International Journal of Engineering

Journal Homepage: www.ije.ir

Lot Streaming in No-wait Multi Product Flowshop Considering Sequence Dependent Setup Times and Position Based Learning Factors

P. Fattahi*[a], V. Azizi[b], M.Jabbari[a]

[a]*Department of Industrial Engineering, Bu-Ali Sina University*
[b]*Department of Industrial Engineering, K.N.Toosi University*

P A P E R I N F O

Paper history:
Received 15 March 2015
Received in revised form 10 May 2015
Accepted 11 June 2015

Keywords:
No-wait Flowshop
Lot Streaming
Sequence dependent Setup Times
Variable Neighborhood Search
Tabu Search, Simulated Annealing

A B S T R A C T

In this paper, a flowshop scheduling problem is studied. The importance of this study is that it considers different constraints simultaneously. These constraints are Lot Streaming, Position based learning factors, sequence dependent setup times and the fact that the flowshop line is no-wait. Lot streaming divide the lots of products into portions called sublots in order to reduce the lead times and work-in-process, and increase the machine utilization rates. The objective is to minimize the makespan. To clarify the system, mathematical model of the problemis presented. Since the problemis strongly NP-hard, two hybrid metaheuristics algorithms are proposed to solve the problem. These algorithms are based on the Variable Neighborhood Search (VNS), which is proved as an effective method for combinatorial optimization problems. In the proposed VNS, an efficient scheme for neighborhood search based on Tabu Search (TS) and Simulated Annealing (SA) is presented to strengthen the local searches. At the last part, computation results are provided to evaluate the efficiency of VNSSA and VNSTS. In order to verify the effectiveness of proposed algorithms, Relative percentage Deviation along with statistical analysis is presented.The computational results show that VNSSA outperforms VNSTS in most instances.

***doi**: 10.5829/idosi.ije.2015.28.07a.09*

1. INTRODUCTION

Scheduling of production systems with multiple products is affected by a number of factors such as learning parameters, sequence dependent setup times and lot streaming. Lot streaming has served an important role in scheduling problems. This technique reduces the required setup times as well as the amount of storage space. In this paper, a real world problem is modeled. In this problem, a no-wait, m-machine flowshop is studied by considering learning effects and sequence dependent setup times in which products must be produced in batches and lot streaming is basically considered in scheduling.

In recent years, there has been a focus on lot streaming in scheduling problems. These studies can be categorized using nine different criteria as mentioned by Surin and Jaiprakash [1].

In this study, the mathematical model of the problem is presented. Since the problem constraints are so complex, this model is useful to solve small-sized problems. For real-sized problems, two hybrid metaheuristics are proposed. These metaheuristics are based on variable neighborhood search (VNS) and take advantage of three neighborhood structures. VNS algorithm is an efficient metaheuristic providing optimal or near optimal solutions for most realistic instances in moderate CPU time. Since VNS algorithm has few parameters, it is known as one of the most practical metaheuristic algorithms. In this study, the exploration structure of VNS is upgraded using Simulated Annealing (SA) and Tabu Search (TS). These metaheuristics enhance the performance of VNS by better local searches.

*Corresponding Author's Email: pfattahi@gmail.com (P. Fattahi)

Please cite this article as: P. Fattahi, V. Azizi, M.Jabbari, Lot Streaming in No-wait Multi Product Flowshop Considering Sequence Dependent Setup Times and Position Based Learning Factors, International Journal of Engineering (IJE), TRANSACTIONS A: Basics Vol. 28, No. 7, (July 2015) 1031-1039

2. LITERATURE REVIEW

No-wait flow shop has several applications in modern manufacturing and servicing environments. According to Rock [2], no-wait flowshop scheduling problem with more than two machines is strongly NP-hard. Gilmore and Gomory [3] provided polynomial time algorithms for two-machine no-wait flow shop scheduling problem. There was no important progress in no-wait flowshop scheduling until recent two decades. Kumar et al. [4] studied a m-machine no-wait flowshop scheduling problem with multi products. They proposed Genetic Algorithm to solve the problem. Aldowasian and Alahverdi [5] studied M-machine no-wait flowshop and proposed a heuristic method and compared it with previous results obtained by other researches. They proved that new method performs better on average. Rabeiee et al. [6] investigated the scheduling of a no-wait two-machine flowshop considering anticipatory sequence-dependent set-up times. Laha and Supkal [7] studied no-wait m-machine flowshop scheduling and presented a constructive heuristic to solve the problem.

There are many researches focusing lot streaming in production system. Among them, Szendrovits [8] research was one of the first researches modeling a single product multi-stage production system. Pots and Baker [9] studied flowshop scheduling with lot streaming in order to minimize the length of flow shop schedule. Yoon and Ventura [10] studied a lot streaming flow shop scheduling problem and proposed a hybrid Genetic algorithm to solve it. Tseng and Liao [11] considered a n-job m-machine lot streaming problem in a flow shop scheduling and proposed a net benefit of movement (NBM) algorithm and a discrete particle swarm optimization (DPSO) algorithm to minimize the total weighted earliness and tardiness. Behnamian et al. [13] have studied the problem of sequence dependent set-up time hybrid flowshop scheduling with the objectives of minimizing the makespan and sum of the earliness and tardiness of jobs. They proposed a hybrid VNS-SA algorithm and compared the results with the previous results obtained by Genetic Algorithm. The comparison showed that the hybrid solution is effective. Hansen et al. [14] investigated about variable neighborhood search in different situations such as combinatorial problems and compared its efficiency, effectiveness and robustness with other algorithms.

Pan et al. [15] proposed a discrete harmony search algorithm to solve lot streaming flow shop. They compare their algorithm with previous results and demonstrate the effectiveness of the proposed DHS. Pan and Ruiz [15] consider a n-job m- machine lot streaming flow shop scheduling with sequence dependent setup times under both the idling and no-idling production cases with the objective of minimizing maximum completion time. Ventura and Yoon [16] investigated a n-job m-machine flowshop scheduling problem and consider equal sublots and limited capacity buffers with blocking and proposed a new Genetic Algorithm to solve the problem. They proved that NGA outperforms GA on the average. Ramezanian et al. [17] discussed a multi-product multi period integrated lot-sizing and scheduling problem considering overlapping in operations. They provided the mixed integer programing model and two heuristic to solve the problem. Vijaychakaravarthy et al. [18] studied a n-job m-machine lot-streaming problem in flowshop with variable size sublots and also to determine the optimal sublot size. They provided an improved sheep flock heredity algorithm (ISFHA) and Artificial Bee Colony (ABC) algorithm to minimize the makespan and total flow time.

Biskup [19] and Cheng and Wang [20] are pioneers in applying learning effects in scheduling problems. Lee and Wu [21] considered learning effect in two-machine flowshop scheduling problem with the criterion of minimizing completion time. Wang and Xia [22] considered the assumption of increasing dominance machine into the flowshop scheduling problem with learning effects. Chen et al. [23] studied two-machine flowshop scheduling problem including learning effect with respect to two criteria: minimizing total completion time and the weighted sum of maximum tardiness. Wang and Liu [24] developed two-machine flowshop scheduling problem with deterioration and learning effects. Isler et al. [25] proposed two-machine flowshop scheduling problem to minimize total earliness and tardiness penalties with learning effects assumption. Li et al [26] studied two-machine flowshop scheduling with truncated learning effects to minimize the total completion time. Cheng et al. [27] addressed two-machine flowshop scheduling with a truncated learning function to minimize the makespan. They described a branch and bound algorithm to obtain the optimal solutions in small size instances and a genetic algorithm to obtain near optimal solutions in large-scale instances.

3. PROBLEM DESCRIPTION

The objective of this paper is to minimize the makespan in a novel model that considers simultaneously no-wait, m-machine flowshop with multiple products considering sequence dependent setup times and learning effect. Besides these, in this problem, each product is divided into sublots and lot streaming should be considered in flowshop schedule. In this problem, a set of N products, $N = \{1,2,\dots,n\}$ are produced on a set of m machines, M = {1,2,…,m} in a way that waiting time between processing of consecutive jobs is not allowed and each product is divided into C sublots.

In order to produce each sublot of products on machines, the setup operation has to be done before starting the process. It should be considered that sequence of products affects the setup times. Therefore, the required setup time of sublot $r(r \in L)$of product j on machine m depends on its previous product (i). Also, if sublot r is in position τ on machine m, its required process time (p_{im}^{τ}) is calculated based on the truncation learning formula,$p_{im}^{\tau} = p_{im}^{\tau}{}' * \max\{\tau^{\alpha_m}, \beta_m\}$, where $p_{im}^{\tau}{}'$is the actual process times, α_m is learning effect parameter for machine m and β_m is the control parameter limiting the learning parameter and do not let the processing time to drop to zero.

In this paper, the following notations are used to formulate the problem:

N: set of products N= {1, 2,…, n};

i,j,l: Product index, $i, j, l \in$ N;

C: Maximum number of sublots

L: set of sublots for each product, L= {1, 2,…, C};

r,p:sublot index, $r, p \in$ L;

T: set of positions T={1,2,…,n*C};

τ: position index $\tau \in$ T;

M: set of machines M={1, 2,…, m};

m:Machine index, $m\in$ M;

$sdst_{ijrm}$: sequence dependent setup time for sublot r of product j on machine m when j is placed after product i;$i \neq j$

$sdst_{jjrm}$: sequence dependent setup time for sublot r of product j on machine m when j is at the first of the sequence;

p_{im}: process time of product i on machine m;

α_m: Learning effect of mth machine

β_m: Truncation factor for learning effect of mth machine.

$LR(\tau, m)$: Learning rate for the sublot in position τ on machine m. $LR(\tau, m) = \max\{\tau^{\alpha_m}, \beta_m\}, \forall\, m \in M$

Variables of the mathematical model are as follows:

${x_{ijr}}^{\tau}$: 1, if sublot r of product j is processed at position τ when product j is processed after product i; 0 otherwise;

${x_{jjr}}^{\tau}$: 1 , if sublot r of product j is processed at position τ when product j is at the first sequence; 0 otherwise

$f_{\tau m}$: Processing completion time of part in position τ on machine m;

$S_{\tau m}$: Start time of doing operation on sublot in position τon machine m;

C_{max} : Makespan

The mathematical model of problem is presented as follows:

$$\text{Min } Z = C_{max} \quad (1)$$

Subject to:

$$\sum_{j=1}^{n} {x_{jjr}}^{\tau} = 1 \qquad r = 1, \tau = 1 \quad (2)$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{r=1}^{C} {x_{ijr}}^{\tau} = 1 \qquad \tau \geq C + 1 \quad (3)$$

$$\sum_{j=1}^{n} {x_{ijr}}^{\tau+1} = \sum_{l=1}^{n} {x_{lip}}^{\tau} \qquad \forall i \in N, \forall \tau \in T,\ r=1, p=c \quad (4)$$

$${x_{ijr}}^{\tau} = {x_{ij(r-1)}}^{\tau-1} \qquad \tau \geq 2, r \geq 2,\ \forall i \in N, \forall j \in N \quad (5)$$

$$\sum_{j=1}^{n}\sum_{r=1}^{C}\sum_{\tau=1}^{n*C} {x_{ijr}}^{\tau} \leq C \qquad \forall i \in N, i \neq j \quad (6)$$

$$\sum_{i=1}^{n}\sum_{r=1}^{C}\sum_{\tau=1}^{n*C} {x_{ijr}}^{\tau} \leq C \qquad \forall j \in N \quad (7)$$

$$s_{\tau m} \geq \sum_{j=1}^{N} {x_{jjr}}^{\tau} * sdst_{jjrm} \qquad \tau = 1, \forall m \in M, r = 1 \quad (8)$$

$$f_{\tau m} = s_{\tau m} + \sum_{j=1}^{n} {x_{jjr}}^{\tau} * LR(\tau, m) * p_{jm}/c \qquad \tau = 1, \forall m \in M, r = 1 \quad (9)$$

$$s_{\tau m} \geq f_{\tau(m-1)} \qquad \tau \geq 2, m \geq 2 \quad (10)$$

$$s_{\tau m} = f_{(\tau-1)m} + \sum_{j=1}^{n}\sum_{j=1}^{n}\sum_{r=1}^{C} {x_{ijr}}^{\tau} * sdst_{ijrm} \qquad \tau \geq 2, \forall m \in M \quad (11)$$

$$s_{\tau m} \geq f_{(\tau-1)(m+1)} + \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{r=1}^{C} {x_{ijr}}^{\tau} * sdst_{ijr(m+1)} - \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{r=1}^{C} {x_{ijr}}^{\tau} * LR(\tau, m) * p_{jm}/c \qquad \tau \geq 2, \forall m \in M \quad (12)$$

$$s_{\tau m} \geq s_{\tau(m+1)} - \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{r=1}^{C} {x_{ijr}}^{\tau} * LR(\tau, m) * p_{jm}/c \qquad \tau \geq 2, \forall m \in M \quad (13)$$

$$f_{\tau m} = S_{\tau m} + \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{r=1}^{C} {x_{ijr}}^{\tau} * LR(\tau, m) * p_{jm}/c \qquad \tau \geq 2, \forall m \in M \quad (14)$$

$${x_{ijr}}^{\tau} \in \{0,1\} \qquad \forall i, j \in N, \forall \tau \in T,\ \forall r \in L \quad (15)$$

$$s_{\tau m} \geq 0 \qquad \forall m \in M, \forall \tau \in T \quad (16)$$

$$f_{\tau m} \geq 0 \qquad \forall m \in M, \forall \tau \in T \quad (17)$$

The objective function (1) indicates minimization of makespan. Equation set (2) assigns a sublot of a product to the first sequence of schedule. Constraint (3) ensures that each position from "C+1"to "C*n" can be occupied by just one sublot. Constraint set (4) indicates that if the position τ is assigned to the last sublot of product i, the first sublot of a next product should be assigned to the next position($\tau + 1$). Constraint set (5) ensures that a product sublot set should process continuously on a set of sequences. Constraints (6) and (7) require each sublot to be assigned to a sequence. Constraint (8) calculates the start time of processing first sublot of first product on first machine. Constraint set (9) calculates the time when the process on first sublot of first product is finished. Constraint (10) and (11) calculate the start time of processing sublots on other machines. Constraints (12), (13) limits the start time of processing on machines considering no-wait conditions. Constraint set (14) calculates finish time of processing sublots in sequence 2 to n*C on all machines. Constraint (15) ensures that ${x_{ijr}}^{\tau}$are binary variables and constraints (16), (17) specify the nonnegative variables.

4. PROPOSED SOLVING APPROACHES

As it is mentioned by Rock [2], a simple form of no-wait flowshop scheduling is NP-Hard. Therefore, this problem that is a developed form of Rock's problem is strongly NP-hard too. In this case, exact methods are not practical in solving real-world problems. In this part, three practical metaheuristics (VNS,TS, SA) are described thoroughly and from these approaches, two hybrid solution procedures are introduced.

Solution representation for both of these proposed methods is an integer string of *n*, where *n* is the number of products and initial solutions are generated randomly.

4. 1. Variable Neighborhood Search VNS is one of the widest applicable metaheuristics known until now. It is based on local search methods. VNS starts from an initial solution and proceed with local changes in order to improve the value of objective function. In this algorithm, N_k neighborhood structures are defined. The local searches are based on these neighborhood structures. Searching procedure is continued until the stopping condition is met. Steps of the basic VNS are as follows.

Select the set of neighborhood structures N_k , k=1,…,k_{max}that will be used in the search; find an initial solution x, choose a stoping condition;

Repeat the following until the stopping condition is met :

(1) Set $k \leftarrow 1$;

(2) Until $k = k_{max}$ repeat the following steps:

(a) *Shaking*. Generate a point $\acute{x}$ at random from the k^{th} neighborhood of x

(b) *Local search*. Apply some local search method with $\acute{x}$ as initial solution; denote with $x^{"}$ the so obtained local optimum;

(c) *Move or not*. If this local optimum is better than the incumbent, move there ($x \leftarrow x$") and continue the search with N_1 ($k \leftarrow 1$); otherwise , set $k \leftarrow k + 1$;

4. 1. 1. Neighborhood Structure Neighborhood structure is a mechanism leading to a new set of neighborhood solutions. This structure should eliminate unnecessary moves. In this paper, three neighborhood structures are used. The first structure randomly selects two elements in current solution and exchanges them (swap). Second structure chooses an element and put it just before another random element (insert). Third one reverses the elements which are located between two randomly chosen elements (reversion).

4. 2. Simulated Annealing Simulated Annealing (SA) was first introduced by Kirkpatrick et al. [28]. It is inspired from process of melting and refreezing materials. SA can escape from being trapped into local optimum solutions by searching for fair solutions, in small probability.

SA procedure starts with random solutions. In each iteration, the moves decreasing the energy will always be accepted while fair moves will only be accepted with a small probability. Therefore, SA will also accept bad solutions with small probability, determined by Boltzmann function, $\exp(-\frac{\Delta}{KT})$where K and T are predetermined constant and the current temperature, respectively. Also Δ is the difference of objective values between the current solution and the new solution.

If the calculated Boltzmann function value is more than a uniform random number between 0 and 1, then the bad solution should be accepted.

4. 2. 1. Cooling Schedule Cooling schedule has a great influence on the success of the SA optimization algorithm. The parameters of cooling schedule consist of initial temperature, equilibrium state and a cooling function. In this article, the initial temperature is defined as the maximum difference between the fitness function value of solution seeds in the initial population.

There are various methods of decreasing temperature in each iteration, such as arithmetical, linear, logarithmic, geometric, non-monotonic, and very slow decrease. In this paper, the logarithmic method is used as Equation (19). In this equation $t_i$is the temperature in the iteration *i* of algorithm.

$$t_i = \frac{t_{i-1}}{\log(i)} \quad (19)$$

4. 3. Tabu Search Tabu search is a metaheuristic method capable of obtaining optimal or near optimal solutions in to a wide variety of problems. This algorithm was first introduced by Glover [29] Using a flexible memory structure, Tabu Search algorithm provides conditions to search the solution space strategically and prevent the procedure from becoming trapped at locally optimal solutions. This memory function can vary over the time span in order to intensify or diversify the search procedure. In this algorithm, there is a forbidden list of potential solutions called Tabu list. The Tabu list will be updated in each iteration of algorithm. One of the most important parameters of Tabu Search is the strategy that set the rules of making Tabu list. Tabu tenure determines how long the corresponding solutions need to be tabooed.

4. 3. 1. Neighborhood Solution Set Size In this paper, Tabu Search is designed in a way that considers all possible states of cods for each problem.

4. 3. 2. Tabu Tenur Tabu tenure determines how long the corresponding solutions need to be tabooed. Furthermore, the size of tabu list is of great importance. In this paper, the tabu tenure strategy is similar to strategy explained by Li et al. [30]. The pseudo-cod tenure strategy is shown in Figure 1.

$begin$

$k = iteration/5$

$if\ t \in (0, k)$

$Tenure = Tenure_{min}$

$Tlength = Tlenght_{min}$

$if\ t \in (k, 3k)$

$g = \frac{Tenure_{max} - Tenure_{min}}{2k}$

$Tenure = Tenure_{min} + g(t-k)$

$Tlength = Tlenght_{min} + g(t-k)$

$if\ t \in (3k, 5k)$

$Tenure = Tenure_{max}$

$Tlength = Tlenght_{max}$

end

Where

$Tenure_{min} = number\ of products/2$

$Tenure_{max} = number\ of\ products$

$Tlength_{min} = number\ of products/2$

$Tlenght_{max} = number\ of\ products$

Figure 1. Pseudo-cod of Tenure Strategy

4. 4. Hybrid Solution

4. 4. 1. VNSSA In order to improve the performance of metaheuristics, Simulated Annealing and Variable Neighborhood Search are joined together. SA has the ability to explore the solution space and improve VNS performance by intensifying local searches. To clarify the hybrid method, consider three steps of VNS described in 2.1, the neighborhood structures are selected at first step.

In the second step, a new solution is generated as described in *shaking* part. For the next step, *local search*, instead of using simple local search, SA algorithm operators do search the solution space. At the third phase, the obtained solution from SA algorithm is compared with the initial solution. If the solution is better, it will be replaced. At the last part the stop condition is checked and these steps will continue until the stop condition is met.

4. 4. 2. VNSTS The hybrid VNS-TS algorithm has many similarities to hybrid VNS-SA algorithm. In the other words, all steps are the same except *local search*. The structure of hybrid VNS-TS algorithm is based on Variable Neighborhood Search. However, hybrid algorithm enhance local search phase by using Tabu Search operators. So, the algorithm starts as VNS and for the local search use TS algorithm instead of simple local search.

5. COMPUTATIONAL RESULTS

Choosing the best parameters for metaheuristics is of great importance. Therefore, SA and TS parameters have been tuned to optimize their performance. One of the best known ways of tuning parameters is Taguchi method. In this paper, Minitab software is used to design the experiments and assign the best level for each size of problems. The results of Taguchi method are shown in Table 1. In order to compare the performance of proposed algorithms, 20 problems are introduced which have different machine, product and sublot sizes. The instances are created randomly and categorized into three sub groups: Small problems, Medium problems and large problems. Table 2 shows the properties of each instance.

In these problems, sequence dependent setup are drawn from a uniform distribution over the interval [1,50] and process times are uniformly distributed over the interval [1,50]. The learning parameter of machines, α, is considered "-0.3" and truncation factor of machines, β, is "0.7" for all machines. To solve the problems, proposed metaheuristics are coded by MATLAB software. Furthermore, the mixed integer linear programming is coded with GAMS IDE (ver. 23.5) software. The instances are solved on a PC with Intel Corei5, 2 GHz and 4G memory.

The obtained results are specified in Table 3. The effectiveness of algorithms is compared using a well-known criteria, Relative Percentage Deviation (RPD). RPD factor is computed as Equation (20) and compare the performance of each procedure to other procedures for each instance.

$$RPD = \frac{algorithm_{solution} - minimum_{solution}}{minimum_{solution}} \quad (20)$$

A statistical test is proposed in this paper to compare the ability of metaheuristics at finding the best solution. This statistical test can be based on ANOVA or nonparametric tests. Before choosing between ANOVA and nonparametric test, the hypothesis of normality should be checked for both VNSSA RPDs and VNSTS RPDs. The test is performed using Minitab software. Figures 2 and 3 show the results of normality test for VNSSA and VNSTS, respectively. These tests are based on the ANDERSON-DARLING method with 0.05 significance level. As it is obvious in the picture, the p-value of tests are less than the required significance level. Therefore, both VNSSA and VNSTS have normal distribution. In order to select proper ANOVA test, the variances of VNSTS and VNSSA are tested. This test identifies whether the variances are same or not.

TABLE 1. Parameter designs

Metaheuristic	Parameter	Appropriate quantity according to Taguchi method		
		Small	**Medium**	**Large**
SA	Population	5	7	15
	Maximum iteration	5	15	15
	Number of movements	10	20	20
	α	0.999	0.999	0.999
	K	1	1	1
TS	Maximum iteration	20	40	100

TABLE 2. Samples properties

	Problem	Number of machines	Number of sublots	Number of products
Small	P1	3	3	3
	P2	3	5	3
	P3	5	3	3
	P4	3	3	5
	P5	3	5	5
	P6	3	3	7
	P7	3	5	7
	P8	3	3	10
	P9	3	5	10
Medium	P10	5	3	12
	p11	5	5	12
	p12	5	7	12
	p13	3	3	15
	p14	3	5	15
	p15	3	7	15
Large	p16	5	3	18
	p17	5	7	18
	p18	3	3	20
	p19	3	5	20
	P20	3	7	20

The results of the test show that the p-value is less than the significance level, the variances are not equal. Considering these observations, the appropriate test of comparing the RPDs of two methods is Two-Sample T-test. In this test, the null hypothesis is the equality of mean of VNSTS RPD and VNSSA RPD. This hypothesis is rejected when the mean of VNSTS RPDs are greater than mean of VNSSA RPDs. The results of ANOVA is summarized in Table 4. The P-Value is less than 0.05 so the null hypothesis is rejected in this test, and the mean of VNSSA RPDs are less than the VNSTS RPDs. This fact illustrates the better performance of VNSSA compared with VSTS. To compare metaheuristics, another important factor is computational time. Figure 4 compares the performance of proposed metaheuristics considering computational time.

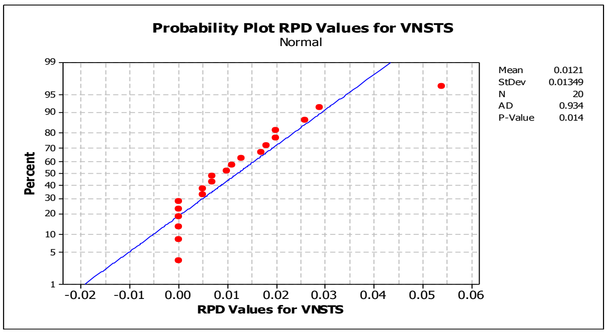

Figure 2. VNSTS normality test

TABLE 3. Computational results

Problem		Cplex		VNSSA			VNSTS		
		C_{max}	CPU time	mean C_{max}	RPD	CPU time	meanC_{max}	RPD	CPU time
Small	P1	174.828	0.080	174.828	0.000	4.948	174.828	0.000	0.635
	P2	162.579	0.050	162.579	0.000	5.988	162.579	0.000	0.745
	P3	166.962	0.19	166.962	0.000	1.855	166.962	0.000	0.641
	P4	277.307	0.050	277.307	0.000	9.114	277.307	0.000	3.388
	P5	268.024	0.060	268.024	0.000	12.559	268.024	0.000	4.852
	P6	366.100	0.090	366.1	0.000	10.765	370.463	0.011	8.139
	P7	369.347	0.120	369.346	0.000	16.506	369.346	0.000	11.477
	P8	496.5092	445.500	497.863	0.002	23.871	501.506	0.010	28.285
	P9	504.9055	603.102	511.744	0.013	30.536	507.743	0.005	41.671
Medium	P10			701.582	0.005	106.024	710.464	0.018	91.121
	p11			703.991	0.010	127.297	711.586	0.020	172.785
	p12			700.374	0.001	166.085	708.466	0.013	207.514
	p13			629.159	0.011	81.682	625.783	0.005	183.952
	p14			695.853	0.009	183.501	701.127	0.017	243.689
	p15			699.638	0.010	263.601	713.143	0.029	356.707
Large	p16			1011.390	0.003	1053.082	1015.174	0.007	1308.563
	p17			1051.163	0.0016	1702.593	1057.848	0.007	1833.369
	p18			909.576	0.004	979.0816	929.793	0.026	1077.534
	p19			929.914	0.011	763.196	938.457	0.020	1369.437
	P20			919.535	0.014	853.232	955.702	0.054	1069.875

TABLE 4. Two-sample T for VNSTS vs. VNSSA

Method	Number	Mean	Standard Deviation	Standard Error of Mean
VNSTS	20	0.0121	0.0135	0.0030
VNSSA	20	0.00473	0.00512	0.0011

Difference = mu (VNSTS) - mu (VNSSA)***Estimate for difference: 0.00737***95% lower bound for difference: 0.00185

T-Test of difference = 0 (vs >): T-Value = 2.28 P-Value = 0.016 DF = 24

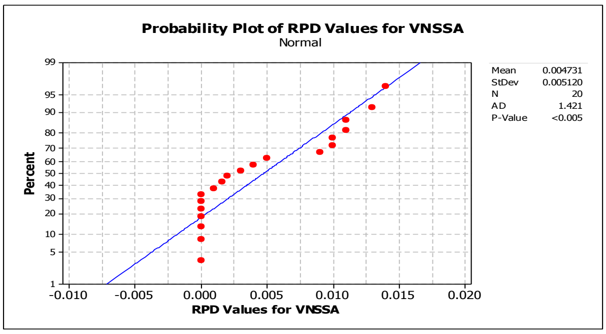

Figure 3. VNSSA normality test

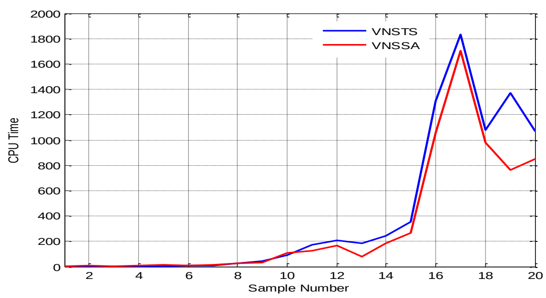

Figure4. Comparison of computational time for VNSSA and VNSTS

6. CONCLUSION

This paper dealt with the "no-wait flowshop scheduling problem" to minimize the makespan with respect to lot streaming, sequence-dependent setup times and truncated learning function. The problem is modeled using mixed integer linear programming. On account of the fact that this problem is strongly NP-hard, therefore, two hybridmetaheuristics are proposed to solve real-word problems.

These metaheuristics are based on the "Variable Neighborhood Search" and are empowered using "Simulated annealing" and "Tabu Search". Different sample problems are provided and solved by each procedure and results are analyzed to compare the performance of procedures in different problem conditions. Furthermore, a thorough investigation basedon ANOVA is presented to determine the best procedure. The results show that for small problems, both methods have approximately same performance. However, for large problems, VNSSA out performs VNSTS.

7. REFERENCES

1. Sarin, Subhash C., and Puneet Jaiprakash. *Flow shop lot streaming*. Springer, 2007.
2. Röck, Hans., "The three-machine no-wait flow shop is NP-complete.", ***Journal of the ACM (JACM)***, Vol. 31, No. 2, (1984), 336-345.
3. Gilmore, Paul C., and Ralph E. Gomory., "Sequencing a one state-variable machine: A solvable case of the traveling salesman problem.", ***Operations Research***, Vol. 12, No. 5, (1964), 655-679.
4. Kumar, Subodha, Tapan P. Bagchi, and C. Sriskandarajah., "Lot streaming and scheduling heuristics for< i> m</i>-machine no-wait flowshops.", ***Computers & Industrial Engineering***, Vol 38, No. 1, (2000), 149-172.
5. Aldowaisan, Tariq, and Ali Allahverdi., "New heuristics for< i> m</i>-machine no-wait flowshop to minimize total completion time.", ***Omega***, Vol. 32, No. 5 (2004), 345-352.
6. Rabiee, M., M. Zandieh, and A. Jafarian., "Scheduling of a no-wait two-machine flow shop with sequence-dependent setup times and probable rework using robust meta-heuristics.", ***International Journal of Production Research***, Vol. 50, No. 24, (2012), 7428-7446.
7. Laha, Dipak, and Sagar U. Sapkal., "An improved heuristic to minimize total flow time for scheduling in the< i> m</i>-machine no-wait flow shop.", ***Computers & Industrial*** Engineering, Vol. 67, (2014), 36-43.
8. Szendrovits, Andrew Z., "Manufacturing cycle time determination for a multi-stage economic production quantity model.", ***Management Science***, Vol. 22, No. 3 (1975), 298-308.
9. Potts, C. N., and K. R. Baker., "Flow shop scheduling with lot streaming.", ***Operations Research Letters***, Vol. 8, No. 6 (1989), 297-303.
10. Yoon, Suk-Hun, and Jose A. Ventura., "An application of genetic algorithms to lot-streaming flow shop scheduling." ***IIE Transactions***, Vol. 34, No. 9, (2002), 779-787.
11. Tseng, Chao-Tang, and Ching-Jong Liao., "A discrete particle swarm optimization for lot-streaming flowshop scheduling problem.", ***European Journal of Operational Research***, Vol. 191, No. 2, (2008), 360-373.
12. Pan, Quan-Ke, Mehmet Fatih Tasgetiren, Ponnuthurai Nagaratnam Suganthan, and Yun-Chia Liang., "Solving lot-streaming flow shop scheduling problems using a discrete harmony search algorithm.", **In** ***Evolutionary Computation (CEC)***, *2010 IEEE Congress on*, pp. 1-6. IEEE, 2010.
13. Behnamian, J., SMT Fatemi Ghomi, and M. Zandieh., "A multi-phase covering Pareto-optimal front method to multi-objective scheduling in a realistic hybrid flowshop using a hybrid metaheuristic." ***Expert Systems with Applications*** Vol.36, No. 8 (2009), 11057-11069.
14. Hansen, Pierre, and Nenad Mladenović. *Variable neighborhood search*. Springer US, (2005).
15. Pan, Quan-Ke, and Rubén Ruiz., "An estimation of distribution algorithm for lot-streaming flow shop problems with setup times.", ***Omega***, Vol. 40, No. 2, (2012), 166-180.
16. Ventura, José A., and Suk-Hun Yoon., "A new genetic algorithm for lot-streaming flow shop scheduling with limited capacity buffers.", ***Journal of Intelligent Manufacturing***, Vol. 24, No. 6, (2013), 1185-1196.
17. Ramezanian, Reza, Mohammad Saidi-Mehrabad, and Ebrahim Teimoury., "A mathematical model for integrating lot-sizing and scheduling problem in capacitated flow shop environments.", ***The International Journal of Advanced Manufacturing Technology***, Vol. 66, No. 1-4, (2013), 347-361.
18. Vijaychakaravarthy, G., S. Marimuthu, and A. Naveen Sait., "Comparison of Improved Sheep Flock Heredity Algorithm and Artificial Bee Colony Algorithm for lot Streaming in m-Machine Flow Shop Scheduling.", ***Arabian Journal for Science and Engineering***, Vol. 39, No. 5, (2014), 4285-4300.
19. Biskup, Dirk., "Single-machine scheduling with learning considerations.", ***European Journal of Operational Research***, Vol. 115, No. 1, (1999), 173-178.
20. Cheng, TC Edwin, and Guoqing Wang., "Single machine scheduling with learning effect considerations.", ***Annals of Operations Research***, Vol. 98, No. 1-4, (2000), 273-290.
21. Lee, Wen-Chiung, and Chin-Chia Wu., "Minimizing total completion time in a two-machine flowshop with a learning effect.", ***International Journal of Production Economics***, Vol. 88, No. 1, (2004), 85-93.
22. Wang, J. B., and Z. Q. Xia., "Flow-shop scheduling with a learning effect.", ***Journal of the Operational Research Society***, Vol. 56, No. 11, (2005), 1325-1330.
23. Chen, P., C. C. Wu, and W. C. Lee., "A bi-criteria two-machine flowshop scheduling problem with a learning effect.", ***Journal of the Operational Research Society***, Vol. 57, No. 9, (2006), 1113-1125.
24. Wang, Ji-Bo, and Li-Li Liu., "Two-machine flow shop problem with effects of deterioration and learning.", ***Computers & Industrial Engineering***, Vol. 57, No. 3, (2009), 1114-1121.
25. İşler, Mesut Cemil, Bilal Toklu, and Veli Çelik., "Scheduling in a two-machine flow-shop for earliness/tardiness under learning effect." ***The International Journal of Advanced Manufacturing Technology***, Vol. 61, No. 9-12, (2012), 1129-1137.
26. Li, Der-Chiang, Peng-Hsiang Hsu, Chin-Chia Wu, and TC Edwin Cheng., "Two-machine flowshop scheduling with truncated learning to minimize the total completion time.", ***Computers & Industrial Engineering***, Vol. 61, No. 3, (2011), 655-662.
27. Cheng, T. C. E., Chin-Chia Wu, Juei-Chao Chen, Wen-Hsiang Wu, and Shuenn-Ren Cheng., "Two-machine flowshop scheduling with a truncated learning function to minimize the makespan.", ***International Journal of Production Economics***, Vol. 141, No. 1, (2013), 79-86.
28. Kirkpatrick, Scott, C. Daniel Gelatt, and Mario P. Vecchi., "Optimization by simmulated annealing.", ***Science***, Vol. 220, No. 4598, (1983), 671-680.
29. Glover, Fred., "Tabu search-part I.", ***ORSA Journal on computing***, Vol. 1, No. 3, (1989), 190-206.
30. Li, Jun-qing, Quan-ke Pan, and Yun-Chia Liang., "An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems.", ***Computers & Industrial Engineering***, Vol. 59, No. 4, (2010), 647-662.

Lot Streaming in No-wait Multi Product Flowshop Considering Sequence Dependent Setup Times and Position Based Learning Factors

P. Fattahi[a], V. Azizi[b], M.Jabbari[a]

[a]*Department of Industrial Engineering, Bu-Ali Sina University*
[b]*Department of Industrial Engineering, K.N.Toosi University*

P A P E R I N F O

Paper history:
Received 15 March 2015
Received in revised form 10 May 2015
Accepted 11 June 2015

Keywords:
No-wait Flowshop
Lot Streaming
Sequence dependent Setup Times
Variable Neighborhood Search
Tabu Search, Simulated Annealing

چکيد

در این مقاله یک مساله زمانبندی جریان کارگاهی مطالعه شده است. اهمیت این مطالعه در نظر گرفتن همزمان چندین محدودیت به طور همزمان است. این محدودیت ها برش در انباشته ها، فاکتورهای یادگیری بر اساس موقعیت، زمان های تنظیم وابسته به توالی و این واقعیت که خط جریان کارگاهی بدون وقفه است. جریان انباشته به منظور کاهش زمان های تحویل و کارهای در حال انجام، انباشته های محصولات را به قسمت هایی که اصطلاحا زیر انباشته نامیده می شوند، تقسیم می کند و با این عمل نرخ بهره وری ماشین را نیز افزایش می دهد. هدف در این مساله کمینه کردن زمان اتمام کل کارها است. به منظور توضیح مفصل سیستم، مدل ریاضی مساله ارائه شده است. از آنجایی که مساله به شدت NP-hardاست، دو الگوریتم فراابتکاری ترکیبی برای حل مساله پیشنهاد شده است. این الگوریتم ها بر اساس جستجوی همسایگی متغیر (VNS)که یک روش کارا و موثر در مسائل بهینه سازی ترکیباتی است، ایجاد شده اند. به منظور تقویت ساختار جستجوی محلی در VNS، از جستجوی ممنوع (TS)و شبیه سازی تبرید تدریجی (SA) استفاده شده است. در قسمت پایانی مقاله، نتایج محاسباتی جهت ارزیابی VNSSAوVNSTSارائه شده اند. برای تایید موثر بودن الگوریتم ها، درصد انحراف نسبی همراه با تحلیل آماری ارائه شده است. نتایج محاسباتی نشان می دهد که VNSSAدر اکثر نمونه ها عملکرد بهتری از خود نشان می دهد.

.doi*: 10.5829/idosi.ije.2015.28.07a.09*