International Journal of Engineering

Journal Homepage: www.ije.ir

Cycle Time Optimization of Processes Using an Entropy-Based Learning for Task Allocation

I. Firouzian*, M. Zahedi, H. Hassanpour

Computer Engineering & IT Department, Shahrood University of Technology, Shahrood, Iran

PAPER INFO

Paper history:
Received 04 May 2019
Received in revised form 16 June 201
Accepted 05 July 2019

Keywords:
Business Process
Entropy-based Optimization
Reinforcement Learning
Resource Allocation
Workflow Management

ABSTRACT

ıe optimization could be one of the great challenges in business process management. Although ıuch research on this subject, task similarities have been paid little attention. In this paper, a ·oach is proposed to optimize cycle time by minimizing entropy of work lists in resource ı while keeping workloads balanced. The idea of the entropy of work lists comes from the fact me it takes for a resource to do similar tasks in a rather consecutive order is less than the time do the same tasks separately. To this end, an entropy measurement is defined, which represents larities on some given work lists. Furthermore, workload balancing is also regarded as an because not only is cycle time optimization important, but also workload fairness should also ;xperimental results on a real-life event log of BPI challenge 2012 showed that the proposed ːads to 32% reduction in cycle time, compared with a reinforcement learning resource allocation ıvolving the entropy.

***doi**: 10.5829/ije.2019.32.08b.05*

1. INTRODUCTION

A business process is a collection of linked activities which find their end in the delivery of a service or a product to a client. Business Process Management (BPM) has emerged to provide a large set of tools and techniques to enact and manage these operational business processes [1]. One of the important issues in business process management systems is cycle time optimization which is highly relevant to process performance. Optimization of cycle time is performed based on process mining which aims to find resource allocation rules or patterns by mining event logs [2].

Cycle time optimization has a close relationship with resource allocation. In this paper, cycle time optimization is done by modifying resource allocation algorithms. There are many algorithms for resource allocation, such as the shortest work list allocation, the shortest processing time allocation and the shortest complete time allocation [3]. All these algorithms confront with some disadvantages or shortcomings, like load imbalance which damages the efficiency of a BPM performance [3]. In order to cope with this problem, workload balancing can also be considered an objective.

Workload balancing is used to dispense the workload dynamically and evenly across all resources available for a certain task to avoid circumstances where some resources are heavily loaded while others are a leisure activity or little work. Note that workload balancing does not guarantee the equality of workloads for all resources at any moment of time; but, it guarantees the whole workload will be dispensed fairly across all resources as much as possible. This is because resources have different specialties but they may have some specialties in common; hence, the resources are assigned to different activities but they may have some activities in common.

Since the resource allocation decisions in processes are a chain of decisions in an interactive environment, a particular work item must be assigned to a particular resource at a particular time. To this end, reinforcement learning is used to make appropriate decisions to allocate resources by trying to minimize long-term entropy, long-term cycle time and to improve the performance of business process execution [4].

*Corresponding Author Email: *iman.firouzian@gmail.com* (I. Firouzian)

In this paper, the idea of the entropy of work lists is introduced, which is based upon task similarities. As the number of similar tasks in a work list increases, the entropy of the work list decreases. It is inspired by the fact that similar tasks may have some subtasks in common, i.e., accessing locally close archive files for human or loading identical files into ram for a machine. Thus, minimizing the entropy of work lists associated with each resource leads to reduce the processing time of rather consecutive similar tasks. Eventually, minimizing the entropy of work lists leads to cycle time optimization.

Based on the discussion above, finding a trade-off point between cycle time and workload balancing is vital in business process management. So, we present an entropy based reinforcement learning for resource allocation through workload balancing, which would lead us to the trade-off point indirectly.

The present paper is organized as follows: section 2 introduces the related works on task allocation and reinforcement learning algorithm. Given the related works and their shortcomings and disadvantages, our new method is defined in section 3. Employing the proposed method experiments are conducted on a real-life event log of BPI challenge 2012 and the results are described in section 4. Finally, Conclusions and future works are discussed in section 5.

2. RELATED WORKS

One solution to the problem of cycle time reduction is closely related to resource allocation. Resource allocation deals with the problem of selecting the best resource for each activity in a business process. The aim of selecting the best resource is to satisfy criteria such as minimizing cost, reducing cycle time, and maximizing the quality of service and workload balancing of involved resources. The resource allocation plays an important role in cycle time reduction and workload balancing. In this section, we investigate the papers related to the subject of cycle time reduction, workload balancing, and reinforcement learning and entropy in the domain of BPM.

There are two main approaches to dealing with resource allocation presented by Kumar et al. [5]. These two basic resource allocation approaches are called push and pull modes.

In the push mode, tasks are pushed to the resources to be accomplished, while in the pull mode, resources are allowed to pull tasks from the task pool. The two modes are applicable when dealing with circumstances like resource shortage or overload [5, 6].

2. 1. Cycle Time Reduction as an Objective

There exist some papers which considered cycle time reduction as their objective [6-8]. Liu et al. [7] showed their technique reduces the cycle time by extracting social relation between resources involved with each trace as a parameter in reinforcement learning algorithm for resource allocation problem. They have examined their own algorithms on BPI Challenge 2012 dataset. They showed that considering social relations lead to cycle time reduction [6]. Muehlen et al. [7] have outlined the major aspects of resource management within workflow applications, following the workflow life cycle. They presented a generic resource Meta model for the initial specification of the resource structure and its population, which combines workflow-oriented with organization-oriented modeling concepts. Finally, they presented strategies for the use of workflow audit trail data to improve resource utilization and develop new process strategies. Xu et al. [9] presented a resource allocation method, which minimized process cost under the limited time constraint. A resource selection rule is presented in the study of Nisafani et al. [10] which considers factors such as workload, queue, working hours, inter-arrival time, and others that affect human resource performance. They used a Bayesian Network (BN) to incorporate those factors into a single model, which we have called the Bayesian Selection Rule (BSR). They showed that the BSR can reduce waiting time, completion time and cycle time. The study conducted by Wibisono et al. [11] proposes an on-the-fly performance-aware resource allocation to manage human resources in BPM. They utilize Naïve Bayes Model in the Naïve Bayes Selection Rule (NBSR) algorithm in order to select an appropriate resource to perform an incoming task.

Low et al. [11] also focused on the decision making on selection of improvement solutions as one of the obstacles hampering the success of process improvement. The paper presents the House of Improvement (HOI) model as a guideline to link decision criteria for the prioritisation of improvement solutions. Three phases in the HOI are applied to facilitate selection and to ensure that suitable and value-added solutions are chosen. Each phase includes procedures for identifying, evaluating, and analysing the elements by establishing a relationship matrix. Nematzadeh et al. [12] tries to evaluate performance and reliability of eflow and BPEL through mapping them to explicit colored petri net. To achieve this goal, colored petri net was enhanced with a new block of immediate transition, called Pick split/join. Then, a transformation table was proposed to show the mapping rules from basic and structured activities in eflow and BPEL to colored petri net. Finally, theory of probability was applied on the model to measure QoS. Mokhtari et al. [13] present a realistic variant of flowshop scheduling which integrates flow shop batch processing machines (FBPM) and preventive maintenance for minimizing the makespan. In order to tackle the given problem, we develop an electromagnetism-like (EM) algorithm, as a recent evolutionary technique, and proposed an enhanced EM algorithm, in which the EM is hybridized with a diversification mechanism, and an effective local search to enhance the efficiency of the algorithm. The study conducted by Kumar et al. [14] introduced a novel and more sophisticated mechanism to distribute work than the ones that are currently employed in workflow systems. Compared to existing workflow management systems that provide pure *push* or *pull* approaches, the mechanism allows on-the-fly balancing of quality and performance considerations. It merges work distribution and delegation into an integrated framework.

2. 2. Workload Balancing as an Objective There are some papers which considered workload balancing as their objective [6-8]. Although cycle time reduction is an important issue in BPM, some workload balancing mechanisms have emerged to resolve the limitation of resources' abilities [15, 16].

The papers presented by Ha et al. [16] showed that the balancing of resources' workloads makes it more resistant to the increased volume of customers' incoming requests [17, 18] Larbi et al. [15] presented a model to solve scheduling problem in identical parallel machines. They solved load balancing in parallel machines by mapping strategy to time Petri-net [15]. Ha et al. [16] presented a method for workload balancing of agents that perform the tasks in the business process. They used an analytic process model based on process specification and execution history data. Then, they used a queuing network built from the analytic process model in order to establish task assignment policies for the efficient execution of the business process. After that, as a practical use of the queuing network, workload balancing is presented to improve overall business process efficiency using a linear programming formula. Finally, a set of simulation experiments is conducted in order to validate the performance with respect to using shared work lists [16, 19].

2. 3. Trade-off between Cycle Time and Workload Balancing There is a trade-off between cycle time and workload balancing. Having workloads balanced, the system makes cycle time get farther away from its optimization point; having cycle time optimized, the system causes workloads to get imbalanced. Yu et al. [20] used process mining to explore the effect of workloads on service time of every resource that is known as "Yerkes-Dodson Law of Arousal". Thus, the need for a heuristic approach to cycle time reduction with a minimum side-effect on workloads seems essential. Some papers considered the relation between resources to reduce cycle time [3, 4, 21-23].

Bozkaya et al. [21] extracted process activities and their logical relations to allocate resources based on resource roles and their coordination. Some researchers highlighted the coordination among resources, ensuring the effective coordination among them. Aalst et al. [22], for instance, calculated parameters relevant to resource coordination level according to causality between activities. Huang et al. [24] calculated prior probabilities of activities between resources and made judgments of the correlation between resources based on what Aalst achieved. Liu and his colleagues [17] could provide a resource model based on the social network to increase cooperation between the resources by improving task allocation algorithm [4, 17]. Liu et al. [17] also presented a strategy for allocating the task to resources which supported workload balancing of resources and considered experiential value [4]. Senkul et al. [23] improved a priori algorithm and produced two types of resource allocation rules: the resource dependency rule and the activity allocation rule. The resource dependency reflected the relationship between resources, which connected the process activities regularly. In addition, by mining event logs to analyze resource allocation rules, they also got preliminary exploration in recommending resources to managers.

2. 4. Business Process Models Resource allocation problem in BPM is a chain of decision problems which is modeled as Markov Decision Processes (MDPs) and queuing network reported in the literature [4, 6, 16]. Since resource allocation in BPM is an interactive problem, reinforcement learning would be an appropriate option to solve the corresponding MDP problem. Ly et al. discovered the task allocation rules from historical data and organizational structure by using the decision tree algorithm. They considered process performers and the type of activities as input, and whether participants are involved in activities as classification results, to learn the resource allocation model inductively. They concentrated on a new aspect of process mining: mining staff assignment rules. They have shown the problem of deriving staff assignment rules using information from audit trail data and organizational information. Huang et al. [24] have used decision tree learning to derive meaningful staff assignment rules. Thus, it is possible to provide staff assignment information about activities enabling a better understanding of the underlying process.Yang et al. [25] used the hidden Markov model to build resource allocation models by mining initialization parameters from event logs, to recommend suitable process resources to activities according to the probability of employees involved in these activities and the transaction among the staff.

Zhao et al. [26] proposed an entropy-based clustering ensemble method for resource allocation. By mining process logs, resource characteristics including both performance elements and competence metrics were analyzed. Moreover, a priority-based schedule algorithm was designed to support dynamic resource allocation in multi-instance process contexts. Their method is evaluated with real-scenario-based experiments, where the proposed method outperforms other related methods and keeps the workload of human resources balanced. Based on theirs proposed approach, it is possible to find suitable resources and make more reasonable operational decisions [26, 27].

To the best of our knowledge, there has been little attention paid to the objectives of this paper altogether in the literature. Therefore, papers related to cycle time reduction, workload balancing and the trade-off between them have been investigated individually. Some papers proposed heuristic methods based on resource relation to reduce cycle time. Since resource allocation in BPM is an interactive problem, reinforcement learning is used to solve the decision problem. Therefore, lack of a novel approach to handling the resource allocation decision problem considering the mentioned objectives is altogether felt.

2. PROPOSED METHOD

This paper proposes a multi-objective approach to the problem of cycle time reduction in BPM. The approach uses reinforcement learning to better handle the interactive environment of resource allocation problem through entropy measure. Since workload fairness is also another objective, resource allocation algorithm is equipped with workload balancing method.

2. 1. Task Similarity Distance

By investigating BPI challenge 2012 dataset, it is observed that 4823 pair tasks are simultaneously executed by one resource. Additional analysis of the pair tasks, this comes out that they are relevant from data perspective and the resources probably intentionally choose to process the pair tasks; therefore, to process the pair tasks faster. The fact is also revealed that some pair tasks take less time when processed simultaneously than individually. Actually, the simultaneous execution of two tasks takes less time than their individual execution if and only if they have some subtasks in common. In other words, simultaneous execution of the two tasks leads to single execution of the common subtasks. As an example, suppose there are two tasks named "Check military service status" and "Check former school" in "Graduation" process of students at a university. In both tasks, students' letters in their file must be checked and this would be a common subtask of the two tasks. Accordingly, to better express the concept of the relevant pairs of tasks, they are called "similar tasks." To measure the similarity between two tasks, a measure of similarity distance should be defined.

The question then arises as how similar work items could be identified from an event log and then to be classified. To extract similar work items from an event log, concurrent pairs of work items for each resource should be identified. Then, for each pair, the processing time of the concurrent pair of work items is compared with the processing time of the pair of work items done independently and not concurrently. If the time of processing both work items in a concurrent way is less than the time of processing the same work items in a concurrent way, it means that concurrency helps both work items to be processed faster and both work items are then related and similar. Therefore, it would be a wise decision to assign both work items into a single work list. Note that, we extract pairwise similar work items. Having identified similar work items, we now extract similar tasks. As you know, a work item is the same as a task when the case is involved. Therefore, when similar work items are made independent from the concept of cases, similar tasks are obtained. A mathematical definition for measuring similarity of tasks is formulated as follows:

$$Similarity\,(t_i, t_j) = {T(t_i, t_j)}\Big/{T(t_i) + T(t_j)} \qquad (1)$$

where $T(t_i)$ and $T(t_j)$ represent the time of processing tasks ti and tj when they are processed independently and not concurrently. $T(t_i)$ is actually the average processing time of all work items instantiated by ti. $T(t_j)$ is also the average processing time of all work items instantiated by tj. $T(t_i,t_j)$ represents the average time of processing all concurrent pairwise work items wti, wtj instantiated by task ti and tj respectively when the work items are processed concurrently. If no concurrent pairwise work items exist, $T(t_i,t_j)$ would be incomputable because taking average of a null set is incomputable. The range of functions represented in Equation (1) is $(0, +\infty)$, in which value $+\infty$ is used in cases where work items ti and tj have not been executed simultaneously in the log and, consequently, $T(t_i,t_j)$ would be incomputable. Therefore, a similarity distance is defined according to Equation (2).

$$Similarity\ Distance\,(t_i, t_j) = {1}\Big/{(1 + Similarity(t_i, t_j))} \qquad (2)$$

The range of distance function in Equation (2) is (0, 1), in which value 0 expresses that the pair of tasks are not similar and value 1 expresses that the pair of tasks are similar.

Event logs contain time information about work items. Therefore, time information about tasks should be extracted from time information about work items by the following formula:

$$T(t_i, t_j) = \frac{\sum_{\substack{\forall\, wi_m, wi_n \\ \text{in the same} \\ \text{worklist}}} \left(len(wi_m) + len(wi_n) - len(overlap(wi_m, wi_n))\right)}{N} \qquad (2)$$
$$,\ wi_m\ instance\ of\ \ t_i\ and\ \ wi_n\ instance\ of\ \ t_j$$

$$T(t_i) = \frac{\sum_{\forall\, wi_k\, instances\, of\, t_i} len(wi_k)}{\mathrm{N}} \tag{3}$$

where len(w) = (w.τcomplete – w.τentrance) is the processing interval time of work item w. By averaging the property of work items, the property would be independent of case concept and associated with task concept. Also, by taking the average of the results of function len over all work items, the results would be associated with task concept.

2. 2. Entropy Measurement of Work List Entropy is an indication of disorder or uncertainty associated with random variables (information theory) [27-30]. In computer science, the arrival and execution of any data is information and entropy is used to compute the amount of disorder or improbability in that information.

The core idea of the entropy of work lists is based upon tasks similarities. As the number of similar tasks in a work list increases, the entropy of the work list decreases. The idea is inspired by the fact that similar tasks may have some subtasks in common, i.e., accessing locally close archive files for human resource or loading identical files into ram for machine resource. Therefore, minimizing the entropy of work lists associated with each resource leads to reducing the processing time of rather consecutive similar tasks. Eventually, minimizing the entropy of work lists leads to cycle time reduction.

Suppose work list w contains n work items and all these work items would be divided into k different classes of work items. Each class contains similar tasks. Intuitively, as the number of classes decreases, so does entropy. More precisely, the mathematical definition of entropy is presented as follows:

$$Ent = -\sum_{i=1}^{k} p_i log_2(p_i) \tag{4}$$

where p_i is the proportion of work item belonging to class *i* and *k* is number of classes of work items (for example number of *task* in process model). By applying the concept of tasks similarities onto resource allocation, the work list of each resource would be arranged with similar work items which have some subtasks in common, and the service rate, hence, increases. This would also lead to average cycle time reduction of all cases. In other words, the aim of this algorithm is to inject similar tasks into work list of each resource. To this end, a table is created for each resource similar to Table 3. In that table, the similarity distance between each two tasks is computed.

In the proposed resource allocation, when a task comes in, the task is allocated to a work list whose entropy would decrease more or in some cases increase less than others after allocation.

To compute the entropy of a work list, the work items of the work list are classified based on the task similarity distance. The entropy of work lists is computed in two ways: one without considering the incoming task and the other with considering the incoming task. The amount of difference between the entropy of the two states would determine the work list which the incoming task would be allocated. The work list whose entropy would decrease more or in some cases increase less than others after allocation is selected.

2. 3. Reinforcement Learning for BPM In this section, we use reinforcement learning based resource allocation mechanism (RLRAM), It presented by Huang et al. [4]. We merge the concept of entropy within the reinforcement learning model. We control learning algorithm to achieve our purposed goal which is minimizing the entropy of work lists by defining of the proper objective function in RLRAM. By just changing the objective function as follows:

$$\begin{gathered} R_a(s_t, s_{t+1}) = \\ \frac{1}{\min\limits_{\forall r \in s.resources}\{Ent_{t+1}(r.worklist_{t+1}) - Ent_t(r.worklist_t)\}} \\ if\ r.workload < \frac{\sum_r r.workload}{N} \\ , and\ \ r.worklist_{t+1} = \ r.worklist_t \cup \{wi\} \end{gathered} \tag{5}$$

where r.worklist$_t$ is the work list of resource r at time t before allocating work item wi. When work item wi is enabled by the system, task allocation algorithm selects a resource to do work item, it selects the best resource whom entropy of his/her worklist is minimized after task assignment (r.worklist$_{t+1}$). $R_a(s_t, s_{t+1})$ is reward function to perform action a for state St in RLRAM.

To better understand the effect of entropy on workload balancing, we start with an example. Suppose a process in which there are three resources R$_1$, R$_2$ and R$_3$ with the same specializations and three types of work items called wi$_1$, wi$_2$ and wi$_3$. The average processing time of work items wi$_1$, wi$_2$ and wi$_3$ is 1 hour, 2 hours, and 3 hours, respectively. Consider an event log including 9 cases of type wi$_1$, 9 cases of type wi$_2$, and 3 cases of type wi$_3$. Table 1 indicates the two types of balanced allocation to three mentioned resources.

TABLE 1. Two types of balanced allocation with work list entropy values

Resource	Balanced work list	Entropy Value	Workload	Entropy based Balancing	Entropy value	Workload
R$_1$	$wi_3wi_1^3wi_2^3$	1.581	12	$wi_1^9wi_2$	1.128	11
R$_2$	$wi_3wi_1^3wi_2^3$	1.581	12	wi_2^6	0	12
R$_3$	$wi_3wi_1^3wi_2^3$	1.581	12	$wi_3^3wi_2^2$	0.619	13
	Average	**1.581**	**12**		**0.672**	**12**

In fact, the resources spend less time and cost to do similar work items. Minimizing entropy helps to arrange similar work items close to each other. Table 1 shows two types of balanced allocation. This paper suggests that processing similar tasks in a rather consecutive order takes less time than when the same tasks are carried out individually. Therefore, balanced work list with minimized entropy takes less time than does balanced work list with uncontrolled entropy.

3. EXPERIMENTS AND RESULTS

To analyse the effectiveness of the proposed method, we consider a real life log. BPI challenge 2012 is the real life log chosen to be the benchmark dataset. The novelty of proposed method lies in applying the concept of entropy in reinforcement learning based resource allocation while keeping workloads balanced. Therefore, to prove the effectiveness of applying the entropy concept, the proposed method is compared with reinforcement learning based resource allocation while keeping workloads balanced without considering entropy.

In section 4.1, the performance of the algorithm in heavy load settings is evaluated against workload balancing algorithm. The BPI challenge 2012 dataset is chosen for the comparison of the two algorithms because the log contains simultaneous execution of tasks by resources. In other words, the concept of tasks similarities is applicable to event logs. In this section, heavy load settings are applied to the two algorithms with decreasing resources from 55 to 10 and increasing inter-arrival rate by changing it from 0.02 to 0.2. Experiments are iterated 10 times to make the evaluations more robust. Finally, comparative results of the proposed method against NBSR algorithm [10] with the related discussions are presented in section 4.2.

3. 1. Case Study The event log of a real-life process of BPI Challenge 2012, adopted from a Dutch financial institute, is used for simulation. Of 6078 cases, 5 activities and 55 resources are acquired from the real-life log. The proposed method is operational in circumstances where the enabling rate of work items is higher than the processing rate of work items. To satisfy this criterion, a number of resources are set to 10 in the simulation runs. In the dataset, no specific role is defined for resources executing tasks. Observations of the dataset reveal that 52 of 55 resources executed almost all the given tasks. Therefore, reducing the number of resources to 10 does not contradict the roles of resources. Consequently, the reduction just increases the workloads of the resources and does not produce any other side effects.

Ten resources are randomly chosen from 52 resources where no specific role is defined. The chosen resources are specified by *Resource_id* in column 2 of Table 2. To obtain the most appropriate probability distribution function, three tests of Kolmogorov- Smirnov, Anderson-Darling, and Chi-square are used. Four probability density functions including Pareto, Lomax, Weibull, and Chi-square are fitted for the distributions by three mentioned tests. According to the results, Lomax probability distribution function with $f(x) = \frac{\alpha\beta^{\alpha}}{(x+\beta)^{1+\alpha}}$ most fitted the distribution and all the simulation runs are based on the Lomax probability distribution function.

In Figure 1, the process model is expressed in terms of a workflow net. The process may start with the '*W Fixing incoming lead*' with probability 0.46 or start with '*W Filling in information for the application*' with probability 0.54 [6, 7].

In fact, different tasks have different self-loop probabilities; but, for the convenience of calculation, the self-loop probability of all tasks is set to 0.2. If discounted factor γ is large, the return has a greater effect on Q value than the immediate payoff. Since the process is not much complicated, the value of γ is set to 0.9 and the value of α learning rate is set to 0.6.

The probability density function of processing each task for each resource can be obtained from the real log (Table 2). Consequently, a model with probabilistic arcs and an estimated probability density function for processing time of each task are obtained from BPI challenge 2012 event log.

Inter-arrival rate parameter actually is the mean arrival time of each two consecutive requests. In this experiment, inter-arrival rate parameter changes between 0.02 and 0.2 by step 0.02 which totally 10 simulation runs for each algorithm are executed.

3. 2. Discussion The real-life process of BPI challenge 2012 contains five different tasks. The entropy measure defined in section 3.1 is calculated based on the identification of similar tasks. The similarity distance between each pair of tasks of the real life process is computed according to Equation (2). Similarity distances between pairs of tasks for resource 10228 are shown in Table 3. As you can see in Table 3, all diagonal elements of the matrix are greater than 0.5, meaning each task is similar to itself.

As an example, the similarity distance between '*W_Calling after sent offers*' and '*W_Fixing incoming lead*' is equal to 0.72; since the value is greater than 0.5, the pair of tasks are considered similar. Irrelevant tasks are shown by mark '<0.5' in the table. For instance, the similarity distance between '*W_Filling in information for the application*' and '*W_Fixing incoming lead*' is lower than 0.5 and the pair of tasks are considered irrelevant and not similar. As the value of similarity distance increases, the level of similarity between the pair of tasks becomes greater.

TABLE 2. Estimated Lomax probability density function for processing time of each task for each resource.

	Resource Id in event log	Pareto 2(Lomax) Parameter	List of Tasks and Lomax Parameter for each resource				
			W_Fixing incoming lead	W_Filling in information for the application	W_Calling after sent offers	W_Assessing the application	W_Calling to add missing information
1	10228	α	1.1659	0.3231	0.3719	11.347	0.7687
		β	0.3515	0.0267	0.5817	4.752	0.1127
2	10909	α	1.3038	1.0285	1.3347	1.5084	6.0896
		β	0.2847	0.2503	0.4467	0.2977	0.7845
3	10914	α	1.7882	0.9165	1.1167	4.688	8.371
		β	0.177	0.4521	0.8534	1.4985	0.093
4	10932	α	0.5695	1.0638	0.4830	0.8754	0.8670
		β	0.1564	0.4327	0.0616	0.2501	0.1685
5	10939	α	2.603	2.0631	1.1875	1.4551	9.9052
		β	0.5022	1.0884	0.4841	0.3718	0.498
6	11009	α	1.5463	1.3662	0.9693	1.6326	4.733
		β	0.4951	0.8358	0.3566	0.3167	1.6424
7	11169	α	0.7350	0.8552	0.7534	0.8220	1.0093
		β	0.1907	0.5319	0.3689	0.2904	0.2613
8	11181	α	3.0658	1.1156	1.5279	2.3491	0.7100
		β	0.5000	0.2221	0.6820	0.4941	0.036
9	11259	α	3.0151	3.8848	1.2281	2.2368	2.7016
		β	0.4227	1.437	0.0974	0.0689	0.4236
10	11302	α	0.7157	0.7134	1.1499	2.3329	1.6438
		β	0.3439	0.0784	0.4358	0.6124	0.3156

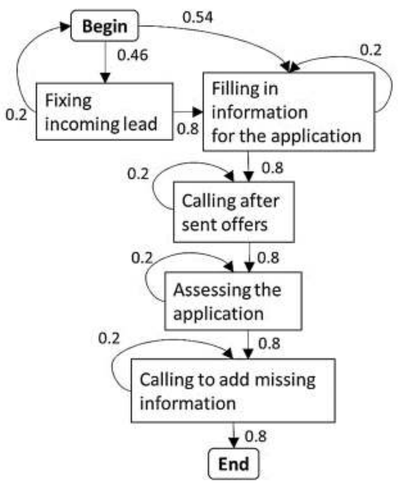
Begin
0.54
0.46
0.2
Fixing incoming lead
0.2
0.8
Filling in information for the application
0.2
0.8
Calling after sent offers
0.2
0.8
Assessing the application
0.2
0.8
Calling to add missing information
0.8
End

Figure 1. A Real-life Financial Process [7]

The proposed method is resource allocation using entropy-based on '*reinforcement learning based resource allocation mechanism'* (RLRAM) [4] while keeping workloads balanced (MinEntropy+RLRAM). To prove the effectiveness of applying the entropy concept, the proposed method is compared with resource allocation using RLRAM while keeping workloads balanced without involving entropy (MinWorkload+RLRAM). The experiment setting for the two methods is the simulated data generated in section 4.1. Simulated tokens are given to the process model with the given inter-arrival time model and given density distribution function for each task according to parameters presented in section 4.1.

Comparative results of the two approaches are shown in Figure 2. The x-axis of Figure 4 represents the time in minutes. At the beginning of time, the first token arrives at the process. As time passes, more tokens arrive at the process at a predetermined rate. In the middle of the arrival of new tokens, some old tokens complete their process instances. Cycle times of the completed tokens are computed. All cycle times of all tokens are shown in the Y-axis of Figure 2. As you can see in Figure 2, the

TABLE 3. Similarity distance matrix of tasks for resource No. 10228

$\boldsymbol{Similarity\ Distance}\ (t_i, t_j) = {}^{\mathbf{1}}/(\mathbf{1} + \boldsymbol{Similarity}(t_i, t_j))$	Fixing incoming lead	Filling in information for the application	Calling after sent offers	Assessing the application	Calling to add missing information
Fixing incoming lead	0.355	0.702	0.416	×	0.796
Filling in information for the application	0.702	×	0.647	×	×
Calling after sent offers	0.416	0.647	0.193	×	0.696
Assessing the application	×	×	×	×	×
Calling to add missing information	0.796	×	0.696	×	0.602

‘*MinEntropy+RLRAM*’ method performs better than the ‘*MinWorkload+RLRAM*’ method. The ‘*MinEntropy+RLRAM*’ method can handle and process all the tokens in less time than the ‘*MinEntropy+RLRAM*’ method. The total processing time of all tokens for the ‘*MinEntropy+RLRAM*’ method is equal to 178.106 time units and the total processing time of all tokens for the ‘*MinEntropy+RLRAM*’ method is equal to 211.31 time units. It shows 16% reduction in the total processing time of 200 tokens with a given arrival time model and given density distribution function for each task according to section 4.1. Note that cycle times of the completed tokens are reduced in the ‘*MinEntropy+RLRAM’* method in contrast to the ‘*MinWorkload+RLRAM’* method.

Figure 3 shows the comparative results of Figure 2 in the average mode. Every point in time represents the average of the cycle times of all completed cases till the given time. The average cycle time of all cases after completion by the ‘*MinEntropy+RLRAM’* method is equal to 32.89, and the average cycle time of all cases after completion by the ‘*MinWorkload+RLRAM’* method is equal to 54.46. It shows 39% reduction in the average cycle time of 200 tokens with a given arrival time model and given density distribution function for each task according to section 4.1.

To elaborate the effect of entropy on the process performance, the entropy of work lists is investigated over time. Figure 4 shows the sum of the entropy of all work lists at every point in time. It clearly shows the

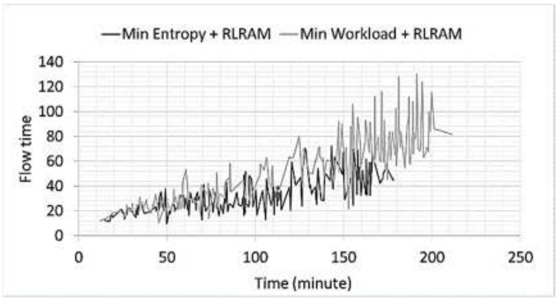

Figure 2. The cycle time of cases as 200 cases arrive at the process

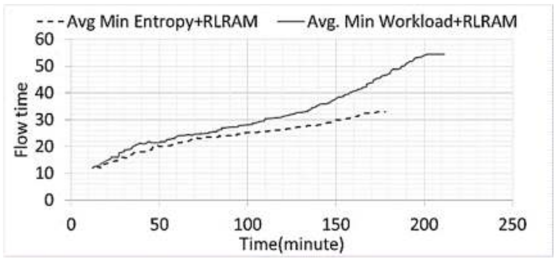

Figure 3. The average cycle time as 200 cases arrive at the process

‘*MinEntropy+RLRAM’* method performs better than the ‘*MinWorkload+RLRAM’* method in handling cases over time. The cycle time reduction for the ‘*MinEntropy+RLRAM’* method in Figure 3 is proportional to the reduction of the entropy of work list as shown in Figure 4. The ‘*MinEntropy+RLRAM’* method arranged work lists with similar tasks.

Results for the two mentioned algorithms with an inter-arrival rate of 0.02 are shown in Figures 2-4. To make the experiments more robust, they are iterated 3 times for each of the 10 experiments with a specific inter-arrival rate. In Figure 5, the horizontal axis represents different inter-arrival rates of tokens and the vertical axis represents the mean cycle time of all cases. As you can see in Figure 5, Random and Order resource allocation algorithms are almost alike. The resource allocation algorithm which chooses the resource with the minimum workload at any moment in time does have a better

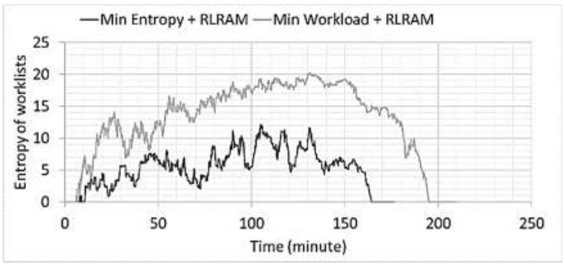

Figure 4. The sum of the entropy of all work lists as 200 cases arrive at the process

performance with respect to Random and Order resource allocation algorithms, especially on inter-arrival rates more than 0.1. The proposed resource allocation algorithm which chooses the resource with the minimum entropy of work lists while having workloads balanced shows predominance with respect to other resource allocation algorithm. The proposed method is even robust in lower inter-arrival rates while resource allocation with the minimum workload fails. The proposed method shows 32% reduction in cycle time with respect to the resource allocation with minimum workload.

As shown in Figure 5, NBSR algorithm [11] obtained better cycle time results in terms of the proposed algorithm for inter-arrival rates of 0.18 and 0.2 while in lower inter-arrival rates (higher request rate), the proposed algorithm takes over NBSR algorithm. The results are as expected. The higher the inter-arrival rates, the lower the probability of queue forming for work lists of resources. Consequently, the probability of assigning similar tasks to the same work list increases and does not show the advantage of the proposed algorithm. In other words, the proposed algorithm by the *'MinEntropy+RLRAM'* method would better perform in heavy load settings with respect to other algorithms.

Entropy-based reinforcement learning is used to reduce cycle time for resource allocation. As mentioned before, workload balancing is defined as another objective to be satisfied. Applying entropy-based reinforcement learning algorithm along with the workload balancing objective has a negative effect on workload balancing and makes workloads deviate from its balance point. The amount of the side effect is statistically computed over time and it is shown in Figure 6. The deviations are computed by Equations (7), (8), and (9). Equation (7) specifies the average of workloads of all resources at time *t*. Equation (8) computes the variance of workloads of all resources at time *t*. The less the value of this equation, the more the balancing is gained.

$$E(t) = \frac{\sum_{\forall r \in Resources} Workload_r(t)}{card(Resources)} \quad (7)$$

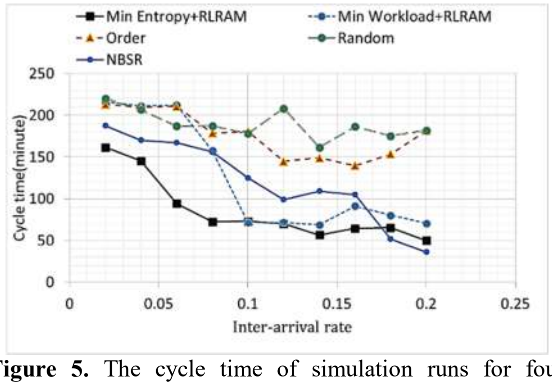

Figure 5. The cycle time of simulation runs for four algorithms with different inter-arrival rates.

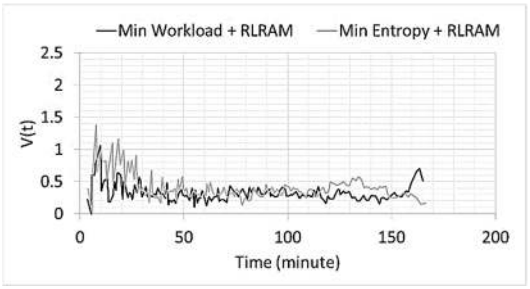

Figure 6. The *V(t)* function as 200 cases arrives at the process

$$V(t) = \frac{\sum_{\forall r \in Resources} |workload_r(t) - E(t)|}{\sum_{\forall r \in Resources} workload_r(t)} \quad (8)$$

$$avgV = \sum_{t1}^{t2} \Delta t \times V(t) \Big/ (t2 - t1) \quad (9)$$

Figure 6 shows that the side effect of applying entropy-based reinforcement learning algorithm along with the workload balancing has the expected negative effect but it only deviated 4.3% from its balance point, whereas 32% reduction in cycle time is achieved. The deviation amount is the difference between the values computed by Equation (8) for both methods mentioned above.

The goal of workload balancing mechanism in resource allocation is to equally balance work lists of resources associated with each activity. Therefore, when a work item comes to a task, the resource with minimum workload associated with the given task is selected. Then the resource is allocated to the work item. In this mechanism, resource allocation decision in each task is independent of the resource allocation decisions in the other tasks of a process. By applying reinforcement learning along with workload balancing in resource allocation, resource allocation decision problem is solved with an approach to considering resource allocation decisions of the other tasks of a process. This enables the workload balancing mechanism in resource allocation to globally solve the problem of resource allocation decision with respect to history. This would also make resource allocation decisions dependent on each other.

Increasing task similarities of each work list in resource allocation leads to cycle time reduction. Minimizing the entropy of similar tasks of work lists increases the probability of co-occurrences of similar tasks in each work list. To globally solve the issue, minimizing the entropy of work lists is applied by the reinforcement learning approach. In the proposed method, the entropy-based reinforcement learning is combined with workload balancing mechanism. Consequently, the proposed method is able to reduce cycle time, while maintaining workloads balanced.

The side effect of the proposed combinational method is shown in Figure 6. To compute the amount of side effect, the average of workloads of all resources is first computed using Equation (7). Then the sum of absolute distances between workloads of resources and the average of workloads is computed as a function of time. Finally, the value is normalized by the sum of workloads of all resources, which is expressed in Equation (8).

The proposed combinational method is compared against reinforcement learning based workload balancing in Figure 6. The values in Figure 6 are obtained from Equation (7). Integrals of two functions are computed by Equation (9) in order to compare the differences. The integral of the proposed method is 4.3% higher than that of reinforcement learning based workload balancing; on the other hand, the increase in the co-occurrence of similar tasks in the work list leads to 32% reduction in cycle time.

4. CONCLUSION

The main contribution of this paper is an entropy-based reinforcement learning approach for the optimization of cycle time in BPM with the aim of workload balancing. A solution to cycle time optimization problem is closely related to resource allocation problem in BPM within its dynamic environment. Q-learning is employed to learn how to interact with this dynamic environment and make the allocation decisions real-time. Simulation experiments were conducted to evaluate the approach with a real life log. Applying entropy-based reinforcement learning algorithm along with the workload balancing achieved 32% reduction in cycle time. The proposed resource allocation method reduces cycle time by satisfying two criteria: cycle time reduction and workload balancing. The proposed method outperforms other resource allocation methods such as NBSR, Order, and Random and 'MinWorkload+RLRAM' methods. Another interesting direction for further research may deal with predicting the entropy in the next states of BPM system and reducing the entropy as much as possible by using a predictor system.

5. REFERENCES

1. Van Der Aalst, W., Van Hee, K.M. and van Hee, K., "Workflow management: Models, methods, and systems, MIT press, (2004).
2. Zhao, W. and Zhao, X., Process mining from the organizational perspective, in Foundations of intelligent systems. 2014, Springer.701-708.
3. Huang, Z., Lu, X. and Duan, H., "A task operation model for resource allocation optimization in business process management", ***Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on***, Vol. 42, No. 5, (2012), 1256-1270.
4. Huang, Z., van der Aalst, W.M., Lu, X. and Duan, H., "Reinforcement learning based resource allocation in business process management", ***Data & Knowledge Engineering***, Vol. 70, No. 1, (2011), 127-145.
5. Wang, J. and Kumar, A., A framework for document-driven workflow systems, in Business process management. 2005, Springer.285-301.
6. Liu, X., Chen, J., Ji, Y. and Yu, Y., "Q-learning algorithm for task allocation based on social relation", ***Process-Aware Systems***, (2015), 49-58, DOI: 10.1007/978-3-662-46170-9_5.
7. Zur Muehlen, M., "Organizational management in workflow applications–issues and perspectives", ***Information Technology and Management***, Vol. 5, No. 3-4, (2004), 271-291.
8. Xu, J., Liu, C. and Zhao, X., "Resource allocation vs. Business process improvement: How they impact on each other", in BPM, Springer. Vol. 2008, (2008), 228-243.
9. Nisafani, A.S., Wibisono, A., Kim, S. and Bae, H., "Bayesian selection rule for human-resource selection in business process management systems", ***Journal of Society for e-Business Studies***, Vol. 17, No. 1, (2014), 53-74.
10. Wibisono, A., Nisafani, A.S., Bae, H. and Park, Y.-J., On-the-fly performance-aware human resource allocation in the business process management systems environment using naïve bayes, in Asia pacific business process management. 2015, Springer.70-80.
11. Azid, S.N.L.S.K.I.A., "House of improvement model to enhance prioritisation of solutions in decision making: A case study", ***International Journal of Engineering-Transactions B: Applications***, Vol. 27, No., (2014), 1195-1204.
12. H Nematzadeh, Z. Nematzadeh, "Deterministic Measurement of Reliability and Performance Using Explicit Colored Petri Net in Business Process Execution Language and Eflow" ***International Journal of Engineering-Transactions A: Basics***, Vol. 28, No. 10, 1439-1446.
13. H. Mokhtari, A. Noroozi and Molla-Alizadeh-Zavardehi, S., "A reliability based modelling and optimization of an integrated production and preventive maintenance activities in flowshop scheduling problem", ***International Journal of Engineering Transactions C: Aspects***, Vol. 28, No. 12, (2015), 1774-1781.
14. Kumar, A., van der Aalst, W.M. and Verbeek, E.M., "Dynamic work distribution in workflow management systems: How to balance quality and performance", ***Journal of Management Information Systems***, Vol. 18, No. 3, (2002), 157-193.
15. Larbi, S. and Mohamed, S., "Modeling the scheduling problem of identical parallel machines with load balancing by time petri nets", ***International Journal of Intelligent Systems and Applications***, Vol. 7, No. 1, (2014), 42-48.
16. Ha, B.-H., Bae, J., Park, Y.T. and Kang, S.-H., "Development of process execution rules for workload balancing on agents", ***Data & Knowledge Engineering***, Vol. 56, No. 1, (2006), 64-84.
17. Liu, Y. and Zhang, K., "Strategy for workflow task assignment based on load balance and experiential value", ***Computer Engineering***, Vol. 21, No. 1, (2009), 1-22.
18. Bellmam, R., *Dynamic programmingprinceton university press*. 1957, Princeton.
19. Ha, B.-H., Bae, J. and Kang, S.-H., Workload balancing on agents for business process efficiency based on stochastic model, in Business process management. 2004, Springer.195-210.
20. Yu, D., Deng, L. and Acero, A., "Using continuous features in the maximum entropy model", ***Pattern Recognition Letters***, Vol. 30, No. 14, (2009), 1295-1300.
21. Bozkaya, M., Gabriels, J. and Werf, J., "Process diagnostics: A method based on process mining", in Information, Process, and Knowledge Management, 2009. eKNOW'09. International Conference on, IEEE., (2009), 22-27.

22. Van Der Aalst, W.M., Reijers, H.A. and Song, M., "Discovering social networks from event logs", ***Computer Supported Cooperative Work (CSCW)***, Vol. 14, No. 6, (2005), 549-593.
23. Senkul, P. and Toroslu, I.H., "An architecture for workflow scheduling under resource allocation constraints", ***Information Systems***, Vol. 30, No. 5, (2005), 399-422.
24. Huang, Z., Lu, X. and Duan, H., "Resource behavior measure and application in business process management", ***Expert Systems with Applications***, Vol. 39, No. 7, (2012), 6458-6468.
25. Yang, H., Wang, C., Liu, Y. and Wang, J., "An optimal approach for workflow staff assignment based on hidden markov models", in On the Move to Meaningful Internet Systems: OTM 2008 Workshops, Springer., (2008), 24-26.
26. Zhao, W., Liu, H., Dai, W. and Ma, J., "An entropy-based clustering ensemble method to support resource allocation in business process management", ***Knowledge and Information Systems***, Vol. 48, No. 2, (2015), 1-26.
27. De Luca, A. and Termini, S., "A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory", ***Information and Control***, Vol. 20, No. 4, (1972), 301-312.
28. Sheng, J., Qi, B., Dong, X.-j. and Tang, L.-r., "Entropy weight and grey relation analysis based load balancing algorithm in heterogeneous wireless networks", in Wireless Communications, Networking and Mobile Computing (WiCOM), 2012 8th International Conference on, IEEE., (2012), 1-4.
29. Zhang, X. and Zheng, G., "Relationship between similarity measure and entropy of interval type-2 fuzzy sets", in Fuzzy Systems and Knowledge Discovery (FSKD), 2013 10th International Conference on, IEEE., (2013), 98-102.
30. Zeng, W. and Li, H., "Relationship between similarity measure and entropy of interval valued fuzzy sets", ***Fuzzy Sets and Systems***, Vol. 157, No. 11, (2006), 1477-1484.

Cycle Time Optimization of Processes Using an Entropy-Based Learning for Task Allocation

I. Firouzian, M. Zahedi, H. Hassanpour

Computer Engineering & IT Department, Shahrood University of Technology, Shahrood, Iran

PAPER INFO

Paper history:
Received 04 May 2019
Received in revised form 16 June 2019
Accepted 05 July 2019

Keywords:
Business Process
Entropy-based Optimization
Reinforcement Learning
Resource Allocation
Workflow Management

چکیده

بهینه سازی چرخه زمانی یکی از مهم ترین چالش های موجود در مدیریت فرآیندهای سازمانی می باشد. اگرچه، تحقیق‌های زیادی در این حوزه انجام شده است، اما به شباهت میان وظایف توجه کمتری شده است. در این مقاله، رویکرد جدیدی به جهت بهینه سازی چرخه زمانی بوسیله کاهش آنتروپی لیست‌های کاری در تخصیص منابع در کنار حفظ برقراری تعادل بارکاری ارائه شده است. ایده آنتروپی لیست های کاری از این حقیقت نشأت می‌گیرد که زمان لازم برای اجرای وظایف مشابه با ترتیب متوالی توسط یک منبع کمتر از زمان لازم برای اجرای همان وظایف توسط همان منبع به صورت کاملا جداگانه می‌باشد؛ بدین جهت، معیار آنتروپی تعریف شد که میزان شباهت میان هر دو وظیفه دلخواه در لیست کاری مفروض را نشان می‌دهد. علاوه‌براین، برقراری تعادل بارکاری هم به عنوان یک هدف بهینه سازی در نظر گرفته شده است، زیرا نه تنها بهینه سازی چرخه زمانی مهم است، بلکه عدالت کاری نیز باید ارضاء شود. نتایج آزمایشگاهی برروی پایگاه‌داده سابقه‌رویداد BPI challenge 2012 نشان می دهد که روش پیشنهادی منجر به ۳۲٪ کاهش چرخه زمانی در مقایسه با حالتی که تخصیص منبع با یادگیری تقویتی بدون در نظرگرفتن آنتروپی بوده است.

doi: *10.5829/ije.2019.32.08b.05*