Subtask Scheduling to Minimize Total Weighted Completion Time for Costumer Order Scheduling

Suma T, Research Scholar, REVA University, Faculty New Horizon College of Engineering , Bangalore, India, sumatbrijesh@gmail.com

Murugesan R, REVA University, Bangalore, India, contactmurugu@gmail.com

Abstract: In the current scenario, for many manufacturing industries, it is challenging task to commit a due date for customer order. To address this key issue, many manufacturing industries are adopting multi variety manufacturing mode. In this mode a number of orders are composed of several product types, these product types have to be scheduled on a different machines, each one capable of processing a single product type and all order should be delivered on time simultaneously. In this study consider a subtask scheduling for customer order scheduling problem with optimization objective to minimize total weighted completion time. A tree structure for order scheduling is discussed with an example and Heuristic based Particle Swarm Optimization (PSO) Algorithm is used for optimal scheduling.

Key words – completion time, manufacturing industries, order scheduling, optimization, PSO Algorithm, task scheduling.

I. INTRODUCTION

Many researchers have been done on customer order scheduling and literature studies shown that even for two machines it is NP- hard problem [1]. As a consequence different Heuristic Algorithms and Models are proposed with optimizing objectives: Minimizing completion time for order scheduling, Minimizing total tardiness, lateness and so on. Currently many researchers are implementing subtask scheduling to achieve above said optimizing strategies[2]. This paper made an attempt to use subtask scheduling to minimize completion time for order scheduling. Subtask is a process in which a task is divided into number of subtasks and these subtasks are assigned to the machines capable of handling it. The task is said to be completed when all of its subtasks are completed. In the process of subtask scheduling g many relations are formed among the subtasks which give rise to number of structures including fork structure used to describe the types of the process. These structures are transformed into tree structures which provide only the aggregate relationship making easier to describe the process type. This paper proposed a problem in which each task is divided into less than or equal to 5 subtasks and the components from subtask say A is used for two consecutive subtasks say B & C which forms fork structure then we convert fork structure into tree structure by dividing the subtask A into two subtask of similar types Say A1 and A2 note that the components from A1 and A2 are utilized for B and C respectively. The detailed example and properties of tree structure is discussed in the next section.

This paper proposes subtask scheduling method to minimize the overall completion time for order scheduling in order to achieve the above said optimizing objective we used Heuristic Particle Swarm Optimization (PSO) algorithm. Parameters are defined, simulation experiments are conducted and results are obtained using MATLAB. Obtained results are verified the effectiveness of the Algorithm.

The main notations and definitions are given in table 1.the rest of the paper is organized as follows. Section II discuses related works done on minimizing overall completion time for order scheduling. Section III describes tree formation for subtask scheduling and Entities. Section VI discuses implementation of PSO algorithm and Section V illustrates simulation settings and results discussion and at last section VI gives the conclusion.

II.RELATED WORKS

Julien and Magazine [4] were the first to define customer order scheduling problems. Later many research has been conducted and important results were discussed.[5]conducted research on order scheduling and proved that minimizing total completion time and tardiness are polynomial solvable and they are unary Np-hard for the machines two and greater than two. However most of the research considered on an order scheduling problem with a single machine and identical parallel machines.[6]considered a multiple job problem with two objectives ,one is to minimize total setup time and second to minimize the customer order ranges and he developed Branch and Bound Algorithm to solve aforementioned optimization problem.[7] show that minimizing total completion time for number of machines greater than are equal to two is strongly NP-hard .[8]proved that minimizing total completion time for three or more machines is strongly NP-hard and proposed two heuristic namely Shortest processing time applied on the machine with largest load(SPTL) and earliest completion time to find near optimal solutions(ECT).[9] Analyzed five heuristic methods to minimize total completion time in which each order has a weight and release date.[10]studied on customer order scheduling with the objective of minimizing the weighted sum of customer order delivery

DOI : 10.18231/2454-9150.2018.0189

times and show that the problem is unary NP-hard and solved several methods for optimal scheduling.[11] Also studied order scheduling with the objective of minimizing the total completion time and given heuristic algorithm to address the problem. Further several studies on minimizing the total completion time for order scheduling is conducted Wang and Cheng show that the problem is unary NP-hard and proposed heuristic method to address the problem. Yoon and Sung proposed Branch and Bound method to solve the problem. In recent year [12] studied dynamic customer order scheduling with different product types in the stochastic medium and derived several optimal properties and proposed three heuristic methods.[13]considered prioritized customer orders arriving at several stations dynamically to minimize the output.[14] proposed a mixed integer linear programming model using linearization technique to define the problem size that can be used to solve the optimality and in addition to that he developed nested partition algorithm to solve large scale problems. Numerical results show that the proposed model and algorithm can provide optimal solutions within the reasonable time. However, only few researches has been done on subtask scheduling to minimize over all completion time for order scheduling and also implementation of PSO algorithm to obtain optimal scheduling .hence this paper made an attempt to implement subtask scheduling model and PSO algorithm for customer order scheduling with the optimization objective of minimizing overall completion time.

Table-I. Main Notations and Definitions:

Notations	Definitions
N_o	Set of orders , $N_o=\{1,2,3,\ldots.j\}$
N_m	Set of machines, $N_m=\{1,2,3\ldots.i\}$
$M_{i,j}$	Machine capable of performing jth order
W_j	Weight of order j
N_{st}	Number of types of subtask
$(ST)_n$	Subtask of n^{th} type
$Pt_{i,j}$	unit production time
$(WL)_{i,j}$	Workload left to be handled in $M_{i,j}$ is $(WL)_{i,j}$.
N_p	total number of process type
P_m	process of m^{th} type
N_T	number of task to be handled
T_k	K^{th} task
Pro_k	process type for T_k
$L_{m,n}$	is the number of layers from $(ST)_n$ to the tree node in P_m.

Fig.1 Subtask Model:

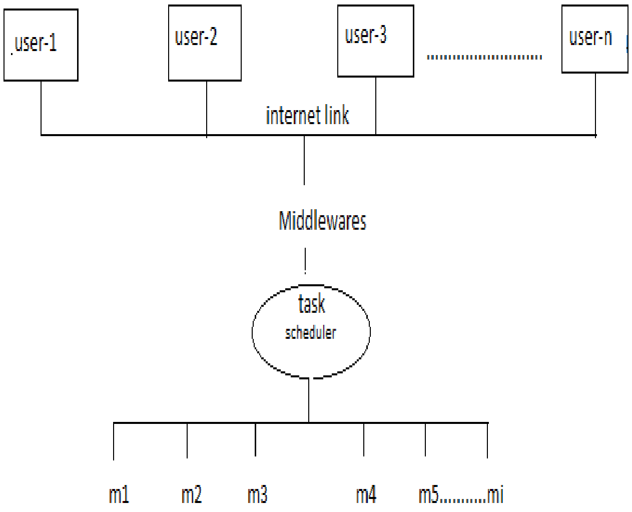


Fig1: The working procedure of subtask model is as follows; A order request from customer is received and sent to the task scheduler .the scheduler begins to work accordingly or on demand .task scheduler performs two functions (i) receiving the order request (ii) analyzing and decomposing each task into subtasks and allocating a machine to each subtasks . Each order consists of two or more products types after analyzing the each task and the current status of the machine where each machine is dedicated to produce only one type of product the scheduler allocates a machine for each subtask. Machine starts handling all the subtasks belongs to them respectively.

A. Entities

1. Given $0 \leq n < N_{ST}$ (N_{ST} total number of subtask types), $(ST)_n$ stands for minimal element handled in machines. A matrix $[N_{ST} \times N_m]$ between (ST)n and machine type $M_{*,j}$ is constructed as shown in **Table II.** If a machine type can handle subtask type the element of the matrix is 1 and otherwise 0.

2. For $0 \leq j < N_m$ (total number of machines types) considering the subtasks a unique metric function (UMF) is introduced as reference to measure the time for handling subtasks. A small component is selected and assigned time for handling it to UMF. the properties of $M_{i,j}$ (machine capable of performing j^{th} order) includes ; A unit production time $Pt_{i,j}$ is time needed by $M_{i,j}$ to produce one UMF, The workload left to be handled in $M_{i,j}$ is $(WL)_{i,j}$.

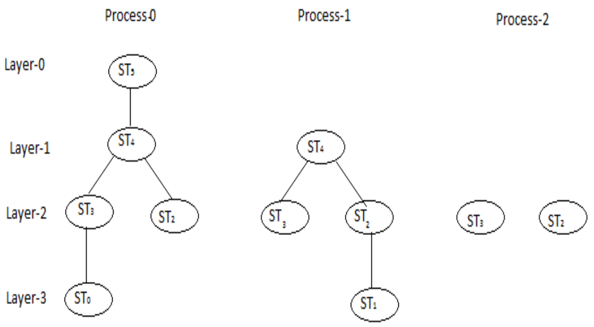
3. Given $0 \leq m < N_p$ (total number of process type), P_m process of m^{th} type is the relation structure between subtask and procedure for final product. Due to complex relations between the subtasks there exist so many structures, used to define the process types. These structures are transformed into tree structures which give rise to aggregation relationship making easy to define process types.

DOI : 10.18231/2454-9150.2018.0189

Table II. Function Mapping B/W machine type and subtask type:

	M1	M2	M3	M4	M5
ST_0	1	0	0	1	1
ST_1	0	1	0	1	0
ST_2	1	0	0	1	1
ST_3	0	0	0	0	1
ST_4	1	0	1	0	0
ST_5	0	1	0	0	1

Fig.2 Example of formation of tree structure of subtask process:



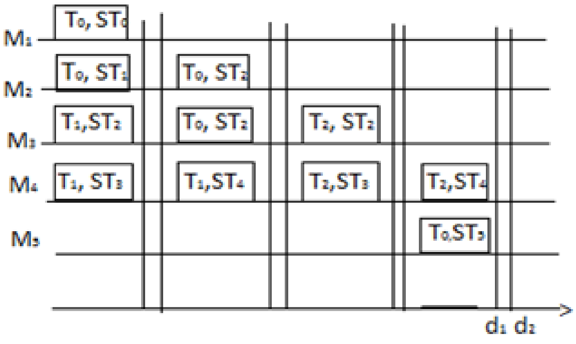
A detailed description is as follows: Given $0 \leq k < N_T$ (number of task to be handled), T_k k^{th} task , If Pro_k (process type for T_k) contains $(ST)_n$ in T_k is scheduled to $M_{i,j}$, we assign a allocation sign $A_{k,n,i}$ =1 and 0 otherwise. We assign a Boolean variable $BP_{m,n}$=1 if $(ST)_n$ exists in P_m and $BP_{m,n}$=0 otherwise and $L_{m,n}$ is the number of layers from $(ST)_n$ to the tree node in P_m. subtask which finishes last in the process is called root node of the tree. And each subtasks are handled according to the specification of the process. The mean layer position for $(ST)_n$ in all the process is

$$(STL)_n = \frac{\sum_{m=0}^{N_p-1} BP_{m,n} L_{m,n}}{\sum_{m=0}^{N_p-1} BP_{m,n}}$$

$(ST)_n$ with greater $(STL)_n$ tends to be handled first in the process.

Based on **Fig.1** and **Fig.2** there are N_p=3, N_{ST}=5 and N_m =5.process types for $task_0$, $task_1$, $task_2$ and $task_3$ are $process_0$, $process_1$, $process_2$ and $process_2$ respectively. Subtask scheduling is shown below in **Fig.3**

Fig.3 Gantt chart of subtask handling:



Gantt chart of subtask handling

B. Optimization Strategy

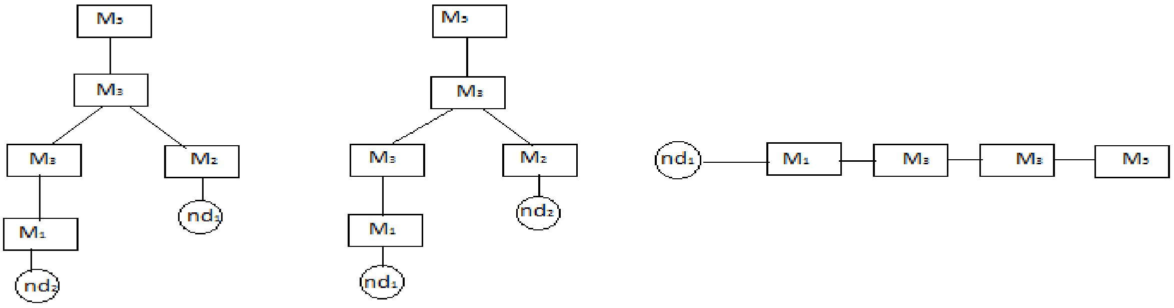
The overall completion time of a subtask for a customer order is the production time in a machine and the overall production time is defined by the subtask which finishes at last. From Fig.3 The subtask in M_5 finishes work last at d_1. The subtask ST_4 of T_0 in M_5 should wait until the completion of all its previous subtasks, the overall completion time for M_4 is less than that for M_5. That means overall production time cannot be calculated by the view of single machine. From the view of subtask Minimized Delay Scheduling exists when all the subtasks finishes at the end are handled without idle waiting. Based on Fig.3 we calculated shortest completion time, and demonstrated in the following example.

1. Machine Tree Construction: If any workload remains in a machine add a node ($nd_{i,j}$) which indicates current load to the corresponding node in machine tree.

2. Weight of the edges of machine tree: d_i ($1 \leq i \leq 6$) weight of the edge is assigned. We calculate the sum d_i of time for handling the subtask without node and in node.

3. Machine tree trimming: we select the trunk having highest weight in machine tree. The optimal completion time Ck for T_k is the sum of the weight of the edges in the trunk for the example in **Fig.4** is $d_1+d_3+d_4+d_6$. The overall completion time that is possible is the Maximum of C_k hence the optimization objective is Min $(Max(C_k))$ s.t $0 \leq k \leq N_T$.

Fig.4 Example for overall completion time:



Example for overall completion time calculation

DOI : 10.18231/2454-9150.2018.0189

IV. IMPLEMENTATION OF PSO FOR OPTIMAL SCHEDULING

The subtask scheduling problem can be considered as an optimal assignment problem for bipartite graph, in which some vertices are subtasks and remaining are machines, each machine can be assigned with more than one subtask, hence it is an NP-hard problem. Many algorithms are proposed to solve this assignment problem, including Genetic Algorithm (GA)[19], Ant Colony Optimization(ACO).and other Heuristic Algorithms[20].This paper implemented PSO to solve this assignment problem.

PSO is a population based algorithm. The population is called Swarm and its individuals are called particles. It was developed in the year 1995 by Kennedy and Eberhart [17]. PSO algorithm searches for the best solution over the complex space Iinitially PSO algorithm creates the initial particle swarm ,it initializes a swarm of particle randomly in the available solution space making each particle an available solution of the optimization problem. The target function determines the fitness value through the target function .Each particle will move in the space of the solution with its direction and distance determined by speed. The general particle will move following the best current particle obtaining the best solution by searching generation by generation. The particle will trace two limited values one of which is the best solution P_{best} found by particle itself and other is the best solution G_{best} found so far by general group Swarm [18].

The algorithm starts by encoding the problem to produce a number of populations, in each Swarm the particles are indexed by k* N_{st}+n, represents $(ST)_n$ in T_k , assigned with i* N_m +j ,represents the $M_{i,j}$ which will handle $(ST)_n$ in T_k. If $(ST)_n$ is not in T_k ,the particle is assigned with negative value. Next particles are initialized with random positions and velocities the equations are given below

$$V_{id} = WV_{id} + C_1 \ r_1(P_{id} - X_{id}) + C_2 r_2(P_{Gd} - X_{id}) \quad \ldots\ldots\ldots\ldots (1)$$

$$X_{id} = X_{id} + V_{id} \quad \ldots\ldots\ldots\ldots (2)$$

Where , V_{id} is the velocity of the particle that is distance to be travelled by particle i from its current position , X_{id} particle position, P_{id} local best solution, P_{Gd} is Global best solution that is best position out of all particles in the swarm .W is Inertial Weight which regulates tradeoff between the global exploration and local exploration capabilities of the Swarm.C_1 and C_2 are acceleration constants which pull each particle towards P_{best} and G_{best} positions.r_1 and r_2 are two random functions with range [0,1].

Next a fitness function is defined for the considered optimization objective as below

$$Fitness = \frac{1}{Max(C_k)}$$

Based on fitness threshold, fitness values for the individual particles are evaluated and compared with Swarms best fitness values. If the fitness value is better than G_{best} then replace G_{best} as current particles fitness value. For new generation change the velocity and position of the particle as per equations (1) & (2) respectively. Again find the fitness function value for the new generation and repeat aforementioned process .When the Algorithm converges terminate the Algorithm. The output will be the optimal schedule scheme with the highest fitness value[21]. Fig.7 represents Flow chart for PSO Algorithm.

C. Encoding Scheme and Generation of Initial Swarm

The Main issue in implementing PSO for scheduling problems is how to encode a schedule to find a suitable configuration among problem solution and PSO particle. To overcome from this issue we have set up a search space of $n \times m$ dimensions for n task on m machines. A particle consists of m segments and each m segments has n task numbers denoted as (n,m) that is processing orders of n tasks on m machines[22].

For Example, consider six tasks and six machines [FTO6] Fig.5 shows one of the possible order on machine-1 to a particles position in PSO domain.

Fig.5 Task assignment V/S PSO particle

Operation on machine-1:

(Job, Operation) = {(1,b),(2,c),(3,d), (4,b),(5,c),(6,d)} $\xrightarrow{\text{Mapping}}$

Dimension: 1 2 3 4 5 6
Position: 1 3 4 5 2 6
Operation: b d b c c d

In order to reduce the number of iterative, a new method is introduced to generate initial Swarms. In which most of feasible tasks are arranged in increasing order of their operations and few operations are reversed, then the task order are arranged according to increasing order of their operations on the machines. If the order of one task operation is same as the other, the orders of two tasks are randomly arranged.

For Example, Fig.6 shows two possibilities to generate an initial particle in new way. If task on all the machines are arranged in this way then the possibility of obtaining feasible schedule, highly increases.

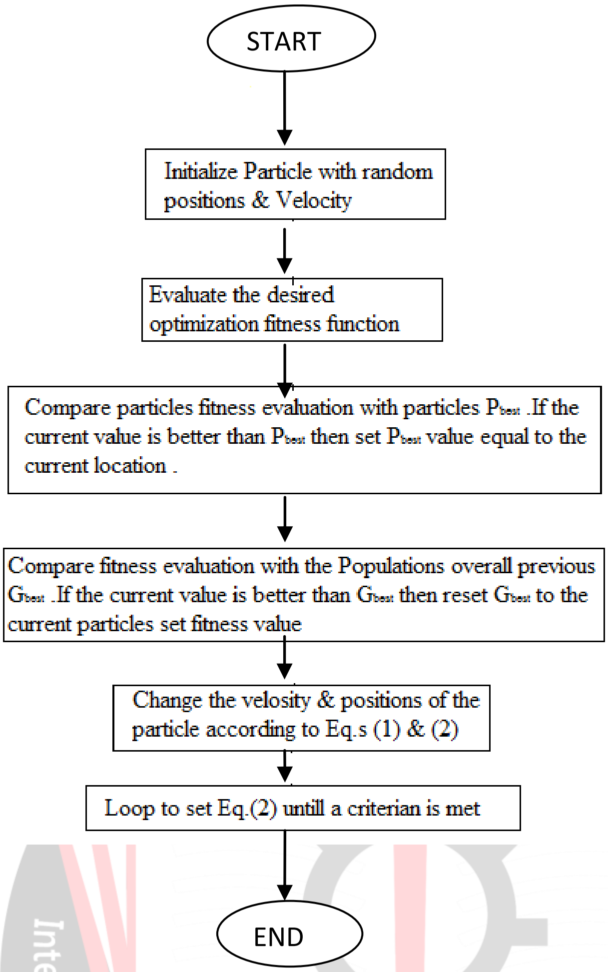
Fig.6 Two possibilities

Initial particle-1
Dimension: 1 2 3 4 5 6
Position: 6 1 4 3 2 5
Operation: d b b d c c

Initial particle-2
Dimension: 1 2 3 4 5 6
Position: 6 4 1 3 5 2
Operation: d b b d c c

DOI : 10.18231/2454-9150.2018.0189

Fig. 7, Flow chart: PSO Algorithm



V. SETTING OF SIMULATION PARAMETERS

An Important parameter for PSO algorithm is Inertial Weight (W) .The appropriate selection of inertial weight give rise to a balance between global search and local search and reduces number of iterations to find an optimal solution .In this paper, for all computations inertial weight is considered as equation given below

$$W = W_{max} - \frac{W_{max} - W_{min}}{I_{max}} \times I$$

Where, W_{max} – initial weighting coefficient value, W_{min} – final weighting coefficient value, I_{max} - maximum number of iterations, I - current iteration

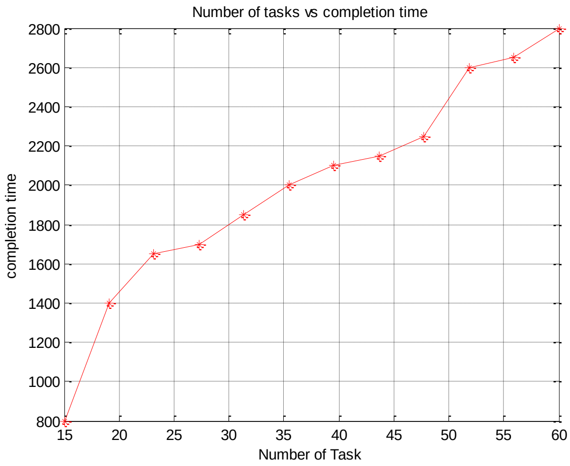
we set simulation experiments on scheduling scheme and optimal solutions are obtained in MATLAB. The fixed parameters for simulation is a given in the table III.

Table III Parameter Setting for Simulation:

Parameter	Values
N_o	5
N_m	30
$M_{i,j}$	10
N_{st}	15
N_p	15
N_T	8
Number of subtask in a process	[5 9]
Number of functions for each machine type	[1 5]

The other parameters are defined as follows .in the construction of process tree it is assumed that five subtask types occurs in each layer with equal probability of 70% .W_j is within the range of [1 ton .However W_j is greater for the subtask which are near to the root of the process tree. Further we obtain 15 different machine type allocation scenarios by assigning types to all 30 machines randomly, by assigning N_T numbers from 15,20,25………30 we get 10 task scenarios for different workloads. To get the optimal solution of PSO algorithm we conduct simulation experiments on these scenarios so as to overcome from possible deficiency fitness function and some other parameters in population initialization. PSO algorithm converges when there is no difference less than 0.5% of fitness values of a new generation. The criteria of convergence for number of generation for PSO algorithm varies from [50 110] to obtain the scheduling strategies we conduct simulation experiments with appropriate schemes and obtain performance indexes of overall completion time .At last, we plot these results as shown in **Fig 8.**

Fig. 8 Number tasks v/s Completion time:



D. Analysis on Scheduling Strategy

Here we considered starting inertial weight value 1.2 and decreases linearly to 0.4 as per equation (2) over the run. Acceleration constants C_1 and C_2 are set each equal to 2.0 with respect to the previous references [22]. In equation (1), V_{id} and X_{id} are user specified parameters and are limited to a maximum velocity V_{max} and Maximum position X_{max} . In this paper V_{max} and X_{max} are set to n number of tasks that is V_{id} value range [-n,n] and X_{id} positive integer value range [1,n]. From **Fig.8**, We notice that the scheduling strategy can obtain the optimal performance for the corresponding optimization objective. When the workload is more, there is congestion of subtasks in machines, so that production time overwhelms the transportation time, and the overall processing time is highly composed of production time.

VI. CONCLUSION

This research paper focuses on customer order subtask scheduling to minimize total weighted completion time .A PSO Algorithm is studied to obtain the optimal schedule for the aforementioned problem. Simulation results shown that, PSO algorithm can be used to obtain the optimal solution for optimization objective of overall completion time. Also it is noticed that the high workload results congestion of subtask in machines, so the production time occupies greater portion in the processing time as the workload increases. Based on our research, future research direction includes, subtask scheduling for priority and arriving pattern of task, optimization performance index of varying production costs, delay and inventory costs.

ACKNOWLEDGMENT

The author is grateful to the management of New Horizon College of engineering and REVA University led by its chairman, Vice –Chairman, secretary, Director and Principal for their support and guidance.

REFERENCES

[1] A. Gerodimos, C. Glass, C. Potts, and T. Tautenhahn, “Scheduling, multi-operation jobs on a single machine,” Ann. Oper. Res., vol. 92, pp. 87–105, 1999.

[2] J. N. Gupta, J. C. Ho, and J. A. van der Veen, “Single machine hierarchical scheduling with customer orders and multiple job classes,” Ann. Oper. Res., vol. 70, pp. 127–143, 1997

[3] J. D. Blocher and D. Chhajed, “The customer order lead-time problem on parallel machines,” Naval Res. Logistics (NRL), vol. 43, no. 5, pp. 629–654, 1996.

[4] F. Julien and M. Magazine, “Scheduling customer orders: An alternative production scheduling approach,” J. Manuf. Oper. Manage, vol. 3, no. 3, pp. 177–199, 1990.

[5] E. Wagneur, C. Sriskandarajah, “Open shops with jobs overlap,” Eur. J. Oper,. Res.71, pp.366–378, 1993.

[6] L. Ching-Jong, “Tradeoff between setup times and carrying costs for finished items,” Comput. Oper. Res., vol. 20, no. 7, pp. 697–705, 1993.

[7] C. S. Sung and S. H. Yoon, “Minimizing total weighted completion time at a pre-assembly stage composed of two feeding machines,” Int. J. Prod. Econ., vol. 54, no. 3, pp. 247–255, 1998.

[8] J. Y. Leung, H. Li, and M. Pinedo, “Scheduling orders for multiple product types to minimize total weighted completion time,” Discrete Appl. Math., vol. 155, no. 8, pp. 945–970, 2007.

[9] G. Wang and T. Cheng, “Customer order scheduling to minimize total weighted completion time,” Omega, vol. 35, no. 5, pp. 623–626, 2007.

[10] R. Ahmadi, U. Bagchi, and T. A. Roemer, “Coordinated scheduling of customer orders for quick response,” Naval Res. Logistics (NRL), vol. 52, no. 6, pp. 493–512, 2005.

[11] S. J. Mason and J.-S. Chen, “Scheduling multiple orders per job in a single machine to minimize total completion time,” Eur. J. Oper. Res., vol. 207, no. 1, pp. 70–77, 2010.

[12] X. Xu, Y. Ma, Z. Zhou, and Y. Zhao, “Customer order scheduling on unrelated parallel machines to minimize total completion time,” IEEE T. Autom. Sci. Eng., in press, DOI: 10.1109/TASE.2013.2291899

[13] J. Sun, Q. C. Zhao, and P. B. Luh, “An enhanced nested partitions method,” in Proc. IEEE Conf. Autom. Sci. Eng., Washington, DC, USA, pp. 377–382, 2008.

[14] Zhongshun Shi,Longfei Wang,Pai Liu, and Leyuan Shi, “Minimizing completion time for order scheduling :Formulation and Heuristic Algorithm, ”IEEE , Vol.14,no 4,pp.1558-1568,2017.

[15] L. Ching-Jong and C. Cheng-Hsing, “Sequencing with setup time and order tardiness trade-offs,” Naval Res. Logistics (NRL), vol. 43, no. 7, pp. 971–984, 1996.

[16] J. Y. Leung, H. Li, and M. Pinedo, “Scheduling orders for multiple product types with due date related objectives,” Eur. J. Oper. Res., vol. 168, no. 2, pp. 370–389, 2006.

[17] J. Kennedy, R. Eberhart, “Particle swarm optimization,” IEEE International Conference on Neural Networks, vol. 4, pp 1942–1948, 1995.

[18] S. Pandey, L. Wu, S.M. Guru, R. Buyy, “A particle swarm optimization-basedheuristic for scheduling workflow applications in cloud computing environments,” Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications, pp 400–407, 2010.

[19] Y.J Laili, L. Zhang, and F. Tao, “Energy adaptive immune genetic algorithm for collaborative design task scheduling in cloud manufacturing system” proceedings IEEE International conference Ind. Eng. Eng. Manage., pp 1912-1916, 2011.

[20] S.S. Zabihzadeh, J. Rezaeian, “Two meta-heuristic algorithms for flexible flowshop scheduling problem with robotic transportation and release times,” Appl.Soft Comput. Vol 40, pp 319–330, 2016.

[21] Q. Ma, X. Lei, Q. Zhang, “Mobile robot path planning with complex constraintsbased on the second-order oscillating particle swarm optimization algorithm”, Word Congr. Comput. Sci. Inf. Eng., pp 244–248, 2009.

[22] Z.-L. Sun, X.-Y. Li, Y. Wang, “Improve simple particle swarm optimization algorithm,” Comput. Sci. 42 pp 86–88 ,2015 .

DOI : 10.18231/2454-9150.2018.0189