Technical Report on Automated Test Assembly of Italian and Mathematics INVALSI Tests for Grades 8 and 10

S.Y. 2017/2018

Mariagiulia Matteucci[1]
Stefania Mignani[1]
Giada Spaccapanico Proietti[1]
Bernard P. Veldkamp[2]
Angela J. Verschoor[3]

November 7, 2018

[1]Department of Statistical Sciences, University of Bologna, Italy
[2]Department of Research Methodology, Measurement and Data Analysis, University of Twente, The Netherlands
[3]Cito, The Netherlands

Preface

In educational measurement, a test is a collection of items developed to measure students' abilities. A key aspect is that, as with any other measurement instrument, different measurements of the same ability should be comparable. In order to meet this requirement, tests should be standardized. A test is considered to be standardized when the test procedures are fixed in such a way that differences among testing conditions, times, and places, do not influence the scores [13]. The accuracy of measurement (test reliability), and the degree to which a test actually measures the ability it is supposed to measure (test validity) are fundamental issues in the development of standardized tests and in the production of a fair scoring system. As the test takers do not necessarily get the same test, it is fundamental that the conditions of reliability and validity are fulfilled through a systematic approach to test development, in which the assembly procedures play a very important role.

According to [3], several phases in such an approach can be distinguished: project plan, content definition, test specifications, item development, item banking, pretesting, item bank calibration, test assembly, test production, test administration and reporting results. Several methods are used for test assembly nowadays. A common practice is selection by hand, usually after item analysis either based on classical test theory (CTT) or item response theory (IRT). Larger testing programs, however, have better access to resources like sophisticated item banking systems, opening the possibility to improve their test assembly process by means of automated test assembly (ATA). ATA has several advantages over manual test assembly. First of all, the test specifications should be defined rigorously early on reducing the need to repeat some phases of the test development. More importantly, ATA is the only way to find optimal or near-optimal solutions starting from large item banks, for which manual assembly is not feasible due to the large number of possible combinations of items. As a consequence, ATA is fundamental to make measurements comparable while reducing operational costs.

The development of computer technologies enabled national test institutes to introduce the use of computers in educational testing. In 2018 the same INVALSI adopted computer based testing (CBT) for grades 8 and 10 instead of the traditional paper and pencil (P&P) testing. This technical report discusses the basics of ATA and the specific characteristics of the assembly procedures used for building the INVALSI Italian and mathematics standardized tests administered in the 2017/2018 school year to grade 8 and grade 10 students. In Section 1, the key theoretical issues of CTT and IRT, which represent the fundamentals of ATA, are briefly introduced. In Section 2, the main features of the optimization models used in ATA are presented. Finally, Section 3 describes the specific characteristics of the ATA procedures used to assemble the INVALSI tests.

Contents

Chapter 1

Test theories

In educational and psychological measurement, the process of test development follows specific steps (see, e.g. [3]) which are guided from strong methodological test theories: classical test theory (CTT) and item response theory (IRT). The statistical framework of test theories was introduced in [7]. The major focus of CTT is on test-level information, while IRT primarily focuses on the item-level information. Especially IRT provides a good framework for automated test assembly methods.

1.1 Classical test theory

The fundamental assumption of CTT is that the true score of a person on a measurement, which is a unobservable variable, is the expected value of the observed raw score, i.e. the expected number-correct score over an infinite number of independent test administrations [8, 7]. Given a person n, the observed score X_n is defined as the sum of the true component and a random error component, as follows

$$X_n = \tau_n + E_n, \tag{1.1}$$

where τ_n is the true score and E_n is the normally distributed error term with expected value equal to zero and constant variance. Measurement errors are assumed to be uncorrelated over repeated administrations.

Within CTT, the test development process is based on checking the test validity and the test reliability. In particular, the fundamental concept of reliability is concerned with the internal consistency of the test, i.e. the degree to which all item scores in a test correlate positively with one another. The most popular index used to assess test reliability is the Cronbach's α [2], which is defined as

$$\alpha = \frac{k}{1-k}\left(1 - \frac{\sum_{i=1}^{k}\sigma_i^2}{\sigma^2}\right), \tag{1.2}$$

where k is the test length, σ^2 the variance of the total test score and σ_i^2 the item variance. The closer α to 1, the higher the test reliability. Unfortunately this function is non-linear with respect to the items and hence is not actively used in the test assembly models, anyway it can be employed to assess the consistency of the assembly results.

Two other item properties play an important role in CTT: item difficulty and item discrimination. For dichotomously scored items, the difficulty is defined as the expected score given by a randomly selected examinee from the population of interest and is usually denoted by π_i or p-value p_i. Item discrimination is operationalized as the point biserial correlation between the item score and the observed test score and denoted as ρ_{it} for item i and test t.

The main limitations of CTT are the test-dependent score, the existence of a single standard error of measurement for the population, and the focus at test level. These drawbacks are overcome by using IRT.

1.2 Item response theory

An exhaustive introduction about IRT models can be found in [7, 4, 5]. Here, we are going to discuss IRT models with the only intention to supply their fundamental ideas and mathematical notation needed to understand the coming sections. Very briefly, an IRT model expresses the relation between the observable variables (the item responses) and the unobservable, latent ability through a probabilistic model. It is assumed that the performance of an examinee can be explained by means of the latent ability. The most popular IRT models are based on the unidimensionality assumption, i.e. the existence of a single latent ability. However, multidimensional IRT models have been developed as well. A second assumption is the local independence, which means that the item responses are statistically independent, conditional to the specification of the correct dimensionality (a single ability or a set of abilities).

A unidimensional IRT model for dichotomous items (e.g. correct-incorrect) expresses the probability of endorsing an item as a function of the underlying ability and a set of item parameters representing the item properties through a S-shaped curve called item characteristic curve (ICC). This function is non linear in the ability and it is monotone increasing because the idea behind these models is that the higher a person is located on the latent trait the higher is the probability she/he will give a correct answer. Different IRT models are characterized by the item response type, the number of item parameters, the latent ability structure, and the functional form.

We focus here on the Rasch model [10] which is used by INVALSI for the national standardized tests. The Rasch model, also known as the one-parameter logistic (1PL) model, has an ICC expressed by the following formula

$$P_i(\theta) = \frac{\exp(\theta - b_i)}{1 + \exp(\theta - b_i)} \tag{1.3}$$

where $P_i(\theta)$ is the probability of a correct answer to item i for an examinee of ability level $\theta \in (-\infty, \infty)$, and the parameter $b_i \in (-\infty, \infty)$ represents the difficulty of the item i. An example of ICCs for the Rasch model is shown in Figure 1.1. Note that the curves differ only by their location on the ability scale: the easiest item is C1 while the most difficult one is the C4. In fact, the Rasch model assumes that the difficulty parameter is the only item characteristic that influences the examinee's performance.

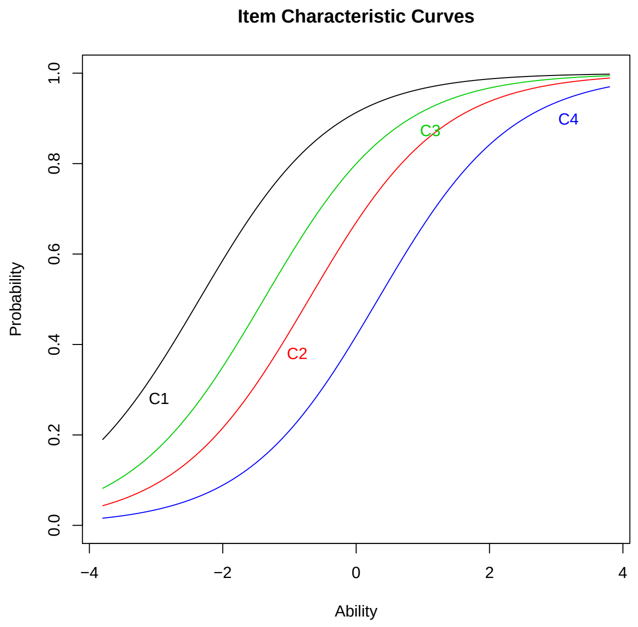


Figure 1.1: ICCs of 4 items C1-C4 with different difficulties according to the Rasch model.

IRT models allow the simultaneous estimation of the item parameters and the examinees' abilities. The *calibration* process usually involves the estimation of item parameters from pre-test response data. The *scoring* phase deals with the estimation of the ability scores of the candidates. Once the item parameters have been estimated, it is possible to understand how precise the test is at various ranges of the latent ability by using the test information function (TIF), which is defined as the sum of the Fisher information for all the items in the test. In fact, under the maximum likelihood (ML) scoring, the Fisher information is asymptotically equal to the inverse of the variance of the ML estimator as follows

$$I(\theta) = \frac{1}{\mathrm{Var}(\hat{\theta}|\theta)}. \tag{1.4}$$

The TIF has a very favorable property that is the additivity (and hence linearity) over the items of a test. Given a test with k items, the TIF is equal to

$$I(\theta) = \sum_{i=1}^{k} I_i(\theta), \tag{1.5}$$

where $I_i(\theta)$ is the item information function (IIF) for item i. Expressions for the IIFs can be easily derived within the framework of IRT. For example, for the Rasch model, the IIF of item i is equal to

$$I_i(\theta) = P_i(\theta)(1 - P_i(\theta)) = \frac{\exp^{(\theta - b_i)}}{[1 + \exp^{(\theta - b_i)}]^2}. \tag{1.6}$$

An example of IIFs for the Rasch model is shown in Figure 1.2. The items are maximally informative (information equal to 0.25) at the ability level corresponding to the difficulty parameter.

Figure 1.3 shows a test information function for a test with 10 items. The Fisher information function is a very important issue for test assembly. Tests can be assembled merely through the selection of appropriate items out of an item bank, one way to do so is to use mathematical programming techniques like 0-1 linear programming (LP) or mixed integer programming (MIP) models. Using these approaches the tests can be built by, for instance, maximizing the TIF at predefined θ points (MAXIMIN in this report), or matching it with known optimal values (MINIMAX) with linear restrictions on the values of items properties.

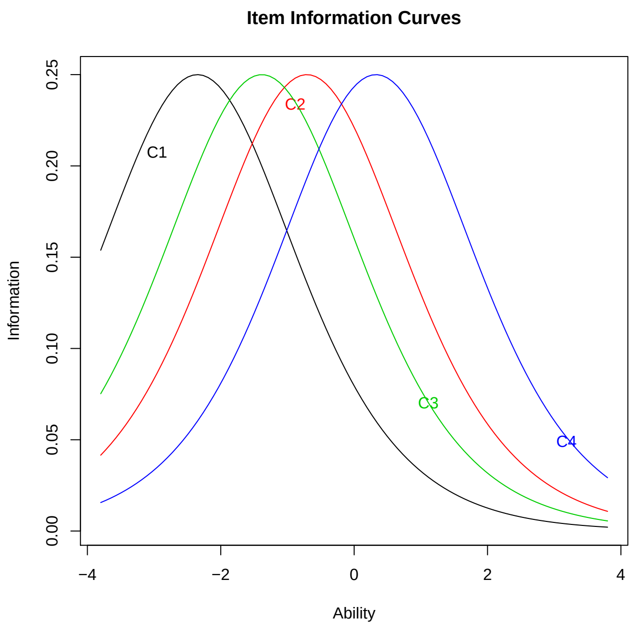


Figure 1.2: IIFs of 4 items C1-C4 according to the Rasch model.

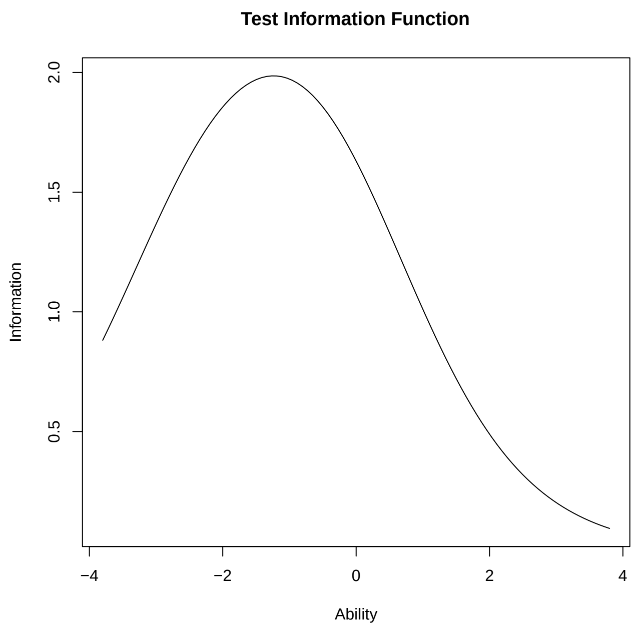


Figure 1.3: An example of test information function (TIF).

Chapter 2

Automated test assembly

In the late 1970s the transition from paper-and-pencil to computer based tests has began in the United States, increasing the efficiency and the accuracy of the assessment tools. First of all the administration of tests by computers streamlined the process of data collection and recording allowing to have scores immediately available and free of data entry errors. Secondly, skills and variables that couldn't be assessed or measured by paper-and-pencil tests like higher-level thinking skills, complex problem-solving and response times now are easily recorded and evaluated thanks to the computers. The growing number of credentialing exams for allowing the practice of a profession, admission tests for granting the access to the universities and standardized national tests for comparing abilities among different settings increased the importance of the use of test scores and hence the content and statistical features of the tests became crucial for the test validity and reliability.

Automated test assembly was born in this framework where fulfilling several specific requirements on the tests such as reducing the length of the test, maximizing the precision of the ability estimates, building tests with the same difficulty level and more over were needed. In practice by ATA models a test developer can impose any content and statistical criteria, from here called constraints, by specifying them in the form of linear (in)equalities. Also, a test developer could choose an objective function to serve as the goal for test assembly. Therefore the computer can find a set of test items that optimally meet these specifications. It is clear that test assembly is at the heart of the test development process but the automatically produced test is only a first draft of tests that could then be reviewed and re-examined by committees.

At the time when the test is assembled, input is required from three other processes: test specification, item construction and test data analysis. A good test needs good specifications, good items and good data.

2.1 The item bank

Once the calibration has been done, the items and their estimated and structural properties are stored in the *item bank* (or item pool). Afterwards, we can move on to the test assembly procedure in which the items will be selected depending on those distinctive characteristics.

Table 2.1 shows an example of the structure of an item bank with I items, where the items are displayed by row while in the columns we find the items features, from left to right: the identifier (*ID*), the IRT difficulty parameter (b) together with its standard error (b_{se}), CTT difficulty (*p-value*), content attributes (*TYPE*, *PROCESS*, *DOMAIN*), and relational attributes that specifies if the item belongs or not to a specific set (*FRIEND SET 1*, *FRIEND SET 2*, *ENEMY SET 1*, *ENEMY SET 2*).

Examinees can get different sets of items because, thanks to the calibration via IRT, the items are set on the same scale and the examinees' scores can be compared.

2.2 Types of assembly models

Given an item bank containing a sufficient number of calibrated items, is it possible to assemble one or more test forms which features can be similar or diverge. When we need to assemble only one test form we are speaking about *single tests assembly* models while if the tests are more than one the models are for *multiple tests assembly*, in particular if the obtained ones are similar under their psychometric characteristics they are called *parallel* [1]. In this work we will focus on the latter category of automated test assembly models.

To solve this type of problems in the last decades three modes of automated test assembly became prevalent: sequential, simultaneous single or multiple test assembly and adaptive tests. By the first two, test forms are entirely built before the administration of the test while in the adaptive framework each test form is assembled during the testing procedure.

Sequential test assembly

The straightforward technique for assembling single or multiple test forms is to populate the forms with items in a sequential way. This is being done by selecting items or groups of items and removing them from the pool, then the model is adapted to the new pool to try to fit the next form. In the case of assembling parallel forms, this method has two serious disadvantages. First, if the forms are assembled one after the other, the value of the objective function for the solution of the model tends to decrease due to the fact that the items

[1]Other details in Section 2.4.

i	*ID*	*b*	b_{se}	*ES*	*FORMAT*	*PROCESS*	*DOMAIN*	*ITEM SET 1*	*ITEM SET 2*	*ENEMY SET 1*	*ENEMY SET 2*	*ENEMY SET 3*
1	M02KL	-2.0	1.02	0.6	Matching	Problem solving	Numbers	1	0	1	0	0
2	M35KL	-1.5	0.12	0.2	Multiple-choice	Knowing	Space and figures	1	0	1	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$I-2$	M03PF	1.2	0.05	0.4	Open-ended	Knowing	Numbers	0	1	1	0	1
$I-1$	M08PF	0.06	0.98	0.35	Multiple-choice	Knowing	Numbers	0	1	1	0	1
I	M10ML	0.75	0.4	0.12	Multiple-choice	Knowing	Space and figures	0	0	1		

Table 2.1: Example of item bank.

with the best values will be selected first. As a consequence, the forms cannot be parallel. A second disadvantage of this approach is the possible infeasibility of later models in the sequence. On the other hand by this type of model a large range of functions can be optimized e.g. the linear structure of the problem can be relaxed. Most of these problems nowadays are solved using ad hoc greedy and heuristic techniques.

Simultaneous test assembly

In the sequential approach an incremental number of different models must be optimized in order to obtained multiple forms causing the disadvantages cited in the last paragraph. Those can be overcome by applying simultaneous test assembly models in which the solution, and hence the test forms, is obtained by solving only one model. They are usually represented by 0-1 integer programming problems and solved by linear programming techniques. The model can be reformulated as a mathematical optimization problem using decision variables. Decision variables are variables defined such that the solution of the optimization problem (i.e., the set of values for which the objective function is optimal and all constraints are satisfied) identifies the best decision that can be made. The decision variables $\{x\}_{it}$ are binary (0-1) since they represent the inclusion (1) or exclusion (0) of the item i in the test t. In the last decades, the algorithms needed to solve such optimization problems were improved, and nowadays powerful implementations of them are available as commercial or open-source software.

Adaptive tests

In the adaptive approach one item is optimally picked from the pool at a time. The person's ability estimate is updated during the test, and each next item is chosen to be maximally informative at the last ability update. The test is then tailored to the candidate. Because the ability estimates converge to the person's true ability level, item selection improves during the test and the ideal of a test with maximum information at the person's true ability level is approached. In the early 1990s computerized adaptive testing (CAT) was implemented in large-scale testing programs, and nowadays large numbers of subjects are tested worldwide using this type of test assembly. One of the most important benefits of this method is that it's more efficient, i.e. it is possible to get more precise ability estimates with less items than the previous standard approaches, but at the cost that it's quite impossible to ensure the congruence between tests since all the test takers will get a different set of items from the pool.

The description of different approaches to assemble tests is not finished here. Because of its advantages the approach dealt with in this report is the simultaneous tests assembly mode. In Section 2.3 and Section 2.4 we introduce the reader to the standard test assembly models used to build first, a single test

1.	Average p-value of the test between .40 and .60
2.	Number of items on applications equal to 24
3.	Reliability of the test should be as high as possible
4.	Items 73 and 100 never in the same test
5.	Test information function as close as possible to the target
6.	Items 33, 45 and 12 must be in the same test

Table 2.2: Example of desiderata.

form and secondly, as a generalization, multiple test forms.

2.3 Single test assembly

As already discussed, a new testing program starts with formulating the set of specifications for the test to be met, that here we call *desiderata*. Sometimes they are verbally expressed as a set of learning objectives or a list of dos and don'ts for the test developer but they can also be well-structured in tables specifying how many items should have a certain content or even better the distribution of items according to their specific characteristics.

Once the desiderata have been collected, they must be translated into a standardized language used in test assembly problems. The standard form in these problems is an *objective function* to be optimized subject to a number of *constraints*, where the latter define a possibly feasible set of tests for a given item bank, and the former expresses our preferences for the tests in this feasible set. If the specifications have been formulated in a simple, concise but complete way it is possible to determine whether they are objectives or constraints. These requirements are really crucial for a correct translation of the desiderata into the standard language for test assembly problems: disambiguations and possible complications may arise in test design if these principles are not satisfied. An example of verbal test specifications is described in the Table 2.2, which is partially extracted from [12].

The specifications in Table 2.2 may represent either objectives or constraints, for example points 1,2,4 are constraints while 3 and 5 are objectives. It is clear now that the objectives involve maximize or minimize some attributes, such as minimize the gap between the test information function at some θ point to its target (5) or maximise the reliability of the test (3) and on the other hand constraints impose a bound on an attribute of the test or of the items, such as limiting the difficulty of the test (1), fixing the number of items having certain characteristics (2) or considering enemy sets (4) or also item (friend) sets (6). An exhaustive classification of specifications together with examples of not well expressed desiderata can be found at pages 34-39 of [12].

optimize	*Objective function*
subject to	
	Constraint 1
	Constraint 2
	⋮
	Constraint N

Table 2.3: Standard form of a test assembly problem.

2.3.1 Standard form of a test assembly problem

If a set of desiderata is well specified they can always be represented in the standard form of Table 2.3.

Only one objective can be optimized at a time; if we have more than one function to optimize some tricks can be applied to transform the objectives into constraints. On the other hand, there is no upper limit for the number of constraints, provided our solver is able to handle the problem. If at least one combination of items that meets all the constraints does exist then the set of these combinations is called *feasible set*; if this set is empty we say that the model in *infeasible*. The subset of tests in the feasible set that optimize the objective function is called *optimal feasible solution*.

2.3.2 Decision variables

Modeling a test assembly problem does not imply only defining objectives and constraints, but we need also the so called *decision variables* which represent the possible combination of items that compose the test form in a mathematical formulation. If we need to assemble a single test from a pool of I items, we can choose as decision variables I binary variables in the form:

$$x_i = \begin{cases} 1 & \text{if item } i \text{ is in the test,} \\ 0 & \text{otherwise.} \end{cases}$$

We can also write the x_i in an algebraic way using the vector x of length I. Therefore we have 2^I possible combinations of values the vector x can take. These combinations decrease in number if we add constraints to the model. In the following section we will use also non binary variables like integer o continuous, which are often useful to reduce the number of objectives in the problem. Once the decision variables have been identified, the process of modeling a test assembly problem goes through the steps of modeling the constraints and objectives and solving the model looking for an optimal solution. In the following section we will present the basic formulations for some common types of constraints and objectives. The last step consists of solving the model by using a computer program which implements some mathematical or numerical algorithms. In our case the software used is the ATA software supplied by CITO (NL), discussed in Section 3.

2.3.3 The model for assembling a single test

The models presented in this section are based mainly on item response theory attributes and we will make an almost exhaustive list of categories of constraints that can be added to a model for single test assembly together with possible objective functions.

The standard model for the assembly of a single test with a generic quantitative objective from a pool of I items indexed by $i = 1, \dots, I$ is

$$\text{optimize} \sum_{i=1}^{I} q_i x_i \quad \text{(objective)} \tag{2.1a}$$

subject to

$$n_c^{\min} \leq \sum_{i \in V_c} x_i \leq n_c^{\max}, \ \forall c \quad \text{(categorical constraints)} \tag{2.1b}$$

$$b_q^{\min} \leq \sum_{i=1}^{I} q_i x_i \leq b_q^{\max}, \quad \text{(quantitative constraints)} \tag{2.1c}$$

$$n^{\min} \leq \sum_{i=1}^{I} x_i \leq n^{\max}, \quad \text{(test length)} \tag{2.1d}$$

$$\sum_{i \in V_e} x_i \leq 1, \quad \forall e \quad \text{(enemy sets)} \tag{2.1e}$$

Then, definition of variables

$$x_i \in \{0, 1\}, \ \forall i \quad \text{(decision variables)}$$

where the sums in the constraints may be bounded only on one side, in which case we only need one of the two inequalities.

For a qualitative variable we use V_c to represent the set of indices of the variable of the items in subset c. Allowing the test to have a number of items with certain qualitative attribute represented by V_c between $n_c^{\min}$ and $n_c^{\max}$ means adding a *categorical constraint* to the model. An example of categorical constraint is “the test must contain at least 10 items in problem solving”, where the set of items in the pool that has the attribute “problem solving” is represented by $V_{\text{problem solving}}$ and $n^{\max}_{\text{problem solving}} = 10$.

On the other hand, if we want to bound the value of a quantitative variable (i.e. that can take numerical values) addressed by the symbol q between the values $b_q^{\min}$ and $b_q^{\max}$ we are constraining the model adding a *quantitative constraint* where q_i is the value that the item i takes for that variable. Suppose we want to assemble a test with the following criterion "the maximum of words must be 300", q_i will serve as the number of words displayed by the item i and $b_q^{\max} = 300$.

The interpretation of the *test length* constraint is straightforward as it is a special case of a quantitative constraint where $q_i = 1$ for each $i = 1, \dots, I$.

If there are *enemy sets* in our pool then the items in one of these sets, called V_e, cannot be picked together, e.g. the desideratum "items 23 and 46 cannot be in the same test" means that items 23 and 46 are enemies and both are in the enemy set V_e, therefore the sum of the corresponding decision variables x_{23} and x_{46} cannot be more than 1.

Item sets

Tests with sets of items organized around common stimuli (known as item sets or friend sets) are popular because of the efficiency of their format. By combining more than one item with the same stimulus, we are able to ask questions using more complex stimuli, such as reading passages, descriptions of cases, or problems with data in a set of tables or graphs, without having to sacrifice too many items for the test to meet the time limit. But the presence of such sets in the item pool complicates the process of assembling the test. In particular, if the items are grouped in S item sets or stimuli indexed by s, there are three ways to deal with this problem (see [12]).

- The *power set method* that consists in summarizing all the quantitative and qualitative attributes by summing or averaging them taking as groups the item sets; if we only need to constraint the test at the set/stimuli level the problem decreases in size since you don't have I decision variables anymore but just S. This method is not efficient if you have variables that cannot be summarized, such as standard errors and if you want to keep some constraints at item level as you need to consider again the original decision variables x_i increasing the size of the model.
- The *pivot-item method* in which each set is represented by a pivot decision variable $x_{i_s^*}$ arbitrarily chosen. Therefore, we can use the variable for a pivot item as a carrier of the attributes of both its stimulus and item set, and we can drop the stimulus variables in the model.
- The *two-stage method* that is a sequential approach with two phases, in the first stimuli are selected and secondly the model choses items with those stimuli.

Given the earlier warnings about sequential approaches we do not suggest the two-stage method and since all the variables in our item bank are summarizable (e.g. the item information function is additive so it is possible to sum it for all the items in the stimulus) we prefer to adopt the power set method which helps to decrease the size of the problem.

The power-set formulation needs a phase of preprocessing of the item bank that, in our work is automatically performed by `ATA`. Instead, for the formal representation of the model, we refer to [12], Section 7.2.

Single test MINIMAX

Once we defined all the constraints and checked that the model is feasible we need to choose an objective to optimize. A first option is to choose absolute targets for the TIF, targets are the values that a goal TIF assumes on a fixed number of θ points along the θ scale, these values must be chosen by test specialists who knows how much precision is required to estimate the abilities of the students at each ability level. This is the reason why absolute targets are used almost exclusively when tests are assembled to be parallel with respect to a known reference test. Formalizing this requirements in the standard form of test assembly models will produce a multi-objective test assembly problem that must be reformulated using the MINIMAX approach explained below.

In particular with the following addition to the model (2.1) we ask that the TIF of the resulting test approximates with minimum negative and positive deviations (i.e. with the highest precision) the chosen goal TIF in a finite set of points V on the θ scale, which we denote as T_k with $k \in V$:

$$\text{minimize} \quad y \quad \text{(objective)} \tag{2.2a}$$

subject to

$$\sum_{i=1}^{I} I_i(\theta_k)x_i \leq T_k + y, \forall k \in V \tag{2.2b}$$

$$\sum_{i=1}^{I} I_i(\theta_k)x_i \geq T_k - y, \forall k \in V \tag{2.2c}$$

$$y \geq 0,$$

More θ points we choose more the TIF of the assembled test will meet the auspicable one, usually 3-5 points are enough to have a good approximation.

Single test MAXIMIN

If a reference test or absolute targets are not available an alternative approach is trying to achieve the best predictive validity for the test, that is maximising the TIF in some chosen theta points. This goal can be met not only setting the location of the peaks of the TIF but also defining its shape in the entire θ scale, which is to say imposing relative targets.

So, denoting with R_k the relative targets for each θ_k with k in the chosen set V of ability points in which we want to control the shape of the TIF, we must fix one of the relative targets to a value (e.g. $R_1 = 1$) and all the other target values must be adjusted correspondingly trying to reproduce the wanted shape.

Like the absolute target model, this method leads to have more than one objective function and also in this case it is possible to rely on a simplifying approach called maximin. The model can be formalized as following:

$$\text{maximise} \quad y \quad \text{(objective)} \tag{2.3a}$$

subject to

$$\sum_{i=1}^{I} I_i(\theta_k)x_i \geq yR_k, \; \forall t, k \in V \tag{2.3b}$$

$$y \geq 0,$$

Also here the practice suggests that the minimum number of θ points in which maximize the TIF must be 3 or 5.

2.4 Multiple simultaneous test assembly

In order to discourage the phenomenon of cheating it is necessary to administer different items to the test takers. This can be achieved building more than one test form that contains different items preserving some overall mutual psychometric features, such as the same test difficulty or same content structure such as same percentage of items of a certain stimulus. These tests are called *parallel* (or interchangeable) and the procedure aimed to perform this task is called "multiple test assembly". In particular, test forms are defined to be *weakly parallel* if their information functions are identical [11]. On the other hand, test forms are *strongly parallel* if they have the same test length and exactly the same test characteristic function [6].

As a consequence, we typically have problems with more objectives than those presented in Section 2.3 for single test assembly. For example, if we assemble T tests and each test has to meet a target for its information function at K ability points (the same points for each test t, with $t = 1, \ldots, T$), the problem has at least $T \times K$ objectives. However, these large multi-objective test assembly problems can be solved using a direct generalization of the approaches for single test assembly. In the following, we will present a general model for simultaneous assembly of a set of tests (that is, as a solution to a single model) that produces an optimal solution. This type of assembly requires a reorganization of the problem using a different version of the decision variables. These variables have double indices[2], one for the items in the pool and the other for the test forms. For the current problem, the variables become

$$x_{it} = \begin{cases} 1, & \text{if item } i \text{ is assigned to test } t \\ 0, & \text{otherwise} \end{cases}$$

for all i and t. As before, we need to complement these variables with a set of constraints that keep their values consistent. The adapted single test assembly model is presented here followed by constraints that arise only in the case of multiple test assembly.

2.4.1 The model for assembling multiple tests

Using the above decision variables, any model for a single test can be reformulated as a model for multiple tests. To illustrate this statement, we reformulate the standard model for a single test in (2.1a)-(2.1e). The model is

$$\text{optimize} \sum_{t=1}^{T}\sum_{i=1}^{I} q_{it}x_{it} \quad \text{(objective)} \tag{2.4a}$$

subject to

$$n_{ct}^{\min} \leq \sum_{i \in V_c} x_{it} \leq n_{ct}^{\max}, \; \forall c, t \quad \text{(categorical constraints)} \tag{2.4b}$$

$$b_{qt}^{\min} \leq \sum_{i=1}^{I} q_i x_{it} \leq b_{qt}^{\max}, \quad \forall t \quad \text{(quantitative constraints)} \tag{2.4c}$$

$$n_{t}^{\min} \leq \sum_{i=1}^{I} x_{it} \leq n_{t}^{\max}, \quad \forall t \quad \text{(test length)} \tag{2.4d}$$

$$\sum_{i \in V_e} x_{it} \leq 1, \quad \forall t, e \quad \text{(enemy sets)} \tag{2.4e}$$

Then, definition of variables

$$x_{it} \in \{0, 1\}, \; \forall i, t \quad \text{(decision variables)}$$

The changes in (2.4) relative to the original model in (2.1a)-(2.1e) are:

1. the replacement of the variables x_i by x_{it};

[2]We use a matrix representation only for a better visual idea, actually they are still vectors but of bigger size.

2. the extension of the objective function to the case of T tests;
3. the indexing of the bounds in the constraints by t to enable to assemble tests with different specifications.

The generalization of the objective function in (2.4a) is simple and consists of taking an (unweighted) sum over the tests.

Item sets

The power-set method presented in 2.3.3 can be easily generalized to the case of multiple tests assembly, for the sake of brevity we prefer to skip the details and still refer to the van der Linden book [12], section 7.2.

Item use

If we want to control the minimum or the maximum number, respectively $n_i^{\min}$ and $n_i^{\max}$, of tests in which an item i can be assigned, we have to consider the following constraints:

$$n_i^{\min} \leq \sum_{t=1}^{T} x_{it} \leq n_i^{\max}, \ \forall i, \quad \text{(item use)} \tag{2.4f}$$

Test Overlap

Since we are creating more than one test form we are not only concerning the properties of each form singularly but also the relationship between them, one of these is the *test overlap*, that is the number of items two forms share. Sometimes it is not important to control for this specification especially if an item use constraint has been fixed. However, if we do not want any overlap between all the pairs, as an example, we have to add these constraints to (2.4)

$$\sum_{t=1}^{T} x_{it} \leq 1, \ \forall i, \quad \text{(no overlap)} \tag{2.4g}$$

The latter is not enough if the test developer wants to keep the same but not null level of overlap between all the possible pairs of tests. In this case, the model has to be modified not only adding new constraints but also a substantial number of variables (luckily binary). In particular controlling for test overlap means adding quadratic constraints of the form:

$$o_{tt'}^{\min} \leq \sum_{i=1}^{I} x_{it}x_{it'} \leq o_{tt'}^{\max}, \ \forall t \neq t', \quad \text{(fixed overlap NON LINEAR)}$$

Those types of constraints must be linearized in order to use the standard LP solvers, so they must be replaced by the following variables

$$z_{itt'} = \begin{cases} 1 & \text{if item } i \text{ is both in test } t \text{ and test } t' \text{ (i.e. } x_{it} = x_{it'} = 1) \\ 0 & \text{otherwise.} \end{cases}$$

This modification increases the size of the model of $I * \binom{T}{2}$ binary variables. Together with the new variables the constraints must be replaced by

$$o_{tt'}^{\min} \leq \sum_{i=1}^{I} z_{itt'} \leq o_{tt'}^{\max}, \ \forall t \neq t', \quad \text{(fixed overlap LINEAR)} \tag{2.4h}$$

Integrality constraints:

$$z_{iit'} \geq x_{it} + x_{it'} - 1 \ \forall i, t \neq t' \tag{2.4i}$$
$$2z_{iit'} \leq x_{it} + x_{it'} \quad \forall i, t \neq t' \tag{2.4j}$$
$$z_{itt'} \in \{0, 1\}, \quad \forall i, t \neq t'$$

The last two constraints are necessary to keep the values of $z_{itt'}$ consistent with their definitions.

Multiple MINIMAX

An example of an objective for multiple test assembly is the generalization of the MINIMAX principle presented in 2.3.3, formally

$$\text{minimize} \quad y \quad \text{(objective)} \tag{2.6a}$$

subject to

$$\sum_{i=1}^{I} I_i(\theta_{kt}) x_{it} \leq T_{kt} + w_t y, \ \forall t, k \in V_t \tag{2.6b}$$
$$\sum_{i=1}^{I} I_i(\theta_{kt}) x_{it} \geq T_{kt} - w_t y, \ \forall t, k \in V_t \tag{2.6c}$$
$$y \geq 0$$

where the target values T_{kt} are indexed by t to allow us to set different targets for different tests. In addition, as different set of values V_t for each test can be given, we can specify the target values at a different set of θ values for each test. Finally, we have added weights w_t to have the option of weighting deviations from target values differently for several tests. If our goal is to assemble parallel tests, the targets and weights will be equal for all t.

Multiple MAXIMIN

The approach used in section 2.3.3 may be generalized to the case of multiple test assembly with this mix of objective and constraints

$$\text{maximise} \quad y \quad \text{(objective)} \tag{2.7a}$$

subject to

$$\sum_{i=1}^{I} I_i(\theta_{kt}) x_{it} \geq y R_{kt}, \ \forall t, k \in V_t \tag{2.7b}$$

$$y \geq 0,$$

where the R_{kt} may be chosen equal between tests, i.e. $R_{kt} = R_{k't'}$ with $t \neq t'$ and $\forall k = k'$ ensuring the parallelism.

Chapter 3

INVALSI automated test assembly

Starting from the school year 2017/2018, the Italian (ITA) and mathematics (MAT) INVALSI tests have been administered via computer-based testing (CBT) for students in grade 8 and 10[1]. The CBT administration is based on automatic testing assembly (ATA). The main aim of the assembly procedure is to provide, for each grade and test, a set of parallel test forms via multiple simultaneous assembly. The assembly work starts from the item banks provided by INVALSI for each grade and test.

3.1 The item banks

The ITA item banks of grade 8 and 10 consist of 367 and 295 items, respectively. The pre-tested items in each bank have been calibrated according to the Rasch model. Each ITA bank contains the following variables for each item:

- Id;
- Domain (reading comprehension, grammar, and lexicon[2]);
- Type (e.g. multiple-choice, short-answer open-ended);
- Text type (e.g. narrative, only for reading comprehension items);
- Text length (short, medium or long depending on the number of words, only for reading comprehension items);

[1]The English test for students in grade 8 is not discussed in the present report as the test assembly followed a different procedure.

[2]Only for grade 10 items.

- Friend set (it defines a group of items with a common stimulus such as a text);
- Enemy set (it defines a group of items with a relation of exclusion, i.e. which cannot be selected for the same test form[3]);
- Lexicon task (A, B, C, only for grade 10 lexical knowledge items);
- Reading subdomain (not considered in ATA);
- Grammar subdomain (not considered in ATA);
- Rasch difficulty parameter estimate (and its standard error);
- CTT difficulty (p-value).

The MAT item banks of grade 8 and 10 consist of 237 and 277 items, respectively. Analogously to the ITA items, the Rasch model was used for calibration. Each MAT bank contains the following variables for each item:

- Id;
- Domain (numbers, space and figures, relationships and functions, data and forecasts);
- Process (knowing, problem solving, arguing and probing);
- Type(e.g. multiple-choice, short-answer open-ended);
- Enemy set (it defines a group of items with a relation of exclusion, i.e. which cannot be selected for the same test form);
- Rasch difficulty parameter estimate (and its standard error);
- CTT difficulty (p-value).

As can be seen from the description of the variables, the information stored in the banks is at item level or at item-set level. In addition, the target population parameters (mean and standard deviation) are available.

3.2 Desiderata

The desiderata represent the set of specifications required for the test forms. They can have a hierarchy, in the sense that some of them are fundamental (primary) while others have a lower importance (secondary). Establishing different levels of desiderata is very useful to deal with infeasibility problems, as

[3]The ITA grade 10 bank does not contain enemy sets.

the objectives and constrains may not be completely satisfied by the items in the bank. To prevent this drawback, the item pool should be built according to the desiderata required in the automated test assembly.

The main desiderata for each grade and test is that at least 12 different test forms are produced. Item overlap is admitted among the test forms to allow all administered items to be on the same scale. Each test form should contain items belonging to all the specified ITA or MAT domains. Each test form contains different item types, with approximately the same distribution among the test forms. Each test form should contain items with different difficulty with approximately the same distribution of difficulty among the test forms. Friend and enemy sets (when available) should be taken into consideration. The ATA procedure allows that all the test forms have the same psychometric properties.

For grade 8 ITA, the following additional desiderata are specified for each test form to be met:

- total test length: about 45-50 items;
- item overlap proposal: about 10;
- number of reading comprehension units: 4;
- number of grammar and lexicon items: about 14;
- number of narrative items: at least 7;
- text length for reading comprehension items: at most one short and one long texts;
- different task types for the lexicon items;
- expected test score about 50%.

Analogously, the following additional desiderata are specified for each test form of grade 10 ITA:

- total test length: about 35-37 items;
- item overlap proposal: about 8;
- number of reading comprehension units: 4;
- number of grammar items: about 5;
- number of narrative items: at least 6;
- text length for reading comprehension items: at most one short and one long texts;
- expected test score about 50%.

With respect to the MAT tests for grade 8 students, the specific desiderata are:

- total test length: about 38 items;
- item overlap proposal: about 5;
- number of items by domain: 10 numbers, 9 space and figures, 10 relationships and functions, 9 data and forecasts;
- number of items by process: 20 knowing, 14 problem solving, 4 arguing and probing;
- number of open-ended items: at most the 60%;
- expected test score about 50%.

Last, the additional desiderata for the MAT tests of grade 10 are:

- total test length: about 35-40 items;
- item overlap proposal: about 6-7;
- same proportion of items by domain;
- number of items by process: about 40% knowing, about 30-40% problem solving, about 20-30% arguing and probing;
- number of open-ended items: at least one;
- number of multiple-choice items: at most 18;
- expected test score about 50%.

For all the tests, the total number of items could be reduced to solve infeasibility problems. Also, where substantial differences in the composition of the test forms by item type were found, proper adjustments were adopted.

3.3 The ATA software

The test assembly problems discussed in this chapter have been solved using the `ATA` software provided by Cito (NL) and developed by Angela Verschoor. `ATA` is a tool that can be used in the test assembly step of the test development process. It assumes that during pretest and analysis data are gathered either according to an IRT model or according to CTT. Yet, as it is not always trivial to collect data in the most efficient way, and not all tests are necessarily based on IRT or CTT, `ATA` offers optimization models for those approaches as well. All the supported models have some elements in common: items are classified according to their content features, such as format, domain or belonging to item sets or enemy sets and also relationships between test forms can be specified such as tests overlap or item use. Thus, the item bank used must contain all relevant data for test assembly: item meta data and test specifications.

The test assembly models that are supported by `ATA` can be divided into three groups: one group based on IRT, one based on CTT and a group not based on any psychometric model at all. In the following, we describe the main test assembly models of `ATA` and the practical steps for using the software.

3.3.1 `ATA` models

Calibration design

The purpose of a calibration design is the parameter estimation, and therefore the most obvious choice for objective function is the minimization of estimation errors.

The direct way of evaluating estimation errors is through the Fisher information function. It shows the extent to which an observation reduces the prior uncertainty regarding the parameter [9]. If ξ is a vector of parameters, and $L(\xi; y)$ denotes the likelihood function associated with $Y = y$, Fisher information takes the form of a matrix I with elements

$$I_{ij} = -E\left[\frac{\partial}{\partial \xi_i} \log L(\xi_i; y) \frac{\partial}{\partial \xi_j} \log L(\xi_j; y)\right] \tag{3.1}$$

where ξ_i and ξ_j are individual parameters in ξ. Equation (3.1) is equal to the inverse of the asymptotic variance-covariance matrix of the ML-estimator of ξ. Minimizing $\text{var}(\hat{\xi} - \xi)$, or minimizing the determinant of the variance-covariance matrix is referred to as the D-optimality criterion ([1]). Thus, asymptotically this can be achieved by maximizing the determinant of the Fisher information matrix. Inspection of I, however, learns that the lowest estimation errors will be achieved by an infinite number of observations. In practical situations the maximum number of observations is limited, but usually not exactly known until the actual administration of the pretests. Therefore, the objective function should reflect Fisher information per participating student.

Fisher information, however, can only be determined if all item parameters are known, i.e. after the pretests have been administered and during analysis. Because item parameters are unknown they are all assumed to be equal to zero, and also all person parameters are assumed to be equal, but not necessarily equal to zero. In this case element I_{ii} is proportional to $u_i = \sum_t x_{it}$, the *incidence rate* of item i. Off-diagonal element I_{ij} is proportional to $u_{ij} = \sum_t x_{it} x_{jt}$, the incidence rate of item pair (i, j) in common test forms. The rows of I are known to sum to zero, therefore we get

$$
\begin{aligned}
I_{ii} &\propto u_i \\
I_{ij} &\propto -\frac{u_i u_{ij}}{\sum_{i \neq j} u_{ij}}.
\end{aligned}
$$

In order to compare different designs, I must be invariant for the number of observations per item. This can be achieved by setting diagonal element I_{ii} to the incidence rate u_i divided by the average incidence rate $\overline{u} = \frac{\sum_{i,t} x_{it}}{M}$. Thus, we obtain

$$
\begin{aligned}
I_{ii} &= u_i / \overline{u} \\
I_{ij} &= -\frac{u_i u_{ij}}{\overline{u} \sum_{i \neq j} u_{ij}}
\end{aligned}
$$

and the proposed optimization problem is

$$
\text{maximize} \quad \det I \tag{3.2}
$$

subject to restrictions in (2.4).

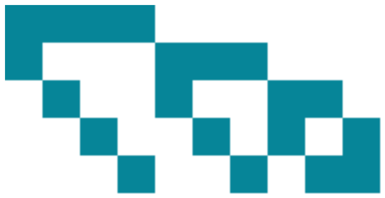
Note that evaluating (3.2) tends to be rather time consuming. Therefore, let us consider some commonly used solutions and their rationales first in order to formulate an alternative objective function that circumvent the disadvantages of achieving strict D-optimality.

Several designs are known to be D-optimal, in case we allow ourselves to disregard many of the restrictions imposed on a problem. Construction of one of these designs is straightforward from a theoretical point of view: If test forms of length L are to be assembled from an item pool of size M, then all combinations of L items out of the available M should be administered to an equal number of test takers. This design is constructed in such a way that all item covariances are observed evenly. It is obvious that the number of test forms soon outgrows the logistical complexity that one can handle. There is, however, a simplification that usually yields a very good approximation to D-optimality: combine all items in blocks of size $\frac{L}{u}$, and combine all sets of u different item blocks into test forms. With $\frac{Mu}{L}$ blocks in the item pool, this will result in a total of $\binom{\frac{Mu}{L}}{u}$ different test forms. This design is referred to as the balanced block (BB) design, whereby often in practical situations $u = 2$ is chosen. In that case $\frac{2(M^2-M)}{L^2}$ test forms are needed. It is also obvious that a BB design is only possible for certain combinations of M, L and u, although this can be overcome by using blocks of slightly different sizes. Despite the fact that the number of test forms is largely reduced compared to a true D-optimal balanced design, the BB design still needs a very large number of test forms. A second complication for the BB design forms the existence of enemy sets. This will prevent direct observation of certain item covariances and hence the existence of a BB design. Therefore, application of a BB design will not always be a viable option.

Close observation of the principles underlying the BB design leads to an alternative design principle. In a BB design, the M items in the pool are divided into T blocks with length $\frac{M}{T} = \frac{L}{u}$. Every set of u blocks is administered to a sample population of in total K students. As there are $\binom{T}{u}$ combinations of blocks, the K students will be divided into $\binom{T}{u}$ groups of equal size. The design matrix D with the total number of item responses per item block i and per group of students j is given by

$$D_{ij} = \begin{cases} 0, & \text{if group } j \text{ does not take item block } i \\ \frac{M}{T} * \frac{K}{\binom{T}{u}} = \frac{MKu!(T-u)!}{TT!}, & \text{else.} \end{cases}$$

Now, consider the transpose D^T, effectively exchanging the role of items and students in the design.



Now, divide the K students into T groups and the M items into $\binom{T}{u}$ blocks, and all pairs of students are observed in D^T, through item blocks of size $\frac{M}{\binom{T}{u}}$. In the BB design, u blocks are combined into a test form, that is, each student takes u blocks. In the transpose, each block is taken by u groups of students. In other words, the incidence rate is u. At the same time, a block with incidence rate u contributes to $\binom{u}{2}$ different overlaps between pairs of test forms. With $\binom{T}{u}$ blocks in the design, the sum of overlaps between all pairs of test forms is $\binom{T}{u}\frac{M}{\binom{T}{u}}\binom{u}{2} = \frac{Mu(u-1)}{2}$. At the same time, the overlap between a pair of test forms is established by $\binom{T-2}{u-2}$ blocks. Therefore, all overlaps have size $\binom{T-2}{u-2}\frac{M}{\binom{T}{u}} = \frac{Mu(u-1)}{T(T-1)}$.

It is clear that not in all circumstances the transpose of a BB design exists, but a few favorable properties are clear:

- The incidence rate of all items should be equal. This can only be reached when $Mu = TL$. If this is not the case, the incidence rate u_i of item i should be as close as possible to the average incidence rate $\overline{u} = \frac{TL}{M}$. Therefore, incidence rates will be restricted in order to enforce a homogeneous item use:

$$\lfloor \overline{u} \rfloor \leqslant \sum_t x_{it} \leqslant \lceil \overline{u} \rceil \quad \forall i. \tag{3.3}$$

- The overlaps of all pairs of test forms should be as close as possible to the average overlap $\overline{v} = \frac{Mu(u-1)}{T(T-1)}$. In first instance the overlaps may be restricted similar to the item use constraints in (3.3):

$$\lfloor \overline{v} \rfloor \leqslant \sum_i x_{is} x_{it} \leqslant \lceil \overline{v} \rceil \quad \forall s, t. \tag{3.4}$$

- The use of enemy sets and friend sets, and extensive use of content constraints may lead to infeasibility. The best option to overcome this is to transform the constraints into the objective function:

$$\text{minimize} \quad \sum_{s,t} \delta^+_{st} + \delta^-_{st} \tag{3.5a}$$

  subject to

$$\sum_i x_{is} x_{it} + \delta^-_{st} \geqslant \lfloor \overline{v} \rfloor \quad \forall s, t \tag{3.5b}$$

$$\sum_i x_{is} x_{it} - \delta^+_{st} \leqslant \lceil \overline{v} \rceil \quad \forall s, t. \tag{3.5c}$$

Thus, Calibration Design models optimize the overlap between test forms according to the objective function in (3.5), subject to constraints in (3.5b), (3.5c), (3.3), (3.4) and (2.4). Note that it may be possible that constraints in (2.4f) and (2.4h)-(2.4j) are redundant.

Expected score

An important prerequisite for the assembly of parallel test forms is the requirement that the test forms should be equally difficult. This is what the Expected Score model tries to establish, and it forms the basis for several other models. Let p_i be the expected score of item i, then $\sum_i p_i x_{it}$ is the expected score of test form t. The maximum of the deviations between the expected score of test t and a desired range is minimized through

$$\text{minimize} \quad \delta \tag{3.6a}$$

subject to

$$\sum_i p_i x_{it} + \delta \geqslant P^\ell_t \quad \forall t \tag{3.6b}$$

$$\sum_i p_i x_{it} - \delta \leqslant P^u_t \quad \forall t, \tag{3.6c}$$

where P^ℓ_t and P^u_t form the lower and upper borders of the desired difficulty range for test form t.

The expected item scores can be determined in various ways. One of the usual methods is data collection and analysis with CTT. A second method of deriving expected scores is through IRT, and in case that no empirical data are available, expert judgments might also be used.

Calibration design / expected score

Close inspection of the Calibration Design model and the solutions it generates shows that the optimal solution is generally not unique: items can be permuted in many ways without violating any of the constraints. This property is utilized by the Calibration Design / Expected Score model. When expected item scores are available, but when data still have to be collected for calibration purposes, the expected item scores can be used to ensure that all test forms are equally difficult. The model is optimized in two phases: in the first phase, an optimal design will be generated without taking any expected test scores in mind. In the second phase, the design will be fixed and items will be permuted in such a way that no constraint will be violated and the deviations from the desired difficulty range is minimized.

Expected score / maximum information

Using classical indices in automated test assembly has a serious shortcoming: generalization of the indices from the context in which they were measured into the new context of the test forms to be assembled is generally not possible without additional assumptions that cannot be evaluated within CTT itself. Therefore, using IRT will generally be preferred as checking whether model assumptions are violated is standard routine in IRT.

The maximum likelihood estimator $\widehat{\vartheta}$ is asymptotically normally distributed with mean equal to ϑ and variance equal to the reciprocal of the Fisher information function, defined as

$$I(\vartheta) = -E\left[\frac{d^2}{d\vartheta^2}\log L(\vartheta; x)\right].$$

Thus, in order to minimize measurement error the information function must be maximized. This maximization does not take place on the continuous function, but on a few strategically selected ability points, for example a cut-off ability.

The expected score / maximum information model is the simplest optimization model based on IRT. It can be used for test specifications involving a cut-off score. This cut-off score should be reached at a predetermined ability level ϑ^*. Thus, the expected score at $\vartheta = \vartheta^*$ must be within – usually a small – interval between P_t^ℓ and P_t^u. At the same time, the measurement error at $\vartheta = \vartheta^*$ should be minimized in order to minimize classification errors.

The model can be formulated as

$$\text{maximise} \quad \delta \tag{3.7a}$$

subject to

$$\delta \leqslant \sum_i I_i(\vartheta_t^*)x_{it} \quad \forall t \tag{3.7b}$$

further subject to constraints in (2.4).

Note that determining an appropriate interval between P_t^ℓ and P_t^ucan lead to a careful balancing act: a relatively large interval may result in large deviations from desired pass rates, while a relatively small interval may result in test forms with large measurement errors or, in extreme cases, even to infeasibility.

3.3.2 Practical steps for `ATA` use

The `ATA` software allows, with a user-friendly interface, to perform a wide selection of operations related to test assembly, in particular, it is possible to:

1. import the item bank;
2. specify the constraints of test forms;
3. select one of the built-in models to optimize;
4. solve the optimization algorithm;
5. interpret and export the results.

Before adding the constraints into the software the item bank must be imported. This process is really straightforward using `ATA` since we only need to open the Windows form interface clicking on the executable `ATA.exe` and, once the window has appeared, clicking on the top left tab `import csv file`. In this way we tell to `ATA` in which file our item bank is stored (that must be structured in the way explained in 2.1), which is the first line of the data and which are the columns that contain core attributes for the ATA, such as Rasch difficulty or discrimination parameters, CTT p-values or also item content features such as friend sets or enemy sets attributes.

Once the item bank has been imported all the specifications verbally expressed in the desiderata (Section 3.2) are easily input in the software by using the third tab `Test specifications`. In the displayed window it is possible to select the desired optimization model in the ones listed in the right drop-down menu `Model`.

After the choice of the optimization function the software allows to define the constraints by fixing:

- if the tests are parallel, by checking the box `Parallel`;
- the maximum number of overlapped items for each pair of tests, by filling the matrix in the `Overlap` panel;
- the maximum and minimum Item use in the `item use` panel;
- the test length in the tab `Tests`, panel `Test length`;
- the desired categorical constraints just by entering the tab `Tests` and typing upper and lower bounds in the designated textboxes;
- if excluding or forcing the presence of certain items in the tests, always in the tab `Tests`, respectively `Fixed items` and `Not in the tests` panels;
- the upper and lower bound for the test expected score in the panel `Expected Score`.

If the assembly requires to maximise the TIF in some ability points, it is needed to input the population expected value and standard deviation of its ability distribution just by clicking on the tab `Define target population` in the main form.

Once all the specifications has been successfully added `ATA` performs the solving procedure just by clicking the button `Assemble tests`. If the model is feasible the results can be printed on text files using the button `Write test report`.

3.4 INVALSI ATA models

The ITA item banks are characterized by the massive presence of friend (item) sets as they contain the reading comprehension items which are related to a common text. On the other hand, the MAT item banks are characterized by the presence of many enemy sets. Also, the ITA item banks include a larger number of items in comparison to the MAT item banks.

The optimization models used for INVALSI test assembly of school year 2017/2018 are the *Score-Info* and the *Design-Score-Info* (`Score/Info` and `Des/Score/Info` in the `ATA` software menu).

The *Score-Info* model was chosen to perform the optimization based on two main targets:

1. minimize the maximum of the deviations between the expected score of each test form and a desired difficulty range (Score);
2. maximize the test information function(TIF) in the selected ability points (Info).

For the definition of expected score in the Score step, the CTT p-value was used. The Score step allows to create parallel forms by requiring equally difficult tests. On the other hand, the Info step allows to minimize the measurement error by maximizing the information function at specified ability values (see Section 3.3.1 for the details). This model follows the principles illustrated theoretically in Section 1.2 and practically in 2.3.3.

The *Design-Score-Info* model is a slight modification of the *Score-Info* model. Specifically, the optimization algorithm allows to build the test forms by first selecting the items that create the design with the lowest overlap between tests (Design) and secondly following the approach described in the *Score-Info* model.

The model works by

1. minimizing the design overlap (Design);
2. minimizing the maximum of the deviations between the expected score of each test form and a desired difficulty range (Score);
3. maximizing the test information function(TIF) in the selected ability points (Info).

The main goal of the assembly using these models is to ensure the maximum precision possible of the tests (and hence their validity) in estimating the ability of respondents controlling for the value of the average expected score (p-value) taking into account the psychometrics features of the population of students under analysis. Therefore, the Design step allows to have test forms which share as less items as possible deterring the phenomenon of cheating.

The test forms produced according to the two ATA models also share approximately the same composition of items with respect to the domain, the type and the other specific features which characterize the item banks, this is due to the small size of the feasible set of solutions generated by the constraints of type (2.4). A qualitative final check on the test forms was performed by the experts in the ITA and MAT fields.

Bibliography

[1] Berger, M. P., Wong, W. K. (Eds.). (2005). *Applied optimal designs.* John Wiley & Sons.

[2] Cronbach, L. J. (1951). *Coefficient alpha and the internal structure of tests. Psychometrika*, 16, 297–334.

[3] Downing, S., Haladyna, T. (2006). *Handbook of Test Development.* Mahwah, NJ: Lawrence Erlbaum Associates.

[4] Hambleton, R. K., Swaminathan, H. (1985). *Item Response Theory: Principles and Applications.* Boston: Kluwer-Nijhoff.

[5] Hambleton, R. K., Swaminathan, H., Rogers, H. J. (1991). *Fundamentals of Item Response Theory.* Newbury Park, CA: Sage Publications, Inc.

[6] Lord, F. M. (1980). *Applications of Item Response Theory to Practical Testing Problems.* Lord Hillsdale, NJ: Erlbaum Associates.

[7] Lord, F. M., Novick, M. R. (1968). *Statistical Theories of Mental Test Scores.* Reading, MA: Addison-Wesley.

[8] Novick, M. R. (1966). *The axioms and principal results of classical test theory.* Journal of Mathematical Psychology, 3(1), 1-18.

[9] Rao, C. R., Rao, C. R., Statistiker, M., Rao, C. R. (1973). *Linear statistical inference and its applications* (Vol. 2, pp. 263-270). New York: Wiley.

[10] Rasch, G. (1960/1980). *Probabilistic Models for Some Intelligence and Attainment Tests.*(Copenhagen, Danish Institute for Educational Research), expanded edition (1980) with foreword and afterword by B.D. Wright. Chicago: The University of Chicago Press.

[11] Samejima, F. (1977). *Weakly parallel tests in latent trait theory with some criticisms of classical test theory.* Psychometrika, 42 (2), 193–198.

[12] van der Linden, W. J. (2005). *Linear Models for Optimal Test Design.* New York: Springer.

[13] Verschoor, A. J. (2007). *Genetic Algorithms for Automated Test Assembly.* PhD thesis, University of Twente.