Genetic Approach for Improved Prediction of Adaptive Learning Activities in Intelligent Tutoring System

Fatima-Zohra Hibbi[1], Otman Abdoun[2], El Khatir Haimoudi[3]
SMARTiLAB, EMSI Rabat, Morocco[1]
Advanced Science and Technologies Laboratory, Polydisciplinary Faculty,
Abdelmalek Essaadi University, Larache, Morocco[1, 2, 3]

***Abstract*—The intelligent tutoring system registers the reference data of the learners in a database. This data is stored for later use in the instructional module. Designing a student model is not an easy task. It is first necessary to identify the knowledge acquired by the learner, then identify the learner's level of understanding of the functionality and finally identify the pedagogical strategies used by the learner to solve a problem. These elements must be taken into account in the development of the learner model. Learner characteristics must be considered in several forms. To build an effective learner model, the system must take into consideration both static (Learner preferences) and dynamic (Compartmental action) student characteristics. The objective of the article is to work out the learner model of the intelligent tutoring system by suggesting a new learning path. This proposal is based on the constructivist approach and the activist style (based on experimentation). According to the KOLB model, the authors propose a list of pedagogical activities depending on the learners' profile. Based on the learners' actions, the system reduces the list of activities based on two criteria: the learner's preference and the presence of one or more activities based on the activist style using genetic algorithm as an evolutionary algorithm. The results obtained led us to improve the learning process through a new conception of the ITS learner model.**

***Keywords*—Intelligent tutoring system; learner model; genetic algorithm; adaptive learning activities**

I. Introduction

e-Learning systems include smart devices for analysing, evaluating and assessing users' knowledge and skills, as well as monitoring and supervising the e-learning process. AI allows using and implementing its techniques to be more efficient educational systems [1], such as genetic algorithms, intelligent tutoring systems and neural networks. However, the problem that online learning systems lack is to optimize the learner activities during a learning process. Meanwhile, various researchers have already addressed this topic by providing several solutions [2], such as eye-tracking technology [3] clustering [4] and classification methods [5] to detect the learner style, etc. These solutions allow the detection of the learning style and subsequently deduce the list of appropriate activities.

Through experiential learning theory, the KOLB model has been the theoretical foundation for several models of learning styles. Among these are the Dunn and Dunn [6] and Felder [7] models that developed the Learning Styles Inventory (LSI) to determine learning styles. This questionnaire consists of a self-evaluation questionnaire with approximately 100 questions that learners must answer. These questions are related to five categories. Several versions of this scale have been developed for adults. However, Felder and Silverman do not consider that all learners fit into predefined categories. For example, a learner may have the active-intuitive-global style; but have a strong preference for the active type, a weak preference for the intuitive and global brand and a moderate preference for the visual style. In this case, because he has a strong preference for the active type, he will likely have "great difficulty" learning in an environment that does not support this style. In the case of moderate visual style preference, he should learn "more easily" in a teaching environment that encourages visual style characteristics. For the intuitive and global styles, it is clear that the learner is well balanced on the scales of these two styles, so they would have no problem learning in an environment that favours one or both types simultaneously [8].

Categorizing learners according to their preferred learning style to match a set of complementary learning activities to each type is a promising idea. The preferred learner style affects the conception and construction of pedagogical activities as well as decision-making about the selection of learning resources [9]. From a pedagogical perspective, the teacher must be able to define a set of activities that correspond to each learning style, adapt the appropriate activity to an identified learning need, and design the most appropriate resources and activities for each individual according to their learning style. It should also be possible for a learner to examine the learning resources and activities associated with a particular learning session and assess whether they are appropriate for their preferred learning style.

The literature review allowed us to justify the uses of the KOLB model during this contribution. The main motivations for this choice are as follows:

A study of four learning style models and the experience of engineering educators in their practical applications are presented in [7].

The finding indicates that KOLB's model LIS helps learners learn the course because they have become aware of their thought processes and helps them develop interpersonal skills.

The objective of this paper is to update the learner model of the intelligent tutoring system in such a way that the activist style will be a required style for the learning process. For this purpose, the authors suggest optimizing the list of activities proposed by the system by strengthening the practice-based exercises to improve the skills acquired by the learner using the evolutionary algorithm: genetic algorithm.

The paper is organized as follow: the second section presents a literature review of the learner module and its triggered problems. The upcoming section describes the choice and the implementation of the genetic algorithm. The fourth section illustrates and interprets the obtained results. Finally, this article is concluded with a summary of the contribution and the future scope.

II. Learner Module

It would be difficult for an intelligent tutoring system to succeed without a good understanding of the learner. The learner model represents the learner's knowledge and skills dynamically. Just as domain knowledge must be represented explicitly to be communicated, the learner model must also be represented in the same way. In principle, the learner model should store aspects of the learner's behaviour and skills so that the ITS can infer the learner's performance and skills [9].

The intelligent tutoring system keeps the learners' data in a database. It stores learner reference data such as name, ID, current level, overall score, course exercises completed, exercises difficulty level achieved, and several questions completed from a course exercise. This data is stored for later use in other modules such as the pedagogical module.

Designing a student model is not an easy task. It must be based on answers to specific questions. What does the learner know? What kind of knowledge would the learner need to solve a problem? The methodology for designing a learner model should be based on those questions. First, it is necessary to identify the knowledge acquired by the learner in terms of the components integrated into the mechanism.

Secondly, it is necessary to identify the learner's level of understanding of the functionality of the mechanism. Finally, it is required to determine the pedagogical strategies used by the learner to solve a problem. Those elements must be taken into account in the development of the learner model [10].

Learner characteristics must be considered in several forms. The system must take into account both static and dynamic features of the learner to construct an effective learner model. Static features include information such as email address, age, and native language and are defined before the learning processes begin. However, dynamic features come from the learners' behaviour during interaction with the system [11]. The Fig. 1 shows the main components of the learner module:

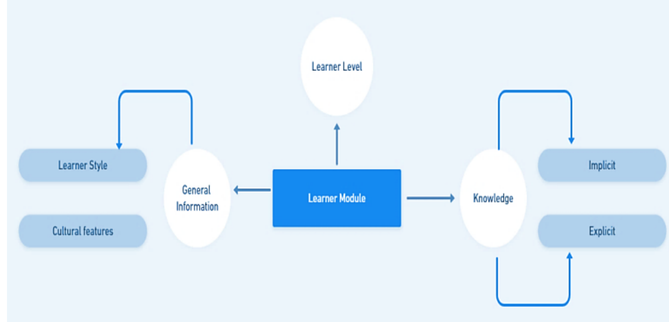

Fig. 1. Learner model component.

The analysis and study carried out in this section have enabled us to formulate various observations. The first one concerns the difficulties encountered by the learner module to extract implicit knowledge. For this, the authors have suggested a solution to convert tacit knowledge into explicit knowledge in the learner model of the intelligent tutoring system with the help of a competitive learning algorithm [12]. The second observation concerns the traces generated by the learners. The study conducted in this sense shows that the detection of learning styles is done automatically, without relying on explicit answers given to the questionnaire by the learners. Therefore, our contribution proposes an optimization of the activities suggested to the learner while keeping his preferred style and adding other parameters to achieve the required pedagogical objective.

III. Methodology

To reach our goal, we will apply evolutionary algorithms; and more precisely the genetic algorithm. Many researchers such as Goldberg, Davis and Michalewicz [13] have developed genetic algorithms (GA). GAs is certainly the best popular example of evolutionary algorithms [14][15][16][17]. The genetic algorithm is defined by a population cycle and involves three main factors: fitness, crossover, and mutation [3]. A population cycle represents the transition from one population to the next generation.

A. Level of Preference

In this subsection, we want to detect the learner's level of preference in each learning style. To do so, we rely on KOLB's model, which classifies the learner into four learning styles: Theorist, Activist, Reflective and Pragmatic. Each learner has a level of preference in a specific learning style. To do this, we prepare a list of activities Ai = {i=1....12} each learning style (LS) contains activities ranging from 1 to 4 (input) and, the output indicates the three levels of preferences which is the total of activities chosen by the learner multiplied by the level of preference. The figure below shows an example of the preference level corresponding to the two learning styles (Activist and Reflectors).

Concerning the Pragmatist learning style, it contains two activities; the corresponding preferences are described in the Fig. 2 to 4.

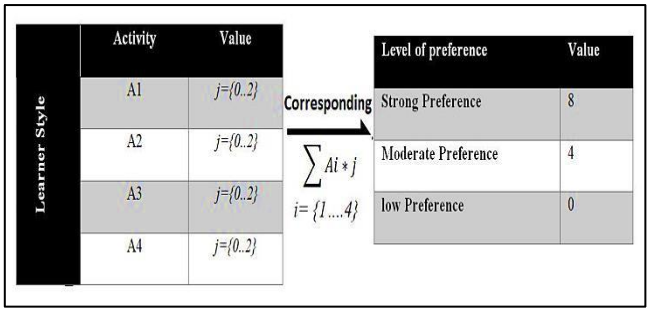

Fig. 2. Level of preference of Activist and Reflector's learner style.

Concerning the Pragmatist learning style; it contains two activities; the corresponding preference is described in the figure below:

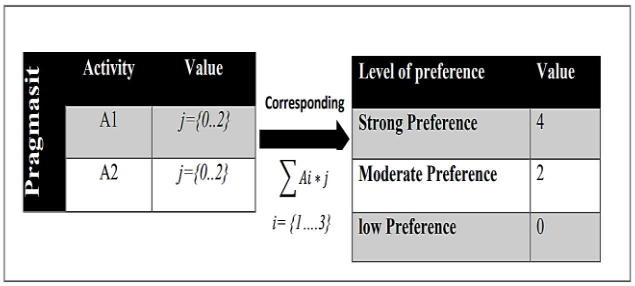

Fig. 3. Level of preference of the Pragmatist learner style.

The last learning style is theoretical; it contains three activities and three preferences levels.

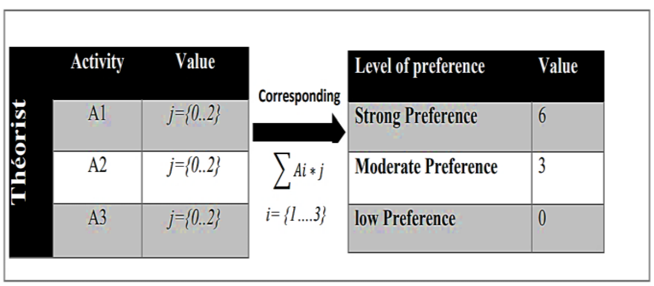

Fig. 4. Level of preference of the theorist learner style.

B. Description of the Genes

The proposed genetic algorithm detects the combination of actions that the student usually performs during a learning sequence using different learning activities. The production of the best-fit chromosome presents the combination of actions preferred by the student.

The first three genes of the chromosome represent the actions that correspond to the Theorist style defined by KOLB. The next four genes represent the actions corresponding to an activist style; the remaining four genes correspond to a reflector style, and the last two genes correspond to a pragmatist's type. With the values of these genes, it is possible to establish the mechanisms of each learning activity that characterize a learner and, consequently, the optimization of these activities and the deduction of the learning style corresponding to the analysed chromosome.

The values of the genes g1, g2 and g3, carry information about the learner's participation in the three activities. If the sum of the genes values is zero, it means that the learner does not participate in any of the three activities. On the other hand, if the sum of the codes for these genes is 6, it means that the learner has made extensive use of these activities and can be considered extremely theorist. If the addition of the two genes results are an intermediate value; the learner is moderately passive, neutral or moderately active, depending on the difference between the value obtained and the extremes. The Table I describes the genes:

TABLE I. DESCRIPTION OF THE GENES

Gene Number	Activities	Gene value	Description
Gene 1	Exercise of Analyzing	0	Does not make it
		1	Makes 50% of the proposed exercise
		2	Accomplish the activity
Gene 2	Exercise based on the extraction of conceptual model	0	Does not make it
		1	Makes 50% of the proposed exercise
		2	Accomplish the activity
Gene 3	Supervise a course (pdf, World, PPT)	0	Only reads the introduction
		1	Reads 50% of the course
		2	Accomplish the course
Gene 4	Collaborative learning	0	Does not participate
		1	Participate, but does not accomplish the task
		2	Accomplish the task
Gene 5	Activities based on games	0	Does not observe
		1	Makes 50% of the proposed activity
		2	Accomplish the activity
Gene 6	Simulation activities	0	Does not do it
		1	Makes 50% of the activity
		2	Accomplish the activity
Gene 7	Activities based on observation	0	Does not observe
		1	Makes 50% of the proposed activity
		2	Accomplish the activity
Gene 8	Access to a video course	0	Watch less than 10% of the video
		1	Watch 50% of the video
		2	finish the video
Gene 9	Production reports	0	Does not make it
		1	Realize 50% of the reports
		2	Accomplish the reports
Gene 10	Brain storming Activity	0	Does not make it
		1	Makes 50% of the activity
		2	Accomplish the activity
Gene 11	Realize the projects	0	Does not make it
		1	Realize 50% of the project
		2	Realize the project
Gene 12	Tutorials	0	Does not do it
		1	Makes 50% of the activity
		2	Accomplish the activity

C. Initial Population

An initial population is a group of chromosomes that represent possible solutions to the problem considered, in this case, as possible combinations of actions that a learner can perform. The size of the initial population and the level of diversity determine the quality of its coverage of the space of possible solutions. Based on the number of genes in the proposed chromosome structure and the number of possible values that each gene can have, it is possible to consider the complete solution space to be of the order of 3^{12}.

For the empirical evaluation of the approach, we used initial populations of two different sizes: 3000 chromosomes and 6000 chromosomes. These values were chosen because they are small compared to the whole space of possible solutions (represent 0.5% and 1% respectively); nevertheless, they allow obtaining a good level of diversity by the random generation of the initial population. The weight of initial population genes has a value of zero because the chromosomes have not yet been evaluated in terms of any academic unit performed by the learner.

D. Fitness Function

The fitness function evaluates each chromosome in a population and gives a score or fitness value to each chromosome based on their ability to solve the problem. In this problem, the fitness value of a chromosome is determined based on how close the chromosome's action combination is to the learner's preferred action combination. To evaluate a particular chromosome in a given population, the fitness function is based on the actions performed by the learner while participating in a specific learning activity.

Then, the fitness value of a chromosome C is defined as the product of the gene weights and represents the level of preference in the chromosome, as shown in Equations 1, 2 and 3. Based on our experience as trainers and teachers in the computer field, we notice that the application-based learning style is more profitable than theory and other types of learning.

$$\mathrm{Fitness(C)} = \sum_{j=1}^{12} \alpha * x \tag{1}$$

$$\text{With:}\quad \alpha = \begin{cases} -1 \text{ if}(j \in [1,3]) \\ 2 \text{ if } (j \in [4,6]) \\ 1 \text{ otherwhise} \end{cases} \tag{2}$$

$$\text{And: } x = \begin{cases} 0 \text{ if the activity is not achieved} \\ 1 \text{ if the half of the activity is achieved} \\ 2 \text{ if the activity is completly achieved} \end{cases} \tag{3}$$

E. Selection

Once chromosome fitness is calculated for a population, a selection method allows the algorithm to randomly select pairs of chromosomes for reproduction. In this contribution, we used three types of selection: uniform selection, roulette wheel selection and turn selection. Fig. 5 shows the example of the selected chromosome. The authors will detail every form in the experimental result section.

F. Crossover

The crossover is the result obtained when two chromosomes share their particularities. It allows the genetic mixing of the population and the application of the principle of heredity of Darwin's theory. In this paper, the authors applied a simple crossover which they subdivided the chromosome on ½. Fig. 6 shows the example of the results of chromosomes.

Weight	1	2	1	4	3	3	2	1	4	2	3	1
Code	2	1	0	2	2	1	0	2	1	1	1	2

Parent 1

Weight	1	2	1	4	3	3	2	1	4	2	3	1
Code	1	1	0	1	2	1	0	1	0	1	1	0

Parent 2

Fig. 5. Example of the selected chromosome (Parent1) & (Parent2).

Weight	2	1	4	2	3	1	1	2	1	4	3	3
Code	0	1	0	1	1	2	1	1	0	1	2	1

Descendants 1

Weight	1	2	1	4	3	3	2	1	4	2	3	1
Code	1	1	0	1	2	1	0	2	1	1	1	0

Descendants 2

Fig. 6. Example of the results chromosome (Descendants 1 and 2).

G. Mutation

The mutation is a genetic modification of the crossing chromosomes. The mutation is a method of introducing genetic diversity by changing the value or code of a randomly selected gene, as described in the Fig. 7. In this work, the authors decided to use the following function:

$$x_i = 2 - x_i \quad \text{With: } x_i = \{0,1,2\} \tag{4}$$

Weight	1	2	1	4	3	2	2	1	4	2	3	1
Code	1	1	0	1	2	1	0	1	0	1	1	0

Fig. 7. Example of a genetic mutation.

IV. Results and Discussion

In order to evaluate the proposed approach, we simulated the actions performed at first by 10 learners generated arbitrary to build artificial data for the experiment. For this task, we considered that each learner has a particular learning style, represented by a set of preferred actions and that they behave according to this style. The actions performed by a learner correspond to their learning style. The resulting data is represented as a two-dimensional matrix, with each column representing the learner's identifier and an array of one particular action from the actions mentioned in the section above with twelve values.

TABLE II. DATA OF THE PREFERRED ACTIVITIES

Preferred Activities												
Learner ID	***A1***	***A2***	***A3***	***A4***	***A5***	***A6***	***A7***	***A8***	***A9***	***A10***	***A11***	***A12***
1	0	1	2	2	1	2	1	2	0	2	1	2
2	2	2	0	1	0	1	1	0	0	0	0	0
3	0	2	0	1	0	1	1	2	2	2	2	0
4	2	0	2	0	2	2	0	2	2	1	2	0
5	0	1	1	0	1	2	1	1	0	0	0	0

The Table II illustrates the data from five learners selected randomly. The third row corresponds to the third learner, for example. This learner does not make the first activity. A2 indicates that he accomplishes the activity. The third activity marked that the learner read only the introduction, and he participates in the fourth activity, but he does not accomplish the task. A5 indicates that the learner does not observe the activity based on the game, but he makes 50% of the proposed activity in A6 and A7. The activities A8, A9, A10, and A11 indicate that the learner finishes the activities, but he does not do the last activity.

The Fig. 8 shows the evolution of the fitness function averaging 50 generations using the genetic algorithm uniform where the fitness value is 15. In the Mutation phase, if we change the value in the interval [4, 6], which has a coefficient (=2) that corresponds to an activist style, the fitness function decreases. On the other hand, if we change the value in the interval [1, 3], which has a coefficient (=-1) that corresponds to a theorist style, the fitness function progresses exponentially. This implies learning must tend towards the activist style.

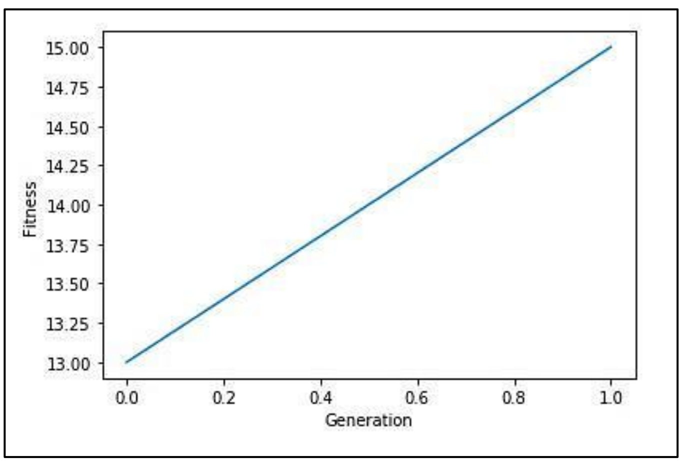

Fig. 8. Accuracy of genetic algorithm uniform.

Fig. 9 and 10 are the developed version of our proposed approach using the selection methods wheel roulette and tournament. We notice that the implementation of these methods improves the fitness function, which allows us to have good results. The fitness value is 21. The results show that the best selected population implies the improvement of the fitness function.

As long as the value of the fitness function is high, the learning tends towards the activist style. This entails the ignorance of theory-based activities and pragmatic activities, which makes it possible to optimize the list of non-selected learning activities; and ultimately, improving learning process.

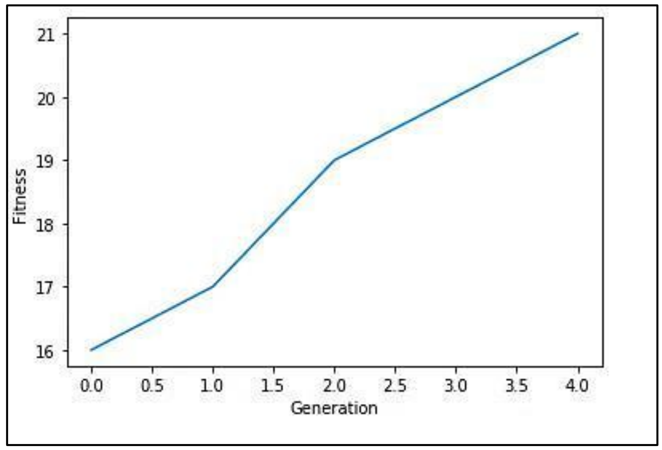

Fig. 9. Accuracy of genetic algorithm using wheel roulette selection.

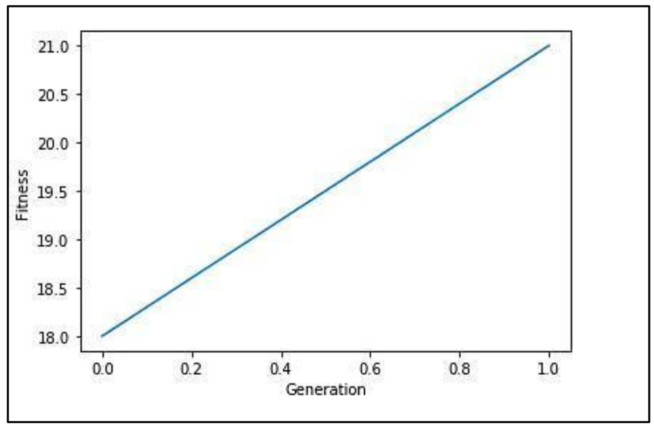

Fig. 10. Accuracy of genetic algorithm using tournament selection.

Table III shows that the 'Tournament' selection method performed better with the mean difference on the test set for the three data sets used compared to the 'Wheel roulette' selection method and the Uniform GA. Fig. 11 shows the variation of the selection methods in GA

TABLE III. COMPARATIVE STUDY BETWEEN THREE VERSIONS OF GA SELECTED METHODS

Comparisons	Mean Diff.	95.00% CI of diff.	Summary	Adjusted P Value
GA Uniform vs. GA-Wheel roulette	-4.000	-5.757 to -2.243	**	0.0017
GA Uniform vs. GA-Tournament selection	-5.500	-6.056 to -4.944	****	<0.0001
GA-Wheel roulette vs. GA-Tournament selection	-1.500	-2.812 to -0.1883	*	0.0308

From the previous researchers [18] and the results as in Fig. 12, the authors see that the tournament selection method (TSM) provides a high fitness value from the first generation compared to GA uniform (GA-U) and wheel roulette selection technique (WRSM). Nevertheless, both methods: TSM and WRSM, show good results at the end of the generation, fitness function =21 as shown in Fig. 12.

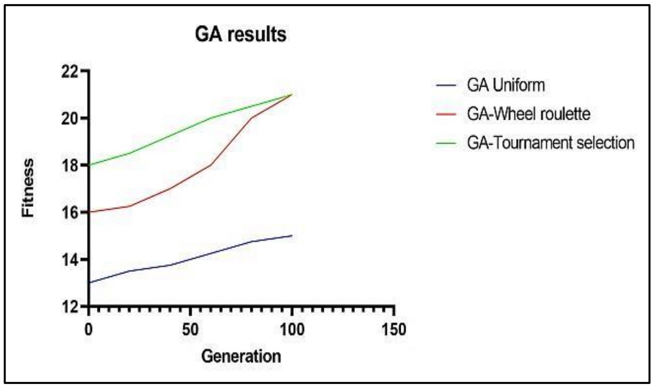

Fig. 11. Variation of the selection methods in GA.

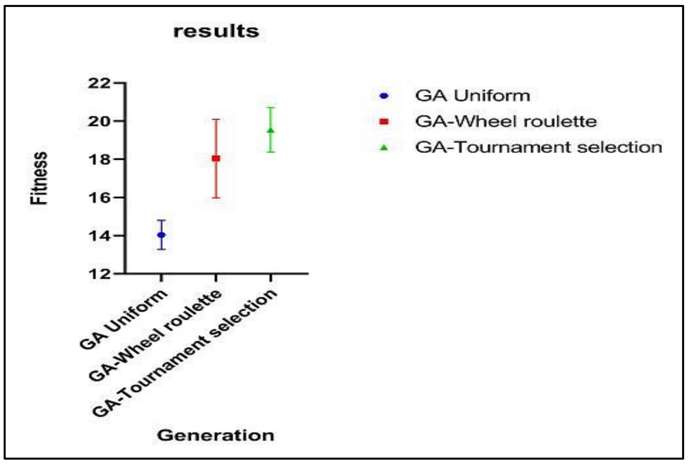

Fig. 12. Comparative result of three version of GA.

The recognition of learning style in an intelligent tutoring system increases the learning effectiveness. However, the adjustment of the content according to learner preferred style does not lead us to have achievable results. From the authors experience in the field of teaching, we recommend using the activist type and respecting the preferences of the learner.

According to the results obtained, we notice that the learners who use the practical activities acquire the expected objective rapidly. For this reason, we have developed and adapted the genetic algorithm by changing in the phase of mutation the activities based on the theory by low values x_i by respecting the following formula: $2\text{-}x_i$.

These results allow us to have an exponential function and converge to practice-based learning. These results urge the learner to perform well the case studies suggested by the intelligent tutoring system.

V. Conclusion

The e-learning system includes intelligent tools for the analysis, evaluation and assessment of the user's knowledge and skills, as well as for the monitoring and supervision of the e-learning process.

The study on data-driven approaches and more specifically on the traces generated by the learners shows that the detection of learning styles is done automatically, without relying on explicit answers given to the questionnaire by the learners. Therefore, our contribution provides a path to be generated from the resource data (learner preference and activist style) towards the desired goal: acquiring the requested skills regarding the pedagogical objectives using genetic algorithms.

This contribution was developed to enhance the learning process of a programming language by presenting a new method for reducing the list of activities proposed to the learner. The authors are convinced that this contribution will lead to a good performance in building the learner model of the intelligent tutoring system. The implementation of these findings in the STS-programming solution is envisaged as future work.

References

[1] J. Ong, S. Ramachandran. “Intelligent tutoring systems: Using AI to improve training performance and ROI,” Stottler Henke Associates, Inc. 2003.

[2] Asselman, A., Khaldi, M. and Aammou, S. Evaluating the impact of prior required scaffolding items on the improvement of student performance prediction. Educ Inf Technol 25, 3227–3249. https://doi.org/10.1007/s10639-019-10077-3, 2020.

[3] Fatima-Zohra Hibbi, Otman Abdoun and El Khatir Haimoudi. “Exploration of Analytical Mechanisms in the Feedback model”, Procedia Computer Science, Volume 148, Pages 201-207, ISSN 1877-0509, 2019.

[4] Hibbi, F.-Z., Abdoun, O., and Haimoudi, E. K. “Smart Tutoring System: A Predictive Personalized Feedback in a Pedagogical Sequence”. International Journal of Emerging Technologies in Learning (iJET), 16(20), pp. 263–268. https://doi.org/10.3991/ijet.v16i20.24783, 2021

[5] Hmedna, B., El Mezouary, A. and Baz, O. “A predictive model for the identification of learning styles in MOOC environments”. Cluster Comput 23, 1303–1328, https://doi.org/10.1007/s10586-019-02992-4, 2020.

[6] Dunn, Rita, et al. “A Meta-Analytic Validation of the Dunn and Dunn Model of Learning-Style Preferences.” The Journal of Educational Research, vol. 88, no. 6, Taylor & Francis, Ltd, pp. 353–62, http://www.jstor.org/stable/27541998, 1995

[7] Felder, Richard. “Learning and Teaching Styles in Engineering Education”, Journal of Engineering Education, vol. 78(7), 674–681, 1988.

[8] Sabine Graf, Tzu-Chien Liu, Kinshuk, Nian-Shing Chen, Stephen J.H. Yang. “Learning styles and cognitive traits – Their relationship and its benefits in web-based educational systems”, Computers in Human Behavior, Volume 25, Issue 6, Pages 1280-1289, ISSN 0747-5632, 2009.

[9] K. Chrysafiadi and M. Virvou. “Student modelling approaches: A literature review for the last decade,” Expert Systems with Applications 40 4715–4729. 2013.

[10] E. Sierra, R. García-Martínez, Z. Cataldi, P. Britos, and A. Hossian, “Towards a methodology for the design of intelligent tutoring systems,” Research in Computing Science Journal, vol. 20, pp. 181–189, 2006.

[11] Fatima-Zohra Hibbi, Otman Abdoun and El Khatir Haimoudi, “Bayesian Network Modelling for Improved Knowledge Management of the Expert Model in the Intelligent Tutoring System” International Journal of Advanced Computer Science and Applications(IJACSA), 13(6), 2022.

[12] Hibbi, F., Abdoun, O., and El Khatir, H. “Extract Tacit Knowledge in the Learner Model of the Smart Tutoring System. International Journal Of Emerging Technologie in Learning (IJET), 15(04), PP.23-240. Doi: http://dx.doi.org/10.3991/ijet.v15i04.11781”, 2020

[13] Goldberg, David Edward. “Genetic algorithms in search, optimization, and machine learning”, Addison-Wesley, 412 p, 1989.

[14] Bck T. Evolutionary Algorithms in Theory and Practice. Oxford University Press, New York, 1996.

[15] J. Dro, A. Ptrowski, P. Siarry, and E. Taillard. “Mta-heuristiques pour l’optimisation difficile ”. Eyrolles, 2003.

[16] O. Abdoun and J. Abouchabaka. “A Comparative Study of Adaptive Crossover Operators for Genetic Algorithms to Resolve the Travelling Salesman Problem”. International Journal of Computer Applications, Vol.31, N.11, 2012.

[17] Yassine Moumen, Otman Abdoun, and Ali Daanoun. “Parallel approach for genetic algorithm to solve the Asymmetric Traveling Salesman Problems”. In Proceedings of the 2nd International Conference on Computing and Wireless Communication Systems. ACM, 2017.

[18] Jinghui Zhong , Xiaomin Hu , Min Gu and Jun Zhang, (2005). “Comparison of Performance between Different Selection Strategies on Simple Genetic Algorithms”, Proceedings of the International Conference on CIMCA-IAWTIC’05.