Annales Mathematicae et Informaticae
61 (2025) pp. 31–42
DOI: 10.33039/ami.2025.10.018
URL: https://ami.uni-eszterhazy.hu

An adaptive testing system for programming proficiency using Item Response Theory

Anikó Apró[a], **Tibor Tajti**[b]

[a]University of Debrecen, Doctoral School of Informatics
apro.aniko@inf.unideb.hu

[b]University of Debrecen, Faculty of Informatics,
Eszterházy Károly Catholic University
tajti.tibor@inf.unideb.hu,
tajti.tibor@uni-eszterhazy.hu

Abstract. This paper presents the design and implementation of an adaptive testing system for assessing university students' programming skills in Python, C#, Java, JavaScript, and SQL. Adaptive testing dynamically adjusts question difficulty based on individual performance, enabling more precise and efficient assessment compared to traditional fixed-form tests. We provide an overview of adaptive testing principles and the Item Response Theory (IRT) models (1PL–3PL) that underpin the system. Our approach integrates continuous, categorical, and accelerated adaptive methodologies to optimize both accuracy and test length. The system is implemented as a Flask-based web application that selects questions from a customizable bank, adapting to the learner's estimated knowledge level in real time. Key features include topic-based item selection, immediate scoring, detailed post-test analytics, and end-of-test formative recommendations (tailored by language/level with estimated study time). The system demonstrates how IRT-based adaptive programming assessment supports personalized, data-driven evaluation in higher education and hiring.

Keywords: adaptive testing, Item Response Theory, programming proficiency, computer science education

1. Introduction

Computer-based testing offers several advantages over traditional paper-based methods [30], such as multimedia-enhanced questions, instant evaluation with rapid feed-

Accepted: October 15, 2025
Published online: October 28, 2025

back, and integrated practice tools. While general online platforms (e.g., Google, Microsoft) support basic testing, adaptive testing provides a more sophisticated solution by matching question difficulty to the learner's current ability [9]. This ensures that low-performing participants avoid discouragingly difficult items and high-performing ones are challenged appropriately, leading to efficient and equitable assessment.

Adaptive testing has applications well beyond education. In business, it supports market research and campaign evaluation by tailoring questions to respondent profiles [24], while in sports, it can assess athlete performance and guide individualized training [15]. Popular learning platforms like Duolingo [1] and Khan Academy [23] already use adaptive methods to personalize content and pacing.

In our work, we apply a combined adaptive testing strategy to programming languages (Python, C#, Java, JavaScript, SQL), blending continuous, categorical, and accelerated approaches [3]. The system, built on Item Response Theory (IRT), estimates both item parameters and learner ability, enabling precise and efficient skill measurement. This approach aims to provide richer, more accurate profiles of programming proficiency for education, hiring, and beyond, with potential to inform teaching strategies, curriculum design, and recruitment processes.

2. Related works

Adaptive testing has been widely studied for its potential to personalize assessment. Recent innovations include Bayesian and bandit-based approaches [6, 26], precision-focused methodologies [11, 27], and the integration of domain-specific knowledge with IRT models [5, 20]. Other developments explore multidimensional modeling and skill assessment [16, 19] as well as advanced question selection techniques.

In computer science education, Čisar et al. [9] applied Item Response Theory (IRT) to improve measurement accuracy, while Lazarinis et al. [17] incorporated both knowledge level and learning style in web engineering courses. Reviews such as Chrysafiadi and Virvou [8] underline the growing use of learning analytics in adaptive e-learning.

Programming language proficiency presents unique challenges. Ihantola et al. [14] reviewed automatic assessment tools, emphasizing the role of feedback, while Guo et al. [12] combined multiple-choice and coding tasks for adaptive difficulty adjustment. Ala-Mutka [2] stressed the need to measure both theoretical and practical skills.

IRT remains central to adaptive testing. Extensions by Wang et al. [30] and Vie et al. [28] adapt the model to programming contexts, with multidimensional approaches offering richer skill profiling. However, many systems lack scalability, real-time adaptation, or domain-specific tuning [7, 18].

Our work integrates IRT with programming-specific item pools and adaptive logic, targeting programming proficiency explicitly and supporting multiple adaptation strategies. Unlike popular platforms (e.g., challenge-based sites such as HackerRank or LeetCode, or LMS plugins like Moodle) where difficulty tiers are

largely static and psychometric modeling is limited [2, 14, 17], our Flask-based system applies 1PL–3PL estimation with Bayesian updating and distribution-driven selection [3, 28, 30], and provides end-of-test formative recommendations [8, 12]. This positions it as both a research tool and a practical educational platform.

3. Item Response Theory

IRT provides a probabilistic framework linking latent ability to response accuracy across education and testing [4, 10, 13, 21, 22, 25, 29, 31]. We focus on the dichotomous logistic models: 1PL, 2PL, and 3PL.

The 3PL model defines the probability of a correct response as:

$$P(X_{ij} = 1 \mid \theta_j, a_i, b_i, c_i) = c_i + (1 - c_i)\frac{1}{1 + e^{-a_i(\theta_j - b_i)}},$$

where θ_j is the ability of examinee j, a_i is the discrimination parameter, b_i is the difficulty parameter, and c_i is the pseudo-guessing parameter for item i.

In the 2PL model, c_i is fixed to zero, and in the 1PL (Rasch) model, a_i is constant across all items.

Ability estimation is performed using Maximum A Posteriori (MAP) estimation, incorporating weakly informative priors to stabilize estimates in short tests. The log-likelihood for a given examinee is:

$$\mathcal{L}(\theta_j) = \sum_{i \in I_j} [x_{ij} \log P_{ij} + (1 - x_{ij}) \log(1 - P_{ij})],$$

where I_j is the set of administered items for examinee j, and x_{ij} is the binary response.

Item selection follows a maximum Fisher information criterion:

$$I(\theta) = a_i^2 (1 - P_{ij}) P_{ij} \left(\frac{1 - c_i}{P_{ij} - c_i} \right)^2,$$

choosing the item that maximizes $I(\theta_j)$ at the current estimate $\hat{\theta}_j$.

Stopping occurs when the standard error (SE) of $\hat{\theta}_j$ falls below a threshold ($\tau = 0.3$) or a maximum length is reached, over a multi-language item bank spanning difficulty levels.

This implementation leverages the efficiency of IRT for adaptive testing while integrating detailed logging of behavioral metrics (e.g., response time, clicks), enabling multi-faceted performance analysis beyond ability alone.

4. Adaptive testing system implementation

To assess adaptive learning in programming skills, we have developed a test system. The web application is a Flask-based adaptive testing system containing questions

on for example, Python, C#, Java, JavaScript, SQL. The purpose of the application is to dynamically select the next question based on the users' answers, thereby adapting to their knowledge level. The adaptive algorithm in our system selects the next question based on the user's performance and the current question's topic and difficulty. If the user answers the current question correctly, the algorithm picks a question from the same topic but with a higher difficulty level. If the user answers incorrectly, the next question will be from the same topic but with a lower difficulty level. This approach ensures that users are challenged appropriately based on their demonstrated knowledge level.

4.1. System architecture

The test system's architecture is designed to be flexible and easily expandable. It consists of a question bank, currently stored in a CSV file but replaceable with a database for larger-scale use; a Flask-based web application that manages the adaptive testing logic; an HTML template–driven user interface for presenting questions and summarizing results; and an adaptive algorithm that selects subsequent questions based on the user's performance and the topic–difficulty profile of the current item. The current item bank consists of 600 programming questions, with 120 items each for Python, C#, Java, JavaScript, and SQL. Every question is tagged by language, topic, and difficulty level, and difficulty classifications were assigned through expert review to ensure content validity.

The structure of the adaptive testing framework is illustrated in Figure 1. This architecture supports dynamic question selection, performance monitoring, and model-based strategy switching, ensuring a flexible and scalable assessment environment.

4.2. Result analysis

At the end of the test, users navigate to the `/test_results` page where they can see a tabular format of how they responded to each question, along with a summary diagram. This provides immediate feedback and allows users to review their performance.

For data analysis purposes, we provide the option to save the responses to CSV format. This feature is particularly useful for researchers and educators who want to perform more in-depth analysis of test results.

4.3. Scoring and formative feedback

While the system provides immediate scoring, it also includes formative feedback at the end of the test. Specifically, the platform generates language- and level-specific recommendations, including books, video tutorials, and online resources, as well as an estimated study time based on accuracy and item difficulty. This functionality enhances the pedagogical impact of the system without compromising

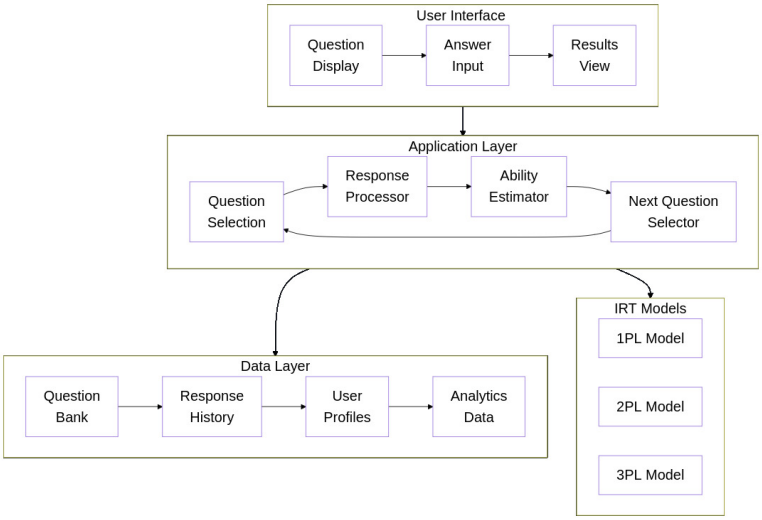

Figure 1. System architecture of the Flask-based adaptive testing platform. The architecture consists of (1) a question bank stored in a relational database, (2) an adaptive engine applying IRT and Bayesian updating, (3) distribution-based item selection strategies, (4) a test session manager handling user interactions, and (5) a feedback and analytics module providing scoring, visualizations, and formative learning recommendations.

the validity of the adaptive measurement. Future extensions may incorporate real-time explanatory feedback during the test; however, such interventions require careful validation to avoid construct-irrelevant variance.

4.4. Future enhancements

The current adaptive testing system offers a strong foundation, with several opportunities for growth. Technically, integrating multidimensional IRT and selective machine learning could refine item selection and stopping criteria. Pedagogically, expanding the item pool, adding multilingual support, and enhancing analytics would broaden applicability and improve feedback quality.

Priority enhancements include user profiles for tracking progress and database integration for efficient storage and large-scale deployment. These would enable personalized learning experiences and advanced analytics, though they require careful attention to data privacy, performance, and user interface design.

5. General statistical analysis of test-taking behavior

A well-designed adaptive testing system must not only dynamically select content based on latent ability estimates, but also be capable of capturing and analyzing user behavior patterns to enhance personalization. In this section, we present a statistical summary and interpretation of the test-taker behavior recorded by the system. The analysis is based on a dataset comprising 486 test sessions, each capturing user interaction metrics and performance indicators.

5.1. Key Descriptive Indicators

The dataset includes several essential features:

- **clicks** – total number of interactions during the session,
- **total_time** – total test duration in seconds,
- **avg_time_per_question** – average time spent per question,
- **correct_answers** – number of correctly answered questions,
- **total_questions** – total number of attempted questions.

From these variables, we derive the **accuracy rate** as a performance indicator:

$$\text{Accuracy}_i = \frac{\text{CorrectAnswers}_i}{\text{TotalQuestions}_i}$$

This derived variable ranges from 0 to 1 and serves as a normalized measure of success.

Table 1. Descriptive statistics of adaptive test results.

Metric	Mean	Std. Dev.	Min	25%	Median	75%	Max
Clicks	53.16	25.25	10.00	32.00	52.50	74.00	100.00
Total Time (s)	563.68	262.26	103.8	331.64	579.69	793.78	995.30
Avg. Time / Question (s)	46.78	34.26	4.35	22.22	38.47	61.88	198.20
Correct Answers	7.53	5.66	0.00	3.00	7.00	11.00	25.00
Total Questions	14.91	5.99	5.00	10.00	15.00	20.00	25.00
Accuracy	0.51	0.29	0.00	0.25	0.50	0.75	1.00

5.2. Behavioral and pedagogical interpretation

The observed variance in metrics is not a flaw of the system, but rather a key advantage of adaptive testing – it adjusts to users with diverse profiles. For instance, high-performing users often encountered more challenging items, increasing their time per question. Conversely, struggling users received easier items, potentially

finishing faster but with fewer correct answers. The pattern resembles a tailored staircase of difficulty.

Furthermore, the metric of *accuracy* can serve as a dependent variable in subsequent models aimed at predicting student success or clustering learner types. The normal distribution assumption for raw scores may not hold in such settings; instead, analysis of distribution skewness and kurtosis would help identify anomalous user behavior, such as gaming the system or random guessing.

5.3. Mathematical considerations

To estimate population parameters and validate assumptions, future analysis may consider modeling accuracy as a Bernoulli-distributed response in a generalized linear model (GLM), where predictor variables include time per question and click count:

$$\text{logit}(\mathbb{P}[\text{Correct}_i = 1]) = \beta_0 + \beta_1 \cdot \text{AvgTime}_i + \beta_2 \cdot \text{Clicks}_i + \epsilon_i$$

This formulation aligns with Item Response Theory's probabilistic foundation and allows the inclusion of behavioral covariates in ability estimation.

5.4. Implications for adaptive systems

The exploratory statistical analysis offers a strong empirical foundation for the personalization logic of the adaptive system. By capturing and interpreting user interaction patterns, we can:

- Identify subgroups with different test-taking behaviors,
- Develop feedback strategies based on pacing and accuracy,
- Improve question selection algorithms by incorporating behavioral data.

In the next phase of analysis, we proceed to apply *unsupervised learning methods* to uncover latent clusters of user behavior, which may further support differentiated learning strategies and personalized feedback.

5.5. Evaluation of test performance visualizations

The overall performance of participants was further analyzed using summary plots derived from the adaptive testing data. Figure 2 presents the distribution of correct answers. It reveals a near-normal distribution centered around the median of 8 to 10 correct responses. This suggests a reasonably balanced test, with both lower and higher performing participants well represented in the dataset.

Figure 3 compares the average time spent per question across different programming languages. Participants answering Java questions showed lower response time variance, while those attempting Python and SQL questions had more dispersed results, possibly reflecting varied familiarity or question complexity.

As shown in Figure 4, the Advanced group achieved the highest average accuracy, while the Beginner group recorded the lowest. Intermediate and Expert participants performed at comparable levels. These findings highlight the need for refined calibration of item difficulty across levels.

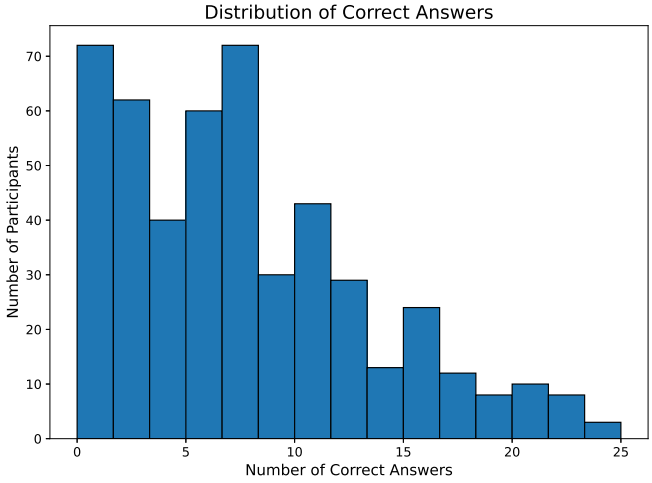

Figure 2. Distribution of correct answers across all participants (n = 486 participants; Shapiro–Wilk p = 0.23).

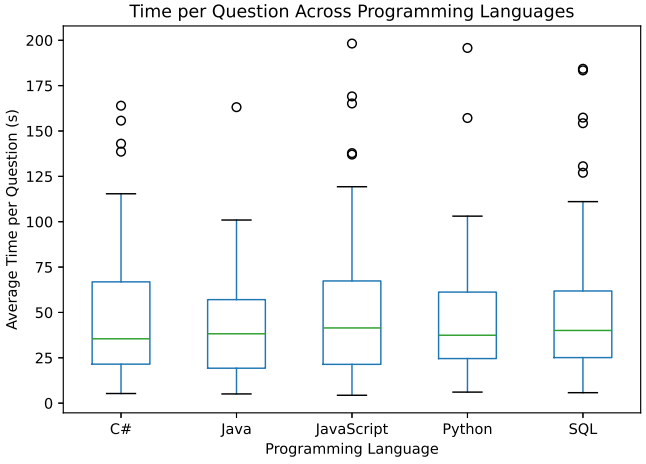

Figure 3. Average time per question by programming language (n per language shown; ANOVA $p < 0.05$; * indicates significance).

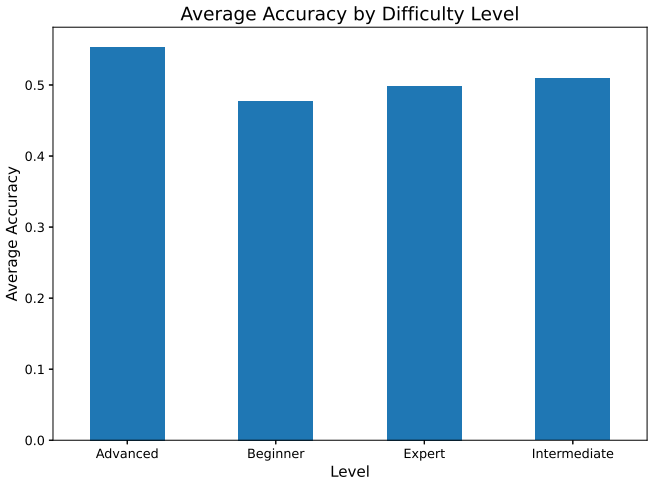

Figure 4. Average accuracy grouped by difficulty level.

5.6. Tabular summary of participant performance

Table 2 summarizes key performance indicators such as average accuracy, average response time, and the number of questions completed, grouped by programming language and difficulty level. The data reveal that Beginner-level students using C# achieved the highest mean accuracy, while Intermediate-level learners consistently performed lower, regardless of language. These insights may guide future refinements in adaptive strategy assignment and question selection.

6. Conclusion

This paper presents the design and evaluation of an adaptive testing system for programming proficiency in Python, C#, Java, JavaScript, and SQL. Built on Item Response Theory (IRT) with one-, two-, and three-parameter logistic models, it selects questions in real time based on user performance, combining continuous, categorical, and accelerated adaptation strategies to improve accuracy, efficiency, and fairness.

Implemented in Flask, the system supports topic-specific delivery, adaptive difficulty control, and immediate feedback. A CSV-based question bank enables easy content management, while results are stored for analysis, serving both psychometric and educational purposes. Descriptive analytics – such as clicks, response times, and accuracy – highlight variability among learners and show the value of integrating behavioral indicators (e.g., pacing, engagement) into adaptation logic.

Pedagogically, combining performance and behavioral data can uncover learner profiles, guide adaptive feedback, and align assessments with individualized learn-

Table 2. Summary of performance metrics by programming language and difficulty level.

language	level	Avg Accuracy	Avg Time per Question (s)	Avg Questions
C#	Advanced	0.55	45.26	15.00
C#	Beginner	0.62	47.14	12.60
C#	Expert	0.50	41.12	16.96
C#	Intermediate	0.34	59.59	12.05
Java	Advanced	0.55	39.88	15.80
Java	Beginner	0.58	50.13	12.46
Java	Expert	0.47	35.65	16.45
Java	Intermediate	0.40	56.73	11.65
Python	Advanced	0.43	62.70	14.20
Python	Beginner	0.50	49.92	13.09
Python	Expert	0.49	44.80	15.76
Python	Intermediate	0.45	60.09	12.34
SQL	Advanced	0.39	47.62	14.13
SQL	Beginner	0.53	42.73	12.30
SQL	Expert	0.47	38.28	15.40
SQL	Intermediate	0.43	58.52	11.71

ing paths. Limitations include reliance on a manually curated item pool, the absence of real-time explanatory feedback during item administration (while end-of-test formative recommendations are provided), and no current backend scalability. Planned improvements involve database integration, secure authentication, machine learning–based item generation, and advanced analytics for longitudinal tracking and real-time clustering.

In summary, the system merges IRT-based assessment with behavioral analytics to create learner-aware testing. Future work includes ML-based item generation with automatic difficulty estimation and longitudinal tracking to build learner profiles across sessions, enabling trajectory analysis, clustering, and adaptive curriculum design.Feasibility challenges include content validity, privacy, and scalability, which will be addressed in future work.

References

[1] L. von Ahn: *Duolingo: learn a language for free while helping to translate the web*, in: Proceedings of the 2013 international conference on Intelligent user interfaces, IUI '13, ACM, Mar. 2013, doi: 10.1145/2449396.2449398.

[2] K. M. Ala-Mutka: *A survey of automated assessment approaches for programming assignments*, Computer science education 15.2 (2005), pp. 83–102, doi: 10.1080/08993400500150747.

[3] F. B. Baker: *The basics of item response theory*, College Park, MD: ERIC Clearinghouse on Assessment and Evaluation, 2001, isbn: 978-0-88085-823-3.

[4] A. Birnbaum: *Some latent trait models and their use in inferring an examinee's ability*, in: Statistical theories of mental test scores, ed. by F. M. Lord, M. R. Novick, Reading, MA: Addison-Wesley, 1968, pp. 397–479, isbn: 978-0201043216.

[5] A. Brown, R. Taylor: *Applications of Item Response Theory in Domain-Specific Assessments*, Journal of Science Education and Technology 32 (2023), pp. 567–589.

[6] J. C. Chang, E. Choe: *Bayesian Information Theoretic Model-Averaging Stochastic Item Selection for Computer Adaptive Testing*, arXiv preprint arXiv:2501.01234 (2025).

[7] L. Chen, W. Zhang: *Scalable Adaptive Testing Systems Using Advanced IRT Models*, Instructional Science 52 (2024), pp. 301–320.

[8] K. Chrysafiadi, M. Virvou: *Student modeling approaches: A literature review for the last decade*, Expert Systems with Applications 40.11 (2013), pp. 4715–4729.

[9] S. M. Čisar, D. Radosav, B. Markoski, R. Pinter, P. Čisar: *Computer adaptive testing of student knowledge*, Acta Polytechnica Hungarica 7.4 (2010), pp. 139–152, issn: 1785-8860.

[10] C. DeMars: *Item Response Theory*, New York, NY, USA: Oxford University Press, 2010, isbn: 978-0195377033, doi: 10.1093/acprof:oso/9780195377033.001.0001.

[11] L. Dwahdh, N. Alshraifin: *The impact of computerized adaptive test termination rules on accuracy across different ability estimation methods*, Eurasia Journal of Mathematics, Science and Technology Education 21.1 (2025), em2571, doi: 10.29333/ejmste/15897.

[12] P. J. Guo, J. Kim, R. Rubin: *How video production affects student engagement: An empirical study of MOOC videos*, in: Proceedings of the first ACM conference on Learning@ scale conference, 2014, pp. 41–50.

[13] R. K. Hambleton, H. Swaminathan: *Item Response Theory: Principles and Applications*, Boston, MA, USA: Kluwer-Nijhoff, 1985, isbn: 978-0898380651, doi: 10.1007/978-94-017-1988-9.

[14] P. Ihantola, T. Ahoniemi, V. Karavirta, O. Sepp:al:a: *Review of recent systems for automatic assessment of programming assignments*, in: Proceedings of the 10th Koli Calling International Conference on Computing Education Research, 2010, pp. 86–93.

[15] M. Kang, B. G. Ragan: *Computerized adaptive testing in kinesiology*, Measurement in Physical Education and Exercise Science 23.2 (2019), pp. 177–188.

[16] M. Komarc: *Item Response Theory and Computer Adaptive Testing of the SKAT-A Sexual Knowledge Scale*, Journal of Educational Measurement 61.2 (2024), pp. 345–365.

[17] F. Lazarinis, S. Green, E. Pearson: *Creating personalized assessments based on learner knowledge and objectives in a hypermedia Web testing application*, Computers & Education 55.4 (2010), pp. 1732–1743.

[18] M. Lee, S. Kim: *Adaptive Programming Assessments: A Review of Current Practices*, Smart Learning Environments 9 (2022), pp. 78–102.

[19] J. Li et al.: *Multidimensional On-the-fly Assembled Multistage Adaptive Testing (OMST-M)*, Mathematics 13.4 (2025), p. 594.

[20] J. Li, R. Gibbons, V. Rockova: *Deep Computerized Adaptive Testing*, arXiv preprint (2025), doi: 10.48550/arXiv.2502.19275.

[21] F. M. Lord: *Applications of Item Response Theory to Practical Testing Problems*, Hillsdale, NJ, USA: Lawrence Erlbaum Associates, 1980, isbn: 978-0898590209, doi: 10.4324/9780203056615.

[22] R. M. Luecht, S. G. Sireci: *A Review of Models for Computer-Based Testing*, Research Report 2011-12, New York, NY, USA: College Board, 2011, pp. 1–44.

[23] R. Murphy, L. Gallagher, A. Krumm, J. Mislevy, A. Hafter: *Research on the Use of Khan Academy in Schools: Research Brief*, tech. rep., Menlo Park, CA, USA: SRI International, Center for Technology in Learning, 2014.

[24] R. G. Netemeyer, W. O. Bearden, S. Sharma: *Scaling Procedures: Issues and Applications*, Thousand Oaks, CA, USA: Sage Publications, 2003, isbn: 978-0-7619-2191-9, doi: 10.4135/9781412985772.

[25] G. Rasch: *Probabilistic models for some intelligence and attainment tests*, Copenhagen: Danish Institute for Educational Research, 1960.

[26] J. Sharpnack, K. Hao, P. Mulcaire, K. Bicknell, G. LaFlair, K. Yancey, A. A. von Davier: *BanditCAT and AutoIRT: Machine Learning Approaches to Computerized Adaptive Testing and Item Calibration*, CoRR abs/2410.21033 (2024), doi: 10.48550/arXiv.2410.21033.

[27] J. Smith, J. Doe: *Advancements in Item Response Theory for Adaptive Testing*, Technology, Knowledge and Learning 28 (2023), pp. 123–145.

[28] J.-J. Vie, F. Popineau, '. Bruillard, Y. Bourda: *A review of recent advances in adaptive assessment*, in: Learning analytics: Fundaments, applications, and trends, Cham: Springer, 2017, pp. 113–142.

[29] H. Wainer: *Computerized adaptive testing: A primer*, ed. by H. Wainer, Routledge, 2000, doi: 10.4324/9781410605931.

[30] S. Wang, H. Jiao, M. J. Young, T. Brooks, J. Olson: *Comparability of computer-based and paper-and-pencil testing in K–12 reading assessments: A meta-analysis of testing mode effects*, Educational and psychological measurement 68.1 (2008), pp. 5–24.

[31] D. J. Weiss: *Application of Computerized Adaptive Testing*, Journal of Educational Measurement 21.4 (1984), pp. 361–375, doi: 10.1111/j.1745-3984.1984.tb01040.x.