

Improved personalized e-course Composition Approach using Modified Particle Swarm Optimization with Inertia-Coefficient

Dheeban S.G, Deepak V, Dhamodharan L, Susan Elias

ABSTRACT

One of the main problems associated with the authoring of e-courses, for e-learning systems, is that the current composition-approaches do not support ‘personalized-learning’ or in other words, the current composition approaches fail to take into consideration the difference in individual learning-capabilities and the background knowledge of the individual learners and thus do not provide materials that exactly meet the demands of the individual learners. In order to provide solution for this problem, in the past, various e-course composition approaches had proposed to use various methods of computational-optimization techniques like Genetic Algorithm and Particle swarm optimization. The primary purpose of this paper is to propose an improved personalized e-course composition approach using modified particle swarm optimization algorithm (MPSO) with inertia-coefficient, which intends to serve as an effective solution to the afore-mentioned problem. Various simulation-based experiments were conducted and the results of these experiments have been furnished at the end of this paper. These results demonstrate that our proposed approach is an effective solution to the problem of ‘personalized-learning’. In addition, these graphs compare our proposed approach with an existing approach which uses Basic particle swarm optimization algorithm (BPSO). These comparisons demonstrate the efficiency of our proposed model.

Keywords

Personalized e-course composition, Particle swarm optimization (PSO), Nature-Inspired optimization techniques, personalized learning, adaptive e-learning.

1. INTRODUCTION

E-learning is the process of learning with the help of electronic gadgets like computers and other devices, aided by technological innovations like the internet [1]. An e-learning system ideally has certain specific duties or functionalities to perform such as delivering appropriate e-learning materials to the user, recording user feedback, managing necessary resources and monitoring all the necessary processes. Besides, for the purpose of facilitating sharing of e-course materials by various e-learning systems, various standards like Content Object Model (COM) and Sharable Content Object Reference Model (SCORM) have been introduced [2]. Simultaneously, efforts have been taken to

develop authoring-systems that aid instructors in composing e-

courses [3, 4]. However, there is one common problem associated with all these composition approaches: all the afore-mentioned composition approaches fail to take into consideration the difference in individual learning capabilities and the background knowledge of the learner while composing the e-courses. As a result the composed courses do not support the individual learner’s demands or in other words, the existing composition approaches do not support ‘personalized-learning’. The idea behind ‘personalized-learning’ is to provide materials that meet the demands of the individual learners [5-7]. This paper moves on these lines. In the past, various approaches have been proposed to provide solution to the above-mentioned problem of ‘personalized learning’ [1, 8]. In 2009, Chu et al proposed a ‘personalized e-course composition approach’ based on basic particle swarm optimization algorithm (BPSO) [9]. In this paper, we propose to modify this ‘personalized e-course composition approach’ using a modified particle swarm optimization algorithm (MPSO) with inertia coefficient [10-12] to improve the solution characteristics. We propose to investigate two different models to vary the inertia coefficient in the MPSO algorithm namely Linear varying inertia coefficient PSO (LPSO) and Random varying inertia coefficient PSO (RPSO) as guided by [13-15] and observe their effects on this specific problem statement of ‘personalized learning’ which have not been observed so far.

The remainder of the paper is organized as follows: In section 2, the paper focuses on the problem statement which this paper aims to solve. In Section 3, the paper discusses the methodology we propose to incorporate in our approach, in order to solve the problem mentioned in section 2. Section 4 is dedicated for discussing details regarding the experiments we performed, to evaluate our approach, and also provides the results of these experiments, to serve as evidence for the effectiveness of our approach. Section 5 concludes the paper.

2. PROBLEM BACKGROUND

In 2009, Chu et al have identified certain factors as the important contributing factors that need to be considered, to effectively solve the problem of ‘personalized learning’ [9]. They are as follows:

- 1) Whether or not the covered e-learning concepts of the personalized e-courses meet the expected learning target of the user: The expected learning target of the user depends on his/her past experience. The past learning experience involves the knowledge which the learner has learnt and has completely understood.
- 2) Whether or not the difficulty level of the e-learning material matches a learner’s ability level: The ability of

the learner depends on age, the level of education and learning subjects and so on. Hence, learners with different ability levels must be provided with different materials with corresponding difficulty levels.

- 3) The limitation of learning time for individuals: Since a learner's ability and attention affect the individual learning time, the expected learning time for each learner is different.
- 4) The weight of the learning concepts covered in a personalized e-course: To avoid the situation that the weight of the learning concepts covered in an e-course is not balance, the balance of the learning concepts need to be considered as an important factor.

Chu et al further represented all the afore-mentioned factors as fitness functions, which facilitate the usage of evolutionary algorithm like particle swarm algorithm (PSO) for this specific problem statement [10-11]. In addition, they identified five parameters that vividly describe the individual learner and five other parameters that describe the characteristic of the e-learning materials that are a part of the e-course to be offered. They are as follows:

2.1 Definition of Parameters

1) Parameters regarding learners

- i. $\{L_1, L_2, \dots, L_k\}$ denotes K learners.
- ii. $\{A_1, A_2, \dots, A_k\}$ denotes the ability level of K learners where $A_k, 1 \leq k \leq K$, denotes the ability level of learner L_k .
- iii. $\{H_1, H_2, \dots, H_k\}$ denotes the expected learning targets of K learners where each H_k has M binary values, $H_k = \{h_{k1}, h_{k2}, \dots, h_{km}\}$, where $h_{km} = 1$, if $1 \leq k \leq K$ and $1 \leq m \leq M$, represents the expected learning target covers the learning concept C_m . Else it is 0.
- iv. Constraint $t_{l,k}, 1 \leq k \leq K$: Lower bound on the expected learning time of an e-course for the learner L_k .
- v. Constraint $t_{u,k}, 1 \leq k \leq K$: Upper bound on the expected learning time of an e-course for the learner L_k .

2) Parameters regarding e-learning materials

- i. $\{C_1, C_2, \dots, C_M\}$ denotes M learning concepts which the learner expects to learn from an e-course. These relate to the specific concepts in a curriculum.
- ii. $\{LM_1, LM_2, \dots, LM_N\}$ denotes the N candidate e-learning materials each of which covers different concepts. From these N e-learning

materials, the most suitable materials for a particular learner L_k are chosen.

- iii. $\{D_1, D_2, \dots, D_N\}$ denotes the difficulty level of the N candidate e-learning materials where $D_n, 1 \leq n \leq N$, denotes the difficulty level of e-learning material LM_n .
- iv. $\{R_1, R_2, \dots, R_N\}$ denotes the covered learning concepts of N e-learning materials where each R_n has M binary values i.e. $R_n = \{r_{n1}, r_{n2}, \dots, r_{nM}\}$ and $r_{nm} = 1, 1 \leq m \leq M$, if the e-learning material LM_n covers the concept C_m and $r_{nm} = 0$ otherwise.
- v. Coefficient $t_n, 1 \leq n \leq N$: Required time for reading the e-learning material LM_n .

3) **Decision variable x_{nk}** , $1 \leq n \leq N$ and $1 \leq k \leq K$: Where $x_{nk} = 1$ if the e-learning material LM_n is to be composed into the e-course of the learner L_k , and $x_{nk} = 0$ otherwise.

2.2 Definition of fitness-function

Chu et al further represented the four criteria mentioned earlier as four sub-fitness functions as follows:

1. Sub-fitness function, F_1 , gives the average difference between the covered learning concept and the expected learning of a learner L_k . This objective function gives an idea about which learning materials cover those learning concepts which are required by the learner.

$$F_1 = \frac{\sum_{m=1}^M \sum_{n=1}^N x_{nk} |r_{nm} - h_{km}|}{\sum_{n=1}^N x_{nk}}, 1 \leq k \leq K$$

2. Sub-fitness function, F_2 , gives the average difference between the difficulty level of the e-learning material and the learner's ability level. This objective function helps in identifying e-learning materials which suits the ability level of the learner L_k .

$$F_2 = \frac{\sum_{n=1}^N x_{nk} |D_n - A_k|}{\sum_{n=1}^N x_{nk}}, 1 \leq k \leq K$$

3. Sub-fitness function, F_3 , gives the required learning time between the lower and upper bound of the expected learning time of the learner L_k . This objective function is to ensure that the total time required for finishing the e-learning materials which are

selected for the particular learner, fall within that learner's expected learning time.

$$F_3 = \left(\max \left(t_{l_k} - \sum_{n=1}^N t_n x_{nk}, 0 \right) \right) + \left(\max \left(0, \sum_{n=1}^N t_n x_{nk} - t_{u_k} \right) \right), 1 \leq k \leq K$$

4. Sub-fitness function, F_4 , is used to balance the weight of the learning concepts. This is used in order to avoid the situation where the learning concepts covered in a personalized e-course are not balanced.

$$F_4 = \sum_{n=1}^m h_{km} \left| \sum_{n=1}^N x_{nk} r_{nm} - \frac{\sum_{n=1}^N \sum_{n=1}^M x_{nk} r_{nm}}{\sum_{n=1}^M h_{km}} \right|, 1 \leq k \leq K$$

The afore-mentioned four sub-fitness functions are aggregated after being multiplied by their corresponding relative weights (w_1, w_2, w_3, w_4) in order to obtain the final fitness function, F .

$$\min F(x) = \sum_{j=1}^4 w_j F_j$$

The basic idea is to minimize each of the sub-fitness functions, which is achieved when the final fitness function is minimized.

3. METHODOLOGY

This section discusses the methodology we propose to incorporate in our approach, in order to provide an effective solution to the problem statement defined in the previous section. We propose to use a modified particle swarm optimization (MPSO) algorithm with an inertia coefficient to improve the solution characteristics of [9]. We incorporate the MPSO algorithm for our problem statement using two models – (1) Linear varying inertia coefficient PSO (LPSO) as guided by [12-14] and (2) Random varying inertia coefficient PSO (RPSO) [15] which are explained below.

3.1 Basic Particle Swarm Optimization

Basic Particle Swarm Optimization (BPSO) algorithm was inspired by the swarm behavior of birds and fishes to find near optimal solutions [10-11]. In this algorithm, the computational equivalent of a bird is called a particle, which represents a vector that refers to a candidate solution. To evaluate the quality of a particle vector, a fitness function is defined to estimate whether or not a particle vector is close to the optimal solution. According to

the derived result of the fitness function for each particle, each particle maintains two best experienced vectors: (1) **pbest_i**: the best vector of a particle_i experienced so far; and (2) **gbest**: the best vector of the entire swarm. According to **pbest_i** and **gbest**, the velocity and position of each particle are iteratively modified to evolve new particles that are closer to optimal solutions. While a terminal criterion is satisfied, a near optimal solution can be derived by the fitness function. In [9], Chu et al proposed to use a basic binary version of PSO as guided by [11] which uses the following formulae (1) for updating the velocity of the particles, where 'i' represents the ith particle and C1, C2 represent acceleration constants, 't' represents the iteration, 'Y' represents the current position and 'd' represents the dimension.

$$(1) V_{i,d}^t = V_{i,d}^{t-1} + C_1 * Rand_1 * (pbest_{i,d} - Y_{i,d}^{t-1}) + C_2 * Rand_2 * (gbest_{i,d} - Y_{i,d}^{t-1})$$

3.2 Modified Particle Swarm Optimization

However, [12, 20] observes that: in population-based optimization methods, proper control of global exploration and local exploitation is crucial in finding the optimum solution efficiently. Shi et al in [12] introduced the concept of inertia coefficient (ω), to the original version of PSO, in the calculation of velocity of particles, in order to balance the local and global search during the optimization process. The above concept is mathematically represented as in (2):

$$(2) V_{i,d}^t = \omega * V_{i,d}^{t-1} + C_1 * Rand_1 * (pbest_{i,d} - Y_{i,d}^{t-1}) + C_2 * Rand_2 * (gbest_{i,d} - Y_{i,d}^{t-1})$$

3.2.1 Linear varying inertia coefficient PSO (LPSO)

Generally, in any population-based optimization methods, considerably high diversity is necessary during the early part of the search to allow use of full range of the search space. On the other hand, during the latter part of the search, when the algorithm is converging to the optimal solution, fine-tuning of the solutions is important to find the global optima efficiently. Considering these facts, Shi et al [12-14] have found a significant improvement in the performance of the PSO method with a linearly varying inertia coefficient (ω) in the calculation of the velocity particles. The above concept is mathematically represented as in (3):

$$(3) \omega = (\omega_{initial} - \omega_{final}) * (Iteration_{maximum} - Iteration_{current}) / Iteration_{maximum} + \omega_{final}$$

Through empirical studies, Shi et al [14] have observed that the optimal solution can be improved by varying the value of ω from 0.9 at the beginning of the search to 0.4 at the end of the search for most problems. The variation of the inertia coefficient (ω) by this method is shown in Fig 1 in the results section.

3.2.2 Random varying inertia coefficient PSO (RPSO)

In 2001, Shi et al observed that replacing the concept of linear variation by random variation of inertia coefficient could significantly improve the performance of MPSO for many real-world problems [15]. The concept is mathematically represented as in (4):

$$(4) \omega = 0.5 + (rand(.) / 2)$$

where rand(.) is a uniformly distributed random number within the range [0,1]. Therefore, the mean value of the inertia coefficient is 0.75. The variation of the inertia coefficient (ω) by this method is shown in Fig 2 in the results section.

4. EXPERIMENTS AND RESULTS

In order to evaluate the approach proposed in the previous section, we conducted a variety of simulation-based experiments. In this section, we give details regarding the environment under which the experiments were performed and also give a vivid account of all the experiments performed to verify our proposed approach. We conclude this section by furnishing the results of the various experiments that show the effectiveness of our proposed approach.

4.1 Experiments

For the purpose of evaluation of our proposed approach, we simulated the three algorithms namely BPSO, LPSO and RPSO under the following environment: Operating System: Windows XP Professional Version 2002 Service Pack 2, CPU: Intel(R) Core(TM) 2, RAM: 1 Giga Byte, Programming Language used: C-language, compiled and ran in Microsoft Visual C++ 6.0.

1. **Experiment 1:** The aim of this experiment is to investigate the quality of the final fitness-value obtained from all the three algorithms – BPSO, LPSO, and RPSO and thus to facilitate a comparison among the three models of PSO for this specific problem statement. In order to achieve that, all the three algorithms were run for 200 e-learning materials spanning 10 concepts and the e-learning materials were divided into 5 difficulty levels. The values for the covered learning concepts R_n and the time required for each e-learning material t_n were assigned randomly. The algorithms were tested for 100 learners who were divided into 5 ability levels. The values, for the expected learning targets for the learners (H_n) and the lower and upper bounds on learning time for each learner were set randomly as guided by [9]. The values of $c1$ and $c2$ were set as 2.0 as informed by [10-14] for all three models of PSO. The number of particles was set to 20. For all three models, the velocity of the particles was restricted in the range of [0,1] as suggested by [11]. For LPSO, the $\omega_{initial}$ was set as 0.9 and ω_{final} was set as 0.4 and the $Iteration_{maximum}$ was

set as 1000. The termination criterion (TC) was set uniformly as 1000 iterations for all three algorithms. Fig 3 shows the graph obtained from this experiment.

2. **Experiment 2:** The aim of this experiment is to observe the performance of each of the three models of PSO when the number of learning materials is increased. Here the all the above mentioned values were kept constant while changing the number of e-learning materials. Fig 4 shows the graph obtained from this experiment.

4.2 Results

In this section we furnish the graphs obtained from the above mentioned experiment. The first two graphs illustrate the variation of inertia coefficient with the number of iterations for LPSO and RPSO respectively. The third graph gives the comparative performance of BPSO, LPSO and RPSO. The fourth graph illustrates the variation of fitness values for all three models when the number of learning materials is varied.

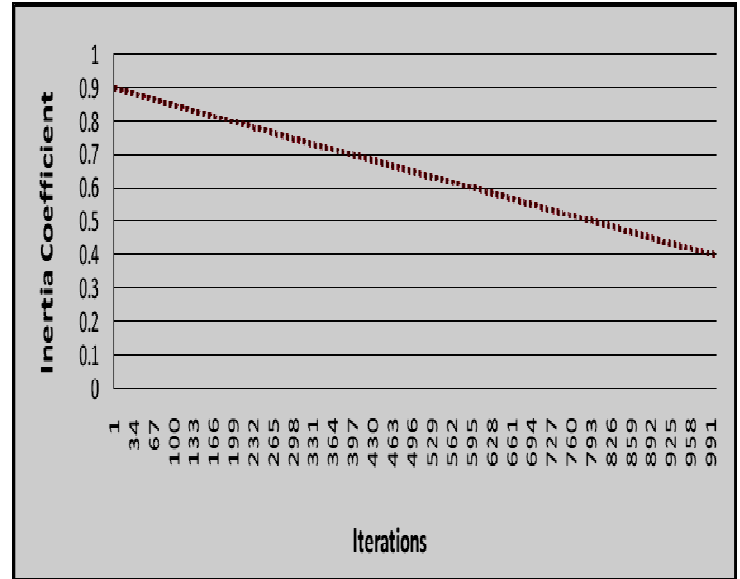


Figure 1: Variation of inertia coefficient with iterations for LPSO

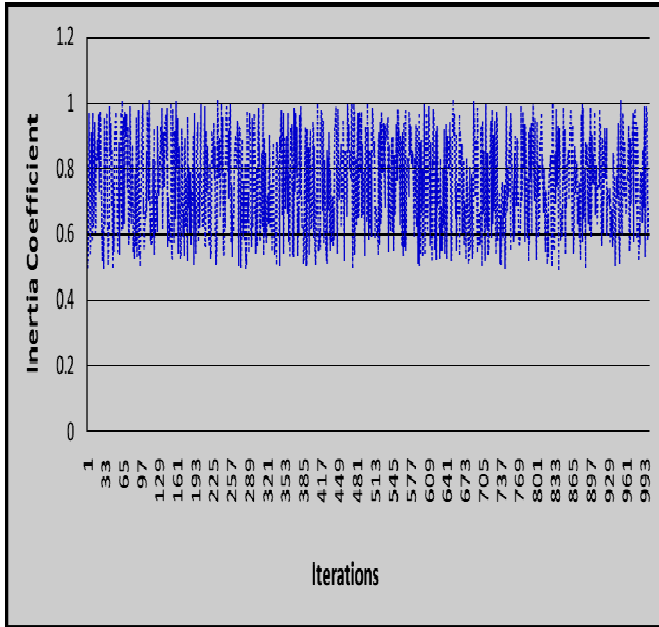


Figure 2: Variation of inertia coefficient with iterations for RPSO

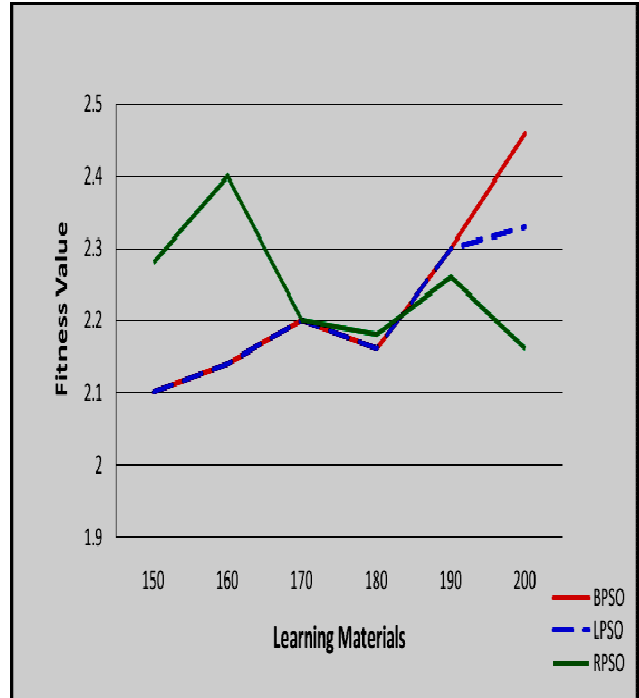


Figure 4: Effect of varying number of learning materials

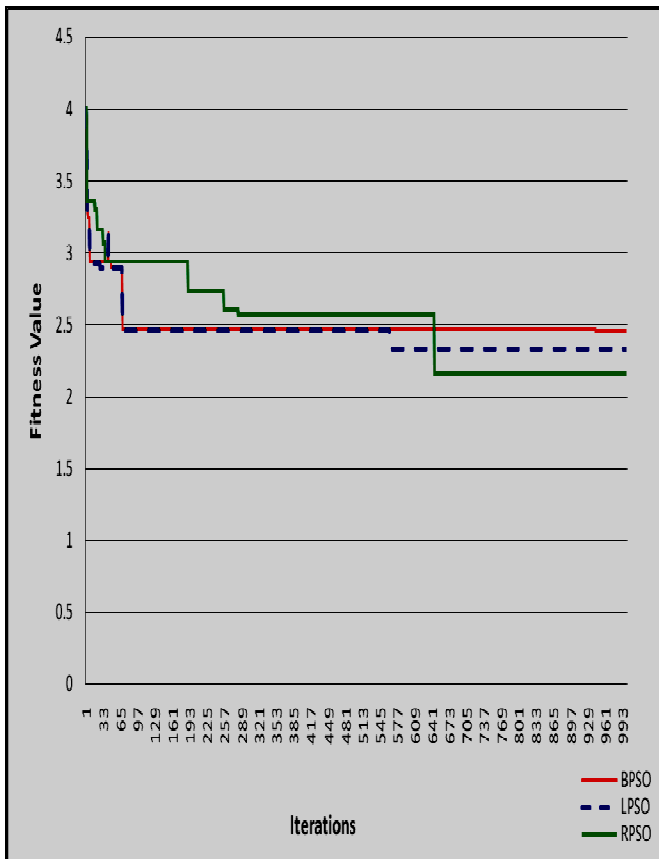


Figure 3: Comparison of fitness values obtained from BPSO, LPSO and RPSO

4.3 Discussion

From Figure 3, we can observe that, when all the 3 models are evaluated under same experimental conditions for a set number of learning materials, the quality of final fitness values provided by our proposed variants of MPSO namely LPSO and RPSO are superior when compared to the currently existing BPSO approach. For our problem statement, such superior values in the final fitness values indicate that the e-courses composed using our proposed approach satisfy the individual learner's demands more than the e-courses composed using the current BPSO approach.

From figure 4, we can observe that as the number of learning materials increases, both our variants clearly out-perform the BPSO. The real significance of this result could be better appreciated by realizing the fact that, in real-time, the learning material count will always be a large quantity and in such scenarios our proposed approach can fare better than the existing BPSO approach.

5. CONCLUSION

In this paper, we have proposed an improved personalized e-course composition approach using modified particle swarm optimization (MPSO). We have investigated the effect of using two models namely Linear inertia coefficient varying PSO (LPSO) and Random inertia coefficient varying PSO (RPSO) for varying the inertia coefficient of the MPSO algorithm. The experimental results indicate that our proposed MPSO approach provides better solution quality than the existing approaches under identical conditions and out-performs the existing BPSO approach, when the learning material count becomes large.

6. ACKNOWLEDGEMENT

We would like to acknowledge and thank the **Defence Research Development Organisation (DRDO)**, New Delhi, India for granting us Extramural Research Funds for carrying out this research work. This work is part of the research project titled “*The design and development of a multimedia presentation system that streams MPEG-21 compatible media-on-demand*”.

7. REFERENCES

- [1] Chen CM, Lee HM, Chen YH (2005) Personalized e-learning system using Item Response Theory. *Comput Educ* 44(3):237–255
- [2] Shareable Content Object Reference Model (SCORM) Version 1.3, Advanced Distributed Learning (ADL) (2004). [Online]. Available: <http://www.adlnet.org/>
- [3] Yang JTD, Chiu CH, Tsai CY, Wu TH (2004) Visualized online simple sequencing authoring tool for SCORM-compliant content package. In: *Proceedings of IEEE international conference on advanced learning technologies*, pp 609–613
- [4] Su JM, Tseng SS, Chen CY, Weng JF, Tsai WN (2006) Constructing SCORM compliant course based on High-Level Petri Nets. *Comput Stand Interfaces* 28(3):336–355
- [5] Brusilovsky P (1999) Adaptive and intelligent technologies for Web-based education. *Kunstl Intell* 13:19–25
- [6] Liu HI, Yang MN (2005) QoL guaranteed adaptation and personalization in E-learning systems. *IEEE Trans Educ* 48(4):676–687
- [7] Brusilovsky P, Eklund J, Schwarz E (1998) Web-based education for all: a tool for development adaptive courseware computer networks and ISDN systems, vol 30, pp 291–300
- [8] Huang MJ, Huang HS, Chen MY (2007) Constructing a personalized e-learning system based on genetic algorithm and case-based reasoning approach. *Expert Syst Appl* 33(3):551–564
- [9] Chu CP, Chang YC, Tsai CC (2009) PC2PSO: personalized e-course composition based on particle swarm optimization. *Applied Intelligence*. Springer Netherlands. ISSN:0924-669X (Print) 1573-7497 (Online) DOI:10.1007/s10489-009-0186-7
- [10] Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Proceedings of IEEE international conference on neural networks*, vol 4, pp 1942–1948
- [11] Kennedy J, Eberhart RC (1997) A discrete binary version of the particle swarm algorithm. In: *Proceedings of IEEE international conference on systems, man, and cybernetics*, vol 5, pp 4104–4108
- [12] Shi Y, Eberhart RC (1998) A modified particle swarm optimizer. *Evolutionary computational proceedings, IEEE* 1998. ISBN: 0-7803-4869-9
- Shi Y, Eberhart RC (1998) Parameter selection in particle swarm optimization in *Evolutionary programming VII*. LNCS, vol 1447, pages 601–610, Springer Berlin.
- [13] Shi Y, Eberhart RC (1998) Empirical study of particle swarm optimization,” in *Proc. IEEE Int. Congr. Evolutionary Computation*, vol. 3, 1999, pp. 101–106.
- [14] Shi Y, Eberhart RC (2001) “Tracking and optimizing dynamic systems with particle swarms,” in *Proc. IEEE Congr. Evolutionary Computation*, Seoul, Korea, pp. 94–97.
- [15] Yin PY, Chang KC, Hwang GJ, Hwang GH, Chan Y (2006) A particle swarm optimization approach to composing serial test sheets for multiple assessment criteria. *Educ Technol Soc* 9(3):3–15
- [16] Huang TC, Huang YM, Cheng SC (2007) Automatic and interactive e-learning auxiliary material generation utilizing particle swarm optimization. *Expert Syst Appl* 35(4):2113–2122
- [17] De-Marcos L, Pages C, Martinez JJ, Gutierrez JA (2007) Competency-based learning object sequencing using particle swarms. In: *Proceedings of 19th IEEE international conference on tools with artificial intelligence 2007 (ICTAI 2007)*, vol 2, pp 111–116
- [18] Cheng SC, Lin YT, Huang YM (2009) Dynamic question generation system for web-based testing using particle swarm optimization. *Expert Syst Appl* 36(1):616–624
- [19] Angeline PJ (1998) Evolutionary optimization verses particle swarm optimization: Philosophy and the performance difference,” in *Lecture Notes in Computer Science*, vol. 1447, *Proc. 7th Int. Conf. Evolutionary Programming—Evolutionary Programming VII*, pp. 600–610.