Appl Intell (2012) 36:979–994
DOI 10.1007/s10489-011-0308-x

Simultaneously construct IRT-based parallel tests based on an adapted CLONALG algorithm

Ting-Yi Chang · You-Fu Shiu

Published online: 14 July 2011

Abstract The simultaneously construct IRT-based (*Item Response Theory*) parallel tests problem requires large numbers of variables and constraints, which leads to high computational complexities and now there is no polynomial time algorithm that exists for finding the optimal solution. This article proposes an adapted CLONALG algorithm to simultaneously construct IRT-based parallel tests. Based on the CLONALG features, the proposed scheme can use a single test construction model to simultaneously construct multiple parallel tests. At the same time, it avoids the inequality problem in the sequential construction and solves the drawback of larger numbers of variables and constraints in the simultaneous construction. The serial experiments show that the proposed scheme has a lower deviation in simultaneously constructing parallel tests than the *Linear Programming* (LP) and the *Genetic Algorithm* (GA). It is also able to construct parallel tests with identical test specifications from a large item bank.

Keywords Parallel tests construction · Item response theory · Test · Clonal selection principle · Heuristic algorithm

1 Introduction

Tests are an important method to appraise teaching performance. The *Item Response Theory* (IRT) [13, 20] is dominant psychometric technology in recent years. It has been used in large-scale tests in various fields such as *Test of English for International Communication* (TOFIC) and *Graduate Record Examinations* (GRE). The well-known *Computerized Adaptive Test* (CAT) is also developed based on IRT. With IRT widely used in the real world to construct a high-quality IRT-based test becomes more important. However, the test construction problem difficulty is that the processing time will increase exponentially with the number of items in the item bank [27, 28]. Numerous authors have therefore proposed various methods to address this problem [1–5, 11, 14–16, 24, 25, 27, 32].

Multiple tests often have to be constructed at the same time in actual testing situations, such as (1) A set of tests that must be administered at different times or locations, (2) A set of tests randomly selected for each person to avoid cheating, (3) A two test set used in an evaluation study or an educational program with a pretest-posttest design, (4) A set of tests constructed for use in a multistage CAT. These tests have similar test specifications and there is no item overlap, which are called "*parallel tests*" [28, 29]. Obviously, constructing parallel tests is much more difficult than constructing a single test. Although parallel tests can be constructed in a sequential fashion, the sequential construction method does not give satisfactory results [7]. Tests constructed later tend not to be as good as earlier constructed tests. To avoid this inequality, simultaneous test construction methods were proposed to balance the assignment of items to the parallel tests [2, 6, 16, 31]. However, Hwang *et al.*'s investigation showed that a near-optimal test is difficult to achieve when the number of items in the item bank is larger than five thousand [16], not to mention the simultaneous test construction from a large item bank. Therefore, these methods have the defect of a high computational complexity and a high deviation between the parallel tests.

T.-Y. Chang (✉) · Y.-F. Shiu
Graduate Institute of e-Learning, National Changhua University of Education, No. 1, Jin-De Road, Changhua City, Taiwan, ROC
e-mail: tychang@cc.ncue.edu.tw

An adapted CLONALG algorithm is proposed to improve the quality and efficiency of the simultaneous test construction method. Based on the parallel-search and independent evolution features in the CLONALG algorithm, the proposed method uses a single test construction model to simultaneously construct multiple parallel tests. Because the simultaneous parallel test construction problem is simplified into a single test construction problem, this method effectively reduces the computational complexity and the constructed parallel tests have very low deviation. Serial experiments showed that the adapted CLONALG algorithm for parallel test construction is more suitable than other algorithms such as the *Linear Programming* (LP) and the *Genetic Algorithm* (GA).

This paper is organized as follows. Section 2 describes the simultaneous test construction problem and reviews the related works. Section 3 describes the theory and features of the original CLONALG. Section 4 considers two kinds of test construction problem which are often used in actual testing situations [29]. Section 5 describes the operators in the adapted CLONALG and how to use the adapted CLONALG to solve the simultaneous test construction problem. In Sect. 6, a series of computational experiments are presented to estimate the adapted CLONALG performance in the construction. GA and LP are compared to the proposed scheme. Finally, we focus on the proposed scheme to discuss its advantages in real practice in Sect. 7.

2 Related works

The test construction problem is a combinatorial optimization problem [17, 19, 21, 22], and now there is no polynomial time algorithm that exists for finding the optimal solution. An IRT-based test construction is composed of three steps. Using the 3-parameter logistic model, each item's discrimination, difficulty, and guessing are translated into item information functions [13]. Second, the instructor determines the objective *Test Information Function* (TIF) according to the test purpose. For example, the commonly used test is the criterion-referenced test whose purpose is to distinguish mastery and non-mastery in groups of examinees. The objective TIF is single-peak shaped, with the center peak located at the designated ability level classification, as shown in Fig. 1. Third, the objective TIF is the sum of the item information in the objective test, so a test construction method selects a set of items from the item bank to compose a test and attempts to approximate the objective TIF.

From the above processes, a high-quality IRT-based test relies on a well-designed item bank and also selects a set of items to approximate the objective TIF in an acceptable time [13, 20]. The IRT-based test construction problem can be formulated as a mathematical programming model. Theunissen [26] first used LP to solve this problem. He used a sequential method to construct multiple parallel tests. In the sequential construction, the first test is constructed and the selected items in the test are removed from the item bank to avoid item-overlap between the parallel tests. The operation is repeated until all parallel tests are constructed. The advantage of sequential construction is easy implementation, but the computation time is linearly increased by the number of parallel tests. However, because these tests are constructed sequentially, the selected items have inequality conditions. That is, after constructing the first test, the good items (i.e., items with high information at a specified ability level) in the item bank are skimmed off and the optimal value of the objective function for the second test will be lower than that for the first one. This problem is called the "*inequality problem*," and the deviation caused by the inequality problem will increase with the number of parallel tests.

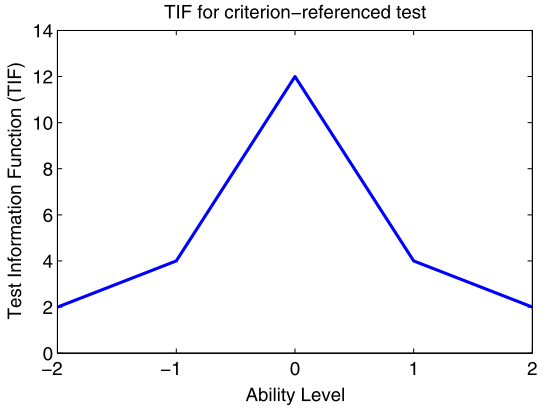


Fig. 1 Using TIF to distinguish two groups of examinees [13]

Therefore, Boekkooi-Timminga [6] added an item-overlap constraint in the mathematical programming model to propose a general model for simultaneous construction of multiple tests. The variables used in the model are defined as follows:

- x_{if} The binary decision variable. $x_{if} = 1$ if ith item is assigned to fth test; otherwise $x_{if} = 0$.
- N The number of items in the item bank.
- F The number of parallel tests.
- $I_i(\theta_k)$ The item information value of ith item at kth ability level.
- $T(\theta_k)$ The objective value of test information at kth ability level.
- n The objective test length.

$$\text{Minimize} \sum_{f=1}^{F}\sum_{k=1}^{K}\sum_{i=1}^{N} I_i(\theta_k)x_{if}, \quad \text{(objective function)} \quad (1)$$

$$\text{s.t.} \sum_{i=1}^{N} I_i(\theta_k)x_{if} \geqslant T(\theta_k),$$
$$k = 1, \ldots, K, f = 1, \ldots, F, \quad \text{(TIF)} \quad (2)$$

$$\sum_{i=1}^{N} x_{if} = n, \quad f = 1, \ldots, F, \quad \text{(test length)} \tag{3}$$

$$\sum_{f=1}^{F} x_{if} \leqslant 1, \quad i = 1, \ldots, N, \quad \text{(item-overlap)} \tag{4}$$

$$x_{if} \in \{0, 1\}, \quad i = 1, \ldots, N, \ f = 1, \ldots, F. \quad \text{(definition of } x_{if}) \tag{5}$$

Equation (1) indicates that the sum of each test's information at each ability level considered a minimum height to approximate $T(\theta_k)$. The actual test information of all tests is summed $\sum_{i=1}^{N} I_i(\theta_k)x_{if}$ in (2) and it must be greater than or equal to $T(\theta_k)$. Equation (3) restricts the test length. Equation (4) is the item-overlap constraint, which indicates that ith item can only be selected once in F parallel tests.

The Boekkooi-Timminga model achieves the simultaneous construction concept and also overcomes the inequality problem occurring in sequentially constructed multiple parallel tests. However, the number of decision variables will substantially grow along with both N and F. The number of item-overlap constraints will also grow along with N. Unless it can be solved using a specified heuristic algorithm, only the test construction problem with smaller N and F can be solved. Oppositely, larger N and F may quickly result in memory management problems and large computation times [29].

Ackerman [1] returned to the sequential construction and proposed a two-stage method to solve the inequality problem. In the first stage, multiple parallel tests are sequentially constructed. In the second stage, items are swapped between tests to minimize the differences between their information functions. This method slightly reduces the deviation caused by the inequality problem. Adema [2] followed Ackerman's two-stage sequential construction further to propose a two-stage simultaneous construction method which does not need a larger number of variables and constraints. His idea is that the system selects the total number of items for F tests directly, and the total number of items is then split into F parallel tests with similar test specifications. In the first stage one large test is constructed. The large test length is F times of the single test length. The objective TIF of the large test is set F times as large as the TIF of the single test. The other objective values for the other constraints (e.g., contents, skills and item types) are the same. The model is essentially a single test construction model as follows:

$$\text{Minimize} \sum_{k=1}^{K} \sum_{i=1}^{N} I_i(\theta_k)x_i, \quad \text{(objective function)} \tag{6}$$

$$\text{s.t.} \sum_{i=1}^{N} I_i(\theta_k)x_i \geq T(\theta_k)F, \quad k = 1, \ldots, K, \quad \text{(TIF)} \tag{7}$$

$$\sum_{i=1}^{N} x_i = nF, \quad \text{(test length)} \tag{8}$$

$$x_i \in \{0, 1\}, \quad i = 1, \ldots, N. \quad \text{(definition of } x_i) \tag{9}$$

In the second stage, the large test is split into F tests which meet the test specification and make the F tests as similar as possible. Basically, the method simplifies the simultaneous parallel tests construction problem into a single test construction problem. This concept successfully solves the inequality problem and does not need a larger number of variables and constraints. However, a drawback of this idea is that it is only applicable to the simple test specification (e.g., only considers TIF, contents and test length). The complex test specification will prevent the large test from being split into F parallel tests with similar test specifications.

van der Linden and Adema [29] applied Adema's idea and proposed the dummy test concept to overcome the inequality problem in sequential construction. The dummy test concept for sequentially constructing multiple tests is that: At each step, the operation simultaneously constructs a single test and a dummy test. The single test is constructed according to the objective test specification, while the dummy test is constructed according to the sum of the test specifications for all remaining tests. As soon as both tests are constructed, the selected items in the dummy test are returned into the item bank. The process is repeated for each sequentially constructed test. The only task for the dummy test is to balance the item selection opportunity between the current constructed test and the remaining tests. This method ensures that earlier constructed tests will not pick off all of the good items. Because the dummy test will select a certain percentage of good items, there are enough good items to return to the item bank for the next sequential construction. This concept avoids the inequality problem and the deviations are much lower than that in Adema's method. This idea sequentially carries out a series of simultaneous two-test constructions. Although it avoids the inequality problem, the calculation is substantially increased.

After reviewing the relevant past papers, only a few studies successfully avoided the inequality problem in sequential test construction. However, other problems such as a larger number of variables and constraints, large deviations between the parallel tests and large complex calculations also arise. To date, there is no method able to construct multiple parallel tests without the above problems. The purpose of this paper is to propose an efficient simultaneous parallel test construction method and successfully solves those problems.

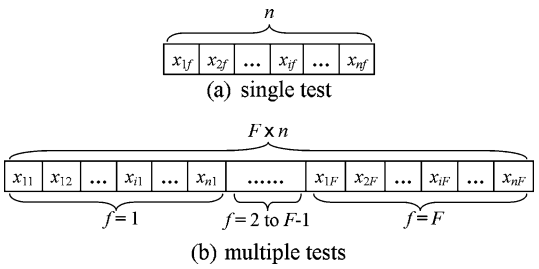


Fig. 2 Concatenate multiple strings to regarded as parallel tests [16]

Because the test construction is a combinatorial optimization problem the number of combinations grows exponentially with the item bank size. Many researchers have applied novel heuristic algorithms such as the neural network technique [24], the Monte Carlo random search [5], the tabu search algorithm [15], the particle swarm optimization algorithm [32], the immune algorithm [18], and GA [11, 14, 25] to determine a near-optimal test in recent years. These studies focused on the single test construction. When they encountered the parallel tests construction problem, they must use a sequential fashion to construct the parallel tests. Uniquely, Hwang *et al*. [16] applied a tabu algorithm with real-valued decision variable (express the item serial number) to simultaneously construct parallel tests. In the coding stage of the tabu algorithm, the test items are coded into real-value strings with one real-value string regarded as a single test, as shown in Fig. 2(a). Hwang *et al.* connected multiple strings into parallel tests, as shown in Fig. 2(b).

x_{if} denotes the real-valued decision variable. This method uses a simple way to avoid item-overlap and does not require extra constraints. This modification simplifies the simultaneous parallel tests construction problem. However, this method causes the search space grows exponentially with F and reduce the quality of constructed tests. For example, let 3 parallel tests that contain 40 items be constructed from 4000 items, that is $N = 4000$, $n = 40$, $F = 3$. Under the same parameters setting, the search space sizes for the sequential construction method and Hwang *et al*.'s method are $F \times C_N^n = 3 \times C_{4000}^{40} = 5.16 \times 10^{36}$ and $C_N^{F \times n} = C_{4000}^{3 \times 40} = 3.56 \times 10^{244}$, respectively. It can be seen that the search space size of Hwang *et al*.'s method is 6.89×10^{207} times of sequential construction method. Therefore, Hwang *et al*.'s parallel tests construction method still cannot solve the computational complexity in a large item bank size.

Moreover, according to the definition by van der Linden [28], the tests are parallel in that each of them meets the identical test specification (include TIF, contents, skills, item type constraints, and so on). Since the past studies were limited to the search capability of the test construction method, the constraints for the test specification are usually set into a set of acceptable and relaxed objective ranges. However, the parallel tests that use the objective range as constraints cannot have identical test specification. Because the parallel tests construction with the identical test specification is more difficult, only a few studies used the objective value as constraints to construct the parallel tests [9, 28, 31].

To improve the quality and efficiency of the simultaneous parallel tests construction method and ensures that the parallel tests have identical test specifications, this paper proposes an adapted CLONALG algorithm. The proposed scheme can use a single test construction model to construct multiple parallel tests simultaneously. That is, the simultaneous parallel tests construction problem into the single test construction problem based on the parallel-search and independent evolution features in CLONALG. The proposed method does not need a larger number of variables and constraints.

3 The original CLONALG algorithm

CLONALG is a kind of heuristic algorithm proposed by de Castro and Von Zuben [10]. The algorithm is based on the clonal selection principle in a biological immune system and designed to find multiple solutions at the same time. We describe the parallel-search and independent evolution features in CLONALG first to explain why CLONALG is well suited to solve the simultaneous parallel tests construction problem.

3.1 The difference between CLONALG and other methods

This paper uses a search problem as an example to explain the differences between CLONALG and other methods.

A. Traditional method A typical method is the gradient method [23], which uses a single-point search. It can find a solution within a very short time. However, the search point has difficulty escaping from the local optimal when it is located near the local optimal. This results in the loss of opportunity to find a global optimal. In Fig. 3, when the initial search point is not located near the global optimal, the final solution is easily trapped in a relatively close local optimal.

B. GA and PSO These methods use the parallel-search strategy which has multiple search points. The parallel-search strategy avoids the search point falling into a local optimal and increases the possibility of finding the global optimal. However, all search points in these methods move to the global optimal, so the multiple solutions found by these methods are similar. In Fig. 4, the five initial solutions spread in a search space, but the five final solutions will be concentrated in near the global optimal. Therefore, these methods cannot find many different solutions at the same time and they should be carried out many times for the parallel tests.

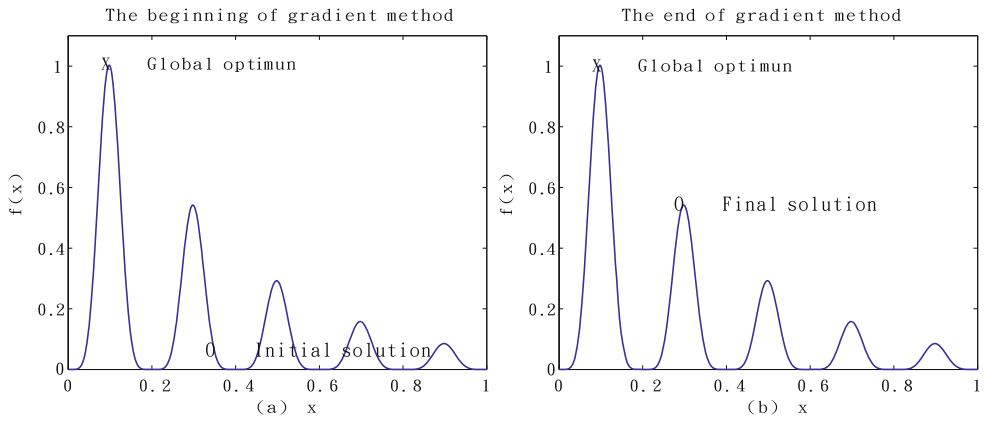


Fig. 3 Search strategy of gradient method [23]

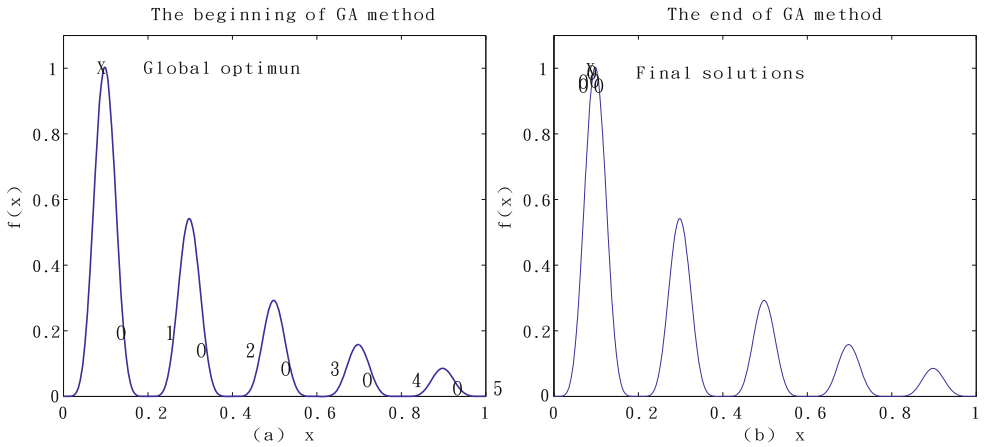


Fig. 4 Search strategy of GA [12]

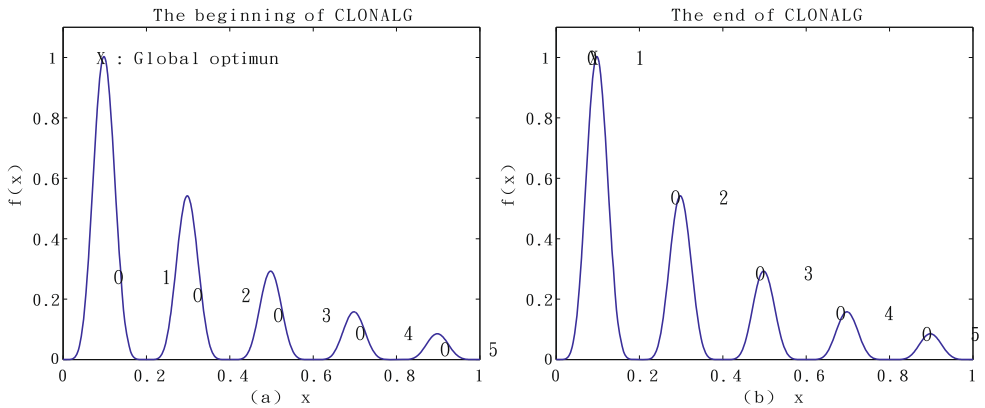


Fig. 5 Search strategy of CLONALG [10]

C. CLONALG CLONALG also uses the parallel-search strategy. The difference is that the search points evolve independently in CLONALG. The independent evolution means that the search point is only compared with itself. All search points are therefore not concentrated near the unique optimal solution. For example, in Fig. 5, the five initial solutions for CLONALG are spread in the search space and the five final solutions are still spread in the search space to find the local optimal solutions in their respective areas. Because CLONALG uses the independent evolution strategy to maintain solution diversity, it is suitable for finding multiple solutions in the search space at the same time. This feature is excellent for simultaneous parallel tests construction. The CLONALG theory and concept are described in detail as follows. The CLONALG core is the clonal selection principle. The clonal selection principle is the antibody evolution procedure for binding a specific antigen in the immune system. The binding strength of an antibody to a specific antigen is called its' "*affinity*". For the test construction problem, an antibody is regarded as a test. A specific antigen is regarded

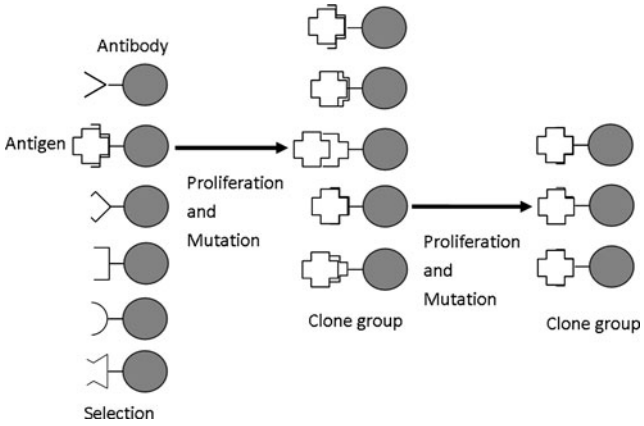


Fig. 6 Procedure of the clonal selection principle [10]

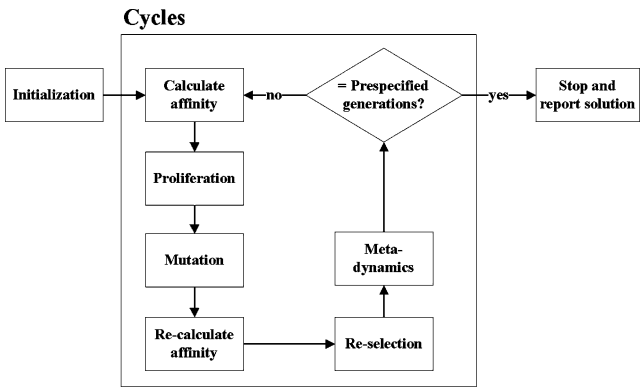


Fig. 7 Flowchart of CLONALG [10]

as an objective test specification. The affinity is the degree of the approximation of a constructed test to the objective test specification.

The clonal selection procedure is described in Fig. 6. Each antibody (test) is able to recognize and bind to a specific antigen (test specification). This binding will stimulate the antibody to reproduce more similar antibodies. These similar antibodies are called "*clones*." These antibodies will be little mutated during the proliferation process. A mutation may enhance the affinity of some antibodies. The higher affinity antibodies will have more chances to perform proliferation and more chances to further enhance the affinity. The main CLONALG based on the clonal selection principle are described in the following.

3.2 CLONALG algorithm

Figure 7 illustrates the flowchart of CLONALG and the detail steps are as follows:

1. *Initialization*: create an initial random population of antibodies. P denotes the number of antibodies in the population.
2. *Calculate affinity*: compute the affinity value for each antibody. In the test construction problem, the affinity value is inversely proportional to the deviation between the constructed test and the test specification. The test specification is formulated as model constraints in the next section.
3. *Clonal selection*: P_c denotes the number of antibodies for clonal selection. The P_c higher affinity antibodies in population are elected to perform below sub steps separately; for each selected antibody:
   - 3.1 *Proliferation*: the antibody will be cloned to compose a clone group G, and the number of clones is proportional to the affinity value. In other words, the antibody with higher affinity value has the larger number of clones.
   - 3.2 *Mutation*: for all antibodies in G, an adaptive mutation probability p_m is used to select some antibodies that will change one or more items randomly. The mutation rate p_m adapts to the affinity for each antibody and it is inversely proportional to the affinity of the antibody. That is, the antibody with higher affinity needs less adjustment to the items.
   - 3.3 *Re-select*: recalculate the affinity of these antibodies in G, and re-select higher affinity antibodies to the population.
4. *Meta-dynamics*: the antibodies that do not perform clonal selection in the population are replaced by new ones.
5. *Cycle*: Repeat from Step 2 to Step 4 until complete the pre-specified generation number.

The CLONALG key for the test construction is the test representation and the calculation of affinity. The test representation was introduced in the previous section (see Fig. 2a). The affinity function is designed for the objective function in the test construction model. The test construction models considered in this paper will be given in the next section.

4 Test construction models

Before introducing the test construction models, the simultaneous parallel tests construction model ((1) to (5)) is greatly simplified due to the following two factors.

1. The decision variable is a real-valued type.

   In general, the item bank size is much larger than the test length. Since the number of binary decision variables is equal to the item bank size, it leads to a very larger number of variables in parallel-search. Therefore, this paper uses the real-valued decision variable (the item serial number) to make the number of decision variables equal to the test length. This greatly reduces the number of variables and also does not need the test length constraint.

2. The variable f and the item-overlap constraints are removed.

   Because this paper modifies CLONALG to avoid item-overlap between tests in later sections and each test (antibody) is constructed independently, the model can only consider a single test construction which does not require the number of parallel tests and the item-overlap constraints. Therefore, the simultaneous parallel tests construction model is greatly simplified to the single test construction model. In other words, the variable f is removed in our model.

The test specification (test construction constraints) considered in this study includes TIF, contents, skills, item types, and word count constraint. To ensure the constructed parallel tests have nearly identical test specifications, this paper uses a set of objective values as constraints to construct parallel tests. It is more accurate than using a set of ranges as constraints.

Two kinds of test construction problems which are applied frequently in real practice are considered in this paper. One is the absolute TIF problem whose object is to minimize the sum of the deviations between the TIF and objective values, and the other is the relative TIF problem whose object is to maintain the relative heights of the TIF and maximize the sum of TIF [2, 29].

4.1 The absolute TIF problem

In this problem, the instructor determines the objective TIF at a series of values θ_k according to the test purpose, and then the problem is formulated as a mathematical programming model. The objective of this model (10) is to minimize the sum of the deviations between the objective TIF and the constructed TIF. The variables used in this model are defined as follows:

- x_i The real-valued decision variable is regarded as an item serial number in the item bank.
- $c_{x_i h}$ The binary decision variable indicating item x_i belongs to hth content attribute ($c_{x_i h} = 1$) or not belong to hth content attribute ($c_{x_i h} = 0$).
- C_h The objective number of items for hth content attribute.
- $s_{x_i v}$ The binary decision variable indicating item x_i belongs to vth skill attribute ($s_{x_i v} = 1$) or not belong to vth skill attribute ($s_{x_i v} = 0$).
- S_v The objective number of items for vth skill attribute.
- $t_{x_i m}$ The binary decision variable indicating item x_i belong to mth type attribute ($t_{x_i m} = 1$) or not belong to mth type attribute ($t_{x_i m} = 0$).
- T_m The objective number of items for mth type attribute.
- r_{x_i} The word count of item x_i.
- R^{u} The upper bound of total words in a test.

$$\text{Minimize} \sum_{k=1}^{K} \sum_{i=1}^{n} I_{x_i}(\theta_k), \quad \text{(objective function)} \tag{10}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} I_{x_i}(\theta_k) - T(\theta_k) \geqslant 0, \quad k = 1, \ldots, K, \quad \text{(TIF)} \tag{11}$$

$$\sum_{i=1}^{n} c_{x_i h} = C_h, \quad h = 1, \ldots, H, \quad \text{(content)} \tag{12}$$

$$\sum_{i=1}^{n} s_{x_i v} = S_v, \quad v = 1, \ldots, V, \quad \text{(skill)} \tag{13}$$

$$\sum_{i=1}^{n} t_{x_i m} = T_m, \quad m = 1, \ldots, M, \quad \text{(type)} \tag{14}$$

$$\sum_{i=1}^{n} r_{x_i} \leqslant R^{\mathrm{u}}, \quad \text{(word count)} \tag{15}$$

$$x_i \in [1, N], \quad i = 1, \ldots, n. \quad \text{(definition of } x_i) \tag{16}$$

Since the variable f is removed in our model, the above equations are simplified and this model becomes a single test construction model. Equation (10) indicates that the sum of each test's information at each ability level considered a minimum height to approximate $T(\theta_k)$. The actual test information of all tests is summed $\sum_{i=1}^{n} I_{x_i}(\theta_k)$ in (11) and it must be greater than or equal to $T(\theta_k)$. Equations (12), (13) and (14) set the objective number of items for contents, skills and types respectively. Equation (15) set that the upper bound for the total word count of constructing a test. This model is very straightforward. However, the instructor may not be able to set a reasonable objective TIF. The model presented in the next subsection is helpful for instructors to focus on the shape of the TIF according to the test purpose.

4.2 The relative TIF problem

The model for absolute TIF has an implicit assumption that the instructor is familiar with the item information distribution in the item bank, so the instructor may not set up an unreasonable objective TIF. However, the instructor may only know the test purpose that s/he wants. In this condition, the instructor can only set the relative heights to describe the shape of the TIF. The model for relative TIF maintains the relative heights as a prerequisite to maximize the sum of TIF. And the relative heights of TIF are called relative efficiencies $r(\theta_k)$ [2, 29]. For instance, if the test purpose is a single-peak TIF in Fig. 1, the relative efficiencies can be set as $r(\theta_k) = \{1, 2, 5, 2, 1\}$ at kth ability level to describe the single-peak shape. After setting $r(\theta_k)$, the series of lower bounds $r(\theta_k)y$ at kth ability level in (18) are imposed on the TIF of the constructed test to maintain the relative height,

and $r(\theta_k)y$ are forced to be as high as possible by maximizing y in the objective function.

$$\text{Minimize } y, \quad \text{(objective function)} \tag{17}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} I_{x_i}(\theta_k) - r(\theta_k)y \geqslant 0, \quad k = 1, \ldots, K. \quad \text{(TIF)} \tag{18}$$

This model also considers the constraints for content, skill, type and word counts in (12) to (16). The difference is that the objective TIF $T(\theta_k)$ in (11) is replaced by a product of $r(\theta_k)$ and y in (18) to form a series of lower bounds for the TIF of the constructed test. And the objective function (17) is to maximize y. The following section describes how to use the adapted CLONALG to solve two models in detail.

5 The adapted CLONALG for simultaneous parallel tests construction

As outlined, the basic problem of the simultaneous construction is that it needs a larger number of constraints to avoid item-overlap between parallel tests. In the original CLONALG, although the antibodies are independent in the construction procedure, three operations: *Initialization*, *Mutation*, and *Meta-dynamics* will import some new items that will probably lead to item-overlap. Therefore, this study modifies these operations to avoid item-overlap between antibodies in the construction procedure so that the adapted CLONALG is suitable for constructing parallel tests at the same time.

To ensure the new items, which are imported by the above three operations, are not selected by other tests, this study proposes the "adaptation of the standard CLONALG" concept. The adapted CLONALG maintains a item list *LIST*, once an item is added into *LIST*, it would be temporarily excluded from the item bank to avoid that item being selected by other tests. For instance, in the initialization operation, the initial antibodies are randomly and sequentially generated. Once an antibody is generated, the selected items in the antibody are added into *LIST* immediately to avoid the item-overlap occurring in the next initial antibody. For the same reason, once an item is selected by an antibody (test) in the mutation or meta-dynamics operations, it would be added into *LIST* immediately until it is released by this antibody. The item will be released when this item is changed by another item in the mutate operation, or the antibody which contains this item is replaced by a new antibody in the meta-dynamic operation.

Note that the adapted CLONALG concept is similar to the sequential construction, but it has no inequality problem. Because the tests select the items simultaneously in CLONALG, the tests have an equal chance to select good items and no test is able to select all of the good items before other tests. Therefore, the adapted CLONALG can construct parallel tests simultaneously without considering the item-overlap constraint. The following section describes the adapted CLONALG for the absolute TIF and relative TIF, respectively.

5.1 The adapted CLONALG for absolute TIF

1. *Initialization*: The operation randomly and sequentially generates P antibodies to compose a population. Each antibody x is composed by n real values $x = [x_1, \ldots, x_n]$ where $x_i \in [1, N]$. Once an antibody (test) is generated, the selected items in the test are added into *LIST* immediately.
2. *Calculate affinity*: The affinity value for each antibody in the population is calculated. The affinity value is a degree of the approximation of an antibody to the test specifications, so it should be designed based on the objective function and all constraints. Since the constraints in the model for absolute TIF are to define the objective values for various item attributes (include contents, skills, item types, word count and TIF in this paper), the affinity function for each constraint is designed as a penalty function for the deviation between the objective test specification and the constructed test.

   – The penalty function for (11), D^{I} denotes the deviation of TIF in all abilities.

   $$\sum_{k=1}^{K} \left| \left[\sum_{i=1}^{n} I_{x_i}(\theta_k) \right] - T(\theta_k) \right| = D^{\mathrm{I}}. \quad \text{(TIF)} \tag{19}$$

   – The penalty function for (12), $\sum_{i=1}^{n} c_{x_i h}$ denotes the number of items for the hth content attribute in the constructed test, D^{C} denotes the deviation of number of items in all content attributes.

   $$\sum_{h=1}^{H} \left| \left(\sum_{i=1}^{n} c_{x_i h} \right) - C_h \right| = D^{\mathrm{C}}. \quad \text{(content)} \tag{20}$$

   – The penalty function for (13), $\sum_{i=1}^{n} s_{x_i v}$ denotes the number of items for the vth skill attribute, D^{S} denotes the deviation of number of items in all skill attributes.

   $$\sum_{v=1}^{V} \left| \left(\sum_{i=1}^{n} s_{x_i v} \right) - S_v \right| = D^{\mathrm{S}}. \quad \text{(skill)} \tag{21}$$

   – The penalty function for (14), D^{T} denotes the deviation of number of items in all type attributes.

   $$\sum_{m=1}^{M} \left| \left(\sum_{i=1}^{n} t_{x_i m} \right) - T_m \right| = D^{\mathrm{T}}. \quad \text{(type)} \tag{22}$$

– The penalty function for (15), D^{R} denotes the deviation between the total word count and the upper bound R^{u}. If the total word count is larger than R^{u}, the deviation D^{R} is greater than zero; otherwise, it is equal to zero.

$$\max\left\{0, \left(\sum_{i=1}^{n} r_{x_i}\right) - R^{\mathrm{u}}\right\} = D^{\mathrm{R}}. \quad \text{(word counts)} \tag{23}$$

– The affinity function.

$$\begin{aligned}\text{affinity}(x) = -\big(&W^{\mathrm{I}}D^{\mathrm{I}} + W^{\mathrm{C}}D^{\mathrm{C}} + W^{\mathrm{S}}D^{\mathrm{S}} \\ &+ W^{\mathrm{T}}D^{\mathrm{T}} + W^{\mathrm{R}}D^{\mathrm{R}}\big)\end{aligned} \tag{24}$$

The constructed test will fully meet the objective test specification when all deviations are equal to zero. The affinity function is the negative sum of weighted deviations for all constraints as shown in (24). The higher the affinity value has the better result. Various W^{I}, W^{C}, W^{S}, W^{T}, and W^{R} are user-defined weights to normalize the scale of different deviations and to enlarge the range of affinity values suitable for the selection operation. Therefore, the weight setting is so flexible that it can be set as a large value or s-shaped function.

3. *Clonal selection*: The P_c antibodies with the higher affinity are elected to perform the clonal selection separately. To enhance the local search ability, modified clonal selection operator based on the fixed test length is proposed to find the local optimums for all antibodies in their areas. For each selected antibody x:
   3.1 *Proliferation*: the number of clones is proportional to the affinity in the original CLONALG. In this study, the number of clones N_c is directly set to the test length n for the follow-up single-point mutation.
   3.2 *Single-point mutation*: in the original CLONALG, the mutate point (item) is decided by an adaptive mutation probability p_m. To enhance the local search ability and simplify the parameter setting in CLONALG, each antibody in the group G is performed the single-point mutation successively in this study, and the mutation point (item) is directly set to the order of antibodies in G. An example of the operations in Steps 3.1 and 3.2 is shown in Fig. 8. Assume that the test length is 4, the selected antibody is cloned n times to compose the clone group G which has 4 antibodies. Suppose the selected items in x are items 2, 4, 7, and 8, these items are added into *LIST* to avoid repeat selection. First, this procedure chooses item 2 in the first antibody directly as the mutation point. An item 3 is then randomly selected to replace item 2 in the first antibody. The follow-up antibody can be explained along similar lines. (In actual condition, *LIST* not only contains the selected items in x, but also contains the items selected by all tests.)
   3.3 *Re-select*: the operation re-calculate the affinity of antibody in G, and it elects the highest affinity antibody x^*. If the affinity value of x^* is larger than its original x, then x^* will replace x. Otherwise, x is maintained for the next cycle.
4. *Meta-dynamics*: This operation is same as that described but easier. It also temporarily excludes all items selected by the population so that item-overlap does not occur.
5. *Cycle*: the operation repeats from Step 2 to Step 4 until complete N_{gen} generations. The F higher-affinity antibodies are elected to regard as final solutions when the algorithm terminates, these antibodies are F parallel tests.

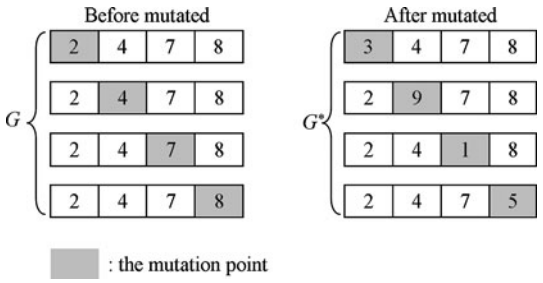


Fig. 8 Example of single-point mutation in the adapted CLONALG

5.2 The adapted CLONALG for relative TIF

In this model, the objective function maximize TIF, and also maintain the shape of TIF. Note that the antibodies evolve independently in CLONALG, so each antibody has its own y. This will lead to multiple tests that have different TIFs. This paper proposes using a two-stage CLONALG to cope with this phenomenon. The first stage finds a feasible target common factor y_{T}, and the second stage uses y_{T} to define the objective TIF and construct a near-optimal test that meets the objective TIF (the model for absolute TIF).

First stage: find the target common factor y_{T} Since the only difference between this stage and the previous model is the affinity function, we introduce this part only. The affinity function in this stage uses the sum of TIF as a reward value to maximize the TIF. The model for relative TIF forces the information in each ability level to be larger than the lower bounds $r(\theta_k)y$ to maintain the relative heights. For this purpose, the affinity function calculates a multiple factor y_k between and $r(\theta_k)$ for $k = 1, \ldots, K$, and then uses the standard deviation of y_k as a penalty value to balance each y_k. The standard deviation, denoted as σ, is proportional to the variance of the common factor y, so the lower σ the more approximate the sharp of $r(\theta_k)$.

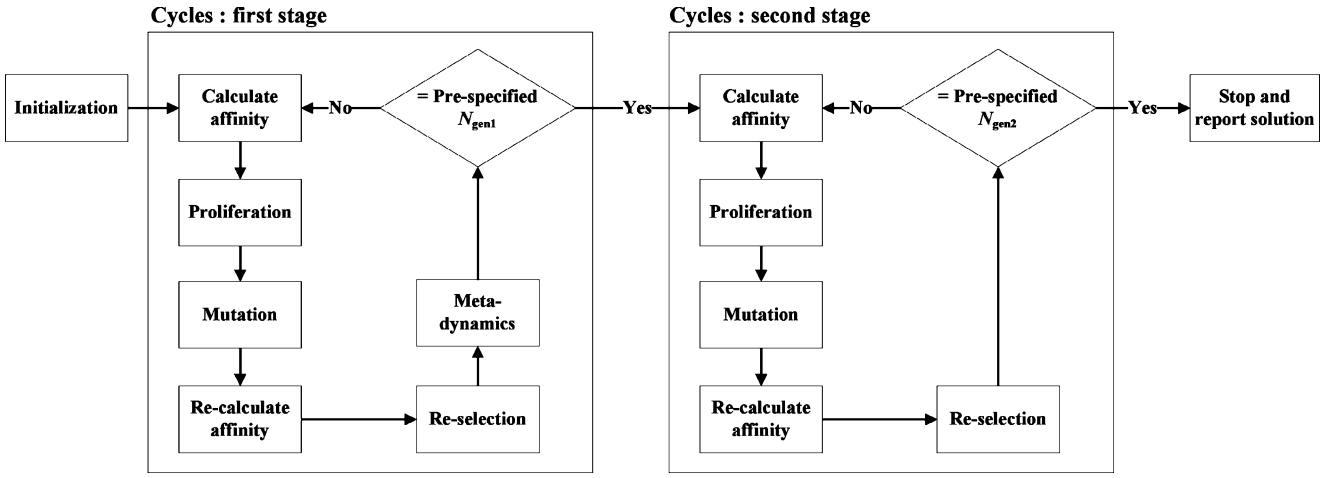


Fig. 9 Flowchart of two-stage CLONALG

– The penalty function for the variance of common factor y.

$$y_k = \frac{1}{r(\theta_k)} \sum_{i=1}^{n} I_{x_i}(\theta_k), \quad k = 1, \ldots, K, \tag{25}$$

$$\sigma = \sqrt{\frac{1}{K} \sum_{k=1}^{K} \left(\sum_{i=1}^{n} I_{x_i}(\theta_k) - \overline{y_k} \right)^2}. \tag{26}$$

Equation (25) calculates all y_k for each ability. Equation (26) calculates the standard deviation of y_k as the penalty value.

– The affinity function.

$$\begin{aligned} \text{affinity}(x) = & \left(W^{\mathrm{I}} \sum_{k=1}^{K} \sum_{i=1}^{n} I_{x_i}(\theta_k) \right) \\ & - (K W^{\mathrm{I}} \sigma + W^{\mathrm{C}} D^{\mathrm{C}} + W^{\mathrm{S}} D^{\mathrm{S}} \\ & + W^{\mathrm{T}} D^{\mathrm{T}} + W^{\mathrm{R}} D^{\mathrm{R}}). \end{aligned} \tag{27}$$

The difference between (24) is that (27) uses the sum of weighted TIF as a reward value, and uses K times weighted σ as a patently value.

The first stage ends when the number of generations N_{gen1} is completed. The target common factor y_{T} must then be calculated. The F high-affinity antibodies are elected from the final population and then the average common factor of these antibodies is calculated at the peak ability level as the target common factor y_{T}.

Second stage: construct the parallel tests for the absolute TIF This stage minimizes the deviations between the TIF and the $T(\theta_k)$, which is the product of y_{T} and $r(\theta_k)$. This stage is similar to the previous model and the affinity function in this stage is (24). But there are four different points:

1. The items that are not selected by the population in first stage are added into *LIST* at the end of the algorithm. Because the objective of the model for relative TIF is maximizing TIF, the items with high information will be collected by the population in the first stage. On the other hand, the items that are not selected by the population are relatively lower. These items are therefore not considered in the second stage, which greatly reduces the search space size.
2. P is set equal to F. That is, the population in this stage only contains F antibodies, and the selected items in other antibodies are returned to the item bank from *LIST*. The reason for putting these items back into the item bank is that these items are likely to have higher information values and can therefore be expected to improve the quality of F antibodies, and the computational complexity of the algorithm be reduced substantially.
3. P_c is also set equal to F. Since the population only contains F antibodies (parallel tests) as final solutions, each antibody must be performed the clonal selection.
4. Since all antibodies perform clonal selection, the second stage does not require the meta-dynamic operator.

The second stage ends when the number of generations N_{gen2} is completed. The F antibodies are as the F parallel tests. Figure 9 shows the flowchart of the two-stage CLONALG procedure.

6 Experiments and evaluation

The experimental results for the adapted CLONALG are presented in this section. According to Sun's experiment design [25], a virtual item bank with 4000 items is generated for constructing various tests. The parameter and attribute values are randomly generated according to a reasonable

Table 1 Attribute of simulate 4000-item bank

	Item parameters*			Item attributes			
	a	*b*	*c*	Content	Skill	Type	Length
Range	0.8∼3.0	−3.0∼3.0	0.1∼0.3	1∼10	1∼6	1∼5	20∼50
Type	real	real	real	integer	integer	integer	integer
Mean	1.9	−0.05	0.2	5.444	3.494	2.967	34.854
Std	0.638	1.734	0.058	2.8917	1.715	1.429	10.149

*The parameters a, b and c are discrimination, difficulty and guessing respectively

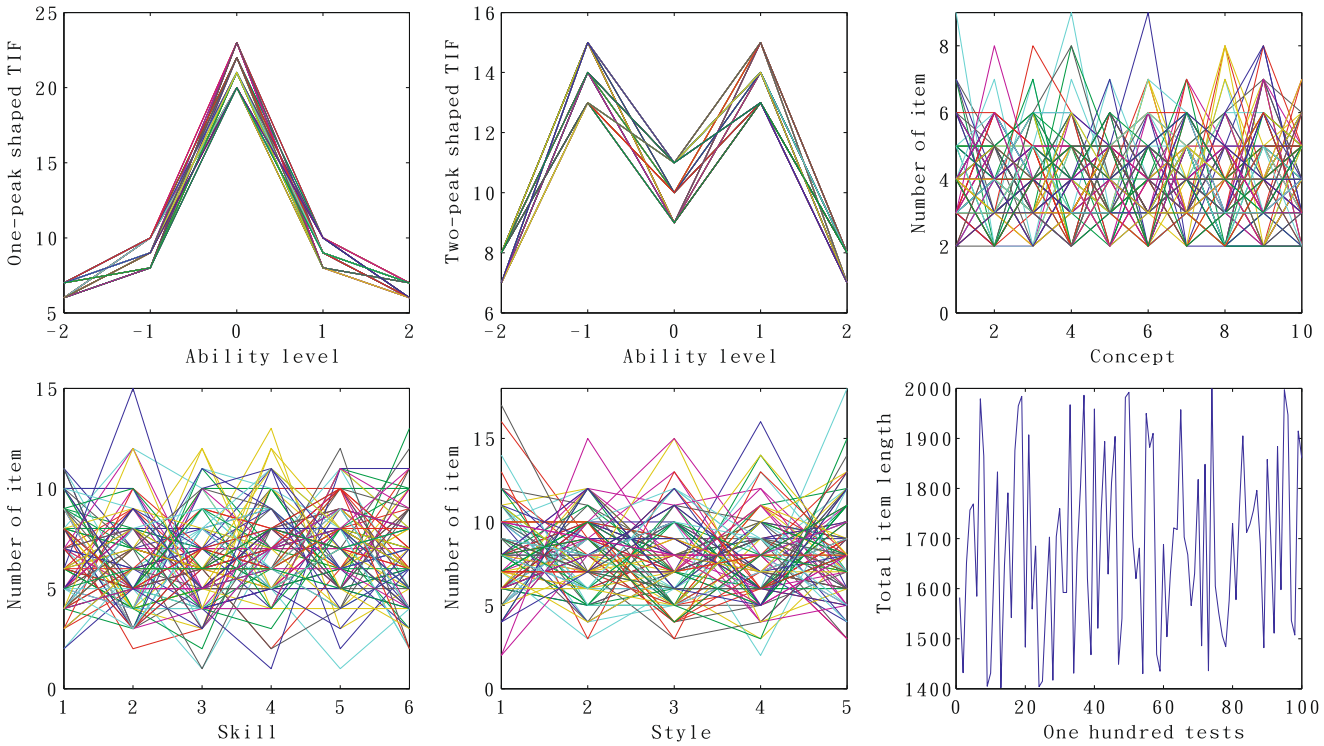


Fig. 10 Distributions of objective value for one hundred test specifications

range defined in Table 1. The performances of models for absolute TIF and relative TIF are evaluated respectively as follows.

6.1 The random test specification for the absolute TIF problem

This experiment considers two types of test purpose. The first one is the single-peak shaped TIF in cases where an instructor wants to separate students into two groups. The second one is the double-peak shaped TIF in cases where an instructor wants to filter two groups of students with abilities at the mid-high and mid-low levels. These two types of TIF were utilized to compose one hundred test specifications. $T(\theta_k)$ were randomly generated in both instances.

For a single-peak shaped TIF, $T(\theta_k)$ varied within the $6 \sim 7$, $8 \sim 10$, $20 \sim 23$, $8 \sim 10$, $6 \sim 7$ for $\theta - 2$, -1, 0, 1, 2 ranges, respectively. For a double-peak shaped TIF, the ranges are $7 \sim 8$, $13 \sim 15$, $9 \sim 11$, $13 \sim 15$, $7 \sim 8$ for $\theta - 2$, -1, 0, 1, 2, respectively. Following the information quantity limitations defined in these two distributions, one hundred $T(\theta_k)$ were randomly generated for each one. The objective number of items for content, skill, and type were also randomly generated. The objective value's distributions (test specifications) are presented in Fig. 10. Each line represents a set of objective values in a test specification.

The adapted CLONALG is evaluated by comparing it with LP and GA. LP is widely used in the test construction problem and GA has been reported an efficient method for solving a large and complex test construction problem in recent years [11, 14, 25, 30]. The Premium Solver Platform with the Large-Scale LP Solver Engine is used as the LP solver tool [8]. However, since the test specification for the parallel tests with the identical test specification is more difficult and the item bank size is up to 4000, the experimental results show that LP takes a week to construct a single

Table 2 GA and CLONALG parameters

GA			CLONALG		
N	4000	The item bank size	N	4000	The item bank size
n	40	Test length	n	40	Test length
P	100	The size of population	P	30	The size of population
p_c	1	The crossover rate	P_c	20	The number of antibodies for the clonal selection
p_m	$1/n$	The mutation rate	N_c	40	The size of clone group
N_{gen}	1000	Generation number	N_{gen}	1500	Generation number

Table 3 GA sequential construction vs. CLONALG simultaneous construction in absolute TIF

	Absolute information			
	Single-peak shaped TIF		Double-peak shaped TIF	
The sum of MSE	GA (Sequential)	CLONALG (Simultaneous)	GA (Sequential)	CLONALG (Simultaneous)
1st	0.68	0.04	0.51	0.03
2st	0.64	0.05	0.73	0.05
3st	0.62	0.07	0.64	0.07
4st	0.51	0.09	0.53	0.08
5st	0.60	0.11	0.52	0.09
Average	0.61	0.07	0.58	0.07
Ratio*	8.71	1	8.28	1

*Ratio = the average MSE of GA divided by the average MSE of CLONALG

test in our experiment. It cannot construct 5 parallel tests simultaneously within an acceptable time. Therefore, LP cannot be used for constructing parallel tests. Although GA and CLONALG are heuristic algorithms, GA is not suitable for simultaneous construction (see Sect. 3.1). Therefore, GA constructs 5 parallel tests sequentially while CLONALG constructs 5 parallel tests simultaneously. To observe the influence of the parameters on the affinity value from a series of preliminary experiments, the GA and CLONALG parameters that can find the best result within an acceptable time are shown in Table 2. One hundred test constructions were executed for the single-peak and double-peak, respectively. For each construction, the sum of TIF deviations in all θ (19) was recorded and used to calculate the mean squared deviation (MSE). The MSE is to quantify the difference between the constructed parallel tests and the objective test. The average MSE of one hundred test constructions is shown in Table 3. The experiment result shows that GA and CLONALG effectively construct parallel tests with identical test specifications. Furthermore, CLONALG constructs parallel tests simultaneously and also greatly reduces the deviation between the TIF and $T(\theta_k)$. The average GA deviation was $0.61/0.07 = 8.71$ times and $0.58/0.07 = 8.28$ times for CLONALG for the single-peak and double-peak, respectively.

6.2 The fixed test specification for the relative TIF problem

This model has a two-stage CLONALG in which the number of generations is divided into N_{gen1} and N_{gen2}. Since the purpose of the first stage is to find y_{T}, $N_{\text{gen1}} = 250$ is enough to find the optimal y_{T}. N_{gen2} must be set to a higher value ($N_{\text{gen2}} = 2000$) to find multiple near-optimal tests that meet $r(\theta_k)y_{\text{T}}$ at kth ability level. P is equal to F leads to a lower computational complexity in the second stage even if N_{gen2} is up to 2000. GA also uses the two-stage strategy. Since GA has a larger P, N_{gen1} and N_{gen2} are set as 50 and 950, respectively, is enough to find the best result. The experiment uses a single-peak $r(\theta_k) = \{1, 2, 5, 2, 1\}$ and a balancing distribution of objective value for other constraints to compare two algorithms. The objective values for other constraints is set as $C_h = \{4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4\}$, $S_v = \{6, 7, 7, 7, 7, 6\}$, $T_m = \{8, 8, 8, 8, 8\}$ and $R^u = 2000$.

Because CLONALG and GA involve some random operations, fifty test constructions were executed and averaged. The average TIF of the constructed tests is shown in Table 4. The deviations in constructed tests at all θ in CLONALG were not more than 0.1 in various F. The result indicates that CLONALG maintains a much more stable TIF shape and a lower deviation than GA. Although the parallel tests constructed using GA have a higher TIF, these tests have an obvious deviation in TIF shape. Conversely, the deviations for all θ in GA were larger than that in CLONALG

Table 4 Average TIF of constructed tests using GA and CLONALG

	Relative shape									
	GA (Sequential)					CLONALG (Simultaneous)				
θ	θ_{-2}	θ_{-1}	θ_0	θ_1	θ_2	θ_{-2}	θ_{-1}	θ_0	θ_1	θ_2
$F=2$	8.6	17.2	43.4	17.3	8.7	8.6	17.2	42.9	17.2	8.5
	8.6	17.2	43.2	17.2	8.7	8.5	17.1	42.9	17.2	8.6
$F=3$	8.5	17.1	43.1	17.2	8.6	8.5	16.9	42.4	16.9	8.5
	8.7	17.3	43.5	17.4	8.6	8.5	16.9	42.4	17.0	8.5
	8.6	17.0	42.9	17.1	8.5	8.5	17.0	42.4	16.9	8.4
$F=4$	8.6	17.2	43.1	17.2	8.6	8.5	17.0	42.4	17.0	8.5
	8.6	17.2	43.0	17.2	8.5	8.5	16.9	42.4	16.9	8.5
	8.6	17.1	43.0	17.2	8.6	8.5	16.9	42.4	17.0	8.4
	8.3	16.6	41.9	16.8	8.3	8.4	17.0	42.4	16.9	8.4
$F=5$	8.7	17.6	44.0	17.5	8.7	8.4	16.7	41.9	16.7	8.4
	8.6	17.0	42.5	16.9	8.4	8.4	16.7	41.8	16.7	8.4
	8.5	17.0	42.5	17.0	8.5	8.3	16.8	41.8	16.7	8.3
	8.2	16.3	40.7	16.2	8.2	8.4	16.7	41.8	16.8	8.3
	8.1	16.1	40.4	16.2	8.1	8.4	16.7	41.8	16.7	8.3

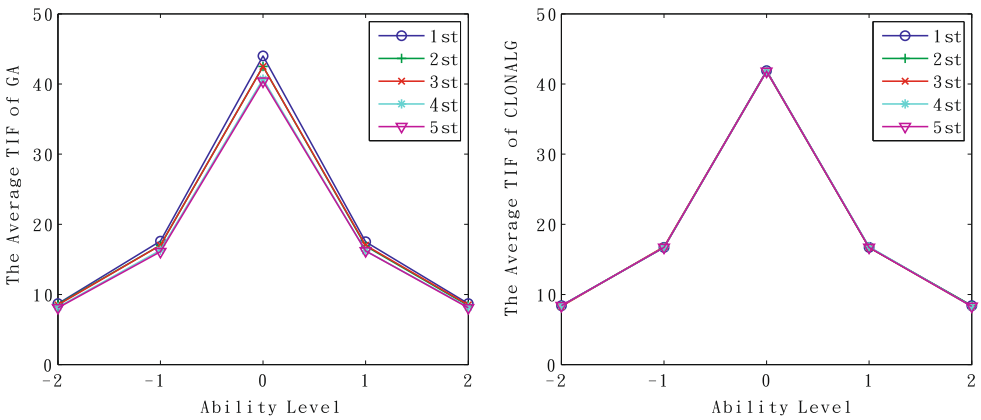


Fig. 11 TIF of constructed tests by GA and CLONALG

and increased with F. Figure 11 shows the TIF of 5 constructed parallel tests using CLONALG and GA. It indicates that the parallel tests using CLONALG have almost identical TIF, and little difference in those produced using GA. More important, the tests constructed using GA had little deviation in the content constraints, which leads these tests to not have identical test specifications. The tests constructed using CLONALG had identical test specifications.

6.3 The random test specification for the relative TIF problem

This experiment follows the same patterns as the previous one. It executed one hundred test constructions for two types of TIF, respectively. For a single-peak, the $r(\theta_k)$ varied within the ranges $0.5 \sim 1$, $1 \sim 1.5$, $2.5 \sim 3$, $1 \sim 1.5$, $0.5 \sim 1$ for $\theta - 2$, -1, 0, 1, 2, respectively; and for a double-peak, $1 \sim 1.5$, $2 \sim 3$, $1.25 \sim 1.75$, $2 \sim 3$, $1 \sim 1.5$ for $\theta - 2$, -1, 0, 1, 2, respectively. The objective values distributions are presented in Fig. 12. One hundred test constructions were executed for these test specifications, the average deviation are shown in Table 5. The average deviation of GA is 7.1 times and 7.89 times larger than CLONALG for the single-peak and double-peak respectively.

This experiment compares the execution times for CLONALG, GA and LP. These methods were executed on a personal computer with Pentium IV processor, 2.0 GHz and 1G RAM, and coding by MATLAB. The average execution time for fifty test constructions is shown in Table 6. The result shows that CLONALG and GA spend almost the same amount of time constructing 5 parallel tests with ab-

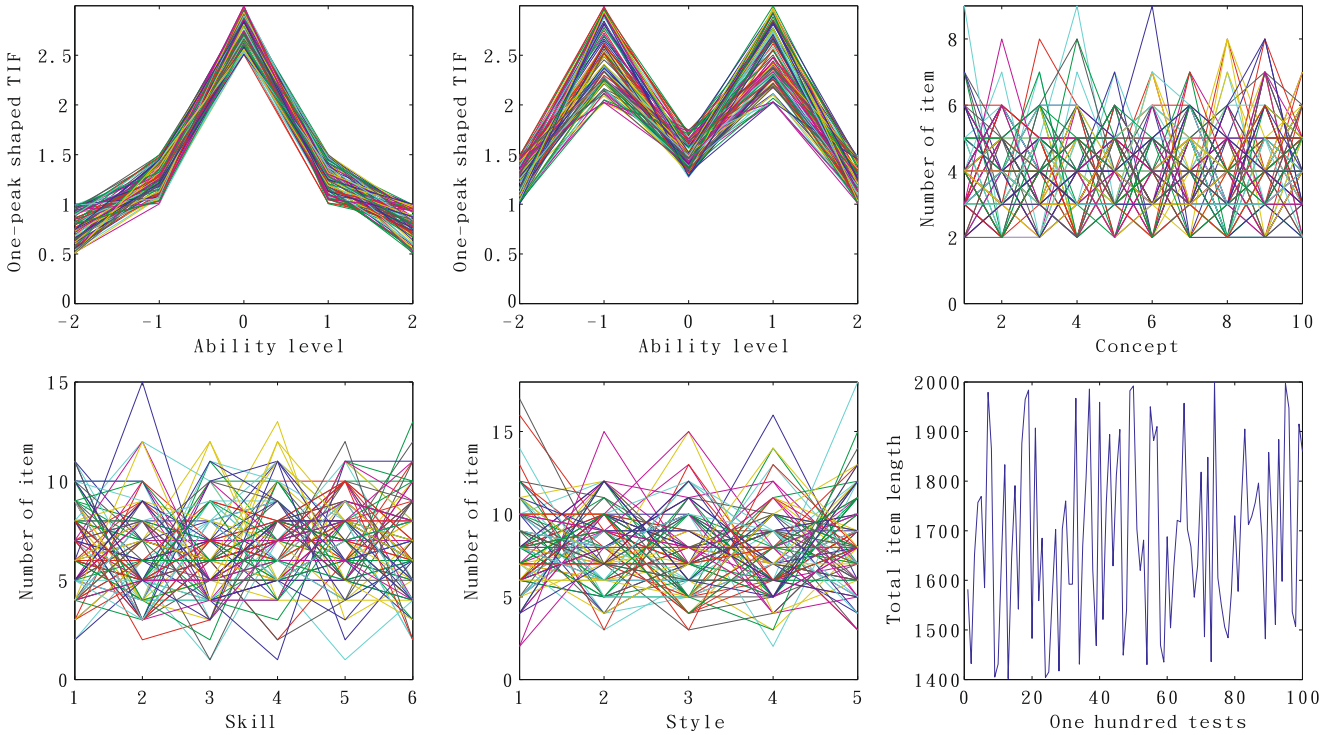


Fig. 12 Distributions of objective values for one hundred test specifications

Table 5 GA sequential construction vs. CLONALG simultaneous construction in relative TIF

	Relative information			
	Single-peak shaped TIF		Double-peak shaped TIF	
The sum of MSE	GA (Sequential)	CLONALG (Simultaneous)	GA (Sequential)	CLONALG (Simultaneous)
1st	1.12	0.07	1.37	0.07
2st	1.22	0.11	1.56	0.10
3st	1.50	0.15	1.40	0.15
4st	1.38	0.22	1.50	0.24
5st	1.30	0.34	1.25	0.32
Average	1.28	0.18	1.42	0.18
Ratio*	7.1	1	7.89	1

*Ratio = the average MSE of GA divided by the average MSE of CLONALG

Table 6 Average execution times for CLONALG, GA and LP

	Absolute information			Relative shape		
	LP	GA	CLONALG	LP	GA	CLONALG
Execution times (mins)	> A week	2.35	2.33	> A week	2.75	2.31

solute TIF, but CLONALG has a more accurate result (see Fig. 11). In the relative TIF experiment, the execution time for CLONALG was shorter than GA, because CLONALG used for relative TIF employs the two-stage strategy and the second stage has a small population size ($P = F$). We can see that CLONALG uses the independent evolution strategy to maintain solution diversity. It is suitable for finding multiple solutions in the search space at the same time. Obviously, the adapted CLONALG takes an additional cost to operate *LIST* which avoids the item is selected by other tests. But it only requires a low cost about 0.7 seconds in our experiment. The result also indicates that even though

the population size is set to a small value, CLONALG still has a great search capability to construct a near-optimal test.

7 Conclusions

This study presented an adapted CLONALG algorithm for simultaneously constructing parallel tests with identical TIF and test specifications. This method greatly simplifies the simultaneous parallel tests construction model into a single test construction model, thereby avoiding the larger number of variables and constraints problem. The proposed algorithm enhances the quality of the constructed parallel tests. The experimental results indicate that the adapted CLONALG is a desirable and stable method for simultaneously constructing near-optimal parallel tests from a large item bank. Compared with GA and LP using sequential construction, the proposed method has a low deviation and can also construct parallel tests with identical test specifications. Overall, the proposed method has several important advantages.

1. *Simultaneous construction*. This method has neither the inequality problem in the sequential construction, nor a larger number of variables and constraints, which greatly decreases the computational complexity in simultaneous construction.
2. *Excellent search capability*. Even though the item bank size is up to 4000 and the objective test specification is more accurate (identical test specifications), CLONALG constructs the parallel tests simultaneously within acceptable time. Moreover, the more the item bank size is increased, the better the proposed method is able to highlight the search capability of heuristic algorithms in solving a large-scale problem.
3. *Flexible*. CLONALG is flexible in solving various constraints in that it only requires inputting the constraint equations into the affinity function.
4. *Uniform sampling*. Because CLONALG selects items from the item bank randomly, this way provides every feasible item combination (test) with an equal chance of being selected during a construction procedure [5].

Based on the quality of the results, CLONALG can be suitably applied to evaluating student learning status and applied to actual evaluation systems in future work.

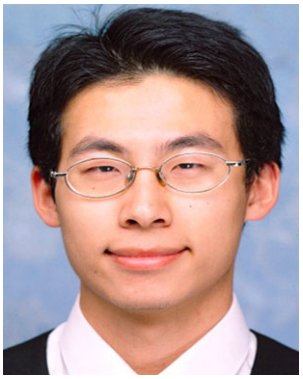
Acknowledgements This research was partially supported by the National Science Council, Taiwan, ROC, under contract no.: NSC99-2622-E-018-005-CC3.

References

1. Ackerman TA (1989) An alternative methodology for creating parallel test forms using the IRT information function. In: The annual meeting of the national council for measurement in education
2. Adema JJ (1992) Methods and models for the construction of weakly parallel tests. Appl. Psychol. Meas. 16:53–63
3. Armstrong RD, Jones DH (1992) Polynomial algorithms for item matching. Appl. Psychol. Meas. 16:365–371
4. Armstrong RD, Jones DH, Li X, Wu IL (1996) A study of a network-flow algorithm and a noncorrecting algorithm for test assembly. Appl. Psychol. Meas. 20:89–98
5. Belov DI, Armstrong RD (2005) Monte Carlo test assembly for item pool analysis and extension. Appl. Psychol. Meas. 29:239–261
6. Boekkooi-Timminga E (1987) Simultaneous test construction by zero-one programming. Methodika 1:101–112
7. Boekkooi-Timminga E (1990) The construction of parallel tests from IRT-based item banks. J. Educ. Behav. Stat. 15:129–145
8. Cor K, Alves C, Gierl MJ (2008) Conducting automated test assembly using the premium solver platform version 7.0 with Microsoft excel and the large-scale LP/QP solver engine add-in. Appl. Psychol. Meas. 32:652–663
9. Cor K, Alves C, Gierl MJ (2009) Three applications of automated test assembly within a user-friendly modeling environment. Pract. Assess. Res. Eval. 14:1–23
10. de Castro LN, Von Zuben FJ (2002) Learning and optimization using the clonal selection principle. IEEE Trans. Evol. Comput. 6:239–251
11. Finkelman M, Kim W, Roussos LA (2009) Automated test assembly for cognitive diagnosis models using a genetic algorithm. J. Educ. Meas. 46:273–292
12. Goldberg D, Korb B, Deb K (1989) Messy genetic algorithms: Motivation, analysis, and first results. The Clearinghouse for Genetic Algorithms (TCGA), Report 89003
13. Hambleton RK, Swaminathan H (1985) Item response theory: principles and applications. Kluwer Academic, Amsterdam
14. Hwang GJ, Lin BMT, Tseng HH, Lin TL (2005) On the development of a computer-assisted testing system with genetic test sheet-generating approach. IEEE Trans. Syst. Man Cybern., Part C, Appl. Rev. 35:590–594
15. Hwang GJ, Yin PY, Yeh SH (2006) A tabu search approach to generating test sheets for multiple assessment criteria. IEEE Trans. Ed. 49:88–97
16. Hwang GJ, Chu HC, Yin PY, Lin JY (2008) An innovative parallel test sheet composition approach to meet multiple assessment criteria for national tests. Comput. Educ. 51:1058–1072
17. Korkmaz EE (2010) Multi-objective Genetic Algorithms for grouping problems. Appl. Intell. 33:179–192
18. Lee CL, Huang CH, Lin CJ (2007) Test-sheet composition using immune algorithm for e-learning application. In: New Trends in Applied Artificial. LNSC, vol. 4570, pp. 823–833
19. Linhares A, Yanasse HH (2010) Search intensity versus search diversity: a false trade off? Appl. Intell. 32:279–291
20. Lord FM (1980) Applications of item response theory to practical testing problems. Lawrence Erlbaum
21. Ombuki BM, Ventresca M (2004) Local search genetic algorithms for the job shop scheduling problem. Appl. Intell. 21:99–109
22. Ray SS, Bandyopadhyay S, Pal SK (2007) Genetic operators for combinatorial optimization in TSP and microarray gene ordering. Appl. Intell. 26:183–195
23. Shewchuk JR (1994) An introduction to the conjugate gradient method without the agonizing pain. Tech Rep
24. Sun KT, Chen SF (1999) A study of applying the artificial intelligent technique to select test items. Psychol. Test. 46:75–88
25. Sun KT, Chen YJ, Tsai SY, Cheng CF (2008) Creating IRT-based parallel test forms using the genetic algorithm method. Appl. Meas. Educ. 21:141–161
26. Theunissen TJJM (1985) Binary programming and test design. Psychometrika 50:411–420

27. van der Linden WJ (1998) Optimal assembly of psychological and educational tests. Appl. Psychol. Meas. 22:195–211
28. van der Linden WJ (2005) Linear models for optimal test design. Statistics for social and behavioral sciences. Springer, Berlin
29. van der Linden WJ, Adema JJ (1998) Simultaneous assembly of multiple test forms. J. Educ. Meas. 35:185–198
30. Verschoor AJ (2007) Genetic algorithms for automated test assembly. PhD thesis
31. Wu IL (2001) A new computer algorithm for simultaneous test construction of two-stage and multistage testing. J. Educ. Behav. Stat. 26:180–198
32. Yin PY, Chang KC, Hwang GJ, Hwang GH, Chan Y (2006) A particle swarm optimization method to composing serial test-sheets for multiple assessment criteria. Educ. Technol. Soc. 9:3–15

Ting-Yi Chang received his M.S. from the Graduate Institute of Computer Science and Information Engineering at Chaoyang University of Technology, and his Ph.D in the Department of Computer Science at National Chiao Tung University, Taiwan.Currently, he is an Associate Professor with the Graduate Institute of e-Learning, National Changhua University, Taiwan. His current research interests include artificial intelligence, e-Learning, information security, cryptography, and mobile communications.



You-Fu Shiu received his M.S. degree from the Graduate Institute of e-Learning at National Changhua University, Taiwan. His research interests include e-Learning and evolutionary algorithms.