mobile grid. These devices move and trade energy resource, computing power and network bandwidth by using the ERAOA algorithm described in Section 4. There is a number of parameters associated with each device such as an energy budget, an expense budget, and a two-dimension position value. Each mobile device’s battery capacity is initialized with a random value in the range of [700, 800], and reduced automatically by a random value in the range [0, 5] in each iteration of the ERAOA algorithm. There are five mobile devices clustered each one servicing 15 mobile devices, all of which contribute resources to the Grid community. The average speed of each device is 5 m/s. The average distance between neighboring devices is 25 m. A LAN consists of 80 nodes all of which contribute resources to the grid. The LAN acts as the main Grid infrastructure into which we want to integrate mobile devices. Device schedulers reside in WLANs, acting as the interface point between the mobile devices. The selective grid applications for simulation are computation-intensive applications such as image processing applications and mpeg players. We assume that each grid application can use any of grid resources including computation, communication and energy resources. The processor capacity varies from 220 to 580 MIPS. The wireless network bandwidth is from 10 kbps to 1 Mbps. The main memory is set to 128 M, 256 M, 512 M, and 2 G. The disk capacity is set to 80 G, 30 G, 20 G. The simulator leaves each application on the mobile device or delegates it to a grid node. There is a total of 150 resources and 600 applications are taken for experimental evaluation of the system. Energy consumption is represented as a percentage of the total energy required to meet all job deadlines [30,15]. Assume that the maximum power, \( P_{\text{max}} \), corresponds to running all jobs with the maximum processing frequency. The maximum frequency is assumed to be \( f_{\text{max}} = 1 \) and the maximum frequency-dependent power is \( P_{\text{max}} = 1 \). When the energy budget for each interval is limited, we can only consume a fraction of \( P_{\text{max}} \) when processing requests during a given interval. Jobs arrive at each site \( s_i \), \( i = 1, 2, \ldots, n \) according to a Poisson process with rate \( \alpha \). To take into account the wide dispersion in the job sizes in real grid applications, the sizes of the jobs are taken randomly from the uniform distribution in the interval [1, 100]. The capacities of the resources were also chosen uniformly in the interval [50, 500]. The resource cost can be expressed in grid dollars that can be defined as the unit processing cost. The initial price of the resource is set from 10 to 500 grid dollars. Users submit their jobs with varying deadlines. The deadlines of users are chosen from 100 ms to 400 ms. The budgets of users are set from 100 to 1500 grid dollars. Each measurement is run 6 times with different seeds. Simulation parameters are listed in Table 5.

### Table 5

<table>
<thead>
<tr>
<th>Table 5</th>
<th>Samples of inputs of grid resources.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource</td>
<td>Energy capacity ( C^E )</td>
</tr>
<tr>
<td>1</td>
<td>0.2</td>
</tr>
<tr>
<td>2</td>
<td>0.4</td>
</tr>
<tr>
<td>3</td>
<td>0.7</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

5. Simulations and analysis

5.1. Simulation settings

In this section, we present the performance evaluation of our energy constrained resource allocation optimization algorithm (ERAOA). Our simulator supports a topology of multiple LANs connected through wired networks and wireless LANs, and bandwidth monitoring. The proposed simulator considers mobile grid environment parameters such as the battery (power) state, the network state (latency and bandwidth), and the system loading state (CPU and memory). The grid simulator is implemented on top of the JAVASIM network simulator [1]. In order to simulate the dynamics and heterogeneity of the Grid, all values of networks can be changed after topology generation. Network generator BRITE [2] generates the computer network topology. BRITE is a random network topology generator used to generate the simulation test bed. In the simulator, different agents are used namely resource provider agents, user agents and the grid scheduler agent which implements resource allocation and scheduling algorithm. The grid scheduler receives the task request, schedules the tasks to the host nodes, and then writes the scheduling records to the files for statistical analysis. The grid scheduler starts a listening thread that listens to the task requests. It receives the task requirements and puts them into the task queue. When the task queue is not empty, the grid scheduler starts the scheduling algorithm to find the right match. When resource agent updates its price, the resource agent forwards the price to user agents; the resource price is put in a packet. Whenever the new price packet passes to user agent, the user agent calculates the utility. According to the algorithm, if the price becomes higher than its maximum willingness to pay, the user agent does not buy the grid resource. The user agent can be informed the price for the next iteration by the next price packets.

We simulate a mobile grid environment with a 2 dimensional area of 500 m × 500 m to study the mobile device’s behavior. Each mobile device in the simulated environment has a maximal radio range of 100 m, and moves following a random-walking mobility model. Mobile devices dynamically enter and leave the mobile grid. These devices move and trade energy resource, computing power and network bandwidth by using the ERAOA algorithm described in Section 4. There is a number of parameters associated with each device such as an energy budget, an expense budget, and a two-dimension position value. Each mobile device’s battery capacity is initialized with a random value in the range of [700, 800], and reduced automatically by a random value in the range [0, 5] in each iteration of the ERAOA algorithm. There are five mobile devices clustered each one servicing 15 mobile devices, all of which contribute resources to the Grid community. The average speed of each device is 5 m/s. The average distance between neighboring devices is 25 m. A LAN consists of 80 nodes all of which contribute resources to the grid. The LAN acts as the main Grid infrastructure into which we want to integrate mobile devices. Device schedulers reside in WLANs, acting as the interface point between the mobile devices. The selective grid applications for simulation are computation-intensive applications such as image processing applications and mpeg players. We assume that each grid application can use any of grid resources including computation, communication and energy resources. The processor capacity varies from 220 to 580 MIPS. The wireless network bandwidth is from 10 kbps to 1 Mbps. The main memory is set to 128 M, 256 M, 512 M, and 2 G. The disk capacity is set to 80 G, 30 G, 20 G. The simulator leaves each application on the mobile device or delegates it to a grid node. There is a total of 150 resources and 600 applications are taken for experimental evaluation of the system. Energy consumption is represented as a percentage of the total energy required to meet all job deadlines [30,15]. Assume that the maximum power, \( P_{\text{max}} \), corresponds to running all jobs with the maximum processing frequency. The maximum frequency is assumed to be \( f_{\text{max}} = 1 \) and the maximum frequency-dependent power is \( P_{\text{max}} = 1 \). When the energy budget for each interval is limited, we can only consume a fraction of \( P_{\text{max}} \) when processing requests during a given interval. Jobs arrive at each site \( s_i \), \( i = 1, 2, \ldots, n \) according to a Poisson process with rate \( \alpha \). To take into account the wide dispersion in the job sizes in real grid applications, the sizes of the jobs are taken randomly from the uniform distribution in the interval [1, 100]. The capacities of the resources were also chosen uniformly in the interval [50, 500]. The resource cost can be expressed in grid dollars that can be defined as the unit processing cost. The initial price of the resource is set from 10 to 500 grid dollars. Users submit their jobs with varying deadlines. The deadlines of users are chosen from 100 ms to 400 ms. The budgets of users are set from 100 to 1500 grid dollars. Each measurement is run 6 times with different seeds. Simulation parameters are listed in Table 5.

5.2. Experiments and analysis

Experiments are conducted to compare our energy constrained resource allocation optimization algorithm (ERAOA) under the constraint of the energy budget, expense budget and the deadline with power-aware hierarchical scheduling algorithm [10]. Energy and deadline constrained scheduling have been studied in embedded systems [26], cluster systems [31,8] and real-time systems [3]. Currently in a mobile grid or wireless grid, energy and deadline constrained scheduling have seldom been studied. Some works focus on energy consumption [25,11,6,9], some works focus on deadline constrained scheduling [4]. But few works in grid scheduling consider both energy and deadline as constraints. The reason for choosing Reference [10] as the comparison is that both our work and Reference [10] deal with energy and deadline constrained scheduling for mobile grids. Reference [10] focuses on task scheduling and resource allocation in a wireless grid, whereas the primary objective is to improve the wireless QoS and minimize the total battery energy used to successfully accomplish a task, as well as to optimize the time deadline. In Reference [10],
Table 4
Sample price values of the resource providers and optimal payments of grid application.

<table>
<thead>
<tr>
<th>Iteration number</th>
<th>Resource provider</th>
<th>Grid application</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Energy price</td>
<td>Bandwidth price</td>
</tr>
<tr>
<td>0</td>
<td>$e_p^{(0)} = 10$</td>
<td>$n_p^{(0)} = 10$</td>
</tr>
<tr>
<td>1</td>
<td>$e_p^{(1)} = 22.3$</td>
<td>$n_p^{(1)} = 27$</td>
</tr>
<tr>
<td>2</td>
<td>$e_p^{(2)} = 37.4$</td>
<td>$n_p^{(2)} = 41.2$</td>
</tr>
<tr>
<td>3</td>
<td>$e_p^{(3)} = 52.1$</td>
<td>$n_p^{(3)} = 55$</td>
</tr>
<tr>
<td>4</td>
<td>$e_p^{(4)} = 73$</td>
<td>$n_p^{(4)} = 76.3$</td>
</tr>
<tr>
<td>5</td>
<td>$e_p^{(5)} = 94$</td>
<td>$n_p^{(5)} = 96.3$</td>
</tr>
</tbody>
</table>

Table 5
Simulation parameters.

<table>
<thead>
<tr>
<th>Simulation parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>200</td>
</tr>
<tr>
<td>Network area</td>
<td>500 m × 500 m</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random-walking mobility</td>
</tr>
<tr>
<td>Average speed of each device</td>
<td>5 m/s</td>
</tr>
<tr>
<td>Network transferring time</td>
<td>200 ms</td>
</tr>
<tr>
<td>Processor capability (MIPS)</td>
<td>[220, 580]</td>
</tr>
<tr>
<td>RAM (B)</td>
<td>[128 M, 256 M, 512 M, 2 G]</td>
</tr>
<tr>
<td>Hard disk (B)</td>
<td>[80 G, 30 G, 20 G]</td>
</tr>
<tr>
<td>Arrival time</td>
<td>[100 ms, 400 ms]</td>
</tr>
<tr>
<td>Total number of applications</td>
<td>600</td>
</tr>
<tr>
<td>Total number of resource providers</td>
<td>150</td>
</tr>
<tr>
<td>Reschedule interval</td>
<td>600 ms</td>
</tr>
<tr>
<td>Initial price of computing power (grid dollar)</td>
<td>[10, 500]</td>
</tr>
<tr>
<td>Deadline</td>
<td>[100, 400]</td>
</tr>
<tr>
<td>Expense budget</td>
<td>[100, 1500]</td>
</tr>
<tr>
<td>Energy budget</td>
<td>[0.1, 1.0]</td>
</tr>
<tr>
<td>Resource capacities</td>
<td>[50, 500]</td>
</tr>
<tr>
<td>Job arrival rate</td>
<td>[0.1, 0.6]</td>
</tr>
</tbody>
</table>

Fig. 3. Energy consumption ratio under various job arrival rates.

Fig. 4. Resource utilization under various job arrival rates.

a power-aware hierarchical scheduling model is proposed to efficiently utilize the energy of wireless nodes with respect to the Quality of Service (QoS). The first level scheduler is responsible for mapping tasks among proxy-nodes and other fixed grid nodes; the second will conduct scheduling in each wireless domain. The first level performs overall scheduling based on the FFSA algorithm (First Input First Service). A revised Min–Min heuristic algorithm is enforced to the second level in order to efficiently map tasks to wireless devices. In the second level algorithm, mobile node selection is targeted to minimize the energy consumed due to communication and computation. Reference [10] adopts the Min–Min algorithm, and the operation of the algorithm in Reference [10] is described as follows: A list of scheduling subtasks is created. Initially this list consists of subtasks with no predecessors. For each subtask $i$ in the above list, across all wireless nodes, it finds the node $j$ that gives the subtask its minimum value $O_{power}$, which is defined as the percentage of energy consumed by each node to complete the entire task, ignoring other subtasks in the list. From among all the subtask/node pairs, it finds the pair that gives the minimum value $O_{power}$, while meeting deadline $D$. The subtask found in the above step is then removed from the list of scheduling subtasks and is mapped to its paired machine. Update the time and energy availability of the wireless node on which the subtask is scheduled and also across all nodes that send global data items to the scheduled subtask. The set of scheduling subtasks is updated to include any other new subtasks.

In the simulation, we compare the energy constrained resource allocation optimization algorithm (ERAOA) with a power-aware hierarchical scheduling algorithm (denoted as PHSA in the simulation) [10] by varying job arrival rate, expense budget and job deadline to study how they affect the performance of two algorithms. The performance metrics include energy consumption ratio, resource utilization, deadline miss ratio and allocation efficiency. Job arrival rate ($a$) is the job arrival speed, which will affect the system load. Job arrival rate varies from 0.1 to 0.6. The job expense budget ($B$) is set from 100 to 1500. The job deadline ($T$) is set from 100 to 400. Deadline Miss Ratio ($DM$) is defined as the ratio of the number of jobs whose deadline constraints are not met over the total number of jobs. The energy consumption ratio ($EC$) is defined as the percentage of consumed energy among total available energy resources. Resource utilization ratio ($RU$) is defined as the percentage of allocated resources among all available resources. Allocation Efficiency ($AE$) is a measure of the efficiency of the allocation process, which is computed using the number of all requests and number of accepted requests. $AE = \frac{\text{requests}}{\text{requests accepted}}$.

The impacts of job arrival rate on energy consumption ratio, resource utilization, deadline miss ratio and allocation efficiency were illustrated in Figs. 3–6 respectively. Fig. 3 shows the effect of job arrival rate ($a$) on the energy consumption ratio. When the job arrival rate increases, more requests need to be processed within one interval and the energy consumption ratio increases. When increasing the job arrival rate by $a = 0.5$, the energy consumption ratio of ERAOA is as much as 26% more than $a = 0.2$. When increasing the job arrival rate by $a = 0.25$, the energy consumption ratio of ERAOA is close to that of PHSA. Fig. 4 shows that as the job arrival rate increases, the resource utilization ratio increases. When $a = 0.5$, the resource utilization of ERAOA is as
When job arrival rate is very large, many jobs will be sent to the system, and resources are busier. Compared with PHSA, the resource utilization of ERAOA decreases more slowly than PHSA when the job arrival rate decreases. When the job arrival rate is 0.1 ($a = 0.1$), the resource utilization of PHSA decreases to 28%, and the resource utilization of ERAOA decreases to 52%. Fig. 5 shows that the deadline miss ratio increases when the job arrival rate increases. When $a = 0.5$, the deadline miss ratio of ERAOA is as much as 11% higher than that of $a = 0.10$. The smaller $a$, is the lower the system load is; grid resources are available for grid users. The requirements of the users can be processed on time and these users experience higher user satisfaction. So the larger $a$ is, the higher is the deadline miss ratio. Under the same job arrival rate ($a = 0.5$), ERAOA has an 8% lower deadline miss ratio than PHSA. Fig. 6 shows when job arrival rate increases ($a = 0.5$), the allocation efficiency of ERAOA is as much as 12% less than that with $a = 0.1$. The allocation efficiency is larger when the job arrival rate $a$ is smaller. When the job arrival rate increases, the system load increases; some user’s requirements cannot be processed on time. Jobs with a low budget cannot be completed before the deadline; this leads to low allocation efficiency. Compared with PHSA, the allocation efficiency of ERAOA decreases more slowly than PHSA when the job arrival rate increases. When the job arrival rate is 0.6 ($a = 0.6$), the allocation efficiency of PHSA decreases to 39%, the allocation efficiency of ERAOA decreases to 70%.

Figs. 7–10 illustrated the effects of user expense budget on energy consumption ratio, resource utilization, deadline miss ratio and allocation efficiency. The grid application budget ($B$) is set from 100 to 1500. Fig. 7 shows the energy consumption ratio under different expense budgets. A larger budget enables the grid user application to use more energy resources with a high price to complete the task before its deadline. When expense budgets are high, the energy consumption ratio is high. When $B = 1000$, the energy consumption ratio of ERAOA is 40% more than energy consumption ratio of $B = 100$. Compared with PHSA, the energy consumption ratio of ERAOA is close to PHSA when the expense budget increases. When expense budget is 800 ($B = 800$), the energy consumption ratio of PHSA increases to 77%, and the energy consumption ratio of ERAOA is up to 76%. Considering the resource utilization, from the results in Fig. 8, as the expense budget is higher, the resource utilization becomes higher. When $B = 1000$, the resource utilization of ERAOA is as much as 22% more than resource utilization of $B = 100$. Because when the budget decreases quickly, the users will be prevented from obtaining expensive resources. When expense budget is 250 ($B = 250$), the resource utilization of PHSA decreases to 51%, the resource utilization of ERAOA decreases to 68%. Fig. 9 is to
show the effect of expense budget on deadline miss ratio. When increasing budget values, the deadline miss ratio becomes lower. A larger expense budget enables a grid user to afford more expensive resources to complete the task before its deadline. When the budget increases \( B = 1500 \), the deadline miss ratio of ERAOA is as much as 37% less than that with \( B = 100 \). Under the same expense budget \( B = 1000 \), ERAOA has 7% lower deadline miss ratio than PHSA. Considering the allocation efficiency, from the results in Fig. 10, when increasing budget values of \( B = 1000 \), the allocation efficiency of ERAOA is 22% more than that of \( B = 100 \). Under same budget value \( B = 1000 \), the allocation efficiency of ERAOA becomes 35% higher than PHSA. A larger budget brings out higher allocation efficiency. Because grid applications can use expensive resources to complete the tasks within the deadline, they can maximize their allocation efficiency.

How the deadline affects the energy consumption ratio, resource utilization, deadline miss ratio and allocation efficiency was illustrated in Figs. 11–14 respectively. Fig. 11 shows how the energy consumption ratio varies under different deadlines. From the results in Fig. 11, when the deadline is low, there is an intensive demand for the resources in a short time, so the user application should choose a more energy-consuming resource to process the jobs, the energy consumption ratio is high. However, when the deadline changes to being higher, it is likely that jobs can be completed before the deadline, so grid job considers using the energy saving resources to complete tasks to maximize the utility, when the energy consumption ratio is low. When deadline is 100 \( (T = 100) \), the energy consumption ratio of PHSA is 83%; the energy consumption ratio of ERAOA is 77%. The energy consumption ratio of ERAOA is close to PHSA when the deadline is low. For the resource utilization under different deadline constraints, from the results in Fig. 12, when increasing the deadlines, the impact on the resource utilization is obvious. A larger deadline values brings out higher resource utilization. When the deadline is 400 \( (T = 400) \), the resource utilization of ERAOA is 24% higher than PHSA. Under the same deadline, ERAOA has a higher resource utilization than PHSA. Because within certain deadlines, grid jobs in ERAOA can choose resources of different prices to maximize the grid application utility; so it can achieve good resource utilization. Fig. 13 is to show the effect of the deadline on deadline miss ratio. When the deadline is low, deadline miss ratios of PHSA and ERAOA are high. When increasing deadlines, deadline miss ratios of the two schemes become lower. Because under a low deadline, more jobs cannot be completed on time. When deadline is 100 \( (T = 100) \), the deadline miss ratio of PHSA increases to 72%, and the deadline miss ratio of ERAOA increases to 67%. From the results in Fig. 14, the allocation efficiency increases when the deadline increases. When the deadline is low, there is intensive demand for the resources in short time, some user’s requirements cannot be processed on time. Jobs with a low budget cannot be completed before the deadline; this leads to low allocation efficiency. When the deadline is 400 \( (T = 400) \), the allocation efficiency of ERAOA is 37% higher than \( T = 100 \). Compared with PHSA, the allocation efficiency of ERAOA decreases more slowly than PHSA when the deadline decreases. When deadline is 100 \( (T = 100) \), the allocation efficiency of PHSA decreases to 25%, the allocation efficiency of ERAOA decreases to 55%.

6. Conclusions

Incorporating mobile devices into grid systems can be achieved in two different ways. One is to enable mobile applications by using fixed grid resources as proxy machines. The other is to take mobile hosts as resource providers to support grid applications. In this paper, we try to address the problem of scheduling grid applications on energy constrained mobile devices. The paper formulates the problem by modeling prices of different resources based on factors such as application budgets, resource capacities and application completion times, etc. We define the utility function for the energy constrained resource allocation problem, and further consider two aspects of the utility function: one is the utility function for maximizing the benefits of grid applications and the other is for optimizing the revenue of resources. Using the utility functions, a price-based algorithm is proposed to solve the defined problem. In the simulation, the performance evaluation of our energy constrained resource allocation optimization is
evaluated. In the future, we will consider moving our method to a real platform to test its feasibility.

Acknowledgments

The authors thank the editor and the anonymous reviewers for their helpful comments and suggestions. The work was supported by the National Natural Science Foundation of China (NSF) under grants (No. 60773211, No. 60970064), Program for New Century Excellent Talents in University, China (NCET080806), and the National Science Foundation of HuBei Province under Grant No. 2008CD5335. Any opinions, findings, and conclusions are those of the authors and do not necessarily reflect the views of the above agencies.

Appendix

Proofs for Theorem 1. We assume that grid application $i$ submits $P_{ei}$ to energy resource $l$, $P_{ci}$ to computing power $j$, and $P_{ni}$ to network resource $k$. Then, $P_{ei} = [P_{ei}^1 \ldots P_{ei}^j]$ represents all payments of grid applications for energy resource $l$, $P_{ci} = [P_{ci}^1 \ldots P_{ci}^k]$ represents all payments of grid applications for computing power $j$, $P_{ni} = [P_{ni}^1 \ldots P_{ni}^k]$ represents all payments of grid applications for network resource $k$. Let $m_i = \sum P_{ci}^1 + \sum P_{ei}^2 + \sum P_{ni}^3$, $m_i$ is the total payment of the ith grid application. $N$ grid applications compete for grid resources with finite capacity. The resource is allocated using a market mechanism, where the partitions depend on the relative payments sent by the grid applications. Let $ep_i, cp_i, np_i$ denote the price of the resource unit of energy resource $l$, the price of the resource unit of computing power $j$ and network resource $k$ respectively. Let the pricing policy, $ep = (ep_1, ep_2, \ldots, ep_n)$, denote the set of resource unit prices of all the energy resources in the grid, $cp = (cp_1, cp_2, \ldots, cp_n)$, denote the set of resource unit prices of all the computing powers, $np = (np_1, np_2, \ldots, np_n)$ is the set of network resource unit prices. The ith grid application receives the resources proportional to its payment relative to the sum of the resource provider’s revenue. Let $e_i, x_i, y_i$ be the fraction of resource units allocated to grid application $i$ by energy $l$, computing power $j$ and network resource $k$.

The time taken by the ith grid application to complete nth job is:

$$t_{ni}^* = \frac{cq_i^np_i}{Cc_i Pe_i} + \frac{bq_i^np_i}{Cn_i Pe_i} + \frac{eq_i^ep_i}{Cp_i Pe_i}.$$

The energy dissipation used by the ith grid user to complete nth job is:

$$e_i^* = er_i t_{ni}^* = er_i \left( \frac{cq_i^np_i}{Cc_i Pe_i} + \frac{bq_i^np_i}{Cn_i Pe_i} + \frac{eq_i^ep_i}{Cp_i Pe_i} \right).$$

Problem Sub 1 can be reformulated as

$$\text{Max } U_{app} = \left( b_i - \sum_{j=1}^{l} P_{ej}^1 - \sum_{j=1}^{l} P_{ej}^2 - \sum_{k=1}^{K} P_{nk}^k \right)$$

$$+ \left( T_i - \sum_{n=1}^{N} \frac{cq_i^np_i}{Cc_i Pe_i} + \frac{bq_i^np_i}{Cn_i Pe_i} + \frac{eq_i^ep_i}{Cp_i Pe_i} \right)$$

$$+ \left( \sum_{j=1}^{l} \frac{cq_i^np_i}{Cc_i Pe_i} + \frac{bq_i^np_i}{Cn_i Pe_i} + \frac{eq_i^ep_i}{Cp_i Pe_i} \right).$$

The Lagrangian for the grid application’s utility is $L_i(P_{ei}, P_{ci}, P_{ni})$.

$$L_i(P_{ei}, P_{ci}, P_{ni}) = \left( b_i - \sum_{j=1}^{l} P_{ej}^1 - \sum_{j=1}^{l} P_{ej}^2 - \sum_{k=1}^{K} P_{nk}^k \right)$$

$$+ \left( T_i - \sum_{n=1}^{N} \frac{cq_i^np_i}{Cc_i Pe_i} + \frac{bq_i^np_i}{Cn_i Pe_i} + \frac{eq_i^ep_i}{Cp_i Pe_i} \right)$$

$$+ \left( \sum_{j=1}^{l} \frac{cq_i^np_i}{Cc_i Pe_i} + \frac{bq_i^np_i}{Cn_i Pe_i} + \frac{eq_i^ep_i}{Cp_i Pe_i} \right).$$


de 

where $\varepsilon_i, \sigma_i, \nu_i$ is the Lagrangian constant. From Karush–Kuhn–Tucker Theorem we know that the optimal solution is given

$$\partial L_i(P_{ei}, P_{ci}, P_{ni}) / \partial P_{ei} = 0$$

for $\varepsilon_i, \sigma_i, \nu_i > 0$.

$$\partial L_i(P_{ei}, P_{ci}, P_{ni}) / \partial P_{ci} = -1 - \nu_i + \frac{eq_i^ep_i}{Cc_i Pe_i} + er_i + \frac{eq_i^ep_i}{Cc_i Pe_i} + \frac{eq_i^ep_i}{Cc_i Pe_i}.$$

Let $\partial L_i(P_{ei}, P_{ci}, P_{ni}) / \partial P_{ci} = 0$ to obtain

$$P_{ci}^* = \left( \frac{1 + \varepsilon_i + \sigma_i + \varepsilon_i er_i eq_i^ep_i}{(1 + \nu_i)Cc_i} \right)^{1/2}.$$

Using this result in the constraint equation, we can determine

$$\theta = (1 + \varepsilon_i + \sigma_i + \varepsilon_i er_i) / (1 + \nu_i)$$

as

$$(\theta)^{-1/2} = \frac{T_i}{\sum_{m=1}^{N} \left( \frac{eq_m^ep_m}{Cc_m} \right)^{1/2}}.$$

We obtain $P_{ci}^*$

$$P_{ci}^* = \left( \frac{eq_i^ep_i}{Cc_i} \right)^{1/2} \frac{\sum_{m=1}^{N} \left( \frac{eq_m^ep_m}{Cc_m} \right)^{1/2}}{T_i}.$$
We obtain $Pc_{i}^*$

$$Pc_{i}^* = \left( \frac{cq_i^2cp_i}{Cc_i} \right) ^{1/2} \frac{1}{T_i} \sum_{m=1}^{N} \left( \frac{m^n cp_m}{Cc_m} \right) ^{1/2}.$$

It means that grid application wants to pay $Pc_{i}^*$ to computing power $j$ for the resource required to execute grid jobs under the completion time constraint.

$$\partial L_1(Pc_i, Pc_j, Pn_k)/\partial Pn_k = -1 + \frac{bq^n npk}{Cn_k(Pc_i)^2} + \epsilon r^n npk + \frac{bq^n npk}{Cn_k(Pc_i)^2} - v_i$$

$$+ \frac{bq^n npk}{Cn_k(Pc_i)^2} + \epsilon r^n npk + \frac{bq^n npk}{Cn_k(Pc_i)^2}.$$

Let $\partial L_1(Pc_i, Pc_j, Pn_k)/\partial Pn_k = 0$ to obtain

$$Pn_k = \left( \frac{1 + \epsilon r + \sigma_i + \epsilon r \epsilon_i}{1 + v_i} \right) \left( \frac{bq^n npk}{Cn_k} \right) ^{1/2}.$$

Using this result in the constraint equation, we can determine $\tau = (1 + \epsilon r + \sigma_i + \epsilon r \epsilon_i)/1 + v_i$ as

$$\tau^{-1/2} = \frac{1}{T_i} \sum_{m=1}^{N} \left( \frac{m^n cp_m}{Cc_m} \right) ^{1/2}.$$

We obtain $Pn_k^*$

$$Pn_k^* = \left( \frac{bq^n npk}{Cn_k} \right) ^{1/2} \frac{1}{T_i} \sum_{m=1}^{N} \left( \frac{m^n cp_m}{Cc_m} \right) ^{1/2}.$$

It means that the grid application wants to pay $Pn_k^*$ to network resource $k$ for the resource required to execute grid jobs under the completion time constraint. □

**Proofs for Theorem 2.** We take the derivative and the second derivative with respect to $x_k$: $U_{resource}(x_k) = Pe_i / x_k$, $U_{resource}''(x_k) = -Pe_i / x_k^2$.

$U_{resource}(x_k) < 0$ is negative due to $0 < x_k$. The extreme point is the unique value maximizing the revenue of energy provider. The Lagrangian for Problem Sub 2 is $L_2(e_i', x_i', y_i')$.

$$L_2(e_i', x_i', y_i') = \sum_{i}(Pe_i \log e_i + Pe_i \log x_i + Pn_i \log y_i)$$

$$+ \lambda_i \left( Ce_i - \sum_{i} e_i \right) + \beta_i \left( Cj_i - \sum_{i} x_i \right) + \phi_i \left( Cn_i - \sum_{i} y_i \right)$$

$$= \sum_{i}(Pe_i \log e_i + Pe_i \log x_i + Pn_i \log y_i - \lambda_i e_i - \beta_i x_i - \phi_i y_i)$$

$$+ \lambda_i Ce_i + \beta_i Cj_i + \phi_i Cn_i$$

where $\lambda_i, \beta_i, \phi_i$ is the Lagrangian constant. From Karush–Kuhn–Tucker Theorem we know that the optimal solution is given as $\partial L_2(e_i', x_i', y_i') / \partial e_i = 0$ for $\lambda_i, \beta_i, \phi_i > 0$.

Let $\partial L_2(e_i', x_i', y_i') / \partial e_i = 0$ to obtain $e_i' = Pe_i / \lambda_i$.

Using this result in the constraint equation $Ce_i \geq \sum e_i$, we can determine $\lambda_i$ as

$$\lambda_i = \frac{\sum_{i=1}^{n} Pe_i}{Ce_i}.$$

We substitute $\lambda$ into $e_i'$ to obtain

$$e_i'' = \frac{Pe_i Ce_i}{\sum_{d=1}^{n} Pe_i}.$$

$e_i''$ is the unique energy allocation for maximizing the revenue of energy provider $i$.

Using a similar method, we can solve the computing power allocation optimization problem.

$$U_{resource}(x_k') = Pe_i / x_k' \quad U_{resource}''(x_k') = -Pe_i / x_k'^2.$$

$U_{resource}(x_k') < 0$ is negative due to $0 < x_k'$. The extreme point is the unique value maximizing the revenue of the computing power provider.

Let $\partial L_2(e_i', x_i', y_i') / \partial x_i = 0$ to obtain $x_i' = Pe_i / \beta_i$.

Using this result in the constraint equation $Cn_i \geq \sum y_i'$, we can determine $\beta_i$ as

$$\beta_i = \frac{\sum_{i=1}^{n} Pe_i}{Cn_i}.$$

We substitute $\beta$ into $x_i'$ to obtain

$$x_i'' = \frac{Pe_i Cn_i}{\sum_{d=1}^{n} Pe_i}.$$

$x_i''$ is the unique optimal computing power allocation for maximizing the revenue of computing power provider $i$.

Using a similar method, we can solve network resource allocation optimization problem.

Let $\partial L_2(e_i', x_i', y_i') / \partial y_i = 0$ to obtain $y_i' = Pe_i / \phi_i$.

Using this result in the constraint equation $Cn_i \geq \sum y_i', we can determine $\phi_i$ as

$$\phi_i = \frac{\sum_{i=1}^{n} Pe_i}{Cn_i}.$$

We substitute $\phi$ into $y_i'$ to obtain

$$y_i'' = \frac{Pe_i Cn_i}{\sum_{d=1}^{n} Pe_i}.$$

$y_i''$ is the unique optimal network resource allocation for maximizing the revenue of network resource provider $k$. □

**References**


