Proceedings of the 2015 International Conference on Operations Excellence and Service Engineering
Orlando, Florida, USA, September 10-11, 2015

Improving a Particle Swarm Optimization model to solve assignment problems in an industrial environment

Javier André; Nathan Suchar; Luis Jiménez; Karol Esquivel and Eldon Caldwell Marin
Industrial Engineering Department
School of Engineering
Operations Engineering Research Group
University of Costa Rica
javier.andreespinoza@ucr.ac.cr, nathan.suchar@ucr.ac.cr, luis.jimenezquesada@ucr.ac.cr, karol.esquivel@ucr.ac.cr, eldon.caldwell@ucr.ac.cr

Abstract

The use of the Particle Swarm Optimization (PSO) model in workload problems resolution is discussed in this article. This kind of problems are usually resolved by a linear programming model, however, as the amount of assignments and machines grow, the workload problem becomes a complex issue, bigger than what a linear programming model can handle. PSO changes the values of the particle's speed and inertia weight to move the swarm along the solution search space. This research proposes the substitution of the classical uniform distribution to initialize the velocity of the particle with the beta distribution. Comparing both models, the results demonstrate statistical equivalences between the convergence values found by both models, however, it is shown how the amount of iterations required to reach the optimal value is lower when using the beta distribution. This paper shows how the proposed model decreases the number of iterations needed to reach a heuristic solution with a coefficient of variation of 84% and a central tendency of 21. Future research lines can include implementing this approach in a more complex model; studying the initialization of the velocity with other probability distributions, and using crossover operators to generate offspring.

Keywords
Beta distribution, job shop, makespan, Particle Swarm Optimization, workload assignment.

1. Introduction

Workload and resource assignment are everyday tasks in any manufacturing company. Haiyan Huang & Zheng Wang (2009) established that the objective of workload assignment is to balance it between the workstations in order to maximize efficiency and authors Askin & Chen (2006) agree with that. However, executing this kind of assignment can be a complex task, depending on the specific characteristics and the size of the production lines.

Furthermore, it is known that workload assignment problems are a special kind of linear programming problem, nevertheless, large assignment problems can be solved much faster with more specified solution procedures, for which case it is recommended to apply them instead of the simplex method.

As mentioned by Romero et al. (2006) since the conception of the first operations research method, there have been a lot of studies regarding different models that currently work as tools for the decision-making. Because of that multiple research has been made about non-conventional methods, such as non-deterministic methods that range from genetic algorithms, differential evolution, artificial neural networks, Markov models and the use of Particle Swarm Optimization (PSO) proposed by Kennedy et al. (1995), up to different variations of PSO like the PSONK proposed by Chutima et al. (2010) or the PSOMS by Kuo et al. (2011).

The main objective of this sort of research lies on minimizing the makespan, which is defined as the difference between the beginning and ending time of a sequence of jobs.

It is believed that non-deterministic algorithms –like Particle Swarm Optimization- are capable of reaching solutions in a quick and efficient way because they generate different outputs from the same inputs, in other words, the solutions obtained are unpredictable and are obtained via iterations that execute various alternatives and they exhaust them to find a solution.

There are two kinds of algorithms in the Swarm Optimization category: the Ant Colony Optimization (ACO) and the Particle Swarm Optimization (PSO) as indicated by Selvi et al. (2010). In spite that both of them have similar basic concepts that derive from the animal behavior during the search of food, ACO works primarily to solve discrete optimization problems, whereas PSO is focused in the solution of continuous optimization problems.

This scientific paper is focused on the application of PSO, specifically for job-shop problems, and analyzes the possibility of improving the efficiency of the model by modifying the distribution used by the algorithm to initialize the particle's velocity, with the end goal of minimizing the amount of iterations required to reach the convergence value.

In Section II it is showed a theoretical framework that includes the most important concepts related to the proposed topic. Then, the application of PSO in workload assignment with a MATLAB code in which a job-shop problem is solved is discussed in Section III. Furthermore, a critic to the classical PSO model is made and it is developed a new model proposed by the authors with its validation in Section IV. Then the conclusions made by the results of this research are exposed in Section V. Finally, further research lines are discussed in Section VI.

2. Theoretical framework

2.1 Particle Swarm Optimization

Heuristic optimization models such as Particle Swarm Optimization (PSO), are optimization mechanisms based on the combinatorial improvement, and are capable of finding a good solution among a finite set of possibilities, without guaranteeing an optimal result.

According to Kennedy & Ebenhart (1995), PSO is closely related to other optimization methods, such as genetic algorithms and evolutionary programming, and it responds as well to the basic principles for Swarm Intelligence models proposed by Millonas (1994), including the proximity principle, quality principle, diverse answer, stability and adaptability. PSO has the capability of moving along a vast solution search space and respond to environmental factors but without changing its behavior with it, only making changes when it is really necessary.

PSO is a method that can change its state by comparing each particle's best local position with the best position of the whole swarm. The particles of the swarm that will be moving along the search space initialize their position and velocity with random numbers, and in every iteration, each particle is accelerated towards the best global position of the swarm, brought by the particle with the best position known as *gbest*.

During an interview with Dr. Mario Villalobos Arias from the Pure and Applied Mathematics Research Center (CIMPA) of the University of Costa Rica, he explained that the criteria described could also be seen as vector interaction as shown by Fig 1.

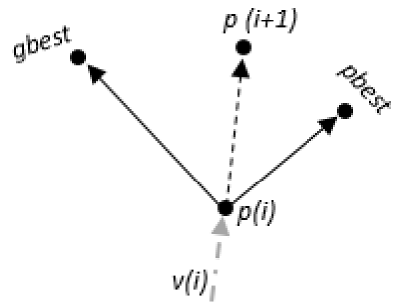

Figure 1. Velocity and position vector of a single particle of the swarm

As shown in Figure 1, PSO compares the current position of the particle p(i) with the gbest and pbest, so the particle knows where should it direct its velocity during the next iteration, shown as p(i+1). The current velocity directed to the best position is represented by the vector v(i).

2.2 Workload assignment

In this research, the assignment problems refers specifically to assigning workload to machines, categorized as a non-polynomial complete problem (NP-Complete). An NP problem means that the solution –if existent- to the problem could only be acquired in non-polynomial period of time, while the solution to an NP-Complete problem can be found in a polynomial period of time.

Usually, JSSP problems involve activities represented by jobs while the resources are represented by machines, and each machine can only process one job at a time, where the main objective is to assign the jobs properly and at the optimal time to the machine in order to reduce the makespan.

The Flow Shop Scheduling Problems are a variation of the JSSP. In this case, every job has the same processing order in every single machine. According to Alharkan (2009), more than two machines are necessary to process the jobs serially.

2.3 Uniform distribution

The Uniform Distribution is used to generate random numbers in the U(0,1) interval. From (1), the value of x (2) can be obtained, concluding that Fx behaves like U(0,1), as shown in (3):

$$(1)\ F_x(x) = \frac{x-A}{B-A}$$

$$(2)\ x = A+(B-A)F_X$$

$$(3)\ F_X \sim U(0,1)$$

According to Ryan T. (2007), there are several special cases for the uniform distribution. One of these cases occurs when the value of A is 0 and the value of B is 1, obtaining U(0,1). For this particular case, the expected value (4) and variance (5) are calculated as follows:

$$(4)\ E(x) = \frac{1}{2}$$

$$(5)\ V(x) = \frac{1}{12}$$

Subsequently, it can be stated that the random numbers generated by the uniform distribution U(0,1) have a variability of 0.0833 and an expected value of 0.5.

2.4 Beta distribution

A given variable X that behaves like a Beta Distribution is presented as indicated in (6).

$$(6)\ X \sim Be(\alpha,\beta)$$

As it is well knownone of the benefits of the beta distribution is that it can adapt depending on the selected values for α and β. Just like the uniform distribution, the beta distribution has an expected value (7) and variance (8).

$$(7)\ E(x) = \frac{\alpha}{\alpha+\beta}$$

$$(8)\ V(x) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

There are also particular cases for this distribution, for example, when α and β have the value of 1, the Beta Distribution becomes a Uniform Distribution as shown by (9)

$$(9)\ X \sim Be(1,1) = U(1,0)$$

3. Current model

Since Kennedy & Ebenhart (1995) proposed the Particle Swarm Optimization original model, multiple studies have been developed in several workload assignment problem applications.

Tasgetiren et al. (2006) implement Particle Swarm Optimization to solve a permutation flowshop problem, aiming to minimize the process's makespan.
Zhang et al. (2008), use an improved PSO algorithm in a flowhop problem in which they include genetic operators, getting better results when working with larger problems than other algorithms such as the genetic algorithm and the heuristic ones. They also apply an adjusted method to accelerate the algorithm, measuring the arrangement of each particle in terms of the makespan.
Zhang et al. (2010), propose using the PSO to minimize the makespan. They use a hybrid method in which they add genetic algorithms and *Annealing Search*. Including genetic algorithms such as crossovers and mutations brings diversity to the swarm and lower the probability of convergence in local minimums.
During 2011, among the rise of PSO investigations, Hu Nai-ping & Wang Pei-li (2010), propose applying the PSO model with a multi-objective optimization method for the resolution of *JSSP* flexible problems. This kind of exercises change the problem´s scheme due to the allowance of performing an operation in many machines at the same time, or that a single machine can perform multiple tasks.
Just like the previously presented investigations, there are many others that have focused on this subject, and despite being a relatively young subject in the academic field, PSO is very relevant for the industrial society.
However, this article is limited to the conventional JSSP workload assignment problems, in which each machine can only process one job simultaneously.
In order to study the application of the PSO model in workload assignment problems, a MATLAB simulation coded by Yang (2004) was acquired. This code simulates a Job-Shop problem being optimized by the classical PSO algorithm.
This model considers 6 jobs that will be processed in 6 different machines, and the jobs can only begin either in the machine 4 or 5, imposing a restriction on the model and forcing the rest of the machines to wait until the jobs are processed. The code analyzes which sequencing is better considering the mentioned restrictions and what is the better way to assign the workload to the machines in order to reduce the processing time and the makespan.
The model performs 100 iterations, and when the simulation ends, two graphics are shown with the obtained results. As shown in Figure 2, the first graphic points out which machine is processing which job at every moment in time. The *y*-axis shows the machine and the *x*-axis indicates the total processing time. Red numbers in the graphic illustrate when a job begins and finishes. A number inside each rectangle is used to designate the job's identification number.
Figure 2 illustrates how the machines that cannot begin a job have downtimes between t=0 and t=5. The path of every job can be tracked in order to study where and when each job is processed.

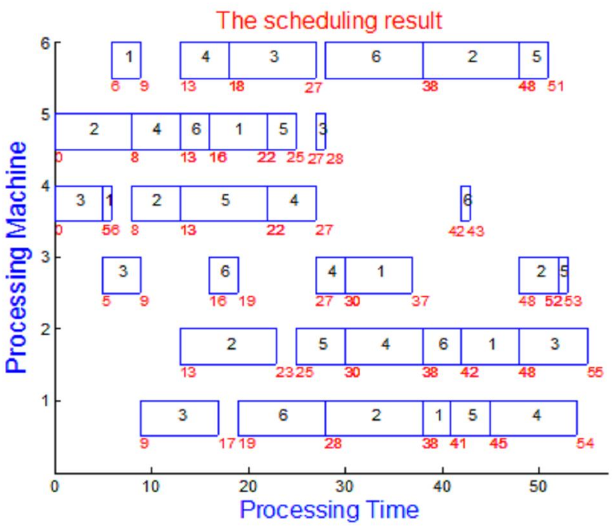

Figure 2. Workload assignment results shown by MATLAB

Figure 3 shows the second graphic presented by MATLAB. This graphic shows the improvement of the model's return value along the iterations. The *x*-axis shows the evolvement generation and in the *y* axis the fitness value can be seen. The convergence speed of the algorithm can be understood by observing and analyzing this image.

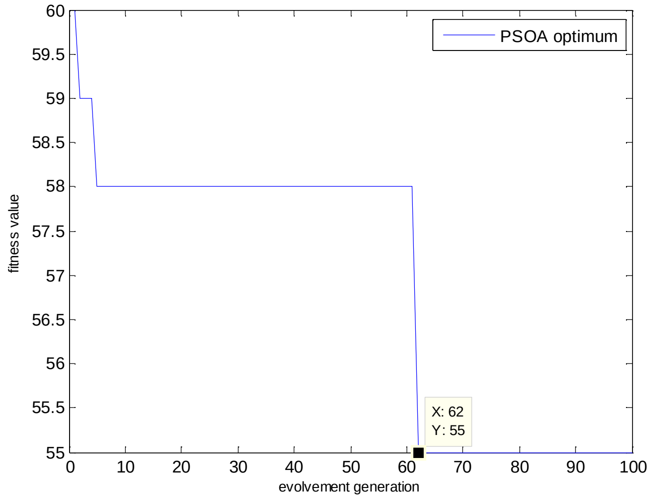

Figure 3. Fitness value shown in the simulation for each iteration

In Figure 3 it is shown how at the beginning, the fitness value converges quickly to 58, where the model gets stuck in this local optimum, until it finally reaches the global optimum during the 62nd iteration

The code also includes a 6x6 matrix "T", in which the processing time for each job in every machine is defined.

$$T=\begin{pmatrix} 1 & 3 & 6 & 7 & 3 & 6 \\ 8 & 5 & 10 & 10 & 10 & 4 \\ 5 & 4 & 8 & 9 & 1 & 7 \\ 5 & 5 & 5 & 3 & 8 & 9 \\ 9 & 3 & 5 & 4 & 3 & 1 \\ 3 & 3 & 9 & 10 & 4 & 1 \end{pmatrix}$$

Figure 4. Processing time matrix T

Each row corresponds to a job and each column to a machine, meaning that the x,y coordinate of the matrix belongs to the processing time for the job *x* in the *y* machine.

If desired, it is possible to alter the matrix's size and the values in it in order to fit the model's parameters to the reality of the user using the code.

4. Current model review and proposal for the new model

Classical PSO models must be initialized with a random position and random velocity for each particle of the swarm. Once this occurs, in the next iterations variables such as the velocity, inertia weight and the positions (pbest & gbest) to move the swarm in the solution search space. When this set of properties are given a less convenient value different scenarios are possible: a less diverse swarm that gets stuck in a local minimum; or, as mentioned by Helwing et al. (2009) it may induce the particles to roam outside of the defined search space, thus generating problematic solutions. The same authors state that one of the most critical variables is the speed, and it is why it is very important that its values are always in a set range for the PSO model to work properly.

Originally, the PSO model initializes the velocity of the particles (v) with (10):

$$(10)\quad V_{j,t+1}= \alpha V_{j,t}+\mathrm{rand}(0,\varphi_1)\left(\theta'_{j,t}-\theta_{j,t}\right)+\mathrm{rand}(0,\varphi_2)\left(\theta'_{g,t}-\theta_{j,t}\right)$$

The speed of each particle has a factor that is determined with the generation of a random number between 0 and 1, with a uniform probability. In addition, usually the generation of random numbers with the uniform distribution assures that the velocities and the positions can be distributed around the whole search area, which balances the exploration and exploitation of the particles of the swarm.

Trelea (2003) defines the exploration as the ability to try several regions of the problem while searching an optimal point and hoping for it to be the global; the exploitation is the ability to concentrate he search around a promising candidate with the idea of finding a global optimum.

Engelbrecht (2012) mentions that the initialization of variables with random numbers evenly distributed with the classic approach is not the best way, especially because it can generate roaming particles –the ones that leave the search space-, which has a direct effect on the convergence time of the model. Furthermore, Helwing et al. (2009) proposed to start the particle's speed at a higher rate, if a better exploration is wanted.

This is why the authors decided to execute tests by initializing the variables with other probability distributions, but varying the appearing probability.

The expected mathematical value of a classic uniform distribution is 0.5, this means that the average value for the initialization of the swarm is 0.5 in the infinite. If this value is raised to a higher number, then it can be achieved a higher velocity for the particles. This way the PSO model can converge quicker, by finding the global optimum without the need of many iterations. However, one must be cautious when the speed is being raised because as said by Helwing et al (2009), really high speed values could impair the search of the area because the particles would not be allowed to analyze the solution area carefully.

The authors chose the beta distribution to generate the random numbers because it can be varied depending on the α and β values chosen; thus generating random numbers with different expected values and at the same time with lower variability than those generated by the uniform distribution.

A beta distribution with α=2, β=1 as its parameters generates random numbers with an expected value of 0.66 instead of 0.5, this is why the speed tends to be higher, allowing a quick convergence of the model. However, by increasing the speed by a 33% in comparison to the uniform distribution the variability of the numbers generated by Be(2,1) is 0.0556, whereas the one for the uniform is 0.0833, which also represents a 33% decrease in variability.

This change of expected value and variability was established in order to balance the exploration and the exploitation of the swarm while searching for the solution because it is generating in average random number with a higher value, but with less variability.

As mentioned earlier, the proposed mathematical model is applied using the software MATLAB. Yang (2004) models a job-shop type of problem that is studied by the authors and then adapted to the new proposed method.

In this code the following equation is stated (11):

$$(11)\quad V_{j,t+1} = eV_{j,t} + 2\times R1\left(\theta'_{j,t} - \theta_{j,t}\right) + 2\times R2\left(\theta'_{g,t} - \theta_{j,t}\right)$$

Where R1 and R2 are matrixes with values between 0 and 1 with a uniformly distributed probability of appearance. The size of this matrixes is given by the size of the swarm and the amount of jobs to be processed:

$$(12)\quad R1 = \mathrm{rand}\left(\mathrm{psize}, \frac{n}{6}\right)$$

$$(13)\quad R2 = \mathrm{rand}\left(\mathrm{psize}, \frac{n}{6}\right)$$

After the changes by the authors to the code where made in MATLAB, the equations for R1 and R2 where as follows:

$$(14)\quad R1 = \mathrm{betarnd}\left(2{,}1{,}\mathrm{psize}, \frac{n}{6}\right)$$

$$(15)\quad R2 = \mathrm{betarnd}\left(2{,}1{,}\mathrm{psize}, \frac{n}{6}\right)$$

Where the first two values inside the *betarnd()* are α and β respectively, followed by the same values the *rand()* had to leave the size of the matrix the same.

5. Validation of the proposed model

For the validation of the proposed model, the MATLAB code was run 50 times for the original one with the uniform distribution and 50 times for the one that was modified with the beta distribution.

The data retrieved from the simulation was put through a normality test with a 95% confidence interval to determine what type of statistical analysis had to be done in order to determine if it really exists a difference between the results of the uniform and the beta distribution.

For both results, the amount of iterations and the value of convergence, the p-value given by the test were lower than 0.005, which means that at a 95% interval of confidence there is no sufficient evidence to consider this data as normal.

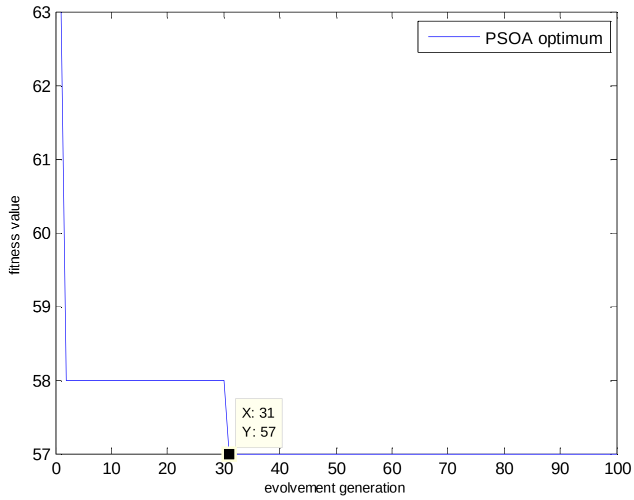

Figure 5. Example of a graphic shown by the simulation with U(0,1)

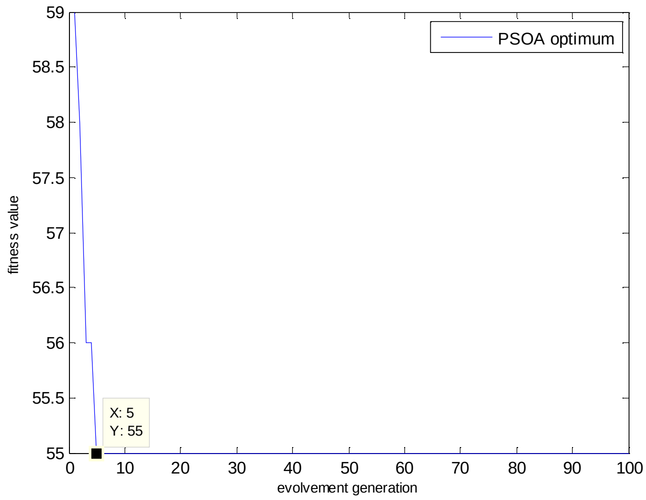

Figure 6. Example of a graphic shown by the simulation with Be(2,1)

Because of this, and with the objective in mind to compare both populations, the authors proceeded to analyze the data with a Mann-Whitney test, which is a non-parametric test in order to prove if the central tendency of both populations is statistically equivalent.

The test for the convergence values indicates that it is significant at 0.0607, which means that at a 95% confidence interval there is not enough evidence to dismiss the null hypothesis, therefore it is said that both populations are statistically equivalent. This means that by using the beta distribution there is no improvement in the convergence values in comparison to the uniform distribution.

However, this test for the amount of iterations indicates that it is significant at 0.0062 therefore the null hypothesis is rejected. This means that there is a significant statistical difference between the two populations thus meaning that the beta distribution does improve the amount of iterations. This can also be demonstrated by the analysis presented in Table 1, where we compare the central tendency parameters of each scenario.

Table 1. Results for central tendency for the amount of iterations

	U(0,1)	Be(2,1)
Mean	15.6	7.1
Mode	3	2
Median	9	5

The results gathered lead us to conclude that the PSO model applied to workload assignment converges to the same value whether it is with U(0,1) or Be(2,1), both of this distributions reach the global minimum.

Nevertheless, by using the beta distribution it is shown a significant difference in the amount of iterations between both models, where the one with beta converges to a global best with fewer iterations thanks to the higher initialization of the velocity of the swarm.

Further analysis of the data shows a standard deviation of the absolute means of 17.71 and a root mean square of 20.88. The absolute differences' coefficient of variation 84.82%. This means that the coefficient of variation for the number of iterations needed to reach a heuristic solution in this model with a central tendency of 20.88 is 84.82%.

6. Conclusions

In the manufacturing sector the production lines are usually constituted by a great amount of tasks that need to be processed by one or many machines. To find the optimal solution to the workload assignment problem where there are a great number of variables such as machines, jobs, capacity, etc., becomes a complex process due to the high number of possible solutions.

The amount of time required to reach a feasible solution to workload assignment problems increases according to the size of the problem, so conventional methods such as linear programming become less effective for the resolution of this problems.

Non-conventional methods such as Particle Swarm Optimization are effective tools for solving job-shop problems where the jobs flow from one machine to the other because the algorithm executes an iterative analysis to find the global best solution in a quick manner.

Although the PSO model is a tool that allows the user to solve workload problems in a fast and effective way, this research has showed that it can be improved to reach the global best in a faster way.

The proposed model has the capacity to generate particles at a higher speed due to the generation of random numbers with a beta distribution with higher expected values and a smaller variability than the distribution that is commonly used.

Thus, by comparing the former PSO model and the one proposed by the authors it is concluded that the latter is able to find an optimal solution faster by reducing the iterations needed to reach the convergence value. By analyzing the results the authors have come to the conclusion that the proposed model improves the number of iterations required by 126% according to the root mean square of the

However, as mentioned earlier, increasing the particles' speed to higher values could impair the study of the area of solutions and this could lead to an early convergence in a local best, thus harming the quality of the solution. This being said, although it is wanted to improve the efficiency of the model by increasing the velocity it cannot be left aside the importance of the quality of the solutions.

7. Future research

The model proposed by the authors utilizes the beta distribution in the initialization of the particle's velocity. This in comparison to the traditional model showed a significant improvement in regard to the reduction of iterations required to reach the global minimum. With the results proposed in this paper it is evident that 17 iterations is a little improvement considering today's technologies, however it could be interesting to develop a research with the approach of this model in a more complex one where there is a great volume of variables involved, perhaps more than 50.

We also encourage other researchers to utilize other probability distributions in the swarm's velocity initialization to evaluate the effect they have on the convergence of the PSO model. Promising results may be found by applying several other ways to initialize the speed of the swarm, such as a faster convergence or maybe better global minimums. Because of this, this kind of research could improve the efficiency of the PSO model.

Another line of research is the application of crossover operators to generate offspring in order to increase the diversity of the particles analyzed and to improve the quality of the solutions. According to Zhang et al. (2008) in their hybrid PSO model with a two-point crossover operator satisfactory results are obtained with a significant improvement over the original PSO model.

Therefore, it is recommended as a future line of research to integrate the proposed model with other operators; it is specially encouraged to do so with a combination of one-point and three-point crossover operators based on what it is proposed by Nearchou (2004), who obtains better experimental results with this combination than with a two-point operator. Said combination has the advantage of coping better with more difficult problems in terms of amount of machines and jobs.

Lastly, the authors would like to propose the further research in newer optimization models like the Ant Colony Optimization in workload assignment. According to Dorigo & Stützle (2004) this method works with the assumption that each individual of a population is an artificial agent that builds in an incremental way the solution to the problem, therefore it could be of interest to apply it to the optimization problem in workload assignment.

Acknowledgements

The authors would like to thank Patricia Ramírez and Víctor Esquivel, both teachers at the Industrial Engineering Department of the University of Costa Rica, for their help towards the elaboration of this paper, their comments where of substantial help for the authors. We would also like to acknowledge the import made by Ph.D. Mario Villalobos Arias from the Pure and Applied Mathematics Research Center (CIMPA) of the University of Costa Rica that helped us understand better how the Particle Swarm Optimization model works.

References

R. Askin and J. Chen, 'Dynamic task assignment for throughput maximization with worksharing', *ELSEVIER*, 2004.

M. Dorigo and T. Stutzle, 'Ant Colony optimization', *Cambridge Mass: MIT press*, 2004.

A. Engerlbrecht, 'Particle Swarm Optimization: Velocity initialization', *IEEE*, 2012.

S. Helwig and R. Wanka, 'Theoretical Analysis of Initial Particle Swarm Behavior', *Springer-Verlag Berlin Heidelberg*, 2008.

S. Helwig, F. Neumann and R. Wanka, 'Particle swarm optimization with velocity adaptation', *University of Erlangen-Nuremberg*, 2009.

N. HU and P. Wang, 'An Algorithm for solving Flexible Job shop Scheduling Problems Based on Multi- Objective Particle Swarm Optimization.', *IEEE Computer Society*, 2010.

H. Huang and Z. Wang, 'Solving coupled task assignment and capacity planning problems for a job shop by using a concurrent genetic algorithm', *IEEE*, 2009.

J. Kennedy and R. Ebenhart, 'Particle Swarm Optimization', *IEEE*, 1995.

M. Millonas, 'Swarms, phase transitions and collective intelligence', *Artificial Life III.*, 1994.

T.P. Ryan, Modern Engineering Statistics. John Wiley & Sons Publishers, New Jersey, 2007.

A. Nearchou, 'The effect of various operators on the genetic search for large scheduling problems', *ELSEVIER*, 2015.

O. Romero, S. Romero and D. Muñoz, 'Introducción a la ingeniería: Un enfoque industrial, *CEGAGE Learning.*, 2006.

F. Tasgetiren, Y. Liang, M. Sevkli and G. Gencyilmaz, 'A particle swarm optimization algorithm for makespan', *ELSEVIER*, 2007.

C. Trelea, 'The particle swarm optimization algorithm: convergence analysis and parameter selection', *ELSEVIER*, 2003.
W. Yang, *Job Shop Matlab Code*. Wiley-Interscience Publishers, New Jersey, 2009.
C. Zhang, J. Ning and D. Ouyang, 'A hybrid alternate two phases particle swarm optimization algorithm', *ELSEVIER*, 2010.
C. Zhang, J. Sun, X. Zhu and Q. Yang, 'An improved particle swarm optimization algorithm for flowshop', *ELSEVIER*, 2008.

Biography

Javier André is an undergraduate student in Industrial Engineering at the University of Costa Rica. He is currently in his junior year. He earned his high-school diploma from the German School in Costa Rica, the Humboldt. Schule, San José, Costa Rica, in 2011. Contacts: javier.andreespinoza@ucr.ac.cr

Nathan Suchar is an undergraduate student in Industrial Engineering at the University of Costa Rica. He is currently in his junior year. He earned his high-school diploma from Institute Dr. Jaim Weizman, San José, Costa Rica. Contacts: nathan.suchar@ucr.ac.cr

Luis Jiménez is an undergraduate student in Industrial Engineering at the University of Costa Rica. He is currently in his junior year. He earned his high-school diploma from the Public School of Tarrazú, Tarrazú, Costa Rica, in 2011. Contacts: luis.jimenezquesada@ucr.ac.cr

Karol Esquivel is an undergraduate student in Industrial Engineering at the University of Costa Rica. She is currently in her junior year. She earned her high-school diploma from the Technical School Eng. Mario Quirós Sasso, San José, Costa Rica, in 2010. Contacts: karol.esquivel@ucr.ac.cr

Eldon Glen Caldwell Marín, Outstanding Service Award Recipient, IEOM Society, full professor (Cathedraticus), University of Costa Rica with over 20 years of teaching and research experience. He is teacher and mentor of the authors of this article on the topic of Operations Engineering at the University of Costa Rica. Bachelor and Master in Industrial Engineering at University of Costa Rica, also with a Master degree in Operations Engineering at ITESM, Mexico, Financial Analysis and Marketing of Services at Inter-american University of Puerto Rico, Health Services Management at UNED, Costa Rica and finally Ph.D. in Industrial Engineering at the University of Nevada, USA. Currently he is a doctoral researcher at the University of Alicante, Spain, and Academic Excellence Prized (2013 and 2014) researcher at the Ph.D. Program in Education at University of Costa Rica. His research interests include production scheduling, information retrieval, robotics and intelligent development of methodologies for implementing lean systems. Contacts: eldon.caldwell@ucr.ac.cr / egcm@alu.ua.es