Maximum Clique Algorithm for Uniform Test Forms Assembly

Takatoshi Ishii[1], Pokpong Songmuang[2], and Maomi Ueno[1]

[1] University of Electro-Communications, Tokyo, Japan
{ishii,ueno}@ai.is.uec.ac.jp
[2] Thammasat University, Pratumthani, Thailand

Abstract. Educational assessments occasionally require "uniform test forms" for which each test form consists of a different set of items, but the forms meet equivalent test specifications (i.e., qualities indicated by test information functions based on item response theory). We propose two maximum clique algorithms (MCA) for uniform test forms assembly. The proposed methods can assemble uniform test forms with allowance of overlapping items among uniform test forms. First, we propose an exact method that maximizes the number of uniform test forms from an item pool. However, the exact method presents computational cost problems. To relax those problems, we propose an approximate method that maximizes the number of uniform test forms asymptotically. Accordingly, the proposed methods can use the item pool more efficiently than traditional methods can. We demonstrate the efficiency of the proposed methods using simulated and actual data.

Keywords: test assembly, uniform test forms, maximum clique problem, item response theory.

1 Introduction

Educational assessments occasionally require "uniform test forms" for which each form consists of a different set of items but which still must have equivalent specifications (e.g., equivalent amounts of test information based on item response theory, equivalent average test score, equivalent time limits). For example, uniform test forms are necessary when a testing organization administers a test in different time slots. To achieve this, uniform test forms are assembled in which all forms have equivalent qualities so that examinees who have taken different test forms can be evaluated objectively using the same scale.

Recently, automatic assembly for test forms has become popular. Automatic assembly assembles test forms to satisfy given test constraints (e.g., number of test items, amount of test information, average test score) to provide equivalent qualities [16,22,9,3,1,2,14,4,24,7,23,8,21,20,6].

In these studies, a test assembly is formalized as a combinational optimization problem. For example, van der Linden [23] proposed the big-shadow-test method using linear programming (LP). This method sequentially assembles uniform

K. Yacef et al. (Eds.): AIED 2013, LNAI 7926, pp. 451–462, 2013.

test forms by minimizing qualitative differences between a current assembled test form and the remaining set of items in an item pool. Although this method assembles uniform test forms in a practically acceptable time, it presents two problems. First, qualitative differences increase with the assembled order of test forms. Secondly, this method does not maximize the number of uniform test forms from the item pool.

To alleviate or ameliorate the first problem, Sun et al. [21] proposed a Genetic Algorithm (GA) for uniform tests assembly that simultaneously assembles uniform test forms as minimizing the differences among the qualities of assembled test forms and user-determined values. Furthermore, Songmuang and Ueno [20] applied the Bees Algorithm to uniform test forms assembly and to improve the performance of the method proposed by Sun et al. [21]. Although these methods [23,21,20] showed effective performance for minimizing the qualitative differences among assembled test forms, no method maximizes the number of uniform test forms from the item pool. These methods do not allow the item pool to be used efficiently to the greatest degree possible.

To maximize the number of test forms, Belov and Armstrong [8] proposed a uniform tests assembly method based on Maximum Set-Packing Problems. Moreover, Belov proposed a random test assembly method [6] to improve the tractability of maximizing the number of uniform test forms. However, these methods [8,6] cannot assemble uniform test forms with overlapping items (i.e., two test forms are allowed to have a common item called an overlapping item). In the non-overlapping conditions, each item is used only at once on assembled test forms. Therefore, the non-overlapping condition strongly restricts the number of assembled test forms. Consequently, the non-overlapping condition interrupts the efficient uses of the item pool.

The goal of this paper is to propose a uniform test forms assembly method that maximizes the number of assembled test forms with overlapping conditions. To achieve this goal, we apply the Maximum Clique Algorithm (MCA). MCA is an algorithm that solves the Maximum Clique Problem. We propose an exact method based on Maximum Clique Problem (ExMCP) for the maximum number of uniform test forms from the item pool.

The unique feature of ExMCP is to generalize Belov and Armstrong's method [8] to maximize the number of uniform test forms with an overlapping condition. Therefore, theoretically, ExMCP can assemble a greater number of test forms than when using traditional methods (e.g., [23,21,8,20]). In fact, ExMCP is expected to use the item pool more efficiently than traditional methods do.

However, the computational time and space costs of ExMCP increase exponentially with the number of "feasible test forms" (i.e., a set of those test forms which satisfy all test constraints except for the overlapping constraint from a given item pool). Therefore, it is difficult to use ExMCP for a large item pool.

To relax this problem, we propose RndMCP by approximating ExMCP using a random search approach (e.g., [19]). RndMCP maximizes the number of uniform test forms asymptotically from the item pool with overlapping conditions, and assembles a greater number of test forms than those of traditional methods (e.g., [23,8]). In addition, RndMCP searches the maximum number of uniform test forms more efficiently than traditional random search methods do [21,20] because the search space of RndMCP is more restrictive than those of the traditional methods.

Moreover, some experiments were conducted to evaluate the proposed methods. The results demonstrate that the proposed methods assemble a greater number of uniform test forms than the traditional methods do.

2 Maximum Clique Algorithm for Uniform Test Forms Assembly

In this section, we propose new methods to maximize the number of assembled uniform test forms with overlapping conditions.

2.1 Maximum Clique Problem

We apply the Maximum Clique Algorithm (MCA) to assemble the maximum number of uniform test forms. The MCA is an algorithm to solve the Maximum Clique Problem (MCP), which is a well-known combinational optimization problem in graph theory [15,11].

As described in this paper, a graph is represented as a pair $G = \{V, E\}$, where V denotes a set of vertices, and E denotes a set of edges.

Maximum Clique Problem searches a special structure called "Maximum Clique" from a given graph. "Clique" is a set of vertices in which each pair of vertices is connected. The "Maximum Clique" is the clique which has the maximum number of vertices in the given graph.

2.2 Maximum Clique Algorithm for Uniform Test Forms Assembly

In our study, the maximum number of uniform test forms is assembled to solve the maximum clique problem.

We assemble the following "Uniform test forms":

1. Any test form satisfies all test constraints.
2. Any two test forms satisfy the overlapping constraint. (i.e., any two test forms have fewer overlapping items than the allowed number in the overlapping constraint).

Accordingly, the maximum number of uniform test forms assembly can be described as the maximum clique extraction from a graph:

$$V = \left\{ \begin{array}{l} s : s \in S, \text{“Feasible test form”}, s \\ \quad \text{satisfies all test constraints} \\ \quad \text{excepting the overlapping constraint} \\ \quad \text{from a given item pool} \end{array} \right\}$$

$$E = \left\{ \begin{array}{c} \{s', s''\} : \text{The pair of } s' \text{ and } s'' \text{ satisfies} \\ \text{the overlapping constraint} \end{array} \right\}.$$

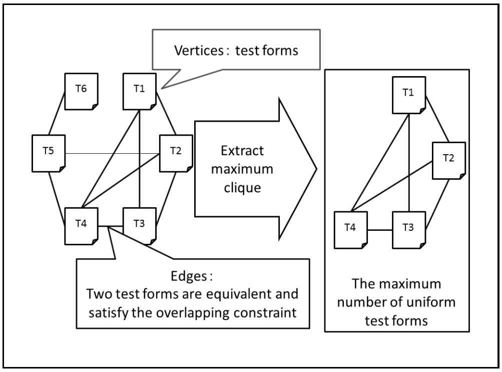

Fig. 1. MCA for Uniform Tests Assembly

This maximum clique problem searches the maximum set of feasible test forms in which any two test forms satisfy the overlapping constraint (i.e., this set is the maximum uniform test forms). Therefore, this optimization problem theoretically maximizes the number of uniform test forms. Figure 1 presents an example of uniform test forms assembly using the maximum clique problem. The graph G has six feasible test forms T1–T6 with nine satisfactions of overlapping constraint and the maximum number of uniform test forms $C_{max} = \{\mathrm{T1}, \mathrm{T2}, \mathrm{T3}, \mathrm{T4}\}$.

Belov and Armstrong's method [8] is a special case of this maximum clique problem when $E = \{$ { v,w }: v and w have no overlap items ($v \cap w = \emptyset$) $\}$. Therefore, our method generalizes Belov and Armstrong's method by relaxing the overlapping constraint.

2.3 Exact Solution: ExMCP

We propose a uniform tests assembly algorithm, "ExMCP", which exactly solves the maximum clique problem described in *Maximum Clique Algorithm for Uniform Test Forms Assembly*. Therefore, ExMCP theoretically maximizes the number of uniform test forms.

ExMCP consists of the following three steps:

Step 1: (assembling feasible test forms)
Step 1 assembles all feasible test forms. We use branch and bound technique (e.g., [3]) to assemble the feasible test forms using test constraints except for the overlapping constraint. Finally, Step 1 stores the feasible test forms into a system memory.

Step 2: (generating a graph that corresponds to a set of feasible test forms with overlapping items)
Step 2 generates the corresponding graph by counting overlapping items among each pair of feasible test forms. The feasible test forms are represented as vertices and satisfactions of the overlapping constraint are represented as edges. Thereby, only if a pair of test forms has fewer common items than

the overlapping constraint do two vertices representing the pair of test forms have an edge.

Step 3: (extracting the maximum clique from the graph)
Step 3 extracts the maximum clique from the graph generated in Step 2. The extracted maximum clique represents the maximum number of uniform test forms that satisfy all test constraints including the overlapping constraint. To obtain the maximum clique, we use Nakanishi and Tomita's algorithm [17], which is the fastest exact algorithm in MCA.

ExMCP guarantees to extract the maximum number of uniform test forms with overlapping conditions from all combinations of feasible test forms from an item pool. However, the computational time and space costs are $O(2^F)$ and $O(F^2)$, where F is the number of feasible test forms from an item pool. Consequently, ExMCP is not available for large item pools.

2.4 Approximate Solution: RndMCP

To relax the computational costs problem, we approximate ExMCP using a random search approach. This method is designated as "RndMCP", which maximizes the number of uniform test forms asymptotically.

Although RndMCP consists of three steps similar to those of ExMCP, RndMCP repeats the three steps using a random search approach until it satisfies the three following constraints for computational costs:

$\mathbf{C_1}$ is the number of feasible test forms assembled in Step 1,
$\mathbf{C_2}$ is the time limit of Step 3,
$\mathbf{C_3}$ is the total time limit of the test assembly.

Details of the steps are the following.

Step 1: (assembling feasible test forms randomly)
Step 1 randomly assembles feasible test forms. Step 1 continues this step until the number of feasible test forms reaches $\mathbf{C_1}$. Finally, Step 1 stores the feasible test forms into the system memory.

Step 2: (generating a graph that corresponds to a set of feasible test forms with overlapping items)
Step 2 generates the corresponding graph by counting the overlapping items among feasible test forms similarly to ExMCP.

Step 3: (extracting the maximum clique)
Although Step 3 extracts the maximum clique from the graph similarly to ExMCP, the computation time of this step is limited by $\mathbf{C_2}$.

Step 4: (controlling the computation time)
Step 4 compares the current largest clique and the result of Step 3. Step 4 stores the larger clique as the largest clique. If the computation time is less than $\mathbf{C_3}$, then jump to Step 1.

The computational time cost of RndMCP is $\mathbf{C_3}$, and the space cost of RndMCP is $O({\mathbf{C_1}}^2)$. By controlling the computational time and space costs, RndMCP relaxes the computational costs problem in ExMCP.

RndMCP repeatedly extracts the maximum number of uniform test forms from subsets that are sampled randomly from all of feasible test forms. Therefore, it asymptotically assembles the maximum number of uniform test forms.

Moreover, this method searches the maximum number of uniform test forms more efficiently than the traditional random search methods [21,20] do because the search space of RndMCP is more restrictive than that of the traditional methods. The traditional methods have $O(2^F)$ search space size, but RndMCP (and ExMCP) has $O(2^{0.19171F})$ search space because this depends on Nakanishi and Tomita's MCA [17]. (This size is an upper bound of the search space size of maximum clique algorithm and might be more restricted when MCA research progresses)

3 Experiments and Results

We demonstrate the respective performances of the proposed methods using two experiments.

We used item response theory (IRT) to measure the quality of test forms similarly to most previous studies of test form assembly (e.g., [25,10,5,4,23,20]). We use simulated item pools in the first experiment and actual item pools in the second experiment.

The items in the simulated and actual item pools have discrimination parameter a and the difficulty parameter b in item response theory. In the simulated item pool, the discrimination parameter a is distributed as $a \sim U(0, 1)$, and the difficulty parameter b is distributed as $b \sim N(0, 1^2)$. The actual item pools use the Synthetic Personality Inventory (SPI) examination [18], which is a popular aptitude test in Japan. Table 2 presents details of the actual item pools.

We compared the performances of ExMCP and RndMCP with those of the traditional methods [23,21,20]. For that comparison, we used CPLEX [12] for the liner programming method in Linden's method. Table 1 shows details of computational environment for all experiments.

3.1 Results for the Simulated Item Pool

In the previous section, we described that ExMCP theoretically maximizes the number of uniform test forms. In this experiment, we present the performances of proposed methods experimentally using the simulated item pools.

We compare the number of assembled test forms with ExMCP, RndMCP, and the traditional methods [23,21,20].

We use six simulated item pools and three constraints. The item pools have the total quantities of items $I = 70$, 80, 90, 100, 110, and 120. The three constraints have common test constraints as follows:

1. The test length was four.
2. The allowed quantities of overlapping items were $0, 1$and 2.

Table 1. Computation Environment

CPU	Intel(R) Xeon(R) E5640 2.67 GHz
System Memory	12.0 GB
OS	Windows 7 SP1 64bit

Table 2. Details of the Actual Item Pool

Item Pool	Parameter a			Parameter b		
Size	Range	Mean	SD	Range	Mean	SD
87	0.15∼0.67	0.35	0.134	-2.09∼4.55	0.73	1.625
93	0.19∼0.69	0.43	0.122	-3.92∼3.61	-0.79	1.196
104	0.13∼1.10	0.59	0.213	-0.18∼4.55	1.50	1.188
141	0.24∼1.09	0.64	0.155	-1.41∼3.91	0.60	0.855
158	0.15∼3.08	0.44	0.255	-4.00∼4.00	-1.12	1.434
175	0.12∼0.93	0.39	0.139	-2.93∼3.12	-0.25	1.113
220	0.16∼0.92	0.46	0.155	-4.00∼2.82	-1.28	1.098

Table 3. Constraints of the Information Function

Constraint	Information Function (Lower /Upper Bound)				
ID	$\theta = -2.0$	$\theta = -1.0$	$\theta = 0$	$\theta = 1.0$	$\theta = 2.0$
1	0.1/0.2	0.2/0.3	0.4/0.5	0.2/0.3	0.1/0.2
2	0.0/0.2	0.1/0.3	0.3/0.5	0.1/0.3	0.0/0.2
3	0.0/0.4	0.1/0.5	0.3/0.7	0.1/0.5	0.0/0.4

In addition, the three constraints have different information constraints among the constraints. The information constraint is described by the lower and upper bounds of the test information function $I(\theta_k)$. Those information constraints are listed in Table 3. These restrict the number of feasible test forms (and assembled test forms) to ID: 1 < ID: 2 < ID: 3.

For the traditional methods [23,21,20], we determined the target values of information function $T(\theta_k)$ as

$$T(\theta_k) = \frac{(Lowerboundsofinformationfunction) + (Upperboundsofinformationfunction)}{2}.$$

The time limitation of test assembly is 6 hr for all methods except for RndMCP.

For RndMCP, we determined the respective computational cost constraints $\mathbf{C_1}$ as 100000, $\mathbf{C_2}$ as 60 s, and $\mathbf{C_3}$ as 1400 s.

Table 4 presents the quantities of test forms assembled by the proposed methods and the traditional methods for the item pool sizes, the overlapping constraint (maximum number of overlap items) and information constraints. In the table, "BST" denotes Linden's method [23],"GA" denotes Sun's method[21], "BA" denotes Songmuang's method[20], "EM" denotes the proposed ExMCP, and "RM" denotes the proposed RndMCP.

In many cases, ExMCP failed the test assembly because it did not complete the calculations in 6 hr (†). Moreover, it was unable to assemble uniform test

Table 4. Results for the Simulated Item Pool

Item Pool	Overlap	Constraint ID: 1					Constraint ID: 2					Constraint ID: 3				
Size	Constraint	BST	GA	BA	EM	RM	BST	GA	BA	EM	RM	BST	GA	BA	EM	RM
70	0	1	0	1	1	1	6	6	7	8†	7	7	7	7	8†	8
	1	2	0	1	2	2	17	26	48	66†	67	17	58	59	0‡	99
	2	3	0	2	3	3	17	66	214	736†	735	17	274	278	0‡	1767
80	0	2	1	2	2	2	7	8	8	9†	9	7	8	8	0‡	9
	1	11	2	11	12†	11	20	40	64	100†	100	20	74	78	0‡	131
	2	20	4	69	88†	88	20	82	242	1462†	1404	20	347	301	0‡	2825
90	0	2	1	2	2	2	8	7	8	10†	10	8	8	9	0‡	10
	1	13	3	11	13†	12	22	40	71	122†	119	22	83	86	0‡	156
	2	22	3	78	107†	107	22	81	251	1949†	1846	22	321	336	0‡	3634
100	0	2	1	2	2	2	8	7	8	10†	10	9	9	9	0‡	11
	1	13	3	11	12†	13	25	36	76	131†	130	25	88	87	0‡	173
	2	25	3	87	118†	118	25	80	292	2325†	2170	25	312	346	0‡	4288
110	0	2	1	2	2	2	8	8	9	10†	10	10	9	10	0‡	11
	1	13	3	11	13†	13	27	34	79	138†	137	27	86	92	0‡	195
	2	27	2	91	123†	123	27	70	308	2632†	2413	27	271	356	0‡	4938
120	0	2	2	2	2	2	9	6	9	11†	11	10	10	11	0‡	13
	1	13	2	10	13†	13	30	29	82	152†	150	30	92	102	0‡	229
	2	30	4	95	129†	127	30	68	336	2913†	2617	30	269	407	0‡	6006

†: The maximum number of uniform test forms detected in 6 hr.
‡: A memory insufficiency problem interrupted the test construction.

forms because the computational environment had insufficient system memory (‡). In † cases, ExMCP detected a greater number of uniform test forms than any other method in the given time limitation. In all cases, RndMCP assembled higher quantities of uniform test forms than the traditional methods did [23,21,20]. In addition, the computational time of RndMCP is less than the other random search methods (e.g., [21,20]). The computational time of RndMCP is $\mathbf{C_3} = 1400$ s, and the time limitations of the other random search methods are 6 hr. Results show that RndMCP provides more accurate results than the other random search methods do. Moreover, the difference of quantities of assembled test forms between the proposed method and the traditional methods increase with the number of assembled test forms (or the scale of assembly).

The results can be summarized as shown below.

1. ExMCP assembles the maximum number of uniform test forms, but it entails a computational cost problem.
2. Even when ExMCP fails a uniform test forms assembly by computational cost problem, RndMCP assembles a greater number of uniform test forms than the traditional methods do. Actually, RndMCP relaxes ExMCPś computational costs problem.
3. RndMCP assembled more quantities of uniform test forms in a shorter time than the other random search methods (e.g., [21,20]) did. Results show that RndMCP provides more accurate results than the other random search methods do.
4. The differences of the number of assembled test forms between the proposed methods and traditional methods increase with the number of feasible test forms (or the scale of test assembly). For large scale assembly, the proposed methods are more efficient than the traditional methods are.

3.2 Results for Actual Item Pool

We assemble uniform test forms using actual item pools to demonstrate the effectiveness of RndMCP in actual situations. ExMCP cannot assemble the test forms in an actual situation because the computational environment has insufficient resources.

We use six actual item pools that have total numbers of items I = 87, 93, 104, 141, 158, 175, and 220. The distributions of item parameters a and b in the item pool are given in Table 2.

We use the same test constraints as in *Results for the Simulated Item pool.* For RndMCP, we determine the computational costs constraint $\mathbf{C_1} = 100000$, $\mathbf{C_2} = 30$ s, and $\mathbf{C_3} = 6$ hr. All other assembly methods are also given 6 hr for calculation times.

Table 5. Results for the Actual Item Pool

Item Pool Size	Overlap Constraint	Constraint ID: 1				Constraint ID: 2				Constraint ID: 3			
		BST	GA	BA	RM	BST	GA	BA	RM	BST	GA	BA	RM
87	0	0	0	0	0	3	3	4	4	3	3	4	4
	1	0	0	0	0	16	10	19	29	14	11	20	27
	2	0	0	0	0	21	36	139	307	21	39	140	309
93	0	0	0	0	0	4	5	5	6	5	5	5	6
	1	0	0	0	0	23	16	33	51	23	16	33	51
	2	0	0	0	0	23	43	211	658	23	54	208	721
104	0	2	2	2	2	6	5	8	10	12	15	15	18
	1	6	5	9	10	26	26	71	131	26	171	140	369
	2	26	14	83	121	26	59	275	2088	26	590	394	8442
141	0	10	3	9	10	18	19	21	27	26	31	27	35
	1	35	5	70	150	6	122	188	589	35	506	239	1014
	2	35	20	268	2307	10	185	393	11426	35	1511	386	19095
158	0	0	0	0	0	6	1	5	6	6	4	7	8
	1	0	0	0	0	22	12	24	40	39	42	75	131
	2	0	0	0	0	39	50	137	316	39	94	279	4877
175	0	2	0	2	2	6	6	7	9	6	6	8	10
	1	12	1	13	15	43	53	96	186	43	65	100	193
	2	43	2	128	234	43	102	303	7030	43	103	283	7413
220	0	2	0	2	2	7	5	8	10	9	8	10	13
	1	8	2	7	17	54	20	87	177	54	57	124	282
	2	54	8	75	136	54	44	309	5889	54	114	334	9938

Table 5 presents the quantities of test forms assembled using the proposed method and the traditional methods for the item pool size, the overlapping constraint and information constraints.

Similar to simulated experiments, in all cases, RndMCP assembled greater quantities of uniform test forms than the traditional methods did [23,21,20]. Moreover, the difference quantities of assembled test forms between the proposed method and the traditional methods increase continuously with the number of assembled test forms.

The results can be summarized as follows:

1. RndMCP assembles a greater number of uniform test forms than the traditional methods do.
2. RndMCP assembled greater quantities of uniform test forms than the other random search methods (e.g., [21,20]) did during an equal time period. Results show that RndMCP provides more accurate results than the other random search methods do.
3. The differences of the number of assembled test forms between the proposed methods and traditional methods increase along with the number of feasible test forms (or the scale of test assembly).

The results show that RndMCP uses an item pool more efficiently than the traditional methods do.

4 Conclusion

We proposed two uniform test forms assembly methods, ExMCP and RndMCP, based on the Maximum Clique Algorithm. The proposed methods exactly or asymptotically maximize the quantities of uniform test forms with an overlapping condition.

ExMCP generalizes Belov's method [8] for overlapping conditions. Furthermore, it maximizes the number of uniform test forms with overlapping conditions. However, ExMCP presents computational costs problems. RndMCP approximates ExMCP using a random search approach to relax this computational costs problem. RndMCP assembles a greater number of uniform test forms than the traditional methods (e.g., [23,21,20]) do. Moreover, RndMCP provides more accurate results than other random search methods (e.g., [21,20]) do.

To demonstrate these features, we conducted two experiments using simulated and actual data. Both experiments show that proposed methods assemble a greater number of uniform test forms than the traditional methods do. Moreover, the differences of the number of assembled test forms between proposed methods and the traditional methods increases with the number of feasible test forms (or the scale of test assembly). This result shows that the proposed methods can assemble a greater number of uniform test forms than the traditional methods can.

In simulated experiments, more cases exist in which ExMCP cannot assemble uniform test forms because of computational cost problems. However in those cases, RndMCP assembles a greater number of uniform test forms than the traditional methods do. This result shows that RndMCP relaxes the computational cost problems of ExMCP.

In simulated experiments, the computational time of RndMCP is less than that of the other random search methods. In actual experiments, RndMCP assembles a greater number of test forms than the traditional methods do, given equal time limitations. Therefore, RndMCP provides more accurate results than other random search methods (e.g., [21,20]) do.

Results show the salient benefits of using the proposed methods.

References

1. Ackerman, T.A.: An alternative methodology for creating parallel test forms using the irt information function. Paper presented at the Annual Meeting of the National Council on Measurement in Education, San Francisco, CA, March 30 (1989)
2. Adema, J.J.: Methods and models for the construction of weakly parallel tests. Applied Psychological Measurement 16(1), 53–63 (1992)
3. Ameda, J.J.: Implementations of the branch-and-bound method for test construction problems. Project Psychometric Aspects of Item Banking, Department of Education, University of Twente, Research Report 89-6 (1989)
4. Armstrong, R.D., Jones, D.H., Kunce, C.S.: Irt test assembly using network-flow programming. Applied Psychological Measurement 22(3), 237–247 (1998)
5. Armstrong, R.D., Jones, D.H., Wang, Z.: Automated parallel test construction using classical test theory. Journal of Educational Statistics 19(1), 73–90 (1994)
6. Belov, D.I.: Uniform test assembly. Psychometrika 73(1), 21–38 (2008)
7. Belov, D.I., Armstrong, R.D.: Monte carlo test assembly for item pool analysis and extension. Applied Psychological Measurement 29, 239–261 (2005)
8. Belov, D.I., Armstrong, R.D.: A constraint programming approach to extract the maximum number of non-overlapping test forms. Computational Optimization and Applications 33, 319–332 (2006)
9. Boekkooi-Timminga, E.: Simultaneous test construction by zero-one programming. Methodika 1, 101–112 (1987)
10. Boekkooi-Timminga, E.: The construction of parallel tests from irt-based item banks. J. Educat. Statist. 15, 129–145 (1990), reports
11. Garey, M.R., Johnson, D.S.: Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York (1990)
12. ILOG: ILOG CPLEX User's Manual 11.0
13. Ishii, T., Songmuang, P., Ueno, M.: A method to extract the maximum number of test forms using maxclique. In: The 23rd Annual Conference of the Japanese Society for Artificial Intelligence (2009)
14. Jeng, H., Shih, S.: A comparison of pair-wise and group selections of items using simulated annealing in automated construction of parallel tests. Psychological Testing 44(2), 195–210 (1997)
15. Karp, R.M.: Reducibility among combinatorial problems. Complexity of Computer Computations 40(4), 85–103 (1972)
16. Lord, F.M.: Applications of Item Response Theory to Practical Testing Problems, 1st edn. Routledge (July 1980)
17. Nakanishi, H., Tomita, E.: An $o(2^{0.19171n})$-time and polynomial-space algorithm for finding a maximum clique. Information Processing Society of Japan SIG Technical Report 2008(6), 15–22 (2008)
18. Recruit: Synthetic Personality Inventory (SPI), http://www.spi.recruit.co.jp/
19. Solis, F.J., Wets, R.J.B.: Minimization by random search techniques. Mathematics of Operations Research 6(1), 19–30 (1981)
20. Songmuang, P., Ueno, M.: Bees algorithm for construction of multiple test forms in e-testing. IEEE Transactions on Learning Technologies 4, 209–221 (2011)
21. Sun, K.T., Chen, Y.J., Tsai, S.Y., Cheng, C.F.: Creating irt-based parallel test forms using the genetic algorithm method. Applied Measurement in Education 21(2), 141–161 (2008)
22. Theunissen, T.J.J.M.: Binary programming and test design. Psychometrika 50(4), 411–420 (1985)

23. van der Linden, W.J.: Liner Models for Optimal Test Design. Springer (2005)
24. van der Linden, W.J., Adema, J.J.: Simultaneous assembly of multiple test forms. Journal of Educational Measurement 35(3), 185–198 (1998)
25. van der Linden, W.J., Boekkooi-Timminga, E.: A maximin model for irt-based test design with practical constraints. Psychometrika 54(2), 237–247 (1989)