Learning Path Model based on Revised Bloom's Taxonomy and Domain Ontologies using Discrete Particle Swarm Optimization

Eko Subiyantoro
Department of Information Technology
PPPPTK BOE VEDC Malang,
Indonesia

Ahmad Ashari
Department of Computer Sciences and Electronics,
Faculty of Mathematics and Natural Sceince,
Universitas Gadjah Mada Indonesia

Suprapto
Department of Computer Sciences and Electronics,
Faculty of Mathematics and Natural Sceince,
Universitas Gadjah Mada Indonesia

***Abstract*— Revised Bloom's Taxonomy (RBT) is present mostly to respond to the demand for future urgencies of the growing education community. RBT provides options for each particular student to develop one's progress and study. It also helps a teacher to prepare appropriate Learning Object (LO). In an unfortunate, common learning process doesn't provide various learning objects with suitable learning paths to comply with students' diverse cognitive abilities. The purpose of this study is to determine learning path recommendations based on Revised Bloom's Taxonomy and ontology learning object using Discrete Particle Swarm Optimization (DPSO). Experimental studies illustrated that the proposed DPSO algorithm can be used to determine the learning path that is in accordance with the cognitive abilities of students through the assessment of the quality of connections between RBT and LO ontology of a subject. The average similarity of learning paths for Course Prerequisites (CP 1, CP 2, CP 3) based on the number of particles was 85.5%.**

Keywords*— *RBT, learning object, ontology, learning path, DPSO

I. INTRODUCTION

Teachers are expected to apply the cognitive Bloom Taxonomy which was revised by Krathwohl [1] in 2002, namely (C1) remember, (C2) understand, (C3) apply, (C4) analyze (analysis), (C5) evaluate (evaluate), and (C6) create (create) during the learning process. These six levels are a series of levels of human thinking. These levels consecutively classify thinking to remember at the lowest level while the highest is to create.

Higher Order Thinking Skills (HOTS) [2][3] is a student thinking activity that involves a high level of cognitive level from Bloom's taxonomy of thinking including (C4) analysing, (C5) evaluating and (C6) creating [4]. HOTS activities sharpen students' skills in seeking knowledge in inductive and deductive reasoning to think of answers or identify and explore scientific examinations of existing facts [5]. Students can process information and make the right and fast decisions in the present. Students need to develop logical thinking and reasoning based on facts.

Education field uses ontology methodology to create conceptual structures of various knowledge domains. Ontology methodology makes semantic relationships among various knowledge concepts. It shows prerequisite relationships, the composition of relationships, etc[6]. LO is a pedagogical tool to help students obtain the concept of learning [7]. Based on this explanation, the ontology approach is applicable to develop LO during the learning process.

Curriculum sequencing (CS) is a technique to provide students in planning the most appropriate sequence of learning tasks individually [8]. CS not only helps students determine the most appropriate learning path but also enable teachers to organize program structure, create content or learning object, and make improvement [9]. The purpose of CS is to replace the structure of rigid, general learning methods, and one suitable model set by the teacher or pedagogical team becomes a more flexible and personalized learning path. So that individualization of teaching materials is challenged in choosing the right LO and making LO sequences that are easy to learn [10]. This suitability of learning paths and students' cognitive abilities will produce an optimal result

Many studies in the CS domain had already applied evolution algorithm (EA) approach include using genetic algorithms, namely pedagogic sequence determination through approaches to matching keywords and difficulty levels [11], pedagogic sequence determination by minimizing the average difference between the level of compatibility of learning objects and participant satisfaction level [12], and pedagogical sequence genetic algorithms through calculating distance in LO [13]. Contrast to the EA method, the swarm intelligence approach emphasizes more on cooperation than competition [14]. In supporting cooperation concept, each agent has equipped with a simple ability to learn from experiences and communicate with fellow agents. The metaheuristic method based on the swarm intelligence concept is Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO).

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 08, August-2019

This study proposes an individual learning model that automatically determines a learning path that best fits students' cognitive abilities based on *Revised Bloom's Taxonomy* using Discrete Particle Swarm Optimization (DPSO). Determination of learning paths that are in accordance with the cognitive abilities of students through optimization of the assessment of the relationship of LO between RBT and the ontology of a subject.

II. FEATURE OPTIMIZATION

A. Particle Swarm Optimization (PSO)

Inspired by bird group social behavior, Dr. Eberhart and Dr. Kennedy developed Particle Swarm Optimization (PSO) is a population-based stochastic optimization technique in 1995[14]. The PSO algorithm works based on particles in the population that work together to solve existing problems disregarding the physical position [15][16]. The PSO algorithm combines local and global search methods that balance exploration (ability to conduct investigations in different areas of the search area to get the best optimal value) and exploitation (ability to concentrate around the search area for fix solution).

The similarity of PSO and GA is that the system starts with a population formed from random solutions, then the system seeks optimization through random generation changes. Each particle holds traces of position in the search space as the interpretation of the best solution (*fitness*) that had been achieved.

There are three stages in the basic algorithm of PSO, namely generation of position and velocity of particles, velocity updates and position updates. First Step, position x_i^t and velocity v_i^t from a collection of particles randomly generated using the upper limit ($xmax$) and the lower limit ($xmin$) of the variable design shown in (1) and (2),

$$x_0^t = x_{min} + rand(x_{max} - x_{min}) \tag{1}$$

$$v_0^t = x_{min} + rand(x_{max} - x_{min}) \tag{2}$$

The second step is to update the latest speed (v_{i+1}) on each particle at time t + 1 based on the previous speed (v_i) and the two best positions that have been searched (P_{best} and G_{best}). The update velocity formulation includes several random parameters, inertia factor (w), self-confidence c_1), swarm confidence (c_2) shown in (3),

$$v_{i,j}^{t+1} = wv_{i,j}^t + c_1r1(Pbest_{i,j}^t - x_{i,j}^t) + c_2r2(Gbest_{g,j}^t - x_{i,j}^t) \tag{3}$$

The third step is to update the particle position (x_i^{t+1}) based on its velocity (v_i^{t+1}). The alteration of particle position is hoped to gain optimal solution. The update of the particle position is shown in (4),

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{4}$$

B. Discrete Particle Swarm Optimization (DPSO)

In 2000, Clerc modified the PSO algorithm which was formulated by Kennedy and Eberhart [18]. Clerc modified the representation of the position of the particles, the shape of the velocity produced by the particles and the effect of velocity on the position of the particles. The expectation of these modifications is to be applied to problems with discrete models especially combinatorial types [19]

$$v_i^{t+1} = c_1 v_i^t \oplus c_2\left(\left(Pbest_i^t + \frac{1}{2}\left(Gbest_g^t - Pbest_i^t\right)\right) - x_i^t\right) \tag{5}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{6}$$

The framework of PSO for discrete optimization problems proposed by Goldbarg et al.[20][21]is shown in figure 2. In this proposal (3) is replaced by (5), the coefficients c_1 and c_2 have the same meaning stated previously and the signal $\oplus$ represents a composition.

In initial applications of the proposed approach, only one of the three primitive moves is associated with each particle of the swarm at each iteration step. Thus, c_1, c_2 ∈ {0,1} and $c_1 + c_2 = 1$ in (5).The assignment is done randomly. Initial probabilities are associated with each possible move and, during the execution, these probabilities are updated. Initially, a high value is set to pr_1, the probability of particle i to follow its own way, a lower value is set to pr_2, the probability of particle i goes towards P_{best} and the lowest value is associated with the third option, to go towards G_{best}. The algorithm utilizes the concept of social neighborhood and the G_{best} of all particles is associated with the best current solution, G_{best}. The initial values set to pr_1, pr_2, and pr_3 are 0.9, 0.05 and 0.05, respectively. As the algorithm runs, pr_1 is decreased and the other probabilities are increased. At the final iterations, the highest value is associated with the option of going towards G_{best} and the lowest probability is associated with the first move option.

```
Procedure Discrete_PSO
/* Define initial probabilities for particles' moves:*/
pr1 ← a1 /*to follow its own way*/
pr2 ← a2 /*to go towards Pbest*/
pr3 ← a3  /*to go towards Gbest*/
/* a1+ a2+ a3=1 */
Initializa the population of particles
do
    for each particle i
        value_i ← Evaluate(x_i)
        if f(value(x_i)) < f(value(Pbest_i) then
            Pbest_i ← x_i
        if f(value(x_i)) < f(value(Pbest_i) then
            Gbest_i ← x_i
    end
    for each particle i
        velocity_i ← define_velocity(pr_1,pr_2,pr_3)
        x_i ← update(x_i, velocity_i)
    end
  /* Update probabilities*/
  pr_1 = pr_1 × 0.95;
  pr_2 = pr_2 × 1.01;
            pr_3 = 1 − (pr_1 + pr_1);
while ( a stop creterion is not satisfied )
```

Figure. 1 Pseudo-code of DPSO

III. RESEARCH METHODS

There are three steps in this research; the first is the analysis of research architecture, the second is the development of learning objects (LO) based on RBT and ontology, and the third is the use of the DPSO algorithm in this study.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 08, August-2019

A. Research Architecture

The model used in this study consists of three components, learning object ontology based on RBT, course prerequisites ,and discrete particle swarm optimization. The general architecture of the proposed model can be seen in Figure 2.

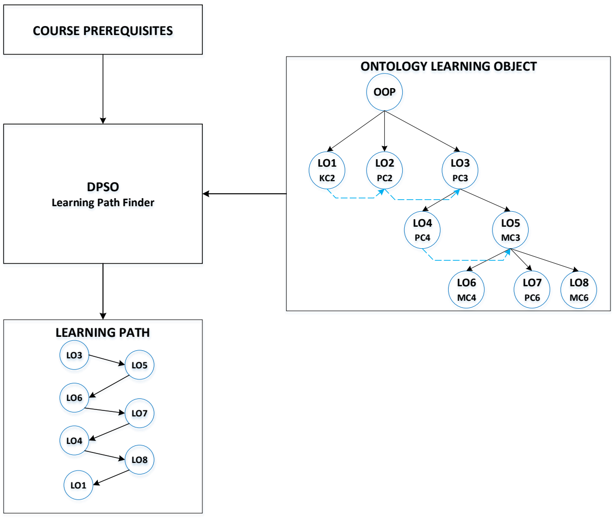


Fig. 2 The architecture of the proposed model

The Discrete Particle Swarm Optimization algorithm is applied to overcome combinatorial problems more practically and regularly in determining learning paths. Determination of learning object sequences through a LO ontology based on initial requirements or Course Prerequisites (CP) and using RBT to assess the quality of connections. The expected final result is that each student gets a recommendation for a learning path that is in accordance with his cognitive level.

B. Learning Object Mapping with RBT

Learning activities often involve both lower order and higher order thinking abilities that include ways of thinking concrete and abstract knowledge. The dimensions of cognitive processes are a continuum in increasing cognitive complexity from low-level thinking skills to higher thinking skills. According to Krathwohl[1], in identifying nineteen specific cognitive processes to clarify the scope of six classification categories. Concept map of analyzing the depth and breadth of learning objectives is shown in Table 1.

TABLE 1. ANALYSIS OF THE DEPTH AND BREADTH OF DETERMINING LEARNING OBJECTS

KNOWLEDGE DIMENSIONS		BREADTH					
		Remember (C1)	Understand (C2)	Apply (C3)	Analyze (C4)	Evaluate (C5)	Create (C6)
DEPTH	Factual	FC1	FC2	FC3	FC4	FC5	FC6
	Conceptual	KC1	LO1	KC3	KC4	KC5	KC6
	Procedural	PC1	LO2	LO3	LO4	PC5	LO7
	Metacognitive	MC1	MC2	LO5	LO6	MC5	LO8

Basic competency is the ability and least learning material that must achieved by students for a subject in each education unit that refers to core competencies. Table 2 presents the relationship between basic competencies with the learning objects in determining competency targets.

TABLE 2. METADATA LEARNING OBJECT

Basic Competency	Learning Object	Competency Target	Position
3.1	Object Oriented Methodology	KC2	(2,2)
3.2	The Basic and Rules in Object Oriented Programming	PC2	(2,3)
3.3	Class and Object	PC3	(3,3)
3.4	Data Encapsulation and Information	PC4	(3,4)
3.5	Inheritance	MC3	(4,3)
3.6	Polymorphism	MC4	(4,4)
3.7	Interface	PC6	(3,6)
3.8	Package	MC6	(4,6)

C. Ontology Learning Object

The ontology of learning objects developed in this study refers to the first semester XI object-oriented programming subjects in software engineering expertise programs at Vocational High Schools (SMK).

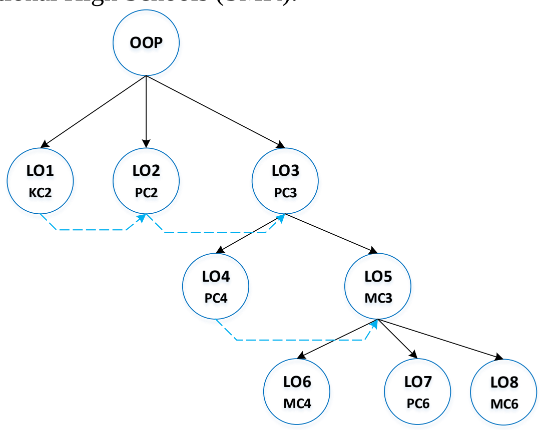


Fig. 3 Ontology learning object with RBT

The distance values in the ontology are: the value of LO parent connected to its LO below it has a value of 1. Subjects distanced more than three levels is declared to have no connection. For those LO that are not connected to each other directly is valued 0.5. Table 3 presents the distance calculation data between LO in the ontology.

TABLE 3. THE DISTANCE VALUE OF EACH LO IN ONTOLOGY

LO	1	2	3	4	5	6	7	8
1	0	0.5	1	2	2	3	3	3
2	0.5	0	0.5	1.5	2	2.5	2.5	2.5
3	0.5	0.5	0	1	1	2	2	2
4	2	1.5	1	0	1.5	1.5	1.5	1.5
5	2	2	1	0.5	0	1	1	1
6	3	2.5	2	1.5	1	0	0.5	1.5
7	3	2.5	2	1.5	1	0.5	0	2
8	3	2.5	2	1.5	1	1.5	2	0

D. The Proposed DPSO Algortihm

The application of the DPSO algorithm in this study, starting with the LO particle representation, updating the velocity and position of the particles by transposition,

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 08, August-2019

calculating the fitness function based on the relationship between RBT and ontology, and finally writing the DPSO algorithm to solve this problem.

1. Particle Representation

The particle representation in this combinatorial problem is to change the arrangement of the positions of each permutation value into an integer form from the solution representation. The solution of the combinatorial problem optimization case is to change the position arrangement of each permutation value into an integer form from the representation of the solution. The Discrete Particle Swarm Optimization algorithm is applicable to overcome combinatorial problems more practically and regularly because there are very structured search and evaluation mechanisms. Figure 4 shows the learning object sequence randomly from three groups of particles.

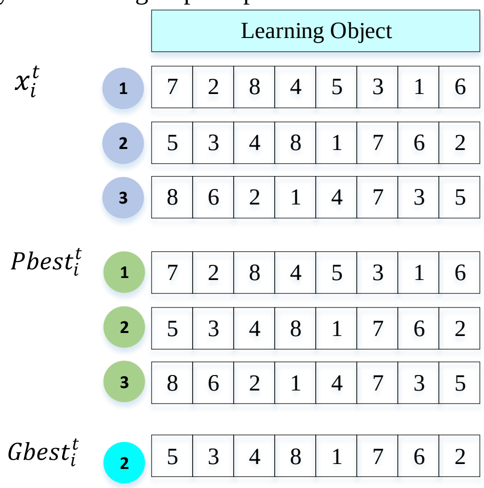


Fig. 4 Particle representation at iteration t = 0

At the 0th iteration (t = 0), the value of all particle is $v_i(t) = \emptyset$ and the starting position of all particle is randomly generated in the form of integer numbers. These numbers represent LO number and uniquely combined. For example, LO $x_{i=1}$ [7 2 8 4 5 3 1 6] means that LO sequence is started from LO7 toward LO 2, 8, 4, 5, 3, 1, 6 and return to LO7.

Connection Weight CW and the amount of unused particle ($UnLO$) from each Course Prerequisites (CP) are counted to determine *Fitness Function*. P_{best} value at 0th iteration (t = 0) is the same value with particle starting position, i.e. ${P_{best}}_i(t) = x_i(t)$.

Pbest with the heightest fitness value determines G_{best} value $\left(k = argMax_i\{fitness\ {P_{best}}_i(t)\} = 2\right)$, so that ${G_{best}}_{g=1}(t=0) = {P_{best}}_{i=2}(t=0)$, i.e. ${G_{best}}_1$ (0) [5 3 4 8 1 7 6 2].

2. Update Position

Figure 5 shows learning object update positions. Transposition pattern allows learning object with particle x_i[7 2 8 4 5 3 1 6] and ${G_{best}}_i$ [5 3 4 8 1 7 6 2] target shifted several times. The shift was started from position (1,5)-(2,6)-(3,4)-(5,7)-(6,7)-(7,8). Equation (5) and (6) will produce particle position of x_{i+1} [5 3 4 8 7 2 1 6].

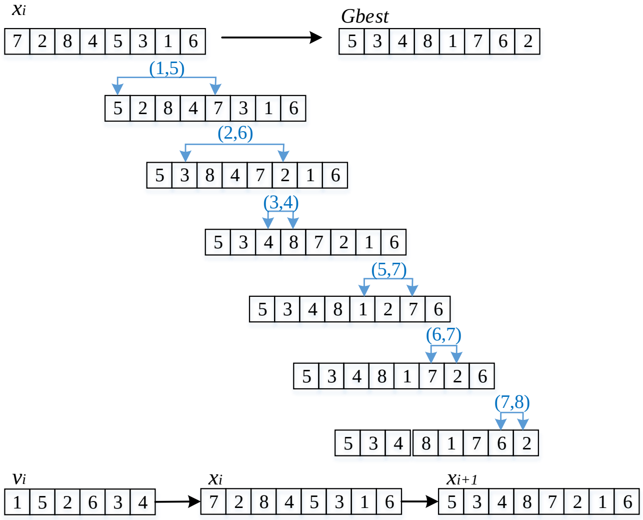


Fig. 5 Learning object position update

3. Fitness Function

Connection Weight (CW) was used to assess the relationship of LO in RBT ontology as cognitive level evaluators [22] in (7). RBT cognitive level evaluators assess only the cognitive levels relationship of (C1, C2, C3, C4, C5, C6) where the value between levels is 1.

$$cw = \frac{k}{t_1 \cdot |DBO| + t_2 \cdot |DBB|} \quad (7)$$

This study uses Distance by Bloom (DBB) to measure the cognitive distance depth and breadth ($d_{k,l}$)between LO using (8),

$$d_{k,l} = \sqrt{(k_2 - k_1)^2 + (l_2 - l_1)^2} \quad (8)$$

Equation 8 is used to calculate the cognitive distance of LO1 with LO3, LO1 with cognitive target KC2 in cognitive position (2.2), while LO3 with cognitive target PC2 in cognitive position (3.2) obtained cognitive distance 1.414.

Distance by Ontology (DBO) is the distance found as the number of levels in an ontology. For example DBO distance calculation between "LO1" and "LO3". LO1 and LO2 in the ontology are not directly connected. The DBO calculation starts from the distance of LO1 to LO2 is 0.5 and LO2 to LO3 is 0.5, so DBO is equivalent to 1 level. The coefficients $t_1$and $t_2$depend on the type of LO which can be both theoretical and practical. For practical LO types, taxonomic distance (DBB) is more important. For theoretical LO types, ontology distance (DBO) is more important.

In the following is how to calculate the *CW* value between "Object Oriented Medotology: KC2" and "Class and Object: PC2". By default, the value of k is 100, the value of t_1 is 1 because LO1 KC2 is theoretical, and the value of t_2 is equivalent to 5 because the LO3 PC2 is practical.

$$CW = \frac{k}{t_1 \cdot |DBO| + t_2 \cdot |DBB|} = \frac{100}{1 \cdot |1| + 5 \cdot |1{,}414|} = 12{,}392$$

The fitness function proposed in this study is to make an individual learning path or route based on RBT and the learning object ontology shown in (9),

$$F = \alpha * CW + \frac{1}{\beta * \sum UnLO} \quad (9)$$

with:

α , β *is* $0 - 1$
CW is connection weight
UnLO is an unused Learning Object based on the Course Prerequisites CP {1, 2, 3}.

4. Application of the DPSO Algorithm
The methodology, steps and strategies of the Discrete Particle Swarm Optimization algorithm in detail are as follows:

Step 1: Initialization.
Initialize population, the number of iterations ($Itermax$), and speed of each particle. Particle position x_i is LO arranged in a random generated array [1…n] randomly based on CP {1,2,3}. Calculate connection weight through DBO and DBB calculations with (7),(8) between LO.

Step 2: Fitness Function Calculation.
Calculate the fitness function of based on CW of each particle with (9).

Step 3: Initialization of P_{best} and G_{best}
The initialati value of Pbest Value is $x_i(t)$, select the P_{best} with highest fitness value to determine

$$G_{best}(k = argMax_i\{fitnessP_{best_i}(t)\})$$

Step 4: Start the iteration, $iter = 1$

Step 5: Velocity Update
Update velocity for each LO with the transposition pattern using (5).

Step 6: Position Update
Update particle position for each LO (6), then calculate the fitness function of each cognitive class based on CW for each particle.

Step 7: P_{best} Update
Change the current particle P_{best} with the current position of the particle if and only if the current fitness value is better than the previous P_{best}.

Step 8: G_{best} Update
Determine G_{best} by choosing one P_{best} with the highest fitness value.

Step 9: Iteration Termination Criteria
If the current iteration of the $iter < Itermax$, then proceed to Step 4, if not continue to Step 10.

Step 10: The outcome of the best G_{best} position.

IV. RESULT AND DISCUSSION

CW testing is used to determine the quality of RBT and ontology relationships that were first discussed, then test and discuss fitness functions based on Course Prerequisites with the number of particles used, and finally, the DPSO algorithm can display learning paths through global best.

A. Testing for Connection Weight

The mechanism for testing connection weight according to in accordance with the procedure shown in Figure 6.

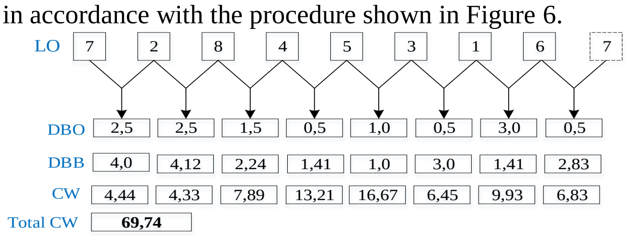


Fig. 6 Testing for CW

The process of calculating CW from the LO starts with finding the value of DBO, DBB, and CW from each LO. Determination of CP will affect the number of LO to be calculated in each iteration, complete CW testing is presented in Table 4. The results of manual CW calculations show the same results as the tests on the DPSO algorithm.

TABLE 4. CONNECTION WEIGHT TESTING DATA

No	Learning Object								CP	CW	
										Manual	DPSO
1	2	6	1	3	5	4			1	64.10	64.1038
2	3	2	1	6	5	4			1	90.56	90.5569
3	2	3	6	1	5	4			1	64.53	64.5303
4	4	5	6	3	7	2	1		2	78.81	78.8128
5	1	3	4	2	6	5	7		2	71.91	71.9115
6	4	6	5	2	1	7	3		2	89.86	89.8589
7	7	2	8	4	5	3	1	6	3	69,74	69,7423
8	5	3	4	8	1	7	6	2	3	94,25	94,2485
9	8	6	2	1	4	7	3	5	3	72,70	72,6956

B. Fitness Function Testing

Testing for the fitness function is done to ensure the fitness function can work properly according to the three proposed requirements.

1. Fitness Function Testing with CP 1

Figure 7 (a) presents a testing of fitness functions for CP 1 with six LOs consisting of 5 groups of particles, whereas Figure 7 (b) tests the fitness function with 10 groups of particles.

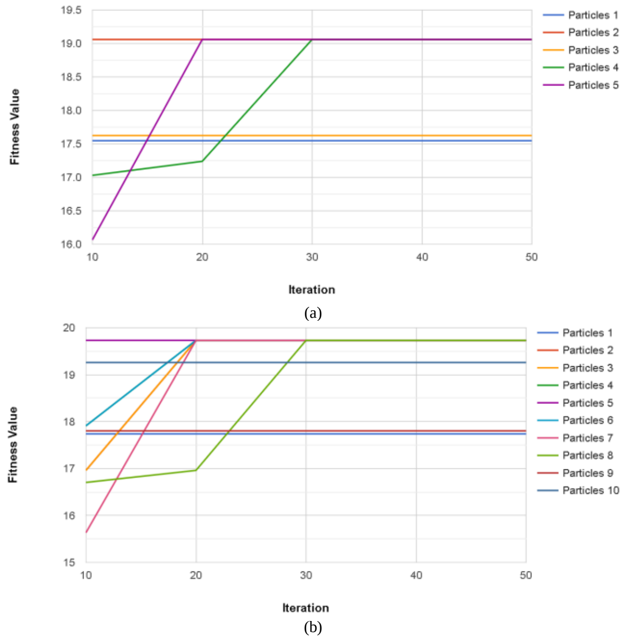
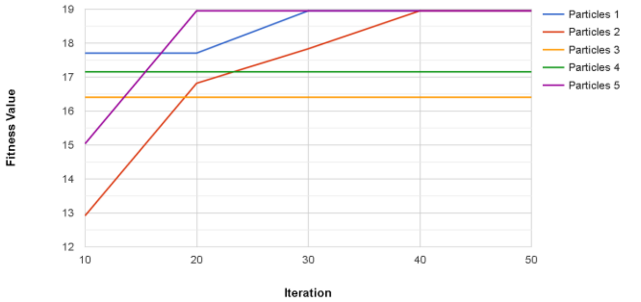


Fig. 7 Testing the P_{best} Fitness Function on CP 1 with 5 particles (a) and P_{best} with 10 particles (b)

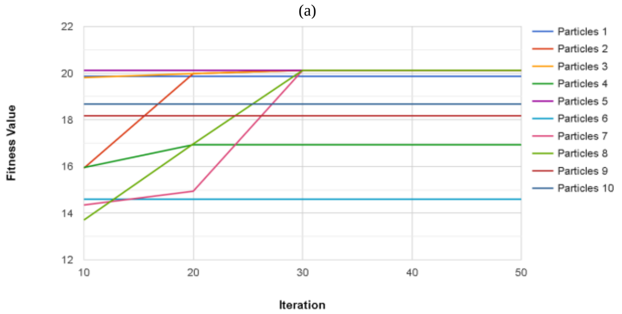
2. Fitness Function Testing with CP 2

Figure 8 (a) presents a testing of fitness functions for CP 1 with seven LO consisting of 5 groups of particles, whereas Figure 8 (b) tests the fitness function with 10 groups of particles.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 08, August-2019



(a)

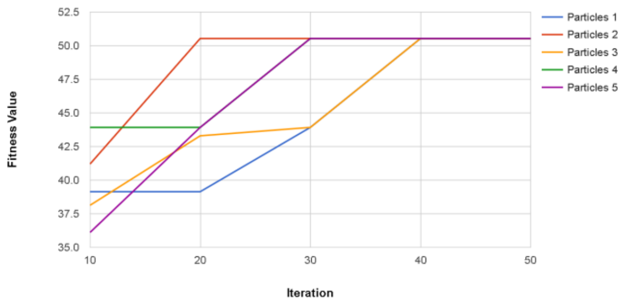


(b)

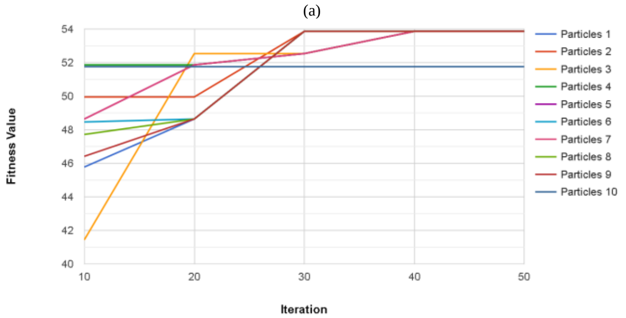
Fig. 8 Testing the P_{best} Fitness Function on CP 2 with 5 particles (a) and P_{best} with 10 particles (b)

3. Fitness Function Testing with CP 3

Figure 9 (a) presents a testing of fitness functions for CP 1 with eight LO consisting of 5 groups of particles, whereas Figure 9 (b) tests the fitness function with 10 groups of particles.



(a)



(b)

Fig. 9 Testing the P_{best} Fitness Function on CP 3 with 5 particles (a) and P_{best} with 10 particles (b)

The testing of the fitness function above shows that the higher the iteration that is used, the more optimal the solution produced by the system with the result of increasing fitness. This is due to the increasing number of iterations that are used to make particles move to find more optimal solutions, allowing particles to find the optimal solution.

C. Learning Path Recomendations

Learning path recommendations shown in Table 5 indicate that an increase in the number of particles affects the value of the learning path sequence generated. Increasing the resulting fitness value can be caused by particle representation or particle evaluation such as strategy randomization and improvement strategies used are able to explore all existing swarm space, or it is possible that the swarm space in this problem has sufficient scope.

TABEL 5. LEARNING PATH RECOMENDATION

No	CP	Number of Particles	Learning Path	
			DPSO	Manual Set
1	1	5	3,2,1,6,5,4	1,2,3,4,5,6
2		10	3,2,1,6,4,5	
3	2	5	6,4,5,2,3,7,1	1,2,3,4,5,6,7
4		10	6,4,5,3,2,7,1	
5	3	5	1,3,2,5,4,7,8,6	1,2,3,4,5,6,7,8
6		10	1,2,3,5,4,7,8,6	

The DPSO algorithm can create a learning path in accordance with the CP required in the manual set. Changes in the number of particles do not really affect the learning path sequence of each CP. The similarity of the learning path sequence based on the number of particles for CP 1 was 83.3%, CP 2 was 85.71%, and CP3 was 87,5%, so that the average similarity of the learning path sequence was 85.5%.

V. CONCLUSION

Discrete Particle Swarm Optimization algorithm was applied to overcome combinatorial problems more practically and regularly in determining learning path. Determination of the learning object sequence through the assessment of the quality of connections between RBT and LO ontology. Experiments show that the models and techniques presented were suitable for finding learning paths that are in accordance with student cognitive abilities.

In the future research can be developed through improving algorithms with Hybrid Discrete Particle Swarm Optimization (HDPSO) to find more accurate solutions, improve more complex ontologies, and implement systems as public services available online.

REFERENCES

[1] Krathwohl, D. R., Anderson, L. W., Airasian, P. W., Cruikshank, K. A., Mayer, R. E., Pintrich, P. R., … Wittrock, M. C. *A Taxonomy For Learning, Teaching, And Assessing: A Revision Of Bloom's Taxonomy Of Educational Objectives*. *New York Longman*, *41*(4), 302, 2002

[2] Saido, *et al.*, "Teaching strategies scale for promoting higher order thinking skills among students in science," *Proceedings of ISER 5th , International Conference*, 91–94, 2015.

[3] D. Sukla and A.P. Dungsungneon, "Students Perceived Level and Teachers Teaching Strategies of Higher Order Thinking Skills A Study on Higher Educational Institutions in Thailand," *Journal of Education and Practkice*, 7(12), 211–219, 2016.

[4] Chinedu, C. C., Olabiyi, O. S., & Kamin, Y. Bin. "Strategies for improving higher order thinking skills in teaching and learning of design and technology education". *Journal of Technical Education and Training*, *7*(2), 35–43, 2015.

[5] Thitima, G., & Sumalee, C. “Scientific Thinking of the Learners Learning with the Knowledge Construction Model Enhancing Scientific Thinking”. *Procedia - Social and Behavioral Sciences*, *46*(1999), 3771–3775, 2012.

[6] Acampora, G., Gaeta, M., & Loia, V. “Hierarchical optimization of personalized experiences for e-Learning systems through evolutionary models”. *Neural Computing and Applications*, *20*(5), 641–657, 2011.

[7] Chen, C. M. “Intelligent web-based learning system with personalized learning path guidance”. *Computers and Education*, *51*(2), 787–814, 2008.

[8] De-Marcos, L., Pages, C., Martínez, J. J., & Gutiérrez, J. A. “Competency-based learning object sequencing using particle swarms”. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, *2*, 111–116, 2007.

[9] Al-Muhaideb, S., & Menai, M. E. B. “Evolutionary computation approaches to the Curriculum Sequencing problem”. *Natural Computing*, *10*(2), 891–920, 2011.

[10] Dukhanov, A., Karpova, M., & Shmelev, V. “An automation of the course design based on mathematical modeling and genetic algorithms”. *Proceedings - Frontiers in Education Conference, FIE*, *2015–Decem*(December),7–10, 2015.

[11] Huang, M. J., Huang, H. S., & Chen, M. Y. “Constructing a personalized e-learning system based on genetic algorithm and case-based reasoning approach”. *Expert Systems with Applications*, *33*(3), 551–564, 2007.

[12] Christudas, B. C. L., Kirubakaran, E., & Thangaiah, P. R. J. “An evolutionary approach for personalization of content delivery in e-learning systems based on learner behavior forcing compatibility of learning materials”. *Telematics and Informatics Jurnal*. https://doi.org/10.1016/j.tele.2017.02.004

[13] Shmelev, V., Karpova, M., & Dukhanov, A. (2015). “An Approach of Learning Path Sequencing Based on Revised Bloom’s Taxonomy and Domain Ontologies with the Use of Genetic Algorithms”. *Procedia Computer Science* (Vol. 66), 2015.

[14] Kennedy, J., & Eberhart, R. “Particle swarm optimization”. *IEEE International Conference on Particle Swarm Optimization*, 1995.

[15] J. Kennedy and R. C. Eberhart, “A discrete binary version of the particle swarm algorithm”, *IEEE, International Conference on Systems*, Man, and Cybernetics, Orlando, FL, vol.5, pp.4104-4108, 1997.

[16] Engelbrecht, A. P. *Computational Intelligence An Introduction* (Second Edi). England: John Wiley & Sons Ltd, 2007.

[17] Li, X., & Deb, K. “PSO Niching algorithms Using Different Position Update Rules”, 2010.

[18] Kennedy, J. and Eberhart, R.C. A discrete binary version of the particle swarm algorithm,1997.

[19] Clerc, M. Discrete Particle Swarm Optimization Illustrated by the Traveling Salesman Problem, 2000.

[20] Goldbarg, E.F.G; Souza, G.R. & Goldbarg, M.C. “Particle swarm for the traveling salesman problem”. Proceedings of the EvoCOP 2006, Gottlieb, J. & Raidl, G.R. (Ed.), Lecture Notes in Computer Science, Vol. 3906, pp. 99-110, ISBN: 3540331786, Budapest, Hungary, April 2006a, Springer, Berlin

[21] Goldbarg, E.F.G; Souza, G.R. & Goldbarg, M.C.”Particle swarm optimization for the bi-objective degree-constrained minimum spanning tree”. Proceedings of Congress on Evolutionary Computation, Vol. 1, pp. 420-427, ISBN: 0780394879, Vancouver, BC, Canada, July 2006b, IEEE

[22] Shmelev, V., Karpova, M., & Dukhanov, A. “An Approach of Learning Path Sequencing Based on Revised Bloom’s Taxonomy and Domain Ontologies with the Use of Genetic Algorithms”. *Procedia Computer Science* (Vol. 66), 2015.