IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 7, 2021, accepted February 2, 2021, date of publication February 5, 2021, date of current version February 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3057362

Spatial-Temporal Adaptive Optimal Allocation of Archival Tasks

CHEN LILAN[1] AND CHEN YONGSHENG[2]
[1]School of Foreign Languages, Guangdong Pharmaceutical University, Guangzhou 510006, China
[2]School of Information Management, Sun Yat-sen University, Guangzhou 510006, China

Corresponding author: Chen Yongsheng (cyszd111@163.com)

This work was supported by one of the Major Projects of National Social Science Fund of China No. 17ZDA200.

ABSTRACT Archival task allocation for Modern Canton Customs (1861-1949) is a heavy workload due to the massive quantities of data. An archival task allocation system is used to make the allocation process easier. However, traditional methods applied in the allocation system lead to lower efficiency and waste of human force because of the irrational hypotheses. This article presents two algorithms of allocating to spatial-temporal system users based on the heuristic rules, namely offline ($ALGO_{OFF}$) and adaptive ($ALGO_{AD}$) allocation algorithms, which significantly improve the performance of the archival task allocation system. With simulation data and authentic data, $ALGO_{AD}$, by employing just 48 percent of the system users, can achieve the same accuracy rate as the commonly used circular policies. And the additional experiments with simulation data composed of the randomly selected system users verify the following conclusions: (1) the adaptive method is better than the offline task allocation method; (2) the adaptive algorithm can save more human force even when the skills of adaptive archive allocation system users and the difficulty of the archival translation tasks are varied; (3) the adaptive algorithm has continuability without affecting its performance; (4) the adaptive method saves resources even when the completion time the adaptive system users spend on archival translation tasks is different.

INDEX TERMS The adaptive archive allocation system, offline allocation, online adaptive allocation.

I. INTRODUCTION

Modern Canton Customs archives (1861-1949) reserved in Guangdong Provincial Archives are massive in quantities, including 16115 volumes and more than 3.7 million pictures [1]. Since the current archival task allocation algorithms make too strict hypotheses to be applied to the archive allocation system, how to efficiently allocate the massive modern Canton Customs archives to the significant number of adaptive archive allocation system users is still a problem to be solved. For instance, they usually make at least one of the following hypotheses, (1) allocate archival tasks in sequence completely; (2) the system users are willing to wait for the archival tasks to be fulfilled; (3) the quality of the system users' archival answers can be calculated in real-time. The three main constraints above lead to lower efficiency and waste of human force in the allocation system.

Law & von Ahn [2] described two models to solve the problem, i.e., the pull model and the push model. With the pull model, the system users actively select archive tasks in accordance with conditions such as the price and keywords; Lu Y [3] built a model of the benefit of each of a plurality of computing tasks under uncertainty as a function of computing resources invested in each of the computing tasks. While the popular push model directly allocates archival tasks to the system users [4], [5]. These two models are based on the source distribution principle applied in many fields: Harberger etc [6] studied monopoly and resource allocation from the view of the economy. R. Rajkumar. etc [7] proposed an analytical model for performance-driven resource allocations, which is basic and flexible. Under cloud computing background, Gawali etc [8] make use of a heuristic approach combining the so-called analytic hierarchy process (MAHP), scheduling optimization, and processing time preemption for a better resource allocation. Ideal archive allocation system [9] will allocate simple tasks such as archival translation tasks to the unskilled and difficult ones to the skilled. The increasing accumulated data on the system users [10]–[12] and the archival tasks make it possible to improve this allocation process.

The associate editor coordinating the review of this manuscript and approving it for publication was Jenny Mahoney.

In essence, the ideal practical archival task allocating system should be totally unsupervised [13], [14]. In addition, the system should allocate the archival tasks to all available system users in parallel, since making the system users wait for archival tasks for long will reduce the system efficiency, frustrating users' enthusiasm. Finally, the archival tasks should be operated in real-time so that the system users not need wait for too long for archival tasks.

The article studies the archival translation task allocation algorithm satisfying these needs to realize the adaptive archive allocation system. The modern Canton Customs archival translation tasks are of different difficulty levels. Suppose the task allocation server can visit the system users with different skills and allocate archival translation tasks to the system users, making all the available users have tasks to fulfill. By observing the system users' responses who have finished the archival translation tasks and collecting votes by majority votes or Bayesian methods, the system estimates the confidence of the system users' archival translation answers. The system's objective is to ensure that the system users can fulfill all the archival translation tasks as accurately as possible after a fixed amount of task allocation. The modern Canton Customs archival translation tasks in the adaptive archive allocation system cover archival documents of different types and different difficulty levels. Suppose the difficulty of archival translation tasks and the system users' skills are set, laying the foundation for other cases and is one practical application. The effective heuristic functions with the series of archival translation tasks have some of the domain-specific features (e.g., Customs jargons [15], proper names [16]the factors and formats of official documents), enabling the system to estimate the difficulty levels of the modern Canton Customs archival translation tasks.

In this work, we propose two algorithms for the system to allocate archival tasks: the offline and the adaptive. 1) $\mathrm{ALGO_{OFF}}$, constructed on the objective function and expected information gain module, can efficiently realize the close-to-the-best allocation. 2) $\mathrm{ALGO_{AD}}$, one adaptive algorithm, is for the adaptive archival task allocation $\mathrm{ALGO_{AD}}$. As the continuation algorithm of $\mathrm{ALGO_{AD}}$, $\mathrm{ALGO_{BIN}}$ is applied to larger scales of the adaptive allocation system users and archival translation tasks. Finally, $\mathrm{ALGO_{RT}}$ is shown as the real-time algorithm for cases when the time system users spend in fulfilling archival translation tasks is different. The experiment with a proper name matching tasks shows that the optimal allocation algorithm in the adaptive archive allocation system uses only 48% of users. The typical circular policy needs to achieve 95% of the maximum accuracy, much saving system users.

The article is organized as follows. Section II makes a introduction of allocation parameters definition and the objective of before math model setup for archival task allocation. In Section III, we show how the two algorithms, $\mathrm{ALGO_{OFF}}$, $\mathrm{ALGO_{AD}}$, and their extension version for different situations, make optimizations for a better performance of the archival translation task allocation system. In Section IV, we validated the effectiveness of the proposed algorithms. Section V provides conclusions for this article and plans for future studies.

II. TASK DEFINITION

The system assumes a group of adaptive users $\mathcal{W}$ and archival translation tasks $\mathcal{Q}$, the difficulty (or potential) of each task $q \in \mathcal{Q}$ is $d_q \in [0, 1]$, the skill parameter for each user (or potential) $w \in \mathcal{W}$ is $\gamma_w \in (0, \infty)$. The assumption of users w, the probability of accurate answers to task q and the task difficulty $d_q \in [0, 1]$ satisfies the monotone increasing function $P(d_q, \gamma_w)$. Thus we adopt formula (1) as the skill function:

$$P(d_q, \gamma_w) = \frac{1}{2}\left(1 + (1+d_q)^{\frac{1}{\gamma_w}}\right) \tag{1}$$

Suppose the system visit the system user pool and ask the users to provide the answers for the allocated archival translation tasks. The user pool is the subset of $\mathcal{W}$, made up of the available system users at a given time, and the user pool is dynamic, that is to say, the system users can come and go at will. We assume that it is of regularity when the users disappear or reappear in the system, and the interval between disappearance and reappearance is equal to the time users spend in finishing translating one archival document, and we term such a task allocation as a round. At the beginning of each round, the algorithms in this article assign the archival translation tasks to all available system users $P_t \subseteq \mathcal{W}$, and the users' translation answers and the probability of the task fulfillment will be collected at the end of each round.

While assigning the archival translation tasks to the system users, the algorithms try to maximize some utilities within a temporal interval T after fixed rounds. In order to make every system user participate in and finish as many archival translation tasks as possible, in each round of $t = 1, \ldots, T$, the methods proposed in this article will allocate each user in the system user pool P_t one archival translation task, and let each allocation algorithm have fixed n task requests in the shared adaptive archive allocation system.

A. THE OPTIMIZATION STANDARDS

Various utility functions $U(S)$ can be used to optimize the adaptive archive allocation system [17], and the natural utility functions are constructed with the expected value gain obtained by observing a group of system users, where $S \in 2^{(\mathcal{Q}\times\mathcal{W})}$ denotes allocating archival translation tasks to system users. Specifically speaking, let $\mathcal{A} = \{A_1, A_2, \ldots, A_{(|\mathcal{Q}|)}\}$ be the random variable set of accurate reference archival translation answers, $\mathcal{X} = \{X_{q,w} \mid q \in \mathcal{Q} \bigwedge w \in \mathcal{W}\}$ is the random variable set corresponding to the system users' archival translation answers, $\mathcal{X}_S$ denotes the subset of the archival translation answers $\mathcal{X}$ corresponding to the archival translation task set S. The prediction uncertainty of set $\mathcal{A}$ is quantified with the information entropy $H(\mathcal{A}) = -\sum_{a\in dom\mathcal{A}} P(a)\, logP(a)$, where the set $\mathcal{A}$ covers all possible task answers to the variables in $\mathcal{A}$, a is a

vector representing an archival translation task. The condition entrop $H((\mathcal{A} \mid \mathcal{X}_S) = -\sum_{\substack{a \in domA \\ x \in domX_s}} P(a,x)\, logP(a|x)$ denotes the prediction uncertainty made after observing the system users' archival translation answers corresponding to the variables in set $\mathcal{X}_s$.

Due to the independence of the reference answers to the archival translation tasks and the system users' archival translation answers, we get $P(a,x) = \prod_{q \in \mathbb{Q}} P(a_q) \prod_{w \in \mathcal{W}\, who\, answered\, q} P(x_{q,w}|a_q)$, where $P(x_{q,w}|a_q)$ depends on the system users' skills and the archival translation task difficulty, as shown in formula (1).

We define the value of the archival translation tasks-system user allocation as the expected reduction of the archival translation task entropy:

$$U_{IG}(S) = H(\mathcal{A}) - H(\mathcal{A}|\mathcal{X}_S) \tag{2}$$

B. THE DIMENSION OF ARCHIVAL TRANSLATION TASKS

This article focuses on the archival translation task allocation in the following three aspects:

1) The offline and online (adaptive) task allocation construction. In the offline (or static) condition, before the first round, the system chooses a complete archival translation task set T and allocates the tasks in it to each system user once, and the task set remains unchanged when the users submit their translation answers. The offline task allocation is the only option when the system users' archival translation answers cannot be collected in real-time. Though this article puts forward the $ALGO_{OFF}$ algorithm for offline archival translation cases, the experiments prove that the adaptive $ALGO_{AD}$ algorithm performs better than the offline algorithm $ALGO_{OFF}$.

2) The linear and parallel employment. One critical problem for online allocation is the number of system users employed in a specific round. Suppose the number of users to be employed is n, the best policy is to allocate tasks to the users sequentially, one user one round, so as to make the best use of such information in allocating the rest archival translation tasks. Such a policy cannot be realized in most situations, especially in the case of larger scales of China's Imperial Maritime Customs archival translation tasks, since if one system user goes wrong, it may waste most users' time demotivating them by making them wait for long. Therefore, this article tries to allocate the archival translation tasks parallelly to the system user set, which contains all available users while allocating certain tasks.

3) The known and latent system users' skills and the difficulty of the archival translation tasks, which are the key to the matching between the archival translation tasks and the system users. The best policy is to allocate the most difficult task to the most skilled and the simple ones to those unskilled. Since the system users generally establish long-term relationships with the task requesters, with the aggregation of the system, EM policy can estimate the accuracy of system users with fewer samples.

When the system users' skills or the difficulty of archival translation tasks are unknown, the task allocation will have the exploration-exploitation balance problem. For instance, asking one golden question with a known answer can estimate information related to system users' skills. Of course, we shouldn't ask questions whose answers are unknown even to the system.

III. OFFLINE ALLOCATION

This article first deals with the simplest method of task allocation, whose solution provides a foundation for the more powerful algorithms. Suppose an adaptive archive allocation system [20] can allocate archival translation tasks to a system user set and can observe their translation answers only after all the users submit their archival translation answers. Thus it cannot allocate new tasks until all tasks in one round are fulfilled because of the inconsistency of the submission time of users' translation answers. This article also supposes that the adaptive archive allocation system can compute the difficulty of archival translation tasks and the system users' skills by itself.

Theorem 1: It is NP difficult to find a way to allocate archival translation tasks to system users to maximize the arbitrary utility function $U(S)$.

Proof: We prove this theorem by simplifying the archive allocation problem as offline adaptive archive allocation system. Suppose an instance of this problem is a set X containing n archival translation tasks and the function $s{:}X \rightarrow \mathbb{Z}^+$, the partition problem is equivalent to whether the set X can be divided into two subsets X_1 and X_2 $(s.t. \sum_{x \in X_1} s(x) = \sum_{x \in X_2} s(x))$.

We construct an offline adaptive archive allocation system to solve the partition problem instance. For each archival document $x_i \in X$, suppose the corresponding system user $w_i \in W$, skills $\gamma_i = s(x_i)/max_{x_i \in X} s(x_i)$, and baseline is 1. At the same time, we set two archival translation tasks with the same difficulty d to be q_1 and q_2, then the utility function $U(S_{q,w})$. If the system user w is assigned the archival translation task q, the probability is 1. Let $f(S,q) = \log[\sum_w S_{q,w}\gamma_w + 1]$, the utility function $U(S) = \sum_q f(S,q)$.

For the optimal solutions S in the adaptive archive allocation system constructed above, if $f(S, q\_1) = f(S, q_2)$, the corresponding partition problem is true.

$\Rightarrow$: Suppose X_1 and X_2 are subsets of the set X, which have the same total number of archival translation tasks. The optimal solution S of the adaptive archive allocation system allocates the two archival translation tasks q_1 and q_2 to systems users with different skills, making their skills equal with monotonous module algorithm and submodule algorithms to improve the utility function $U(S)$ and proves S is suboptimal, thus proved.

$\Leftarrow$: Suppose S is the optimal solution to archival translation task $s.t. f(S, q_1) = f(S, q_2)$ in the adaptive archive allocation system, then the system users' skills for each archival translation task are equal, and the solution to the partition

problem is reasonable, i.e., the corresponding partition problem is true.

Considering the intractability of this problem, we put forward the approximation method. The idea of the submodular combination is that if each $A \subseteq B \subseteq \mathcal{N}, e \in \mathcal{N} : f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B)$, then the function $f{:}2^{\mathcal{N}} \rightarrow R$ is submodular. If each $A \subseteq B \subseteq \mathcal{N} : f(A) \leq f(B)$, then f is monotonous.

Theorem 2: When the system users' skills and the difficulty of archival translation tasks are known, the utility function $U_{VG}(S)$ of the value of archival translation answers are monotonous submodular in the set S.

Proof: Suppose the system users' skills, the difficulty of archival translation tasks, the real reference answers to the archival translation tasks, and the system users' votes are independent. The expected value gain function $U_{VG}(S)$ of the archival translation answers are submodular and non-degressive.

The proposed optimization is constrained by the baseline T, i.e., the system cannot allocate system users more than T tasks. Suppose $M = (\mathcal{N}, I)$, where M is the linear independence in the vector space, $\mathcal{N}$ is the baseline set, $I \in 2^{\mathcal{N}}$ is the task subset in set $\mathcal{N}$. If we partition $\mathcal{N} = \mathcal{Q} \times \mathcal{W}$ into the mutually disjoint set $B_w = \{w \times \mathcal{Q}\}$, which corresponds to the archival translation task set possibly allocated to system user w, then we can construct the expected partition matrix by defining an independent set containing at most T tasks in each set.

With the greedy selection process shown in Algorithm 1, ALGO_{OFF} can induce the following theorem:

Algorithm 1 Offline Algorithm ALGO_{OFF}

Input: system user set $\mathcal{W}$,the prior $P(a)$ for all answers, the unobserved vote set $\mathcal{X}_R$, the baseline T and the initial value S;
Output: the archival translation tasks in set $\mathcal{S}$ allocated to system users.

```
for sorted w in (W.key = γ_w) do
        for i = 1 to T do
    X_w ← {X_{q,w'} ∈ X_R | w = w'}
  for X_{q,w} ∈ X_w in vote set X_R do
  ΔX ← H(A_q | x) − H(X_{q,w}, S, x)
  end for
  X* ← arg max{ΔX : X ∈ X_w}
  Set S ← S ∪ {X*} and X_R ← X_R\{X*}
 end for
end for
```

Theorem 3. Suppose S^* is the archival translation – system user allocation, which has the highest value gain of archival translation answer U_{VG}, the archival translation task set S is found out by greedily maximizing U_{VG}, then $U_{VG}(S) \geq \frac{1}{2} U_{VG}(S^*)$.

ALGO_{OFF} allocates archival translation tasks to each system user independently by taking advantage of the monotonously increasing property of formula (1) in the system users' skills and improves its performance with the submodular evaluation. In Algorithm 1, the greedy algorithm selects system users by the descending order of system users' skills from high to low. But the experimental results prove that arranging system users' skills by the ascending order from low to high, allocating simpler tasks to system users with lower skills can greatly improve the performance of ALGO_{OFF}.

Algorithm 2 The Adaptive Algorithm ALGO_{AD}

Input: system user set $\mathcal{W}$, the prior $P(a)$ for all answers, the unobserved vote set $\mathcal{X}_R$, the baseline T, $x \leftarrow \emptyset$.

```
for t = 1 to T do
 S_t ← call ALGO_OFF (T = 1, S = ∅)
 Observe x_{S_t} and set x ←x⋃ x_{S_t}
 X_R ← X_R\S_t
 Update prior as P(a|x)
end for
```

IV. THE ADAPTIVE (ONLINE) ALLOCATION

The adaptive ALGO_{OFF} algorithm is the static allocation and cannot be adjusted with the system users' archival translation answers. However, most adaptive archive allocation systems [21] can acquire the interim output while system users try to fulfill the assigned archival translation tasks. The task allocation may assign a simple archival translation task with the answers of different qualities to more system users. This article tries to prove the adaptive archive allocation algorithms can attract all available system users at each point in time.

Unlike the offline setting, the optimal algorithm of online setting aims to compute an adaptive way to establish adaptive allocation tasks rather than fixed tasks. This problem can be seen as a POMDP formally. Set the state space $\mathcal{A}$ to be the set of answers to all archival translation tasks in set $\mathcal{Q}$, the state of POMDP keeps unchanged, with the purpose of estimating the hidden state a^*, which is the real reference answer vector to all archival translation tasks in set $\mathcal{Q}$. In each round t, the observable actions correspond to the whole task set $S_t \in 2^{\mathcal{Q} \times P_t}$ assigned to all the available system user pool $P_t \subseteq \mathcal{W}$ s.t., constituting the vector of system users' answer set X_{S_t}. Every system user can be assigned at most one task each time. The optimal solution to POMDP is π^*, the received observation set in each round up to round t is $\cup_{i=1}^{t-1} X_i$, and π^* satisfies the condition

$$(\cup_{i=1}^{t} x_i) = argmax_{\pi} E\left[U(\cup_{j=t}^{H}\{S_j \leftrightarrow \pi(\cup_{i=1}^{t-1} x_i, \cup_{i=t}^{j-1} x_i)\}\right].$$

Almost all the current POMDPs methods only adapt to the increasing linear utility function U, while the actions (the assigned archival documents) in each round in the adaptive archive allocation system is exponential, making it difficult to handle such a dilemma in such cases.

Though the selected archival documents increase and their states are observed, if the conditional expected marginal benefits won't rise as the selected tasks increase, then the function f won't satisfy the adaptive submodule of probability allocation p, thus drawing the following conclusion:

Theorem 4: Though the difficulty of archival translation tasks and the system users' skills are known, the utility function of the value of the archival translation answers $U_{VG}(S)$ is still not adaptive submodular.

Proof: To guarantee $U(S)$ is adaptive submodular, the conditional expected marginal benefit of the archival translation answers submitted by system users won't decrease with the increase of observations. As a counter-example, suppose there is a binary question with difficulty $d = 0.5$ and two system users, whose skills are $\gamma_1 = 1$ and $\gamma_2 \to \infty$ respectively. If the prior probability of the answer to the question is $P(A = True) = 0.5$, since he/she always provides accurate answers, the expected information gain of the second user's votes $H(A) - H(A|X_2)$ is initially 1. Nevertheless, if the second system user can observe the first user's votes before he/she votes, because the posterior probability $P(A = True)$ has changed, the expected information gain of the second system user will be smaller than 1.

The experiments in the following part show that $\mathrm{ALGO_{AD}}$ saves more system user force than baseline algorithms.

The $\mathrm{ALGO_{AD}}$ algorithm can quickly allocate archival translation tasks to all available system users, enabling users to finish such tasks efficiently. The algorithm constructs the utility function of archival translation tasks by estimating the difficulty of archival translation tasks and the skills of system users with questionnaires. When the archival translation tasks' scale is large, then the tasks will be divided into many subtasks to reduce their complexity. When there are a great number of system users and archival translation tasks, this article further puts forward the advanced algorithm $\mathrm{ALGO_{BIN}}$ based on $\mathrm{ALGO_{AD}}$, reducing the allocation cost of the archival translation tasks with the partition method $(|\mathcal{W}|^2 + |\mathcal{W}|C^2)$, where C is the user parameter, denoting the number of partitions. The C^2 in ALGOBIN refers to the cross product of the difficulty partition C and the reliability partition of archival translation tasks C (uniformly allocated between 0 and 1).

In the initialization process, suppose the archival translation tasks are evenly allocated, the system arranges the system users from high to low according to the average difficulty and the reliability of archival translation tasks.

V. UNITS

A. EXPERIMENTS

This article tries to answer the following questions: (1) How the adaptive method proposed in this article performs well in practice? It's answered by asking the system users to finish a challenging matching exercise on proper names in China's Imperial Maritime Customs archives. (2) What are the advantages of adaptive methods compared with the offline allocation method? It's answered by observing the average performance of many experiments with simulation data. (3) How much benefit can the adaptive method proposed in this article gain from the changes in the system users' skills and the difficulty of archival translation tasks? It's answered by simulating the different allocation of the system users' skills and difficulty of archival translation tasks. (4) To what extent are the methods proposed in this article suitable for larger scales of system users and tasks? It's answered by first estimating the policy to allocate only one task to each system user each round and then evaluating how the partition's speed gain influences the quality of the archival translation answers.

Customs archive proper names matching:

Column A	**Column B**
1. Fei-sze-lae	1) 总统令
2. Hu Pih	2) 惠州
3. Hk Tls.	3) 保税关栈
4. existing duty and likin	4) 年度贸易报告
5. Lappa	5) 通事
6. transit dues	6) 现行关税及厘金
7. Yamên (1863)	7) 关平银
8. bonded godowns	8) 北海地区
9. Yu-ch'uan Pu	9) 邮传部
10. Annual Reports on Trade	10) 关卡
11. linguists	11) 费士来
12. Weichow	12) 拱北
13. barrier	13) 子口半税
14. Pakhoi district	14) 总理衙门
15. Presidentaial Mandate	15) 湖北

1) BENCHMARK

To evaluate the performance of the adaptive policy, this article will compare the $\mathrm{ALGO_{AD}}$ based on the archival translation value gain with other policies.

The Random policy (Ra). It refers to allocating each system user one archival translation task randomly at each round.

The Circular policy (Cir). It arranges the archival translation tasks according to the votes obtained up to now, and iteratively allocates each system user random archival translation tasks which obtain the least votes.

The Uncertainty policy ($\mathrm{U_{NC}}$). It arranges the system users in the order from the lowest to the highest skills and allocates users who are the most unskilled but haven't assigned tasks yet with archival translation tasks which have the highest degree of label uncertainty but haven't been assigned, and the uncertainty is measured by the entropy of the posterior allocation of labels already received. It is different from the value-gain-based $\mathrm{ALGO_{AD}}$ in two aspects: (a) it never allocates the same tasks to two system users in one round; (b) the task's difficulty is neglected while fulfilling the archival translation tasks. Hence the Uncertainty policy can be regarded as a simpler version of $\mathrm{ALGO_{AD}}$. When the difficulty of all archival translation tasks is generally at the same level, the Uncertainty policy's performance and that of the $\mathrm{ALGO_{AD}}$ is similar.

The Accuracy-Gains policy (AG). It is similar to the value-gain-based ALGO_{AD}, but tries to greedily optimize the expected accuracy gains in each round, i.e., it tries to find the archival translation tasks-system user allocation S to maximize the formula $\mathbb{E}[\sum_{q\in\mathcal{Q}}(P(a_q|\mathcal{X}_S, 1-P(a_q|\mathcal{X}_S)))]$. Unlike the value-gain-based ALGO_{AD}, the Accuracy-gains policy is not submodular, and the greedily maximizing system user allocation process doesn't have constant error even in one single round. Despite this, due to its great heuristic power, it provides a good option for the archival translation value gains of the ALGO algorithms.

The following section will mark the value-gain-based ALGO_{AD} as VG so as to compare it with other baseline policies. Though the Random policy and the Circular policy seem priorly inferior to VG, the relative advantages of the other two policies are not known yet, and provide important approaches for the practical operation of VG.

In the following experiments, all the baseline policies and VG themselves will allocate archival translation tasks that haven't been allocated yet, until all the archival translation tasks are allocated once, and then these policies perform normally. This can guarantee all the predictions are based on at least one observation and provide empirical advantages.

2) COMPARISON OF ADAPTIVE POLICIES

To test the performance of authentic system users and archival translation tasks, this article selects a proper name matching test. Since we want to compare the different policies on the same data, we select 15 test questions, 120 system users, with each user completing the whole matching test.

This article adopts the golden test answers to evaluate the system users' skills and the difficulty of archival translation tasks and computes the maximum likelihood estimation of these parameters with a gradient descent method. As mentioned above, by modeling the adaptive archive allocation system and the archival translation tasks, these parameters can be estimated with few figures.

Suppose every system user will be assigned to any of the 15 test questions in each round. After each round, for each policy, by applying the same EM process to the users' answers, each question's most possible answer is computed. Based on the prediction, the accuracy of each policy in each round can be calculated.

Figure 1 shows the performance comparison of the adaptive policies with the votes observed in the live experiment. This article selects randomly 80 users from the 120 users to do simulation experiments many times, getting the means of the results and then computing the probability. Firstly, it asks each system user to answer each question to acquire the maximum accuracy, computes the expected accuracy as part of the maximum accuracy, and then calculates the votes saved by the adaptive policies compared with the Circular policy. To get the same accuracy rate, all adaptive policies [22] need less than 50% of the Circular policy's votes to get 95% of the maximum accuracy.

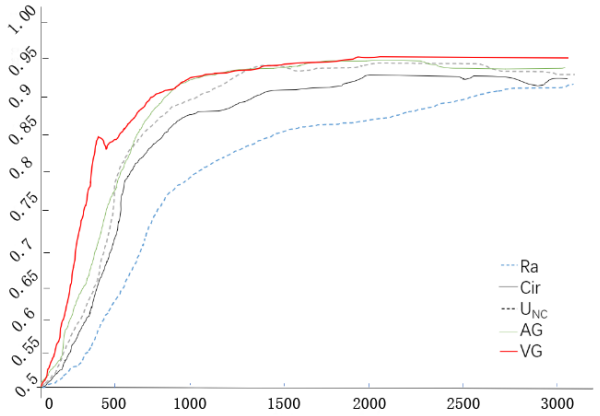

FIGURE 1. Comparison of votes by different policies and the expected accuracy function.

In Figure 1, compared with Cir policy, ALGO_{AD} (VG) needs only 48% of the Cir policy's votes to achieve 95% of the maximum accuracy.

This article also uses the parameters estimated from the live experiment to generate simulation data for the 30 test questions and 120 systems users to distinguish the adaptive policies by randomly sampling the system users' skills and the difficulty of archival translation tasks. Figure 2 displays the experiment results. To achieve 95% to 97% of the total accuracy (two-tailed paired test, $p<0.0001$), on the part of the votes saved, VG is better than AG, while AG is better than U_{NC}, compared with the Cir policy. The results also show the following three points:

ALGO_{AD} (VG) saves the system users the most, followed by AG and U_{NC}.

All these three adaptive algorithms perform better than non-adaptive baselines.

Though the ALGO_{AD} (VG) wins in all aspects, the other variants of ALGO_{AD} (AG and U_{NC}) perform quite well.

In Figure 2, the experiment with simulation data of the system users' skills and the difficulty of archival translation

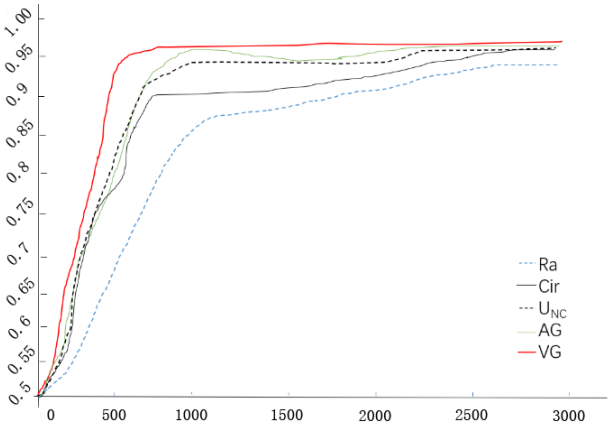

FIGURE 2. Comparison of votes by different policies and the expected accuracy function.

tasks estimated from the live experiment shows that $ALGO_{AD}$ (VG) performs better than AG and U_{NC}.

This article also compares these policies in the $ALGO_{RT}$ setting, shown in Figure 3. To simulate the completion time system users need in fulfilling tasks, this article adjusts the completion time observed with log normal allocation and samples from the allocation. Though there are minor relative gains, the ranking of these policies is consistent with that round-based setting, proving the methods proposed in this article can be extended to authentic settings where the time system users spend in completing tasks is different.

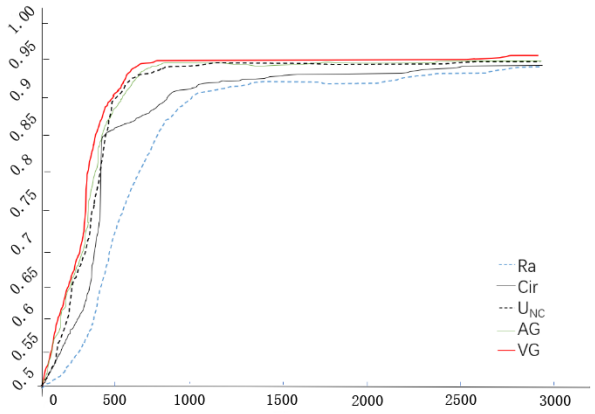

FIGURE 3. Comparison of votes by different policies and the expected accuracy function in $ALGO_{RT}$ setting.

As Figure 3 shows, in the live experiment where the time system users spend in fulfilling tasks is different, $ALGO_{RT}$ (VG) performs better than other policies even though the system needs to allocate tasks to users in real-time when they finish fulfilling tasks.

Figure 4 summarizes the relative savings of the policies proposed in this article in the situations mentioned above – the actual votes, simulation votes, and the simulation votes for changeable completion time. Due to VG's obvious cross-scene advantages, the following section will describe its performance with more simulation.

Figures 4 shows the comparison of savings among VG, AG and U_{NC} compared with the Cir policy to achieve 95% of the total accuracy.

3) THE BENEFITS OF ADAPTIVE POLICIES

To determine the benefits of adaptive policies, by randomly sampling the system users' skills and the difficulty of archival translation tasks, this article generates the simulation data of 15 test questions and 120 system users.

Regarding the expected accuracy as part of the maximum accuracy obtained by each user fulfilling one archival translation task, Figure 5 shows the relative savings of $ALGO_{AD}$ and $ALGO_{OFF}$ compared with the Cir policy. The simulation in this article defines the task difficulty uniformly, setting the system users' skills to be the reciprocal $1/\gamma \sim \mathcal{N}$

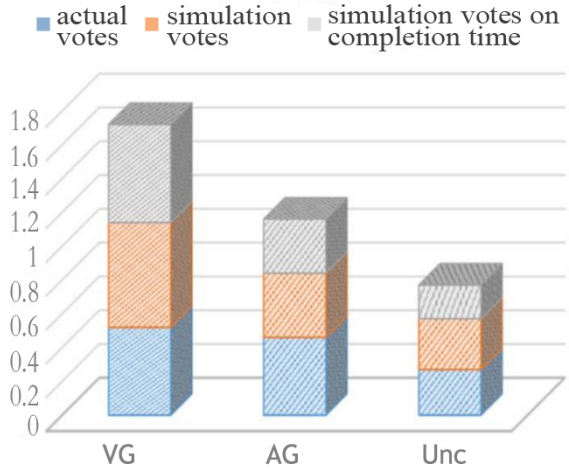

FIGURE 4. The comparison of savings among VG, AG and U_{NC}.

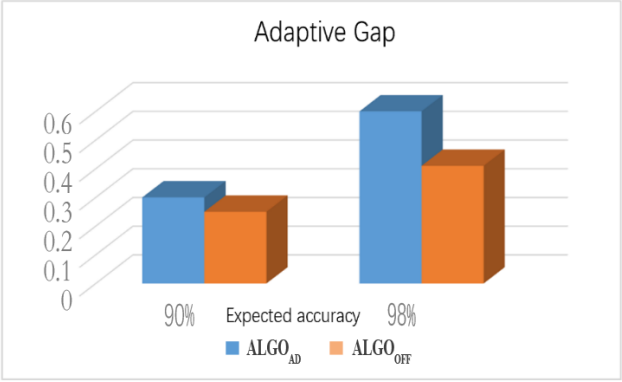

FIGURE 5. The comparison of savings between $ALGO_{AD}$ and $ALGO_{OFF}$ in different accuracy settings.

(0.79,0.29), which is feasible, satisfying the estimation of live experiments.

As Figure 5 displays, by adaptively observing and replying tasks, $ALGO_{AD}$ saves more system users than $ALGO_{OFF}$.

4) THE SYSTEM USERS' SKILLS AND THE DIFFICULTY OF ARCHIVAL TRANSLATION TASKS

This section investigates how different parameter allocations influence the performance of adaptive policies. With the simulation data of 15 test questions and 120 system users, as shown in Figure 6, and with the system users' skills more diverse, the votes $ALGO_{AD}$ saves increase compared with the Cir policy. Sampling again to do experiments with the same task difficulty and the previous system users' skill allocation, changing the standard deviation from $\sigma = 0$ to $\sigma = 0.2$. As mentioned above, the accuracy is still regarded as part of the maximum accuracy of the given group of system users. The larger the σ, the better the performance of the algorithms proposed in this article is, since they can make optimal allocation by taking advantage of system users' differences.

Figure 6 displays that the gains brought about by $ALGO_{AD}$ increase as the system users' skills are more diverse.

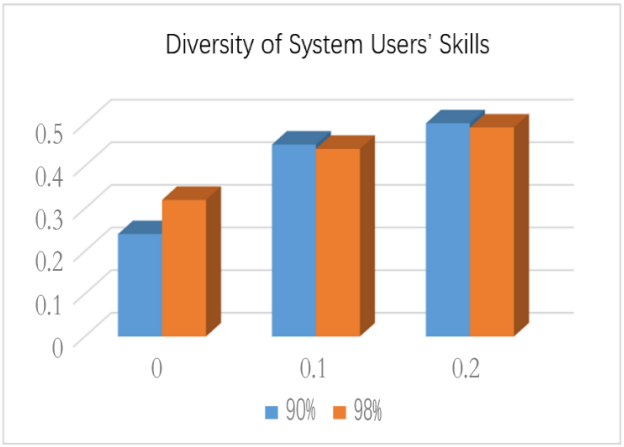

FIGURE 6. System users' skill diversity and the gain changes of $ALGO_{AD}$.

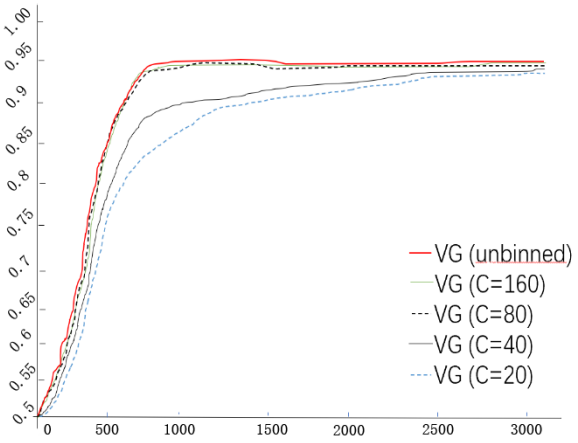

FIGURE 7. The influence of the increase of parameter C on the performance of $ALGO_{BIN}$.

Maintaining the skill allocation unchanged and increasing the difficulty diversity of tasks can also save votes. For instance, compared with the Cir policy, sampling the difficulty of archival translation tasks from the β allocation with a single peak saves much less than uniform sampling or sampling from bimodal allocation.

5) THE CONTINUABILITY OF LARGER SCALES OF ARCHIVAL TRANSLATION TASKS AND SYSTEM USERS

To testify the speed and allocation performance of the $ALGO_{BIN}$ policy proposed in this article, we conduct data simulation with larger scales of system users and archival translation tasks (100 and 2000 respectively) with the same archival translation task difficulty and system users' skill setting as in previous experiments with adaptive policies. Without constraining the number of system users simultaneously performing a specific archival translation task, $ALGO_{AD}$ policy is not feasible for such scales of tasks. We first examine, in each round, whether it's the case that the limitation to the number of system users to the same archival task leads to the quality decrease of archival translation tasks. Limiting the number of system users who fulfil the same archival translation task to one, two or three does not lead to a statistically significant difference (at the significance level of 0.05, the single-tailed paired t-test was used to measure the votes saved to reach different thresholds).

Limiting the number of system users to each archival translation task to a smaller figure can reduce the allocation time $ALGO_{AD}$ needs to 10-15 seconds in each round (100 system users), which is longer than a fraction of a second, the allocation time for 120 system users and 15 test questions. For different numbers of system user sets ($C \in \{20, 40, 80, 160\}$) experimented in this article, $ALGO_{BIN}$ can allocate tasks to system users within 3 seconds. $ALGO_{BIN}$ adopts $O(|\mathcal{W}|^2)$ algorithm to allocate tasks, which performs much better than the non-partition $O(|\mathcal{W}||\Omega|)$ algorithm.

Figure 7 shows that increasing parameter C that dominates the number of partitions can extend $ALGO_{BIN}$ to approximate $ALGO_{OFF}$.

The smaller the partition set, the closer to the true utility value, thus when the value of parameter C increases, the performance of $ALGO_{BIN}$ is close to the non-partition algorithm. Figure 7 shows the experiment results. Compared with the non-partition method, the accuracy of $ALGO_{BIN}$ is slightly lower, proving that such complex methods as adaptive partition may also bring about more benefits.

VI. CONCLUSION

In the adaptive archive allocation system, it is important to allocate simple tasks to the unskilled and the difficult ones to experts. Since the users are not so patient, the system should allocate the archival translation tasks parallelly to all available system users to fulfill archival translation tasks. Unfortunately, it is difficult to optimize the allocation of archival translation tasks to users, even if the users' skills and difficulty are known.

This article introduces the adaptive archive allocation system, describing its space based on the adaptiveness, parallelism, and the amount of known information, and proves that the submodularity of the objective function (the expected value maximization of archival translation answers) can be rationally approximated even if the offline task allocation is NP-hard. This article introduces two algorithms, $ALGO_{OFF}$ and $ALGO_{AD}$, to perform offline and online parallel task allocation. The experiment with proper names matching task shows that the optimal allocation algorithm in the adaptive archive allocation system uses only 48% of users that the standard circular policy needs to achieve 95% of the maximum accuracy, significantly saving system users. Besides, this article also brings about $ALGO_{BIN}$ (a system that can be extended to larger scales of system users and archive allocation tasks) and $ALGO_{RT}$ (a suitable system for situations

when the time users spend in fulfilling archive tasks is different).

In the future adaptive archival task allocation, it is hoped to relax the perfect hypotheses about system users and archival tasks. At the same time, this article hopes to further deep learning [17] and new technology [18], [19] to study such situations as different adaptive archival task requesters use the same system and the allocation algorithms need to balance system users among different types of archival tasks.

REFERENCES

[1] *Modern Guangdong Customs Archive*, Guangdong Arch., 2019, pp. 1–261.
[2] E. Law and L. V. Ahn, "Human computation," *Synth. Lectures Artif. Intell. Mach. Learn.*, vol. 5, no. 3, pp. 1–121, 2011.
[3] Y. Lu, S. T. Maguluri, M. S. Squillante, and C. W. Wu, "Inventors; international business machines corp, assignee. Allocating resources among tasks under uncertainty," U.S. Patent Appl. 15 081 827, Sep. 28, 2017.
[4] S. J. Choi *et al.*, "Does majority voting improve board accountability?" *Univ. Chicago Law Rev.*, pp. 1119–1180, 2016.
[5] Y. Ertimur, F. Ferri, and D. Oesch, "Does the director election system matter? Evidence from majority voting," *Rev. Accounting Stud.*, vol. 20, no. 1, pp. 1–41, Mar. 2015.
[6] G. F. Cooper and E. Herskovits, "A Bayesian method for constructing Bayesian belief networks from databases," in *Uncertainty Proceedings 1991*. San Mateo, CA, USA: Morgan Kaufmann, 1991, pp. 86–94.
[7] A. C. Harberger, "Monopoly and resource allocation," in *Essential Readings in Economics*. London, U.K.: Palgrave, 1995, pp. 77–90.
[8] R. Rajkumar, C. Lee, J. Lehoczky, and D. Siewiorek, "A resource allocation model for QoS management," in *Proc. Real-Time Syst. Symp.*, 1997, pp. 298–307.
[9] M. B. Gawali and S. K. Shinde, "Task scheduling and resource allocation in cloud computing using a heuristic approach," *J. Cloud Comput.*, vol. 7, no. 1, p. 4, Dec. 2018.
[10] S. Ghobadi and S. Jahangiri, "Optimal allocation of resources using the ideal-solutions," *J. New Researches Math.*, vol. 5, no. 20, pp. 121–134, Sep. 2019.
[11] Q. Jiang and B. Huang, "Distributed monitoring for large-scale processes based on multivariate statistical analysis and Bayesian method," *J. Process Control*, vol. 46, pp. 75–83, Oct. 2016.
[12] S. Choochotkaew, H. Yamaguchi, and T. Higashino, "A selforganized task distribution framework for module-based event stream processing," *IEEE Access*, vol. 7, pp. 6493–6509, 2018.
[13] M. Makki Alamdari, B. Samali, J. Li, Y. Lu, and S. Mustapha, "Structural condition assessment using entropy-based time series analysis," *J. Intell. Mater. Syst. Struct.*, vol. 28, no. 14, pp. 1941–1956, Aug. 2017.
[14] J. A. Grant, D. S. Leslie, K. Glazebrook, R. Szechtman, and A. N. Letchford, "Adaptive policies for perimeter surveillance problems," *Eur. J. Oper. Res.*, vol. 283, no. 1, pp. 265–278, May 2020.
[15] K. L. Chan, Y. W. Si, and M. Dumas, "Simulation-based evaluation of workflow escalation strategies," in *Proc. IEEE Int. Conf. e-Business Eng.*, Oct. 2009, pp. 75–82.
[16] V.-I. Chan and Y.-W. Si, "Simulation-based evaluation of resource allocation strategies for archival management workflow: The Macau case," in *Proc. IEEE 7th Int. Conf. E-Business Eng. (ICEBE)*, Nov. 2010, pp. 338–344, doi: 10.1109/ICEBE.2010.31.
[17] J. Jiao, K. Venkat, Y. Han, and T. Weissman, "Maximum likelihood estimation of functionals of discrete distributions," *IEEE Trans. Inf. Theory*, vol. 63, no. 10, pp. 6774–6798, Oct. 2017.
[18] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM J. Optim.*, vol. 26, no. 3, pp. 1835–1854, Jan. 2016.
[19] W. Cao, Q. Liu, and Z. He, "Review of pavement defect detection methods," *IEEE Access*, vol. 8, pp. 14531–14544, 2020.
[20] R. Wang, M. Shen, T. Wang, and W. Cao, "L1-norm minimization for multi-dimensional signals based on geometric algebra," *Adv. Appl. Clifford Algebras*, vol. 29, no. 2, p. 33, Apr. 2019.
[21] H. Chen, G. Wu, W. Pedrycz, P. N. Suganthan, L. Xing, and X. Zhu, "An adaptive resource allocation strategy for objective space partition-based multiobjective optimization," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Mar. 12, 2019, doi: 10.1109/TSMC.2019.2898456.
[22] R. Ramírez-Velarde, A. Tchernykh, C. Barba-Jimenez, A. Hirales-Carbajal, and J. Nolazco-Flores, "Adaptive resource allocation with job runtime uncertainty," *J. Grid Comput.*, vol. 15, no. 4, pp. 415–434, Dec. 2017.
[23] A. Shirali, J. K. Kordestani, and M. R. Meybodi, "Self-adaptive multi-population genetic algorithms for dynamic resource allocation in shared hosting platforms," *Genetic Program. Evolvable Mach.*, vol. 19, no. 4, pp. 505–534, Dec. 2018.
[24] C. Huang, S. Lucey, and D. Ramanan, "Learning policies for adaptive tracking with deep feature cascades," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 105–114.

CHEN LILAN received the B.A. degree from Henan Normal University, in July 2001, and the M.A. and Ph.D. degrees from Sun Yat-sen University, in June 2007 and in June 2020, respectively. She is currently a Lecturer with the School of Foreign Languages, Guangdong Pharmaceutical University. Her current research interests include the integration of information resources and the digitalization of archives and language studies.

CHEN YONGSHENG was the Dean of the Institute of Big Data, Sun Yat-sen University, where he was an Assistant from 1983 to 1988, a Lecturer from 1988 to 1992, and an Associate Professor from 1992 to 1994. Since 1994, he has been a Professor of Sun Yat-sen University, where he is currently a Professor with the School of Information Management. His current research interests include the integration of information resources, digitalization of archive, information security, and secrecy management.