

# Structural properties of the optimal resource allocation policy for single-queue systems

Ran Yang · Sandjai Bhulai · Rob van der Mei

Published online: 6 August 2011

© The Author(s) 2011. This article is published with open access at Springerlink.com

**Abstract** This paper studies structural properties of the optimal resource allocation policy for single-queue systems. Jobs arrive at a service facility and are sent one by one to a pool of computing resources for parallel processing. The facility poses a constraint on the maximum expected sojourn time of a job. A central decision maker allocates the servers dynamically to the facility. We consider two models: a limited resource allocation model, where the allocation of resources can only be changed at the start of a new service, and a fully flexible allocation model, where the allocation of resources can also change during a service period. In these two models, the objective is to minimize the average utilization costs whilst satisfying the time constraint. To this end, we cast these optimization problems as Markov decision problems and derive structural properties of the relative value function. We show via dynamic programming that (1) the optimal allocation policy has a work-conservation property, and (2) the optimal number of servers follows a step function with as extreme policy the bang-bang control policy. Moreover, (3) we provide conditions under which the bang-bang control policy takes place. These properties give a full characterization of the optimal policy, which are illustrated by numerical experiments.

**Keywords** Constrained Markov decision problems · Monotonicity · Optimal resource allocation · Shared server pool · Queueing theory

## 1 Introduction

Today, multimedia data is rapidly gaining importance along with recent deployment of publicly accessible digital television archives, surveillance cameras in public locations, and automatic comparison of forensic video evidence (Snoek et al. 2006). Typical services include iris-scan and fingerprint systems that make high-resolution scans and require processing of the data to identify a person; these services operate in a real-time environment and run under very strict time constraints. To meet such constraints, these large-scale applications

---

R. Yang · S. Bhulai (✉) · R. van der Mei  
VU University Amsterdam, Amsterdam, The Netherlands  
e-mail: [sbhulai@few.vu.nl](mailto:sbhulai@few.vu.nl)

typically are being executed on centralized clusters consisting of large collections of computing resources. In large-scale systems, the applications can reserve a number of processing resources to process the data. This gives rise to a new class of models in which the application has to decide how many resources should be reserved. In this decision making, one has to deal with the following trade-off: if the number of computing resources allocated to a facility is too low, then the processing power is insufficient to meet strict processing time requirements; if the number of computing nodes allocated is too high, then it brings forth high resource allocation costs. In this context, the challenge is to identify policies that meet the service requirements at minimal allocation costs.

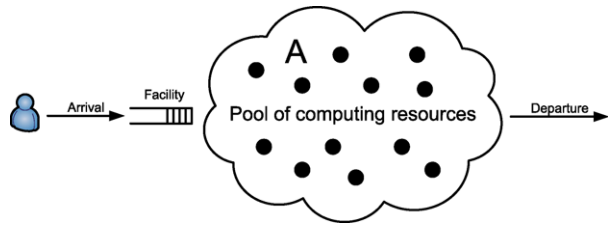
In the literature, a lot of research has been devoted to resource allocation problems. In Daniel and Chronopoulos (2003), Park (2003), Rana et al. (2002), the authors investigate resource allocation problems in the context of protocol design. The problem from an architectural point of view is studied in Czajkowski et al. (1999), Foster et al. (2000), Wang and Luo (2004). Security assurance in Grid/Cluster job scheduling is studied in Song et al. (2006), Xie and Qin (2008). Other research is focused on economic models in a Grid computing environment, e.g., (Sandholm et al. 2006; Buyya et al. 2001; Guo and Conitzer 2010). Several papers are focused on optimization problems in the context of resource allocation. In Aziz and El-Rewini (2006), a framework for resource allocation and task scheduling is presented, where the objective function is to minimize the job completion time. Nurmi and co-authors (Nurmi et al. 2007, 2008) propose a statistical method determining when a job should be submitted to a batch queue to ensure that it will be running at a particular time in the future. Wakamiya et al. (2002) propose a resource-allocation scheme to share resources fairly among users by solving the utilization-maximization problem where the utilization ratio is a function of the video quality and the resource allocation costs.

In this paper, we study a model with a single FCFS queue with a common pool of computing resources in which for each job a number of computing resources has to be allocated dynamically such that a mean sojourn time requirement is met against minimal allocation costs. In this context, we study the following two models: (1) a system in which a chosen allocation cannot be changed during a service of a job, and (2) a system in which it is allowed to change the allocation during the service of a job. For both models, we show via dynamic programming that (i) the optimal allocation policy has a work-conservation property that implies when the system is not empty, the optimal policy is not allowed to keep all computing resources idle, (ii) the optimal number of servers follows a step function with as extreme policy the bang-bang control policy, which means a facility receives all computing resources or none at all, and moreover (iii) we also provide the conditions under which the bang-bang control policy is optimal. The techniques to prove such results are based on monotonicity properties of the dynamic programming relative value function (see, e.g., Koole 1998, 2006; Rykov 2001).

The contribution of the paper is two-fold. First, on the methodological side we provide a full characterization of the optimal policy for a single-queue system. Interestingly, the derivation of these results is not obtained via standard induction-based arguments, but is based on a combination of direct arguments and induction. The results provide new and valuable insight into optimal resource allocation problems with time constraints. Second, on the application side we have structured policies that are easy to be implemented in systems that are highly relevant in practice.

In recent work, the authors have studied similar resource allocation problems in different settings. In Yang et al. (2011), we study a resource allocation problem in which resources need to be reserved in advance before they are available. In Yang et al. (2011), we study

**Fig. 1** Arrival and departure of the jobs



a resource allocation problem in a multi-queue setting in which multiple queues compete for the same pool of shared resources. The distinguishing feature in this model is that the optimal resource allocation policy partitions the resource pool in subsets each of which is dynamically allocated to the different queues. The model under consideration is a special case in which there is only a single queue, and therefore allows for a full characterization of the optimal policy which is not directly applicable in the multi-queue setting. This is due to the techniques used to prove the results which are not generalizable to multiple queues. These observations make the added value of the present paper compared to the two papers mentioned above evident.

This paper is organized as follows. In Sect. 2 we formulate the models for the different cases. Next, we derive the special properties of the optimal policy in Sect. 3. In Sect. 4 we illustrate these results by numerical experiments. Finally, in Sect. 5 we conclude the paper.

**2 Model formulation**

Consider a service facility at which jobs arrive according to a Poisson process with arrival rate  $\lambda$  and have exponentially distributed service requirements. The facility has a single infinite-sized FCFS queue and a pool of  $A \geq 1$  parallel and identical servers that can work together to process a single job. There is a central decision maker that can assign a number of servers to a job. The service rate of a job depends on the number of servers assigned to that job. More precisely, when a job has been allocated  $a$  servers, the service duration of that job is exponentially distributed with parameter  $\mu(a)$ , which is strictly increasing in  $a$ . Without loss of generality, we assume that  $\mu(0) = 0$ . After a job has completed its service, it leaves the system. Throughout the paper, it is assumed that the stability condition  $\lambda < \mu(A)$  is met, and that the system is in steady state. Figure 1 gives an illustration of the system.

We consider the following cost structure in the facility. When the facility uses  $a$  resources a cost of  $c(a)$  is incurred by the system per unit time. Here  $c(a)$  is a strictly increasing function of  $a$ . Without loss of generality, we assume that  $c(0) = 0$ . Let  $S$  denote the sojourn time of an arbitrary job at the facility. The problem in the system is to find a server assignment policy that minimizes that long-run average costs subject to  $\mathbb{E}S \leq \alpha$  for some  $\alpha > 0$ .

When the system works at full speed, all  $A$  computing resources are allocated at each moment in time. In this case, the average sojourn time is given by the mean sojourn in an  $M|M|1$  queue with arrival rate  $\lambda$  and service rate  $\mu(A)$ . Therefore, by using all  $A$  computing resources the average sojourn time is equal to  $\frac{1}{\mu(A)-\lambda}$ . Hence, to ensure that there exists at least one allocation strategy that meets the time constraint  $\alpha$ , the total number of compute resources  $A$  should satisfy the following necessary and sufficient stability condition:

$$A > \lceil \mu^{-1}(\lambda + 1/\alpha) \rceil,$$

where  $\mu^{-1}(\cdot)$  is the inverse function of  $\mu(\cdot)$ . Throughout the paper, it is assumed that this condition is met.

The objective of the decision maker is to derive an optimal policy based on the number of jobs in the system. More precisely, when the decision maker observes  $x$  jobs in the system, he can decide to allocate  $0 \leq i(x) \leq A$  servers to the first job in the queue. We emphasize that at any moment in time, there is at most one job in service.

We study two models. First, we study the case in which the resource allocation policy cannot be changed when a job is being served; only *upon the start of the service* of the next job the resource allocation policy can be modified. The second case deals with the fully flexible case in which the system can change the resource allocation policy *at any moment in time*, thus also during the service of a job.

### 2.1 Limited resource allocation policy

In this subsection we focus on the case in which the resource allocation for a service facility can only be changed upon the start of the service of a new job. Hence, adding or removing resources during a service is not allowed. To study this case, we cast the resource allocation problem as a Markov decision problem.

Define the state space  $\mathcal{X} = \mathbb{N}_0 \times \{0, \dots, A\}$ , where  $(x, a) \in \mathcal{X}$  denotes the state in which there are  $x$  jobs at the facility with  $a$  resources allocated to the first job. When the system is in state  $(x, a) \in \mathcal{X}$  the decision maker can choose actions from the action space  $\mathcal{A}_{(x,a)} = \{b \in \mathbb{N}_0 \mid a + b \leq A \text{ and } ab = 0\}$ , where action  $b \in \mathcal{A}_{(x,a)}$  denotes the available number of resources for allocation. Here, the restriction  $ab = 0$  models the fact that when a service is ongoing (i.e.,  $a > 0$ ), the service allocation cannot be changed (i.e.,  $b = 0$ ). The transition rates when the system is in state  $(x, a) \in \mathcal{X}$  and action  $b \in \mathcal{A}_{(x,a)}$  is chosen are given by

$$p((x, a), b, (x', b')) = \begin{cases} \lambda, & x' = x + 1 \text{ and } b' = a + b, \\ \mu(a + b), & x' = [x - 1]^+ \text{ and } b' = 0, \\ 0, & \text{otherwise,} \end{cases}$$

with  $[x]^+ = \max\{x, 0\}$ . The first line in the expression above models arrivals, the second line models service completions, and the third line prohibits any other state transitions. Note that when a service completes, the resource allocated for that facility is released completely. Finally, when the system is in state  $(x, a) \in \mathcal{X}$  and action  $b \in \mathcal{A}_{(x,a)}$  has been chosen, the direct cost is  $r((x, a), b) = c(a + b)$  per time unit. The quadruple  $(\mathcal{X}, \mathcal{A}, p, r)$  completely describes the Markov decision process.

Define a decision rule  $\pi_{(x,a)}$  as a probability distribution on  $\mathcal{A}_{(x,a)}$ , i.e., when the system is in state  $(x, a) \in \mathcal{X}$ , the decision maker chooses action  $b \in \mathcal{A}_{(x,a)}$  with probability  $\pi_{(x,a)}(b)$ . Let the policy  $\pi$  denote the collection of decision rules for all states. Let  $u_t^\pi(x, a)$  denote the total expected costs up to time  $t$  when the system starts in state  $(x, a)$  under policy  $\pi$ . Note that for any stable and work-conserving policy, the Markov chain satisfies the unchain condition, so that the average expected costs  $g(\pi) = \lim_{t \rightarrow \infty} u_t^\pi(x, a)/t$  is independent of the initial state  $(x, a)$  (see Proposition 8.2.1 of Puterman (1994)). The goal is to find a policy  $\pi^*$  that minimizes the long-term average costs under the time constraint, thus

$$\min_{\pi} g(\pi) \quad \text{subject to} \quad \mathbb{E}S \leq \alpha.$$

Note that due to Little’s Law the number of jobs  $L$  in the facility can be related to the sojourn time  $S$  by  $\mathbb{E}L = \lambda \mathbb{E}S$ . Using this knowledge, the constrained Markov decision problem can be rewritten as an unconstrained Markov decision problem using Lagrange multipliers (see Sect. 12.6 of Altman (1999)). To this end, we uniformize the system (see Sect. 11.5 of

Puterman (1994)). Therefore, assume that the uniformization constant  $\lambda + \mu(A) = 1$ ; This can always be obtained by scaling. Uniformizing is equivalent to adding dummy transitions (from a state to itself) such that the rate out of each state is equal to 1; then we can consider the rates to be transition probabilities. Now, let  $V(x, a)$  be a real-valued function defined on the state space. This function will play the role of the relative value function, i.e., the asymptotic difference in total costs that results from starting the process in state  $(x, a)$  instead of some reference state. The long-term average optimal actions are a solution of the optimality equation (in vector notation)  $g \cdot e + V = TV$ , where  $T$  is the dynamic programming operator acting on  $V$ , defined as follows

$$\begin{aligned} TV(x, a) &= \tau \frac{x}{\lambda} + c(a) + \lambda H(x + 1, a) + \mu(a)H([x - 1]^+, 0) \\ &\quad + (1 - \lambda - \mu(a))V(x, a) \\ &= \tau \frac{x}{\lambda} + c(a) + \lambda H(x + 1, a) + \mu(a)H([x - 1]^+, 0) \\ &\quad + (\mu(A) - \mu(a))V(x, a), \end{aligned} \tag{1}$$

where  $\tau$  is the Lagrange multiplier, and where the function  $H(\cdot, \cdot)$  is given by

$$H(x, a) = \min_{b \in \mathcal{A}(x,a)} \{V(x, a + b)\}.$$

The first term in the dynamic programming operator (1) corresponds to the service requirement  $\mathbb{E}S \leq \alpha$ . When  $\tau$  increases, the value of  $\mathbb{E}S$  decreases. Therefore, there exists a value of  $\tau^*$  such that  $\mathbb{E}S^{\tau^*} > \alpha$  for all facilities and there exists another  $\tau'$  such that  $\mathbb{E}S^{\tau'} < \alpha$ , where  $\tau' = \tau^* + \epsilon$  for a small  $\epsilon \geq 0$ . Note that we can obtain the expected sojourn time at the facility for a given policy by setting  $r((x, a), b) = 0$  for all states  $(x, a)$  and actions  $b$  in the optimality equation. The optimal policy is to randomize between the associated policies  $\pi^{\tau^*}$  and  $\pi^{\tau'}$  so that the equality  $\mathbb{E}S = \alpha$  is achieved for the facility. This is consistent with the fact that the optimal policy will be randomized in exactly one state (see Sect. 12.6 of Altman (1999)). The second term represents the cost of using  $a$  resources. The third term is involved with the decision making upon arrival of a job. The fourth term deals with the decision making when a job has completed its service. The final term is the dummy term due to uniformization. Note that the decision making is modeled uniformly through the function  $H(\cdot, \cdot)$ .

### 2.2 Fully flexible resource allocation policy

In this section we study the case in which the service facility has full flexibility in the resource allocation policies. The resource facility can change the resource allocation during a service of a job, and it does not have to wait for the job to be finished. Since our system has Poisson arrivals and exponential service times, it suffices to consider only the moments that an event occurs. Therefore, the only difference with the previous case, discussed in Sect. 2.1, is that this system allows to change the allocation at arrival instants.

In the fully flexible case, the state space is given by  $\mathcal{X} = \mathbb{N}_0$ , where  $x \in \mathcal{X}$  denotes that there are  $x$  jobs at the facility. The action space is given by  $\mathcal{A}_x = \{0, \dots, A\}$ , where action  $a \in \mathcal{A}_x$  denotes the number of resources that one has allocated in state  $x \in \mathcal{X}$ . The transition

rates when the system is in state  $x \in \mathcal{X}$  and action  $a \in \mathcal{A}_x$  is chosen are given by

$$p(x, a, x') = \begin{cases} \lambda, & x' = x + 1, \\ \mu(a), & x' = [x - 1]^+, \\ 0, & \text{otherwise.} \end{cases}$$

Finally, when the system is in state  $x \in \mathcal{X}$  and action  $a \in \mathcal{A}_x$  has been chosen, the direct cost is  $r(x, a) = c(a)$  per time unit. The tuple  $(\mathcal{X}, \mathcal{A}, p, c)$  completely describes the Markov decision process for this problem.

Let  $V(x)$  denote the relative value function in this case. Then, the dynamic programming operator acting on  $V$ , defined as follows:

$$TV(x) = \tau \frac{x}{\lambda} + \lambda V(x + 1) + \min_{a \in \mathcal{A}_x} T_a(x), \tag{2}$$

where

$$\begin{aligned} T_a(x) &= \mu(a)V([x - 1]^+) + (1 - \lambda - \mu(a))V(x) + c(a) \\ &= \mu(a)V([x - 1]^+) + (\mu(A) - \mu(a))V(x) + c(a). \end{aligned}$$

The first term in the expression  $TV(x)$  corresponds to the constraint  $\mathbb{E}S \leq \alpha$  of the facility, similar to the limited resource allocation model, see (1). The second term models the arrivals of jobs to the facility. The first term in the expression  $T_a(x)$  denotes the departure of a job in case action  $a$  has been chosen. The second term in  $T_a(x)$  is the uniformization constant. The last term in  $T_a(x)$  models the direct cost.

### 3 Structural properties of the optimal policy

In the previous section, we described two models and a solution technique to obtain the optimal policy. However, the optimal policy also possesses structural properties that provide fundamental insight. Moreover, this also enables one to determine the optimal policy with less computational effort due to a reduction of the solution search space. Therefore, in this section, we derive a full characterization of the optimal policy for both models.

#### 3.1 Limited resource allocation policy

The structure of the optimal policy for a service facility with the limited resource allocation policy is more intricate than the case with dedicated resources. In order to study the structure, in principle, one needs to solve the optimality equation  $g + V = TV$  with  $TV$  given by (1). However, the optimality equation is hard to solve analytically in practice. Alternatively, the optimal actions can also be obtained by recursively defining  $V_{l+1} = TV_l$  for arbitrary  $V_0$ . For  $l \rightarrow \infty$ , the maximizing actions converge to the optimal ones (for existence and convergence of solutions and optimal policies we refer to Puterman (1994) and Aviv and Federgruen (1999)). The backward recursion equation is given by

$$\begin{aligned} V_{n+1}(x, a) &= \tau \frac{x}{\lambda} + c(a) + H_n(x + 1, a) + \mu(a)H_n([x - 1]^+, 0) \\ &\quad + (\mu(A) - \mu(a))V_n(x, a), \end{aligned} \tag{3}$$

where the function  $H_n(\cdot, \cdot)$  is given by

$$H_n(x, a) = \min_{b \in \mathcal{A}(x, a)} \{V_n(x, a + b)\}.$$

For ease of notation, we also define the set of optimal actions in state  $(x, a)$  by  $\arg H_n(x, a)$  defined as

$$\arg H_n(x, a) = \arg \min_{b \in \mathcal{A}(x, a)} \{V_n(x, a + b)\}.$$

The backward recursion equation (3) allows us to prove structural properties of the relative value function  $V$  through induction on  $n$  in  $V_n$ . By using this, we can show that the optimal allocation policy has the following properties:

**Property (i) monotonicity:** *this implies that the optimal policy is a step function. Thus for all  $a \in \arg H(x + k, 0)$  and  $b \in \arg H(x, 0)$ , we have  $a \geq b$  for all  $k \geq 0$ .*

**Property (ii) work-conservation:** *this implies that if the system is not empty, then the optimal policy is not keeping all computing resources idle. Thus the minimal element of  $\arg H(x, 0)$  is strictly positive for all  $x > 0$ .*

**Property (iii) bang-bang control:** *if the condition  $c(a)\mu(a + 1) - \mu(a)c(a + 1) \geq 0$  holds for all  $a \in \mathcal{A}(x, b)$ , then the optimal policy is a bang-bang control policy, which means using all servers or no server at all.*

Monotonicity Property (i) has already been studied in Yang et al. (2011) in a multi-queue setting. In a single-queue setting, the proofs can be slightly generalized by not relying on the fact that  $\arg H$  is a singleton. Therefore, for completeness, we have added the proof of Property (i) in Appendix A. In order to prove Properties (ii) and (iii), we need the following additional properties.

**Property 3.1** (non-decreasingness) *The relative value function  $V$  is increasing in the number of jobs, i.e.,*

$$V(x + 1, a) - V(x, a) \geq 0,$$

for all  $x \in \mathcal{X}$  and  $0 \leq a \leq A$ .

**Property 3.2** (convexity-related properties) *For a single-queue system, the following properties hold*

- (i)  $V(x + 1, a) - 2V(x, a) + V(x - 1, a) \geq 0$  for all  $x \geq 1$  and  $a \geq 0$ ,
- (ii)  $V(x, a) - V(\lceil x - 1 \rceil^+, a) - H(\lceil x - 1 \rceil^+, 0) + H(\lceil x - 2 \rceil^+, 0) > 0$  for all  $x \geq 0$  and  $a > 0$ ,
- (iii) *The minimal element of  $\arg H(x, 0)$  is strictly positive for all  $x \geq 2$ , and*
- (iv)  $H(x + 1, 0) - 2H(x, 0) + H(x - 1, 0) \geq 0$  for all  $x \geq 1$ .

**Property 3.3** (submodularity) *The relative value function satisfies a version of submodularity, namely,*

$$V(x, a + k) - V(x, a) - V(x + 1, a + k) + V(x + 1, a) > 0,$$

for all  $x \geq 1$ ,  $a > 0$ , and  $k > 0$ .

**Property 3.4** (monotonicity) *If the service rate  $\mu(a)$  and cost function  $c(a)$  are strictly increasing functions in  $a$ , then for all  $a \in \arg H(x + k, 0)$  and  $b \in \arg H(x, 0)$ , we have  $a \geq b$  for all  $k \geq 0$ .*

Now, we proceed to prove the work-conservation Property (ii) and the bang-bang control Property (iii) of the optimal policy. Recall that we assumed  $c(0) = \mu(0) = 0$ . First, we start with the proof of the work-conservation property.

**Theorem 3.5** (work-conservation) *If the service rate function  $\mu(\cdot)$  and cost function  $c(\cdot)$  are strictly increasing functions, then the minimal element of  $\arg H(x, 0)$  is strictly positive for all  $x > 0$ .*

*Proof* Based on part (iii) of Property 3.2, we have that the minimal element of  $\arg H(x, 0)$  is strictly positive for all  $x \geq 2$ . Therefore, we only need to prove that each element of  $\arg H(1, 0)$  is strictly positive. This can be proven by the contradiction method. Assume that  $0 \in \arg H(1, 0)$ . Then it follows that  $H(1, 0) = V(1, 0)$ . Based on the optimality equation and the assumption that  $c(0) = \mu(0) = 0$ , we have

$$V(1, 0) - V(0, 0) = \frac{\tau}{\lambda} + \lambda[H(2, 0) - H(1, 0)] + \mu(A)[V(1, 0) - V(0, 0)].$$

Since  $\mu(A) + \lambda = 1$ , it follows

$$\lambda[V(1, 0) - V(0, 0) - H(2, 0) + H(1, 0)] = \frac{\tau}{\lambda}.$$

Because we have  $\tau > 0$ . Then, it holds that

$$V(1, 0) - V(0, 0) - H(2, 0) + H(1, 0) > 0. \quad (4)$$

However, based on our assumption  $H(1, 0) = V(1, 0)$  and the fact that  $V(0, 0) \geq H(0, 0)$ , it holds that

$$V(1, 0) - V(0, 0) - H(2, 0) + H(1, 0) \leq 2H(1, 0) - H(0, 0) - H(2, 0).$$

Based on part (iv) of Property 3.2, it follows that  $V(1, 0) - V(0, 0) - H(2, 0) + H(1, 0)$  is non-positive. This is in contradiction with (4). Therefore, we conclude that each element of  $\arg H(1, 0)$  is strictly positive, which completes the proof.  $\square$

Next, we prove that the optimal allocation policy has the bang-bang control property. To this end, the following result is needed.

**Lemma 3.6** *The following inequality holds*

$$V(x, a) - H(x - 1, 0) > \frac{c(a)}{\mu(a)},$$

for all  $x \geq 1$  and  $a > 0$ .

*Proof* Since  $a > 0$ , it holds that  $V(x, a) = H(x, a)$  for all  $x$ . Therefore, based on the optimality equation, we have



$$V(x, a) - V(0, 0) = \tau \frac{x}{\lambda} + c(a) + \lambda[V(x + 1, a) - H(1, 0)] + \mu(a)H([x - 1]^+, 0) + [\mu(A) - \mu(a)]V(x, a) - \mu(A)V(0, 0).$$

Since  $\lambda + \mu(A) = 1$ , the equation above implies that

$$\lambda[H(1, 0) - V(0, 0) + V(x, a) - V(x + 1, a)] = \tau \frac{x}{\lambda} + c(a) + \mu(a)[H([x - 1]^+, 0) - V(x, a)]. \tag{5}$$

Because of the definition of the value function, it holds that  $V(0, 0) \geq H(0, 0)$ . This implies that

$$H(1, 0) - V(0, 0) + V(x, a) - V(x + 1, a) \leq H(1, 0) - H(0, 0) + V(x, a) - V(x + 1, a). \tag{6}$$

Based on parts (i) and (ii) of Property 3.2, we obtain that

$$V(x + 1, a) - V(x, a) \geq V(2, a) - V(1, a) \geq H(1, 0) - H(0, 0). \tag{7}$$

Combining (6) and (7), we conclude that

$$H(1, 0) - V(0, 0) + V(x, a) - V(x + 1, a) \leq 0.$$

Because  $\tau > 0$ , (5) implies that

$$c(a) + \mu(a)[H(x - 1, 0) - V(x, a)] < 0,$$

and hence  $V(x, a) - H(x - 1, 0) > \frac{c(a)}{\mu(a)}$ . □

We are now ready to prove the optimality of the bang-bang control policy.

**Theorem 3.7** (bang-bang control) *For a single-queue system, the following properties hold*

- (i)  $\arg H(0, 0) = \{0\}$ .
- (ii) if  $c(a)\mu(a + 1) - \mu(a)c(a + 1) \geq 0$  for all  $a \in \mathcal{A}_{(x,b)}$ , then  $\arg H(x, 0) = \{0, A\}$  for all  $x > 0$ .

*Proof* We use the contradiction method to prove this theorem. To start, let  $a \in \arg H(0, 0)$ . Then  $H(0, 0) = V(0, a)$ . Now assume that  $a > 0$ , then based on the definition of the value function, we have

$$\begin{aligned} V(0, 0) - V(0, a) &= -c(a) + \lambda[H(1, 0) - H(1, a)] - \mu(a)H(0, 0) \\ &\quad + \mu(A)[V(0, 0) - V(0, a)] + \mu(a)V(0, a) \\ &= -c(a) + \lambda[H(1, 0) - H(1, a)] + \mu(A)[V(0, 0) - V(0, a)]. \end{aligned}$$

Since  $\lambda + \mu(A) = 1$ , we have that

$$\lambda[V(0, 0) - V(0, a)] = -c(a) + \lambda[H(1, 0) - H(1, a)].$$

Because  $H(1, 0) - H(1, a) \leq 0$  and  $c(a) > 0$  for  $a > 0$ , we have  $V(0, 0) - V(0, a) < 0$ , which is in contradiction with the assumption that  $0 < a \in \arg H(0, 0)$ . Therefore, we conclude that  $\arg H(0, 0) = \{0\}$ , which proves part (i).

To prove part (ii), we assume that there exists  $x \geq 1$  and  $a \in \arg H(x, 0)$  such that  $0 < a < A$ . Then we have  $V(x, a) = H(x, 0)$  for all  $0 < a < A$ . Because of the definition of the value function,  $V(x, a) = H(x, a)$  for all  $x$ . Based on the optimality equation, we have

$$\begin{aligned} V(x, a + 1) - V(x, a) &= c(a + 1) - c(a) + \lambda[V(x + 1, a + 1) - V(x + 1, a)] \\ &\quad + [\mu(a + 1) - \mu(a)]H(x - 1, 0) \\ &\quad + \mu(A)[V(x, a + 1) - V(x, a)] \\ &\quad - \mu(a + 1)V(x, a + 1) + \mu(a)V(x, a). \end{aligned}$$

This implies that

$$\begin{aligned} &\lambda[V(x, a + 1) - V(x, a) - V(x + 1, a + 1) + V(x + 1, a)] \\ &= c(a + 1) - c(a) + [\mu(a + 1) - \mu(a)]H(x - 1, 0) - \mu(a + 1)V(x, a + 1) \\ &\quad + \mu(a)V(x, a). \end{aligned}$$

Since  $V(x, a) = H(x, 0)$ , we have  $V(x, a + 1) \geq V(x, a)$ , and hence

$$\begin{aligned} &\lambda[V(x, a + 1) - V(x, a) - V(x + 1, a + 1) + V(x + 1, a)] \\ &\leq c(a + 1) - c(a) + [\mu(a + 1) - \mu(a)]H(x - 1, 0) - \mu(a + 1)V(x, a) + \mu(a)V(x, a) \\ &= c(a + 1) - c(a) + [\mu(a + 1) - \mu(a)][H(x - 1, 0) - V(x, a)]. \end{aligned} \tag{8}$$

Based on Property 3.3, it holds that  $V(x, a + 1) - V(x, a) - V(x + 1, a + 1) + V(x + 1, a) \geq 0$ . Therefore, inequality (8) implies that

$$c(a + 1) - c(a) + [\mu(a + 1) - \mu(a)][H(x - 1, 0) - V(x, a)] \geq 0,$$

which is equivalent to

$$V(x, a) - H(x - 1, 0) \leq \frac{c(a + 1) - c(a)}{\mu(a + 1) - \mu(a)}. \tag{9}$$

Based on the condition  $c(a)\mu(a + 1) - \mu(a)c(a + 1) \geq 0$  of part (ii), we have

$$\begin{aligned} &c(a)\mu(a + 1) - c(a)\mu(a) + c(a)\mu(a) - \mu(a)c(a + 1) \\ &= c(a)[\mu(a + 1) - \mu(a)] - \mu(a)[c(a + 1) - c(a)] \geq 0, \end{aligned}$$

which implies that

$$\frac{c(a + 1) - c(a)}{\mu(a + 1) - \mu(a)} \leq \frac{c(a)}{\mu(a)}, \quad \text{and hence } V(x, a) - H(x - 1, 0) \leq \frac{c(a)}{\mu(a)},$$

which is in contradiction with Lemma 3.6. This proves part (ii). □

The following corollary follows directly from combining Theorems 3.5 and 3.7.

**Corollary 3.8** *Under the assumption  $c(a)\mu(a + 1) - \mu(a)c(a + 1) \geq 0$  it holds that  $\arg H(x, 0) = \{A\}$  for all  $x > 0$  and  $a \in A_{(x,b)}$ .*

### 3.2 Fully flexible resource allocation policy

In this section we focus our attention to optimal allocation strategies for the system in which the allocation of the number of servers is allowed to change when a job is already in service. We shall adopt the same techniques in deriving the structure of the optimal policy as in the previous section.

We adopt the same techniques of proof as described in Sect. 3.1. We start by rewriting (2) for the fully flexible system as a set of backward recursion equations. This set of equations is given by

$$V_{n+1}(x) = \tau \frac{x}{\lambda} + \lambda V_n(x+1) + \min_{a \in \mathcal{A}_x} T_a^n(x), \quad (10)$$

where  $T_a^n(x)$  is given by

$$T_a^n(x) = \mu(a)V_n([x-1]^+) + [\mu(A) - \mu(a)]V_n(x) + c(a).$$

By performing backward recursion, the optimal allocation policy can be obtained, which has the following properties.

**Property (i) monotonicity:** *this implies that the optimal policy is a step function. Thus for all  $a \in \arg \min_{a \in \mathcal{A}_{x+1}} \{T_a(x+1)\}$  and  $b \in \arg \min_{a \in \mathcal{A}_x} \{T_a(x)\}$  we have  $a \geq b$  for all  $x \geq 0$ .*

**Property (ii) work-conservation:** *this implies that if the system is not empty, then the optimal policy is not keeping all compute resources idle. Thus, the minimal element of  $\arg \min_{a \in \mathcal{A}_x} \{T_a(x)\}$  is strictly positive for all  $x > 0$ .*

**Property (iii) bang-bang control:** *if  $c(a)\mu(a+1) - \mu(a)c(a+1) \geq 0$  holds for all  $a \in \mathcal{A}_x$ , then the optimal policy is a bang-bang control policy, which means using all servers or no server at all.*

The proof of monotonicity Property (i) is given in Appendix B (the ideas are similar to the multi-queue setting studied in Yang et al. (2011)). Additionally, we need the following properties of the relative value function from the same paper to prove Properties (i) and (ii).

**Property 3.9** (convexity) *Assume that the functions  $\mu(\cdot)$  and  $c(\cdot)$  are strictly increasing functions, then  $V(x)$  is a convex increasing function in  $x$ , i.e.,  $V(x) > V(x-1)$  and  $V(x+1) - 2V(x) + V(x-1) > 0$  for all  $x \geq 1$ .*

**Property 3.10** (monotonicity) *Assume that the functions  $\mu(\cdot)$  and  $c(\cdot)$  are strictly increasing functions in  $a$ . Then the optimal resource allocation strategy is given by a non-decreasing curve, i.e., for each  $a \in \arg \min_{a \in \mathcal{A}_{x+1}} \{T_a(x+1)\}$  and  $b \in \arg \min_{a \in \mathcal{A}_x} \{T_a(x)\}$  we have  $a \geq b$  for all  $x \geq 0$ .*

We proceed to derive parts (ii) and (iii). We start with the work-conservation property. Recall that without loss of generality, it is assumed that  $c(0) = \mu(0) = 0$ . The following result shows that if the system is not empty, then the optimal policy will not keep all computing resources idle.

**Theorem 3.11** (work-conservation) *The minimal element of  $\arg \min_{a \in \mathcal{A}_x} \{T_a(x)\}$  is strictly positive for all  $x > 0$ .*

*Proof* We use the contradiction method to prove this theorem. Assume there exists  $x > 0$  such that the minimal element of  $\arg \min_{a \in \mathcal{A}_x} \{T_a(x)\}$  is 0. Then, Property 3.10 implies that  $\arg \min_{a \in \mathcal{A}_x} \{T_a(0)\} = \{0\}$ . Therefore, by applying the optimality equation, we obtain

$$V(x) - V(0) = \tau \frac{x}{\lambda} + \lambda[V(x+1) - V(1)] + \mu(A)[V(x) - V(0)].$$

Since  $\lambda + \mu(A) = 1$ , this is equivalent to

$$\lambda[V(1) - V(0) + V(x) - V(x+1)] = \tau \frac{x}{\lambda}.$$

Because the time constraint of the facility is finite, we have  $\tau > 0$ . Hence, it holds that  $V(1) - V(0) + V(x) - V(x+1) > 0$ . However, Property 3.9 directly implies that  $V(1) - V(0) + V(x) - V(x+1) \leq 0$ , which contradicts the convexity of Property 3.9. This completes the proof.  $\square$

In order to prove that the optimal allocation policy has the bang-bang control property, we need the following lemma.

**Lemma 3.12** *Let  $b \in \arg \min_{a \in \mathcal{A}_x} \{T_a(x)\}$  for arbitrary  $x \geq 1$ , then the following inequality holds*

$$V(x) - V(x-1) \geq \frac{c(b)}{\mu(b)}.$$

*Proof* From the definition of the value function, we have

$$T_a(0) = \mu(a)V(0) + [\mu(A) - \mu(a)]V(0) + c(a) = \mu(A)V(0) + c(a),$$

which implies that  $\arg \min_{a \in \mathcal{A}_x} \{T_a(0)\} = \{0\}$ . For  $x > 0$ , let  $b \in \arg \min_{a \in \mathcal{A}_x} \{T_a(x)\}$ . Then Theorem 3.11 implies that  $b > 0$ . Moreover, based on the optimality equation, it holds that

$$V(x) - V(0) = \tau \frac{x}{\lambda} + \lambda[V(x+1) - V(1)] + \mu(b)V(x-1) + \mu(A)[V(x) - V(0)] - \mu(b)V(x) + c(b).$$

Since  $\lambda + \mu(A) = 1$ , this is equivalent to

$$\lambda[V(1) - V(0) + V(x) - V(x+1)] = \tau \frac{x}{\lambda} + c(b) + \mu(b)[V(x-1) - V(x)].$$

Based on  $\tau > 0$  and Property 3.9, it holds that  $V(1) - V(0) + V(x) - V(x+1) \leq 0$ , and hence

$$c(b) + \mu(b)[V(x-1) - V(x)] < 0,$$

which is equivalent to  $V(x) - V(x-1) > \frac{c(b)}{\mu(b)}$ . This completes the proof.  $\square$

We are now ready to prove the optimality of the bang-bang control policy.

**Theorem 3.13** (bang-bang control) *The following properties hold*

- (i)  $\arg \min_{a \in \mathcal{A}_0} \{T_a(0)\} = \{0\}$ .

(ii) If  $c(a)\mu(a + 1) - \mu(a)c(a + 1) \geq 0$  holds for all  $a \in \mathcal{A}_x$ , then  $\arg \min_{a \in \mathcal{A}_x} \{T_a(x)\} = \{0, A\}$  for all  $x > 0$ .

*Proof* First, we prove that  $\arg \min_{a \in \mathcal{A}_0} \{T_a(0)\} = \{0\}$ . To this end, the definition of the value function implies

$$T_a(0) = \mu(a)V(0) + [\mu(A) - \mu(a)]V(0) + c(a) = \mu(A)V(0) + c(a),$$

which immediately shows that  $\arg \min_{a \in \mathcal{A}_0} \{T_a(0)\} = \{0\}$ . This concludes the proof of part (i).

To prove part (ii), we use the contradiction method. Assume that there exists  $x > 0$  and  $b = \arg \min_{a \in \mathcal{A}_x} \{T_a(x)\}$  such that  $0 < b < A$ . Then  $0 < b < A$ , which implies that  $T_{b+1}(x) - T_b(x) \geq 0$ . Since

$$T_{b+1}(x) - T_b(x) = c(b + 1) - c(b) + [\mu(b + 1) - \mu(b)][V(x - 1) - V(x)],$$

this implies that

$$V(x) - V(x - 1) \leq \frac{c(b + 1) - c(b)}{\mu(b + 1) - \mu(b)}.$$

Based on the condition of the lemma, we have  $c(a)\mu(a + 1) - \mu(a)c(a + 1) \geq 0$  for all  $a \in \mathcal{A}_x$ . Therefore, it holds that  $\frac{c(b+1)-c(b)}{\mu(b+1)-\mu(b)} \leq \frac{c(b)}{\mu(b)}$ . Thus  $V(x) - V(x - 1) \leq \frac{c(b)}{\mu(b)}$ . However, based on Lemma 3.12 it holds that  $V(x) - V(x - 1) > \frac{c(b)}{\mu(b)}$  for all  $x > 0$ , which is in contradiction. This completes part (ii) of the proof.  $\square$

By combining Theorems 3.11 and 3.13 we obtain the following result.

**Corollary 3.14**  $\arg \min_{a \in \mathcal{A}_x} \{T_a(x)\} = \{A\}$  for all  $x > 0$ .

### 4 Numerical results

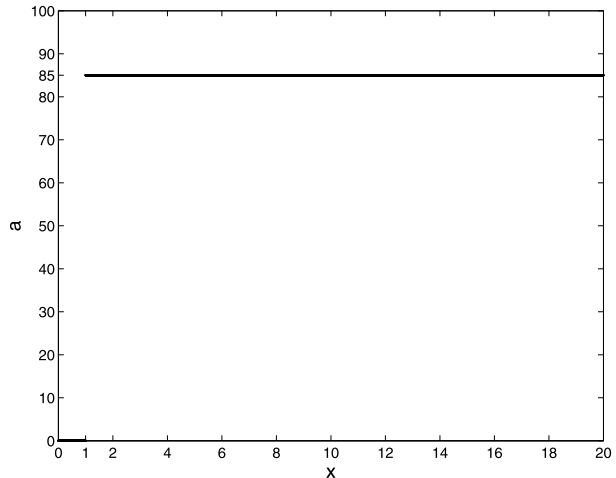
In this section, we illustrate the structural properties of the optimal allocation policies. To this end, we have conducted numerical experiments for a variety of cost functions  $c(\cdot)$  and service rates  $\mu(\cdot)$  having different characteristics. These parameter choices are given in Table 1 for 7 experiments. The other parameters in our experiments are set as follows:  $A = 85$  (this represents the number of servers in the DAS-3 cluster that we used, see The Distributed ASCI Supercomputer 2011),  $\lambda = 2$ ,  $\mu = 0.7$ , and  $\tau = 1$ . Note that  $\mu$  is defined as  $\mu(1)$ , the service rate of using a single server (used for normalization of the server speed). Moreover, we fixed the value of  $\tau$ , since we are only interested in the structural properties of the optimal allocation policy.

In Experiments 1–4, the functions  $c(\cdot)$  and  $\mu(\cdot)$  satisfy the conditions of the bang-bang control property (see Theorems 3.7 and 3.13). For these experiments, the results for both models are shown in Fig. 2. In this figure, we see that the results are in agreement with Corollaries 3.8 and 3.14.

In Experiments 5–7, the conditions of the bang-bang control property are not satisfied. Since  $c(\cdot)$  and  $\mu(\cdot)$  are strictly increasing functions, the optimal allocation policy satisfies the monotonicity property, which implies that the optimal policy follows a step function. The experimental results of Experiments 5–7 for the limited resource allocation model and the fully flexible resource allocation model are shown in Figs. 3 and 4, respectively.

**Table 1** Parameter choices for  $c(\cdot)$  and  $\mu(\cdot)$ 

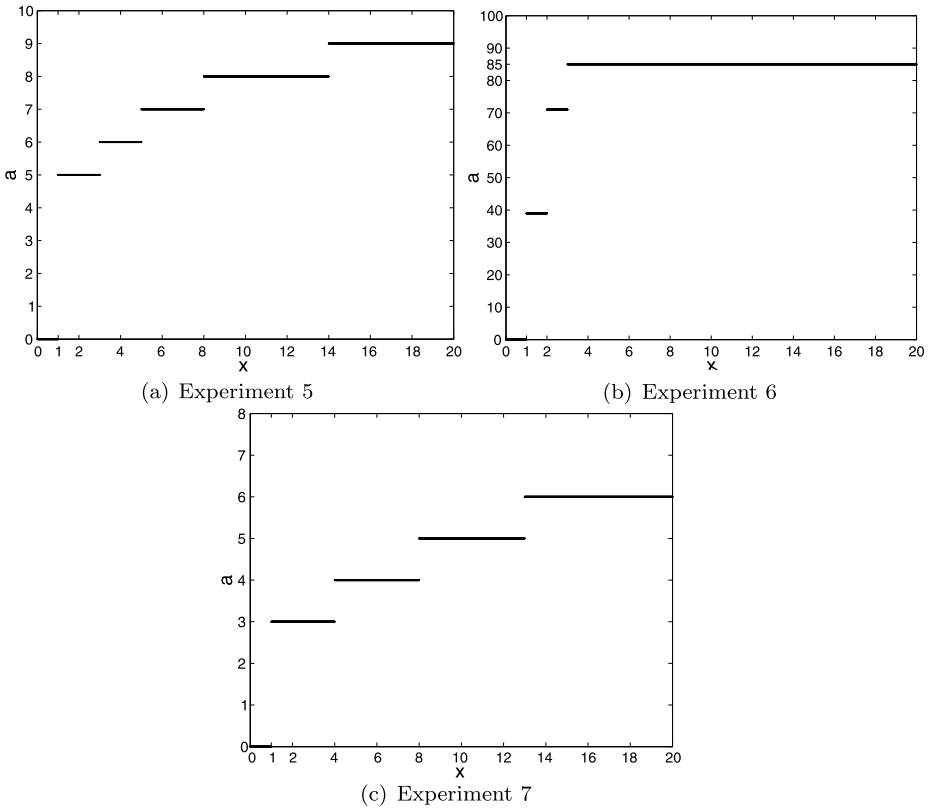
Experiment 1	$c(a) = a$	$\mu(a) = a\mu$
Experiment 2	$c(a) = \sqrt{a}$	$\mu(a) = a^2\mu$
Experiment 3	$c(a) = \sqrt[3]{a}$	$\mu(a) = \sqrt{a}\mu$
Experiment 4	$c(a) = a^{1.5}$	$\mu(a) = a^2\mu$
Experiment 5	$c(a) = a^2$	$\mu(a) = \sqrt{a}\mu$
Experiment 6	$c(a) = \sqrt{a}$	$\mu(a) = \sqrt[3]{a}\mu$
Experiment 7	$c(a) = a^2$	$\mu(a) = a^{1.5}\mu$

**Fig. 2** Optimal action  $a$  as a function of  $x$  for experiment 1 to 4

Note that in the experiments we did not consider the service time requirement that is imposed in the system. Evidently, adding the restriction  $\mathbb{E}S \leq \alpha$  subject to the stability condition discussed in Sect. 2 will lead to a different optimal policy. However, the optimal policy under minimal costs when meeting the service requirement still has the *same structural properties*, i.e., a non-decreasing step function. The only difference with the results presented here is that this policy will be randomized in exactly one state (see Sect. 12.6 of Altman (1999)).

## 5 Conclusion

In this paper we have explored the structural properties of the optimal allocation policy in two different systems. Both systems are capable to change the number of computing resources dynamically. However, one system deals with updating the number of computing resources only upon the start of the service of a new job whilst in the other system that can be done during the service of a job. In both systems, one needs to optimize the resource allocation costs on the one hand while satisfying a service requirement on the sojourn time of a job on the other hand. We applied dynamic programming to show that the optimal resource allocation policy has a work-conservation property. It follows a step function with as extreme policy the bang-bang control policy. Also, we provide the conditions under which the bang-bang control policy is optimal. The techniques to show these results are not obtained



**Fig. 3** Optimal action  $a$  as a function of  $x$  for the limited resource allocation model

by standard induction arguments, and thus provide a foundation for studying generalized systems in which these techniques can be applied.

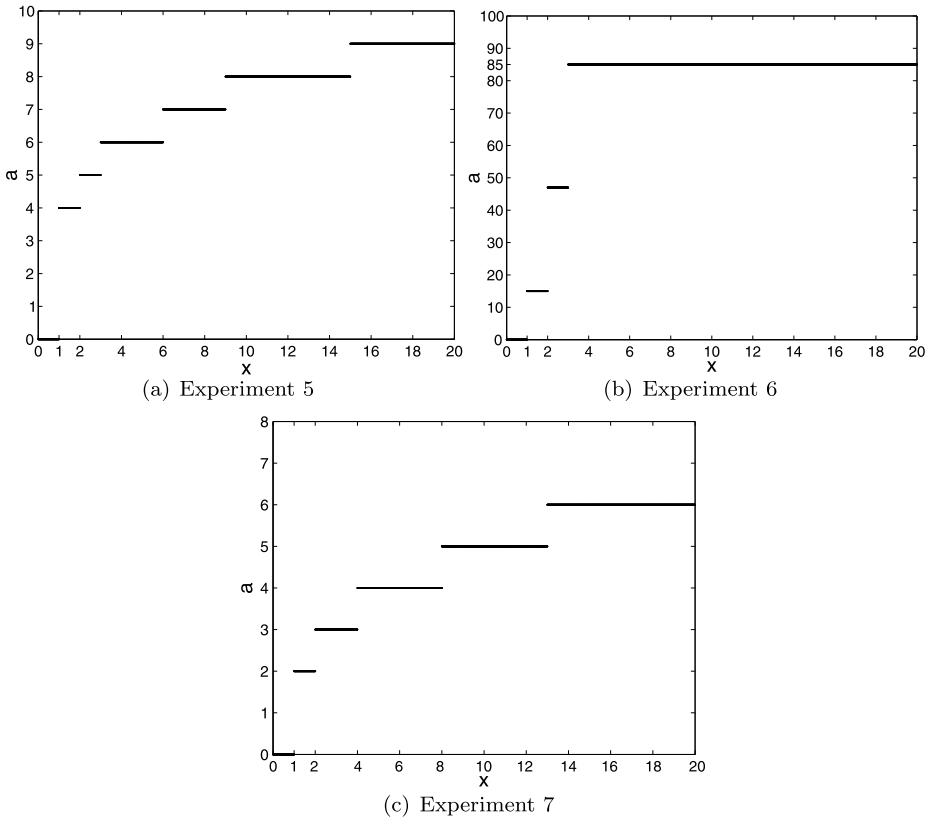
**Open Access** This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

**Appendix A: Limited resource allocation policy**

In the appendix, we add the proofs of Property 3.1, 3.2, 3.3, and 3.4.

*Proof of Property 3.1* The proof is by induction on  $n$  in  $V_n$ . Define  $V_0(x, a) = 0$  for all states  $x$  and actions  $a$ . Then, clearly,  $V_0(x, a)$  is increasing in all components of  $x$ . Now, assume that  $V_n(x + 1, a) - V_n(x, a) \geq 0$  for some  $n \in \mathbb{N}$ . Now, we prove that  $V_{n+1}(x, a)$  satisfies the increasingness property as well. Then

$$\begin{aligned}
 V_{n+1}(x + 1, a) - V_{n+1}(x, a) &= \frac{\tau}{\lambda} + \lambda[H_n(x + 2, a) - H_n(x + 1, a)] \\
 &\quad + \mu(a)[H_n(x, 0) - H_n([x - 1]^+, 0)] \\
 &\quad + [\mu(A) - \mu(a)][V_n(x + 1, a) - V_n(x, a)]. \quad (11)
 \end{aligned}$$



**Fig. 4** Optimal action  $a$  as a function of  $x$  for the fully flexible resource allocation model

Note that the first term  $\tau/\lambda$  and the last term with  $V_n(x + 1, a) - V_n(x, a)$  are positive. Hence, based on the induction hypothesis, we have

$$\begin{aligned}
 V_{n+1}(x + 1, a) - V_{n+1}(x, a) &> \lambda[H_n(x + 2, a) - H_n(x + 1, a)] \\
 &\quad + \mu(a)[H_n(x, 0) - H_n([x - 1]^+, 0)].
 \end{aligned}
 \tag{12}$$

Let  $b \in \arg H_n(x + 2, a)$  and  $c \in \arg H_n(x, 0)$ . Then,

$$\begin{aligned}
 V_{n+1}(x + 1, a) - V_{n+1}(x, a) &> \lambda[V_n(x + 2, a + b) - V_n(x + 1, a + b)] \\
 &\quad + \mu(a)[V_n(x, c) - V_n([x - 1]^+, c)] \\
 &\geq 0.
 \end{aligned}
 \tag{13}$$

Clearly, the inequality above holds because of the induction hypothesis. Hence, we conclude, by taking the limit as  $n \rightarrow \infty$ , that  $V(x, a)$  is increasing in  $x$ . □

Property 3.1 shows that the costs that the system incurs increases as the number of customers in the system increases. In fact, more can be said about the rate at which the costs increase; the increase in costs is higher when more customers are in the system. Hence, this



implies that the relative value function is a convex function. In the sequel we will show that this is indeed true. However, before doing so, we need a preparative lemma.

**Lemma A.1** *The value function satisfies the following property:*

$$H(x + 1, 0) - H(x, 0) - V(x, 0) + V(x - 1, 0) < 0,$$

for all  $x \geq 1$ .

*Proof* Let  $x \geq 1$ , then

$$V(x, 0) - V(x - 1, 0) = \frac{\tau}{\lambda} + \lambda[H(x + 1, 0) - H(x, 0)] + \mu(A)[V(x, 0) - V(x - 1, 0)].$$

Since  $\lambda + \mu(A) = 1$ , the equation above implies,

$$\lambda[V(x, 0) - V(x - 1, 0)] = \frac{\tau}{\lambda} + \lambda[H(x + 1, 0) - H(x, 0)].$$

Therefore,

$$\lambda[H(x + 1, 0) - H(x, 0) - V(x, 0) + V(x - 1, 0)] = -\frac{\tau}{\lambda}.$$

Thus,  $H(x + 1, 0) - H(x, 0) - V(x, 0) + V(x - 1, 0) < 0$ , since  $-\tau/\lambda < 0$ . □

Lemma A.1 is almost the inequality that represents convexity of the value function. This would be the case when  $H$  would be replaced by  $V$ . However, for the proof of convexity, we need three additional properties to hold as well. The following proof makes these properties explicit.

*Proof of Property 3.2* The proof is by induction on  $n$  in  $V_n$ . Define  $V_0(x, a) = 0$  for all states  $x$  and actions  $a > 0$  and  $V_0(x, 0) = \epsilon > 0$  for all  $x$ . Then, clearly,  $V_0(x, a)$  satisfies all properties. Now suppose that the properties hold for some  $n \in \mathbb{N}$ . We prove that the properties also hold for  $n + 1$ . Therefore, we start with convexity first.

**Property (i)** *Let  $x \geq 1$  and suppose that  $a = 0$ . Then,*

$$\begin{aligned} &V_{n+1}(x + 1, 0) - 2V_{n+1}(x, 0) + V_{n+1}(x - 1, 0) \\ &= \lambda[H_n(x + 2, 0) - 2H_n(x + 1, 0) + H_n(x, 0)] \\ &\quad + \mu(A)[V_n(x + 1, 0) - 2V_n(x, 0) + V_n(x - 1, 0)] \\ &\geq \lambda[H_n(x + 2, 0) - 2H_n(x + 1, 0) + H_n(x, 0)] \\ &\geq 0. \end{aligned} \tag{14}$$

The equality following by expanding  $V_{n+1}$  into  $V_n$ . The first inequality follows by using Property (i) of the induction hypothesis. The last inequality follows by using Property (iv) of the induction hypothesis.

Now let  $x \geq 1$  and suppose that  $a > 0$ . Then

$$V_{n+1}(x + 1, a) - 2V_{n+1}(x, a) + V_{n+1}(x - 1, a)$$

$$\begin{aligned}
 &= \lambda[V_n(x + 2, a) - 2V_n(x + 1, a) + V_n(x, a)] \\
 &\quad + \mu(a)[H_n(x, 0) - 2H_n([x - 1]^+, 0) + H_n([x - 2]^+, 0)] \\
 &\quad + [\mu(A) - \mu(a)][V_n(x + 1, a) - 2V_n(x, a) + V_n(x - 1, a)] \\
 &\geq \mu(a)[H_n(x, 0) - 2H_n([x - 1]^+, 0) + H_n([x - 2]^+, 0)] \\
 &\geq 0.
 \end{aligned} \tag{15}$$

The equality following by expanding  $V_{n+1}$  into  $V_n$ . The first inequality follows by using Property (i) of the induction hypothesis. The last inequality follows by using Property (iv) of the induction hypothesis. Thus, for all  $x \geq 1$  and  $a \geq 0$ ,  $V_{n+1}(x + 1, a) - 2V_{n+1}(x, a) + V_{n+1}(x - 1, a) \geq 0$ .

**Property (ii)** Let  $x \geq 0$  and suppose  $a > 0$ . Then, based on the optimality equation, we have

$$\begin{aligned}
 V_{n+1}(x, a) - V_{n+1}([x - 1]^+, a) &= \frac{\tau}{\lambda} + \lambda[V_{n+1}(x + 1, a) - V_{n+1}(x, a)] \\
 &\quad + \mu(a)[H_{n+1}([x - 1]^+, 0) - H_{n+1}([x - 2]^+, 0)] \\
 &\quad + [\mu(A) - \mu(a)][V_{n+1}(x, a) - V_{n+1}([x - 1]^+, a)].
 \end{aligned}$$

Recall that the uniformization constant  $\lambda + \mu(A) = 1$ . Thus, the equation above is equivalent to

$$\begin{aligned}
 \lambda[V_{n+1}(x, a) - V_{n+1}([x - 1]^+, a)] &= \frac{\tau}{\lambda} + \lambda[V_{n+1}(x + 1, a) - V_{n+1}(x, a)] \\
 &\quad + \mu(a)[H_{n+1}([x - 1]^+, 0) - H_{n+1}([x - 2]^+, 0)] \\
 &\quad - \mu(a)[V_{n+1}(x, a) - V_{n+1}([x - 1]^+, a)].
 \end{aligned}$$

The equation above implies that

$$\begin{aligned}
 &\mu(a)[V_{n+1}(x, a) - V_{n+1}([x - 1]^+, a) - H_{n+1}([x - 1]^+, 0) + H_{n+1}([x - 2]^+, 0)] \\
 &= \frac{\tau}{\lambda} + \lambda[V_{n+1}(x + 1, a) - 2V_{n+1}(x, a) + V_{n+1}([x - 1]^+, a)].
 \end{aligned}$$

Hence, by using Property (i) of the induction hypothesis and because of  $\tau > 0$ , the right-hand side of the equation is positive. Hence, we have  $V_{n+1}(x, a) - V_{n+1}([x - 1]^+, a) - H_{n+1}([x - 1]^+, 0) + H_{n+1}([x - 2]^+, 0) > 0$ .

**Property (iii)** We prove the property by means of contradiction. Assume that there exists a  $x \geq 2$  such that the minimal element of  $\arg H(x, 0)$  is 0, i.e.,  $\min\{\arg H(x, 0)\} = 0$ . This, by definition, implies that  $V_{n+1}(x, 0) = H_{n+1}(x, 0)$ . Therefore,

$$\begin{aligned}
 &H_{n+1}(x, 0) - H_{n+1}(x - 1, 0) - V_{n+1}(x - 1, 0) + V_{n+1}(x - 2, 0) \\
 &\geq V_{n+1}(x, 0) - 2V_{n+1}(x - 1, 0) + V_{n+1}(x - 2, 0) \\
 &\geq 0.
 \end{aligned}$$

The first inequality follows by taking action  $a = 0$  in the second term  $H_{n+1}(x - 1, 0)$ . The second inequality follows by Property (i) of the induction hypothesis. However, based on Lemma A.1 we know that  $H_{n+1}(x, 0) - H_{n+1}(x - 1, 0) - V_{n+1}(x - 1, 0) + V_{n+1}(x - 2, 0) < 0$ . Therefore, we conclude that  $\min\{\arg H(x, 0)\} > 0$  for  $x \geq 2$ .

**Property (iv)** Let  $x \geq 1$ . Since  $x - 1 \geq 0$ , we have  $x + 1 \geq 2$ . Thus by using Property (iii) of the induction hypothesis, we have  $a^*(x) := \min\{\arg H(x, 0)\} > 0$ . Therefore,

$$\begin{aligned} &H_{n+1}(x + 1, 0) - 2H_{n+1}(x, 0) + H_{n+1}(x - 1, 0) \\ &\geq V_{n+1}(x + 1, a^*(x + 1)) - V_{n+1}(x, a^*(x + 1)) - H_{n+1}(x, 0) + H_{n+1}(x - 1, 0) \\ &\geq 0. \end{aligned}$$

The first inequality follows by taking action  $a^*(x + 1)$  in  $H_{n+1}(x, 0)$ . The second inequality follows by Property (ii) of the induction hypothesis.

We conclude the proof by taking the limit as  $n \rightarrow \infty$ . □

Property 3.2 shows that the relative value function is convex. However, in proving this one needs three additional properties simultaneously in the proof by induction Property (i) depends on (iv), which depends on (ii) and (iii). Now, we are ready to study the submodularity property of the optimal policy.

*Proof of Property 3.3* We prove the submodularity property by induction on  $n$  in  $V_n$ . Let  $V_0(x, a) = 0$ . Clearly, the submodularity property holds. Now assume that the property holds for some  $n \in \mathbb{N}$  and for all  $x \geq 0$ . We proceed to prove that  $V_{n+1}(x, a)$  satisfies the property as well. Therefore, fix  $x \geq 1$  and  $a > 0$ , then

$$\begin{aligned} &V_{n+1}(x, a + k) - V_{n+1}(x, a) - V_{n+1}(x + 1, a + k) + V_{n+1}(x + 1, a) \\ &= \lambda[V_n(x + 1, a + k) - V_n(x + 1, a) - V_n(x + 2, a + k) + V_n(x + 2, a)] \\ &\quad + [\mu(a + k) - \mu(a)][H_n(x - 1, 0) - H_n(x, 0)] \\ &\quad + \mu(A)[V_n(x, a + k) - V_n(x, a) - V_n(x + 1, a + k) + V_n(x + 1, a)] \\ &\quad - \mu(a + k)V_n(x, a + k) + \mu(a)V_n(x, a) \\ &\quad + \mu(a + k)V_n(x + 1, a + k) - \mu(a)V_n(x + 1, a) \\ &= \lambda[V_n(x + 1, a + k) - V_n(x + 1, a) - V_n(x + 2, a + k) + V_n(x + 2, a)] \\ &\quad + [\mu(a + k) - \mu(a)][H_n(x - 1, 0) - H_n(x, 0)] \\ &\quad + \mu(A)[V_n(x, a + k) - V_n(x, a) - V_n(x + 1, a + k) + V_n(x + 1, a)] \\ &\quad - [\mu(a + k) - \mu(a)]V_n(x, a + k) - \mu(a)V_n(x, a + k) + \mu(a)V_n(x, a) \\ &\quad + [\mu(a + k) - \mu(a)]V_n(x + 1, a + k) + \mu(a)V_n(x + 1, a + k) - \mu(a)V_n(x + 1, a) \\ &= \lambda[V_n(x + 1, a + k) - V_n(x + 1, a) - V_n(x + 2, a + k) + V_n(x + 2, a)] \\ &\quad + [\mu(A) - \mu(a)][V_n(x, a + k) - V_n(x, a) - V_n(x + 1, a + k) + V_n(x + 1, a)] \\ &\quad + [\mu(a + k) - \mu(a)][V_n(x + 1, a + k) - V_n(x, a + k) - H_n(x, 0) \\ &\quad + H_n(x - 1, 0)]. \end{aligned} \tag{16}$$

The first equality follows from expanding  $V_{n+1}$  into  $V_n$ . The second equality follows from adding and subtracting  $\mu(a)V_n(x, a + k)$  and  $\mu(a)V_n(x + 1, a + k)$ . The third equality follows from standard algebraic manipulations. Based on the induction hypothesis, we have

$$\begin{aligned} &V_{n+1}(x, a + k) - V_{n+1}(x, a) - V_{n+1}(x + 1, a + k) + V_{n+1}(x + 1, a) \\ &\geq [\mu(a + k) - \mu(a)][V_n(x + 1, a + k) - V_n(x, a + k) - H_n(x, 0) \\ &\quad + H_n(x - 1, 0)]. \end{aligned} \tag{17}$$

By using Property (ii) of Property 3.2 and by noting that  $\mu(\cdot)$  is a strictly increasing function, we obtain

$$V_n(x + 1, a + k) - V_n(x, a + k) - H_n(x, 0) + H_n(x - 1, 0) > 0. \tag{18}$$

Thus, we have shown that  $[V(x, a + k) - V(x, a)] - [V(x + 1, a + k) - V(x + 1, a)] > 0$ .  $\square$

Based on the submodularity property of the optimal policy, we can study monotonicity properties of the optimal policy. The following proof formalizes this statement posed in Property 3.4.

*Proof of Property 3.4* For  $x = 0$ , the only feasible action in  $\arg H(0, 0)$  is  $a_0 = 0$ , since there are no customers to serve. For  $x = 1$ , if  $a_1 = 0 \in \arg H(1, 0)$ , then all  $a_2 \in \arg H(2, 0)$ , we have  $a_2 > a_1$  [Property (iii) of Property 3.2]. Hence,  $a_2 > a_1 \geq a_0$ . Now, for  $x \geq 1$  and  $a_1 > 0$  it suffices to show that the relative value function satisfies an extension of submodularity, namely

$$[V(x, a + k) - V(x, a)] - [V(x + 1, a + k) - V(x + 1, a)] > 0, \tag{19}$$

for all  $k > 0$ . If this property holds, then since  $V(x + 1, a_{x+1} + k) - V(x + 1, a_{x+1}) \geq 0$ , we have that  $V(x, a_{x+1} + k) - V(x, a_{x+1}) > V(x + 1, a_{x+1} + k) - V(x + 1, a_{x+1}) \geq 0$  with  $a_{x+1}$  the minimal element of  $\arg H(x + 1, 0)$ . Hence, this implies that all minimizing actions  $a_x \in \arg H(x, 0)$  in state  $x$  satisfy  $a_x \leq a_{x+1}$ .

Based on Property 3.3, we complete the proof as this property implies that  $a_x$  is an increasing function in  $x$ .  $\square$

### Appendix B: Full flexible resource allocation policy

In the appendix, we add the proofs of Properties 3.9 and 3.10.

*Proof of Property 3.9* The proof is by induction on  $n$ . Let  $V_0(x) = x^2$  for all  $x \geq 0$ . Then  $V_0(x + 1) - V_0(x) = 2x + 1 > 0$ , and  $V(x + 1) - 2V(x) + V(x - 1) = 2 > 0$ . Therefore,  $V_0$  is a convex increasing function in  $x$ . Now, assume that the statement holds for  $k$ , then we prove that the statement also holds for  $k + 1$ . To this end, define

$$T_a^k(x) = \mu(a)V_k([x - 1]^+) + [\mu(A) - \mu(a)]V_k(x) + c(a).$$

Then, for  $x \geq 0$ , we have

$$V_{k+1}(x + 1) - V_{k+1}(x) = \frac{\tau}{\lambda} + [\lambda V_k(x + 2) - \lambda V_k(x + 1)]$$

$$\begin{aligned}
 &+ [\min_a\{T_a^k(x + 1)\} - \min_a\{T_a^k(x)\}] \\
 &> \min_a\{T_a^k(x + 1)\} - \min_a\{T_a^k(x)\}.
 \end{aligned}$$

The inequality holds because the first two terms are positive. Let  $a^*$  be the smallest element of  $\arg \min_a\{T_a^k(x + 1)\}$ , then, for  $x \geq 0$ , we have

$$\begin{aligned}
 V_{k+1}(x + 1) - V_{k+1}(x) &> T_{a^*}^k(x + 1) - \min\{T_a^k(x)\} \\
 &\geq T_{a^*}^k(x + 1) - T_{a^*}^k(x) \\
 &= \mu(a^*)V_k(x) + [\mu(A) - \mu(a^*)]V_k(x + 1) + c(a^*) \\
 &\quad - \mu(a^*)V_k([x - 1]^+) - [\mu(A) - \mu(a^*)]V_k(x) - c(a^*) \\
 &> 0.
 \end{aligned}$$

Therefore, by induction, we derive that  $V(x + 1) - V(x) > 0$ .

Now we proceed to prove convexity of the relative value function. Assume that convexity holds for  $k$ , then we need to prove convexity for  $k + 1$ . Then, for  $x \geq 0$ , we have

$$\begin{aligned}
 &V_{k+1}(x + 1) - 2V_{k+1}(x) + V_{k+1}(x - 1) \\
 &= [\lambda V_k(x + 2) - 2\lambda V_k(x + 1) + \lambda V_k(x)] \\
 &\quad + [\min_a\{T_a^k(x + 1)\} - 2\min_a\{T_a^k(x)\} + \min_a\{T_a^k(x - 1)\}] \\
 &> \min_a\{T_a^k(x + 1)\} - 2\min_a\{T_a^k(x)\} + \min_a\{T_a^k(x - 1)\}.
 \end{aligned}$$

The inequality holds because the first expression between the brackets is positive due to the induction hypothesis. Now assume that  $a_1^*$  is the smallest element of  $\arg \min_a\{T_a^k(x + 1)\}$  and  $a_2^*$  is the smallest element of  $\arg \min_a\{T_a^k(x - 1)\}$ . Then, we have

$$\begin{aligned}
 &\min_a\{T_a^k(x + 1)\} - 2\min_a\{T_a^k(x)\} + \min_a\{T_a^k(x - 1)\} \\
 &\geq [c(a_1^*) - c(a_1^*) - c(a_2^*) + c(a_2^*)] \\
 &\quad + [\mu(a_1^*)V_k(x) - \mu(a_1^*)V_k([x - 1]^+) - \mu(a_2^*)V_k([x - 1]^+) + \mu(a_2^*)V_k([x - 2]^+)] \\
 &\quad + [(\mu(A) - \mu(a_1^*))V_k(x + 1) - (\mu(A) - \mu(a_1^*))V_k(x) - (\mu(A) - \mu(a_2^*))V_k(x) \\
 &\quad + (\mu(A) - \mu(a_2^*))V_k([x - 1]^+)] \\
 &= (\mu(a_1^*) - \mu(a_2^*)) [V_k(x) - V_k([x - 1]^+)] \\
 &\quad + \mu(a_2^*) [V_k(x) - 2V_k([x - 1]^+) + V_k([x - 2]^+)] \\
 &\quad + (\mu(A) - \mu(a_1^*)) [V_k(x + 1) - V_k(x)] \\
 &\quad - [\mu(A) - \mu(a_1^*) + (\mu(a_1^*) - \mu(a_2^*))] (V_k(x) - V_k([x - 1]^+)) \\
 &= \mu(a_2^*) [V_k(x) - 2V_k([x - 1]^+) + V_k([x - 2]^+)] \\
 &\quad + (\mu(A) - \mu(a_1^*)) [V_k(x + 1) - 2V_k(x) + V_k([x - 1]^+)] \\
 &\geq 0.
 \end{aligned}$$

The first inequality follows from taking a potentially suboptimal action in the second term of  $\min_a \{T_a^k(x+1)\} - 2 \min_a \{T_a^k(x)\} + \min_a \{T_a^k(x-1)\}$ . The two equalities follow by rearranging the terms. The last inequality follows by the induction hypothesis and by noting that  $\mu(A) - \mu(a_1^*)$  is positive. Hence, using mathematical induction we have proved that  $V(x)$  is a convex increasing function in  $x$ .  $\square$

*Proof of Property 3.10* Let  $a \in \arg \min_{a \in \mathcal{A}_{x+1}} \{T_a(x+1)\}$  and  $b \in \arg \min_{a \in \mathcal{A}_x} \{T_a(x)\}$  be an arbitrary optimal allocation in states  $x+1$  and  $x$ , respectively. The proof is by contradiction. Suppose that  $b$  and  $a$  such that  $a < b$ , then

$$\begin{aligned} T_a(x) - T_b(x) &= [\mu(b) - \mu(a)][V(x) - V([x-1]^+)] - [c_2(b) - c_2(a)] \\ &\geq 0. \end{aligned}$$

By Property 3.9 we have that  $V$  is a convex increasing function in  $x$ . Together with the fact that  $\mu(\cdot)$  is a strictly increasing function, and thus  $\mu(b) - \mu(a) > 0$ , we have

$$\begin{aligned} T_a(x+1) - T_b(x+1) &= [\mu(b) - \mu(a)][V(x+1) - V(x)] - [c_2(b) - c_2(a)] \\ &> [\mu(b) - \mu(a)][V(x) - V([x-1]^+)] - [c_2(b) - c_2(a)] \\ &\geq 0. \end{aligned}$$

However, this implies that  $a$  is not optimal for state  $x+1$ , since  $b$  has a smaller value. Hence,  $a \geq b$ , and this establishes the non-decreasing curve as stated in the theorem.  $\square$

## References

- Altman, E. (1999). *Constrained Markov decision processes*. London: Chapman and Hall.
- Aviv, Y., & Federgruen, A. (1999). The value-iteration method for countable state Markov decision processes. *Operations Research Letters*, 24(5), 223–234.
- Aziz, A., & El-Rewini, H. (2006). Grid resource allocation and task scheduling for resource intensive applications. In *Proceedings of the 2006 international conference on parallel processing workshops (ICPPW'06)*.
- Buyya, R., Abramson, D., & Giddy, J. (2001). A case for economy grid architecture for service oriented grid computing. In *Proceedings on the 15th international symposium on parallel and distributed processing*, April (pp. 776–790).
- Czajkowski, K., Foster, I., & Kesselman, C. (1999). Resource co-allocation in computational grids. In *Proceedings of the eighth IEEE international symposium on high performance distributed computing (HPDC-8)* (pp. 219–228).
- Daniel, G., & Chronopoulos, A. T. (2003). Algorithmic mechanism design for load balancing in distributed systems. *IEEE Transactions on Systems, Man and Cybernetics Part B Cybernetics*, 34(1), 77–84.
- Foster, I., Roy, A., & Sander, V. (2000). A quality of service architecture that combines resource reservation and application adaptation. In *Proceedings of the 8th international workshop on quality of service (IWQOS)* (pp. 181–188), Pittsburgh, PA, June.
- Guo, M., & Conitzer, V. (2010). Optimal in expectation redistribution mechanisms. *Artificial Intelligence*, 174, 363–381.
- Koole, G. M. (1998). Structural results for the control of queueing systems using event-based dynamic programming. *Queueing Systems*, 30, 323–339.
- Koole, G. M. (2006). Monotonicity in Markov reward and decision chains: theory and applications. *Foundations and Trends in Stochastic Systems*, 1, 1–76.
- Nurmi, D., Brevik, J., & Wolski, R. (2007). QBETS: Queue bounds estimation from time series. In *Proceedings of job scheduling strategies for parallel processing (JSSPP)*, June.
- Nurmi, D., Wolski, R., & Brevik, J. (2008). Probabilistic advanced reservations for batch-scheduled parallel machines. In *Proceedings of the 13th ACM SIGPLAN symposium on principles and practice of parallel programming* (Vol. 20–23, pp. 289–290), PPOPP 2008) Salt Lake City, UT, USA, February.

- Park, J. (2003). A scalable protocol for deadlock and livelock free co-allocation of resources in internet computing. In *Proceeding of the Symposium on Applications and the Internet*, January (pp. 66–73).
- Puterman, M. L. (1994). *Markov decision processes: discrete stochastic dynamic programming*. New York: Wiley.
- Rana, O. F., Winikoff, M., Padgham, L., & Harland, J. (2002). Applying conflict management strategies in BDI agents for resource management in computational grids author. In *Proceedings of the 8th international workshop on quality of service (IWQOS)* (pp. 205–214). Los Alamitos: IEEE Computer Society Press.
- Rykov, V. V. (2001). Monotone control of queueing systems with heterogeneous servers. *Queueing Systems*, 37(4), 391–403.
- Sandholm, T., Lai, K., Andrade, J., & Odeberg, J. (2006). Market-based resource allocation using price prediction in a high performance computing grid for scientific applications. In *Proceedings of the IEEE international symposium on high performance distributed computing (HPDC)* (pp. 19–23), Paris, France.
- Snoek, C. G. M., Worring, M., Geusebroek, J. M., Koelma, D. C., Seinstra, F. J., & Smeulders, A. W. M. (2006). The semantic pathfinder: Using an authoring metaphor for generic multimedia indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28, 1678–1689.
- Song, S., Hwang, K., & Kwok, Y. K. (2006). Risk-resilient heuristics and genetic algorithms for security-assured grid job scheduling. *IEEE Transactions on Computers*, 55, 703–719.
- The Distributed ASCI Supercomputer 3 (2011). <http://www.cs.vu.nl/das3>.
- Wakamiya, N., Yamashita, T., Murata, M., & Miyahara, H. (2002). Integrated resource allocation scheme for real-time video multicast. In *Global telecommunications conference IEEE GLOBECOM* (Vol. 2, pp. 1455–1459).
- Wang, X., & Luo, J. (2004). Architecture of Grid resource allocation management based on QoS. *Grid and Cooperative Computing*, 2, 81–88.
- Xie, T., & Qin, X. (2008). Security-aware resource allocation for real-time parallel jobs on homogeneous and heterogeneous clusters. *IEEE Transactions on Parallel and Distributed Systems*, 19, 682–697.
- Yang, R., Bhulai, S., & van der Mei, R. D. (2011). Optimal resource allocation for multi-queue systems with a shared server pool. *Queueing Systems*, 68(2), 133–163.
- Yang, R., Bhulai, S., van der Mei, R. D., & Seinstra, F. (2011). Optimal resource allocation for time-reservation systems. *Performance Evaluation*, 68(5), 414–428.