Journal of King Saud University – Computer and Information Sciences (2017) 29, 74–84

King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

Executed-time Round Robin: EtRR an online non-clairvoyant scheduling on speed bounded processor with energy management

Pawan Singh *, Berhane Wolde-Gabriel

School of Informatics, Institute of Technology, Hawassa University, Ethiopia

Received 20 July 2015; revised 11 March 2016; accepted 17 March 2016
Available online 31 March 2016

KEYWORDS

Weighted flow time;
Power Management;
Non-clairvoyant scheduling;
Online scheduling;
Potential analysis

Abstract Energy conservation has become a prime objective due to excess use and huge demand of energy in data centers. One solution is to use efficient job scheduling algorithms. The scheduler has to maintain the machine's state balance to obtain efficient job schedule and avoid unnecessary energy consumption. Although the practical importance of non-clairvoyant scheduling problem is higher than clairvoyant scheduling, in the past few years the non-clairvoyant scheduling problem has been studied lesser than clairvoyant scheduling. In this paper, an online non-clairvoyant scheduling problem is studied to minimize total weighted flow time plus energy and a scheduling algorithm Executed-time Round Robin (EtRR) is proposed. Generally, weights of jobs are system generated and they are assigned to jobs at release/arrival time. In EtRR, the weights are not generated by the system, rather by the scheduler using the executed time of jobs. EtRR is a coupling of weighted generalization of Power Management and Weighted Round Robin (WRR). We adopt the conventional power function $P = s^{\alpha}$, where s and $\alpha > 1$ are speed of a processor and a constant, respectively. EtRR is $O(1)$-competitive, it is using a processor with the maximum speed $(1 + \tau/3)T$, where the maximum speed of optimal offline adversary is T and $0 < \tau \leqslant (3\alpha)^{-1}$.

1. Introduction

"What matters most to the computer designers at Google is not speed, but power, low power, because data centers can consume as much energy as a city." (Markoff and Lohr, 2002). In the current epoch, energy conservation is a key issue in designing modern processors. Dynamic speed scaling is adapted by many chip manufacturers and they produce associated software also such as AMD's PowerNow. These softwares ease an operating system to scale the processor's speed and obtain energy efficiency. Modern scheduling

* Corresponding author.
E-mail addresses: pawansingh3@yahoo.com, pawansingh51279@gmail.com, dr_pawansingh@hu.edu.et (P. Singh), berhane.woldegabriel@gmail.com (B. Wolde-Gabriel).
Peer review under responsibility of King Saud University.

ELSEVIER Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jksuci.2016.03.002
1319-1578

algorithms comprise of two components: first, a job selection policy that determines which job to execute; second, a speed scaling policy that computes the execution speed of a processor, at any time.

An operating system has dual conflicting objectives to solve such problems: first, to optimize some scheduling Quality of Service (QoS) objective; second, Power Management (PM) objective, such as total weighted flow time and total energy used, respectively (Bansal et al., 2009). Scheduling jobs becomes complicated, if QoS, speed scaling and energy usage efficiency are considered at once. A scheduler arranges jobs in some order to optimize a certain QoS metric, such as throughput, makespan, slowdown, flow time or weighted flow time. In most of the operating systems (such as UNIX), when job arrives there is no information about the job's size. In clairvoyant (non-clairvoyant) scheduling algorithms the sizes of jobs are known (not known) at the release time. Unlike online, offline algorithms know complete job sequence in advance, which is not possible in most of the practical problems. Yao et al. (1995) initiated the theoretical study of speed scaling and proposed a model, wherein the processor's speed s can vary from zero to infinity, i.e., $[0, \infty)$. The traditional power consumption function is $P = s^{\alpha}$, where $\alpha > 1$ a constant, s speed of a processor and P is the power consumed (the value of $\alpha = 2$ or 3 for CMOS-based chips (Pruhs et al., 2008)). There are two speed models, unbounded speed and bounded speed model, where the speed ranges are $[0, \infty)$ and $[0, T]$, respectively (Bansal et al., 2009). Kalyanasundaram and Pruhs (2000) introduced an idea to augment the resources of the non-clairvoyant scheduler by increasing the processor's speed. As per Kalyanasundaram and Pruhs (2000), if a non-clairvoyant scheduler is allowed $(1 + \tau)$ times faster processor, then it can attain a response time within a $(1 + 1/\tau)$ factor achievable by the best possible clairvoyant algorithm.

Motwani et al. (1994) first analyzed non-clairvoyant scheduling algorithm for the objective of mean response time and showed that Round Robin (RR) has a performance ratio of $\left(2 - \frac{2}{(n+1)}\right)$, which is optimal for deterministic non-clairvoyant algorithms; they proved that the lower bounds remain equal for the jobs of bounded sizes, i.e. the ratio of the largest to the smallest execution time is bounded by some small constant. RR makes $\Omega(x)$ preemptions for a job of size x. The randomized algorithms have the same performance ratio as RR. Any deterministic non-clairvoyant dynamic algorithm has performance ratio $\Omega(n^{1/3})$, while any randomized non-clairvoyant dynamic algorithm has a performance ratio $\Omega(\log n)$. Muthukrishnan et al. (1999) studied uniprocessor online job scheduling algorithm with slowdown or stretch as their objective and showed that SRPT is 2-competitive but in clairvoyant settings. Berman and Coulston (1999) considered the problem of online preemptive non-clairvoyant scheduling (Balance) on a uniprocessor model for the objective of minimizing the total response time. Balance schedules the least processed job first. Berman and Coulston (1999) proved that if the Balance runs υ times faster than the clairvoyant algorithm then the competitive ratio is $(\upsilon/(\upsilon - 1))$ at most and for $\upsilon \geqslant 2$ the competitive ratio of Balance is $(2/\upsilon)$; they concluded that adequately high speed is more powerful than clairvoyance. Edmonds (2000) achieved $\Omega(\sqrt{n})$ lower bound on competitive ratio of sequential and parallelizable jobs for randomized non-clairvoyant schedulers; if the speed of processor is $(1 + \tau)$, then the lower bound is $\Omega(\frac{1}{\tau})$. Edmonds (2000) proved that after the resource augmentation, when speed of a processor $s > 2$, the Equi-partition and Processor Scheduling (Round Robin), which shares the processor equally among all jobs, becomes competitive. In case, if there are p processors of $(2 + \tau)$ speed, then the competitive ratio of Equi-partition is between $\frac{2}{3}\left(1 + \frac{1}{\tau}\right)$ and $\left(2 + \frac{4}{\tau}\right)$; with extra augmentation, when $s \geqslant 4$ the competitive ratio of Equi-partition is between $\left(\frac{2}{s}\right)$ and $\left(\frac{16}{s}\right)$.

As per Kalyanasundaram and Pruhs (2003) and Becchetti and Leonardi (2004), randomized version of Multi Level Feedback Queue algorithm is $O(logn)$-competitive. Yun and Kim (2003) proposed that it is NP-hard to calculate a minimum energy schedule for jobs with fixed priority. Becchetti et al. (2006) showed the modification in Bansal's algorithmic result and gave $O(\alpha^2/log^2\alpha)$ competitive algorithm with resource augmentation for the objective of minimizing weighted flow time plus energy. Bansal et al. (2007) showed that the algorithm Optimal Available (OA) is -competitive using the potential analysis and the competitive ratio is $\mu_\tau\gamma$, where $\gamma = \max\left\{2, \frac{2(\alpha-1)}{\alpha-(\alpha-1)^{1-1/\alpha-1}}\right\}$ and $\mu_\tau = \max\{(1 + 1/\tau), \ (1 + \tau)\alpha\}$ for any $\tau > 0$. Bansal et al. (2009) assumed that allowable speeds are countable collection of disjoint subintervals in range $[0, \infty)$ and they have taken the power functions that are non-negative, continuous and differentiable. Bansal et al. (2009) used SRPT for job selection and the speed scaling such that at any time the speed is equal to one plus number of unfinished jobs, their algorithm is $(3 + \tau)$-competitive for the objective of total flow time plus energy. Bansal et al. (2009) considered Highest Density First (HDF) also for job selection and the speed scaling such that at any time the speed is equal to fractional weigh of unfinished jobs, and gave a $(2 + \tau)$-competitive algorithm for the objective of fractional weighted flow time plus energy.

In multiprocessor systems, a new concept of sleep management, QoS and energy consumption were used by Albers (2010). The non-clairvoyant speed scaling scheduling algorithm LAPS proposed by Gupta et al. (2012) is $(1 + \tau)$-speed, $O(1/\tau^5)$-competitive for the objective of minimizing the flow time plus energy on related machines. Gupta et al. (2012) gave the first scalable non-clairvoyant algorithm for speed-scalable heterogeneous processors for fixed-speed related machines and suggested that scheduling heterogeneous multiprocessors might be inherently more complex than scheduling homogeneous multiprocessors, or at least, require significantly unlike algorithms. Chan et al. (2013) gave an online non-clairvoyant deterministic algorithm Scheduling with Arrival Time Alignment (SATA) with sleep management, which is $(1 + \tau)$-speed, $O(1/\tau^2)$-competitive for the objective of minimizing total flow time plus energy. SATA uses mechanism called the arrival-time-alignment to ensure the even jobs distribution when a job arrives or finishes, wherein it migrates each job at most four times on an average. In classical settings with no sleep management SATA is $(1 + \tau)$-speed, $8(1 + 1/\tau)^2$-competitive for the objective to minimize flow time only.

Fox et al. (2013) proposed a non-clairvoyant algorithm Weighted Latest Arrival Processor Sharing with Energy (WLAPS + E), which is $(1 + 6\tau)$-speed $(5/\tau^2)$-competitive, where $0 < \tau \leqslant 1/6$, for the objective of weighted flow time plus energy. WLAPS + E schedules late arriving jobs and a job can use number of machines proportioned by job weight. In WLAPS + E all processors are not taken in use, rather some processors remain inactive to save energy. Sun et al. (2014) studied non-clairvoyant scheduling algorithm Non-uniform Equi-partitioning (N-EQUI) for a set of parallel jobs in two circumstances: first, where all jobs are released at the same time; second, where jobs are coming over time, i.e. with arbitrary release time. Sun et al. (2014) proved that N-EQUI is $O(ln^1/\alpha P)$-competitive and $O(lnP)$-competitive for the objective of minimizing the total flow time plus energy in first and second circumstances, respectively, where P is the total number of processors. Bell and Wong (2014) proposed an $2^{4\alpha}(log^\alpha P + \alpha^\alpha 2^{\alpha-1})$-competitive deterministic online energy efficient deadline scheduling algorithm Dual-Classified Round Robin (DCRR) on multiprocessors, where P is the ratio of the maximum to the minimum job size. DCRR uses traditional power function and classify the jobs according to densities as well as sizes. Im et al. (2015) gave the first analysis of the instantaneously fair algorithm Round Robin (RR), which is $2k(1 + 10\tau)$-speed $O(k/\tau)$-competitive for all $k \geqslant 1$ and the l_k-norms of flow time for temporal fairness in the multiple identical machines setting (general meaning of l_k-norms of flow time considered is $(\sum_j (C_j - r_j)^k)^{1/k}$). At any time, if jobs are more than machines, allocate machines to jobs equally or process each job on a machine completely. Angelopoulos et al. (2015) proposed a framework to study online scheduling algorithms on a uniprocessor system. This framework is based on primal–dual and dual-fitting techniques for design and analysis of algorithms to solve generalized flow time problems (GFP). The proofs are independent of potential functions and based on intuitive geometric interpretations of the primal/dual objectives. In their primal–dual approach when a new job arrives, the dual variables for jobs can be updated without affecting the past portion of the schedule. Angelopoulos et al. (2015) proved that Highest Density First (HDF) is $(1 + \tau)$-speed $(1 + \tau/\tau)$-competitive for GFP with concave functions; this reflects the improvement in analysis of WLAPS (Im et al., 2014b), which is $(1 + \tau)$-speed $O(1 + \tau/\tau^2)$-competitive.

We study online non-clairvoyant speed scaling algorithm against an offline adversary. The objective considered is to minimize weighted flow time plus energy consumption. In this paper, the analysis of online non-clairvoyant algorithm is presented using competitive analysis, i.e. the worst case comparison of an online algorithm and optimal offline algorithm. To minimize the cost function of weighted flow time plus energy, an online algorithm is c-competitive, if for any input the cost incurred is never more than c times the cost of optimal offline algorithm. The objective of minimizing weighted flow time plus energy consumption has a natural interpretation, as it can be measured in monetary terms (Chan et al., 2011a). The assumption perceives that the user is eager to pay a unit of energy to decrease certain units (say ρ units) of weighted flow time. Energy is of more concern if there is a large value of ρ and if $\rho = 0$ then the problem is converted to the traditional weighted flow time scheduling. In this paper, an online non-clairvoyant scheduling algorithm Executed-time Round Robin (EtRR) is proposed, wherein the weights of jobs are not system generated, rather they are generated using the executed time of a job by the scheduler, i.e. current time minus release time of a job. The resource augmentation is used along with speed bounded model.

The rest of the paper is divided into the following sections: Section 2 describes notations used in our paper and definitions necessary for discussion. In Section 3, we have given some scheduling algorithms related to our work and their results. In Section 4, we present the online non-clairvoyant algorithm Executed-time Round Robin (EtRR) and compare EtRR against an optimal offline algorithm Opt using amortized analysis (potential function). Section 5 draws some concluding remarks and future scope of our study.

2. Definitions and notations

We study an online non-clairvoyant job scheduling in a uniprocessor environment, wherein jobs arrive over time, job sequence is not known until job arrives, release time is known at job arrival only and size of job j is known only when job j completes. The speeds of a processor, used by the offline adversary and EtRR, can range from zero to T i.e. $[0, T]$ and from zero to $\left(1 + \frac{\tau}{3}\right)T$, respectively. Jobs can preempt with no penalty. The notations used in this paper are listed in Table 1. We use the traditional power function $P = s^\alpha$, where $\alpha > 1$, a fixed constant and s speed of a processor. Per unit time, a processor executes s units of work, if processor's speed is s. Consider that there is some schedule S of any job set I. At any given time t, a job j is active if its release time is less than current time and job is not completed, i.e., $r(j) \leqslant t$ and $p(j, t) > 0$. The executed time $e_{xj}(t)$ or $e_x(j)$ of a job j up to time t is current time minus release time, i.e., $(t - r(j))$. The weight of a job j at any time t is one plus executed time of job j, i.e., weight $we(j) = (1 + e_x(j)) = (1 + t - r(j))$. The flow time $F(j)$ of a job j is the time elapsed since j arrived and until job j is completed. The total weighted flow F is $\sum_{j\in I} we(j)F(j)$ or equivalently $int_0^\infty we(t)dt$. Our objective is to minimize total weighted flow time plus energy, denoted by $G = F + E$. The total energy usage E for the scheduling is $\int_0^\infty s(t)^\alpha dt$.

Let Opt be an optimal offline algorithm such that for any job set/sequence I, weighted flow plus energy $F_{Opt(I)} + E_{Opt(I)}$ of Opt is minimized among all schedules of I. An algorithm ALG is said to be c-competitive for any $c \geqslant 1$, if for all jobs sequence I the following inequality is satisfied-

$$F_{ALG(I)} + E_{ALG(I)} \leqslant c \cdot (F_{Opt(I)} + E_{Opt(I)})$$

3. Related work

Lam et al. (2008) showed that an online clairvoyant scheduling SRPT-AJC is $\frac{2(\alpha+1)}{(\alpha-\beta)}$-competitive, where $\beta = \frac{(\alpha-1)}{(\alpha+1)^\gamma}$ and $\gamma = \frac{1}{(\alpha-1)}$, for the objective of minimizing flow time plus energy in bounded speed model, using the traditional power function. SRPT-AJC is using SRPT for job selection and AJC for speed scaling. Lam et al. (2008) gave a non-clairvoyant scheduling algorithm RR-AJC, which is 2-competitive for the objective of minimizing flow time plus energy in bounded speed model using traditional power function. RR-AJC has a big constraint that all the jobs are required to be released at time $t = 0$, this make it non-pure online job scheduling. Lam et al. (2009) proposed a job scheduling model where processor can be in either of three states: working state, idle state and sleep state. The transaction of states depends on inactive flow, wake-up energy and idling energy. A processor transits from idle to working

Table 1 Notations used in EtRR.

Notations	Meaning
t	The current time
j	Any job
$r(j)$ or r_j	Release time/arrival time of a job j
C_j	Completion time of a job j
$p(j)$	Processing requirement (size) of a job j
P	Power of a processor at speed s
$s(t)$ or s	Speed of a processor at time t
α	A constant, commonly believed that its value is 2 or 3
τ	A constant, its value depends on the value of α
I	A set of jobs
S	A schedule of set of jobs in I
$p(j, t)$	The remaining work of a job j at time t
$pdw_a(j, t)$	The pending work of a job j in EtRR at time t
$pdw_o(j, t)$	The pending work of a job j in Opt at time t
$F(j)$	The flow time of a job j
F	The total weighted flow
$e_{xj}(t)$ or $e_x(j)$	Executed time of a job j up to time t
$we_j(t)$ or $we(j)$	Weight of a job j at time t
$we_a(t)$ or we_a	Weight of active jobs at time t
we_l	Total weight of active jobs in L at time t
$n_a(t)$ or n_a	The number of active jobs at time t
$s_a(t)$ or s_a	The speed of a processor for EtRR at time t
$s_o(t)$ or s_o	The speed of a processor for Opt at time t
$e(t)$	The total weight of all active jobs n_a at time t
E	The energy consumed by processor
G	Total weighted flow plus energy
c	The competitiveness
T	The maximum speed of Opt
μ	A fixed constant, its value is $\left(\frac{514}{512}\right)$
σ	A constant ($0 < \sigma < 1$), its value depends on the value of τ
$G_a(t)$ or G_a	The weighted flow time plus energy acquired till time t by the EtRR
$G_o(t)$ or G_o	The weighted flow time plus energy acquired till time t by the Opt
$\frac{dG_a}{dt}$	The rate of change of G_a due to EtRR
$\frac{dG_o}{dt}$	The rate of change of G_o due to Opt
γ	A constant (>0)
L	A set of lagging jobs in EtRR
c_i	The coefficient of a job j_i at time t
x_i	The difference of pending work of a job j_i in EtRR and Opt at time t
δ	A constant depends on α, its value is $\left(\frac{1}{2\alpha}\right)$
l	Number of lagging jobs at time t
Φ	Potential value
$\frac{d\Phi_o}{dt}$	The rate of change of Φ due to Opt
$\frac{d\Phi_a}{dt}$	The rate of change of Φ due to EtRR
S_z^j	The full size of a job j
S_{oz}^j	The units of work of a job j processed by Opt till time t
S_{az}^j	The units of work of a job j processed by EtRR till time t

state when inactive flow equals to idling energy; from idle to sleep state if idling energy exceeded the wake-up energy; working to idle state, if there is no active job; sleep to working state, if inactive flow equals to wake-up energy. Using this model Lam et al. (2009) showed that IdleLonger (SLS) an online non-clairvoyant scheduling is $(3 + (4\alpha^3 + \alpha)(1 + (1 + 3/\alpha)^{\alpha}))$-competitive for the objective of flow time plus energy when using traditional power function in unbounded speed model.

Chan et al. (2011a) gave an online clairvoyant algorithm Uniform Penalty and Unit Weight (UPUN) which is 6-competitive for the objective of minimizing flow time plus energy plus penalty. Chan et al. (2011b) proved that the competitive ratio, using an amortized local competitiveness argument, of online non-clairvoyant algorithm Latest Arrival Processor Sharing (LAPS) is 8 for $\alpha = 2$, 13 for $\alpha = 3$ and $(2\alpha^2/\ln\alpha$ for $\alpha > 3$. LAPS was studied using the traditional power function and unbounded speed model where speed of a processor can range $[0, \infty)$ for the objective of minimizing weighted flow time plus energy. Lam et al. (2013) showed that AJC (Active Job Count) algorithm with no sleep management is $\beta(1 + 1/\alpha)$-competitive, where $\beta = \frac{2}{(1-\gamma)}$ and $\gamma = \frac{1-1/\alpha}{(\alpha+1)^{1/(\alpha-1)}}$, for the objective of minimizing flow time plus energy in the bounded speed model; AJC runs the active jobs using SRPT and the speed is $n_a{}^{1/\alpha}$, where n_a is number of active jobs.

Im et al. (2014a) proposed a concept of migration of jobs and gave an online non-clairvoyant algorithm SelfishMigrate, which is $O(\alpha^2)$-competitive using traditional power function for the objective of minimizing total weighted flow plus energy on unrelated machines. In SelfishMigrate, jobs migrate selfishly until they attain equilibrium. A virtual queue is maintained by every machine where new jobs are added at the tail and a modified Weighted Round Robin (WRR) is used to schedule jobs in queue. Their main innovation was a coordination game on virtual utilities (utility means real speed).

Azar et al. (2015) proposed an online non-clairvoyant uniprocessor algorithm NC, wherein all the jobs arrives with uniform density (*i.e.* *weight/size* = 1). $NC_{fractional}$ *and* $NC_{integral}$ are $(2 + 1/(\alpha - 1))$-competitive and $(3 + 1/(\alpha - 1))$-competitive using traditional power function for the objective of minimizing fractional flow time plus energy and integral weighted flow time plus energy, respectively. NC is using unbounded speed model. In NC, all jobs are arriving with uniform density (*i.e. weight/size* = 1), hence density gives indirect information about size.

We propose an online non-clairvoyant job scheduling EtRR for the objective of weighted flow time plus energy using traditional power function in bounded speed model. EtRR is $\left(\mu \cdot \left(1 + \frac{\tau}{3}\right) \cdot \left(1 + \left(1 + \frac{\tau}{3}\right)^{\alpha}\right)\right)$-competitive, where $0 < \tau \leqslant (3\alpha)^{-1}$ and $\mu = \left(\frac{514}{512}\right)$. Generally, in scheduling algorithms the computation of total flow time depends only on summation of flow time, not on weight and the computation of total weighted flow time depends on summation of flow time multiplied by weight, where the weight of every job is provided by the system on arrival of a job, this weight remains fixed for the life time of the job. In EtRR the total flow time is calculated by summation of flow time multiplied by artificially created weight. In EtRR the weights of jobs are not provided by the system at arrival time, rather scheduler generates them using the difference of current time and release time. The weights of jobs do not change with time linearly, rather re-evaluated when any job arrives or completes, therefore weights change discretely. EtRR calculates the total flow time plus energy using the methodology of weighted flow time plus energy. The summary of results is given in Table 2.

Table 2 Summary of results.

Uniprocessor Algorithms	Competitiveness					
	Clairvoyant			Non-Clairvoyant		
	General α	$\alpha = 2$	$\alpha = 3$	General α	$\alpha = 2$	$\alpha = 3$
SRPT-AJC	$\frac{2(\alpha+1)}{\left(\alpha - \frac{(\alpha-1)}{(\alpha+1)^{\frac{1}{(\alpha-1)}}}\right)}$	3.6	4	–	–	–
RR-AJC	–	–	–	–	2	2
IdleLonger(SLS)	–	–	–	$(3 + \beta\gamma)$ where $\beta = 4\alpha^3 + \alpha,\ \gamma = 1 + (\frac{3}{\alpha})^{\alpha}$	249.5	1002
UPUW	6	6	6	–	–	–
LAPS	–	–	–	$\frac{2\alpha^2}{\ln \alpha}$ (for $\alpha > 3$)	8	13
AJC	$\frac{2\left(\frac{1}{\alpha}+1\right)}{\left(1 - \frac{(1-\frac{1}{\alpha})}{(\alpha+1)^{\frac{1}{(\alpha-1)}}}\right)}$	3.6	4		–	–
SelfishMigrate	–	–	–	α^2	4	9
$NC_{fractional}$	–	–	–	$(2 + \frac{1}{\alpha-1})$	3	2.5
$NC_{integral}$	–	–	–	$(3 + \frac{1}{\alpha-1})$	4	3.5
EtRR (this paper)	–	–		$\mu\rho(1 + \rho^{\alpha})$ where $\rho = (1 + \frac{\tau}{3})$ and $\tau = \frac{1}{3\alpha}$	2.24	2.20

4. An $O(1)$-competitive algorithm

This section contains a non-clairvoyant online algorithm Executed-time Round Robin (EtRR), where the weights of jobs are created artificially using executed time of jobs by the scheduler. The motive behind creating artificial weights is that the process which is executing for a longer duration may be bigger in size and desires more share of the processor's speed to get fully executed in a smaller amount of time and reduces the total flow time. In clairvoyant setting the job's size is known and processor speed decreases with job's size, whereas in non-clairvoyant the job's size is not known until completion of execution of job and job starts with less speed which increases with an increase in execution to create the reverse effect of clairvoyant. EtRR scheduling is $O(1)$-competitive for the objective of minimizing weighted flow time plus energy (F + E) when using a processor with the maximum speed $\left(1 + \frac{\tau}{3}\right)T$, where $\tau = 1/3\alpha$). At any time t, the weight assigned to a newly arrived job is 1. The values of created weights do not change linearly with time, rather the weights ($\forall j$, $we_j(t)$) of all jobs are re-evaluated/recalculated discretely using $we(j) = (1 + e_x(j)) = (1 + (t - r(j)))$ (only when a job arrives or finishes).

4.1. Algorithm EtRR

At any time t, the processor speed is set to $s_a(t) = \left(1 + \frac{\tau}{3}\right) \cdot \min((we_a(t))^{1/\alpha}, T)$, i.e. $s_a(t) = \left(1 + \frac{\tau}{3}\right) \cdot \min\left(\left[\sum_{i=1}^{n_a}(1 + e_{xi})\right]^{1/\alpha}, T\right)$, where $e_{xi} = (t - r_i)$ and $we_a(t) = \sum_{i=1}^{n_a}(1 + e_{xi})$ are the executed time of job i and total weight of active jobs n_a (total executed time of all active jobs). The processor executes all active jobs such that every active job i shares processor's speed equal to $s(t) \cdot \left(\frac{we(i)}{we_a(t)}\right)$, i.e. $s(t) \cdot \left[\frac{(1+e_{xi})}{\sum_{i=1}^{n_a}(1+e_{xi})}\right]$. It is considered that the weights of jobs and speed of a processor will be re-evaluated, when there is a change in count of active jobs n_a (i.e., either on arrival or on completion of a job). EtRR is compared against an optimal offline algorithm Opt, which uses maximum processor speed T.

Theorem 1. *When* $0 < \tau \leqslant (3\alpha)^{-1}, \mu = \left(\frac{514}{512}\right)$ *and* $\alpha > 1$, *using a uniprocessor with maximum speed* $\left(1 + \frac{\tau}{3}\right)T$, *EtRR is c-competitive for weighted flow plus energy, where* $c = \mu \cdot \left(1 + \frac{\tau}{3}\right) \cdot \left(1 + \left(1 + \frac{\tau}{3}\right)^{\alpha}\right) = O(1)$.

The remaining part of this section is committed to prove Theorem 1. We will drop the parameter t as it is clear that t is current time only. To prove that EtRR is c-competitive, it will be sufficient to provide a potential function $\Phi(t)$ which satisfied the following three conditions (Chan et al., 2011a).

(a) *Boundary Condition:* At the beginning before any job is released and at the end after all jobs are completed $\Phi = 0$.
(b) *Job Arrival and Completion Condition:* The value of Φ does not increase when a job arrives or completes.
(c) *Running Condition:* At any other time when no job arrives or completes $\frac{dG_a(t)}{dt} + \gamma \cdot \frac{d\Phi}{dt} \leqslant c \cdot \frac{dG_o(t)}{dt}$, where $\gamma > 0$.

4.2. Potential function $\Phi(t)$

Let $pdw_a(j, t)$ and $pdw_o(j, t)$ be pending work of j in EtRR and Opt, respectively, at any time t and for any job j. At any time t, an active job j is considered as a lagging job if EtRR has processed less than Opt on j up to time t, i.e., $pdw_a(j, t) - pdw_o(j, t) > 0$ (the difference can be calculated using *Lemma* 2). Let $L = \{j_1, j_2, \ldots, j_l\}$ be a set of lagging jobs in EtRR and they are arranged in ascending sequence of latest time when the job is converted into lagging job. $(\forall j_i \in L)\exists x_i = pdw_a(j_i, t) - pdw_o(j_i, t) > 0$. Our potential function $\Phi(t)$ for weighted flow plus energy is as follows:

$$\phi(t) = \Sigma_{i=1}^{l} c_i \cdot x_i \qquad (1)$$

Where $\quad x_i = \max\{0, (pdw_a(j_i, t) - pdw_o(j_i, t))\}$ (2)

$$\& \qquad c_i = \begin{cases} (we_i)^{1-1/\alpha}, & \text{if } we_i \leqslant T^\alpha, \quad \text{where } we_i = \sum_{k=1}^{i}(1 + e_{xk}) \text{ and } e_{xk} = (t - r_k) \\ \left(\frac{we_i}{T-\delta T}\right), & \text{otherwise}, \quad \text{where } \delta = \frac{1}{2\alpha} \end{cases} \tag{3}$$

Note: c_i is called the coefficient of j_i and monotonically increases with i.

There is no active job before any job is released and after all jobs are completed, therefore in both the cases the value of $\Phi = 0$, hence the boundary condition follows. At some time t, on arrival of any job j_i in I, x_i tends to zero as its executed time is zero, therefore $[pdw_a(j_i, t) - pdw_o(j_i, t) = 0]$. On arrival of any job j_i, it will be added at the end of I, therefore the coefficient of all other jobs remains changed hence Φ does not change. On completion of a job j_i (it leaves I), x_i will be zero and coefficient of any other lagging job will either remain the same or decrease, therefore Φ does not increase, hence the arrival or completion condition follows. The only condition left to check is the running condition at time t, when no job arrives or completes, i.e., Φ does not have discrete changes. Let $we_l = \sum_{i=1}^{l} we(j_i) = \sum_{i=1}^{l}(1 + e_{xi})$ be the weight of all jobs in L. Since number of lagging job $l \leqslant n_a, we_l \leqslant we_a$. As per the previous discussion, $\frac{dG_a}{dt} = we_a + s_a^\alpha$ and $\frac{dG_o}{dt} = we_o + s_o^\alpha$. Bounding the rate of change of Φ by observing how Φ changes, first due to Opt only (*Lemma 3*) and then due to EtRR (*Lemma 4*). Total rate of change of Φ due to Opt and EtRR is $\frac{d\Phi}{dt} = \frac{d\Phi_o}{dt} + \frac{d\Phi_a}{dt}$.

Lemma 1 (*Young's Inequality (Steele, 2004)*). *For some positive real numbers a, b, x and y, if* $\frac{1}{a} + \frac{1}{b} = 1$ *holds, then*

$$x \cdot y \leqslant \frac{1}{a} \cdot x^a + \frac{1}{b} \cdot y^b \tag{4}$$

Lemma 2. *The difference of pending work of any job j in EtRR and Opt can be calculated in non-clairvoyant scheduling setting (without knowing the actual full size of job j).*

Proof. Let a job j of size S_z^j (which is not known in non-clairvoyant scheduling setting) is released at arrival time r_j. After t unit of time, S_{az}^j and S_{oz}^j units of work of a job j are processed by EtRR and Opt, respectively. Then pending work can be calculated as,

$$pdw_a(j, t) = \left(S_z^j - S_{az}^j\right)$$

$$pdw_o(j, t) = \left(S_z^j - S_{oz}^j\right)$$

The difference of pending work can be calculated as,

$$pdw_a(j, t) - pdw_o(j, t) = \left(S_z^j - S_{az}^j\right) - \left(S_z^j - S_{oz}^j\right)$$

$$\Rightarrow pdw_a(j, t) - pdw_o(j, t) = \left(S_{oz}^j - S_{az}^j\right)$$

The difference of pending work of a job j in EtRR and Opt is equal to difference of units of work of a job j processed by Opt and EtRR. Hence, Lemma follows.

Lemma 3.

(a) If $we_l \leqslant T^\alpha$ holds, then $\frac{d\Phi_o}{dt} \leqslant \frac{s_o^\alpha}{\alpha} + \left(1 - \frac{1}{\alpha}\right) \cdot we_l$;
(b) If $we_l > T^\alpha$ holds, then $\frac{d\Phi_o}{dt} \leqslant \left(\frac{we_l}{1-\delta}\right)$, where $\delta = (1/2\alpha)$.

Proof. To calculate the upper bound of $\frac{d\Phi_o}{dt}$, it is required to observe the worst case in which Opt is executing the job j_l with the biggest coefficient c_l. At this time, the rate of change of x_i will be s_o (only due to Opt), therefore $\frac{d\Phi_o}{dt} \leqslant s_o \cdot c_l$.

(a) When $we_l \leqslant T^\alpha, c_l = we_l^{1-1/\alpha}$, and thus $\frac{d\Phi_o}{dt} \leqslant s_o \cdot we_l^{1-1/\alpha}$. On applying Young's Inequality (*Lemma 1*), where $a = \alpha, b = \alpha/(\alpha - 1), x = s_o, y = we_l^{1-1/\alpha}$. Using Eq. (4) we have

$$\frac{d\Phi_o}{dt} \leqslant \frac{s_o^\alpha}{\alpha} + \left(1 - \frac{1}{\alpha}\right) \cdot we_l \tag{5}$$

(b) When $we_l > T^\alpha,\ c_l = \frac{we_l}{(T-\delta T)} = \frac{we_l}{(1-\delta)T}$.

Since $s_o \leqslant T$, $\frac{d\Phi_o}{dt} \leqslant s_o \cdot c_l \leqslant T \cdot c_l \leqslant \frac{we_l}{(1-\delta)}$

$$\Rightarrow \frac{d\Phi_o}{dt} \leqslant \frac{we_l}{(1 - \delta)} \tag{6}$$

Lemma 4.

(a) If $we_l \leqslant T^\alpha$ holds, then $\frac{d\Phi_a}{dt} \leqslant -\frac{s_a}{(2-1/\alpha)} \cdot \left(\frac{we_l^{2-1/\alpha}}{we_a}\right)$;

(b) If $we_l > T^\alpha$ holds, then $\frac{d\Phi_a}{dt} \leqslant -\frac{\left(1+\frac{\tau}{3}\right) \cdot we_l^2}{(2-1/\alpha) \cdot we_a}$.

Proof. To compute the upper bound of $\frac{\mathrm{d}\Phi_\mathrm{a}}{\mathrm{dt}}$, it is required to observe that every lagging job j_i is processed at the rate of $s_a \cdot \left(\frac{we(j_i)}{we_a}\right)$ (due to only EtRR), and consequently x_i is varying at the rate of $-s_a \cdot \left(\frac{we(j_i)}{we_a}\right)$. To make the discussion uncomplicated, let $f_i = \sum_{k=1}^{i} we(j_k)$, thus $f_0 = 0, f_l = we_l$ and for any $1 \leqslant i \leqslant l,\ f_i - f_{i-1} = we(j_i)$.

(a) For every $j_i \in L,\ c_i = f_i^{1-1/\alpha}$. If $we_l \leqslant T^\alpha$, then using Eq. (1)

$$\begin{aligned}
\frac{d\Phi_a}{dt} &= \sum_{i=1}^{l} c_i \cdot x_i \\
\frac{d\Phi_a}{dt} &= \sum_{i=1}^{l} f_i^{1/\alpha} \cdot \left(-s_a \cdot \frac{we(j_i)}{we_a}\right) \\
\frac{d\Phi_a}{dt} &= -\frac{s_a}{we_a} \sum_{i=1}^{l} f_i^{1-1/\alpha} \cdot we(j_i) \\
\frac{d\Phi_a}{dt} &= -\frac{s_a}{we_a} \sum_{i=1}^{l} f_i^{1-1/\alpha} \cdot (f_i - f_{i-1}) \\
&\leqslant -\frac{s_a}{we_a} \sum_{i=1}^{l} \int_{f_{i-1}}^{f_i} h^{1-1/\alpha} dh \quad \text{(Since } h^{1-1/\alpha} \text{ is monotonically increasing)} \\
&\leqslant -\frac{s_a}{we_a} \int_0^{f_l} h^{1-1/\alpha} dh \\
&= -\frac{s_a}{we_a} \cdot \frac{f_l^{2-1/\alpha}}{(2-1/\alpha)} \\
&= -\frac{we_l^{2-1/\alpha}}{(2-1/\alpha)} \frac{s_a}{we_a} \\
\Rightarrow \frac{d\Phi_a}{dt} &\leqslant -\frac{s_a}{(2-1/\alpha)} \left(\frac{we_l^{2-1/\alpha}}{we_a}\right)
\end{aligned} \tag{7}$$

(b) If $we_l > T^\alpha$, in this situation $we_a \geqslant we_l > T^\alpha$ and $f_l = we_l > T^\alpha$, therefore

$$s_a = \left(1+\frac{\tau}{3}\right)\cdot \min((we_a(t))^{1/\alpha}, T) = \left(1+\frac{\tau}{3}\right)\cdot T \quad (8)$$

Let g be a biggest integer so that $z_g \leqslant T^\alpha$, then using Eq. (1)

$$\frac{d\Phi_a}{dt} = \sum_{i=1}^{l} c_i\cdot x_i = \sum_{i=1}^{l} c_i\cdot\left(-s_a\cdot\frac{we(j_i)}{we_a}\right)$$
$$= -\sum_{i=1}^{l} c_i\cdot we(j_i)\cdot\left(\frac{s_a}{we_a}\right)$$
$$= -\sum_{i=1}^{l} c_i\cdot we(j_i)\cdot\left(\frac{(1+\frac{\tau}{3})\cdot T}{we_a}\right) \quad \text{(by using equation (8))}$$

The set of l lagging jobs is divided into two sets. First set of g jobs are following $f_g \leqslant T^\alpha$ and the rest of $(l-g)$ jobs in second set are following $f > T^\alpha$ then

$$= -\left(\sum_{i=1}^{g} we(j_i)\cdot f_i^{1-1/\alpha} + \sum_{i=g+1}^{l}\frac{1}{(1-1/2\alpha)}\cdot\frac{we(j_i)}{T}\cdot f_i\right)\cdot\left(\frac{(1+\frac{\tau}{3})\cdot T}{we_a}\right)$$

$$= -\left(\sum_{i=1}^{g} f_i^{1-1/\alpha}\cdot(f_i - f_{i-1}) + \frac{1}{(1-1/2\alpha)\cdot T}\cdot\sum_{i=g+1}^{l} f_i\cdot(f_i - f_{i-1})\right)\cdot\left(\frac{(1+\frac{\tau}{3})\cdot T}{we_a}\right)$$

$$\leqslant -\left(\int_0^{f_g} h^{1-1/\alpha}dh + \frac{1}{(1-1/2\alpha)\cdot T}\cdot\int_{f_g}^{f_l} h\,dh\right)\cdot\left(\frac{(1+\frac{\tau}{3})\cdot T}{we_a}\right)$$

$$= -\left(\frac{f_g^{2-1/\alpha}}{(2-1/\alpha)} + \frac{1}{(1-1/2\alpha)\cdot T}\cdot\left(\frac{f_l^2 - f_g^2}{2}\right)\right)\cdot\left(\frac{(1+\frac{\tau}{3})\cdot T}{we_a}\right)$$

$$= -\frac{1}{(2-1/\alpha)}\cdot\left(\frac{f_g^2}{f_g^{1/\alpha}} + \frac{f_l^2 - f_g^2}{T}\right)\cdot\left(\frac{(1+\frac{\tau}{3})\cdot T}{we_a}\right)$$

(since $f_g^{1/\alpha} \leqslant T$)

$$\leqslant -\frac{1}{(2-1/\alpha)}\cdot\left(\frac{f_g^2}{T} + \frac{f_l^2 - f_g^2}{T}\right)\cdot\left(\frac{(1+\frac{\tau}{3})\cdot T}{we_a}\right)$$

$$= -\frac{(1+\frac{\tau}{3})\cdot f_l^2}{(2-1/\alpha)\cdot we_a}$$

$$= -\frac{(1+\frac{\tau}{3})\cdot we_l^2}{(2-1/\alpha)\cdot we_a}$$

$$\Rightarrow \frac{d\Phi_a}{dt} \leqslant -\frac{(1+\frac{\tau}{3})\cdot we_l^2}{(2-1/\alpha)\cdot we_a} \quad (9)$$

Lemma 5. *By assuming* $\gamma = \left(\frac{\alpha-1}{512\alpha}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)$, *at any time when* Φ *does not have discrete changes* $\frac{dG_a}{dt} + \gamma\cdot\frac{d\Phi}{dt} \leqslant c\cdot\frac{dG_o}{dt}$, *where* $c = \mu\cdot\left(1+\frac{\tau}{3}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)$ *and* $\mu = \left(\frac{514}{512}\right)$.

Proof. We have divided the analysis into three possibilities depending on whether $we_a > T^\alpha$ and $we_l > T^\alpha$. The possibilities are further divided on the basis of whether $we_l > (1-\sigma)\cdot we_a \Rightarrow we_l > \left(\frac{\tau}{3+\tau}\right)\cdot we_a \Rightarrow we_l > \left(\frac{\tau}{3+\tau}\right)\cdot we_a \Rightarrow we_l > \left(\frac{1}{1+9\alpha}\right)\cdot we_a$, where $\sigma = \frac{3}{(3+\tau)}$, $0 < \sigma < 1$ and $0 < \tau \leqslant (3\alpha)^{-1}$. In view of this fact that any non-lagging active job in EtRR must also be active in Opt, therefore

$$we_o \geqslant (we_a - we_l) \geqslant [we_a - (1-\sigma)\cdot we_a] \geqslant \sigma\cdot we_a$$
$$= \frac{3}{(3+\tau)}\cdot we_a \Rightarrow we_a \leqslant \left(1+\frac{\tau}{3}\right)\cdot we_o \quad (10)$$

$$\gamma = \left(\frac{\alpha-1}{512\alpha}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right) \quad (11)$$

$$c = \mu\cdot\left(1+\frac{\tau}{3}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right) \quad (12)$$

$$\sigma = \frac{3}{(3+\tau)} \quad (13)$$

Case 1: $we_l \leqslant we_a \leqslant T^\alpha$, where

$$s_a = \left(1+\frac{\tau}{3}\right)\cdot\min((we_a)^{1/\alpha}, T) = \left(1+\frac{\tau}{3}\right)\cdot(we_a)^{1/\alpha} \quad (14)$$

(**a**) If $we_l > \left(\frac{1}{1+9\alpha}\right)\cdot we_a \quad (15)$

Then using *Lemma* 3 and *Lemma* 4,

$$\left(\frac{dG_a}{dt} + \gamma\cdot\frac{d\Phi}{dt}\right) = \frac{dG_a}{dt} + \gamma\cdot\left(\frac{d\Phi_o}{dt} + \frac{d\Phi_a}{dt}\right)$$
$$\left(\frac{dG_a}{dt} + \gamma\cdot\frac{d\Phi}{dt}\right) \leqslant (we_a + s_a^\alpha) + \gamma\cdot\left[\frac{s_o^\alpha}{\alpha} + \left(1-\frac{1}{\alpha}\right)\cdot we_l - \frac{s_a}{(2-1/\alpha)}\cdot\left(\frac{we_l^{2-1/\alpha}}{we_a}\right)\right] \quad \text{(by using equations (5) and (7))}$$
$$\leqslant we_a + \left(1+\tfrac{\tau}{3}\right)^\alpha\cdot we_a + \tfrac{\gamma}{\alpha}\cdot s_o^\alpha + \gamma\cdot\left(1-\tfrac{1}{\alpha}\right)\cdot we_l - \frac{(1+\frac{\tau}{3})\cdot we_a^{1/\alpha}}{(2-1/\alpha)}\cdot\frac{\gamma\cdot we_l^{2-1/\alpha}}{we_a} \quad \text{(by using equation (14))}$$
$$\leqslant \left(1+\left(1+\tfrac{\tau}{3}\right)^\alpha\right)\cdot we_a + \tfrac{\gamma}{\alpha}\cdot s_o^\alpha + \gamma\cdot we_a - \frac{we_a^{1/\alpha}}{(2-1/\alpha)}\cdot\frac{\gamma\cdot\left[\left(\frac{1}{1+9\alpha}\right)\cdot we_a\right]^{2-1/\alpha}}{we_a} \quad \text{(by using equation (15))} \quad (16)$$
$$= \tfrac{\gamma}{\alpha}\cdot s_o^\alpha + \left(1+\left(1+\tfrac{\tau}{3}\right)^\alpha\right)\cdot we_a + \gamma\cdot we_a - \frac{\gamma\cdot we_a\cdot\left(\frac{1}{1+9\alpha}\right)^{2-1/\alpha}}{(2-1/\alpha)}$$
$$= \tfrac{\gamma}{\alpha}\cdot s_o^\alpha + \left[\left(1+\left(1+\tfrac{\tau}{3}\right)^\alpha\right) + \gamma - \frac{\gamma\cdot\left(\frac{1}{1+9\alpha}\right)^{2-1/\alpha}}{(2-1/\alpha)}\right]\cdot we_a$$

$$\begin{cases} \because a > 1 \Rightarrow \frac{1}{1+9\alpha} < 1 \Rightarrow \left(\frac{1}{1+9\alpha}\right)^2 < 1 \Rightarrow \left(1 - \frac{1}{2}\cdot\left(\frac{1}{1+9\alpha}\right)^2\right) = \frac{162\alpha^2+36\alpha+1}{162\alpha^2+36\alpha+2} < 1 & (17) \\ \because \frac{1}{(1-1/2\alpha)} > 1 \Rightarrow \frac{1}{(2-1/\alpha)} > \frac{1}{2} & (18) \\ \because \left(\frac{1}{1+9\alpha}\right)^{2-1/\alpha} > \left(\frac{1}{1+9\alpha}\right)^2 & (19) \end{cases}$$

Using the results of Eqs. (10), (19) and (18) in (16), we have

$$\left(\frac{dG_a}{dt}+\gamma\cdot\frac{d\Phi}{dt}\right) \leqslant \frac{\gamma}{\alpha}\cdot s_o^\alpha + \left[\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)+\gamma-\frac{1}{2}\cdot\gamma\cdot\left(\frac{1}{1+9\alpha}\right)^2\right] \cdot\left(1+\frac{\tau}{3}\right)\cdot we_o = \frac{\gamma}{\alpha}\cdot s_o^\alpha + \left[\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)+\gamma\cdot\left(1-\frac{1}{2}\cdot\left(\frac{1}{1+9\alpha}\right)^2\right)\right] \cdot\left(1+\frac{\tau}{3}\right)\cdot we_o \quad (20)$$

Using the result of Eq. (17) in (20), we have

$$\begin{aligned}
\left(\frac{dG_a}{dt}+\gamma\cdot\frac{d\Phi}{dt}\right) &\leqslant \frac{\gamma}{\alpha}\cdot s_o^\alpha+\left[\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)+\gamma\right]\cdot\left(1+\frac{\tau}{3}\right)\cdot we_o \\
&= \frac{\gamma}{\alpha}\cdot s_o^\alpha+\left[\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right) + \left(\frac{\alpha-1}{512\alpha}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)\right]\cdot\left(1+\frac{\tau}{3}\right)\cdot we_o \\
&\quad \text{(by using equation (11))} \\
&= \frac{\gamma}{\alpha}\cdot s_o^\alpha+\left[\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right) + \frac{1}{512}\cdot\left(1-\frac{1}{\alpha}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)\right]\cdot\left(1+\frac{\tau}{3}\right)\cdot we_o \\
&\leqslant \frac{\gamma}{\alpha}\cdot s_o^\alpha+we_o\cdot\left[\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right) + \frac{1}{512}\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)\right]\cdot\left(1+\frac{\tau}{3}\right) \quad (21) \\
&= \frac{\gamma}{\alpha}\cdot s_o^\alpha+we_o\cdot\left[\frac{513}{512}\cdot\left(1+\frac{\tau}{3}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)\right] \\
&\leqslant \frac{\gamma}{\alpha}\cdot s_o^\alpha+we_o\cdot\left(\mu\cdot\left(1+\frac{\tau}{3}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)\right) \\
&\quad \text{(by using equation (12))} \\
&= \frac{\gamma}{\alpha}\cdot s_o^\alpha+we_o\cdot c
\end{aligned}$$

$$\left(\frac{dG_a}{dt}+\gamma\cdot\frac{d\Phi}{dt}\right) \leqslant \frac{\gamma}{\alpha}\cdot s_o^\alpha+we_o\cdot c$$

To calculate the value of γ/α for Eq. (21) we are using Eq. (11) as follows:

$$\begin{aligned}
\frac{\gamma}{\alpha} &= \frac{1}{\alpha}\cdot\left(\frac{\alpha-1}{512\alpha}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right) \leqslant \frac{1}{512}\cdot\left(1-\frac{1}{\alpha}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right) \\
&\leqslant \frac{1}{512}\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right) \leqslant \frac{1}{512}\cdot\left(1+\frac{\tau}{3}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right) \\
&\leqslant \frac{514}{512}\cdot\left(1+\frac{\tau}{3}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right) = \mu\cdot\left(1+\frac{\tau}{3}\right) \\
&\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right) = c \Rightarrow \frac{\gamma}{\alpha} \leqslant c
\end{aligned} \quad (22)$$

Using the results of Eq. (22) in (21), we have

$$\Rightarrow \left(\frac{dG_a}{dt}+\gamma\cdot\frac{d\Phi}{dt}\right) \leqslant c\cdot s_o^\alpha + c\cdot we_o \leqslant c\cdot\frac{dG_o}{dt}$$

$$\Rightarrow \left(\frac{dG_a}{dt}+\gamma\cdot\frac{d\Phi}{dt}\right) \leqslant c\cdot\frac{dG_o}{dt}$$

(**b**) If $we_l \leqslant \left(\frac{1}{1+9\alpha}\right)\cdot we_a \quad (23)$

Then in this case, we are adopting the result of *Lemma* 4 as $\frac{d\Phi_a}{dt} \leqslant 0$ (due to negative value).

$$\left(\frac{dG_a}{dt}+\gamma\cdot\frac{d\Phi}{dt}\right) = \frac{dG_a}{dt}+\gamma\cdot\left(\frac{d\Phi_o}{dt}+\frac{d\Phi_a}{dt}\right)$$

$$\left(\frac{dG_a}{dt}+\gamma\cdot\frac{d\Phi}{dt}\right) \leqslant \frac{dG_a}{dt}+\gamma\cdot\left(\frac{d\Phi_o}{dt}\right)$$

Using *Lemma* 3 we have

$$\begin{aligned}
\left(\frac{dG_a}{dt}+\gamma\cdot\frac{d\Phi}{dt}\right) &\leqslant (we_a+s_a^\alpha)+\gamma\cdot\left[\frac{s_o^\alpha}{\alpha}+\left(1-\frac{1}{\alpha}\right)\cdot we_l\right] \\
&\quad \text{(by using equation (5))} \\
&\leqslant we_a+\left(1+\frac{\tau}{3}\right)^\alpha\cdot we_a+\frac{\gamma}{\alpha}\cdot s_o^\alpha+\gamma\cdot\left(1-\frac{1}{\alpha}\right)\cdot we_l \\
&\quad \text{(by using equation (14))} \\
&\leqslant \left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)\cdot we_a+\frac{\gamma}{\alpha}\cdot s_o^\alpha+\gamma\cdot we_a\left(1-\frac{1}{\alpha}\right) \\
&\leqslant \left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)\cdot we_a+\frac{\gamma}{\alpha}\cdot s_o^\alpha+\gamma\cdot we_a \\
&= \frac{\gamma}{\alpha}\cdot s_o^\alpha+\left(\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)+\gamma\right)\cdot we_a \\
&= \frac{\gamma}{\alpha}\cdot s_o^\alpha+\left(\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right) + \left(\frac{\alpha-1}{512\alpha}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)\right)\cdot we_a \\
&\quad \text{(by using equation (11))} \\
&= \frac{\gamma}{\alpha}\cdot s_o^\alpha+\left(\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right) + \frac{1}{512}\cdot(1-1/\alpha)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)\right)\cdot we_a \\
&\leqslant \frac{\gamma}{\alpha}\cdot s_o^\alpha+\left(\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right) + \frac{1}{512}\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)\right)\cdot\left(1+\frac{\tau}{3}\right)\cdot we_o \\
&\quad \text{(by using equation (10))} \\
&= \frac{\gamma}{\alpha}\cdot s_o^\alpha+we_o\cdot\left(\frac{513}{512}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)\cdot\left(1+\frac{\tau}{3}\right) \\
&\leqslant \frac{\gamma}{\alpha}\cdot s_o^\alpha+we_o\cdot\left(\mu\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)\cdot\left(1+\frac{\tau}{3}\right)\right) \\
&= \frac{\gamma}{\alpha}\cdot s_o^\alpha+c\cdot we_o \quad \text{(by using equation (12))} \\
&\leqslant c\cdot s_o^\alpha+c\cdot we_o \quad \text{(by using equation (22))} \\
&= c\cdot(s_o^\alpha+we_o) = c\cdot\frac{dG_o}{dt} \Rightarrow \left(\frac{dG_a}{dt}+\gamma\cdot\frac{d\Phi}{dt}\right) \leqslant c\cdot\frac{dG_o}{dt}
\end{aligned}$$

Case 2: $we_a > T^\alpha$ and $we_l \leqslant T^\alpha$, where

$$s_a = \left(1+\frac{\tau}{3}\right)\cdot\min((we_a(t))^{1/\alpha}, T) = \left(1+\frac{\tau}{3}\right)\cdot T \quad (24)$$

(**a**) If $we_l > \left(\frac{1}{1+9\alpha}\right)\cdot we_a \quad (25)$

Then using previous *Lemma* 3 and *Lemma* 4,

$$\left(\frac{dG_a}{dt}+\gamma\cdot\frac{d\Phi}{dt}\right)\leqslant(we_a+s_a^\alpha)+\gamma\cdot\left[\frac{s_o^\alpha}{\alpha}+\left(1-\frac{1}{\alpha}\right)\cdot we_l-\frac{s_a}{(2-1/\alpha)}\cdot\left(\frac{we_l^{2-1/\alpha}}{we_a}\right)\right]$$

(by using equations (5) *and* (7))

$$\leqslant\left(we_a+\left(1+\tfrac{\tau}{3}\right)^\alpha\cdot T^\alpha\right)+\gamma\cdot\left[\frac{s_o^\alpha}{\alpha}+we_l-\frac{\left(1+\frac{\tau}{3}\right)\cdot T}{(2-1/\alpha)}\cdot\frac{we_l^{2-1/\alpha}}{we_a}\right]$$

(by using equation (24))

$$\leqslant\left(we_a+we_a\cdot\left(1+\tfrac{\tau}{3}\right)^\alpha\right)+\gamma\cdot\left[\frac{s_o^\alpha}{\alpha}+we_l-\frac{\left(1+\frac{\tau}{3}\right)\cdot we_l^{1/\alpha}}{(2-1/\alpha)}\cdot\frac{we_l^{2-1/\alpha}}{we_a}\right]$$

$$=\left(we_a+we_a\cdot\left(1+\tfrac{\tau}{3}\right)^\alpha\right)+\gamma\cdot\left[\frac{s_o^\alpha}{\alpha}+we_l-\frac{\left(1+\frac{\tau}{3}\right)}{(2-1/\alpha)}\cdot\frac{we_l^2}{we_a}\right] \quad (26)$$

$$\leqslant we_a\cdot\left(1+\left(1+\tfrac{\tau}{3}\right)^\alpha\right)+\gamma\cdot\left[\frac{s_o^\alpha}{\alpha}+we_a-\frac{\left(1+\frac{\tau}{3}\right)}{(2-1/\alpha)}\cdot\frac{\left(\frac{1}{1+9\alpha}\right)^2\cdot we_a^2}{we_a}\right]$$

(by using equation (25))

$$=\frac{\gamma}{\alpha}\cdot s_o^\alpha+we_a\cdot\left[\left(1+\left(1+\tfrac{\tau}{3}\right)^\alpha\right)+\gamma-\frac{\gamma\left(\frac{1}{1+9\alpha}\right)^2}{(2-1/\alpha)}\cdot\left(1+\tfrac{\tau}{3}\right)\right]$$

$$\leqslant\frac{\gamma}{\alpha}\cdot s_o^\alpha+\left[\left(1+\left(1+\tfrac{\tau}{3}\right)^\alpha\right)+\gamma-\frac{\gamma\left(\frac{1}{1+9\alpha}\right)^{2-1/\alpha}}{(2-1/\alpha)}\right]\cdot \mathrm{we}_a$$

Using the result of Eqs. (10), (19) and (18) in (26), we have

$$\left(\frac{dG_a}{dt}+\gamma\cdot\frac{d\Phi}{dt}\right)\leqslant\frac{\gamma}{\alpha}\cdot s_o^\alpha+\left[\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)+\gamma-\frac{1}{2}\cdot\gamma\cdot\left(\frac{1}{1+9\alpha}\right)^2\right]$$
$$\cdot\left(1+\frac{\tau}{3}\right)\cdot we_o=\frac{\gamma}{\alpha}\cdot s_o^\alpha+\left[\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)+\gamma\cdot\left(1-\frac{1}{2}\cdot\left(\frac{1}{1+9\alpha}\right)^2\right)\right]$$
$$\cdot\left(1+\frac{\tau}{3}\right)\cdot we_o \quad (27)$$

Using the result of Eq. (17) in (27), we have

$$\left(\frac{dG_a}{dt}+\gamma\cdot\frac{d\Phi}{dt}\right)\leqslant\frac{\gamma}{\alpha}\cdot s_o^\alpha+\left[\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)+\gamma\right]\cdot\left(1+\frac{\tau}{3}\right)\cdot we_o$$
$$=\frac{\gamma}{\alpha}\cdot s_o^\alpha+\left[\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)+\left(\frac{\alpha-1}{512\alpha}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)\right]$$
$$\cdot\left(1+\frac{\tau}{3}\right)\cdot we_o \quad \text{(by using equation (11))}$$
$$=\frac{\gamma}{\alpha}\cdot s_o^\alpha+\left[\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)+\frac{1}{512}\cdot\left(1-\frac{1}{\alpha}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)\right]$$
$$\cdot\left(1+\frac{\tau}{3}\right)\cdot we_o$$
$$\leqslant\frac{\gamma}{\alpha}\cdot s_o^\alpha+we_o\cdot\left[\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)+\frac{1}{512}\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)\right]$$
$$\cdot\left(1+\frac{\tau}{3}\right)$$
$$=\frac{\gamma}{\alpha}\cdot s_o^\alpha+we_o\cdot\left[\frac{513}{512}\cdot\left(1+\frac{\tau}{3}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)\right]$$
$$\leqslant\frac{\gamma}{\alpha}\cdot s_o^\alpha+we_o\cdot\left[\mu\cdot\left(1+\frac{\tau}{3}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)\right]$$
$$\leqslant\frac{\gamma}{\alpha}\cdot s_o^\alpha+we_o\cdot c \quad \text{(by using equation (12))}$$
$$\leqslant c\cdot s_o^\alpha+we_o\cdot c \quad \text{(by using equation (22))}$$
$$=c\cdot(s_o^\alpha+we_o)\Rightarrow\left(\frac{dG_a}{dt}+\gamma\cdot\frac{d\Phi}{dt}\right);\leqslant c\cdot\frac{dG_o}{dt}$$

(**b**) If $we_l\leqslant\left(\frac{1}{1+9\alpha}\right)\cdot we_a$ (28)

Then in this case, we are adopting the result of *Lemma* 4 as $\frac{d\Phi_a}{dt}\leqslant 0$ (due to negative value). We become familiar with $we_a > T^\alpha$ and $we_a \geqslant we_l$.

Using *Lemma* 3, we have

$$\left(\frac{dG_a}{dt}+\gamma\cdot\frac{d\Phi}{dt}\right)\leqslant(we_a+s_a^\alpha)+\gamma\cdot\left[\frac{s_o^\alpha}{\alpha}+\left(1-\frac{1}{\alpha}\right)\cdot we_l\right]$$

(by using equation (5))

$$=\left(we_a+\left(1+\frac{\tau}{3}\right)^\alpha\cdot T^\alpha\right)+\gamma\cdot\left[\frac{s_o^\alpha}{\alpha}+\left(1-\frac{1}{\alpha}\right)\cdot we_a\right]$$

(by using equation (24))

$$\leqslant\left(we_a+we_a\cdot\left(1+\frac{\tau}{3}\right)^\alpha\right)+\gamma\cdot\left[\frac{s_o^\alpha}{\alpha}+we_a\right]$$
$$=\frac{\gamma}{\alpha}\cdot s_o^\alpha+we_a\cdot\left[\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)+\gamma\right]$$
$$\leqslant\frac{\gamma}{\alpha}\cdot s_o^\alpha+\left[\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)+\gamma\right]\cdot\left(1+\frac{\tau}{3}\right)\cdot we_o$$

(by using equation (10))

$$=\frac{\gamma}{\alpha}\cdot s_o^\alpha+\left[\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)+\left(\frac{\alpha-1}{512\alpha}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)\right]$$
$$\cdot\left(1+\frac{\tau}{3}\right)\cdot we_o \quad \text{(by using equation (11))}$$
$$=\frac{\gamma}{\alpha}\cdot s_o^\alpha+\left[\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)+\frac{1}{512}\cdot(1-1/\alpha)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)\right]$$
$$\cdot\left(1+\frac{\tau}{3}\right)\cdot we_o$$
$$\leqslant\frac{\gamma}{\alpha}\cdot s_o^\alpha+\left[\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)+\frac{1}{512}\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)\right]$$
$$\cdot\left(1+\frac{\tau}{3}\right)\cdot we_o$$
$$=\frac{\gamma}{\alpha}\cdot s_o^\alpha+\frac{513}{512}\cdot\left(1+\frac{\tau}{3}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)$$
$$\cdot\left(1+\frac{\tau}{3}\right)\cdot we_o \quad \text{(by using equation (13))}$$
$$\leqslant\frac{\gamma}{\alpha}\cdot s_o^\alpha+\mu\cdot\left(1+\frac{\tau}{3}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^\alpha\right)$$
$$\cdot we_o \quad \text{(by using equations (12) and (22))}$$
$$\leqslant c\cdot s_o^\alpha+c\cdot we_o=c\cdot(s_o^\alpha+we_o)$$
$$\Rightarrow\left(\frac{dG_a}{dt}+\gamma\cdot\frac{d\Phi}{dt}\right)\leqslant c\cdot\frac{dG_o}{dt}$$

Case 3: $we_a > T^\alpha$ and $we_l > T^\alpha$ where

$$s_a=\left(1+\frac{\tau}{3}\right)\cdot\min\left((we_a(t))^{1/\alpha},T\right)=\left(1+\frac{\tau}{3}\right)\cdot T$$
$$\leq\left(1+\frac{\tau}{3}\right)\cdot we_a^{1/\alpha} \quad (29)$$

(**a**) If $we_l>\left(\frac{1}{1+9\alpha}\right)\cdot we_a$ (30)

Then using previous *Lemma* 3 and *Lemma* 4,

$$\left(\frac{dG_a}{dt}+\gamma\cdot\frac{d\Phi}{dt}\right)=\frac{dG_a}{dt}+\gamma\cdot\left(\frac{d\Phi_o}{dt}+\frac{d\Phi_a}{dt}\right)\quad\left(\frac{dG_a}{dt}+\gamma\cdot\frac{d\Phi}{dt}\right)\leqslant(we_a+s_a^\alpha)+\gamma$$
$$\cdot\left[\frac{we_l}{(1-\delta)}-\frac{\left(1+\frac{\tau}{3}\right)\cdot we_l^2}{(2-1/\alpha)\cdot we_a}\right]\quad\text{(by using equations (6) and (9))}$$
$$\leqslant\left(we_a+\left(1+\tfrac{\tau}{3}\right)^\alpha\cdot we_a\right)+\gamma\cdot\left[\frac{we_a}{(1-1/2\alpha)}-\frac{\left(1+\frac{\tau}{3}\right)\cdot we_l^2}{(2-1/\alpha)\cdot we_a}\right]$$

(by using equation (29))

$$\leqslant\left(1+\left(1+\tfrac{\tau}{3}\right)^\alpha\right)\cdot we_a+we_a$$
$$\cdot\frac{\gamma}{(1-1/2\alpha)}-\frac{\gamma}{(2-1/\alpha)}\cdot\frac{\left(\left(\frac{1}{1+9\alpha}\right)\cdot we_a\right)^2\cdot\left(1+\frac{\tau}{3}\right)}{we_a}\quad\text{(by using equation (30))}$$
$$=\left[\left(1+\left(1+\tfrac{\tau}{3}\right)^\alpha\right)+\frac{\gamma}{(1-1/2\alpha)}-\frac{\gamma}{(2-1/\alpha)}\cdot\left(\frac{1}{1+9\alpha}\right)^2\cdot\left(1+\tfrac{\tau}{3}\right)\right]\cdot e_a$$
$$(31)$$

$$\begin{cases}\because\alpha>1\Rightarrow 2\alpha-1>1\Rightarrow(1/2\alpha-1)<1\\ \Rightarrow\frac{1}{(1-1/2\alpha)}=\frac{2\alpha}{(2\alpha-1)}=1+\frac{1}{(2\alpha-1)}<2\end{cases}\quad(32)$$

Using the result of Eqs. (10), (18) and (32) in (31), we have

$$\left(\frac{dG_a}{dt}+\gamma\cdot\frac{d\Phi}{dt}\right)\leqslant\left[\left(1+\left(1+\frac{\tau}{3}\right)^{\alpha}\right)+2\gamma-\frac{1}{2}\cdot\gamma\cdot\left(\frac{1}{1+9\alpha}\right)^{2}\cdot\left(1+\frac{\tau}{3}\right)\right]\cdot\left(1+\frac{\tau}{3}\right)\cdot we_o$$
$$\leqslant\left[\left(1+\left(1+\frac{\tau}{3}\right)^{\alpha}\right)+2\gamma-\frac{1}{2}\cdot\gamma\cdot\left(\frac{1}{1+9\alpha}\right)^{2}\right]\cdot\left(1+\frac{\tau}{3}\right)\cdot we_o$$
$$=\left[\left(1+\left(1+\frac{\tau}{3}\right)^{\alpha}\right)+\gamma\cdot\left(2-\frac{1}{2}\cdot\left(\frac{1}{1+9\alpha}\right)^{2}\right)\right]\cdot\left(1+\frac{\tau}{3}\right)\cdot we_o \quad (33)$$

Using the result of Eq. (17) in (33), we have

$$\left(\frac{dG_a}{dt}+\gamma\cdot\frac{d\Phi}{dt}\right)\leqslant\left[\left(1+\left(1+\frac{\tau}{3}\right)^{\alpha}\right)+\gamma\cdot(1+1)\right]\cdot\left(1+\frac{\tau}{3}\right)\cdot we_o$$
$$=\left[\left(1+\left(1+\frac{\tau}{3}\right)^{\alpha}\right)+2\cdot\left(\frac{\alpha-1}{512\alpha}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^{\alpha}\right)\right]\cdot\left(1+\frac{\tau}{3}\right)\cdot we_o \quad \text{(by using equation (11))}$$
$$=\left[\left(1+\left(1+\frac{\tau}{3}\right)^{\alpha}\right)+\frac{2}{512}\cdot(1-1/\alpha)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^{\alpha}\right)\right]\cdot\left(1+\frac{\tau}{3}\right)\cdot we_o$$
$$\leqslant\left[\left(1+\left(1+\frac{\tau}{3}\right)^{\alpha}\right)+\frac{2}{512}\cdot\left(1+\left(1+\frac{\tau}{3}\right)^{\alpha}\right)\right]\cdot\left(1+\frac{\tau}{3}\right)\cdot we_o$$
$$=\frac{514}{512}\cdot\left(1+\frac{\tau}{3}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^{\alpha}\right)\cdot we_o$$
$$=\mu\cdot\left(1+\frac{\tau}{3}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^{\alpha}\right)\cdot we_o=c\cdot we_o$$
$$\leqslant c\cdot(s_o^{\alpha}+we_o)\leqslant c\cdot\frac{dG_o}{dt}\Rightarrow\left(\frac{dG_a}{dt}+\gamma\cdot\frac{d\Phi}{dt}\right)\leqslant c\cdot\frac{dG_o}{dt}$$

(**b**) If $we_l \leqslant \left(\frac{1}{1+9\alpha}\right)\cdot we_a$ (34)

Then in this case, we are adopting the result of *Lemma* 4 as $\frac{d\Phi_a}{dt}\leqslant 0$ (as the value of it is negative).

Then using *Lemma* 3 we have

$$\left(\frac{dG_a}{dt}+\gamma\cdot\frac{d\Phi}{dt}\right)=\frac{dG_a}{dt}+\gamma\cdot\left(\frac{d\Phi_o}{dt}+\frac{d\Phi_a}{dt}\right)$$
$$\left(\frac{dG_a}{dt}+\gamma\cdot\frac{d\Phi}{dt}\right)\leqslant\frac{dG_a}{dt}+\gamma\cdot\left(\frac{d\Phi_o}{dt}\right)$$
$$\left(\frac{dG_a}{dt}+\gamma\cdot\frac{d\Phi}{dt}\right)\leqslant(we_a+s_a^{\alpha})+\gamma\cdot\frac{we_l}{(1-\delta)} \quad \text{(by using equation (6))}$$
$$\leqslant\left(we_a+\left(1+\frac{\tau}{3}\right)^{\alpha}\cdot we_a\right)+\gamma\cdot\frac{we_l}{(1-1/2\alpha)} \quad \text{(by using equation (29))}$$
$$\leqslant\left(1+\left(1+\frac{\tau}{3}\right)^{\alpha}\right)\cdot we_a+\frac{\gamma}{(1-1/2\alpha)}\cdot we_a$$
$$\leqslant\left(1+\left(1+\frac{\tau}{3}\right)^{\alpha}\right)\cdot we_a+2\gamma\cdot we_a \quad \text{(by using equation (32))}$$
$$\leqslant\left[\left(1+\left(1+\frac{\tau}{3}\right)^{\alpha}\right)+2\left(\frac{\alpha-1}{512\alpha}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^{\alpha}\right)\right]\cdot\left(1+\frac{\tau}{3}\right)\cdot we_o \quad \text{(by using equations (11) and (10))}$$
$$=\left[\left(1+\left(1+\frac{\tau}{3}\right)^{\alpha}\right)+\frac{2}{512}\cdot(1-1/\alpha)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^{\alpha}\right)\right]\cdot\left(1+\frac{\tau}{3}\right)\cdot we_o$$
$$\leqslant\left[\left(1+\left(1+\frac{\tau}{3}\right)^{\alpha}\right)+\frac{2}{512}\cdot\left(1+\left(1+\frac{\tau}{3}\right)^{\alpha}\right)\right]\cdot\left(1+\frac{\tau}{3}\right)\cdot we_o$$
$$=\frac{514}{512}\cdot\left(1+\frac{\tau}{3}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^{\alpha}\right)\cdot we_o$$
$$=\mu\cdot\left(1+\frac{\tau}{3}\right)\cdot\left(1+\left(1+\frac{\tau}{3}\right)^{\alpha}\right)\cdot we_o$$
$$=c\cdot we_o \quad \text{(by using equation (12))}$$
$$\leqslant c\cdot(s_o^{\alpha}+we_o)\leqslant c\cdot\frac{dG_o}{dt}$$
$$\Rightarrow\left(\frac{dG_a}{dt}+\gamma\cdot\frac{d\Phi}{dt}\right)\leqslant c\cdot\frac{dG_o}{dt}\alpha=2\alpha=2$$

Analytical proofs of all three cases (possibilities) reflect that the running condition is satisfied. Combining job arrival and completion condition with boundary condition and *Lemma* 5, we conclude that Theorem 1 follows. We have calculated the results of related algorithms and EtRR on $\alpha = 2$ and 3 and the summary of comparison of their results is shown in Table 2. Among all clairvoyant scheduling algorithms, the competitive value of EtRR is least. Among all non-clairvoyant scheduling algorithms, RR-AJT and EtRR are having minimum values of competitiveness. RR-AJT is having a big limitation that in RR-AJT all jobs must be released at time $t = 0$, which is not possible practically. EtRR is free from such limitations; hence EtRR is having best (minimum) competitive value.

5. Conclusion and future scope

In this work, we propose an online non-clairvoyant job scheduling algorithm Executed-time Round Robin (EtRR). EtRR is using a variant of WRR, where the weights of jobs are generated and assigned by scheduler using executed time of a job. The objective of EtRR is to minimize weighted flow time plus energy. EtRR is $O(1)$-competitive when using a processor at maximum speed $(1 + \tau/3)T$, where $0 < \tau \leqslant (3\alpha)^{-1}$. In EtRR there is a limitation that the speed must be re-evaluated at discrete level of time (when any job releases or finishes). Even the weights are not provided they are artificially generated, the competitive ratio of EtRR is comparatively smaller than others. The numeric values of competitiveness calculated on $\alpha = 2$ and $\alpha = 3$ show that our proposed algorithm outperforms the existing algorithms. The futuristic enhancement of our study can be to evaluate the working of EtRR in the multi-processor environment and conducting the experiments in real environment. One open problem is to reduce the competitive ratio achieved in this work.

Acknowledgments

The authors are obliged to the Editor-in-Chief and anonymous reviewers for their perceptive comments and precious suggestions which help us to enhance the content and the presentation of this paper. We would like to thank Mr. R. Kumar for the linguistic review of our research paper.

References

Albers, S., 2010. Energy-efficient algorithms. Commun. ACM 53 (5), 86–96.

Angelopoulos, S., Lucarelli, G., Thang, N.K., 2015. Primal-dual and dual-fitting analysis of online scheduling algorithms for generalized flow-time problems. In: Proceedings of 23rd Annual European Symposium (ESA '15), pp. 35–46.

Azar, Y., Devanue, N.R., Huang, Z., Panighari, D., 2015. Speed scaling in the non-clairvoyant model. In: Proceedings of the 27th Annual ACM Symposium on Parallelism in Algorithms and Architectures (SPAA '15), pp. 133–142.

Bansal, N., Kimbrel, T., Pruhs, K., 2007. Speed scaling to manage energy and temperature. J. ACM 54 (1), 3.

Bansal, N., Chan, H.L., Pruhs, K., 2009. Speed scaling with an arbitrary power function. In: Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '09), pp. 693–701.

Becchetti, L., Leonardi, S., 2004. Nonclairvoyant scheduling to minimize the total flow time on single and parallel machines. J. ACM 51 (4), 517–539.

Becchetti, L., Leonardi, S., Spaccamela, A.M., Pruhs, K., 2006. Online weighted flow time and deadline scheduling. J. Discrete Algorithms 4 (3), 339–352.

Bell, P.C., Wong, P.W.H., 2014. Multiprocessor speed scaling for jobs with arbitrary sizes and deadlines. J. Comb. Optim. 29 (4), 739–749.

Berman, P., Coulston, C., 1999. Speed is more powerful than clairvoyance. Nordic J. Comp. 6 (2), 181–193.

Chan, S.H., Lam, T.W., Lee, L.K., 2011a. Scheduling for weighted flow time and energy with rejection penalty. In: Proceedings of the 28th International Symposium on Theoretical Aspects of Computer Science (STACS '11), pp. 392–403.

Chan, H.L., Edmonds, J., Lam, T.W., Lee, L.K., Marchetti-Spaccamela, A., Pruhs, K., 2011b. Nonclairvoyant speed scaling for flow and energy. Algorithmica 61 (3), 507–517.

Chan, S.H., Lam, T.W., Lee, L.K., Zhu, J., 2013. Nonclairvoyant sleep management and flow-time scheduling on multiple processors. In: Proceedings of the 25th Annual ACM Symposium on Parallelism in Algorithms and Architectures (SPAA 2013), pp. 261–270.

Edmonds, J., 2000. Scheduling in the dark. Theor. Comput. Sci. 235 (1), 109–141.

Fox, K., Im, S., Moseley, B., 2013. Energy efficient scheduling of parallelizable jobs. In: Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '13), pp. 948–957.

Gupta, A., Im, S., Krishnaswamy, R., Moseley, B., Pruhs, K., 2012. Scheduling heterogeneous processors isn't as easy as you think. In: Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2012), pp. 1242–1253.

Im, S., Kulkarni, J., Munagala, K., Pruhs, K., 2014a. SelfishMigrate: a scalable algorithm for non-clairvoyantly scheduling heterogeneous processors. In: Proceedings of 55th IEEE Annual Symposium on Foundations of Computer Science (FOCS '14), pp. 531–540.

Im, S., Moseley, B., Pruhs, K., 2014b. Online scheduling with general cost functions. SIAM J. Comput. 43 (1), 126–143.

Im, S., Kulkarni, J., Moseley, B., 2015. Temporal fairness of round robin: competitive analysis for Lk-norms of flow time. In: Proceedings of the 27th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA' 15), pp. 155–160.

Kalyanasundaram, B., Pruhs, K., 2000. Speed is as powerful as clairvoyant. J. ACM 47 (4), 617–643.

Kalyanasundaram, B., Pruhs, K., 2003. Minimizing flow time nonclairvoyantly. J. ACM 50 (4), 551–567.

Lam, T.W., Lee, L.K., To, I.K.K., Wong, P.W.H., 2008. Speed scaling functions for flow time scheduling based on active job count. In: Proceedings of the 16th Annual European Symposium on Algorithms (ESA '08), pp. 647–659.

Lam, T.W., Lee, L.K., Ting, H.F., To, I.K.K., Wong, P.W.H., 2009. Sleep with guilt and work faster to minimize flow plus energy. In: Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP '09), pp. 665–676.

Lam, T.W., Lee, L.K., To, I.K.K., Wong, P.W.H., 2013. Online speed scaling based on active job count to minimize flow plus energy. Algorithmica 65 (3), 605–633.

Markoff, J., Lohr, S., 2002. Intel's huge bet turns iffy. In: New York Times. http://www.nytimes.com/2002/09/29/business/intel-s-huge-bet-turns-iffy.html.

Motwani, R., Phillips, S., Torng, E., 1994. Nonclairvoyant scheduling. Theor. Comput. Sci. 30 (1), 17–47.

Muthukrishnan, S., Rajaraman, R., Shaheen, A., Gehrke, J., 1999. Online scheduling to minimize average stretch. In: Proceeding of IEEE 40th Annual Symposium on Foundations of Computer Science (FOCS '99), pp. 433–442.

Pruhs, K., Uthaisombut, P., Woeginger, G., 2008. Getting the best response for your erg. ACM Trans. Algorithms 4 (3), 38.

Steele, J.M., 2004. The Cauchy–Schwarz Master Class: An Introduction to the Art of Mathematical Inequalities, Kindle ed. Cambridge University Press, p. 136.

Sun, H., He, Y., Hsu, W.J., Fan, R., 2014. Energy-efficient multiprocessor scheduling for flow time and makespan. Theor. Comput. Sci. 550, 1–20.

Yao, F., Demers, A., Shenker, S., 1995. A scheduling model for reduced CPU energy. In: Proceedings of 36th Annual Symposium on Foundations of Computer Science (FOCS '95), pp. 374–382.

Yun, H.S., Kim, J., 2003. On energy-optimal voltage scheduling for fixed priority hard real-time systems. ACM T. Embed. Comput. S. 2 (3), 393–430.