

Integrating Answer Set Programming and Item Response Theory for Adaptive Assessment in E-Learning

Fabrizio Marino^{1,†}, Giovanni Melissari^{1,†} and Kristian Reale^{2,*,†}

¹DLVSystem L.T.D., Via della Resistenza 19/C, Rende (CS), Italy

²Department of Mathematics and Computer Sciences, University of Calabria, Rende (CS), Italy

Abstract

Adaptive assessment systems should accurately estimate learner ability while also satisfying pedagogical requirements such as prerequisite relationships, balanced topic coverage, and personalized learning paths. This paper presents an Adaptive Quiz Framework for personalized e-learning assessment that integrates Item Response Theory (IRT) and Answer Set Programming (ASP). The framework dynamically selects the next question based on the estimated ability of each learner, item difficulty, and associated learning topic. Quiz content can be sourced from existing question banks or created through a graphical authoring interface, optionally supported by a Large Language Model (LLM) to generate topic structures, author assessment items, and provide preliminary difficulty estimates. The proposed integration of adaptive modeling and declarative reasoning illustrates how adaptive assessment can jointly address measurement accuracy and pedagogical coherence.

Keywords

Adaptive Assessment, Adaptive Quiz, E-Learning, Item Response Theory, Answer Set Programming

1. Introduction

The increasing adoption of digital learning environments has transformed the way educational activities are delivered and assessed. Learning Management Systems (LMSs) and online assessment platforms have become essential tools for supporting teaching activities, particularly by enabling the creation and administration of digital quizzes. Among the assessment strategies enabled by these platforms, Computerized Adaptive Testing (CAT) [1, 2] has long been recognized as an effective approach for tailoring test difficulty to the ability of each individual learner, dynamically selecting the next item according to the current ability estimate and previous responses. Most existing assessment solutions, however, still rely on static question sets or predefined evaluation strategies, providing limited support for adapting the assessment process to the evolving knowledge state of individual learners.

Where adaptivity is supported, it typically rests on Item Response Theory (IRT) [3, 4], a well-established statistical framework for estimating latent learner abilities and evaluating the relationship between item characteristics and response probabilities. Traditional IRT-based adaptive testing approaches, however, mainly optimize the statistical efficiency of item selection and often lack mechanisms for incorporating pedagogical aspects, such as prerequisite relationships between topics, balanced coverage of learning objectives, instructor-defined priorities, and personalization constraints for learners with special educational needs. Some of these limitations have been partially addressed in the literature, although usually in isolation. Content balancing, for instance, has been studied through automated test assembly, where item selection is typically formulated as an integer linear programming problem [5]. Separately, the propagation of proficiency estimates across related topics has been explored through

AI-LEAP 2026 - 1st International Workshop on AI Learning, Education And Practice, co-located with AIXIA 2026, University of Perugia, October 2026

*Corresponding author.

[†] These authors contributed equally.

✉ fabrizio.marino@dlvsystem.it (F. Marino); giovanni.melissari@dlvsystem.it (G. Melissari); kristian.reale@unical.it (K. Reale)

ORCID 0009-0001-8140-6859 (F. Marino); 0000-0001-5307-8491 (G. Melissari); 0000-0002-5988-2429 (K. Reale)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

graph-informed extensions of IRT, such as T-SKIRT [6], and through prerequisite-aware knowledge tracing models [7]. While some MIP-based approaches address these requirements, to the best of our knowledge, no existing approach jointly supports prerequisite reasoning, content balance, and personalization within a single declarative and extensible specification.

Answer Set Programming (ASP) [8, 9, 10] is a declarative paradigm widely adopted in Artificial Intelligence (AI), Knowledge Representation and non-monotonic Reasoning, in which a problem is specified as a set of facts, rules, and constraints, and answer sets represent solutions. Hard constraints exclude unacceptable candidates, while weak constraints rank the rest by priority, allowing several, potentially conflicting requirements to be combined within a single specification rather than reduced to one score. ASP is therefore well suited to closing the gap discussed above.

Building on this idea, this paper presents an Adaptive Quiz Framework proposal for personalized e-learning assessment that integrates IRT and ASP. IRT is used to estimate learner ability and to quantify the statistical informativeness of candidate items, while an ASP program encodes the actual item-selection logic, enforcing prerequisite relationships within a topic dependency graph, maintaining balanced topic coverage weighted by instructor-assigned topic importance, and satisfying additional personalization constraints that are difficult to capture through a purely statistical selection strategy. The framework has been implemented as a working prototype, and this paper focuses on its architecture and the integration of IRT and ASP. Evaluation with real assessment scenarios is left to future work.

2. Preliminary Concepts

2.1. Answer Set Programming and DLV2

Answer Set Programming (ASP) is a declarative programming paradigm for knowledge representation and reasoning rooted in logic programming under the stable model semantics [11]. An ASP program is a finite set of rules of the form

$$\alpha_1 \mid \dots \mid \alpha_k \text{ :- } \beta_1, \dots, \beta_n, \text{ not } \beta_{n+1}, \dots, \text{ not } \beta_m,$$

where α_i and β_j are atoms and *not* denotes default negation. The atoms in the left-hand side form the rule *head*, while those in the right-hand side define its *body*. Rules with an empty head are *constraints*, used to eliminate invalid solutions, whereas rules with an empty body represent *facts*. ASP solutions are represented by *answer sets*, sets of atoms satisfying all rules under stable model semantics.

ASP supports additional constructs for modeling combinatorial and optimization problems. In particular, *choice rules* enable the representation of alternative decisions, *aggregates* such as *#count*, *#sum*, *#min*, and *#max* allow reasoning over sets of atoms and numerical constraints, and *weak constraints* express optimization criteria by associating penalties with undesirable solutions. These features make ASP suitable for declarative modeling of complex decision-making problems [10].

In this work, we rely on DLV2 [12], an ASP solver compliant with the ASP-Core-2 standard [9] and extended with support for *external atoms* [13]. External atoms allow a logic program to invoke computations defined outside the ASP solver. They are expressed as

$$\&p(t_0, \dots, t_n; u_0, \dots, u_m),$$

where terms before and after the semicolon denote input and output parameters, respectively. The corresponding custom external function, implemented in Python, computes the output values, enabling the integration of ASP reasoning with auxiliary computational modules.

2.2. Item Response Theory Concepts

Item Response Theory. Item Response Theory (IRT) [3, 14, 4] models the probability of a correct response to an item as a function of an unobserved latent trait of the respondent, commonly referred to as *ability* and denoted by $\theta \in \mathbb{R}$. In the one-parameter logistic model (1PL), also known as the Rasch model, the probability of correctly answering item i is

$$P_i(\theta) = \frac{1}{1 + e^{-(\theta - b_i)}}, \quad (1)$$

where b_i is the difficulty parameter of item i .

Expected A Posteriori. During an adaptive assessment process, the estimate of the latent ability can be progressively updated as new responses arise. In particular, given a sequence of responses $u_1, \dots, u_n \in \{0, 1\}$, where $u_i = 1$ denotes a correct response, to items with known difficulty parameters $b_1, \dots, b_n$, the likelihood of the observed response sequence for a candidate ability value θ is

$$L_n(\theta) = \prod_{i=1}^n P_i(\theta)^{u_i} (1 - P_i(\theta))^{1-u_i}. \quad (2)$$

The Expected A Posteriori (EAP) estimator for IRT [14, 15] estimates the latent ability by computing the expected value of the posterior distribution. Since the posterior distribution does not generally admit a closed-form solution, EAP approximates it by evaluating the distribution over a finite grid of candidate ability values

$$\Theta = \{\theta_1, \dots, \theta_m\}.$$

Each grid point θ_k is associated with an initial prior weight w_k , obtained from a prior distribution over the ability parameter, typically a standard normal distribution $\theta \sim \mathcal{N}(0, 1)$. After observing the response sequence, the posterior weight associated with each θ_k is updated according to

$$\pi_k = \frac{L_n(\theta_k)w_k}{\sum_{j=1}^m L_n(\theta_j)w_j}. \quad (3)$$

The EAP estimate is then obtained as the posterior mean

$$\hat{\theta}_{EAP} = \sum_{k=1}^m \theta_k \pi_k. \quad (4)$$

After each new response, the likelihood is updated and the posterior weights are recomputed, allowing the ability estimate to be progressively refined throughout the adaptive assessment process.

Prerequisite Relationship. While conventional IRT represents learner proficiency through a single latent ability value, this assumption may be too restrictive when an assessment covers multiple knowledge areas. In such cases, different knowledge components may require different proficiency estimates, especially when dependencies exist among them.

To address this limitation, T-SKIRT [6] extends the IRT formulation by introducing a structured representation of learner knowledge. Instead of modeling proficiency through a single latent variable, the learner state is represented as a vector of concept-level abilities:

$$\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_n),$$

where each component θ_t denotes the latent ability associated with concept t .

T-SKIRT organizes concepts through expert-defined prerequisite relationships, represented as a directed acyclic graph $G = (T, E)$, where T is the set of concepts and $E \subseteq T \times T$ is the set of prerequisite relations. An edge $(p, t) \in E$ indicates that concept p is a prerequisite for concept t . Rather than estimating the abilities of different concepts independently, T-SKIRT assumes that related concepts should exhibit correlated proficiency levels. This dependency is modeled through a Gaussian prior over the ability vector:

$$\log P(\boldsymbol{\theta}) = -\lambda \sum_{p \in T} \theta_p^2 - \gamma \sum_{(p,t) \in E} (\theta_p - \theta_t)^2, \quad (5)$$

where λ controls the shrinkage of concept abilities toward the prior mean, while γ regulates the influence of the prerequisite structure.

The first term acts as a global regularization component, preventing unbounded ability estimates, whereas the second term encourages similar ability values for concepts connected by prerequisite relationships. Consequently, evidence collected for one concept can indirectly influence the estimation of related concepts, producing learner models that better capture the structure of the underlying knowledge domain.

3. The Adaptive Quiz Framework

We now describe the proposed Adaptive Quiz Framework, detailing its architecture and the methodology underlying the adaptive assessment process.

3.1. System Architecture

Figure 1 illustrates the overall architecture of the proposed Adaptive Quiz Framework, which is organized into an *initialization phase* and an *iterative adaptive assessment loop*.

During initialization, three categories of inputs are provided to the system. The Quiz Sources component provides the pool of assessment items, either retrieved from an existing item bank or generated together with associated metadata through the LLM-assisted authoring interface. The Knowledge Model encodes the domain structure as a topic prerequisite graph, capturing dependency relationships among the topics covered by the assessment. The User Configuration component collects learner-specific preferences and constraints (e.g., desired topic coverage or importance weights) that inform the personalization of the assessment path. These three inputs are jointly passed to the *Adaptive Selection Engine*, which constitutes the core of the framework.

Within the *Adaptive Selection Engine*, the IRT Model maintains and updates the estimate of the learner ability based on the current response pattern, and provides this estimate, together with item difficulty parameters, to the ASP Logic module. This module performs constraint-based reasoning over the topic prerequisite graph and the user configuration to select, among the candidate items, the next question that best matches the estimated ability while satisfying prerequisite ordering, topic balance, and personalization constraints. The selected item is then delivered to the learner as the Next Question.

Once the learner submits a Student Response, the answer is used to perform an *Ability Estimation Update*, which revises the IRT ability estimate accordingly. The updated estimate is fed back into the IRT Model, closing the loop and enabling the engine to select the subsequent item adaptively. This cyclical process (item selection, response collection, and ability re-estimation) continues until a predefined stopping criterion is met, allowing the framework to jointly pursue measurement precision, through the statistical foundation of IRT, and pedagogical coherence, through the declarative constraints of ASP.

3.2. System Details and Proposed Methodology

3.2.1. Quiz Item Sources

Quiz content can be obtained from two complementary sources. The first is an existing item bank, populated in this work with publicly available Mathematics, Italian, and English items developed by the Italian National Institute for the Evaluation of the Education System (INVALSI) and collected from the GestInv repository [16]. Each item is associated with a curricular topic and, when available, an empirically calibrated IRT difficulty parameter b (Section 2.2), so these items can be used directly by the adaptive engine without an additional difficulty elicitation step. The second source consists of items authored by instructors through a graphical interface. The instructor provides the item content and assigns it to a topic, optionally receiving LLM-based topic suggestions. For manually authored items, the difficulty is specified through five qualitative levels: *Very Easy*, *Easy*, *Medium*, *Hard*, and *Very Hard*, which are mapped to the IRT difficulty parameter b as

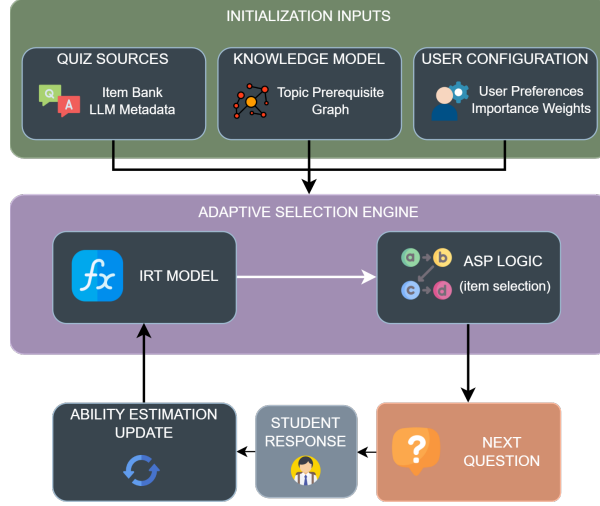


Figure 1: Framework Architecture. An initialization phase is followed by an iterative assessment loop, where the Adaptive Selection Engine combines items, domain structure, and learner preferences, and the ASP Logic module selects each question based on pedagogical constraints and IRT-based ability updated after every response.

$$b(l) = b_{\min} + \frac{l-1}{4}(b_{\max} - b_{\min}), \quad (6)$$

where $l \in \{1, \dots, 5\}$ denotes the difficulty level and $b_{\min}$ and $b_{\max}$ are the minimum and maximum difficulty values of b . The mapping provides equally spaced values on the IRT scale and represents a preliminary design choice to be validated empirically.

3.2.2. Modeling Topic Prerequisite Relationships

We now introduce our proposed *Prerequisite (weighted) Topic Graph*, a structured representation of topic dependencies that captures prerequisite relationships and their relative influence within the learning process. Building on the prerequisite relationships introduced in Section 2.2, the graph models the organization of the assessment domain by associating topics with dependency relations weighted as percentages (Figure 2). While T-SKIRT models dependencies among concepts through a simple prerequisite concept graph, in our framework we introduce the notion of *topic* as the educational unit associated with assessment items and organize topics through a weighted dependency structure. In our setting, a topic represents a broader educational area to which one or more questions refer, rather than the content of a single item. Examples of topics include “Geography” and “Cultural Heritage”.

This Topic Graph is represented as a directed weighted graph $G = (T, E, W)$, where T is the set of topics, $E \subseteq T \times T$ is the set of prerequisite relations, and W is the set of associated dependency weights. An edge $(p, t) \in E$ indicates that topic p is a prerequisite for topic t . Each edge is associated with an instructor-defined weight $w(p, t) \in [0, 1]$, representing the relative influence of prerequisite topic p on the acquisition of topic t . For each topic t , the incoming weights are normalized as:

$$\sum_{p:(p,t) \in E} w(p, t) = 1,$$

so that the set of prerequisites contributing to a topic defines a normalized distribution of dependencies. Therefore, stronger prerequisite relations have a larger influence in the knowledge estimation and reasoning processes. Note that, for readability, Figure 2 reports dependency weights as percentages (i.e., $w(p, t) \times 100$). The percentage-based representation is also the one adopted in the ASP program described in Section 3.3, since ASP does not support real-valued quantities.

In addition to prerequisite dependencies, the instructor may assign an *importance value* to each topic to support topic coverage during adaptive item selection. For manually authored items, an LLM can

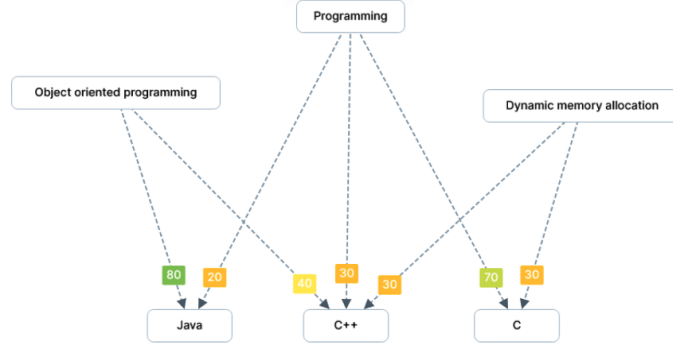


Figure 2: Example of a topic graph, showing topics connected by weighted dependency relations.

suggest topics from the item content and refine dependency relationships among them.

Building upon the weighted Topic Graph and the T-SKIRT framework described in Section 2.2, we extend the model with a weighted knowledge propagation mechanism. Unlike T-SKIRT, which considers an unweighted prerequisite graph, our approach associates each relation with an instructor-defined weight representing the dependency strength between topics. Therefore, the regularization term in Equation (5) is extended by weighting each prerequisite contribution accordingly.

Specifically, let $\hat{\theta}_t$ denote the topic-specific ability estimate independently obtained through the EAP procedure described in Section 2.2. We define the following weighted graph regularization objective:

$$\mathcal{J}(\boldsymbol{\theta}) = \lambda \sum_{p \in T} \theta_p^2 + \gamma \sum_{(p,t) \in E} w(p,t) (\theta_p - \theta_t)^2 + \sum_{p \in T} (\theta_p - \hat{\theta}_p)^2, \quad (7)$$

where $w(p,t) \in [0, 1]$ represents the dependency weight of the prerequisite relation (p,t) .

The first two terms play the same role as in Equation (5), now weighted by $w(p,t)$, while the third anchors each estimate to its independent EAP value. Its coefficient is fixed to 1 without loss of generality: multiplying $\mathcal{J}(\boldsymbol{\theta})$ by any constant does not change where its minimum lies. The regularized ability estimates $\boldsymbol{\theta}$ are obtained by minimizing $\mathcal{J}(\boldsymbol{\theta})$, approximated iteratively: at each iteration, every θ_p is recomputed simultaneously using the values of its neighbors from the previous iteration only, for a fixed number of iterations R (we set $R = 10$ in our implementation). The values of λ , γ , and R were set heuristically; their principled, data-driven calibration is left to future work.

3.2.3. User Preferences

User preferences represent learner-specific requirements that allow the adaptive assessment process to consider individual needs beyond the estimated knowledge level. In a school context, these preferences can capture accessibility requirements, learning constraints, or teacher-defined adaptations that make the assessment more suitable for a specific learner. Examples include limiting the maximum length of a question statement, restricting the acceptable difficulty range, or selecting preferred item formats. In the ASP formulation, these preferences are encoded as learner-specific facts and integrated into the reasoning process through declarative constraints, ensuring that only compatible items are considered during the selection phase.

3.2.4. LLM-Assisted Authoring

During the experiments, the application integrated OpenAI GPT-4.1 mini to support the authoring and initialization of the assessment knowledge model. The LLM was used to generate the set of topics and their dependency relationships, assign an importance value to each topic, and estimate the difficulty of the assessment items. These functionalities support the preparation of the content and metadata

Table 1

Input predicates of the ASP program, encoding learner information, item metadata, topic relations.

Name	Description
<code>user(User)</code>	Identifies the learner involved in the selection process
<code>user_pref(User, Preference, Value)</code>	Encodes preferences, such as difficulty range and question length limits
<code>topic(Topic)</code>	Represents the available topics
<code>topic_graph(Prerequisite, Topic, Weight)</code>	Represents weighted prerequisite relations between topics
<code>importance(Topic, Importance)</code>	Stores the instructor-defined importance of each topic
<code>question(Item, Grade, Subject, Topic, TextLength)</code>	Stores metadata associated with candidate items
<code>item(Item, Difficulty)</code>	Associates each item with its IRT difficulty parameter
<code>given_item(User, Item)</code>	Identifies items already administered to the learner
<code>mastery_threshold(Threshold)</code>	Defines the minimum mastery required for prerequisite topics
<code>mastery_high_threshold(Threshold)</code>	Defines the mastery level above which a topic is considered covered
<code>max_information_point(Value)</code>	Defines the reference value for maximum uncertainty in item selection

required by the framework, while the adaptive assessment process itself relies on the IRT and ASP components described above.

3.3. Adaptive Item Selection via Answer Set Programming

Item selection is encoded as an Answer Set Program combining the item bank, the topic graph, current learner ability estimates, and topic importance values into a single specification of the next item to administer. IRT-related quantities are obtained through external function calls, denoted by the `&` prefix, allowing the ASP program to interact with the Python-based ability estimation module. Since ASP requires integer values, all real-valued quantities are discretized by multiplying by 100 and truncating before being asserted or computed; external atoms already return values in this discretized form. The ASP program relies on the predicates reported in Table 1 to encode the domain knowledge and learner information provided as input facts. The program consists of the rules described below.

```

1 1 <= { best_item(User,I) : item(I,_) } <= 1 :- user(User).
2 :- best_item(User,I), given_item(User,I).
3 correct_answer_prob(User,I,T,P) :- &calculateProbabilityASP(User,T,B;P), item(I,B), user(User),
   question(I,_,_,T,_).
4 new_theta(User,I,T,"correct",Theta) :- &newThetaASP(User,T,B,1;Theta), user(User), item(I,B),
   question(I,_,_,T,_).
5 new_theta(User,I,T,"incorrect",Theta) :- &newThetaASP(User,T,B,0;Theta), user(User), item(I,B),
   question(I,_,_,T,_).
6 theta_update_gain(User,I,T,G) :- new_theta(User,I,T,"correct",ThetaC),
   new_theta(User,I,T,"incorrect",ThetaI), G = ThetaC-ThetaI.
7 topic_mastery(T,P) :- user(User), &calculateProbabilityASP(User,T,0;P), topic(T).
8 blocked_topic(T) :- topic_graph(P,T,_), topic_mastery(P,ProbP), mastery_threshold(Th), ProbP < Th.
9 sufficiently_covered(T) :- topic_mastery(T,Prob), mastery_high_threshold(Th), Prob >= Th.
10 :- best_item(User,I), question(I,_,_,T,_), blocked_topic(T).
11 :- best_item(User,I), question(I,_,_,L), user_pref(User,"max_length",LMax), L > LMax.
12 :- best_item(User,I), item(I,B), user_pref(User,"min_difficulty",Min), B < Min.
13 :- best_item(User,I), item(I,B), user_pref(User,"max_difficulty",Max), B > Max.
14 prop(X,Y,W) :- topic_graph(X,Y,W).
15 prop(X,Y,W) :- topic_graph(X,Z,W1), topic_graph(Z,Y,W2), W = (W1*W2)/100.
16 priority(T,S) :- prop(T,Target,W), importance(Target,Imp), S = W*Imp.
17 :~ best_item(User,I), question(I,_,_,T,_), sufficiently_covered(T). [1@4]
18 :~ best_item(User,I), question(I,_,_,T,_), priority(T,S). [-S@3]
19 :~ best_item(User,I), theta_update_gain(User,I,T,G). [-G@2]
20 info_dist(User,I,D) :- correct_answer_prob(User,I,_,P), max_information_point(P0), D = (P-P0)*(P-P0).
21 :~ Dmin != #min { D : info_dist(User,I,D) }, best_item(User,I), info_dist(User,I,Dmin). [1@1]

```

Rule 1 is a choice rule that selects exactly one item for each user from the available items, while Rule 2 is an integrity constraint that prevents the selection of already administered items.

Rules 3–5 define IRT-related quantities through the external atoms mechanism of DLV2 (see Section 2). Rule 3 computes the probability of a correct response for each candidate item according to the current topic-specific ability estimate. Rules 4 and 5 estimate the updated ability after a hypothetical correct or incorrect response allowing the reasoning module to evaluate the expected impact of each item.

Rule 6 computes `theta_update_gain`, which measures the difference between the two possible updated ability estimates. A larger gain indicates a greater potential refinement of the learner model.

Rule 7 estimates current topic mastery by evaluating the Rasch probability model at a neutral item

difficulty ($b = 0$). Under the Rasch model, this reduces $P_i(\theta)$ to $1/(1 + e^{-\theta})$, which is the logistic function applied directly to the topic ability estimate and maps it to a value in $[0, 1]$. Thus, $P = 0.5$ corresponds to $\theta = 0$, which represents the mean ability level under the EAP prior (Section 2.2). This provides one possible operational definition of topic mastery, with the resulting probability serving as the mastery estimate and allowing mastery levels (e.g., “below 40%”) to be easily defined and interpreted.

Rule 8 states that a topic is blocked until its prerequisites are mastered, while Rule 9 identifies topics that have already been sufficiently covered according to predefined mastery thresholds.

Rules 10–13 define hard constraints that filter invalid candidates: Rule 10 excludes items associated with blocked topics, while Rules 11–13 enforce learner preferences on question length and difficulty.

Rules 14 and 15 compute direct and indirect prerequisite relations in the topic graph by propagating dependency weights through intermediate topics. Note that these weights are already discretized as integers in the range 0–100.

Rule 16 computes a priority score for each topic by combining propagated dependency weights with instructor-defined importance values (integers), allowing the selection process to consider both topic relevance and prerequisite roles.

Rules 17–21 are organized in two stages. Rules 17–18 determine *which topic* to consider next: Rule 17 penalizes topics already sufficiently covered, and Rule 18 favors items on higher priority topics, weighted by the priority score S . Rules 19–21 then determine *which item* to select: Rule 19 rewards items with larger expected ability variation, Rule 20 defines $\text{info_dist}/3$ as the squared distance between the correct-answer estimate and the maximum information point ($\text{max_information_point}/1$), and Rule 21 minimizes this distance. This ordering is a deliberate design choice: topic-level decisions are prioritized to address curriculum coverage and relevance before optimizing item-level informativeness.

4. Conclusions and Future Work

This paper presented an Adaptive Quiz Framework that integrates Item Response Theory (IRT) and Answer Set Programming (ASP) to address, in a unified manner, the two central requirements of adaptive assessment in e-learning: measurement accuracy and pedagogical coherence. While IRT provides a statistically grounded estimate of learner ability and drives the selection of items that maximize information gain, it offers limited support for expressing structured pedagogical requirements such as prerequisite ordering, topic balance, and learner-specific constraints. The proposed framework addresses this limitation by delegating such requirements to a declarative ASP encoding, which reasons jointly over the item bank, the topic prerequisite graph, and user preferences to select, at each step, the item that best reconciles statistical informativeness with pedagogical soundness. The resulting framework illustrates how adaptive modeling and declarative reasoning can be combined without compromising either component: IRT retains its statistical rigor in ability estimation, while ASP provides a flexible and extensible layer for expressing pedagogical knowledge. In addition, support for LLM-assisted authoring of topics, items, and difficulty estimates reduces the burden of populating the knowledge model and the item bank, making the framework easily applicable to new learning domains.

As future work, we plan to empirically validate the proposed framework through simulation studies and pilot deployments with real learners, in order to assess the impact of the weighted knowledge propagation mechanism on ability estimation accuracy, evaluate the computational scalability of the approach to large item banks, and calibrate the qualitative-to-quantitative difficulty mapping introduced in Equation (6). This evaluation will ideally include comparisons with appropriate baseline adaptive-assessment strategies and consider the application of the framework in real educational settings, with secondary schools representing a particularly suitable target context due to the diversity of subjects and knowledge areas typically covered at this educational level.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT and Claude solely for translation and grammatical/linguistic refinement. These tools did not contribute in any way to the formulation of the paper's content, therefore the authors take full responsibility for the publication's content.

References

- [1] D. J. WEISS, G. G. KINGSBURY, Application of computerized adaptive testing to educational problems, *Journal of Educational Measurement* 21 (1984) 361–375. doi:10.1111/j.1745-3984.1984.tb01040.x.
- [2] H. Wainer, N. Dorans, D. Eignor, R. Flaugher, B. Green, R. Mislevy, L. Steinburg, D. Thissen, *Computerized Adaptive Testing: A Primer*, Taylor & Francis, 2000. URL: <https://books.google.it/books?id=ui-RAgAAQBAJ>.
- [3] F. M. Lord, A Theory of Test Scores, number 7 in *Psychometric Monographs*, Psychometric Society, 1952.
- [4] G. Rasch, Probabilistic Models for Some Intelligence and Attainment Tests, *Studies in mathematical psychology*, Danmarks Paedagogiske Institut, 1960. URL: <https://books.google.it/books?id=aB9qLgEACAAJ>.
- [5] W. J. van der Linden, *Linear models for optimal test design*, Statistics for Social and Behavioral Sciences, Springer, Germany, 2005. doi:10.1007/0-387-29054-0.
- [6] C. Ekanadham, Y. Karklin, T-SKIRT: online estimation of student proficiency in an adaptive learning system, *CoRR* abs/1702.04282 (2017). URL: <http://arxiv.org/abs/1702.04282>. arXiv:1702.04282.
- [7] P. Chen, Y. Lu, V. W. Zheng, Y. Pian, Prerequisite-driven deep knowledge tracing, in: *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*, IEEE Computer Society, 2018, pp. 39–48. URL: <https://doi.org/10.1109/ICDM.2018.00019>. doi:10.1109/ICDM.2018.00019.
- [8] V. Lifschitz, *Answer Set Programming*, Springer, Cham, 2019. doi:10.1007/978-3-030-24658-7.
- [9] F. Calimeri, W. Faber, M. Gebser, G. Ianni, R. Kaminski, T. Krennwallner, N. Leone, M. Maratea, F. Ricca, Asp-core-2 input language format, *Theory and Practice of Logic Programming* 20 (2019) 1–16. doi:10.1017/S1471068419000450.
- [10] G. Brewka, T. Eiter, M. Truszczynski, Answer set programming at a glance, *Commun. ACM* 54 (2011) 92–103. doi:10.1145/2043174.2043195.
- [11] M. Gelfond, V. Lifschitz, The stable model semantics for logic programming (1988). URL: <https://api.semanticscholar.org/CorpusID:261517573>.
- [12] M. Alviano, F. Calimeri, C. Dodaro, D. Fuscà, N. Leone, S. Perri, F. Ricca, P. Veltri, J. Zangari, The asp system dlvs, in: *International Conference on Logic Programming and Non-Monotonic Reasoning*, 2017. URL: <https://api.semanticscholar.org/CorpusID:38430482>.
- [13] F. Calimeri, D. Fuscà, S. Perri, J. Zangari, External computations and interoperability in the new dlvs grounder, in: *International Conference of the Italian Association for Artificial Intelligence*, 2017. URL: <https://api.semanticscholar.org/CorpusID:12374568>.
- [14] Y. Chen, X. Li, J. Liu, Z. Ying, Item response theory-a statistical framework for educational and psychological measurement, *Statistical Science* 40 (2025) 167 – 194. URL: <https://imstat.org/journals-and-publications/statistical-science/>. doi:10.1214/23-sts896, published online.
- [15] R. D. Bock, R. J. Mislevy, Adaptive eap estimation of ability in a microcomputer environment, *Applied Psychological Measurement* 6 (1982) 431 – 444. URL: <https://api.semanticscholar.org/CorpusID:122840004>.
- [16] GestInv, Gestinv – archivio delle prove invalsi, <https://www.gestinv.it>, 2026. Accessed: 2026-06-01.