Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)

Collective Biobjective Optimization Algorithm for Parallel Test Paper Generation

Minh Luan Nguyen
Data Analytics Department
Institute for Infocomm Research
Singapore

Siu Cheung Hui
School of Computer Engineering
Nanyang Technological University
Singapore

Alvis C. M. Fong
School of Computing Science
University of Glasgow
Scotland, UK

Abstract

Parallel Test Paper Generation (k-TPG) is a biobjective distributed resource allocation problem, which aims to generate multiple *similarly optimal* test papers automatically according to multiple user-specified criteria. Generating high-quality parallel test papers is challenging due to its NP-hardness in maximizing the *collective objective functions*. In this paper, we propose a Collective Biobjective Optimization (CBO) algorithm for solving k-TPG. CBO is a multi-step greedy-based approximation algorithm, which exploits the submodular property for biobjective optimization of k-TPG. Experiment results have shown that CBO has drastically outperformed the current techniques in terms of paper quality and runtime efficiency.

1 Introduction

With the rapid growth of the Internet and mobile devices, Web-based education has become a ubiquitous learning platform in many institutions to provide students with online learning courses and materials through freely accessible educational websites such as Khan Academy, or online classes such as Coursera and Udacity. To make learning effective, it is important to assess the proficiency of the students while they are learning the concepts. As there may have many students in an online class [Martin, 2012], to ensure the assessment reliability of large-scale Web-based testing, pedagogical practitioners have suggested that it is necessary to compose multiple tests from a large question pool with equivalent properties. One promising approach to support large-scale Web-based testing is parallel test paper generation (k-TPG), which generates k *similarly optimal* test papers automatically according to a user specification based on multiple criteria. Specifically, it aims to find k disjoint subsets of questions from a question database to form multiple test papers according to a user specification based on total time, topic distribution, difficulty degree, discrimination degree, and so on.

k-TPG is a challenging problem especially with large number of questions and large number of generated test papers (i.e. large k) due to its NP-hardness. Different from other multiobjective optimization problems, k-TPG has *collective objective functions* to be defined based on the evaluation objectives of k generated test papers instead of a single test paper. In fact, k-TPG is close to the spirit of optimal distributed allocation problems [Feige and Vondrak, 2006] with collective objective functions. Formally, k-TPG is a biobjective optimal distributed resource allocation problem, which aims to simultaneously maximize two objective functions under a multidimensional Knapsack constraint. The first objective can be formulated as a Welfare Allocation problem [Vondrak, 2008], which aims to maximize the total quality of the generated test papers. The second objective can be formulated as a Fairness Allocation problem [Asadpour and Saberi, 2010; Bertsimas *et al.*, 2011], which aims to maximize the fairness quality of the generated test papers. Traditionally, optimizing these two objectives is NP-hard and considered separately. To the best of our knowledge, there has been no attempt to solve both of the objectives simultaneously.

Submodular-based approximation algorithms, which are polynomial time algorithms that have performance guarantees, have been studied for distributed optimization problems [Nedic *et al.*, 2010]. Inspired by this novel idea, we propose a greedy-based approximation algorithm, called Collective Biobjective Optimization (CBO), for k-TPG. The key idea of CBO is twofold. Firstly, it analyzes the properties of the two collective objectives and reformulates the k-TPG problem such that it can be solved effectively. Secondly, we propose an effective multi-step framework for the reformulated optimization problem. More importantly, we exploit submodular optimization techniques to design greedy-based approximation algorithms for enhancing each step in the proposed framework. In this paper, we will discuss the proposed CBO algorithm for k-TPG and its performance evaluation.

2 Related Work

For the past few years, heuristic-based intelligent techniques such as Tabu Search [Hwang *et al.*, 2008], Particle Swarm Optimization [Ho *et al.*, 2008] and Ant Colony Optimization [Hu *et al.*, 2009] have been proposed for k-TPG. However, the quality of the generated parallel test papers is often unsatisfactory [Hwang *et al.*, 2008; Ho *et al.*, 2008; Hu *et al.*, 2009] according to users' test paper specifications. The main issue of the current techniques is the exhaustive search to optimize the biobjective functions simultaneously in the very large space of possible candidates with multicriteria constraints. Although these heuristic-based techniques are straightforward to implement, they suffer from some drawbacks. They tend to get stuck in a local optimum as they are mainly based on traditional weighting parameters for multiobjective optimization [Ishibuchi *et al.*, 2008].

3 Parallel Test Paper Generation

Let $\mathcal{Q} = \{q_1, q_2, .., q_n\}$ be a dataset consisting of n questions, $\mathcal{C} = \{c_1, c_2, .., c_m\}$ be a set of m different topics, and $\mathcal{Y} = \{y_1, y_2, .., y_k\}$ be a set of k different question types. Each question $q_i \in \mathcal{Q}$, where $i \in \{1, 2, .., n\}$, has 6 attributes $\mathcal{A} = \{q, e, t, d, c, y\}$ defined as follows:

- *Question* q: It is used to store the question identity.
- *Discrimination degree* e: It is a positive integer value to distinguish user proficiency.
- *Question time* t: It is a positive integer value to indicate the average time in minutes needed to answer.
- *Difficulty degree* d: It is a positive integer value to indicate the question difficulty.
- *Related topic* c: It is a set of related topics of a question.
- *Question type* y: It is the type of a question.

A *test paper specification* $\mathcal{S} = \langle N, T, D, C, Y\rangle$ consists of 5 attributes including the number of questions N, total time T, average difficulty degree D, topic distribution $C = \{(c_1, pc_1), (c_2, pc_2), .., (c_M, pc_M)\}$ and question type distribution $Y = \{(y_1, py_1), (y_2, py_2), .., (y_K, py_K)\}$, which are defined based on the attributes of the selected questions.

In 1-TPG, it selects an optimal subset of questions to form a test paper P, which maximizes the average discrimination degree $f(P) = \frac{1}{N}\sum_{i=1}^{N} e_i, q_i \in P$ while satisfying multiple user-specified criteria in a test paper specification $\mathcal{S}_P = \mathcal{S}$.

In k-TPG, user specification is given as a pair of $\langle k, \mathcal{S}\rangle$, where k is an integer parameter in addition to the test paper specification $\mathcal{S}$. The k-TPG aims to select k disjoint subsets of questions $P_1, P_2, .., P_k$ from the question dataset $\mathcal{Q}$ to form k test papers with specification $\mathcal{S}_{P_i}$, which satisfies the test paper specification such that $\mathcal{S}_{P_i} = \mathcal{S}$, $\forall i \in [1, k]$. Apart from maximizing the objective function as in single test paper generation, k-TPG aims to generate the k test papers with similar optimal quality according to the specification. It aims to provide similar quality of test papers for different users. As such, there are two important *collective objective functions* that need to be maximized in k-TPG: *total quality maximization* and *fairness quality maximization*.

Total Quality Maximization. It aims to maximize the sum of discrimination degrees of k test papers $\sum_{i=1}^{k} f(P_i)$, where $f(P_i)$ is the discrimination degree of test paper $P_i, i \in [1, k]$. Formally, the k-TPG problem is stated as follows:

$$\underset{P_1,..,P_k}{\text{maximize}} \sum_{i=1}^{k} f(P_i) \tag{1}$$

Fairness Quality Maximization. It aims to maintain the fairness among the generated test papers with equivalent discrimination degrees. However, solely maximizing the total quality objective does not guarantee the fairness quality. According to [Asadpour *et al.*, 2008; Bertsimas *et al.*, 2011] , it is more effective to optimize a fair allocation of the discrimination degrees of k generated test papers as:

$$\underset{P_1,..,P_k}{\text{maximize}} \min_{1 \le i \le k} f(P_i) \tag{2}$$

subject to the local multicriteria constraints:

$$P_i \cap P_j = \emptyset, \forall i \neq j \tag{3}$$

$$\mathcal{S}_{P_i} = \mathcal{S} \tag{4}$$

4 Proposed CBO Approach

To simultaneously optimize the total quality and fairness quality objective functions is a challenging problem. However, by exploiting the relationship between the two objective functions based on the submodular property [Lovasz, 1983], we can devise an effective and efficient approach.

4.1 Problem Reformulation

Before reformulating the k-TPG problem, we review the fundamental concept of submodular function in optimization.

Definition 1 (Submodular Function Optimization). Given a set X, a non-negative function $f : 2^X \to \mathbb{R}^+$, $T \subseteq X$, and $x \in X$. Let $\frac{\partial f(S)}{\partial x} = \triangle_f(x|S) = f(S \cup \{x\}) - f(S)$ be the *decreasing marginal value* of f at S with respect to x. A non-negative function $f : 2^X \to \mathbb{R}^+$ is *submodular* if for every set $S, T \subseteq X$, we have:

$$f(S \cup T) + f(S \cap T) \le f(S) + f(T)$$

Although NP-hard, the submodular property has led to the existence of a Greedy-based Approximation Algorithm [Fujishige, 2005] for the maximization. It is widely known that a linear function as well as the combination of linear functions are aslo submodular [Fujishige, 2005]. [Nguyen *et al.*, 2013] shows that 1-TPG can be reformulated as an integer linear program which aims to maximize a linear objective function $f(P)$ under multiple knapsack constraints. Hence, the quality function $f(P)$ is submodular due to the linearity property.

Lemma 1. Given a test paper P generated from a question set $\mathcal{Q}$ ($P \subseteq \mathcal{Q}$), the quality evaluation function of discrimination degree $f(P) : 2^{\mathcal{Q}} \to \mathbb{R}^+$ is submodular and monotone.

Hence, the total quality objective function is also submodular due to a linear combination of k submodular functions.

Corollary 1. The total quality objective function of k test papers $\sum_{i=1}^{k} f(P_i), P_i \subseteq \mathcal{Q}$ is submodular and monotone.

To jointly optimize the biobjective functions, we take the advantage of submodularity to reformulate the total quality objective such that it also integrates the fairness objective.

Let $f_\phi(P) = \min\{f(P), \phi\}$ be a truncated function, where ϕ is a non-negative constant. Note that $f_\phi(P)$ is also submodular and monotone due to [Fujishige, 2005]. For any constant ϕ, we have the following important observation:

$$\min_{l \in \{1,..,k\}} f(P_l) \ge \phi \iff \sum_{l=1}^{k} f_\phi(P_l) = k\phi \tag{5}$$

As k is a constant, this means that the fairness quality objective value of a paper is larger than or equal to ϕ if the total quality objective value is $k\phi$ and vice versa. Therefore, we have the following result in Theorem 1:

Theorem 1. Given the optimal fairness value ϕ^*, i.e., $\max_{P_1,..,P_k} \min_l f(P_l) = \phi^*$, then it is sufficient to jointly optimize the two objective functions of the original k-TPG problem by solving the following equivalent problem:

g-17pt

$$\underset{P_1,..,P_k}{\text{maximize}} \sum_{l=1}^{k} f_{\phi^*}(P_l) \text{ s.t. } P_i \underset{\forall i \neq j}{\cap} P_j = \emptyset \;,\; \mathcal{S}_{P_i} = \mathcal{S} \quad (6)$$

Proof Sketch: The original k-TPG problem is equivalent to the problem given in (6) since, by (5), there exists k test papers $P_1, .., P_k$ such that $\sum_{l=1}^{k} f_{\phi^*}(P_l) = k\phi^*$. In addition, because of the property of the truncated function $f_{\phi^*}(P_l)$, we have $\max \sum_{l=1}^{k} f_{\phi^*}(P_l) \leq k\phi^*$. Thus, for any optimal generated papers $P_1, .., P_k$ of the problem in (6), it must satisfy that $\sum_{l=1}^{k} f_{\phi^*}(P_l) = k\phi^*$. Hence, we have $\min_l f(P_l^*) = \phi^*$ due to (5).

4.2 Collective Biobjective Optimization

In this paper, we propose a novel Collective Biobjective Optimization (CBO) approach for k-TPG. We observe that the reformulated problem (6) can be solved effectively by an iterative multi-step framework. More importantly, we exploit the submodular optimization technique to design greedy-based approximation algorithm for enhancing the steps in the proposed approach. Figure 1 shows the 5 main steps of the CBO approach: Optimal Fairness Value Estimation, Infeasible Allocation Detection, Total Quality Maximization, Fairness Quality Balancing and Local Constraint Improvement.

In the first step, after each feedback loop, CBO either: (1) reduces half of the search space on ϕ if Infeasible Allocation Detection fails or (2) finds a better solution in terms of both of the objectives. In the remaining 4 steps, CBO will generate k test papers progressively by using the greedy-based algorithm for 1-TPG and adjusting the fairness value ϕ. Due to NP-hardness of 1-TPG, when ϕ approaches its optimal value, we can only achieve a fraction $\beta \leq 1$ of the optimal objective value of ϕ, where β is a constant that will be determined in Section 4.6. As such, our goal is to allocate questions into k test papers such that for all k papers, we have $f_\phi(P_l) \geq \beta\phi, \forall l = 1..k$.

4.3 Optimal Fairness Value Estimation

In each iteration, a new set of k papers will be generated according to a new fairness value ϕ. This step aims to find a new optimal value of ϕ to continue the co-optimization process. Generally we do not know the optimal value ϕ^*. Therefore, we need to search for it using the binary search strategy, starting with the interval $[\phi_{min}, \phi_{max}] = [0, N \times \max_{q \in \mathcal{Q}} f(q)]$. In each step, we test the center $\phi = (\phi_{min} + \phi_{max})/2$ of the current interval $[\phi_{min}, \phi_{max}]$ of possible fairness value.

4.4 Infeasible Allocation Detection

This step aims to detect whether it is possible to generate k test papers such that $f_\phi(P_l) \geq \beta\phi, \forall l = 1..k$. Specifically, it checks whether the estimated value of ϕ is appropriate or not. To overcome this, we propose a greedy-based approximation

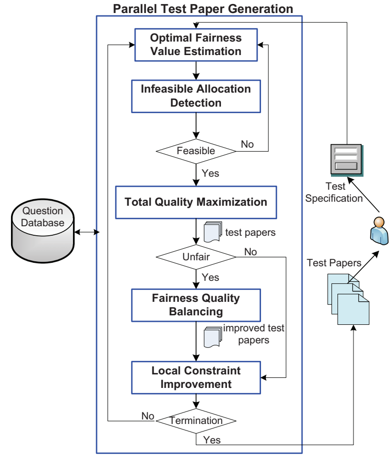


Figure 1: Proposed CBO Approach

Algorithm 1: Infeasible_Allocation_Detection

Process:

$P_l \leftarrow \emptyset$ for all $l = 1..k$;
for *i=1 to k*N* **do**
  foreach *question* $q \in \mathcal{Q} \setminus (P_1 \bigcup ... \bigcup P_k)$ ***and*** $l \in \{1..k\}$ **do**
    if *q satisfies the cardinality constraint and local constraints* P_l **then**
      Compute the improvement:
      $\Psi_{l,q} \leftarrow w(P_l + \{q\}) - w(P_l)$;
  $(l^*, q^*) \leftarrow \text{argmax}_{(l,q)} \Psi_{l,q}$;
  $P_{l^*} \leftarrow P_{t^*} \bigcup \{q^*\}$;
  $\xi = \sum_{l=1}^{k} f_\phi(P_l)$;
  if $\xi < \frac{1}{2}k\phi$ **then return** ***Yes*** **else return** ***No***

algorithm that generalizes the basic greedy-based approximation algorithm for submodular function. For the sake of simplicity, we consider only the topic and question type constraints. As such, the result obtained is the upper bound of the actual approximate solution. However, it is sufficient for the early infeasible allocation detection purpose.

The algorithm is outlined in Algorithm 1. In each iteration, we first check that whether the question q can be possibly allocated to a paper P_t. To verify whether q can be possibly allocated to P_t, we first check whether P_t is full, i.e., $|P_t| = N$. Then, we check whether allocating q to P_t will violate the two constraints. If yes, we then compute the marginal improvement value $\Psi_{t,q}$ of allocating question q into P_t. Next, we greedily choose the best allocation of test paper and question (P_{t^*}, q^*) with the maximum $\Psi_{t,q}$ value.

We provide a theoretical result of this step. For general cases, i.e., $k \geq 2$, we can prove that it achieves a weaker approximation ratio of $\frac{1}{2}$.

Theorem 2. Consider only the content constraints with $k \geq 2$, Algorithm 1 can obtain a set of generated test papers $P_1, .., P_k$ with the total quality value such that:

$$\sum_{i=1}^{k} f_\phi(P_i) \geq \frac{1}{2} \max_{P'_1,..,P'_k} \sum_{i=1}^{k} f_\phi(P'_i) = \frac{1}{2} OPT$$

where OPT refers to the optimal value.

Proof Sketch: Let H be the original problem of allocating $k * N$ out of n questions to k papers $P_1, .., P_k$ such that it maximizes the total quality objective $\sum_{i=1}^{k} f_\phi(P_i)$. Let H' be the problem on the $n-1$ remaining questions after the first question q_t is selected for paper P_j. On the problem H', the quality evaluation function $f_j(P_j) = f_\phi(P_j)$ is replaced by $f'_j(P_j) = f_j(P_j \cup \{q_t\}) - f_j(\{q_t\})$. All other quality evaluation function $f_i(P_i)$, $i \neq j$, are unchanged.

Let $VAL(H)$ be the value of the allocation produced by Algorithm 1 and $OPT(H)$ be the value of the optimal allocation. Let $p = f_j(\{q_t\})$. By definition of H', it is clear that $VAL(H) = VAL(H') + p$. By submodularity, we can show that $OPT(H) \leq OPT(H') + 2p$. This proof is completed by induction on H' since H' is also a submodular function:

$$OPT(H) \leq OPT(H') + 2p \leq 2VAL(H') + 2p = 2VAL(H)$$

Theorem 2 means that Algorithm 1 can achieve 1/2-approximation ratio for the total quality maximal objective based on a given fairness value ϕ. Based on this, we can decide whether it is feasible to allocate k papers corresponding to the current fairness ϕ.

4.5 Total Quality Maximization

It progressively solves k problem instances of 1-TPG to generate k papers $P_1, P_2, .., P_k$. Based on the submodular property of $f_\phi(P)$, we greedily select questions that maximize the objective function while paying attention to satisfying the multiple knapsack constraints. This is motivated by an algorithm that maximizes a submodular function under a knapsack constraint: $\max_S f(S)$ s.t. $\sum_{x \in S} c(x) \leq b$ where $c(x) \geq 0$ is a cost function and $b \in \mathbb{R}$ is a budget constraint. Sviridenko [Sviridenko, 2004] proposed the greedy-based algorithm for solving this problem with an approximation ratio of $1 - \frac{1}{e}$. It defines the marginal gain $\frac{\Delta_f(x|S)}{c(x)}$ when adding an item x into the current solution S as a ratio between the discrete derivative $\Delta_f(x|S)$ and the cost $c(x)$. This algorithm starts with $S = \emptyset$, and then iteratively adds the element x that maximizes the marginal gain ratio among all elements that satisfy the remaining budget constraint:

$$S_{i+1} = S_i \cup \{ \operatorname*{arg\,max}_{x \in V \setminus S_i : c(x) \leq b - c(S_i)} \frac{\Delta(x|S_i)}{c(x)} \}$$

We extend the above algorithm of submodular function maximization under a knapsack constraint for solving the case of multiple knapsack constraints. Algorithm 2 gives the greedy-based algorithm for this step. This algorithm maintains a set of weights that are incrementally adjusted in a multiplicative manner. These weights capture the possibility that each constraint is close to be violated when maximizing the objective function of generating a paper. The algorithm is based on a main loop in Line 4. In each iteration, the algorithm selects an available question that maximizes the sum of marginal gain ratio normalized by the weights as follows:

$$P_l^{i+1} = P_l^i \cup \{ \operatorname*{arg\,max}_{q_j \in \mathcal{Q}} \sum_{i=1}^{m} \frac{\Delta_{f_\phi}(q_j|P_l^i)}{A_{ij} h_i} \}$$

Algorithm 2: Total_Quality_Maximization

Process:
1 **for** *l=1 to k* **do**
2   $P_l \leftarrow \emptyset$;
3   **for** *i=1 to m* **do** $h_i = 1/b_i$;
4   **while** $\sum_{i=1}^{m} b_i h_i \leq \mu$ *and* $|P_l| \leq N$ **do**
5     $q_j \leftarrow \operatorname*{arg\,max}_{q_j \in \mathcal{Q}} \sum_{i=1}^{m} \frac{\Delta_{f_\phi}(q_j|P_l)}{A_{ij} h_i}$;
6     $P_l \leftarrow P_l \cup \{q_j\}$;
7   **for** *i=1 to m* **do** $h_i \leftarrow h_i \mu^{\frac{A_{ij}}{b_i}}$;
8     $\mathcal{Q} \leftarrow \mathcal{Q} \setminus \{q_j\}$
9 **return** $P_1, P_2, .., P_k$

where A is a matrix and b is the vector of multiple knapsack constraints. Here, $\Delta_{f_\phi}(q_j|P_l)$ is the submodular incremental marginal value of question q_j to the paper P_l.

The parameter μ in Algorithm 2 is important as it ensures that multiple knapsack constraints will not be violated while maximizing the submodular objective function. Let $H = \min\{b_i/A_{ij} : A_{ij} > 0\}$ be the width of the knapsack constraints. By setting the parameter $\mu = e^H m$, Algorithm 2 can achieve near-optimal results. [Azar and Gamzu, 2012] has proved the following theoretical result:

Lemma 2. Algorithm 2 (Line 2-8) attains an approximation guarantee such that

$$f_\phi(P_i) \geq (1 - \frac{1}{e}) f_\phi(P_i^*) \approx f_\phi(P_i^*) > \frac{1}{2} f_\phi(P_i^*)$$

where P_i^* refers to the optimal solution of each main loop.

Corollary 2. The total quality objective function of k test papers $P_1, .., P_k$ attained by Algorithm 2 satisfies:

$$\sum_{i=1}^{k} f_\phi(P_i) \geq \frac{1}{2} \max_{P_1^*, .., P_k^*} \sum_{i=1}^{k} f_\phi(P_i^*) = \frac{1}{2} OPT$$

Proof Sketch: Let P_i^* be the optimal solution corresponding to each main loop of Algorithm 2. By using Lemma 2, we can directly derive the result.

4.6 Fairness Quality Balancing

This step aims to employ a novel fairness balancing strategy to improve the fairness quality objective. Recall that our goal is to allocate questions into k test papers such that for all papers, we have $f_\phi(P_l) \geq \beta\phi, \forall l = 1..k$. However, this is not easy to achieve especially when ϕ approaches the optimal value. This is because in the small and less abundant pool of questions, Algorithm 2 greedily selects all questions with high value $f_\phi(q)$ for some of the generated papers $P_1, P_2, .., P_s$ to satisfy $f_\phi(P_l) \geq \beta\phi, \forall l = 1..s$. As a result, the subsequent papers $P_{s+1}, .., P_k$ might not have enough good questions to satisfy this requirement. We need to swap some of the questions from the satisfied papers with questions from unsatisfied papers to achieve a fair allocation.

We will move questions from satisfied papers to unsatisfied papers and vice versa, until all papers satisfy the requirement $f_\phi(P_l) \geq \beta\phi, \forall l = 1..k$. Let's define the *swapping operation*

Algorithm 3: Fairness_Quality_Balancing

Process:

while $\exists i, j : f_\phi(P_i) \geq 3\beta\phi; f_\phi(P_j) < \beta\phi$ **do**

  foreach $q^i \in P_i$ **do**

    $P_j \leftarrow \{P_j \setminus \{q_j\}\} \cup \{q_i\}$;

    $P_i \leftarrow \{P_i \setminus \{q_i\}\} \cup \{q_j\}$;

    if $f_\phi(P_j) \geq \beta\phi$ **then** break;

return $P_1, P_2, .., P_k$

as follows. Select a satisfied paper $P_i = \{q_1^i, q_2^i, ...q_N^i\}$ for which $f_\phi(P_i) \geq 3\beta\phi$. Such a paper is always ensured through appropriate choice of α and β as shown in Lemma 3. Then, we select an unsatisfied paper P_j, i.e., $f_\phi(P_j) \leq \beta\phi$. In the paper P_i, choose $t < N$ such that $f_\phi(\{q_1^i, q_2^i, ...q_{t-1}^i\}) < \beta\phi$, $f_\phi(\{q_1^i, q_2^i, ...q_t^i\}) \geq \beta\phi$, and $f_\phi(\{q_t^i\}) < \beta\phi$. Let $\Lambda_i = \{q_1^i, q_2^i, ...q_t^i\}$. As $f_\phi(\emptyset) = 0$, the set Λ_i is not empty. In the reversed direction, let $\Lambda_j = \{q_1^j, q_2^j, ...q_t^j\}$ be a set of questions in P_j such that each pair of questions $q_l^i, l = 1..t$, and $q_l^j, l = 1..t$, has the same topic and question type. We reallocate the questions of papers P_i and P_j by swapping questions of the two sets Λ_i and Λ_j. We note that the swapping operation does not violate the multiple knapsack constraints. More importantly, it improves the fairness among all papers. The swapping operation is given in Algorithm 3.

By some analysis, we can prove that swapping questions does not decrease the value of $f_\phi(P_i)$ by more than $2\beta\phi$. Thus, P_i is still satisfied. Moreover, the previously unsatisfied test paper becomes satisfied by this operation. We can also make sure that we can always perform this operation until all test papers are satisfied by choosing $\beta = \alpha/3$, where $\alpha = 1/2$ is the approximation ratio of Algorithm 1.

Lemma 3. If we choose $\beta = \alpha/3 = 1/6$, it is guaranteed that after at most t swapping operations, all test papers will be satisfied, i.e., $f_\phi(P_l) \geq \beta\phi, \forall l = 1..k$.

Lemma 3 ensures that there always exists a test paper such that $f_\phi(P_l) \geq 3\beta\phi$. It also ensures that all test papers will be satisfied after at most t swaps.

4.7 Local Constraint Improvement

So far, we have achieved a set of k test papers, having guarantee on the biobjective values and satisfying the multiple knapsack constraints. However, these test papers have not yet satisfied the total time and difficulty degree constraints.

Here, we propose an effective method for satisfying these constraints. Given a test paper $P^0 = P_l = \{q_1, q_2, .., q_N\}, l = 1..k$, let $q = \{0, 1\}^n, |q| = N$ be the binary representation of the initial solution P. From the previous step, we have $\mathbf{A}q^0 \leq \mathbf{b}$, which is the 0-1 ILP formulation of the corresponding 1-TPG problem. This step aims to turn the existing solution q^0 to q^1 such that $\mathbf{A}q^1 = \mathbf{b}$. This is the Subset Vector Sum problem, which is NP-hard. Here, we only need to optimize the total time and difficulty degree constraints. Let $\sum_{l=1}^n a_{il}q \leq b_i$ be the total time constraint and $\sum_{l=1}^n a_{jl}q \leq b_j$ be the difficulty degree constraint. To optimize constraint satisfaction while ensuring the objective value $f(P_l)$ will not decrease, we reformulate the 1-TPG problem

	#Questions	#Topics	#Question Types	Distribution
D_1	20000	40	3	uniform
D_2	30000	50	3	normal
D_3	40000	55	3	uniform
D_4	50000	60	3	normal

Table 1: Test Datasets

to unconstrained submodular optimization by introducing two weighting parameters λ_1 and λ_2:

$$f_\xi(P) = f(P) - \lambda_1 |\sum_{l=1}^n a_{il}q - b_i| - \lambda_2 |\sum_{l=1}^n a_{jl}q - b_j|$$

It is not difficult to show that the problem $\max f_\xi(P_l)$ is an unconstrained non-monotone submodular maximization, which can be solved by an effective deterministic approximation local search proposed by Feige et al. [Feige *et al.*, 2007].

4.8 Termination

The process is repeated until the termination condition is reached: $\phi_{max} - \phi_{min} \leq \delta$, where δ is a predefined threshold. An optimal value of $\delta = 0.05$ was found experimentally.

4.9 Theoretical Analysis

We summarize the approximation results of the proposed CBO approach for biobjective optimization of k-TPG.

Theorem 3. The proposed CBO approach has achieved the following theoretical biobjective approximation results of the total quality maximization and fairness quality maximization:

$$\sum_{i=1}^k f(P_i) \geq \frac{1}{2} \max_{P_1', .., P_k'} \sum_{i=1}^k f(P_i') = \frac{1}{2} OPT$$

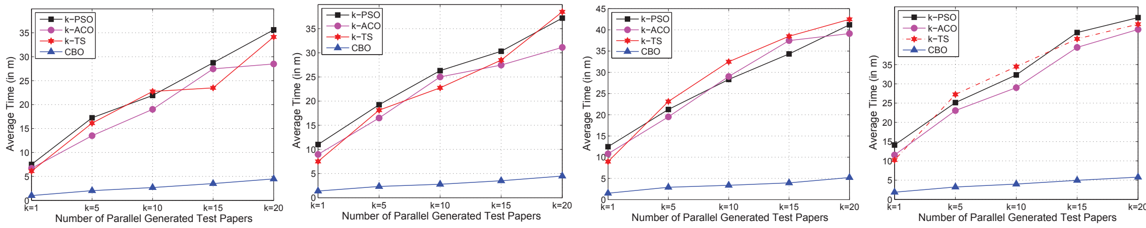
$$\min_{l=1..k} f(P_l) \geq \frac{1}{6} \max_{P_1', .., P_k'} \min_{l=1..k} f(P_l')$$

5 Performance Evaluation

The performance of CBO is compared with other techniques including the following 3 re-implemented algorithms for k-TPG: k-TS [Hwang *et al.*, 2008], k-PSO [Ho *et al.*, 2008] and k-ACO [Hu *et al.*, 2009] based on the published articles.

As there is no benchmark data available, we generated 4 large-sized synthetic datasets, namely D_1, D_2, D_3 and D_4, for performance evaluation. In datasets D_1 and D_3, the value of each attribute is generated according to a uniform distribution. On the other hand, in datasets D_2 and D_4, the value of each attribute is generated according to a normal distribution. Table 1 summarizes the 4 datasets.

We analyze the quality and runtime efficiency of the CBO approach for k-TPG based on the 4 large-scale datasets by using different specifications. Here, we have designed 12 test specifications in the experiments. We vary the parameters in order to have different test criteria in the test specifications. The number of topics is specified between 2 and 40. The total time is set between 20 and 240 minutes, and it is also set proportional to the number of selected topics for each specification. The average difficulty degree is specified randomly between 3 and 9. We have conducted the experiments according to 5 different values of k, i.e., $k = 1, 5, 10, 15, 20$.



(a) Dataset D_1 (b) Dataset D_2 (c) Dataset D_3 (d) Dataset D_4

Figure 2: Performance Results Based on Average Runtime

To evaluate the quality for k-TPG, we use the common metrics including the Average Discrimination Degree and Average Constraint Violation for 1-TPG [Nguyen *et al.*, 2013] based on the same user specification $\mathcal{S}$. From the common metrics, for a given dataset $\mathcal{D}$ and k, we define the following metrics for k-TPG based on the average and deviation of the quality of generated test papers: the Average Discrimination Degree $\mathcal{M}_d^{k,\mathcal{D}}$, the Deviate Discrimination Degree $\mathcal{V}_d^{k,\mathcal{D}}$, the Average Constraint Violation $\mathcal{M}_c^{k,\mathcal{D}}$ and the Deviate Constraint Violation $\mathcal{V}_c^{k,\mathcal{D}}$. Specifically, given a dataset $\mathcal{D}$ and for each k, we generate 12 test paper sets $P_1, P_2, ..., P_k$ from the 12 specifications. For each of the 12 test paper sets, we measure the mean and deviation of quality of k test papers according to the discrimination degree and constraint violation. Finally, we obtain the performance results based on the 4 measures by averaging the means and deviations over the 12 test paper sets.

For high quality parallel test papers, the Average Discrimination Degree should be high and the Average Constraint Violation should be small. Similarly, the Deviate Discrimination Degree and the Deviate Constraint Violation should be small.

5.1 Performance Results based on Runtime

Figure 2 compares the runtime performance of the 4 techniques based on the 4 datasets. The results have clearly shown that the proposed CBO approach significantly outperforms the other heuristic techniques in runtime for the different datasets. CBO generally requires less than 6 minutes to complete the parallel paper generation process. Moreover, the proposed CBO approach is quite scalable in runtime on different dataset sizes and distributions. In contrast, the other techniques are not efficient to generate high quality parallel test papers. Particularly, the runtime performance of these techniques degrades quite badly as the dataset size or the number of generated parallel test papers gets larger, especially for imbalanced datasets D_2 and D_4.

5.2 Performance Results based on Quality

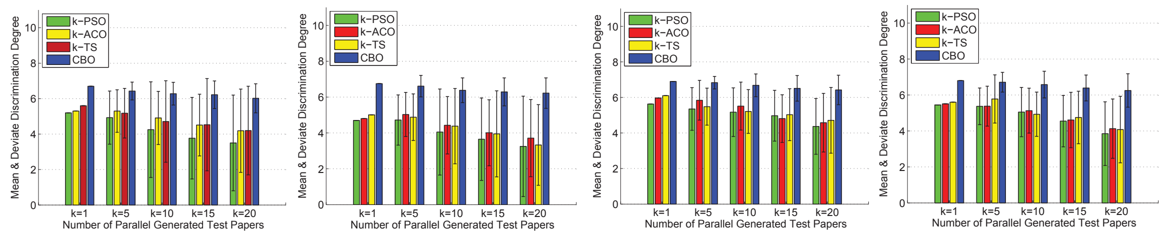
Figure 3 shows the quality performance results of the 4 techniques based on the Mean Discrimination Degree $\mathcal{M}_d^{k,\mathcal{D}}$ and Deviate Discrimination Degree $\mathcal{V}_d^{k,\mathcal{D}}$. As can be seen from Figure 3, CBO has consistently achieved higher Mean Discrimination Degree $\mathcal{M}_d^{k,\mathcal{D}}$ and lower Deviate Discrimination Degree $\mathcal{V}_d^{k,\mathcal{D}}$ than the other heuristic k-TPG techniques for the generated parallel test papers. Particularly, CBO can generate high quality test papers with $\mathcal{M}_d^{k,\mathcal{D}} \approx 7$. Note that the lower Deviate Discrimination Degree $\mathcal{V}_d^{k,\mathcal{D}}$ value indicates that the generated parallel test papers have similar quality in terms of discrimination degree. Generally, for a specific value of k, we observe that the quality of the generated parallel test papers of all 4 techniques based on the Mean Discrimination Degree $\mathcal{M}_d^{k,\mathcal{D}}$ and the Deviate Discrimination Degree $\mathcal{V}_d^{k,\mathcal{D}}$ tend to be improved when the dataset size gets larger.

Figure 4 gives the quality performance results of the 4 techniques based on the Mean Constraint Violation $\mathcal{M}_c^{\mathcal{D},k}$ and Deviate Constraint Violation $\mathcal{V}_c^{\mathcal{D},k}$. We also observe that CBO has consistently outperformed the other techniques on Mean Constraint Violation $\mathcal{M}_c^{\mathcal{D}}$ and Deviate Constraint Violation $\mathcal{V}_c^{\mathcal{D},k}$ based on the 4 datasets. The Mean Constraint Violation of CBO tends to decrease whereas the Mean Constraint Violations of the other 3 techniques increase quite fast when the dataset size or the number of specified constraints gets larger. In particular, CBO can generate high quality parallel test papers with $\mathcal{M}_c^{\mathcal{D},k} \leq 10$ for all datasets. Also, CBO is able to generate higher quality parallel test papers on larger datasets while the other techniques generally degrade on the quality of the generated test papers when the dataset size gets larger. In addition, we find that when k gets larger, the Mean Constraint Violation $\mathcal{M}_c^{k,\mathcal{D}}$ on a specific dataset $\mathcal{D}$ tends to decrease. Similarly, the Deviate Constraint Violation $\mathcal{V}_c^{k,\mathcal{D}}$ quality tends to increase when k gets larger. These results have shown that the quality based on constraint violation on a specific dataset $\mathcal{D}$ tends to degrade when k gets larger.

The good performance of CBO is due to 2 main reasons. Firstly, as CBO is an approximation algorithm with constant performance guarantee, it can find the near-optimal solution for objective functions of k-TPG effectively and efficiently while satisfying the multiple constraints without using weighting parameters. As such, CBO can achieve better paper quality and runtime efficiency as compared with other heuristic-based k-TPG techniques. Secondly, CBO is a submodular greedy-based algorithm, which is able to produce good solution in efficient polynomial runtime. Thus, CBO can also improve its computational efficiency on large-scale datasets as compared with the other k-TPG techniques.

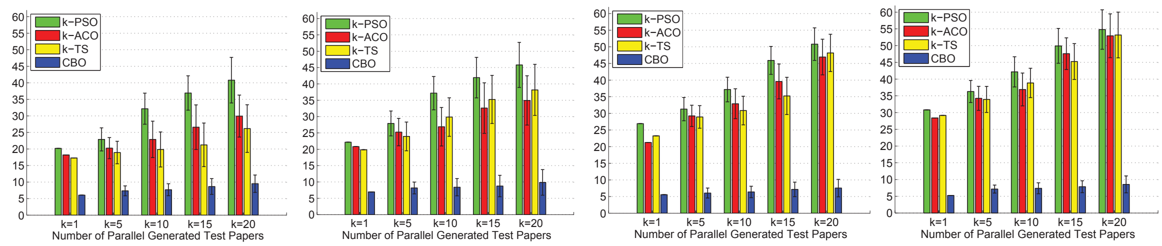
6 Conclusion

In this paper, we have proposed an effective and efficient Collective Biobjective Optimization algorithm for solving k-TPG. The key success of CBO lies in the synergy of the effective problem reformulation with the exploitation of submodular property for *collective biobjective* optimization. CBO incorporates submodular optimization mechanism and submodular



(a) Dataset D_1 (b) Dataset D_2 (c) Dataset D_3 (d) Dataset D_4

Figure 3: Results on Quality based on Mean Discrimination Degree $\mathcal{M}_d^{k,\mathcal{D}}$ and Deviate Discrimination Degree $\mathcal{V}_d^{k,\mathcal{D}}$



(a) Dataset D_1 (b) Dataset D_2 (c) Dataset D_3 (d) Dataset D_4

Figure 4: Results on Quality based on Mean Constraint Violation $\mathcal{M}_c^{k,\mathcal{D}}$ and Deviate Constraint Violation $\mathcal{V}_c^{k,\mathcal{D}}$

fairness balancing to jointly optimize the *total quality maximization* and the *fairness quality maximization* objectives. To the best of our knowledge, CBO is a pioneering greedy-based approximation algorithm, which can achieve provably near-optimal solutions in polynomial runtime for k-TPG. The performance results on various datasets have shown that the CBO approach has achieved generated parallel test papers with not only high quality, but also runtime efficiency when compared with other k-TPG techniques.

References

[Asadpour and Saberi, 2010] A. Asadpour and A. Saberi. An approximation algorithm for max-min fair allocation of indivisible goods. *SIAM Journal on Computing*, 39(7):2970–2989, 2010.

[Asadpour *et al.*, 2008] A. Asadpour, U. Feige, and A. Saberi. Santa claus meets hypergraph matchings. *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 10–20, 2008.

[Azar and Gamzu, 2012] Yossi Azar and Iftah Gamzu. Efficient submodular function maximization under linear packing constraints. In *Automata, Languages, and Programming*, pages 38–50. Springer, 2012.

[Bertsimas *et al.*, 2011] D. Bertsimas, V. F. Farias, and N. Trichakis. The price of fairness. *Operations research*, 59(1):17–31, 2011.

[Feige and Vondrak, 2006] U. Feige and J. Vondrak. Approximation algorithms for allocation problems: Improving the factor of 1-1/e. In *47th Annual IEEE Symposium on Foundations of Computer Science*, pages 667–676, 2006.

[Feige *et al.*, 2007] U. Feige, V.S. Mirrokni, and J. Vondrak. Maximizing non-monotone submodular functions. *SIAM Journal on Computing*, 40(4):1133–1153, 2007.

[Fujishige, 2005] S. Fujishige. *Submodular functions and optimization*. Elsevier Science, 2005.

[Ho *et al.*, 2008] T. F. Ho, P. Y. Yin, G. J. Hwang, S. J. Shyu, and Y. N. Yean. Multi-objective parallel test-sheet composition using enhanced particle swarm optimization. *Journal of Educational Technology and Society*, 12(4):193–206, 2008.

[Hu *et al.*, 2009] X. M. Hu, J. Zhang, H. S. H. Chung, O. Liu, and J. Xiao. An intelligent testing system embedded with an ant-colony-optimization-based test composition method. *IEEE Transactions on Systems, Man, and Cybernetics*, 39(6):659–669, 2009.

[Hwang *et al.*, 2008] G. J. Hwang, H. C. Chu, P. Y. Yin, and J. Y. Lin. An innovative parallel test sheet composition approach to meet multiple assessment criteria for national tests. *Computers and Education*, 51(3):1058–1072, 2008.

[Ishibuchi *et al.*, 2008] H. Ishibuchi, N. Tsukamoto, and Y. Nojima. Evolutionary many-objective optimization: A short review. In *IEEE World Congress on Evolutionary Computation*, pages 2419–2426, 2008.

[Lovasz, 1983] L. Lovasz. Submodular functions and convexity. *Mathematical programming: the state of the art*, pages 235–257, 1983.

[Martin, 2012] F. G. Martin. Will massive open online courses change how we teach? *Communication of the ACM*, 55(8):26–28, August 2012.

[Nedic *et al.*, 2010] A. Nedic, A. Ozdaglar, and P. A. Parrilo. Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55(4):922–938, 2010.

[Nguyen *et al.*, 2013] M. L. Nguyen, S. C. Hui, and A. C. M. Fong. Large-scale multiobjective static test generation for web-based testing with integer programming. *IEEE Transactions on Learning Technologies*, 6(1):46–59, 2013.

[Sviridenko, 2004] M. Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1):41–43, 2004.

[Vondrak, 2008] J. Vondrak. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 67–74, 2008.