The Good, the Bad, and the Ugly: New Item Response Theory based algorithms for the development of short test forms

Author Note

The authors have no conflict of interest to disclose.

Correspondence concerning this article should be addressed to: Ottavia M. Epifania, Department of Philosophy, Sociology, Education, and Applied Psychology, Via Venezia 14, Padova, Italy. E-Mail: ottavia.epifania@unipd.it

Abstract

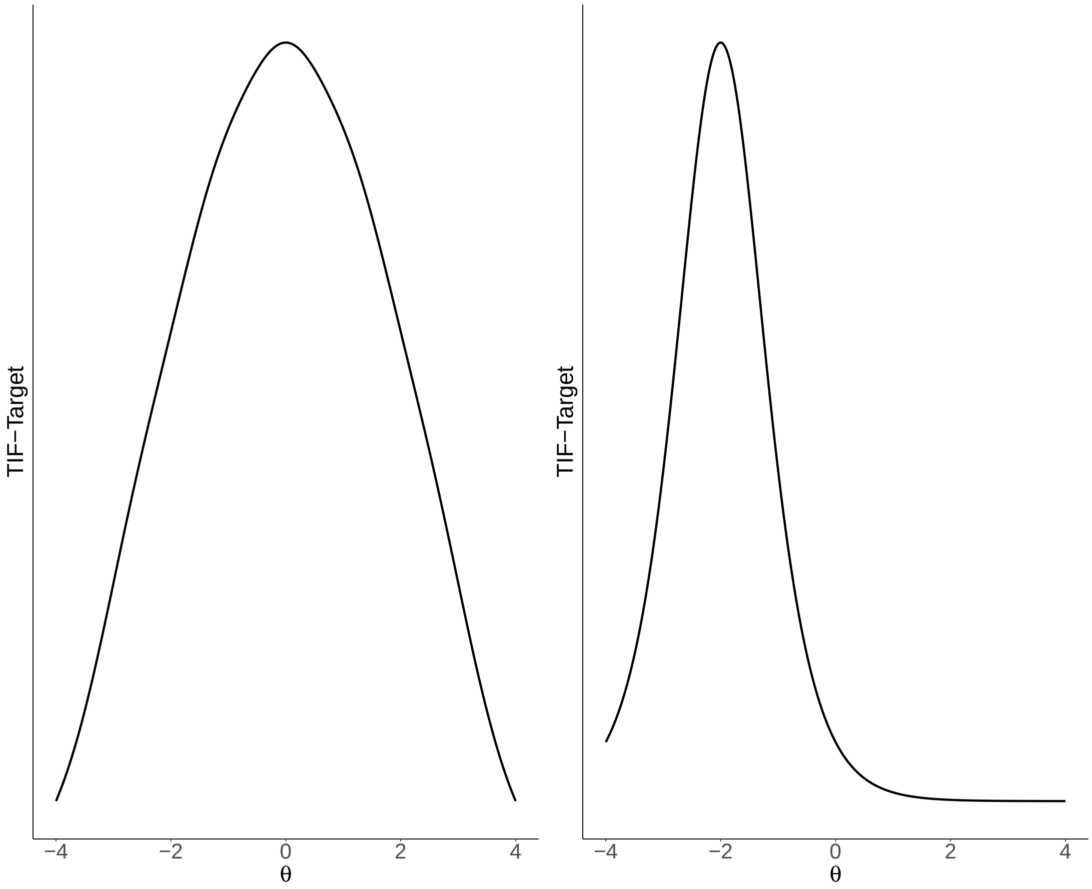
This manuscript presents Item Response Theory-based algorithms for the development of short test forms (STFs) from full-length tests. All algorithms aim at minimizing both the number of items included in the STF and the minimum average distance from a test information target that describes the desired characteristics of the STF, denoted as test information function target (TIF-target). The algorithms differ according to the method used for the item inclusion in the STF at each iteration, while they share the same termination criterion. The item selection is based on the ability of each item to reduce the distance from the TIF-target considering either one specific latent trait level defined at each iteration or the entire latent trait level. The algorithms stop when the last selected item does not contribute in reducing the distance from the TIF-target. The performance of the algorithms is compared against that of a brute force algorithm, which selects the item combination best able to reduce the distance from the TIF-target among all the possible item combinations. Results of a simulation study show that the algorithm that selects the items according to their ability of reducing the distance from the TIF-target throughout the entire latent trait outperforms the other algorithms. Limitations of the study and future lines of research are discussed.

Keywords: Item Response Theory; Information Functions; Algorithms; Short Test Forms

The Good, the Bad, and the Ugly: New Item Response Theory-Based Algorithms for the Development of Short Test Forms

Item Response Theory (IRT) offers detailed insights into the measurement precision of individual items across varying levels of the latent trait, making it an ideal framework for test development (e.g., Baker & Kim, 2017; van der Linden, Theunissen, Boekkooi-Timminga, & Kelderman, 1987). By leveraging its robust analytical capabilities, IRT facilitates the construction of tailored assessments, such as tests designed from item banks or the derivation of short test forms (STFs) from existing assessments. This flexibility ensures that tests can be optimized for specific purposes, populations, or contexts while maintaining precision and validity. The STFs developed from tests can be classified into two main types. On the one hand, there are the adaptive STFs developed within the computerized adaptive testing framework, which tailor the item administration to maximize measurement precision for each respondent (e.g., Magis & Barrada, 2017). Therefore, the items may differ from one respondent to another. On the other hand, static STFs are composed of the same subset of items, which is chosen to maximize the measurement precision across all the administered respondents. This article presents three new IRT-based algorithms for the development of STFs. All algorithms attempt at recreating the desired characteristics of a test, as defined through a target information function, by iteratively selecting the items that reduce as much as possible the distance between the information function of the STF and the target one. In this application, the focus is on the development of static STFs from full-length tests, although the introduced algorithms also apply to the development of static tests from item banks.

In common IRT procedures (e.g., Chiesi, Lau, & Saklofske, 2020; Colledani, Robusto, & Anselmi, 2018; Silvia, 2021), the items are selected for inclusion in the STF according to their measurement precision with respect to different levels of the latent trait that are of interest for the assessment. To pursue this aim, the information functions of all the items of the full-length test are inspected, and the items with the information functions best able to comply with the desired characteristics of the STF are selected for inclusion. Although this procedure offers high degrees of freedom (e.g., the number of items in the STF can be not specified in advance), it strongly relies on the expertise and subjectivity of the researchers, which lead to issues related to replicability. Moreover, since this procedure is not automated, it might be particularly demanding, especially if the full-length test comprises many items. Recently, Epifania, Anselmi, and Robusto (2023) introduced a procedure, denoted as θ-target procedure, which integrates the principles of computerized adaptive testing (i.e., tailoring the item administration to specific latent trait levels) for the development of static STFs from full-length tests, thus allowing for its automation. The newly introduced procedure grounds the item selection on the measurement precision of each item in the full-length test with respect to specific latent trait levels of interest, denoted as θ targets, which are defined a priori according to the aim of the assessment. The procedure selects an optimal item (i.e., the item that provides the highest measurement precision) for each of the identified θ targets. Despite being promising, the discrete definition of the θ targets raises different issues that cannot be ignored. Firstly, since an optimal item is selected for each of the θ targets, the actual number of θ targets determines the number of items included in the STFs. As such, the more the STF is meant to precisely measure larger regions of the latent trait, the larger the number of selected θ targets and the larger the number of items that need to be included in the STF. Moreover, the functioning of the θ-target procedure strongly depends on the availability in the item bank of informative items with respect to the defined θ targets. If the item bank does not contain informative items with respect to one or more θ targets, sub-optimal items might be included in the final item selection. As such, the STF might contain more items than actually needed or items that might not appropriately assess the levels of interest. Finally, the θ targets are defined as discrete values to recreate a sort of "overall" information function of the STF, which is continuous by nature. In this light, it might be more convenient to directly refer to a continuous function describing the desired characteristics of the STF by means of a target information function to be recreated with the final selection of items in the STF (e.g., Lord, 1977). The specification of a TIF-target allows for shifting the focus on different regions of the latent trait, hence STFs addressing different aims can be developed. For instance, Figure 1 illustrates two TIF-targets, which have been specified for obtaining either precise measurements of different levels of the latent trait. Figure 1a illustrates a TIF target for obtaining a precise measurement of the θ levels between -2 and 2. Figure 1b illustrates a TIF target highly informative for θ levels around -2



(a) (b)

Figure 1: Examples of target information functions. Panel 1a illustrates a target information function aimed at obtaining a good measurement precision of latent trait levels between -2 and 2. Panel 1b illustrates a target information function focused on a specific level of the latent trait, $\theta = -2$.

standard deviation below the average population mean of 0 (a cut off at 2 standard deviations below the population mean is a criterion sometimes used in diagnostic tests, see e.g., Balboni et al., 2022; Tassé et al., 2016).

Although the specification of the desired characteristics of the STF through a target information function is reasonable, its actual definition might not be straightforward. While there might be an idea concerning the shape of the target information function (i.e., the regions of the latent trait on which the highest measurement precision is desired), the definition of its height (i.e., the amount of information for different levels of the latent trait) might be troublesome (van ver Linden & Boekkooi-Timminga, 1989). In this sense, if both the shape and height of the target information function can be defined, the test or STF development is performed with the idea of recreating an *absolute information* (Stocking, Swanson, & Pearlman, 1991; van der Linden et al., 1987). Conversely, if only the shape of the target information function is known and can be defined, the test or STF development is performed with the idea of recreating the shape of a *relative information* (van ver Linden & Boekkooi-Timminga, 1989).

This manuscript presents three new IRT-based algorithms for the development of STFs, which ground the item selection on the definition of the general desired characteristics of the STF as expressed by a target test information function. The continuous definition of the desired characteristics allows for overcoming the issues related to the a priori specification of θ targets, hence releasing the constraint on the optimal number of items. As the target information function is empirically defined with a specific shape and height, these algorithms attempt at finding the item selection to include in the STF that is best able to reduce the distance from an absolute information. All algorithms consider the distance between the target information function and the information function obtained on the items selected up to that iteration with the aim of bridging the gap between the two information functions (i.e., bringing the information function of the STF as close as possible to the target one). As it will be further illustrated, they differ according to how they consider the distance between the information functions and hence on how they consider the items for inclusion in the STF. The performance of the algorithms is evaluated in a simulation study, where it is compared against that of a brute force algorithm. Since the brute force algorithm selects the item combination best able to reduce the distance from the target information function among all the possible item combinations of different length, its item selection is considered as the gold standard.

The manuscript is organized as follows. The next section illustrates the IRT model used as reference in this work, along with the definition of the related information functions. The proposed algorithms are then presented, followed by a simulation study that compares their performance in STFs development. The manuscript concludes with final remarks.

Item Response Theory and Information Functions

Different IRT models are available for the analyses of dichotomous responses, which differentiate according to the characteristics of the items that they take into account. In this application, the 2-Parameter Logistic model (2-PL; Birnbaum, 1968) is used as a reference, which considers item difficulty and item discrimination. According to this model, the probability of observing a positive response by person p on item i is formalized as:

$$P(x_{pi} = 1|\theta_p, b_i, a_i) = \frac{\exp[a_i(\theta_p - b_i)]}{1 + \exp[a_i(\theta_p - b_i)]}, \tag{1}$$

where θ_p is the ability parameter of person p (i.e., the latent trait level of p), b_i is the difficulty parameter of item i (i.e., the location of i on the latent trait), and a_i is the discrimination parameter of item i (i.e., the ability of i to distinguish respondents with different levels of θ). The probability of observing a positive response changes as the distance between the ability of the person θ_p and the location of the item on the latent trait b_i changes, such as it is of the 50% when the ability of person p matches the difficulty of item i.

The precision with which each item measures different levels of the latent trait is described by the *item information function*, $\text{IIF}_i = a_i^2 P(\theta, b_i, a_i)[1 - P(\theta, b_i, a_i)]$. In the 2-PL, the IIF reaches its maximum when $\theta_p = b_i$. The precision with which the test as a whole measures the entire latent trait is expressed by the *test information function*, which is the sum of the IIFs of all the items in a test or item bank, $\text{TIF} = \sum_{i \in B} \text{IIF}_i$ (where B is the set of items in a test or in an item bank). Given the dependency between the location of the items on the latent trait, their discrimination parameters, and their informativeness with respect to different latent trait levels, the shape and height of the TIF depends on the spread of the items along the latent trait continuum. Since the TIF is the sum of the IIFs, the higher the number of items in test, the higher the TIF. Alternatively, a mean TIF can be computed by averaging for the number of items. As it will be further illustrated, the computation of the mean TIF might be more convenient when developing STFs.

The Algorithms

The three introduced algorithms aim at approximating as best as possible the desired characteristics of a test, as expressed by a target information function (TIF-target), by selecting the items best able to reduce the distance between the information function of the STF and the target one. Two of the algorithms (item locating algorithm – ILA – and item selecting algorithm – ISA –) can be seen as the natural evolution of the θ-target procedure in that they ground the item selection on the characteristics of each item with respect to a specific θ target, which is defined at each iteration as the latent trait level where the maximum distance between the two information functions is observed. The third one (Frank) overcomes the definition of the θ targets altogether by considering the entire latent trait for the item inclusion at each iteration. As such, it accounts for the measurement precision of each item with respect to the entire latent trait and to the TIF-target. All three algorithms attempt at bridging the gap between the TIF-target and the temporary information function obtained from the items selected up to that iteration. By grounding the item selection on a specific θ target, ILA and ISA bring the two information functions closer considering only one latent trait level at the time. On the other hand, by grounding the item selection on its contribution to the temporary information function, Frank bridges the gap between the two functions considering the entire latent trait.

The ability of ILA, ISA, and Frank to approximate an absolute TIF-target is evaluated by comparing their performance against that of a brute force algorithm. This algorithm tries every possible combination of items without repetition and selects the one best able to bridge the gap between the two information functions among all the possible item combinations attainable from the full-length test. As such, its resulting STF is considered as the gold standard.

All algorithms but the brute force start with an empty item collection, Q, in which an item is included at each iteration until the termination criterion is met. Before any item can be included in Q, the termination criterion tests its usefulness in reducing the distance from the TIF-target by concurrently considering the TIF-target, the TIF obtained from the STF with the last item considered for inclusion (i.e., provisional item), and the TIF obtained from the STF without this last item. If the distance between the former two is greater than or equal to the distance between the latter two, the provisional item is not included in the STF and the algorithm stops. Vice versa, the item is included and the algorithms proceeds to a new iteration. The reasoning is as follows. If the distance between the TIF-target and the STF including the provisional item is greater than or equal to the difference between the TIF-target and the STF without the provisional item, the provisional item does not contribute in reducing this distance. Since the provisional item is the best one among the available options according to the selection method of each algorithm and does not contribute to reduce the distance from the TIF-target, the algorithm stops. When the termination criterion is reached, the final item selection included in the STF is the one in Q up to the last iteration, which does not include the provisional item.

In what follows, the notation $||X||$ is used to denote the cardinality of a generic set X, B denotes the set of items in the full-length test, Q_x indicates the subset $Q_x \subset B$ of items included in the STF by each algorithm $x \in \{\text{Bruto}, \text{ILA}, \text{ISA}, \text{Frank}\}$, and TIF* indicates the TIF-target. The TIF-target is used to express the desired characteristics of the STF in terms of absolute information across the latent trait. Since the focus of the manuscript is on the development of STFs from full-length tests, Q_x is taken to be a proper subset of the items in the test, such that it is not possible to obtain a STF that contains all the items of the test. If the algorithms are applied for the generation of informative tests from an item bank, Q_x can be the subset of B, and all the items can be included in the test.

The brute force algorithm (i.e., Bruto) does not have a proper termination criterion given that it generates all the possible STFs of different length and then chooses the one that allows for minimizing the distance from the TIF*. It follows that Bruto does not have iterations either. The number of iterations performed by the other algorithms cannot be known in advance since they iteratively compare the TIF obtained by including the provisional item with the TIF* until the termination criterion is met. The only constraints on the number of iterations pertains the lower and upper bound. Specifically, since all the algorithms start from an empty set of items, they are forced to perform at least two iterations such that at least one item can be included in the STF. Moreover, the upper constraint is logically imposed by the number of items in the full-length test, such that the maximum number of iterations is equal to the cardinality of the full-length test. Importantly, if the algorithms reach the last iteration and the termination criterion is not met, it is considered as a failure in the their ability to find a STF able to reproduce the TIF*.

Finally, since the TIF is usually obtained by summing the IIFs together, the more the items in a test, the higher the TIF. As such, when the TIF is used as criterion for deciding the best subset of items to include in a STF, the risk is to favor STFs that include more items. To avoid this risk, the mean TIF of the STF is used in the comparison against the TIF-target. In what follows, the mean TIF is simply indicated as TIF.

Bruto

Bruto compares the TIF of all the possible item combinations attainable from the items in B against the TIF* and selects the item combination best able to reduce the average distance from the TIF*, as follows:

$\forall Q \in \mathcal{Q} = 2^B \setminus \{\emptyset, B\}$, compute

1. $TIF^Q = \frac{\sum_{i \in Q} IIF_i}{||Q||}$ and

2. $\overline{\Delta}_{\text{TIF}^Q} = mean(|TIF^* - TIF^Q|)$.

Then, $Q_{bruto} = \arg\min_{Q \in \mathcal{Q}} \overline{\Delta}_{\mathbf{TIF}^Q}$.

For all the possible item combinations in $\mathcal{Q}$ that are attainable from the item bank B (excluding the empty set and the full set of items), Bruto computes: (1.) the mean TIF, TIF^Q and (2.) the average distance $\overline{\Delta}_{\text{TIF}^Q}$ between the TIF* and the TIF^Q. Then, the final subset of items of the STF, Q_{bruto}, is the one that allows for obtaining the least value of $\overline{\Delta}_{\text{TIF}^Q}$, hence for minimizing the distance from the TIF*.

Given that Bruto attempts all the possible item combinations in $\mathcal{Q}$, it ensures to find the one that is best able to minimize the average distance from the TIF*.

Item Locating Algorithm

At each iteration k, the item locating algorithm (ILA) selects the item that minimizes the distance on the latent trait from a specific θ target (denoted as θ_{target}), as follows:

At $k = 0$: TIF$^0(\theta) = 0\,\forall\theta$, $Q^0 = \emptyset$. For $k \geq 0$,

1. $\theta_{target} := \arg\max|\text{TIF}^* - \text{TIF}^k|$
2. $i^* := \arg\min_{i \in B \setminus Q^k} |\theta_{target} - b_i|$
3. pTIF$_{i^*} = \frac{TIF^k + IIF_{i^*}}{||Q^k||+1}$
4. Termination Criterion: $|\text{TIF}^* - \text{pTIF}_{i^*}| \geq |\text{TIF}^* - \text{TIF}^k|$:
    - FALSE: $Q^{k+1} = Q^k \cup \{i^*\}$, TIF^{k+1} = pTIF$_{i^*}$, iterates 1-4
    - TRUE: Stop, $Q_{ILA} = Q^k$

ILA starts at iteration $k = 0$, where $Q^0 = \emptyset$ and TIF$^0(\theta)$ is 0 for all the values of θ. At each iteration: (1.) the θ_{target} is defined as the latent trait level for which the highest distance between the TIF* and the TIF^k is observed; (2.) The item $i \in B \setminus Q^k$ for which the minimum distance from the θ_{target} is observed is considered as the provisional item i^*; (3.) A provisional TIF, pTIF$_{i^*}$ is computed as the average TIF considering the items in Q^k and item i^*, with the condition $i \in B \setminus Q^k$ ensuring that the items are selected only once; and (4.) The termination criterion tests whether the provisional item i^* is useful for reducing the distance from the TIF*. Specifically, if the distance between the TIF* and the pTIF$_{i^*}$ (TRUE) is greater than or equal to the distance between the TIF* and the TIF^k (i.e., the TIF obtained by considering the item in Q^k, without item i^*), the item in i^* does not help in reducing the distance from the TIF*. As such, ILA stops, and the subset of items selected by ILA, Q_{ILA}, is the one in Q^k. Conversely (FALSE), if the distance between the TIF* and the pTIF$_{i^*}$ is lower than that between the TIF* and the TIF^k, then the item in i^* is useful in reducing the distance from the TIF-target, it is included in the set of selected items, and a new iteration starts, $Q^{k+1} = Q^k \cup \{i^*\}$

Item Selecting Algorithm

The item selecting algorithm (ISA) follows the same procedure as ILA. However, ILA and ISA differentiate according to the method with which the provisional items i^* are considered for inclusion at each iteration. While ILA selects the item that minimizes the distance on the latent trait from the θ_{target}, ISA selects the item that maximizes the IIF with respect to the θ_{target}, as follows:

At $k = 0$: $\text{TIF}^0(\theta) = 0\,\forall\theta$, $Q^0 = \emptyset$. For $k \geq 0$,

1. $\theta_{target} := \arg\max|\text{TIF}^* - \text{TIF}^k|$
2. $i^* := \arg\max_{i \in B \setminus Q^k} \text{IIF}_i(\theta_{target})$
3. $\text{pTIF}_{i^*} = \frac{TIF^k + \text{IIF}_{i^*}}{||Q^k||+1}$
4. Termination Criterion: $|\text{TIF}^* - \text{pTIF}_{i^*}| \geq |\text{TIF}^* - \textbf{TIF}^k|$:
    - FALSE: $Q^{k+1} = Q^k \cup \{i^*\}$, $\text{TIF}^{k+1} = \text{pTIF}_{i^*}$, iterates 1-4
    - TRUE: Stop, $Q_{ISA} = Q^k$

Since the procedure is the same as that described for ILA, it will not be discussed further. The only difference concerns the second step of the algorithm (2.), that is the inclusion of the items $i \in B \setminus Q^k$ in $\{i^*\}$. At each iteration, the provisional item i^* that is considered for inclusion in Q^k is the one that maximizes the IIF for the specific θ_{target}.

Frank

Instead of grounding the item selection at each iteration on a single latent trait level (i.e., the θ_{target}) as ILA and ISA do, Frank considers the entire latent trait for the item selection, in that it selects the item whose IIF is best able to reduce the distance from the TIF^* along the entire latent trait, as follows:

At $k = 0$: $\text{TIF}^0(\theta) = 0\,\forall\theta$, $Q^0 = \emptyset$. For $k \geq 0$,

1. $A^k = B \setminus Q^k$

2. $\forall i \in A^k, pTIF_i^k = \frac{\text{TIF}^k + \text{IIF}_i}{||Q^k||+1}$

3. $i^* = \arg\min_{i \in A^k} |\text{TIF}^* - \text{pTIF}_i|$

4. Termination criterion: $|\text{TIF}^* - \text{pTIF}_{i^*}| \geq |\text{TIF}^* - \text{TIF}^k|$:

    - FALSE: $Q^{k+1} = Q^k \cup \{i^*\}$, $TIF^{k+1} = pTIF_{i^*}$, iterates 1-4
    - TRUE: Stop, $Q_{Frank} = Q^k$

When Frank starts ($k = 0$), the subset of items Q^0 is empty and the TIF^0 is 0 for all the θ levels. At each iteration k: (1.) a set of available items is generated as the items in the item bank that have not been included in the STF yet, $A^k = B \setminus Q^k$; (2.) An average provisional TIF, pTIF, is computed by adding the IIF of each of the items in the set of the available items A^k, one at the time, to the TIF obtained from the items in Q^k (The denominator is obtained by adding 1 to the cardinality of Q^k); (3.) Among all the items in A^k, the one that allows for minimizing the distance between pTIF and TIF* is included in i^*; (4.) The termination criterion is tested. If the distance between the TIF* and the pTIF$_{i^*}$ is greater than or equal to the distance between the TIF* and the TIF^k (i.e., the TIF obtained from the items in the subset Q^k, without item i^*) (TRUE), then the item i^* does not contribute in the reduction of the distance from the TIF*, the algorithm stops, and the final item selection is the one without the item in i^*, $Q_{frank} = Q^k$. Conversely (FALSE), the item in i^* does contribute in the reduction of the distance from the TIF*, hence it is included in the set of items and a new iteration starts, $Q^{k+1} = Q^k \cup \{i^*\}$.

Simulation Study

Simulation Design

The performance of the algorithms was compared in 100 replications. For each replication a full-length test B ($||B|| = 11$) was generated by randomly drawing the difficulty b_i^* and discrimination a_i^* parameters from uniform distributions, $\mathcal{U}(-3, 3)$ and $\mathcal{U}(0.9, 2)$, respectively. The TIF* was generated by randomly selecting some of the items from the full-length test B, which is also the set from which the items are selected for inclusion in the STF by each of the algorithms. If the item parameters are not modified for the generation of the TIF*, there is the risk of having implausible instances where the TIF of the STF perfectly reproduces the TIF*. On the other hand, if the item parameters are slightly modified to obtain the TIF*, a more realistic and ecological scenario is obtained where the items selected for the inclusion in the STF can approximate the TIF* without reaching a perfect overlap. To reproduce this last ecological scenario, the difficulty b_i^* and discrimination a_i^* parameters of the items randomly selected for the generation of the TIF* were modified by adding values b_i^+ and a_i^+ randomly drawn from uniform distributions, $\mathcal{U}(-0.20, 0.20)$. A constraint was made on the discrimination parameters to avoid negative discrimination parameters. Specifically, if a discrimination parameter was found to be negative, it was multiplied by -1. For instance, suppose that Items 1, 2, 3 with parameters $b_1^* = -1.0$, $b_2^* = 0.0$, $b_3^* = 1.0$ and $a_1^* = 0.90$, $a_2^* = 1.0$, $a_3^* = 1.50$ are extracted from B and that the random values $b_1^+ = -0.15$ $b_2^+ = 0.02$ $b_3^+ = 0.18$ and $a_1^+ = 0.10$ $a_2^+ = -0.18$ $a_3^+ = 0.01$ are randomly drawn to be added to the original item parameters, then the TIF* is computed based on the $b_1^* + b_1^+ = -1.15$, $b_2^* + b_2^+ = 0.02$, $b_3^* + b_3^+ = 1.18$ and $a_1^* + a_1^+ = 1.00$, $a_2^* + a_2^+ = 0.82$, $a_3^* + a_3^+ = 1.51$ modified parameters.

Comparison Analyses

At each replication, Bruto, ILA, ISA, and Frank were applied. The algorithms were compared in terms of computational time and proportion of successes in finding an item selection able to reproduce the TIF*. A failure is registered anytime an algorithm runs out of items from B without finding a STF able to reduce the mean distance from the TIF*.

Given that Bruto tries every possible item combination $Q \in \mathcal{Q}$, the item selection Q_{Bruto} is the one best able to minimize the average mean distance from the TIF* among all the possible item combinations. All the item selections generated by Bruto can be ordered in terms of increasing average distance from the TIF*, $\forall (Q, Q') \in \mathcal{Q}, Q \preceq Q' \Rightarrow mean(|TIF^* - TIF_Q| \leq |TIF^* - TIF_{Q'}|)$. The positions of the item selections $Q_x, x \in \{\text{ILA}, \text{ISA}, \text{Frank}\}$ can be found among the ranking of the STFs developed by Bruto, and their respective percentile rank can be computed. The percentile ranks of the item selections obtained with the other algorithms are considered for comparing their performance in terms of average distance from the TIF* among all the possible item combinations.

Considering the item selection in Q_{Bruto} as the gold standard, the symmetric distance from Q_{Bruto} of the item selections Q_x, $x \in \{\text{ILA}, \text{ISA}, \text{Frank}\}$ can be computed. The symmetric distance is the cardinality of the set obtained from the union between the set of items that are selected by x but not by Bruto and the set of items that are selected by Bruto but not by x, $Q_x \Delta Q_{Bruto} = ||\{\{Q_x \setminus Q_{Bruto}\} \cup \{Q_{Bruto} \setminus Q_x\}\}||$. A symmetric distance of 0 indicates that Bruto and x selected the same exact set of items, and the lower the value, the better in terms of ability of x to select the same set of items from B as Bruto. Take for instance an item pool $B = \{1, 2, 3, \ldots, 11\}$ ($||B|| = 11$). By applying Bruto, ILA, and Frank, the following item selections are obtained: $Q_{\text{Bruto}} = \{1, 2, 3, 6, 9, 10\}$, $Q_{ILA} = \{1, 4, 5, 7, 9\}$, $Q_{Frank} = \{1, 2, 3, 9\}$, with $||Q_{\text{Bruto}}|| = 6$, $||Q_{\text{Ila}}|| = 5$, and $||Q_{\text{Frank}}|| = 4$. The symmetric distances between the item selection provided by Bruto and those provided by the two other algorithms are $Q_{\text{ILA}} \Delta Q_{Bruto} = ||\{\{Q_{\text{ILA}} \setminus Q_{Bruto}\} \cup \{Q_{Bruto} \setminus Q_{\text{ILA}}\}\}|| = ||\{\{4, 5, 7\} \cup \{2, 3, 6, 10\}\}|| = 7$ and $Q_{Frank} \Delta Q_{Bruto} = ||\{\{Q_{\text{Frank}} \setminus Q_{Bruto}\} \cup \{Q_{Bruto} \setminus Q_{\text{ILA}}\}\}|| = ||\{\{\emptyset\} \cup \{6, 10\}\}|| = 2$, hence suggesting that Frank showed a better performance in selecting and excluding the same items as Bruto than ILA. Precisely, the fact that $Q_{\text{Frank} \setminus Q_{Bruto}} = \emptyset$ implies that $Q_{Frank} \subset Q_{Bruto}$.

Although the symmetric distance provides an overall overview on the ability of the algorithms to retrieve the item selection of Bruto, it cannot disentangle whether the algorithm failed to select items from B that were included by Bruto (underselection) or whether the algorithm failed to exclude the same items as Bruto from B (overselection). This information can be obtained by computing ad hoc statistics from the contingency table between Q_x and Q_{Bruto}:

		Q_{Bruto}	
		$q \in B \setminus Q_{Bruto}$	$q \in Q_{Bruto}$
Q_x	$q \in B \setminus Q_x$	$\vert\vert B \setminus \{Q_x \cup Q_{\text{Bruto}}\}\vert\vert$	$\vert\vert(Q_{Bruto} \setminus Q_x)\vert\vert$
	$q \in Q_x$	$\vert\vert(Q_x \setminus Q_{Bruto})\vert\vert$	$\vert\vert\{Q_x \cap Q_{Bruto}\}\vert\vert$

The main diagonal of the contingency table contains the number of items that have been excluded by both x and Bruto ($||B \setminus (Q_x \cup Q_{\text{Bruto}})||$) and the number of items that have been included by both ($||Q_x \cap Q_{\text{Bruto}}||$). The secondary diagonal contains the discrepancies between the selections provided by x and Bruto, specifically: (i) the number of items that have been excluded by x but have been selected by Bruto ($||(B \setminus Q_x) \cup Q_{\text{Bruto}}||$), and (ii) the number of items that have been excluded by Bruto and selected by x ($||(B \setminus Q_{\text{Bruto}}) \cup Q_x||$). The statistics that can be computed from this contingency table are: (i) Specificity, the proportion of items that have been excluded by both Bruto and x ($||\{B \setminus (Q_x \cup Q_{bruto})\}||/||\{B \setminus Q_{Bruto}\}||$), (ii) sensitivity, the proportion of items that have been selected by both Bruto and x ($||\{Q_x \cap Q_{Bruto}\}||/||Q_{bruto}||$), and (iii) accuracy, the degree to which the item selections of Bruto and x overlaps ($||\{(Q_x \cap Q_{bruto}) \cup B \setminus (Q_x \cup Q_{bruto})\}||/||B||$).

The accuracy, sensitivity, specificity indexes range between 0 and 1, where 1 indicates a perfect correspondence between the item selections of x and Bruto. The accuracy is 1 if both specificity and sensitivity are 1, which means that $Q_x = Q_{bruto}$ (i.e., x selected the same items from B and excluded the same items from B as Bruto). A decrease in the accuracy might occur in different scenarios: (i) a decrease in both the specificity and sensitivity, (ii) a decrease in the sensitivity but not in the specificity, and (iii) a decrease in the specificity but not in the sensitivity. To compute these statistics form the item selections Q_{Bruto}, Q_{ILA}, and Q_{Frank}, two contingencies tables can be obtained, one comparing ILA vs. Bruto (top panel, $Q_{\text{ILA} \times Q_{\text{Bruto}}}$), and one comparing Frank vs. Bruto (top panel, $Q_{\text{Frank} \times Q_{\text{Bruto}}}$).

		Q_{Bruto}		
		$q \in B \setminus Q_{Bruto}$	$q \in Q_{Bruto}$	
Q_{Ila}	$q \in B \setminus Q_{\text{ILA}}$	2	4	6
	$q \in Q_{\text{ILA}}$	3	2	5
		5	6	11

		Q_{Bruto}		
		$q \in B \setminus Q_{Bruto}$	$q \in Q_{Bruto}$	
Q_{Frank}	$q \in B \setminus Q_{\text{Frank}}$	5	2	7
	$q \in Q_{\text{Frank}}$	0	4	4
		5	6	11

The specificity of ILA vs. Bruto is
$||\{B \setminus (Q_{\text{ILA}} \cup Q_{Bruto})\}||/||\{B \setminus Q_{Bruto}\}|| = ||\{8, 11\}||/||\{4, 5, 7, 8, 11\}|| = 2/5 = .40$, the sensitivity is $||\{Q_{ILA} \cap Q_{Bruto}\}||/||Q_{bruto}|| = ||\{1, 9\}/\{1, 2, 4, 6, 9, 10\}|| = 2/6 = .33$, and

the accuracy is $(2+2)/11 = .37$. In a similar vein, the specificity, sensitivity, and accuracy of Frank vs. Bruto are computed, resulting in $5/5 = 1$, $4/6 = .67$, and $(5+4)/11$, respectively. Overall, Frank provides a better accuracy than ILA, due to better specificity and sensitivity of the former than the latter one. Specifically, Frank has a specificity of 1, indicated that it excludes the same items as Bruto from the item selection.

Finally, McNemar tests were run to test the differences in the accuracy, sensitivity and specificity of the algorithms with respect to Bruto across all the replications, with $\alpha = .05$. The McNemar test is based on an approximation of the χ^2 probability distribution with 1 degree of freedom. The critical value is $\chi^2_{.95} = 3.84$ and it is used to test the null hypothesis that the performance of the two considered algorithms with respect to the gold standard are equal (i.e., the two algorithms have the same error rate).

The simulation study was run in R, version XX on a Windows 10 PC (DETTAGLI TECNICI SULLA RAM E ALTRE COSE).

Results

Table 1 reports the computational time and proporton of successes in finding a STF for all the algorithms.

Table 1: Computational time (minutes) and percentage of successes of the algorithms in finding an STF able to minimize the distance from the TIF-target.

Algorithm	Time Range	Mean Time	Percentage of successes
Bruto	10.5 - 12.5	10.8±0.287	100
ILA	0.002 - 059	0.006±0.007	93
ISA	0.002 - 0.055	0.007±0.008	82
Frank	0.008 - 0.517	0.027±0.052	100

Note: The computational time has been computed on the 100 replications. $||\mathcal{Q}|| = 2^{11} \setminus \{\emptyset, 11\} = 2,046$

Not surprisingly, Bruto always found an item combination able to minimize the distance from the TIF-target, as well as Frank. On the other hand, ILA and ISA failed in the 7% and 18% of the replications. Interestingly, the 7 replications for which ILA was not able to recover a STF are included within the 18 failed replications of ISA. The 18 replications for which a STF could not be found according to ILA and ISA were excluded from the comparison with

Bruto. The following results are hence based on the 82 replications for which all the algorithms could find a STF.

Figure 2 illustrates the percentile rank (y-axis) displayed in increasing distances between the STFs developed by Bruto in each of the 82 replications (x-axis) and those developed by ILA (square), ISA (dot), and Frank (triangle).

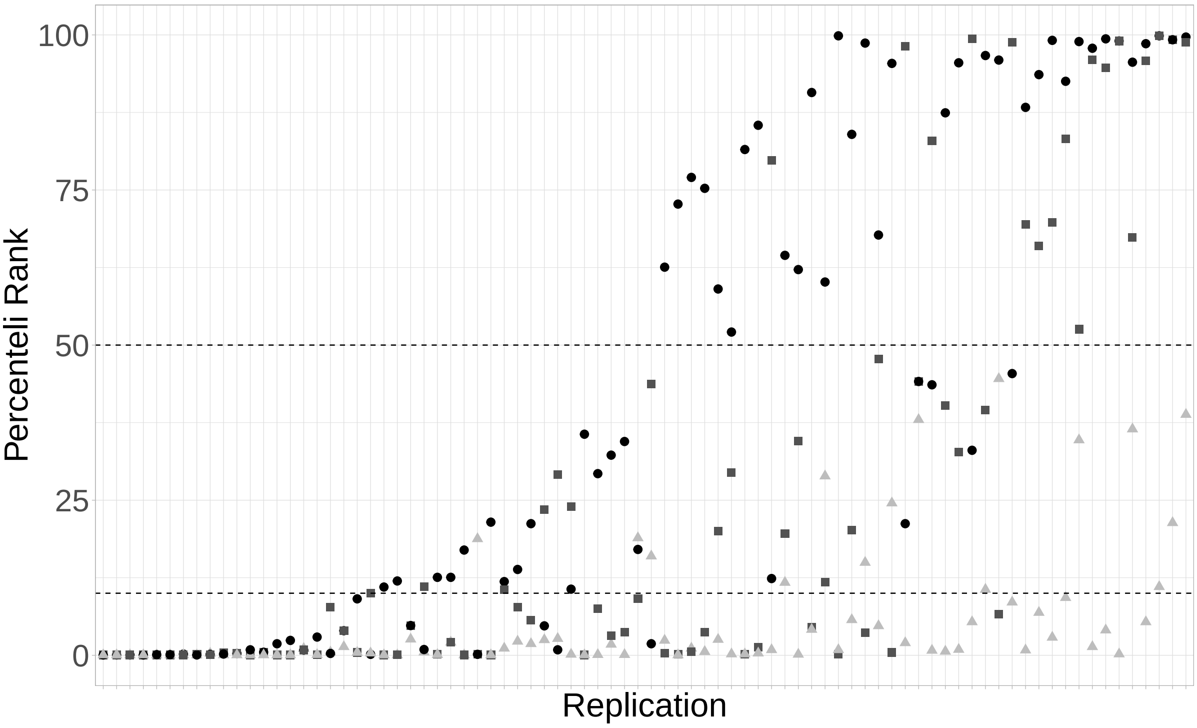


Figure 2: Percentile ranks of the distances from the STFs chosen by Bruto and the STFs produced by ILA (square), ISA (dot), and Frank (triangle). The horizontal dashed lines represent the 10th and 50th percentiles.

All the STFs developed by Frank were within the 50 percentile of distance, while the STFs developed by ILA and ISA exceeded the 50 percentile in the 22% and 42% of cases. Frank resulted in STFs over the 10 percentile of distance in the 18% of cases, ILA and ISA in the 44% and 67% of cases, respectively.

The average symmetric distances from Bruto were: $Q_{\text{ILA}} \Delta Q_{\text{Bruto}} = 2.70$, $Q_{\text{ISA}} \Delta Q_{\text{Bruto}} = 2.87$ and $Q_{\text{Frank}} \Delta Q_{\text{Bruto}} = 2.46$. The item selections of all the algorithms, on average, presented at least 2 items of difference from the selection provided by Bruto, with ISA showing the worst performance and Frank the best one. However, by only considering the symmetric distance it is not possible to ascertain whether these results were mostly ascribable to an over selection of items not included in bruto (lack of sensitivity) or an under selction of items included in Bruto (lack of sensitivity). To better understand the results on the symmetric distance, Table 2 reports the mean accuracy, sensitivity (i.e., items selected by Bruto and x), and specificity (i.e., items selected by neither Bruto nor x).

Table 2: Accuracy, sensitivity and specificity of the algorithms with respect to the best selection provided by Bruto.

Algorithm	Accuracy	Specificity	Sensitivity
ILA	0.56 ± 0.33	0.68 ± 0.39	0.43 ± 0.36
ISA	0.54 ± 0.32	0.64 ± 0.38	0.41 ± 0.37
Frank	0.59 ± 0.34	0.68 ± 0.38	0.50 ± 0.37

Across the 82 replications, Frank provided the highest average accuracy, followed by ILA and ISA. It is worth noting that the accuracy of all algorithms was below the 60%. The McNemar test suggested that, when compared against Bruto, Frank performed better than both ILA and ISA ($\chi^2 = 4.90$ and $\chi^2 = 10.58$, respectively), while no significant differences were found between ISA and ILA ($\chi^2 = 2.43$). To best understand the implications of the results on the accuracy of the algorithms (i.e., whether they are mostly due to an over inclusion of items or an over exclusion of items), the average specificity and sensitivity should be considered. Frank and ILA resulted in the same average specificity, while the sensitivity of ILA was lower than that of Frank. In other words, Frank and ILA, on average, had the same ability of not selecting the items for inclusion in the STF as Bruto. However, ILA was less able than Frank in selecting for the inclusion in the STF the same items as Bruto. The lower accuracy of ILA compared to Frank is hence mostly due to ILA's inability of selecting for inclusion the same items as Bruto. ISA showed the least accuracy and specificity of all the algorithms, although its sensitivity was in line with that of ILA.

Final Remarks

This manuscript presented new IRT-based algorithms for the development of STFs. The functioning of the algorithms depends on the specification of the desired characteristics through the definition of a target information function. While all algorithms aim at reducing as much as possible the distance between the provisional information function of the STF and the target information function, they differentiate according to the methods employed for considering the items for inclusion in the STF at each iteration. Specifically, two algorithms (ILA and ISA) consider either the location or the informativeness of the items in the full-length test with respect to specific latent trait levels, while the third one (Frank) considers the ability of each item in the full-length test of getting the provisional and target information functions closer.

The results of a simulation study where the performance of the algorithms has been compared against that of a brute force algorithm showed that Frank was the only one (besides the brute force one) to always retrieve a STF. Moreover, Frank is the algorithm best able to recreate the TIF-target and to get as close as possible to the item combination selected by the brute force algorithm.

Even among the replications where all the algorithms identified a STF, Frank tended to identify the item selection that was closest, in terms of average distance, from that identified by the brute force algorithm. This result is not surprising given that Frank considers the contribution of each item in reducing the distance from the TIF-target across the entire latent trait rather than focusing on a specific level. By focusing on a specific level of the latent trait, like ILA and ISA do, and selecting the item that only helps in reducing the distance from the TIF-target for that level, the risk is that even items that are not close to the level, and hence do not help in bridging the gap from the TIF-target, can be selected for the evaluation. As such, not useful items (i.e., items that do not reduce the distance) might be included, leading to the termination criterion. On the other hand, Frank considers for the inclusion the item that contributes to reduce the distance from the TIF-target across all levels of the latent trait.

The TIF of a STF would always be lower than the TIF that could be obtained from the full-length test, meaning that generally a STF is less precise than the full-length test in measuring the latent trait. The true advantage of using a STF lies in the fact that respondents would get less tired, and their response accuracy should be maintained during the administration. This may be particularly useful when multiple tests are administered to reduce the overall cognitive burden on the respondents. This point should be further investigated in future simulation and empirical studies, where the tiredness of the respondents during the administration is accounted for by the algorithm for each administered item.

References

Baker, F. B., & Kim, S.-H. (2017). *The basics of Item Response Theory using R.* Springer.

Balboni, G., Bacherini, A., Anselmi, P., Brovedani, P., Buono, S., Micheletti, S., ... Tassé, M. J. (2022). Italian diagnostic adaptive behavior scale: reliability and diagnostic accuracy compared with the vineland-ii. *Research in Developmental Disabilities*, *123*, 104185. doi: https://doi.org/10.1016/j.ridd.2022.104185

Birnbaum, A. (1968). Some latent trait models and their use in inferring an examinee's ability. In F. M. Lord & R. Novick (Eds.), *Statistical theories of mental test scores.* Reading:MA: Addison-Wesley Publishing.

Chiesi, F., Lau, C., & Saklofske, D. H. (2020). A revised short version of the compassionate love scale for humanity (CLS-H-SF): Evidence from Item Response Theory analyses and validity testing. *BMC Psychology*, *8*(1), 1–9. doi: https://doi.org/10.1186/s40359-020-0386-9

Colledani, D., Robusto, E., & Anselmi, P. (2018). Development of a new abbreviated form of the Junior Eysenck Personality Questionnaire-Revised. *Personality and Individual Differences*, *120*, 159–165. doi: https://doi.org/10.1016/j.paid.2017.08.037

Epifania, O. M., Anselmi, P., & Robusto, E. (2023). Pauci sed boni: An item response theory approach for shortening tests [Conference paper]. In (Vol. 422, p. 75 – 83). doi: 10.1007/978-3-031-27781-8_7

Lord, F. M. (1977). Practical applications of item characteristic curve theory. *Journal of Educational Measurement*, 117–138.

Magis, D., & Barrada, J. R. (2017). Computerized adaptive testing with R: Recent updates of the package catR. *Journal of Statistical Software*, *76*, 1–19. doi: https://doi.org/10.18637/jss.v076.c01

Silvia, P. J. (2021). The self-reflection and insight scale: Applying Item Response Theory to craft an efficient short form. *Current Psychology*, 1–11. doi: https://doi.org/10.1007/s12144-020-01299-7

Stocking, M. L., Swanson, L., & Pearlman, M. (1991). Automated item selection using Item

Response Theory. *ETS Research Report Series*, *1991*(1), i–31.

Tassé, M. J., Schalock, R. L., Thissen, D., Balboni, G., Bersani Jr, H., Borthwick-Duffy, S. A., ... Navas, P. (2016). Development and standardization of the diagnostic adaptive behavior scale: Application of item response theory to the assessment of adaptive behavior. *American Journal on Intellectual and Developmental Disabilities*, *121*(2), 79–94. doi: https://doi.org/10.1352/1944-7558-121.2.79

van der Linden, W. J., Theunissen, T., Boekkooi-Timminga, E., & Kelderman, H. (1987). IRT-based test construction.

van ver Linden, W. J., & Boekkooi-Timminga, E. (1989). A maximin model for irt-based test design with practical constraints. *Psychometrika*, *54*, 237–247.