OUTLAY MINIMIZATION ALGORITHMS FOR INFORMATION MIDPOINT ORGANIZATION

R.Jayasri [1], K.Sathya [2]

[1]M.Phil, Research Scholar, Department of Computer Science, Padmavani Arts and Science College for Women, Salem, TN

[2] Assistant Professor, Department of Computer Science, Padmavani Arts and Science College for Women, Salem, TN

Abstract*—suitable to the increasing usage of cloud computing applications, it is significant to reduce energy cost obsessive by a data midpoint, and simultaneously, to improve quality of repair via data midpoint organization. One capable advance is to switch some servers in a information midpoint to the idle mode for saving force while to keep a suitable number of servers in the energetic mode for providing timely service. In this paper designed both online and offline algorithms for this problem. For the offline algorithm, we formulate information midpoint administration as a outlay minimization problem by considering energy asking price, delay outlay (to measure service quality), and switching cost (to change servers' active/idle mode). Then, we examine convinced properties of an optimal resolution which lead to a active programming based algorithm. Moreover, by revising the solution procedure, we successfully eliminate the recursive system and complete an optimal offline algorithm with a polynomial density. For the online algorithm, We propose it by considering the worst case scenario for potential workload. In simulation, we show this online algorithm can always present near-optimal solutions.

***Index Terms*—information midpoint organization, offline algorithm, Runtime programming, Online Applications**

I.INTRODUCTION

The goals of information midpoint organization may include minimizing energy cost and improving quality of service. Power cost is a major part of a data center's budget, which should be minimized to decrease service provider's cost, and more importantly, to keep our Earth green. One advance to reduce force price is to change several servers since dynamic mode to inactive type each time feasible. These switching decisions are entire based on location of the servers, such as system position or storage space position. Meanwhile, we desire to achieve superior service excellence, which can be careful by the standard wait of serves' responding point. For this function, there must be sufficient dynamic servers in arrange to procedure tasks initiate by trade in point. To complete both goal of information midpoint organization, we should sustain a appropriate quantity of dynamic servers and then dispense jobs to these dynamic servers.

We focal point on intra-center organization and consider active workload more than a interlude of point. In the minimization trouble, the price includes force price, interruption price and switching price.

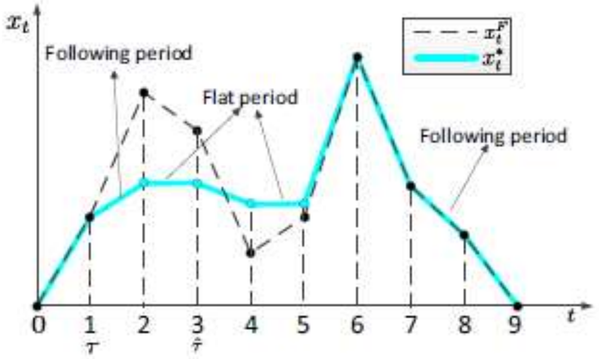


Figure 1 OUTLAY MINIMIZATION ALGORITHMS

The we will design an offline algorithm with the consideration on integer requirement. The offline problem is an integer optimization, which is NP-hard in general. But we will show that our costume designed offline algorithm can optimally solve this problem in polynomial-time. We formulate the cost minimization problem as a mixed integer optimization model and we call it the Original Problem (OP) model. Then, in order to obtain a better formulation, we apply reformulation techniques and get an integer optimization model and we call it the Reformulated Problem (RP) model. The RP model has a much smaller problem size (much less numbers of variables and constraints). We discuss two special cases. The first special case is that there is no switching cost and we call it the Zero Switching Cost (ZSC) model.

RELATED WORK

We design a dynamic programming based optimal offline Algorithm and we call it the Dynamic Programming (DP)- offline algorithm. We further discuss how to avoid the recursive process in dynamic programming, from which an optimal offline algorithm with a low complexity O(KN2) is obtained, where K is the number of local maximum points of xF t and N is the number of servers in the data center. We call it the Non-Recursive DP (N RDP)-offline algorithm. We design an online algorithm, which makes decisions by considering the worst case scenario for future workload, and we call it the Worst Case (WC)-online algorithm.

In simulation, we compare solutions by this online algorithm with optimal offline solutions and show the nearoptimal performance of this online algorithm. Moreover, the cost achieved by another online solution xF t (with no consideration on switching cost) is more than twice of the minimum cost achieved by the optimal solution, which indicates the importance of taking the switching cost into the consideration.

Considering a data center with multiple servers and time varying workload (measured by the number of jobs at different time), the data center management problem is on how to distribute jobs to each server in the data center. Servers with job assignment are in the active mode while servers without job assignment can be in the power-saving idle mode. For simplicity, we call switch operations between these two modes as “turn on” and “turn off,” although a server is not off when it is in the idle mode.

Denote the number of servers in a data center as N. We consider a time slot based scheme, i.e., at the beginning of each time slot, the workload is estimated and assigned to each server. Denote T as the number of all time slots and the workload in time slot t as γt. We set the initial workload γ0 = 0 and the ending workload γT = 0. Consider homogenous servers in a data center, i.e., each server has the same capacity C (the maximum number of jobs that can be served in a time slot), the same operating cost function p(γ), where γ ∈ [0,C] is the assigned workload, and the same switching costs βon, βoff ≥ 0 to turn on or off a server. Assume there are enough number of servers, i.e., γt ≤ NC. Operating cost function p(γ) should be non-negative and non-decreasing. We further assume that p(γ) is a convex function that may include energy cost and delay cost [11], where energy cost is consisted of costs for energy consumption, for cooling, and for power distribution, and delay cost measures the quality of service. The operating cost when a server is in the idle mode, p(0), is usually not zero.

III.METHODOLOGY

The integer requirement is missing in problem formulation, algorithm design, and performance analysis.As a consequence, the obtained solution usually is infeasible. Even for the rare case that the obtained solution is feasible, that solution may not have a constant approximation bound as claimed in. This is because that the authors thought that the problem is convex and then made a performance analysis based on its dual problem. But any optimization problem with non-continuous variables is non-convex and cannot be analyzed by its dual problem due to unknown duality gap. where *xt*, *xt*on, and *xt* off are integer variables; *T, λt, β*on*, β*off, *N*, *x*0, and *xT* are constants. Comparing with the OP model, his Reformulated Problem (RP) model has less variables and constraints and all its constraints are linear constraints. The RP model is an integer optimization problem due to the integer requirement on *xt* (the number of active servers), *xt*on and *xt* off. Such an integer optimization problem is NP-hard in the data center management problem is on how to distribute jobs to each server in the data center. Servers with job assignment are in the active mode while servers without job assignment can be in the power-saving idle mode. For simplicity, we call switch operations between these two modes as “turn on” and “turn off,” although a server is not off when it is in the idle mode.

1. *Problem Formulation*

Denote the number of servers in a data center as N. We consider a time slot based scheme, i.e., at the beginning of each time slot, the workload is estimated and assigned to each server. Denote T as the number of all time slots and the workload in time slot t as γt. We set the initial workload γ0 = 0 and the ending workload γT = 0. Consider homogenous servers in a data center, i.e., each server has the same capacity C (the maximum number of jobs that can be served in a time slot), the same operating cost function p(γ), where γ ∈ [0,C] is the assigned workload, and the same switching costs βon, βoff ⩾ 0 to turn on or off a server. Assume there are enough number of servers, i.e., γt ⩽ NC.

For a time slot t, the workload assignment constraint is ΣN i=1 γt;i = γt (1 ⩽ t ⩽ T − 1) , (1) where γt;i is the workload assigned to a server i. The number of active servers xt is equal to the number of positive γt;I values, i.e., xt = ΣN i=1 I(γt;i > 0) (1 ⩽ t ⩽ T − 1) , (2) where I(·) is the indicator function. Since γ0 = γT = 0 and (1), we have γ0;i = γT ;i = 0 and x0 = xT = 0. Denote xt on = max{xt − xt□1, 0} (1 ⩽ t ⩽ T) (3) xt off = max{xt□1 − xt, 0} (1 ⩽ t ⩽ T) (4) as the number of servers switched on/off at time t, respectively. Then the offline problem can be formulated as follows. Min ΣTt=1 [ΣN i=1 p(γt;i) + βon · xton + βoff · xt Off] (5) s.t. (1), (2), (3), (4) γ t;i ∈ [0,C], xt ∈ [⌈ t C ⌉,N] (1⩽t⩽T − 1, 1⩽i⩽N) xton, xt off ⩾ 0 (1 ⩽ t ⩽ T) , where γt;i are continuous variables; xt, xton, xt off are integer variables; T,N, βon, βoff, γt,C, γT ;i, x0, xT are constants. Eq. (5) is a straightforward nonlinear formulation. We call it the Original Problem (OP) model.

2. *Reformulation*

The OP model is not in a good form for optimization. In particular, constraints (2), (3), and (4) are nonlinear. In this section, we apply several reformulation approaches to simplify the problem structure in terms of line arising all constraints, and to reduce the problem size in terms of reducing the number of constraints and variables. First define normalized workloads and a new cost for the developed minimization delay organization function we follows as to delay minimization of data

$$
\begin{aligned}
\lambda_t &= \frac{\gamma_t}{C} && (0 \le \gamma_t \le NC) && (6)\\
\lambda_{t,i} &= \frac{\gamma_{t,i}}{C} && (1 \le i \le N, 0 \le \gamma_{t,i} \le C) && (7)\\
f(\lambda_{t,i}) &= p(\gamma_{t,i}) - p(0) && (1 \le i \le N, 0 \le \lambda_{t,i} \le 1). && (8)
\end{aligned}
$$

It’s easy to verify that f(0) = 0. Then we separate the objective function as the operating cost ΣT t=1 ΣN i=1 p(γt;i) and the switching cost ΣT t=1(βon · xton + βoff · xt off). For the operating cost, we

$$
\begin{aligned}
&\sum_{t=1}^{T}\sum_{i=1}^{N} p(\gamma_{t,i}) = \sum_{t=1}^{T}\sum_{i=1}^{N}(f(\lambda_{t,i}) + p(0))\\
&= \sum_{t=1}^{T-1}\sum_{\substack{1\le i\le N}}^{\lambda_{t,i}>0} f(\lambda_{t,i}) + TNp(0) = \sum_{t=1}^{T-1} x_t f\left(\frac{\lambda_t}{x_t}\right) + TNp(0)
\end{aligned}
$$

where the first equation holds by (8), the second equation holds by f(0) = 0 and γT ;i = 0, and the third equation holds by the convexity of f(·) and (1), (2). Since TNp(0) is a constant, we can remove it from the objective function.

$$x^t_{\textbf{on}} \geq x_t - x_{t-1} \quad (1 \leq t \leq T-1)\,, \tag{9}$$

$$x^t_{\textbf{off}} \geq x_{t-1} - x_t \quad (2 \leq t \leq T)\,. \tag{10}$$

Note that with objective (5), constraints (9) and (10) can guarantee (3) and (4) respectively.

After these reformulations, we have

$$\text{Min} \quad \sum_{t=1}^{T-1} x_t f\left(\frac{\lambda_t}{x_t}\right) + \sum_{t=1}^{T} (\beta_{\text{on}} \cdot x^t_{\text{on}} + \beta_{\text{off}} \cdot x^t_{\text{off}}) \tag{11}$$

$$\begin{aligned} \text{s.t.} \quad & (9), (10) \\ & x_t \in [\lceil \lambda_t \rceil, N], x^t_{\text{on}} \geq 0 \qquad (1 \leq t \leq T-1) \\ & x^t_{\text{off}} \geq 0 \qquad (2 \leq t \leq T)\,, \end{aligned}$$

The RP model is an integer optimization problem due to the integer requirement on *xt* (the number of active servers), *xt*on and *xt* off.

3. *Optimal Solution When No Switching Cost*

As a starting point, we consider a special case of βon = βoff = 0 (i.e., no switching cost). The result on this special case can be used in developing an optimal offline algorithm for the general case of βon, βoff ⩾ 0. First let us denote $F(x) = xf\left(\frac{1}{x}\right) \quad (x \geq 1)\,.$

Given that *f*(*x*) is a convex function, it can be verified that *F*(*x*) is also a convex function. Then the operating cost can be re-written as $\sum_{t=1}^{T-1} x_t f\left(\frac{\lambda_t}{x_t}\right) = \sum_{t=1}^{T-1} \lambda_t \frac{x_t}{\lambda_t} f\left(\frac{\lambda_t}{x_t}\right) = \sum_{t=1}^{T-1} \lambda_t F\left(\frac{x_t}{\lambda_t}\right)$ and thus the problem is

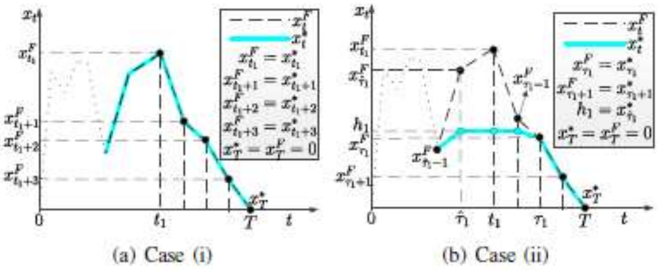
$$\begin{aligned} \min \quad & \textstyle\sum_{t=1}^{T-1} \lambda_t F\left(\frac{x_t}{\lambda_t}\right) \qquad (13) \\ \text{s.t.} \quad & x_t \in [\lceil \lambda_t \rceil, N] \quad (1 \leq t \leq T-1)\,, \end{aligned}$$

IV. EXPERIMENTAL SETUP

we will design the optimal offline algorithm and online algorithm based on the models and the lemmas we have concluded before. In this section, we assume that we know all γt values, 1 ≤ t ≤ T, to design the optimal offline algorithm. In Section V, we assume we only know the past and current γt values to design an online algorithm. Although the optimal offline algorithm cannot be implemented for online running, its solution can provide a performance benchmark for any online algorithm. Our optimal offline algorithm is based on the dynamic programming, which will be discussed in Section IV-A. We name this algorithm as the dynamic programming (DP) based offline algorithm. However, the DP-offline algorithm has an exponential complexity. So in Section IV-B we improve the solution procedure and get a polynomial complexity algorithm. We name the new offline algorithm as the non-recursive DP (NRDP) based offline algorithm. In Section V, we design the online algorithm based on considering the worst case scenario for future workload.

Optimal offline algorithm based on dynamic programming

The following is the design of the DP-offline algorithm, which contains a recursive process. First let us define a sub problems as follows. Consider a sub problem of minimizing the total cost for τ ∈ [0, t] under given workload γ_ and specified ending xt = h ≥ γt. Denote the minimum total cost for this sub problem as c(t, h). For our problem, we have xT = 0 and thus our problem is to determine c(T, 0). Note that lemmas proved in Section III-C also hold for a sub problem.



(a) Case (i)

(b) Case (ii)

Fig. 5. Two cases for calculating $c(T, 0)$.

The first step. We first develop a recursive formula for *c*(*T*, 0). By (14) and *γT* = 0, we have *xF T* = 0. Then we define that

t_1 is the smallest value such that $x^F_{t_1} \geq x^F_{t_1+1} \geq \cdots \geq x^F_T$. (15)

Since *x_T* = 0 = *xF T* , an optimal solution *x_ t* must fall into Zither one of the following two categories:

(i) *x_ t* = *xF t* for all *t* ∈ [*t*1, *T*] (see Fig. 5(a)) or (ii) there is a *τ*1 ∈ [*t*1 + 1, *T*] and *h*1 ∈ [1, *xF t*1 − 1]

(ii) such that *x_ t* = *xF t* for *t* ∈ [*τ*1, *T*] and *x_ _*1□1 = *h*1 < *xF* _1□1 (see Fig. 5(b)).

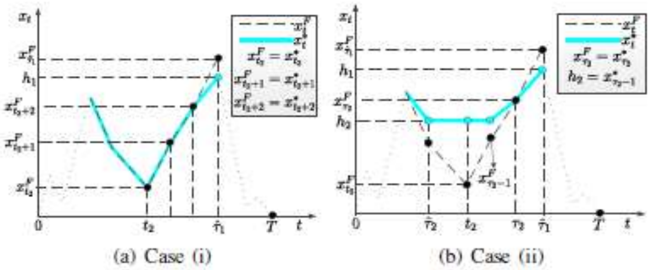
Case (i): the minimum total cost for *t* ∈ [0, *t*1] with *xt*1 = *xF t*1 is *c*(*t*1, *xF t*1). For *t* ∈ [*t*1, *T*], we have *x_ t* = *xF t* and thus can calculate the total cost (not including the operating cost at time *t*1 since this cost is counted in *c*(*t*1, *xF t*1) already). Denote this cost as *C*(*t*1;*xF t* 1);(*T*;0). Then for *c*(*T*, 0), we have *c*(*T*, 0) = *c*(*t*1, *xF t*1) + *C*(*t*1;*xF t* 1);(*T*;0) .

Case (ii): We define that

Such a $\hat{\tau}_1$ always exists since $x^F_0 = 0 < h_1 < x^F_{t_1}$. Moreover, it is easy to verify that $x^F_t > h_1$ for $t \in [\hat{\tau}_1 + 1, \tau_1 - 1]$. Then we have $x^*_{\tau_1-1} = x^*_{\tau_1-2} = \cdots = x^*_{\hat{\tau}_1} = h_1$ by Lemma 5. But since $x^F_{\hat{\tau}_1} = h_1 \neq x^F_{\hat{\tau}_1-1}$ (or $x^F_{\hat{\tau}_1} > h_1 > x^F_{\hat{\tau}_1-1}$), $x^*_{\hat{\tau}_1-1}$ may not be equal to $x^*_{\hat{\tau}_1}$ by Lemma 4 (or Lemma 6), i.e., the flat period starts at $\hat{\tau}_1$. Thus, we can say that $\hat{\tau}_1$ defined in (16) is the point where the flat line with height h_1 intersects with curve (t, x^F_t). The minimum total cost for $t \in [0, \hat{\tau}_1]$ with $x_{\hat{\tau}_1} = h_1$ is $c(\hat{\tau}_1, h_1)$. Since $x^*_t = h_1$ for $t \in [\hat{\tau}_1, \tau_1 - 1]$ and $x^*_t = x^F_t$ for $t \in [\tau_1, T]$, we can calculate the total cost for $t \in [\hat{\tau}_1, T]$ (not including the operating cost at time $\hat{\tau}_1$ since this cost is counted in $c(\hat{\tau}_1, h_1)$). Denote this cost as $c_{(\hat{\tau}_1, h_1), (T, 0)}$. Then we have

$$c(T, 0) = c(\hat{\tau}_1, h_1) + c_{(\hat{\tau}_1, h_1), (T, 0)}\,.$$

$\hat{\tau}_1$ is the largest value in $[1, t_1]$ such that $x^F_{\hat{\tau}_1} = h_1 \neq x^F_{\hat{\tau}_1-1}$ or $x^F_{\hat{\tau}_1} > h_1 > x^F_{\hat{\tau}_1-1}$. (16)



(a) Case (i) (b) Case (ii)

Fig. 6. Two cases for calculating $c(\hat\tau_1, h_1)$.

By considering both cases, we have $c(T,0) = \min\{c(t_1, x^F_{t_1}) + c_{(t_1,x^F_{t_1}),(T,0)}, \min_{h_1\in[1,x^F_{t_1}-1]}(c(\hat\tau_1,h_1) + c_{(\hat\tau_1,h_1),(T,0)})\}$.

Actually, Cases (i) and (ii) can be combined together since the former one is a special version of the later to some extent. The $\hat\tau_1$ defined by $h_1 = x^F_{t_1}$ and (16) is t_1. Therefore, we can rewrite $c(t_1, x^F_{t_1}) + c_{(t_1,x^F_{t_1}),(T,0)}$ as $c(\hat\tau_1, x^F_{t_1}) + c_{(\hat\tau_1,x^F_{t_1}),(T,0)}$. Then we have

$$c(T,0) = \min_{h_1\in[1,x^F_{t_1}]} \left(c(\hat\tau_1, h_1) + c_{(\hat\tau_1,h_1),(T,0)}\right) . \quad (17)$$

The above equation gives us a recursive approach to calculate $c(T,0)$ based on some $c(\hat\tau_1, h_1)$ values, where $h_1 \in [1, x^F_{t_1}]$ and $\hat\tau_1$ is defined by (16), i.e., we have $\hat\tau_1 \in [1, t_1]$, $h_1 \in [1, x^F_{t_1}]$, and $x^F_{\hat\tau_1} = h_1 \neq x^F_{\hat\tau_1-1}$ or $x^F_{\hat\tau_1} > h_1 > x^F_{\hat\tau_1-1}$.

Note that although the range for h_1 is $[1, x^F_{t_1}]$ in (17), some small value of h_1 is impossible to yield an optimal solution due to the constraint set by γ_t. That is, if we assume a small value for h_1, we can find a ^τ_1 and $x_t = h_1$ for $t \in$ [^τ_1, $\tau_1 - 1$]. If $\gamma_t > h_1$ for any $t \in$ [^τ_1, $\tau_1 - 1$], then this hypothetical optimal solution is in fact infeasible, i.e., this h_1 is too small. As a result, we can skip the calculation of $c($^$\tau_1, h_1) + c_{(\text{^\_}1;h_1);(T;0)}$ for this h_1.

The second step.

To calculate a particular c(^τ1, h1) in (17), we define that

t_2 is the smallest value such that $x^F_{t_2} \le x^F_{t_2+1} \le \cdots \le x^F_{\hat\tau_1}$. (18)

An optimal solution x^*_t for $c(\hat\tau_1, h_1)$ must fall into either one of the following two categories: (i) $x^*_t = x^F_t$ for all $t \in [t_2, \hat\tau_1 - 1]$ (see Fig. 6(a)) or (ii) there is a $\tau_2 \in [t_2 + 1, \hat\tau_1 - 1]$ and $h_2 \in [x^F_{t_2} + 1, h_1]$ such that $x^*_t = x^F_t$ for $t \in [\tau_2, \hat\tau_1 - 1]$ and $x^*_{\tau_2-1} = h_2 > x^F_{\tau_2-1}$ (see Fig. 6(b)).[1] We can further analyze each case as follows.

Case (i): The minimum total cost for $t \in [0, t_2]$ with $x_{t_2} = x^F_{t_2}$ is $c(t_2, x^F_{t_2})$. Since $x^*_t = x^F_t$ for $t \in [t_2, \hat\tau_1 - 1]$ and $x_{\hat\tau_1} = h_1$, we can calculate the total cost for $t \in [t_2, \hat\tau_1]$ (not including the operating cost at time t_2 since this cost is counted in $c(t_2, x^F_{t_2})$ already). Denote this cost as $c_{(t_2,x^F_{t_2}),(\hat\tau_1,h_1)}$. Then we have

$$c(\hat\tau_1, h_1) = c(t_2, x^F_{t_2}) + c_{(t_2,x^F_{t_2}),(\hat\tau_1,h_1)} \cdot$$

Case (ii): we can prove that any $h_2 > \max_{t=1}^{t_2} x^F_t$ cannot yield an optimal solution (see Lemma 7). For $h_2 \in [x^F_{t_2} + 1, \min\{h_1, \max_{t=1}^{t_2} x^F_t\}]$, we define that

$\hat\tau_2$ is the largest value in $[1, t_2]$ such that $x^F_{\hat\tau_2} = h_2 \neq x^F_{\hat\tau_2-1}$ or $x^F_{\hat\tau_2} < h_2 < x^F_{\hat\tau_2-1}$. (19)

Such a $\hat\tau_2$ always exists since there is an $x^F_t (= \max_{\tau=1}^{t_2} x^F_\tau) \ge h_2 > x^F_{t_2}$. Moreover, it is easy to verify that $x^F_t < h_2$ for $t \in [\hat\tau_2+1, \tau_2-1]$. then we have $x^*_{\tau_2-1} = x^*_{\tau_2-2} = \cdots = x^*_{\hat\tau_2} = h_2$ by Lemma 5. But since $x^F_{\hat\tau_2} = h_2$ (or $x^F_{\hat\tau_2} > h_2 > x^F_{\hat\tau_2-1}$), $x^*_{\hat\tau_2-1}$ may not be equal to $x^*_{\hat\tau_2}$ by Lemma 4 (or Lemma 6), i.e., the flat period starts at $\hat\tau_2$. Thus, we can say that $\hat\tau_2$ is the point where the flat line with height h_2 intersects with curve (t, x^F_t). The minimum total cost for $t \in [0, \hat\tau_2]$ with $x_{\hat\tau_2} = h_2$ is $c(\hat\tau_2, h_2)$. Since $x^*_t = h_2$ for $t \in [\hat\tau_2, \tau_2 - 1]$, $x^*_t = x^F_t$ for $t \in [\tau_2, \hat\tau_1 - 1]$, and $x^*_{\hat\tau_1} = h_1$. We can calculate the total cost for $t \in [\hat\tau_2, \hat\tau_1]$ (not including the operating cost at time $\hat\tau_2$ since this cost is counted in $c(\hat\tau_2, h_2)$). Denote this cost as $c_{(\hat\tau_2,h_2),(\hat\tau_1,h_1)}$. Then we have

$$c(\hat\tau_1, h_1) = c(\hat\tau_2, h_2) + c_{(\hat\tau_2,h_2),(\hat\tau_1,h_1)} \cdot$$

By considering both cases, we have

$$\begin{aligned} & c(\hat\tau_1, h_1) \\ &= \min\Bigg\{c(t_2, x^F_{t_2}) + c_{(t_2,x^F_{t_2}),(\hat\tau_1,h_1)}, \\ &\qquad \min_{h_2\in[x^F_{t_2}+1,\min\{h_1,\max_{t=1}^{t_2} x^F_t\}]} \left(c(\hat\tau_2, h_2) + c_{(\hat\tau_2,h_2),(\hat\tau_1,h_1)}\right)\Bigg\} \\ &= \min_{h_2\in[x^F_{t_2},\min\{h_1,\max_{t=1}^{t_2} x^F_t\}]} \left(c(\hat\tau_2, h_2) + c_{(\hat\tau_2,h_2),(\hat\tau_1,h_1)}\right). \quad (20) \end{aligned}$$

Non-recursive approach

we designed the DP-offline algorithm. However, this algorithm contains recursive progress which may leads to large complexity. In this section, we will show how to avoid the recursive process in dynamic programming and analyze the developed NRDP-offline algorithm's complexity. We first identify all $c(t, h)$ that should be determined during the calculation of $c(T, 0)$. The entire recursive process requires us to obtain some $c(t, h)$ values, $t \in [1, t_1]$, that satisfy one of the following three cases: $x^F_t = h \neq x^F_{t\square 1}$, $x^F_{t\square 1} < h < x^F_t$, or $x^F_{t\square 1} > h > x^F_t$. Thus, if $x^F_t = x^F_{t\square 1}$ for a particular t, we do not need to calculate any $c(t, h)$ for this t; if $x^F_t > x^F_{t\square 1}$, we need to calculate $c(t, h)$ for $h \in [x^F_{t\square 1} + 1, x^F_t]$; while if $x^F_t < x^F_{t\square 1}$, we need to calculate $c(t, h)$ for $h \in [x^F_t, x^F_{t\square 1} - 1]$. For the example in Fig. 8, we have $t_1 = 14$, i.e., we need some $c(t, h)$ values for $t \le 14$. In particular, for $t = 12$, since $x^F_{11} = 9$ and $x^F_{12} = 6$, we need to calculate $c(12, 8)$, $c(12, 7)$, and $c(12, 6)$.

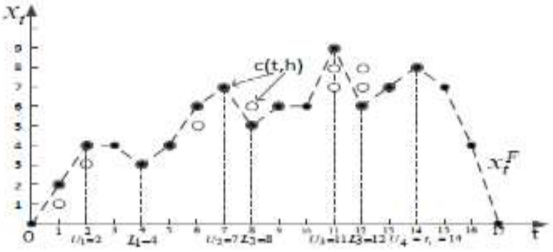
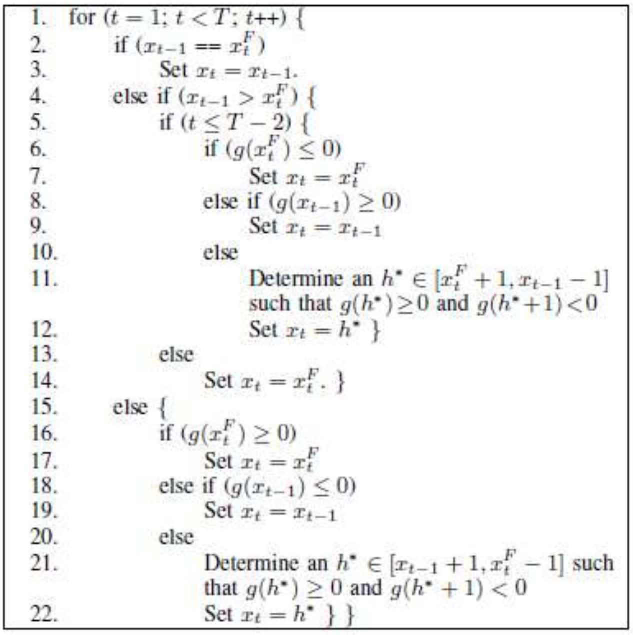


Fig. 8. An example for all $c(t, h)$ determined during the calculation of $c(T, 0)$.

[1] For the extreme case that $\tau_2 = \hat\tau_1$, $[\tau_2, \hat\tau_1 - 1]$ is an empty set and thus there is no t such that $x^*_t = x^F_t$, i.e., no following period.



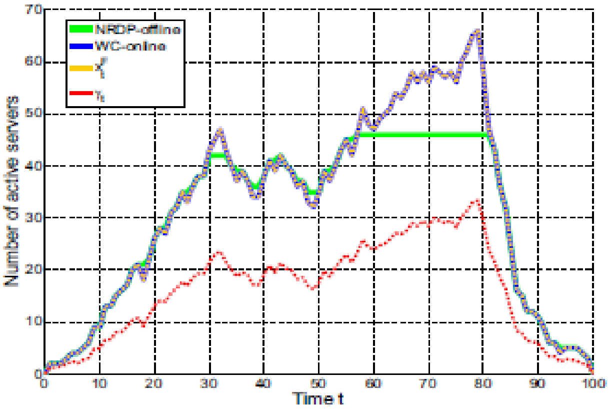
```
for $(t = 1; t < T; t{+}{+})$ {
    if $(x_{t-1} == x_t^F)$
        Set $x_t = x_{t-1}$.
    else if $(x_{t-1} > x_t^F)$ {
        if $(t \leq T - 2)$ {
            if $(g(x_t^F) \leq 0)$
                Set $x_t = x_t^F$
            else if $(g(x_{t-1}) \geq 0)$
                Set $x_t = x_{t-1}$
            else
                Determine an $h^* \in [x_t^F + 1, x_{t-1} - 1]$
                such that $g(h^*) \geq 0$ and $g(h^* + 1) < 0$
                Set $x_t = h^*$ }
        else
            Set $x_t = x_t^F$. }
    else {
        if $(g(x_t^F) \geq 0)$
            Set $x_t = x_t^F$
        else if $(g(x_{t-1}) \leq 0)$
            Set $x_t = x_{t-1}$
        else
            Determine an $h^* \in [x_{t-1} + 1, x_t^F - 1]$ such
            that $g(h^*) \geq 0$ and $g(h^* + 1) < 0$
            Set $x_t = h^*$ } }
```

Fig. 10. The WC-online algorithm.

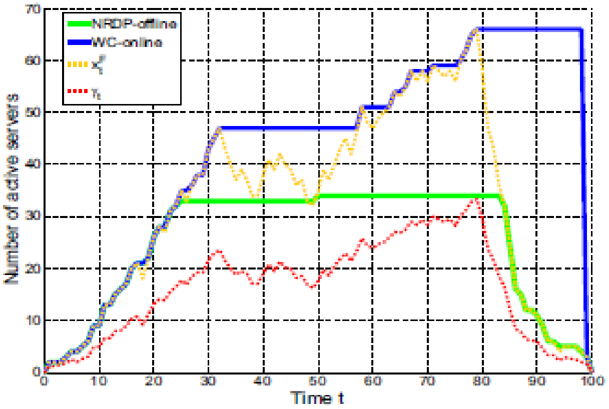
the convexity of $F(\cdot)$, it is clear that operating cost at time t is decreased because we move x_t closer to x_{Ft} .

Numerical results for one simple network

The first case has a linear function $p(\gamma) = 2\gamma$ with zero switching cost, i.e., βon + βoff = 0. This is the special case discussed in Section III-A. The second case has a linear function $p(\gamma) = 2\gamma$ with fixed switching cost βon + βoff = 6. This is the special case discussed in Section III-B. The third case was defined in [11], i.e., $p(\gamma) = \gamma$ max { 1 1□ 1.5, 0 } + 1 and βon + βoff = 6. The forth case has p(γ) = γ3 and βon + βoff = 6.



(a) The first scenario



(b) The second scenario

Evaluation Measures

For the first case, the total costs of the offline optimal solution and the online solution are both 13258.8, which means the performance ratio is 1. _ For the second case, the total cost of the offline optimal solution is 13666.8 while the total cost of the online solution is 14050.8. So the performance ratio of the online solution is 14050:8 13666:8 = 1.028. _ For the third case, the total cost of the offline optimal solution is 14976.8 while the total cost of the online solution is 17748.5. So the performance ratio of the online solution is 17748:5 14976:8 = 1.185. _ For the fourth case, the total cost of the offline optimal solution is 11080.4 while the total cost of the online solution is 11389.8. So the performance ratio of the online solution is 11389:8 11080:4 = 1.028.

Numerical results for more networks

Results for 100 randomly generated workload networks. For each network, we apply NRDP-offline algorithm to obtain an optimal solution with the minimum total cost and apply WC-online algorithm to obtain a feasible solution with certain total cost. We still use the four different cost functions and calculate the performance ratio achieved by our online algorithm. We find for the first case, the performance ratios are always 1, and for the other three cases, the performance ratios are almost the same (all within [1.0, 1.2]). Thus, results for the third case in Fig. 12 are shown. Results for other three cases are similar and thus omitted. We can see that our WC-online algorithm has the average performance ratio 1.151.

Note that we can use x^F_t as an online solution. However, the average performance ratio by x^F_t is 2.197 since such solutions only minimizes the operating cost. Thus, it is important to consider both operating cost and switching cost in an online algorithm such that the total cost can be close to the minimum total cost achieved by the optimal offline solution.

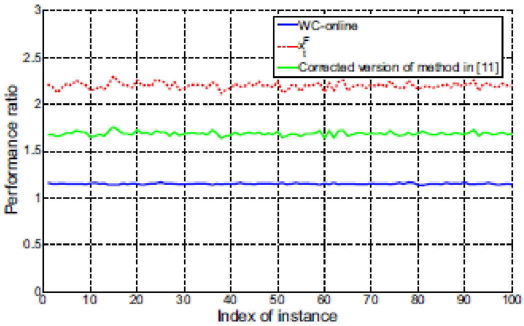


Fig. 12. Performance ratio results for 100 randomly generated workload network with the third scenario.

V.CONCLUSION

We considered information midpoint organization problem to minimize energy cost and to improve value of service by adjusting the number of active servers and distributing workload to these active servers. We first formulated this problem as a cost minimization problem, with the consideration on energy cost, delay cost, and switching cost. Then after analyzed some properties of optimal solution, both online and offline algorithms are designed. To the offline solution, we designed a dynamic programming based algorithm (DP-offline algorithm) and further revised the solution procedure to have an optimal offline algorithm with a polynomial complexity (NRDP-offline algorithm). To the online solution, we designed the WC-online algorithm by considering the worst case scenario and optimizing the performance for this case. The WC-online algorithm is shown to be able to achieve near-optimal performance in simulation.

VI. REFERENCES

[1] M.A. Adnan, R. Sugihara, R. Gupta, Energy efficient geographical load balancing via dynamic deferral of workload, in: Proc. of CLOUD’12, IEEE, 2012.

[2] M.A. Adnan, R. Sugihara, Y. Ma, R. Gupta, Energy-optimized dynamic deferral of workload for capacity provisioning in data centers, in: Proc. of IGCC’13, IEEE, 2013.

[3] “Data Center Locations,” http://www.google.com/about/data centers/inside/locations/index.html.

[4] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu, “No Power Struggles: Coordinated Multi-level Power Management for the Data Center,” in Proceeding of the 13th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS). ACM, 2008, pp. 48–59

[5] I. Marshall and C. Roadknight, “Linking cache performance to user behaviour,” Computer Networks and ISDN System, vol. 30, no. 223, pp. 2123 – 2130, 1998.

[6] H. Jin, T. Cheocherngngarn, D. Levy, A. Smith, D. Pan, J. Liu, and N. Pissinou, “ Joint Host-Network Optimization for Energy Efficient Data Center Networking,” in Proceeding of the 27th International Symposium on Parallel Distributed Processing (IPDPS), 2013, pp. 623–634.

[7] D. Niu, C. Feng, B. Li, Pricing cloud bandwidth reservation under demand uncertainty, in: Proc. of SIGMETRICS’12, ACM, 2012. H. Qian, D. Medhi, Server operational of cost optimization for cloud computing service providers over a time horizon, in: Proc. of HOTICE’11, ACM, 2011.

[8] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, B. Maggs, Cutting the electric bill for Internet-scale system, in: Proc. of SIGCOMM’09, ACM, 2009.

[9] L. Rao, X. Liu, M.D. Ilic, J. Liu, Distributed coordination of Internet data center under multiregional electricity markets, Proc. IEEE 100 (2011) 269–282.

[10] L. Rao, X. Liu, L. Xie, W. Liu, Minimizing electricity cost: optimization of distributed Internet data center in a multi-electricity-market environment, in: Proc. of INFOCOM’10, IEEE, 2010.

[11] L. Rao, X. Liu, L. Xie, W. Liu, Coordinated energy cost management of distributed Internet data center in smart grid, IEEE Trans. Smart Grid 3 (2012) 50–58.

[12] S. Ren, Y. He, F. Xu, Provably-efficient job scheduling for energy and fairness in geographically distributed data center, in: Proc. of ICDCS’12, IEEE, 2012.

[13] N. Tziritas, S.U. Khan, C.Z. Xu, T. Loukopoulos, S. Lalis, On minimizing the resource consumption of cloud applications using process migrations, J. Parall. Distrib. Comput. 73 (12) (2013) 1690–1704. Elsevier.

[14] Z. Xu, W. Liang, Minimizing the operational cost of data centers via geographical electricity price diversity, in: Proc. of CLOUD’13, IEEE, 2013.