Advances and Applications in Mathematical Sciences
Volume 17, Issue 1, November 2017, Pages 147-163

ONE-DIMENSIONAL CUTTING STOCK PROBLEM WITH SINGLE AND MULTIPLE STOCK LENGTHS USING DPSO

T. ASVANY[1], J. AMUDHAVEL[2] and P. SUJATHA[3]

[1,3]Department of Computer Science
Pondicherry University
Puducherry, India
E-mail: asvanytandabani@gmail.com
spothula@gmail.com

[2]Department of CSE
KL University
Andhra Pradesh, India
E-mail: info.amudhavel@gmail.com

Abstract

In our world the day to day life many industries use a significant problem which is termed as Cutting stock problem. Here both the single sheet length and multiple sheet length cutting patterns of the one-dimensional cutting stock problem is executed and the results compared with other algorithms like Genetic Algorithm (GA), Particle Swarm Optimization algorithm (PSO) and Cuckoo search. The main objectives of CSP problem are minimization of total wastage and minimization of number of stocks utilization. The proposed DPSO algorithm is discussed with 20 problem set in detail and results are compared with the other evolutionary algorithms, the DPSO algorithm attains the efficient experimental results.

1. Introduction

The past years, to solve this problem they mainly used two approaches namely heuristic and exact methods. The heuristic procedure was successfully applied standard stock length item have greater flexibility in a specific constraints of a problem and produce a good optimal solution and its

2010 Mathematics Subject Classification: 68N13, 97N50.
Keywords: Cutting stock problem (CSP), Discrete particle swarm optimization (DPSO), Genetic Algorithm (GA), Cuckoo Search (CS).
Received March 10, 2017; Accepted July 20, 2017

computational effort for many instances. Heuristic methods produce better and good, but not all the time essentially produces optimal solutions. Exact algorithms are mainly used branch and bound techniques and dynamic/linear programming. Linear programming relaxation is a well-known general technique which is strongly depending to find improved cutting planes of the branch and bound algorithm.

Cutting Stock Problem (CSP) concerns with realistic issues of how to cut a material based on required demand of customers from given stock with minimum wastage. Kantorovich was first well maintained the CSP problem and be a later the two people Gilmore and Gomory made a changes in the CSP problem. Some of the application domains of cutting stock problem include Textile industries, Wood, Rod, Glass and Steel industries etc., the main objective of CSP is to minimize the wastage and minimum number of sheets utilization is main objective of the problem.

In this paper we describe the one-dimensional cutting stock problem of smaller items with known dimensions. It consists of random selection and implementing different options to produce a demand with minimum wastage. The main objective of the CSP is to reduce the waste too minimum and maximize the material usage. The mathematical representation of the one dimensional CSP is:

$$\text{Minimize } \sum_{i=1}^{n} x_i. \tag{1}$$

Subject to

$$\sum_{i=1}^{n} a_{ij} x_i \geq b_i \quad \forall\, i = 1, 2, \ldots, m \tag{2}$$

$$x_i \geq 0, \text{ integer.} \tag{3}$$

Notations used are n-Number of cutting patterns, m-Number of items of different types, l_i-length of item $i, i = 1, \ldots, m, b_i$-demand of item $i, i = 1, \ldots, m, L$-Length of sheet a_{ij}-number of types of item i in j^{th} pattern. Let x_j is the number of items to cu according to the j^{th} pattern cut, $j = 1, \ldots, n$.

The objective function of the CSP is represented in equation one. Equation two represents the set of constraints make sure the total number of items produced demand with minimum waste. Equation three represent non negative integer of items to be cut.

Particle swarm optimization algorithm is used to solve the problems on combinatorial optimization concepts; it is substitute to other conventional heuristic approaches. PSO admires the swarm behaviour of bird flocking or fish schooling; it is a population based search method. PSO is basically considering a random search algorithm in natural evolutionary process and solving the complex and difficult optimization problems.

Use of this optimization mechanism has the positive behaviours such that fast convergence, high computation time, and easy in implementation. The different types of problems can be solved by the PSO variants. Compared to other evolutionary optimization algorithms PSO performs better solution and success rate.

This paper is organized as follows. In section 2 a review of previous work is discussed. The proposed approach elaborately says in section 3 with solution representation and followed experimental results on benchmark problems discussed in section 4. Finally, Section 5 concludes the proposed approach and the future research of possible issues.

2. Literature Review

Cutting Stock Problem grasps a wide range of research area since it is an *NP*-Hard problem. Different techniques were being proposed for solving CSP is found in literature. Some of the deterministic algorithms which are based on the techniques Dynamic Programming, Branch and Bound where been used to solve this *NP*-Hard problem. At this instantaneous the beneficiaries move to Meta-heuristic approaches which gains knowledge from previous experience.

Yaodong Cui [1] proposed (2006) an exact algorithm for two-segment cutting patterns to solve the problem in two stages. First stage refers guillotine sheer which cuts the plates into strips. Second stage refers punching of required items in desired dimension from the strips. At this juncture dynamic programming technique is used to choose optimal pattern and optimal strip on various lengths of segments.

Mobasher et al., [2] proposed mixed integer linear program which concerns Column generation procedure based heuristic algorithm and two local search algorithm on considering total setup cost for solving cutting stock problem solved a nonlinear cutting stock problem.

Moretti, et al., [3] proposed a Nonlinear Cutting Stock Problem model to minimize the number of different patterns and objects. They club problem linearization, column generation, Augmented Lagrangain method and a heuristic method for solving Nonlinear Cutting Stock problem. In this method heuristics were being applied to generate integer values which represent the solution.

The literal at ease makes another study on CSP by Michael Adaowicz, et al., [4]. A Solution of the Rectangular Cutting-Stock Problem which holds the objective of minimizing the wastage was proposed for two dimensional (Rectangular) Cutting Stock problems. The way of solution described at this point avoids exhaustive search procedures by employing an advance utilizing a constrained dynamic programming algorithm to place out groups of rectangles called strips.

Yaodong Cui, et al. [5] presented recursive branch and bound algorithm mainly concerns the material utilization which are supposed to be maximized. This technique thought-out every plate as block, which block is selected with the intention of crammed the bottom left corner. Vertical and horizontal cuts are included in the regions.

One-dimensional cutting stock problem Cerqueira et al. [6] developed heuristic approaches to minimize the solution of different patterns. Based on order the items split two put out of joint group. Depends in the lead the grouping they generate a pattern it must be with minimum wastage. Pattern reducing procedure applied, whose demands are not satisfied.

Yaodong Cui [7] developed a solution for manufacturing industries. They used to cut circular and sectorial blanks. Implicit enumeration method to determine the optimal combination of blank rows in the strips, the strip numbers and directions in the pattern. In the patterns more than one row of equal blanks can become visible in a strip. A strip is in one of the two at 90-degree angle directions, namely X- or Y-direction.

Haessler [8] planned (1975) pattern generating heuristic that one after another adds latest cutting patterns to the current solution awaiting all demand is met. In each step, the method selects a cutting pattern, whose trim loss is tiny and occurrence is high.

Bioinspired algorithms like ACO Amudhavel et al. [17] introduced various related cloud environment Raju et al. [21, 22], web services Amudhavel et al. [18, 19, 20] and other domain related prolems.

3. The Proposed Algorithm DPSO

The concept of fish schooling or bird flocking used in the algorithm called Particle Swarm Optimization. The modified version of Particle Swarm Optimization is Discrete PSO (DPSO) this technique is mainly used for discrete variables. The real world optimization techniques used to develop a variable and arrange a particle in a search space called discrete. In the complex search spaces, the individuals are used to find the optimal solutions.

This DPSO first developed by Kennedy and Eberhart (1997) and used only for binary particle values. The main usage of the DPSO developed to solve the discrete problems. The DPSO is mainly concentrate velocity and position of the particles. Compared to other optimization techniques like GA etc., this DPSO has lot of advantages. The DPSO has excellent memory power compare with GA. The working of DPSO remembers their own neighbourhood best value and at the same time previous best value remembers each and every particle have their own. Compared with *GA* the DPSO implementation is very easy as well as limited adjust in the parameters and its structure is simple. In the DPSO every particle maintains the information of some most successful particle information to efficiently improve the swarm diversity.

The DPSO algorithms broadly classified and implemented in five different categories. The first one is DPSO with crossover and mutation techniques (Lian et al., 2006 [9]; Lian et al., 2008 [10];) and second one is binary valued DPSO (Kennedy and Eberhart, 1997 [11]) and third one is modified continuous PSO with smallest position value rule (Tasgetiren et al., 2007 [12]) and fourth one is DPSO with dummy variable to transition from combinatorial to discrete state and vice-versa (Jarboui et al., 2007 [13]) and the fifth one is various DPSO models which includes other variety such as fuzzy DPSO (Anghinolfi and Paolucci, 2009 [14]).

3.1. Solution Representation

In our proposed algorithm DPSO randomly generated the population are used. The Solution representation of the Discrete PSO for Cutting Stock Problem is $\mathrm{Par}_{11} \ldots\ldots\ldots\ldots\ldots \mathrm{Par}_{1n}$. Where the number of dimensions in a particle will be the total number of items in the dataset. The order of the items in the particle plays a main role in the representation process.

The description of the algorithm follows:

Step 1. Initialization

Step 1-1. Initialize the particle where "n" represents the dimension and "m" represents the number of population in iteration.

$$\text{Particle} = \begin{bmatrix} \mathrm{Par}_{11} \ldots\ldots \mathrm{Par}_{1n} \\ \mathrm{Par}_{m1} \ldots\ldots \mathrm{Par}_{mn} \end{bmatrix}.$$

Step 2. The matrix elements are randomly generated and satisfy the conditions

$$P_{ij} \leftarrow \{1, \ldots\ldots\ldots, d\},$$

where "d" represents dimension and "ij" represents the i^{th} particle in the j^{th} iterations. The Initialization of Velocity is $V = \{v_1, v_2, \ldots\ldots\ldots, v_n\}$. The matrix elements are generated randomly and the following matrix shows the representation

$$V = \begin{bmatrix} \mathrm{Vel}_{11} \ldots\ldots \mathrm{Vel}_{1n} \\ \mathrm{Vel}_{m1} \ldots\ldots \mathrm{Vel}_{mn} \end{bmatrix}.$$

Random position matrix is updated in each and every particle the initialization of random velocity is

$$V = V_{11} \ldots\ldots\ldots V_{1n}$$

$$\text{Velocity} \rightarrow R,$$

where "R" represents the Real Number. In the DPSO algorithm two parameter values are used. The value of parameter is represented by

$$C_1 = 0.5 \text{ Cognitive Parameter},$$

$$C_2 = 0.5 \text{ Social Parameter}$$

Step 3. Repeat the same process

Step 4. Calculate the fitness population for the use of following equations

$$\text{Fitness formula} = \text{Calculate } W$$

$$= lb + (ub - lb) * \frac{(\# \, iter \; i)}{(\# \, iter - 1)}.$$

Where the above formula representation is

$$i = \text{Current iteration}$$

$$\# \, iter = \text{Total number of best}$$

$$ub = \text{Upper bound}$$

$$lb = \text{Lower bound.}$$

For each and every particle value is calculated and check the condition the fitness value is less than the local best particle value.

$$if \; (\text{fitness } (i) < P \text{ best } (i))$$

$$P \text{ best } (i) = \text{fitness } (i)$$

end.

Some particles of local best position value are lesser than the global best the value is updated local best to global best.

$$g \text{ best } = \min \{P \text{ best}_{ij}\}.$$

Here the "i" represents the iteration number and "j" represents the total number of population. The new velocity calculation the formula is

$$\text{New Vel}_i = W * (V_i) + C_1 * \text{rand}(\;) * (P_i - \text{Par}_i) + C_2 * \text{rand}(\;) * (P_g - \text{Par}_i)$$

$$\text{New Par}_i = \text{Current Par}_i + \text{New } V_i.$$

Step 5. Until terminate the condition satisfies

Step 6. end

The workings of proposed DPSO algorithm have a modification in pbest and gbest values. In the proposed algorithm, from the large search space random positions of velocity and swarm values are initialized. Then, Calculating the fitness value for each and every particle has been done. In step 3, latest fitness value is compared to the previous best value. The steps to update the pbest value for each iteration of the particle is as follows: let us assume for example the sequence of swap operator randomly produced values [(2, 5), (4, 6), (7, 9)]. The values already stored in the array assume the sequence is [1 2 3 4 5 6 7 8 9 10] after applying the swap operator produced the set of values to the sequence it generates combination of the swap sequence1 for pbest.

$$\begin{aligned} \text{Position update} &= [1\,2\,5\,3\,4\,6\,7\,8\,9\,10] \\ &= [1\,6\,2\,5\,3\,4\,7\,8\,9\,10] \\ &= [1\,9\,6\,2\,5\,3\,4\,7\,8\,10]. \end{aligned}$$

The produced values of swap sequence1 for pbest, the best values are update to the population to generate the next iterations. For swap sequence 2 the best values of sequence 1 are updated in sequence 2 the current gbest value calculates the fitness again. Fr satisfies the iteration compared the results shown which value is updated in gbest.

4. Experimental Results

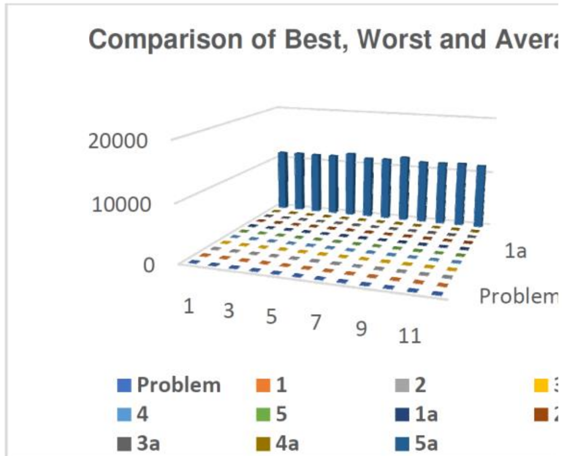
The proposed algorithm was implemented in MATLAB 8.3 and computed results in individual system. Here benchmark results are collected from different instances like some journal papers and some data set in library, all instances are related to the cutting stock problem. Randomly generated instances and computational trials were performed. Each problem instances can have tested maximum of 10 trails in 100 iterations for small problems. Each and every trail can take one best value up to the maximum level of iterations. For each iteration best value is chosen and compared to all trails, minimum value is finally selected as best result. In this Cutting stock problem the related dataset is implemented in evolutionary algorithms namely *GA*, Cuckoo, PSO and DPSO. The most popular basic above mentioned algorithms are implemented in mat lab and tested in collected dataset, basic parameters of the algorithm like Best, Worst, Average, Convergence, Average Convergence and Computational Time results.

First the authors tested CSP problem in Genetic algorithm, the basic *GA* parameters are probability of Mutation is 20%, probability of Crossover is 60% and the Selection probability is 80%. Second here chosen parameters of Cuckoo algorithm, are used in this probability of abonded are 0.5 and the value of gamma is 0.01. Third algorithm PSO the normal standard parameters used in the particle swarm optimization algorithm is LB, UB, C1 and C2. The Lower Bound value set as 0.4, and the upper bound value is 0.9, after the parameters C1 is 2 and C2 is 0.5. Finally, the proposed DPSO algorithms have some changes in the inertia coefficient and nearest particle values also changes.

CSP problem datasets are taken from Hinterding and Khan [15], Liang et al., [16] for testing sample 20 dataset problems are included. In the benchmark problem two different sets of problem instances are tested. The first set of problems 1-5 are multiple stock length CSP's and 1a-5a are single stock length problems. The first set of problems is small size in range so here tested only for minimum of 100 iterations itself. For minimum iterations the proposed DPSO taken the exact result of the problems. The second set problems also have ten datasets but this problem have larger in size and difficult to cutting because of the number of items required in high range. The same 6-10 sets are multiple and 6a-10a is single stock length and maximum of 500 iterations were performed in this part.

First ten comparison dataset 1-5 the multiple stock sizes is minimum in all stages like 0 wastage. In single stock size 1a-5a some variations occur.

Graph 1 shows comparison of best, worst and average of single and multiple dataset problems. Here while comparing 1a-4a the same repeated values occurs for all the four tested evolutionary algorithms. Only value changes in the last problem 5a, the repeated values come under in the case of average in *GA* and Cuckoo algorithm. For taking the dataset problems 6-10 have large number of requested stock items, so the value must be different in each iteration. In this 6-10 and 6a-10a the dataset items are tested in maximum number of 500 iterations.



Graph 1. Both Single and Multiple Dataset Comparison of Best, Worst and Average.

The *GA* algorithm values should not change in the worst case. For each iteration ranges 100th to 500th range the worst case value of *GA* doesn't change. It constantly had shown single same value for minimum to maximum ranges. The Genetic algorithm doesn't change the value for iterations in worst case, comparing to the PSO worst case also have the same status.

Furthermore, the four evolutionary algorithm evaluate the performance, the authors tested the ten problems (i.e., problems 6-10 and 6a-10a) are compared. According to the results the best, worst and average values were compared, the DPSO algorithm expressively better than the other three evolutionary algorithms for all the dataset problems with multiple stock lengths (dataset problems 6-10), the single stock length (dataset problems 6a-10a) problems also performed better in results. The four algorithms are tested in matlab code the representation of iteration value changes in trails. The best value for above mentioned four algorithms, single stock length dataset problem of 6a to 10a the values in DPSO occurs 189, 50, 212, 142 and 400. In the same problem the Cuckoo value is 192, 68, 256, 168 and 426 and PSO value is 196, 72, 262, 170 and 434. The GA value is very high compared to other three algorithms. The worst case particularly the Cuckoo algorithm value doesn't change in most of the trails. PSO and GA the value changes in the trails but compared to the DPSO algorithm the value is very high.

Table 1. The best results for problem 1-5(100 iterations) and 6-10 over 500 iterations for CSPs, where “Found” indicates the average number of occurrences the best solution and “Time” indicates the maximum number of time taken for the best solution.

Problem No	Best	Worst	Average	Found	Time
1	0	0	0	10/10	8.23
2	0	0	0	10/10	9.25
3	0	0	0	10/10	10.15
4	0	0	0	10/10	11.52
5	0	0	0	10/10	9.24
6	98	104	100	7/10	18.21
7	30	38	33	8/10	19.86
8	92	124	96	7/10	20.45
9	102	128	106	9/10	22.15
10	189	202	192	7/10	24.85

The existing novel EP algorithm is used for CSP and results of 10a dataset are reduced 1037.20 to 643.60. In our proposed algorithm DPSO algorithm results of 10a problem is reduced to 643.60 to 400. The problem of 9a instances is also have some minimum values when compared to previous algorithms, total wastage of cutting a sheet is reduced 730.00 to 432.40 using novel EP algorithm. The proposed DPSO algorithm tested the cutting pattern reduced from 432.40 to 142 the total wastage of single stock sheet and multiple stock sheet is 102 reduced in total wastage. Dataset problem 9a and 10a (single stock sheet) the total number of stock utilization is 150 and 204.

The DPSO algorithm the best solution occurs in the progress of increasing the inertia values have some changes while computation time and best solutions occurs of better performance.

Table 1 represents the multiple stock sheet usage for dataset problems, in that table last column shows computation time for problems 1-10 taken time duration. The maximum time duration it took only for problem 10, time is

24.85. First 1-5 problems values show null, so problems 6-10 have the deviation in the occurrences of best solution found.

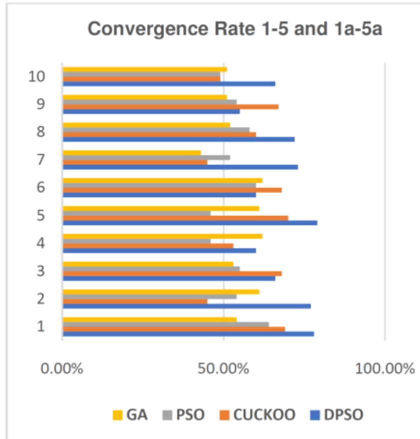
Table 2. The best results for problem 1a-5a (100 iterations) and 6a-10a over 500 iterations for CSPs, where "Found" indicates the average number of occurrences the best solution and "Time" indicates the maximum number of time taken for the best solution.

Problem No	Best	Worst	Average	Found	Time
1a	3	5	4	8/10	37.19
2a	13	17	14	8/10	58.83
3a	0	0	0	10/10	65.56
4a	11	13	12	8/10	67.00
5a	11250	11330	11280	7/10	137.90
6a	189	210	192	7/10	181.55
7a	50	74	62	6/10	177.90
8a	212	236	220	5/10	335.93
9a	142	156	150	4/10	292.75
10a	400	436	416	5/10	569.21

Table 2 result shows single stock sheet problems, here first set of problem 1a-5a taken some deviation in values, next set 6a-10a problems take larger computation time. The highest time taken for dataset problem 10a is 569.12. Proposed DPSO algorithm compared to other three evolutionary algorithm results, minimum time duration for best solution occurs.

Compared to the occurrences found maximum number of best solution found in proposed algorithm repeatedly. The Best, worst and Average value of problem 1a-4a single stock sheet value for other three evolutionary algorithms also produced same results. The computation time and number of times best solution repeatedly produced only have better results in proposed DPSO algorithm.

Graph 2 shows both single stock sheet and multiple stock sheet dataset problems for 1-5 and 1a-5a, the convergence rate for all the above mentioned evolutionary algorithms.

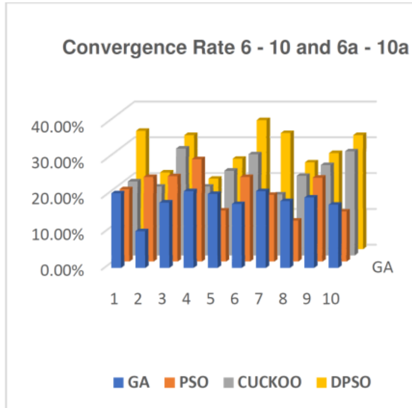


Graph 2. The evolutionary algorithm results for problems 1-5 and 1a-5a, both Single and Multiple Dataset Comparison of Convergence Rate.

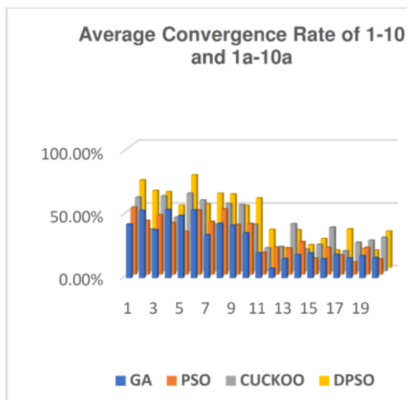
The convergence rate percentage is calculated by use of maximum number of iterations and minimum numbers of generations are worked. The graph based solution representation of the convergence rate is purely depending on iteration based values. The DPSO value is compared to all other three evolutionary algorithms; the proposed algorithm reaches performance wise better and convergence rate produced a good value.

The proposed DPSO algorithm techniques maintains efficiency of the method is one of best factor is to improve the speed of convergence. Here to improve convergence rate in a successive iterative method for approximations and useful to increase the rate of convergence.

The convergence rate for the 6-10 and 6a-10a the problem instances are described clearly. Compared to the entire above mentioned algorithm the DPSO gives the better results.



Graph 3. The evolutionary algorithm results for problems 6-10 and 6a-10a, both single (100 iterations) and Multiple (500 iterations) Dataset Comparison of Convergence Rate.



Graph 4. The overall best results of the Average Convergence Rate of four evolutionary algorithms for dataset problems 1 to 10 (Multiple stock sheets) and 1a to 10a (Single stock sheets) over 100 iterations to 500 iterations.

Convergence rate of PSO and Cuckoo algorithm produces same result at the beginning of the iterations; later cuckoo algorithm beat the PSO algorithm while changing the levy distribution. The proposed DPSO algorithm higher in range while compared to all other evolutionary algorithm mentioned in this paper.

The average convergence rate of 20 dataset problems and their values described in Graph 4. For example, the same dataset problems implemented in first set of algorithm namely *GA* and PSO the average convergence ratio percentage is having maximum of 12 in the range difference. The second set algorithm PSO and Cuckoo algorithm compared in the range it has some deviation, maximum number of deviation occurs in range is 8 to 10. While Cuckoo algorithm are compared to DPSO, first and second set of algorithm values in higher in range and the proposed DPSO algorithm only have minor changes. An experimental result of the proposed algorithm in range is 4 to 8.

For calculating Average Convergence Rate minimum of two values needed. First one is maximum total number of iterations and second one is minimum total number of generations used in the maximum number of iterations. It is very interesting to make a note on different results obtained by the four evolutionary algorithms. It appears that a small effect on the total number of stocks utilization, while they had an important change on the total wastage. When stock utilization was considered, the total wastage also came down considerably for some dataset problems.

5. Conclusion

A modified DPSO algorithm which is capable of solving optimization problems with discrete solutions. Here presents a discrete PSO algorithm to solve Cutting Stock Problem. Authors done the work in two phases. In the first phase, a proposed DPSO algorithm solves the single stock sheet problems. In the second phase, multi stock sheet dataset problems are solved. Also, it was observed that the convergence time and the number of iterations are randomly chosen variable which is principally rest on such parameters of algorithms and swarm based initialization in the beginning. The proposed DPSO algorithm performs best results in 1). Effective in improving the total material utilization 2). It is able to reduce the total wastage of material 3) the first and second point reaches in a reasonable computation time.

References

[1] Y. Cui, Generating optimal multi-segment cutting patterns for circular blanks in the manufacturing of electric motors, European Journal of Operational Research 169(1) (2006), 30-40.

[2] A. Mobasher and A. Ekici, Solution approaches for the cutting stock problem with setup cost, Computers & Operations Research 40(1) (2013), 225-235.

[3] A. C. Moretti and L. L. D. Salles Neto, Nonlinear cutting stock problem model to minimize the number of different patterns and objects, Computational & Applied Mathematics 27(1) (2008), 61-78.

[4] M. Adamowicz and A. Albano, A solution of the rectangular cutting-stock problem, IEEE Transactions on Systems, Man and Cybernetics (4) (1976), 302-310.

[5] Y. Cui, Y. Yang, X. Cheng and P. Song, A recursive branch-and-bound algorithm for the rectangular guillotine strip packing problem, Computers & Operations Research 35(4) (2008), 1281-1291.

[6] G. R. L. Cerqueira and H. H. Yanasse, A pattern reduction procedure in a one-dimensional cutting stock problem by grouping items according to their demands, Journal of Computational Interdisciplinary Sciences 1(2) (2009), 159-164.

[7] Y. Cui, Generating optimal T-shape cutting patterns for circular blanks, Computers & Operations Research 32(1) (2005), 143-152.

[8] R. W. Haessler, Controlling cutting pattern changes in one-dimensional trim problems, Operations Research 23(3) (1975), 483-493.

[9] Z. Lian, X. Gu and B. Jiao, A similar particle swarm optimization algorithm for permutation flow-shop scheduling to minimize makespan, Applied Mathematics and Computation 175(1) (2006), 773-785.

[10] Z. Lian, X. Gu and B. Jiao, A novel particle swarm optimization algorithm for permutation flow-shop scheduling to minimize makespan, Chaos, Solitons & Fractals 35(5) (2008), 851-861.

[11] J. Kennedy and R. C. Eberhart, A discrete binary version of the particle swarm algorithm, In Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation, 1997 IEEE International Conference on (Vol. 5, pp. 4104-4108), IEEE.

[12] M. F. Tasgetiren, Y. C. Liang, M. Sevkli and G. Gencyilmaz, A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem, European Journal of Operational Research 177(3) (2007), 1930-1947.

[13] B. Jarboui, M. Cheikh, P. Siarry and A. Rebai, Combinatorial particle swarm optimization (CPSO) for partitional clustering problem, Applied Mathematics and Computation 192(2) (2007), 337-345.

[14] D. Anghinolfi and M. Paolucci, A new discrete particle swarm optimization approach for the single-machine total weighted tardiness scheduling problem with sequence-dependent setup times, European Journal of Operational Research 193(1) (2009), 73-85.

[15] R. Hinterding and L. Khan, Genetic algorithms for cutting stock problems: with and without contiguity, In Progress in Evolutionary Computation (pp. 166-186), Springer Berlin Heidelberg, (1995).

[16] K. H. Liang, X. Yao, C. Newton and D. Hoffman, A new evolutionary approach to cutting stock problems with and without contiguity, Computers & Operations Research 29(12) (2002), 1641-1659.

[17] J. Amudhavel, S. Padmapriya, R. Nandhini, G. Kavipriya, P. Dhavachelvan and V. S. K. Venkatachalapathy, Recursive Ant Colony Optimization Routing in Wireless Mesh Network, In Proceedings of the Second International Conference on Computer and Communication Technologies (2016), (pp. 341-351). Springer India.

[18] J. Amudhavel, U. Prabu, N. Saravanan, P. Dhavachelvan, R. Baskaran and V. S. K. Venkatachalapathy, A Distributed Spanning Tree-Based Dynamic Self-Organizational Framework for Web Server, In Proceedings of the Second International Conference on Computer and Communication Technologies (2016), (pp. 333-339), Springer India.

[19] J. Amudhavel, U. Prabu, P. Inbavalli, N. Moganarangan, V. Ravishankar, R. Baskaran and P. Dhavachelvan, Survey and Analysis of web service Composition Strategies: A state of art performance study, Indian Journal of Science and Technology 9(11) (2016).

[20] J. Amudhavel, U. Prabu, P. Dhavachelvan, N. Moganarangan, V. Ravishankar and R. Baskaran, (2015, April). Non-homogeneous hidden Markov model approach for load balancing in web server farms (NH 2 M 2-WSF), In Communication Technologies (GCCT), 2015 Global Conference on (pp. 843-845), IEEE.

[21] R. Raju, J. Amudhavel, N. Kannan and M. Monisha, Interpretation and evaluation of various hybrid energy aware technologies in cloud computing environment-*A* detailed survey, In Green Computing Communication and Electrical Engineering (ICGCCEE), 2014 International Conference on (pp. 1-5), IEEE. (2014).

[22] J. Amudhavel, V. Brindha, B. Anantharaj, P. Karthikeyan, B. Bhuvaneswari, M. Vasanthi and D. Vinodha, A survey on intrusion detection system: State of the art review, Indian Journal of Science and Technology 9(11) (2016).