Don't Fire Me, A Kernel Autoregressive Hybrid Model For Optimal Layoff Plan

Zhiling Luo
College of Computer Science
Zhejiang University
Hangzhou, China
Email: luozhiling@zju.edu.cn

Ying Li
College of Computer Science
Zhejiang University
Hangzhou, China
Email: cnliying@zju.edu.cn

Ruisheng Fu
College of Computer Science
Zhejiang University
Hangzhou, China
Email: 190949008@qq.com

Jianwei Yin
College of Computer Science
Zhejiang University
Hangzhou, China
Email: zjuyjw@cs.zju.edu.cn

***Abstract*—Job cutting occurs when a modern service enterprise reduces the employing labour cost by firing some staffs. Making an appropriate layoff plan is always quite difficult since a bad job cutting has a serious impact on not only the organization but also the business process executing efficiency. Therefore, in this paper, we address the problem of making an optimal layoff plan with the least influence on the executing of the business process. The key challenge is estimating the process throughput under a layoff plan. We overcome this challenge by two steps: regressing the activity throughput by the stuff number and inferring process throughput by the maximum flow or minimum cut algorithm on the Directed Acyclic Graph of process. In the regressing step, a kernel autoregressive hybrid model is proposed, whose MSE is 30% lower than SVM. After that, an augmenting path based algorithm is introduced to make an optimal layoff plan. To evaluate the accuracy of our model, we conduct an external experiment on a real dataset from the workflow system employed in the government of Hangzhou City in China, which results in 9750969 logs from 2050 activities and 16295 employees in two years.**

***Keywords*-Machine Learning; Layoff; Autoregressive Model; Kernel Method**

I. Introduction

Layoff, a common phenomena occurred in both modern service enterprises and government departments to reduce the labour resource cost, is accompanied by giant changes and some potential influences on this organization [15]. Since both the importance and influence on the business process vary from one staff to another, making decisions about who should be fired is really tough, from the perspective of the manager. An awful layoff plan will decrease the efficiency of the organization and even shut the business down. Therefore how to build an optimal layoff plan is becoming an interesting topic for both the managers and researchers.

To specify our discussion, we illustrate a layoff plan by following three questions:

- Q1: **Which** positions are redundant in staffs?
- Q2: **How many** staffs need to be laid off on each position?
- Q3: **Who** should be fired?

Q1 can be answered qualitatively by theories of management science. Specifically, human resource management, a major branch in the management science, derives some basic principles and useful tools [13] on layoff. These theories, built on the experience of managers, are either qualitative or grain-coarse [4], [12] and out-perform poorly in quantitatively analysing.

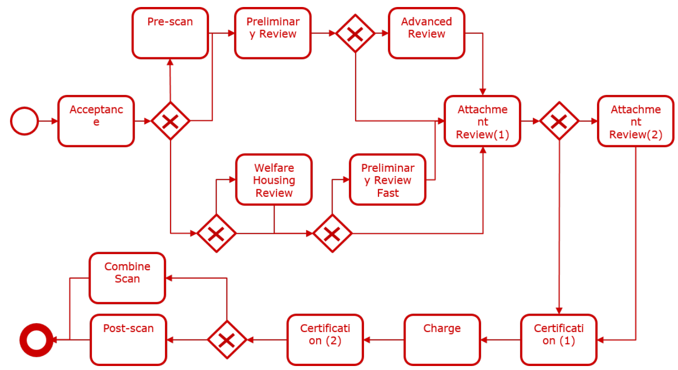

Figure 1: Real Estate Transactions Approval Process in BPMN diagram of Hangzhou Government, China. This diagram is in the format of BPMN, in which activity is presented by the rectangle and gateway is presented by the prism. The circular is start node and the bound circular is the end node. There are 13 activities and 6 gateways in RetAP.

To answer Q2, we need step further and construct a concreted and quantitative theory. In this work, we introduce Workflow Management System(WfMS) as the theory foundation. Devised decades ago, Workflow Management System [19] has experienced a rapid development and now is widely used in enterprises and government departments to support the business process management. WfMS provides not only a process management tool but also more possibilities with quantitative resource management techniques [17]. Specially, WfMS records all the staff operations in its event log supporting deep analysis on the process [19]. Therefore, we employ WfMS and its log helping to find out the potential redundant position.

As for Q3, the personal effectiveness and other non-job factors are involved. It is beyond the scale of this paper and we leave this part in near future work. In this work, we ignore the different performance between different staffs.

Therefore, in this paper, we study the layoff problem on the foundation of WfMS and introduce a model based on the event logs. To evaluate the influence of layoff, we employ the **throughput** or **frame rate** [9], namely the completed instances number of an activity per unit time. Further, the process throughput, the achieved process instances per unit time, can be defined and calculated by the activity throughput. To guarantee the regular running, the different between the process throughput before and after layoff should be as small as possible. The formalization of this problem is an optimization problem, as we explained in section III-B. Its complexity comes from two challenges. For better understanding the challenges, we have extracted a Real Estate Transactions Approval Process (RetAP), illustrated by Fig. 1, charged by the Real Estate Department of Hangzhou Government, China.

- **Given the staff number, estimating the activity throughput is difficult.** There are three important facts. First, although intuitively the activity throughput and staff number have the positive correlation, it is not a linear relation.Second, we witness that the activity throughput in different day sharing the same staff number. The last but not least, the activity throughput is time sequence related. In other words, the activity throughput of day t is not related to the staff number, but also influenced by the activity throughput at $t-1$, $t-2$ and so on.
- **Given the activity throughput, inferring the process throughput is another challenge.** This challenge comes from two facts. On one hand, although activity throughput is studied in previous literature, the definition and calculation of process throughput is rarely discussed before. Therefore, introducing a reasonable definition itself is not an easy task. On the other hand, process structure has a important influence on its throughput. In other words, in estimating the process throughput, we should take not only the activity throughput but also the process structure into consideration.

To overcome the first challenge, referring the **AutoRegress Model** (AR) or (ARM) [1], we introduce an autoregressive model to present the activity throughput and append the staff number by a non linear kernel. In this way, we build a autoregressive and kernel hybrid model to estimate activity throughput. To overcome the second challenge, we convert the process structure to a Directed Acyclic Graph and employ a promoted maximum flow/ minimum cut algorithm to infer the process throughput.

Contribution: Our major contributions in this paper are summarized as follows.

- A kernel autoregressive hybrid model to predict the activity throughput given the staff number. Comparing to other process mining techniques related to the labour resource, our model supports presenting the relation of self regression.
- A metric indicator, process throughput, to evaluate the executing efficiency. Its calculation can be inferred by applying maximum flow or minimum cut algorithm on Directed Acyclic Graph (DAG) of this process.
- An augmenting path based algorithm helps to select the optimal layoff plan. This algorithm's worst time complexity is $O(V^2E^2)$, which is acceptable for most practical environment.

The rest of this paper is organized as follows: Section 2 describes the source of our dataset and its basic static characteristics. Section 3 introduces our model. Section 4 discusses the evaluation on our approaches. Finally, related works and conclusion are respectively given in section 5 and 6.

II. Working Log Data and Characteristics

The dataset used in this paper, contains more than 7 million (7,650,969) event logs, and is extracted from the WfMS of Hangzhou Government, China, which spans from May. 2013 to Apr. 2015. It consists of 666 processes, 1,129,509 process instances, 12,836 activities, 7,650,969 activities instances and 3995 staffs. Figure 1 is an example process in our dataset and it contains 13 activities. A process instance is a concreted case of a process and an activity instance is a case of an activity [21]. A data fragment is presented in Fig. 2. An event log, recording the life cycle of an activity instance, contains following attributes:

- *InstActivityId*: the identity number of this activity instance,
- *DefActivityId*: the identity number of the activity,
- *InstProcessId*: the identity number of the process instance that it belongs to,
- *DefProcessId*: the identity number of the process that it belongs to,
- *StaffId*: the identity number of the staff who take responsibility of this activity,
- *Signin*: the timestamp that this activity instance created and
- *Complete*: the timestamp that this activity instance completed.

III. Model

We first introduce the key features in layoff in Section III-A, and then formalize the layoff problem in Section III-B. After that, the kernel autoregressive hybrid model is introduced in Section III-C. The inference algorithm and the predicting method are presented in Section III-D and Section III-E. Finally, we present the optimization approach in Section III-F. All the notations used in this paper are listed in Table I.

InstActivityId	DefActivityId	InstProcessId	DefProcessId	StaffId	Signin	Complete
2c90e4db43672ba901436f9075025983	actif1e7b7c88770	2c90e4db43672ba901436b8a77792f31	defpfa0a22cc8670	2c918a053463e2e7013463eeb53902ef	2014/9/10 15:10:03	2014/9/10 15:30:01
2c90e4da451670a40145205a7a466b24	actif1e7b7c88770	2c90e4da451670a40145205a79db6b23	defpfa0a22cc8670	20130511025034usrKFS100184	2014/9/10 15:12:35	2014/9/10 16:10:03
2c90e4dc45166f5b01451c30f67363cf	actif3bd1374877e	2c90e4dc45166f5b01451c30f61963cc	defpfa0a22cc8670	2c90e4da3e4b39fc013e4e097e24026f	2014/9/10 15:13:40	2014/9/10 15:19:12
2c90e4dc45183d7001451c01412f29f7	actif1e7b7c88770	2c90e4dc44f254d30144f2d9be520975	defpfa0a22cc8670	2c90e4db3b3fba44013b40a964d0127b	2014/9/10 15:13:42	2014/9/10 15:32:23

Figure 2: The dataset fragment consists of 4 activity instances from 2 activities in RetAP.

Symbol	Description
A,S	activity throughput, staff number
P	process
N	activity number
C	layoff constraints
t,Δt, T	time, time difference, max time
$A_i^{(t)}$	throughput of activity i at time t
$S_i^{(t)}$	staff number of activity i at time t
$\mathcal{A}$	the throughput schema
$\mathcal{S}$	the staff schema
$\mathcal{P}$	the process throughput
v_{ij}	the transition probability from activity j to activity i
$i \leftarrow j$	activity i starts after j
$\hat{.}$	the value after arrangement
$.^*$	the predicted value

Table I: Symbols used in Section III.

A. Features

This part introduces three basic features: activity throughput, staff number and process throughput. A process P, e.g. RetAP in Section 1, is a set of activities arranged in a determined order, according the concept definitions in workflow system. In following discussion, we use the indicator $i \in [1, N]$ to denote an activity in the process P. And N is the number of activities. For RetAP, N is equal to 13 as illustrated by Fig. 1.

Definition III.1 ***activity throughput*** $A_i^{(t)}$ *is the instance number of activity i completed per unit time Δt at time t [9].*

Activity throughput $A_i^{(t)}$ is extracted from the logs described in above section. To simplify the calculation, we set unit time Δt equals one day. And the coarse of time is chosen as a day, namely $t \in [1, T]$ is t'st day. For example, $t = 2$ means the second day in all logs.

Definition III.2 *the staff number $S_i^{(t)}$ means the staff number at time t of the activity i.*

To simplify our description, we introduce the schema $\mathcal{S} = (s_1, s_2, ..., s_N)'$ to represent the staff allocation for a process. It means allocation s_i staffs for activity i. And we use $\mathcal{A} = (a_1, a_2, ..., a_N)'$ to represent the activity throughput, where

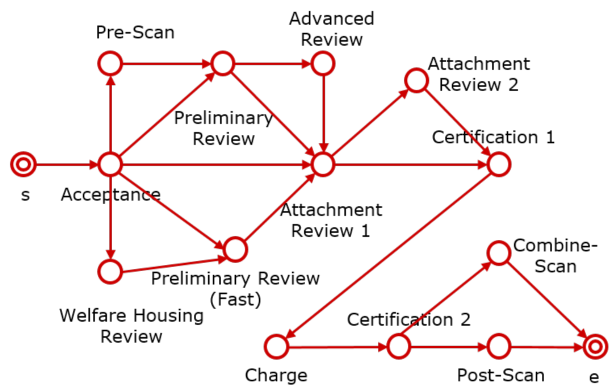

Figure 3: The DAG diagram of RetAP contains 15 nodes (13 normal nodes, 1 start node and 1 end node) and 21 directed edges.

activity i has a_i throughput.

Definition III.3 *The process throughput $\mathcal{P}^{(t)}$ is the process instance number completed per unit time Δt at time t.*

According to the buckets effect, namely the bucket's capacity is limited by the shortest board, the process throughput is exactly determined by least throughput activity. From the perspective of network, the process throughput is the maximum flow in the network composed by activities. At first, we employ the **Directed Acyclic Graph (DAG)** introduced by Yu and Rajkumar [24] to describe the executing order of the process. Each node represents an activity and a directed arc between two nodes means the one activity cannot be executed until another activity is completed. In some researches, DAG in process is also called **Activity On Vertex (AOV)** network [18]. Figure 3 is the DAG diagram of the example process mentioned in Section 1. The directed arc from 'Acceptance' to 'Pre Scan' means the latter is executed followed the former. We use the dependency relation $i \leftarrow j$ to denote the directed arc from activity j to i. With this definition, j is called the **predecessor activity** of i. Witnessing the difference of the conditional probability, existing the dependency relation $i \leftarrow j$ between given i and given j, we introduce $\mathbf{v}$ where

$$v_{ij} = Pr(i \leftarrow j|j). \tag{1}$$

The physical meaning of $\mathbf{v}$ is the probability of an in-

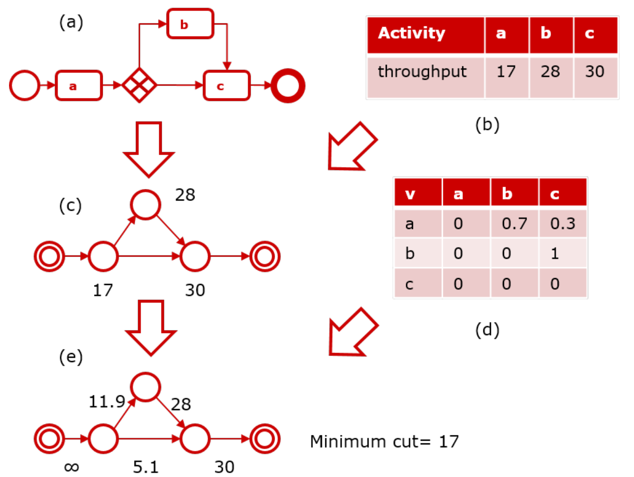

Figure 4: Inferring process throughput procedure. (a) is an original BPMN process diagram with 3 activities and 1 gateway. (b) is the activity throughput. (c) is the DAG with activity throughput marked. (d) is the transition probability $\mathbf{v}$. (e) is the DAG with edge weight marked. The minimum cut is presented by the dash line in (e).

stance of activity i comes from activity j given the instances of j. To simplify our model, we use the statical variable to represent $\mathbf{v}$, namely v_{ij} is determined by dividing the number of instances of i which come from j by the number of all j instances. An interesting fact is that once we employ the **topological sort** on activities, $\mathbf{v}$ is upper triangular matrixes, which is guaranteed by DAG.

Then, we introduce the weight $E(i, j)$ on edges $< A_i, A_j >$ in DAG by

$$E(i, j) = A_i * v_{ij}. \tag{2}$$

In DAG the weights of edges from start vertex are infinite. At last, we get the calculation of process throughput:

Proposition III.1 *The process throughput is exactly the* ***maximum flow*** *or* ***minimum cut****, of corresponding DAG from start vertex to end vertex.*

For better understanding this procedure, we introduce a simple case in Fig. 4. The DAG, illustrated by (c), is extracted from the original BPMN diagram in (a). Then the activity throughput is appended on each node in (c). With the help of transition probability v in (d), we can mark the edge with by Eq. 2 in (e). Intuitively the minimum cut is the arc marked in (e). So we get the process throughput as 17.

B. Problem Description

Thanks to the feature definitions and notations, we can now introduce the problem description in formalization.

Definition III.4 *Given a process P, the original staff schema $\mathcal{S}$ and constraint C, the goal of the layoff arrangement is to find a new staff schema $\hat{\mathcal{S}}$ satisfy that $\|\mathcal{S} - \hat{\mathcal{S}}\| = C$ to minimize the process throughput difference $\|\mathcal{P} - \hat{\mathcal{P}}\|$.*

In this formalized definition, $\|.\|$ is the ℓ_1 norm. Our goal is to find out the optimal staff schema with least process throughput reduction and at the same time satisfying layoff constraint.

C. Kernel Autoregressive Hybrid Model

Noticing that both $A^{(t)}$ and $S^{(t)}$ are not stationary random process, we apply one order difference on $A^{(t)}$ and $S^{(t)}$, namely first gradient. Figure 5 illustrates both the original activity throughput of 'Acceptance' and the first gradient. In fact we can repeat this trick r times, and get the r-st gradient. However the promotion of using high-order difference is inapparent in our dataset, therefore we set $r = 1$. Let's consider our observation on an activity denoted by $\{s_t, a_t\}_{t=1}^{T}$, here s_t is the first gradient, equaling to $S_i^{(t)} - S_i^{(t-1)}$ and a_t is equal to $A_i^{(t)} - A_i^{(t-1)}$, we consider the autoregressive model,

$$\Phi^p(B)a_t = e_t + \omega'\phi(s_t), t = 1, 2, ..., T, \tag{3}$$

where $\Phi^p(B)$ is a polynomial in back shift operator B with parameters $\rho_i, i = 1, ..., p$, such that

$$\Phi^p(B)a_t = a_t - \rho_1 a_{t-1} - \rho_2 a_{t-2} - ... - \rho_p a_{t-p}, \tag{4}$$

and e_i, called white noise, is assumed to follow independently normal distribution, namely $e_i \sim \mathcal{N}(0, \sigma)$. $\phi(s)$ is the kernel, often $K(s_i, s_j) = \phi(s_i)'\phi(s_j)$. In layoff problem, we select RBF (Radial Basis Function) kernel,

$$K(s_i, s_j) = \exp(-\frac{||s_i - s_j||_2^2}{2\lambda^2}). \tag{5}$$

Therefore, we name our model as RBF-AR(p).

D. Parameter Estimation

The estimate of parameter ω in Eq. 3, is obtained by minimizing the regularized negative log likelihood,

$$\begin{aligned} L(\omega) &= \sum_{t=1}^{p}(\Phi^t(B)a_t - \omega'\phi(s_t))^2 \\ &+ \sum_{t=p+1}^{T}(\Phi^p(B)a_t - \omega'\phi(s_t))^2 + \lambda||\omega||^2 \end{aligned} \tag{6}$$

where λ is a nonnegative constant which controls the trade-off between the goodness-of-fit on the data and $||\omega||^2$. Literature [11] guarantees that the minimizer of the regularized negative log likelihood to be $\phi(s)\omega = K\alpha$ for some vector α. Therefore Eq. 6 becomes obtaining α to minimize

$$L(\alpha) = (B_\rho\mathbf{a} - K\alpha)'(B_\rho\mathbf{a} - K\alpha) + \lambda\alpha'K\alpha \tag{7}$$

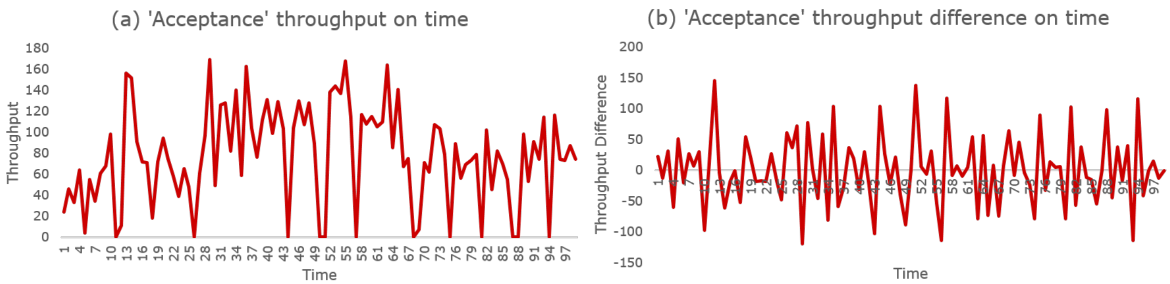

Figure 5: (a) is the 'Acceptance' activity throughput on time and (b) is its first gradient. Noticing that latter process is more stable than the former, we use the first gradient in follow discussion instead of original data.

where

$$B_\rho = \begin{pmatrix} 1 & 0 & 0 & ... & ... & 0 \\ -\rho_1 & 1 & 0 & ... & ... & 0 \\ 0 & -\rho_1 & 1 & 0 & ... & 0 \\ & & ... & & & \\ 0 & 0 & ... & 0 & -\rho_1 & 1 \end{pmatrix} \text{ for RBF-AR(1)}, \tag{8}$$

$$B_\rho = \begin{pmatrix} 1 & 0 & 0 & ... & ... & 0 \\ -\rho_1 & 1 & 0 & ... & ... & 0 \\ -\rho_2 & -\rho_1 & 1 & 0 & ... & 0 \\ & & ... & & & \\ 0 & 0 & ... & -\rho_2 & -\rho_1 & 1 \end{pmatrix} \text{ for RBF-AR(2)}. \tag{9}$$

and $K_{ij} = K(s_i, s_j)$. The estimation value of α for the mean function can be found as

$$\alpha^* = (KK + \lambda K)^{-1} K B_\rho \mathbf{a}, \tag{10}$$

where $\mathbf{a} = (a_1, a_2, ...a_T)'$.

E. Prediction

Given s_t and $a_{t-1}, a_{t-2}, ...a_{t-p}$, the predicted value of a_t is obtained as

$$a_t^* = \rho_1 a_{t-1} + \rho_2 a_{t-2} + ...\rho_p a_{t-p} + K(s_t, s)\alpha^*, \tag{11}$$

where s is the vector of s_i in parameter estimation. Noticing that a_t^* is the first gradient, we need add it back to the previous activity throughput

$$A_i^{(t)*} = A_i^{(t-1)} + a_t^*. \tag{12}$$

In predicting the process throughput, we can use various maximum flow algorithms, e.g. Ford-Fulkerson algorithm. And we employ Edmonds-Karp algorithm [7] to calculate the maximum flow on DAG and its time complexity is $\mathcal{O}(VE^2)$, where V is the number of vertex, namely activities and E is the edge number. In essential, after trained, our model provides a function from staff schema $\mathcal{S}$ to the process throughput $\mathcal{P}$. Therefore, we can use $\mathbf{P}(\mathcal{S})$ to denote this function.

F. Optimization

To solve problem III.4, we convert it as an optimization problem.

$$\begin{aligned} &minimize_{\hat{\mathcal{S}}} \mathbf{P}(\mathcal{S}) - \mathbf{P}(\hat{\mathcal{S}}) \\ &s.t. \|\mathcal{S} - \hat{\mathcal{S}}\| = C \end{aligned} \tag{13}$$

To solve problem 13, we propose an approximation algorithm which follows steps and is illustrated by algorithm 1:

- **Step 1**: Selecting an activity i from the inverse topological order, whose staff number is larger than 1. If no activity satisfy this condition, stoping this algorithm, which means unsatisfied. If existing two or more, picking one randomly. Decreasing its staff number by one, namely $S_i = S_i - 1$. Updating the constraints $C = C - 1$. Updating its throughput as A_i^* by RBF-AR(p) and its connecting edge weights.
- **Step 2**: Using breadth-first search to finding the augmenting path from start node to end node. Decreasing each edge weights on the augmenting path by the least weight of edges in this path.
- **Step 3**: Repeating step 2 until no augmenting path can be found. Finding out those activities whose connecting out edge weight is larger than 0.
- **Step 4**: Initialize counter B as 1 and store a copy of current staff number $\mathcal{S}$ as $\mathcal{S}bak$. Iterating those activities in topological order and executing step 5.
- **Step 5**: Decreasing the selected activity's staff number by 1. Updating its activity throughput by RBF-AR(p). If in this condition, the edge weight is less or equal than 0, switching to another activities found in step 4. Growing the counter B by 1. If B is now equal to C, then outputs the current staff number as $\hat{\mathcal{S}}$
- **Step 6**: Restoring the staff number $\mathcal{S}$ from $\mathcal{S}bak$ and repeating step 1.

Figure 6 illustrates a simple example of applying algorithm 1. In this case, the process consists of three activities (a,b,c). Original staff allocation schema is $\mathcal{S} = (5, 6, 5)'$, the transition probability $\mathbf{v}$ and constraint $C = 3$. The process

Algorithm 1 Layoff Arrangement Algorithm

Input: $\mathcal{S}$: Current staff schema. C: Constraints
Output: $\hat{\mathcal{S}}$: Arranged staff schema

```
1: while i = findPossibleLastActivity(S) is not null do
2:     S_i = S_i − 1;
3:     C = C − 1;
4:     DAG = constructDAG(S);
5:     while p = findAugmentingPath(DAG) is not null do
6:         cap = findMinWeight(p);
7:         DAG = updateWeightDAG(DAG, p, cap);
8:     end while
9:     B = 0;
10:    Sbak = S;
11:    while j = findExistingActivity(DAG) is not null do
12:        for k = 0 to S_j do
13:            if PredictA(S_j) − PredictA(S_j − k) < getWeight(DAG, j) then
14:                B = B + k − 1;
15:                S_j = S_j − k;
16:                if B > C then
17:                    Ŝ = S
18:                    return Ŝ
19:                end if
20:            end if
21:        end for
22:    end while
23:    S = Sbak;
24: end while
25: return Unsatisfied
```

throughput, as we discussed, is the minimum cut, which is equal to 17. In step 1, we sort activities in topological order, namely (a,b,c). We select the last activity with staff number larger than 1, which is c in this case. Then we set staff number in c to 4, constraint C=2 and update the DAG including activity throughput. In step 2&3, we infer the residue graph and witness that activity b and c are both redundant. In step 4,5&6, we try to reduce the staff number in b and c with guaranteing the edge weights in residue graph are both larger or equal than 0. We get the new staff allocation schema $\hat{\mathcal{S}} = (5, 5, 3)'$ and the process throughput is still 17. Finally, $\hat{\mathcal{S}}$ is optimal layoff plan.

In Alg. 1, the iterations on line 1 are V in the worst situation. We compute activity throughput V times, by applying Eq. 12 at each activity, on line 4, wasting $O(V)$. The time complexity from line 5 to line 7 is $O(VE^2)$ according to Edmonds-Karp algorithm The iterations from line 11 to line 22 are $O(V)$. Thus, the whole time complexity is $O(V^2E^2)$ for alg. 1.

IV. Experiment

In this section, we report the evaluation of RBF-AR(p) on the accuracy. Our goal here is to predict the activity throughput based on historic records and staff number. We set 70% activity throughput as the train data and left 30% as test data. And we run our experiment on a 3.3GHz Intel Core i5 processor and 8G RAM under Windows 8. All algorithms are implemented in R language.

A. Baseline Methods

We compare our model, RBF-AR(p) with two methods.

- **SVM**: Supported Vector Machine uses the staff number as the attribute to train a regression model and apply it to predict the activity throughput in the test data. We also select the RBF as the kernel in SVM in this experiment.
- **AR**: Autoregressive Model uses the previous activity throughput to train the autoregressive parameters and apply them to predict the next activity throughput in the test data.

The parameter p donates the order of autoregressive coefficient. It is influenced by the data characteristics and there are plenty of researches on selecting the optimal p in autoregressive model [2]. We omit the discussion on p because it is beyond the scale of this paper and simply set $p = 2$ for both AR and RBF-AR in activity throughput prediction.

B. Prediction Performance

We quantitatively evaluate the performance of inferring activity throughput in terms of Mean Square Error (MSE):

$$MSE = \frac{1}{N}\sum_{t=1}^{N}(A^{(t)} - A^{(t)*})^2, \tag{14}$$

where N is the size of test data. Figure 7 shows the results for activity throughput prediction. From Fig. 7, we see that our method clearly outperforms the baseline methods on both cases. RBF-AR(2) achieves a 26% reduction compared with SVM and AR(2) in terms of MSE. Figure 8 intuitively illustrates the comparison of SVM, AR(2) and RBF-AR(2) in terms of MSE. As depicted by Fig. 9, we make a comparison between the actual value and RBF-AR(2) predicted value of activity throughput in 'Acceptance' in RetAP. In Fig. 9, the light line is the actual value and the dark line is the predicted value. It is clear that our model can guarantee a high accuracy in activity throughput estimation.

V. Related Works

In this section, we make a simple review of researches on layoff from two perspectives of (a) resource in WfMS and (b) efficiency of WfMS.

As we discussed, from the perspective of human resource management, staff is a kind of resource in WfMS [4]. In WfMS, resource management attracts computer researchers' attention decades ago. Russell et. al. [17] introduced a group of resource pattern stand on Aalst's researches about workflow patterns [20]. Li et. al. [14] provides a workflow model and specification with resource constraint considered. Liu et. al. [23] took the problem of work item sharing in resource management by applying association rule mining techniques to workflow event log. A resource and control flow integrated model, called SPDL, was introduced in [22] Gao et. al. [8] proposed a resource allocation approach based on polychromatic sets theory. A method to define resource

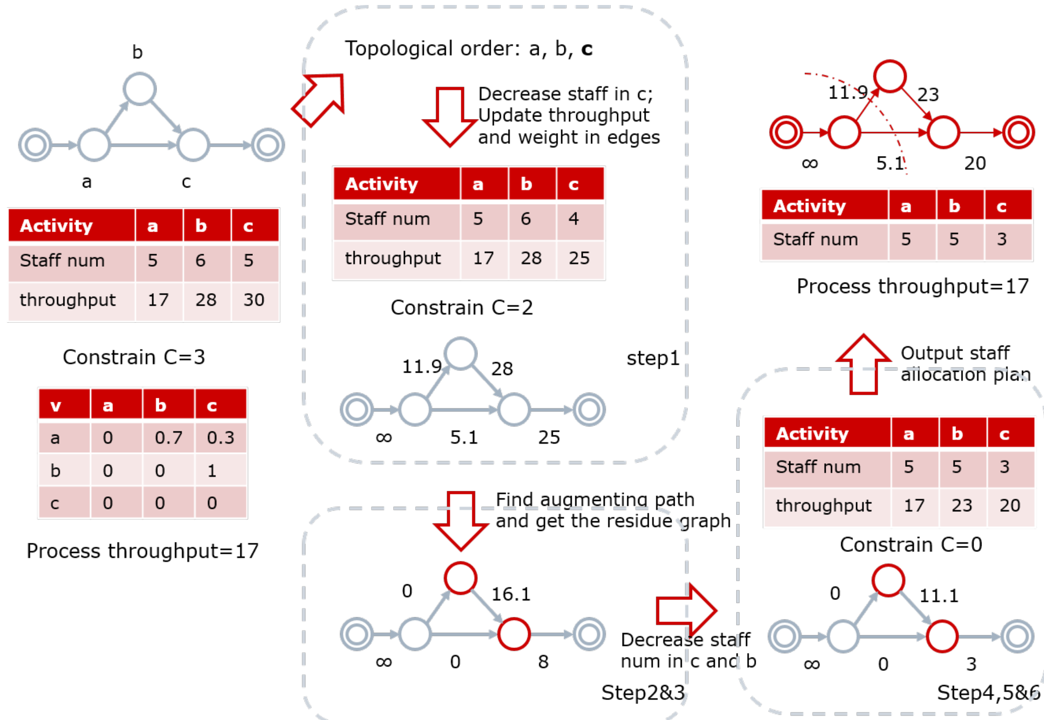

Figure 6: The simple case of algorithm 1

	Acceptance	Pre-Scan	Advanced Review	Attachment Review 1	Attachment Review 2	Certification 1	Charge	Certification 2	Combine-Scan
SVM	1293.91	1079.99	32.19	2449.60	2373.31	1808.30	3306.32	3355.97	1810.69
AR(2)	1165.34	779.20	38.49	1699.89	1625.54	1319.61	3059.31	4452.98	1626.43
RBF-AR(2)	**593.84**	**540.99**	**28.85**	**1165.40**	**1024.90**	**898.56**	**2756.60**	**4405.70**	**1284.70**

Figure 7: Activity throughput prediction performance of different methods.

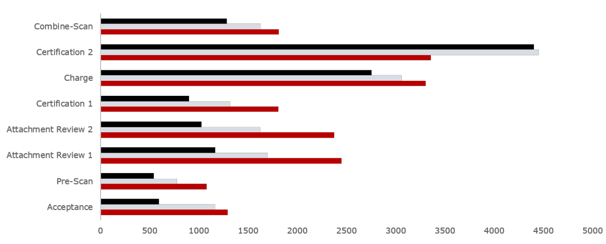

Figure 8: MSE comparison of different methods

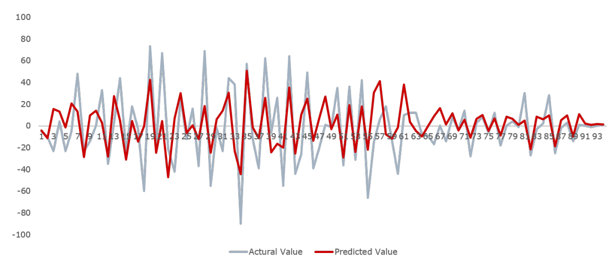

Figure 9: Actual value and Predicted value of activity throughput in 'Acceptance' in RetAP

assignment is introduced in [3]. These models, however, both concentrate on specification and designing of resource instead of concreted allocation approach. Our model, on the contrary, can provide a practical and executable staff allocation plan.

Efficiency of WfMS is another interesting and challenging topic. Usually, researchers employ process time to identify the efficiency. Eder et. al. [6] discussed the time management, especially time constraints management in workflow system. Pang et. al. [16] made applications of time workflow model using Timing Constraint Petri Nets. An activity running duration predicting approach was proposed in [5]. Jin et. al. [10] introduced a technique appending server in workflow system to guarantee time constraints. In these researches, time and workflow efficiency are considered as the optimal goal. We, on the contrary, set the efficiency, namely process throughput as the constraint condition to reduce staff number.

VI. Conclusions

In this paper, we study and formalize the problem of layoff in enterprise and government department. Activity throughput and process throughput are introduced as the constraint condition. The goal is to find out the optimal layoff plan to lay off staffs on some positions. At first, we propose a kernel autoregressive hybrid model, called RBF-AR(p), in predicting activity throughput by given staff number using historical records. Then, we introduce an algorithm based on DAG to allocate the staff in different activity to guarantee the least process throughput loss. At last, the accuracy of our model is evaluated by comparing with SVM and AR on the real data set of Hangzhou government, China. In summary, our model can provide an optimal layoff plan with least throughput loss, given the historical data. Since we can answer Q1 and Q2, mentioned above, to answer Q3, the future work will focus on determining the personal efficiency and firing those with least efficient.

ACKNOWLEDGMENTS

This work was supported by Zhejiang Provincial Natural Science Foundation of China under grant No.LY15F02007 and National Science and Technology Supporting Program of China (NO. 2015BAH18F02)

REFERENCES

[1] H. Akaike. Fitting autoregressive models for prediction. *Annals of The Institute of Statistical Mathematics*, 21(1):243–247, 1969.

[2] G. E. Box, G. M. Jenkins, and G. C. Reinsel. *Time series analysis: forecasting and control*, volume 734. John Wiley Sons, 2011.

[3] C. Cabanillas, M. Resinas, and A. Ruiz-Corts. Defining and analysing resource assignments in business processes with ral. In *Service-Oriented Computing*, Service-Oriented Computing, pages 477–486. Springer, 2011.

[4] G. Dessler. *Human resource management*. Pearson Higher Ed, 2014.

[5] P. A. Dinda. Online prediction of the running time of tasks. pages 383–394, 2001.

[6] J. Eder, E. Panagos, H. Pozewaunig, and M. Rabinovich. *Time management in workflow systems*. Springer, 1999.

[7] J. Edmonds and R. M. Karp. *Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems*. Combinatorial Optimization. 2001.

[8] X. Gao, L. Xu, X. Wang, Y. Li, M. Yang, and Y. Liu. Workflow process modelling and resource allocation based on polychromatic sets theory. *Enterprise Information Systems*, 7(2):198–226, 2013.

[9] Y. Gu and Q. Wu. Maximizing workflow throughput for streaming applications in distributed environments. In *Computer Communications and Networks (ICCCN), 2010 Proceedings of 19th International Conference on*, Computer Communications and Networks (ICCCN), 2010 Proceedings of 19th International Conference on, pages 1–6. IEEE, 2010.

[10] J. Hyun Son and M. Ho Kim. Improving the performance of time-constrained workflow processing. *Journal of Systems and Software*, 58(3):211–219, 2001.

[11] G. KIMELDORF. Some results on tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33(1):82–95, 1971.

[12] M. J. Koch and R. G. McGrath. Improving labor productivity: Human resource management policies do matter. *Strategic management journal*, 17(5):335–354, 1996.

[13] H.-S. Lee, J.-H. Yu, and S.-K. Kim. Impact of labor factors on workflow. *Journal of construction engineering and management*, 130(6):918–923, 2004.

[14] H. Li, Y. Yang, and T. Y. Chen. Resource constraints analysis of workflow specifications. *Journal of Systems and Software*, 73(2):271–285, 2004.

[15] B. Low, E. Ostrom, C. Simon, and J. Wilson. Redundancy and diversity: do they influence optimal management. *Navigating social-ecological systems: Building resilience for complexity and change*, pages 83–114, 2003.

[16] H. Pang, H. Li, and X. Yang. Research and application of time workflow model based on timing constraint petri nets. *Procedia Engineering*, 15:1371–1376, 2011.

[17] N. Russell, W. M. van der Aalst, A. H. Ter Hofstede, and D. Edmond. Workflow resource patterns: Identification, representation and tool support. In *Advanced Information Systems Engineering*, Advanced Information Systems Engineering, pages 216–232. Springer, 2005.

[18] H. Shiizuka and M. Suzuki. Modeling of agv networks in flexible manufacturing systems. *Computers and industrial engineering*, 27(1):81–86, 1994.

[19] W. Van Der Aalst. *Process mining: discovery, conformance and enhancement of business processes*. Springer Science Business Media, 2011.

[20] W. M. Van Der Aalst, A. H. Ter Hofstede, B. Kiepuszewski, and A. P. Barros. Workflow patterns. *Distributed and parallel databases*, 14(1):5–51, 2003.

[21] S. A. White. Introduction to bpmn. *IBM Cooperation*, 2(0):0, 2004.

[22] J. Yin, Z. Luo, Y. Li, B. Fan, W. Liu, and Z. Wu. Towards a service pattern model supporting quantitative economic analysis, 2014 2014.

[23] L. Yingbo, Z. Li, and W. Jianmin. Mining workflow event log to facilitate parallel work item sharing among human resources. *International Journal of Computer Integrated Manufacturing*, 24(9):864–877, 2011.

[24] J. Yu and B. Rajkumar. Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms. *Scientific Programming*, 14(3-4):217 – 230, 2006.