Supply Chain for a High Stakes Testing Agency

RONALD ARMSTRONG
Management Science and Information Systems
Rutgers University
Newark, NJ 07102-1895
USA

DMITRY BELOV
Psychometric Research
Law School Admission Council
662 Penn Street
Newtown, PA 18940

USA

MABEL KUNG
Information Systems and Decision Sciences
California State University at Fullerton
Fullerton, CA 92834

USA

Abstract: - A valuable resource for any testing agency is the item pool which yields the questions for its tests. This paper considers the problem of developing the supply chain that provides these items. In particular, the life cycle of items used in the Law School Admission Test (LSAT) is reviewed. Guidelines for the optimal development of items over a finite horizon are given. The assembly of the tests and the problems associated with supply chain development will be formulated as integer programming models. All problems will be solved with commercially available software.

Key Words: Supply Chain Management, Test Assembly, Item Response Theory, Integer Programming

1. Introduction

To deal with the complexity of supply chain life cycles, new strategies gaining wide acceptance when driving continuous improvement generally include the following key areas - situation analysis, planning and implementing changes, monitoring the results, and developing a closed-loop control systems [2,7,9]. Nevertheless, limited attention has been given to the issue of measuring the effectiveness of information sharing and performance evaluations across the supply chain.

The supply chain in this research corresponds to the development of items to maximize usage of the pool which produces the test questions. Proper management of the supply chain requires recognition of item characteristics because certain items are likely candidates to be selected on an operational test form, and other items not so likely. Earlier studies established the maximum number of non-overlapping forms in a pool [5]. However, new item development costs or the timing of the specifications were not considered. These issues are included in this paper.

A typical item pool would allow many ways to combine items to make a form, for example, heuristic methods [8], network flow and Lagrangian relaxation [1,3], large scale mixed integer programming (MIP) software [10]. The focus of this paper takes the last approach using CPLEX (ILOG, [6]) as the tool to help manage the supply chain. The results were generated from a Microsoft Access database and processed on a laptop (Pentium 2.13GHz, 2GB RAM, Windows XP). All MIP problems were solved with the CPLEX library.

2. Item Life Cycle

Items reaching an operational level pass through the writers, test specialists and psychometricians. In this study, items with multiple choice responses are considered and an item is either responded to correctly or incorrectly.

After the initial review, the items go to the "pre-test" stage where they are assigned to a *nonscored* section of the test. When examinees take the test, they are not aware of which sections are not scored or which are scored. Thus, reliable response data are collected on the new items without affecting examinees' scores. Most testing agencies utilize an item response theory (IRT) model to determine the difficulty and discrimination power of an item. This study uses the popular three-parameter logistics model where each examinee has an underlying latent trait, denoted by θ, that determines their ability to answer an item correctly. The values for θ can be placed on a standardized scale, so most examinees have a latent trait value between -3.0 and +3.0. The three IRT parameters for item *i* are given by a_i, b_i and c_i. The value of a_i indicates the discrimination power of the item, b_i relates to item difficulty and c_i provides a pseudo-guessing estimate. A Bernoulli random variable U_i is defined to equal 1 for a correct response and 0 otherwise. The equation used to determine the probability of a correct response to an item given θ is the following.

$$P(U_i = 1|\theta) = p_i(\theta) = c_i + \frac{1-c_i}{1+\exp(-1.7a_i(\theta - b_i))} \quad (1)$$

Further, the concept of information as a function of θ can be defined for the response model.

$$I_i(\theta) = (1.7a_i)^2 \left[\frac{1.0 - p_i(\theta)}{p_i(\theta)}\right]\left[\frac{p_i(\theta) - c_i}{1-c_i}\right]^2 \quad (2)$$

This study assumes that responses are conditionally independent. This allows total information and total expected number correct to be obtained by summing over the items on a test.

Based on the responses received in the pre-testing stage, statistically estimated values for a_i, b_i and c_i are obtained for each item. An item may be rejected at this point if the responses to the item differ significantly from the IRT model. The statistical analysis is performed by psychometricians in consultation with test specialists. If the item is rejected at this point, an attempt is made to identify the probable cause and communicate this to the writers.

3. Single Linear Test Form

This segment considers the problem of assembling a single linear test form from an item pool. The model is designed specifically for the Law School Admission Council that currently administers the Law School Admission Test (LSAT) in a linear pencil and paper format. The test consists of four scored sections; two in Logical Reasoning (LR), one in Analytical Reasoning (AR), and one in Reading Comprehension (RC). The AR and RC sections are set-based with 4 stimuli (passages) in each section, and the LR sections are composed of discrete items. All items are multiple choice and all recent administrations of the LSAT have contained 101 scored items. The LSAT is a high stakes test given to more than 130,000 prospective law school students. A descriptive summary of the constraints for the test assembly is reviewed here [3].

Pre-test positioning. Items must appear on the test form at approximately the same position in the operational form as in the pre-test form.

Cognitive skill content. A distribution of the cognitive skills being tested must be satisfied.

Item set specifications. When a stimulus is assigned to a form, an upper bound and a lower bound on the total number of items from the associated item set is required. Also, there may be items within the item set that must be used on the form.

Word count. A lower limit and an upper limit on the number of words in each section are specified.

Answer key count distribution. A constraint on the distribution of the multiple-choice answer keys is imposed.

Topic. The AR and RC sets are categorized according to topics. Each AR and RC section must have a specified number of stimuli of each topic.

Diversity. Certain stimuli are oriented toward a diversity group. Every RC section must have a specified diversity representation.

Targets. Each section has target information functions and target characteristic curves. These targets are enforced at discrete points on the latent trait scale -over -3.0 to +3.0 in steps of 0.3.

Enemies. There are items pairs and stimuli pairs that should not appear on the same form.

Overall mean test score. The overall mean score on the test should be within a specified range.

Number of items on the test. All forms have a fixed length (101 items).

An objective function used in the following analysis assigns random costs to the items.

$$Minimize \sum_{j \in J} u_j x_j , \quad (3)$$

where u_j is a uniform random number between 0 and 1. The set J gives the indices for all the items in the pool.

The study considered assembling a single form from an item pool with 1336 discrete LR items, 110 AR stimuli with 951 items, and 108 RC stimuli with 1021 items. For the assembly with one of the diversity restrictions enforced, the MIP problem had 5037 variables and 1936 constraints.

4. Multiple Forms

The assembly of a single test form is close to the assembly of four separate sections. The constraints that link the sections are constraints on the overall expected test score and overall IRT target goals. Thus, a viable approach for assembling multiple tests is to create many non-overlapping sections and combine them ensuring satisfaction of the constraints across the test. Suppose the maximum number of non-overlapping sections for each item type (AR, RC or LR) can be determined. Let MAX_{AR}, MAX_{RC} and MAX_{LR} represent these numbers. If the number of non-overlapping test forms equals $\min(MAX_{AR}, MAX_{RC}, MAX_{LR}/2)$, the maximum number of non-overlapping forms from the item pool would have been assembled. This observation leads to the test assembly approach of this paper.

The number of feasible *overlapping* sections that can be assembled from the item pool is large. However, if all feasible sections with unique stimuli combinations are generated, the maximum number of *non-overlapping* sections can be extracted from this set of overlapping sections. The following outlines the approach to assemble the maximum number of non-overlapping test forms from the item pool.

1. Assemble all, or many, overlapping sections satisfying the constraints on the individual item type. Do this for all three item types and save the sections in the database. Use the random cost objective for the items as given by (3) with new costs generated for each assembly.

2. Extract the maximum number of non-overlapping sections from the sections assembled in step 1. Since several section groupings may yield the maximum number of non-overlapping sections, more than one maximum group may be extracted. Do this for all three item types and save the extracted sections in a separate database table.

3. Combine the non-overlapping sections identified in step 2 to create the complete set of feasible forms using these sections.

4. Extract the maximum number of feasible non-overlapping forms from the set of forms created in step 3.

5. Multiple Overlapping Sections

Each AR and RC section contains exactly 4 stimuli with the associated set-based items. If the set-based constraints were satisfied by 4 stimuli, CPLEX was called to assemble a feasible section using only those 4 stimuli. If a feasible section was assembled with *n* items, the network was modified to consider the assembly of a section with the number of items restricted to equal other values feasible for the section. In our particular case, this meant 2

additional MIP problems were solved every time a feasible section was found. The problem of assembling a section given the 4 stimuli took less than 1 second. The total enumeration for the RC section required 4.5 hours and created 31,221 sections while the AR section required 20 hours and created 93,270 sections. The RC enumeration required considerably less time because the diversity and topic constraints eliminated many item set combinations before calling the MIP code.

As the size of the RC and AR sections increases, the complete enumeration approach may not be feasible. For example, an AR section of a pool with 106 stimuli has 4,967,690 combinations of 4 stimuli to consider. If the number of AR stimuli increased to 175, there would be 37,752,925 combinations of 4 stimuli. The complete enumeration of the LR sections with discrete items is not possible. Random costs and usage penalties were applied to obtain a large number of feasible sections, with 1200 LR sections assembled. Each MIP problem had the same constraints but the uniform random costs were generated anew for each problem and the item usage rate was multiplied by 10.0 to penalize frequently used items. The process was restarted after 100 sections were assembled. Every MIP problem had 1397 variables, 64 rows and 36,141 nonzero entries in the constraints. The time to obtain a solution within at least 50% of the optimal was 5 seconds, the total time to assemble 1,200 sections was 4 hours.

6. Non-Overlapping Sections

An MIP problem is created for each of the three item types. This zero-one programming problem is often referred to as the maximum set packing problem, or a maximum clique problem [11]. The overlapping sections were indexed sequentially as they were assembled. The problem statement is:

$$Maximize \quad \sum_{j=1}^{NS} \alpha_j \tag{4}$$

subject to

$$\sum_{j \in JS(i)} \alpha_j \le 1, \quad i = 1, ..., MS \tag{5}$$

$$\alpha_j = 0 \text{ or } 1 \quad j = 1, ..., NS \tag{6}$$

The value of NS gives the number of overlapping sections assembled as described previously, and MS is the number of stimuli used in the assembly of the sections for the item type. The index set $JS(i)$ gives all sections that contain stimulus i. The binary variable α_j equals one if section j is included in the set of non-overlapping sections with maximum cardinality.

The AR (RC, LR) section maximum set packing problem had 93270 (31221, 1200) zero-one variables and 105 (106, 1333) constraints. There were 5 (3, 3) AR (RC, LR) stimuli that could not be used in any section. These unusable stimuli could be removed from the pool. The total problem construction and solution took 20 (4, 30) minutes. There were 16 (17, 32) non-overlapping AR (RC, LR) sections. A solution yielding the maximum number of non-overlapping sections is referred to as a *maximum packing*.

If the rotation of the diversity found in the RC sections is added as an additional restriction, the number of non-overlapping RC sections dropped from 17 to 15. Denote the index sets of sections with one diversity orientation by $JD1$, the sections with another orientation by $JD2$, and the sections with the third orientation by $JD3$. The following constraints were added to the problem to allow an acceptable diversity rotation.

$$\sum_{j \in JD1} \alpha_j - \sum_{j \in JD2} \alpha_j + sd1 = 1 \tag{7}$$

$$\sum_{j \in JD1} \alpha_j - \sum_{j \in JD3} \alpha_j + sd2 = 1 \tag{8}$$

$$0 \le sd1 \le 2, \quad 0 \le sd2 \le 2. \tag{9}$$

Since complete enumeration was used for the AR and RC sections, the maximum number of non-overlapping tests that can be assembled is 15. It cannot be verified that the maximum number of non-overlapping LR sections is 32, because complete enumeration was not possible. However, if the additional diversity rotation restrictions were not enforced, it could still be verified that 16 is an upper bound on the number of non-overlapping test forms, because there were a maximum of 16 unique AR sections.

The objective coefficients of (4) can be perturbed to obtain a maximum packing with desired individuality. Each LSAT section has an upper and lower limit on the number of items in the section, but a fixed number of items must appear on a form. The assembly of a complete form would be facilitated if the number of items in the sections obtained from the maximum packing were likely to

sum to the number of items on a form ($NTEST$). Let $ns_{AR} + ns_{RC} + 2ns_{LR} = NTEST$. Similarly, the expected raw score for the sections should have a good likelihood of summing to the desired expected raw score on the form, $SCORE$. Let the raw score goal for each section be such that $ps_{AR} + ps_{RC} + 2ps_{LR} = SCORE$. The perturbed objective function coefficient was taken to be 1.0-0.01$|ps_{AR} - ps_j|$-0.01$|ns_{AR} - ns_j|$ for the AR sections, 1.0-0.01$|ps_{RC} - ps_j|$-0.01$|ns_{RC} - ns_j|$ for the RC sections, and 1.0-0.01$|ps_{LR} - ps_j|$-0.01$|ns_{LR} - ns_j|$ for the LR sections, where ps_j and ns_j are the mean score and number of items for section *j*, respectively. The mean section score is the sum of the section's mean item scores.

7. Non-Overlapping Tests

The final step in the test assembly process was to combine sections to create linear test forms that have no items or stimuli in common and satisfy the test specifications. The set of feasible sections that were used consisted of three distinct maximum packings for AR, two distinct maximum packings for RC, and a single maximum packing for LR. The combinations of these sections can be enumerated to determine all feasible test forms. The only constraints that need be checked are the overall test constraints on IRT targets, overall expected score and enemy constraints across the two LR sections. Let NT be the number of feasible forms and let MT be the number of stimuli appearing more than once in these forms. Define $JT(i)$ to be the index set of forms containing stimulus *i*. To determine the number of non-overlapping forms possible by selecting from these forms, another maximum set packing problem is solved.

$$Maximize \quad \sum_{j=1}^{NT} \beta_j \tag{10}$$

subject to

$$\sum_{j \in JT(i)} \beta_j \le 1, \quad i = 1, ..., MT \tag{11}$$

$$\beta_j = 0 \text{ or } 1, \quad j = 1, ..., NT \tag{12}$$

The binary variable β_j=1 if form *j* was chosen for the maximum packing and β_j=0, otherwise. The total problem had 64,573 variables (one for each feasible form), 966 rows and 3,796,694 nonzero constraint coefficients. Before the use of the branch and cut procedure, CPLEX reduced the problem to 131 rows and 529,503 nonzero coefficients. The solution to (10), (11) and (12) did produce the 15 non-overlapping forms in less than 1 hour. The enumeration was not overly time consuming, but querying the database to establish the $JT(i)$'s was the main task. After dropping the requirement of diversity rotation and redoing the process with constraints (7), (8), and (9) omitted, 16 non-overlapping test forms were created. Since there are four administrations in one year, the inventory of tests at this point covers four years.

8. Items in the Supply Chain

The characteristics of items developed by the writers drive the supply chain. Consider the item development process over a one-year horizon. Four tests, a one-year supply, are taken from the set of tests assembled as described in the previous section. The items associated with these test are removed from the item pool. Some of the items removed from the pool have characteristics critical for the assembly of future tests and some of the items may not be critical. The objective is to add new items that minimize the cost of development and provide the four additional tests; that is, the tests removed from the pool are replenished. The following method is used to identify critical item and stimulus characteristics.

Let JN represent the index set of new items proposed for addition to the item pool. Some of these items may have already been written, but not yet pre-tested. Most of the desired item characteristics can be controlled by the writers. The IRT parameters are the most critical item characteristics to increase pool usage [4]. Experienced writers and test specialist provide reasonable parameter estimates. Item difficulty modeling can assist in approximating the IRT parameters prior to pre-testing.

The fundamental approach to assist in the management of the supply chain is to add items that shift the distributions pool characteristics to more closely resemble the distributions found in the assembled forms. Define a pseudo item to be an item without text but with all the characteristics required for test assembly. Add pseudo items to create the desired distributions for the pool.

Let m represent the number of forms desired from the pool augmented with the pseudo items. Repeat the assembly process described above, but with the following modification when creating the non-overlapping forms. Let h_j represent the anticipated development costs for the items in the j^{th} section assembled at step 1. All items currently in the pool have a cost of zero, and those that are already under contract but not pre-tested have their cost calculated without the sunken costs included.

$$Minimize \quad \sum_{j=1}^{NT} h_j \beta_j \tag{13}$$

subject to

$$\sum_{j \in JT(i)} \beta_j \leq 1, \quad i = 1, ..., MT \tag{14}$$

$$\sum_{j=1}^{NT} \beta_j = m \tag{15}$$

$$\beta_j = 0 \text{ or } 1, \quad j = 1, ..., NT \tag{16}$$

All item indices $i \in JN$ appearing in section j with $\beta_j = 1$ correspond to items that should be added to supply chain demand.

9. Conclusion

This paper utilizes quality control in a supply chain context. A performance procedure is presented to assist in the management of the delivery and assembly of test forms in the decision making of job flows for the LSAT. Methods for data manipulation, model building, and preprocessing are given. The knowledge of the items to be developed and the timing of the development enable cost reductions and an increased flexibility in administering forms. The sequential assembly of test forms may have a tendency to select, in the first few forms, items with statistical characteristics that make them good candidates. The assembly of the maximum number of non-overlapping forms would disperse these "good" items across multiple forms.

It is important for testing agencies to appraise the strength of their pools and assess future needs. Items are a valuable commodity and the future development of items should not duplicate the attributes of the unused items [10]. This research was designed to assemble test forms meeting specifications. The objective function may be varied to promote better curve fitting at some expense to solution time. The research perspectives include the pertinence of action toward a framework in a procedural dimension of a supply chain performance life cycle.

Acknowledgements: This Research has been funded by the Law School Admission Council.

References:

[1]. Ahuja, R., Magnanti, T. & Orlin, J, *Network Flows: Theory, Algorithms and Applications*, Prentice Hall, 2000.

[2]. Angerhofer, B. & Angelides, M., A model and a performance measurement system for collaborative supply chains, *Decision Support Systems*, Vol.42, 2006, pp.283-301.

[3]. Armstrong, R., Belov, D., & Weissman, A., Developing and Assembling the Law School Admission Test, *Interfaces*, Vol.35, 2005, pp.140-151.

[4]. Belov, D. & Armstrong, R., Monte Carlo Test Assembly for Item Pool Analysis and Extension, *Applied Psychological Measurement*, Vol. 29, 2005, pp.239-261.

[5]. Belov, D. & Armstrong, R., A constraint programming approach to extract the maximum number of non-overlapping test forms, *Computational Optimization and Applications*, Vol.33, 2006, pp.319-332.

[6]. ILOG, Inc. (2003). *CPLEX 9.0 [Computer program and manual]*. Mountain View, CA: Author. [www.ilog.com]

[7]. Lohman, C., Fortuin, L., & Wouters, M., Designing a performance measurement system: a case study, *European Journal of Operational Research*, Vol.156, 2004, pp.267-286.

[8]. Luecht, R., Computer assisted test assembly using optimization heuristics, *Applied Psychological Measurement*, Vol.22, 1998, pp.224-236.

[9]. Sahin, F. & Robinson, E., Information sharing and coordination to make to order supply chain, *Journal of Operations Management*, Vol.23, 2005, pp.579-598.

[10]. van der Linden, W., Veldkamp, B. & Reese, L., An integer programming approach to item bank design. *Applied Psychological Measurement*, Vol.24, 2000, pp.139-150.

[11]. Wood, D., An algorithm for finding a maximum clique in a graph. *Operational Research Letters*, Vol.21, 1997, pp.211-217