Advanced Logistic Systems, Vol. 8, No. 1 (2014), pp. 111-120.

EXAMINATION OF SCHEDULING METHODS FOR PRODUCTION SYSTEMS

ZOLTÁN VARGA[1]–PÁL SIMON[2]

Abstract: Nowadays manufacturing and service companies pay more attention to meet logistical demands. The widespread lean philosophy establishes claims to reduce production and logistic costs. The biggest cost reduction can be obtained by effective scheduling algorithms and logistics optimization. Several similarities and a close relationship can be seen between the two research areas. The aim of production scheduling can be defined as the allocation of available production resources in order to satisfy the criteria set by demands. These criteria contain a lot of logistical aspects, which also play important roles. Typically, the scheduling problem involves a set of tasks and an objective function, which aims to find a balance between early completion, stock and frequent production changeovers. Since the production processes can be diverse and unique, there are several different production models and scheduling algorithms. The aim of this article is to present and compare the nowadays applied different scheduling algorithms, with which the effectiency of production systems can be increased.
Keywords: scheduling, job shop, flow shop

1. Relationship between logistic and production scheduling

One of the special fields of logistics belongs to manufacturing systems, and it is called production logistics. It can be defined as the sum of the essential materials and production tools used in production processes, and the materials necessary for concordance of the sub processes of the production process and the related information flow processes. The task of the production logistics system is to cover a sufficient material supply during production. Accordingly, taking into consideration the requirements of production scheduling, it defines, for example, from which store it should be delivered, and what types of material handling equipment have to be used in a production process. Further tasks of the production logistics systems include supply reduction and reducing lead-times, makespan and expenses. These tasks appear in production scheduling, and they are of crucial importance in order to achieve optimal production.

2. Scheduling

Scheduling is the allocation of shared resources over time to competing activities. It has been the subject of a significant amount of literature in the operations research field. Emphasis has been on investigating machine scheduling problems where jobs represent activities and machines represent resources; each machine can process at most one job at a time.

The scheduling problem is one of the most important and hardest combinatorial optimization problems on account of its complexity and frequency in practical applications.

[1] PhD. student, University of Miskolc
zoli.varga86@gmail.com
[2] PhD student, University of Miskolc
simon11@iit.uni-miskolc.hu
H-3515 Miskolc-Egyetemváros, Hungary

The purpose of scheduling generally is to allocate a set of resources to tasks by the definition of Pinedo [1]. Since the first appearance of the systematic method to scheduling problems was in the mid-1950s, thousands of articles on different scheduling problems have arisen in the literature, which can be categorized in accordance with shop environments, including single machine, parallel machines, open shop, job shop, flow shop and others.

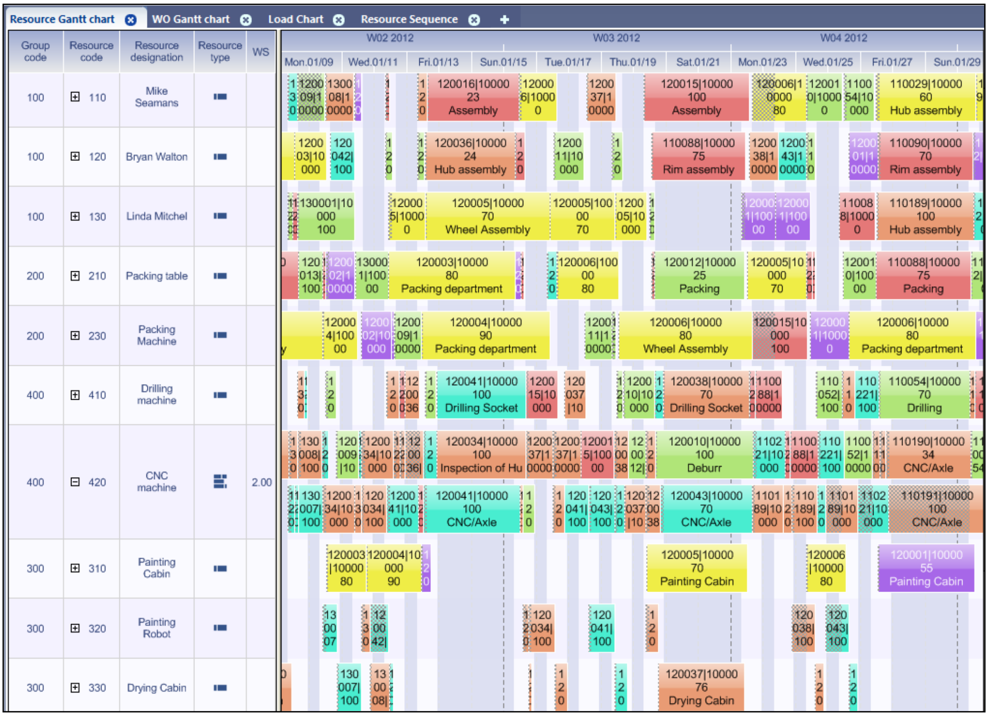

Figure 1. Example Gantt chart in Microsoft Dynamics NAV

2.1. Single-machine scheduling. The concepts of scheduling was a relevant research area in the late 70’s when the basic concepts was introduced [2]. There are given n jobs, which are denote by $J_1, \dots, J_n$ and there are given a set of machines, which can handle one job at a time and the jobs have to be scheduled on it. Depending on the machine configuration, we distinguish between single-machine scheduling problem, parallel-machine problem and shop model. Every job J_i $(i = 1, \dots, n)$ has a processing time p_i, that is the processing period length of a J_i job.

Given a π schedule, where the starting time of a J_i job in π is $S_i(\pi)$ and the completion time of π is $C_i(\pi)$, the argument π is omitted when it is clear to which schedule are referred. When interruption is not allowed then $C_i = S_i + p_i$. A J_i job execution depends on a release time r_i which is a lower bond on the starting time or a deadline di which is a upper bound on the starting time. Job J_i may have a weight w_i to express its importance. For a given schedule, $L_i = C_i - d_i$ is defined as the lateness of job J_i, and the tardiness T_i of J_i in a given schedule is defined as $T_i = max\{0, C_i - d\}$. Maximum lateness can be generalized to maximum cost f_{max} which is defined as $f_{max}(C) = max_i\{f_i(C_i \mid i = 1, \ldots, n)$, where each job J_i $(i=1, \ldots, n)$ has its own cost function $f_i(C_i)$. We use an indicator function U_i to denote when job J_i is tardy $(Ui = 1)$ or on time $(U_i = 0)$ in a given schedule. The opposite of the tardiness for a job J_i is earliness, which is defined as $E_i = max\{0, d_i - C_i\}$. The following performance criteria appear frequently in the literature:

- maximum completion time or makespan:

$$C_{max} = max\{ C_i \mid i = 1, \ldots, n \}, \tag{1}$$

- total (weighted) completion time:

$$\sum_{i=1}^{n} w_i C_i, \tag{2}$$

- maximum lateness:

$$L_{max} = max\{ L_i \mid \mid i = 1, \ldots, n \}, \tag{3}$$

- maximum tardiness:

$$T_{max} = max\{ T_i \mid i = 1, \ldots, n \}, \tag{4}$$

- maximum cost:

$$f_{max} = max\{ f_i(C_i) \mid i = 1, \ldots, n \}, \tag{5}$$

- total (weighted) tardiness:

$$\sum_{i=1}^{n} w_i T_i, \tag{6}$$

- maximum earliness:

$$E_{max} = max\{ E_i \mid i = 1, \ldots, n \}, \tag{7}$$

- total (weighted) earliness:

$$\sum_{i=1}^{n} w_i E_i, \tag{8}$$

- (weighted) number of tardy jobs:

$$\sum_{i=1}^{n} w_i U_i. \tag{9}$$

It is not necessary to use one criteria, but any combination of two criteria out of this list is possible. A natural one is to consider a combination of a tardiness and an earliness performance criteria, which reflects the just-in-time objective, as both an early delivery and a tardy delivery are penalized.

The single-machine scheduling have received considerable attention for many years since the 70s, when the fundamentals are presented [3]. During the last decades these principles have been supplemented by studies which have important aspects in practical production planning problems. In these studies the release times of the jobs are related to the amount of resources consumed, see for example Panwalkar and Rajagopalan [4] Li et al. [5], Biskup [6] and Kuo and Yang [7].

2.2. Parallel machine scheduling. In the parallel-machine scheduling problem more than one machine are available, for processing the jobs, they are identical and parallel, denote $M_1, \ldots, M_m$. There are given n jobs, denote $J_1, \ldots, J_n$ and each of these jobs have a processing time $p_1, \ldots, p_n$ to be processed on the m identical, parallel machines. The same criteria can be used like at the single-machine scheduling.

Nowadays the parallel-machine scheduling is not very often used. The most of the researches were dealt in the 90s [8][9]. In these articles the authors have been presented this is an NP-hard problem and some heuristic algorithms were presented [10][11]. In the last decade the research turned towards the unrelated parallel-machine scheduling problem [12][13][14].

2.3. Shop scheduling. A schedule is an assignment of operations to time intervals on the machines. A simple job routing example can be seen on *Figure 2.* The problem is to find a schedule of minimal time to complete all jobs. The shop scheduling including job shop, flow shop problems, which are widely used for modelling industrial production processes. All of these problems are special cases of the general shop problem.

The general shop problem can be defined as follows. There are given n jobs $J_1, \ldots, J_n$ and m machines $M_1, \ldots, M_m$. Each job i consists of a set of operations O_{ij} $(j=1, \ldots, n)$ with processing times p_{ij} . Each O_{ij} operation must be processed on a machine from the $\{M_1, \ldots, M_m\}$ set. Each job can be processed only by one machine at a time and each machine can only process one job at a time. The objective is to find a feasible schedule that correspond with some criteria, the regular criteria can be seen in Section 2.1.

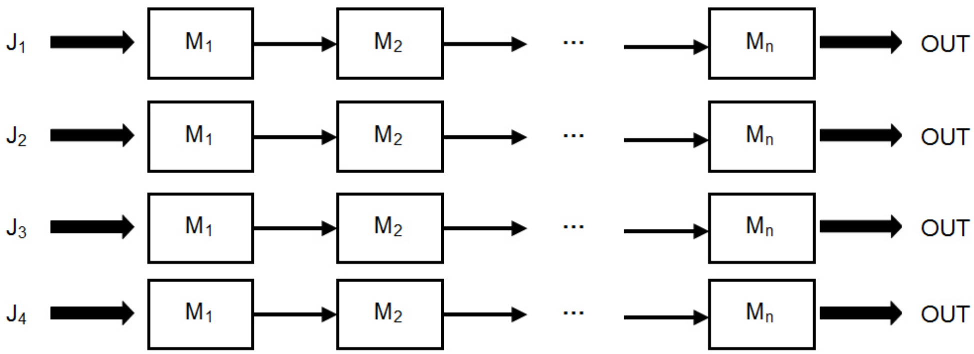

Figure 2. Job routing

3. Job shop scheduling problem

The Job Shop Scheduling problem is one of the most difficult ones among all the scheduling problems [15]. It is a nondeterministic polynomial time (NP) hard problem using combinatorial optimization. The literature uses the JSS, JSP and JSSP abbreviation as well. Hereafter we use the JSSP abbreviation. The Job Shop Scheduling problem can be defined as a set of jobs with several consecutive operations must be processed on a set of machines. Each operation should be processed by a particular machine, once started, the operation cannot be interrupted and each machine can only handle one operation type.

We also can define it as the following. The ordinary job shop scheduling model definition considers *n* jobs to be process on *m* machines (*n x m* operations) while minimizing some function of completion time of jobs subject to following technological constraints and assumptions:

- Each machine can perform only one operation at a time on any job.
- An operation of a job can be performed by only one machine at a time.
- Once an operation has begun on a machine, it must not be interrupted.
- An operation of a job cannot be performed until its preceding operations are completed.
- There are no alternate routings, i.e. an operation of a job can be performed by only one type of machine.
- Operation processing time and number of operable machines are known in advance.

The JSSP has attracted many optimization methods, because it still exists in most of manufacturing systems in various forms. There are various methods and solutions for JSSP, such as dispatching rules, mathematical formulas, branch and bound. Some of these formulas are exact, and there are many artificial intelligence techniques, like artificial neural networks, ant colony or bee colony algorithms, usually these are heuristic solutions. Fuzzy logic techniques are also good solutions for the problem. These methods are introduced to obtain an optimum, or mostly a near to optimum solution. The classical JSSP deals with only one performance criteria. This can be named single objective job shop scheduling problem.

Unfortunately nowadays the single objective job shop scheduling problem is not so close to reality, but the multi objective job shop scheduling is much closer. The goal of the multi objective job shop scheduling is to find many different promising schedules as possible, considering different criteria at the same time.

In literature we can find different categories of multi objective job shop scheduling, like dynamic job shop scheduling, or flexible job shop scheduling. In the following paragraphs we present these special job shop scheduling problems with examples and solutions from researchers.

3.1. Multi objective JSSP and Dynamic JSSP. One special multi objective job shop scheduling type is dealing with real time events such as random job arrivals and machine breakdowns are ignored. Taking into account these events, JSSP shifts to a new kind of problem that is well-known as dynamic job shop scheduling problem (DJSSP). In DJSSP, due to changes in problem condition during planning horizon, using a scheduling method with parameters set at their optimum value as used in most researches [16] can reduce performance of the selected method. Preventing this problem many researchers used their own algorithms mostly based on artificial intelligence methods. Some researcher used learning agents, others used neural networks with genetic algorithm [17] [18]. Adibi, Zandieh and Amiri used variable neighbourhood search techniques [19].

3.2. Flexible JSSP. As another generalization of JSSP, the flexible job shop scheduling problem (FJSSP) is also more realistic than the JSSP. The FJSP is an extension of the classical job shop scheduling problem. Flexibility allows an operation to be processed by any machine from a given set. Flexibility also means, the FJSSP will be much more complex than the JSSP, and will be strongly NP hard problem. Despite its combinatorial complexity, the FJSSP is suited to practical job shops, because most machines can perform more than one task type. Moreover, in the FJSSP, jobs can be transferred to other machines when a machine breaks down, there by avoiding blocking and production interruptions [20].

In 2014 Yang and Gu used a novel quadspace cultural genetic Tabu algorithm (QSCGTA) to solve a FJSSP problem. Their algorithm provides a different structure form the original algorithm, in containing double brief spaces and population spaces. The spaces are dealing with different levels of populations globally and locally by applying genetic algorithm and tabu search. They have presented bidirectional shifting for decoding the job shop scheduling process [21]. Yazdani, Amiri, Zandieh also examined the JSSP. They used parallel variable neighbourhood search algorithm to solve the FJSSP makespan time. The parallelization of their algorithm is based on the application of multiple independent searches increasing the exploration in the search space [22]. Jia and Hu also solved a multi object flexible job shop scheduling problem. They used a novel path-relinking algorithm based on the classical Tabu search, combined it with back-jump tracking. The routing problem was identified by problem specific neighbourhood search [20]. Pérez and Raupp used a new hierarchical and heuristic algorithm to solve a multi-objective job shop scheduling problem. Their proposed method is based on the Newtons's method for continuous multi objective unconstrained optimization problems [23].

4. Flow Shop Scheduling problem

The flow shop scheduling problem is a specialization of the job shop problem which considers n jobs to be process on m machines while minimizing some function of completion time of jobs subject to following technological constraints and assumptions:

- Each job consists of *m* operations with some processing times and they are assigned to a machine.
- The order is specified in the assignment of jobs to machines, i.e. each job is first processed on machine 1, then on machine 2, then on machine 3, etc.

Flow shop scheduling problem is one of the most popular machine scheduling problems with extensive engineering relevance, representing nearly a quarter of manufacturing systems, assembly lines, and information service facilities in use nowadays [24], [25] and [26]. In a simple case when there are two machines we can use the Johnson algorithm, which provides an exact solution. However, if there are more than two machines, the task is too complex to solve it with exact methods, therefore heuristic and metaheuristic algorithms have been developed over time. Such methods are simulated annealing (SA), tabu search (TS) and genetical algorithm (GA). The SA method simulates the cooling of metals to find the near-optimal solution in case of the general flow shop problem [27] [28]. Contrarily TS is used in permutation flow shop problems, where all jobs have the same ordering sequence on all machines. It is easier to implement and it does not produce an appreciably worse performance than the optimal general flow shop schedule [29] [30].

After the SA and TS the GA was also used to solve the FSSP. Reeves was the first who used it and his studies carried out is that SA and GA produce comparable results for the flow shop sequencing problem for most sizes and types of problem, but that GA will perform relatively better for large problems, and that it will reach a near-optimal solution rather more quickly [31].

A special case of the FSSP when there are multiple parallel machines per stage usually referred to as the hybrid flow shop problem (HFSSP). Rubén and Vázques-Rodrígues presents a literature review on exact, heuristic and metaheuristic methods that have been proposed for its solution. The paper briefly discusses and reviews several variants of the HFSSP, each in turn considering different assumptions, constraints and objective functions [32].

Several methods have been used to solve the HFSSP, such as the SA that Mirsanei et al. presents a novel algorithm which uses a new effective neighbourhood function to obtain better results [33]. Bozejko el al. presented a TS algorithm which takes advantage of the parallel computing [34]. Engin el al. presented a GA algorithm with new crossover operations which also takes the advantage of the parallel computing [35]. Liu et la. presented a particle swarm optimalization (PSO) algorithm with special local searching operators and an adaptive local search to perform exploitation [36].

Nowadays the extensive use of just-in-time (JIT) system in manufacturing, the performance measure related to both earliness and tardiness penalties. Lot-streaming is one of the effective techniques to meet these requirements. The job splitting into sublots process is usually called lot-streaming, which is one of the effective techniques used to implement the time-based strategy in today’s era of global competition [37]. By splitting into subplots it allows overlapping operations of a job, which reduces machine waiting time.

To solve these problems swarm intelligence algorithms are usually achieves good result. Tseng and Liao proposed a discrete PSO algorithm with a so-called net benefit of movement algorithm which is efficient for obtaining the optimal starting and completion times of sublots for a given job sequence [38]. Pan et al. proposed a discrete artificial bee colony algorithm (ABC) to solve the lot-streaming FSP with the criteria of total weighted earliness and tardiness penalties under both the idling and no-idling cases. Unlike the original ABC algorithm, the proposed DABC algorithm represents a food source as a discrete job permutation and applies discrete operators to generate new neighbouring food sources for the employed bees, onlookers and scouts [39].

5. Conclusions

Nowadays manufacturing and supply companies pay more attention to production logistics tasks and for those solutions. In our paper we have presented scheduling problems, like single machine, parallel machine, and shop scheduling problem. We have presented the methods which are applied for these problems. These scheduling problems are usually NP hard problems, hence exact solutions don't exists, or they are too slow to solve the problems in real-time. Therefore metaheuristic methods have been used, in the late 70's, early 80's, like simulated annealing, tabu search, and genetic algorithms. However, nowadays in most researches the swarm intelligence solutions are used, with parallel computing. The metaheuristic methods don't obtain the optimum, just a near to optimum solution, therefore many new algorithms can be developed in the future.

Acknowledgements
This research was partially carried out in the framework of the Center of Excellence of Mechatronics and Logistics at the University of Miskolc.

References

[1] Pinedo, M. (2012) *Scheduling: theory, algorithms, and systems.* Springer, e-ISBN 978-1-4614-2361-4.
[2] Graham, R. L.; Lawler, E. L.; Lenstra, J. K.; Kan, A. H. G. (1979) *Optimization and approximation in deterministic sequencing and scheduling: a survey.* Annals of discrete mathematics, Vol. 5. pp. 287-326. ISBN 978-0-08-086767-0
[3] Baker, K. R.; Baker, K. R. (1974) *Introduction to sequencing and scheduling.* New York: Wiley. p. 305
[4] Panwalkar, S. S.; Rajagopalan, R. (1992) *Single-machine sequencing with controllable processing times.* European Journal of Operational Research, Vol. 59(2) pp. 298-302.
[5] Li, C. L.; Sewell, E. C.; Cheng, T. C. E. (1995) *Scheduling to minimize release-time resource consumption and tardiness penalties.* Naval Research Logistics (NRL), Vol. 42(6) pp. 949-966.
[6] Biskup, D. (1999) *Single-machine scheduling with learning considerations.* European Journal of Operational Research, Vol. 115(1) pp. 173-178.
[7] Kuo, W. H.; Yang, D. L. (2006) *Minimizing the total completion time in a single-machine scheduling problem with a time-dependent learning effect.* European Journal of Operational Research, Vol. 174(2) pp. 1184-1190.
[8] Chen, Z. L. (1996) *Parallel machine scheduling with time dependent processing times*. Discrete Applied Mathematics, Vol. 70(1) pp. 81-93.
[9] Cheng, T. C. E.; Sin, C. C. S. (1990) *A state-of-the-art review of parallel-machine scheduling research.* European Journal of Operational Research, Vol. 47(3) pp. 271-292.
[10] Cheng, R.; Gen, M. (1997) *Parallel machine scheduling problems using memetic algorithms.* Computers & Industrial Engineering, Vol. 33(3) pp. 761-764.
[11] Mosheiov, G. et al. (2001) *Parallel machine scheduling with a learning effect*. Journal of the Operational Research Society, Vol. 52(10) pp. 1165-1169.
[12] Rabadi, G.; Moraga, R. J.; Al-Salem, A. (2006) *Heuristics for the unrelated parallel machine scheduling problem with setup times.* Journal of Intelligent Manufacturing, Vol. 17(1) pp. 85-97.
[13] Fanjul-Peyro, L.; Ruiz, R. (2010) *Iterated greedy local search methods for unrelated parallel machine scheduling*. European Journal of Operational Research, Vol. 207(1) pp. 55-69.

[14] Vallada, E.; Ruiz, R. (2010) *A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times*. European Journal of Operational Research, Vol. 211(3) pp. 612-622.

[15] French, S. (1982) *Sequencing and scheduling: an introduction to the mathematics of the job-shop.* Chichester: Ellis Horwood.

[16] Dominic, P. D. D.; Kaliyamoorthy, S.; Saravana Kumar, M. (2004) *Efficient dispatching rules for dynamic job shop scheduling.* International Journal of Advanced Manufacturing Technology, Vol. 24. pp. 70–75.

[17] Sha, D. Y.; Liu, C. H. (2005) *Using data mining for due date assignment in a dynamic job shop environment.* International Journal of Advanced Manufacturing Technology, Vol. 25. pp. 1164–1174.

[18] Dimitrov, T.; Baumann, M. (2011) *Genetic algorithm with genetic engineering technology for multi-objective dynamic job shop scheduling problems*. In: Proceedings of the 13th annual conference companion on Genetic and evolutionary computation. ACM, pp. 833-834.

[19] Adibi, M. A.; Zandieh, M.; Amiri, M. (2010) *Multi-objective scheduling of dynamic job shop using variable neighborhood search.* Expert Systems with Applications, Vol. 37(1) pp. 282-287.

[20] Jia, S.; Hu, Z. H. (2014) *Path-relinking Tabu search for the multi-objective flexible job shop scheduling problem.* Computers & Operations Research, Vol. 47. pp. 11-26.

[21] Yang, Y.; Gu, X. (2014) *Cultural-Based Genetic Tabu Algorithm for Multiobjective Job Shop Scheduling.* Mathematical Problems in Engineering, Vol. 2014, Article ID 230719, 14 pages, 2014. doi:10.1155/2014/230719

[22] Yazdani, M.; Amiri, M.; Zandieh, M. (2010) *Flexible job-shop scheduling with parallel variable neighborhood search algorithm.* Expert Systems with Applications, Vol. 37(1) pp. 678-687.

[23] Pérez, M. A. F.; Raupp, F. M. P. (2014) *A Newton-based heuristic algorithm for multi-objective flexible job-shop scheduling problem.* Journal of Intelligent Manufacturing, pp. 1-8. DOI: 10.1007/s10845-014-0872-0

[24] Lee, W. C.; Wu, C. C. (2009) *Some single-machine and m-machine flowshop scheduling problems with learning considerations.* Information Sciences, Vol. 179(22) pp. 3885-3892.

[25] Tavakkoli-Moghaddam, R.; Rahimi-Vahed, A.; Mirzaei, A. H. (2007) *A hybrid multi-objective immune algorithm for a flow shop scheduling problem with bi-objectives: weighted mean completion time and weighted mean tardiness.* Information Sciences, Vol. 177(22) pp. 5072-5090.

[26] Yin, Y. et al. (2009) *Some scheduling problems with general position-dependent and time-dependent learning effects*. Information Sciences, Vol. 179(14) pp. 2416-2425.

[27] Osman, I. H.; Potts, C. N. (1989) *Simulated annealing for permutation flow-shop scheduling*. Omega, Vol. 17(6) pp. 551-557.

[28] Ishibuchi, H.; Misaki, S.; Tanaka, H. (1995) *Modified simulated annealing algorithms for the flow shop sequencing problem.* European Journal of Operational Research, Vol. 81(2) pp. 388-398.

[29] Nowicki, E.; Smutnicki, C. (1996) *A fast tabu search algorithm for the permutation flow-shop problem.* European Journal of Operational Research, Vol. 91(1) pp. 160-175.

[30] Ben-Daya, M.; Al-Fawzan, M. (1998) *A tabu search approach for the flow shop scheduling problem.* European Journal of Operational Research, Vol. 109(1) pp. 88-95.

[31] Reeves, C. R. (1995) *A genetic algorithm for flowshop sequencing*. Computers & operations research, Vol. 22(1) pp. 5-13.

[32] Ruiz, R.; Vázquez-Rodríguez, J. A. (2010) *The hybrid flow shop scheduling problem*. European Journal of Operational Research, Vol. 205(1) pp. 1-18.

[33] Mirsanei, H. S. et al. (2011) *A simulated annealing algorithm approach to hybrid flow shop scheduling with sequence-dependent setup times*. Journal of Intelligent Manufacturing, Vol. 22(6) pp. 965-978.

[34] Bożejko, W.; Pempera, J.; Smutnicki, C. (2013) *Parallel tabu search algorithm for the hybrid flow shop problem.* Computers & Industrial Engineering, Vol. 65(3) pp. 466-474.
[35] Engin, O.; Ceran, G.; Yilmaz, M. K. (2011) *An efficient genetic algorithm for hybrid flow shop scheduling with multiprocessor task problems.* Applied Soft Computing, Vol. 11(3) pp. 3056-3065.
[36] Liu, B.; Wang, L.; Jin, Y. H. (2007) *An effective PSO-based memetic algorithm for flow shop scheduling.* Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, Vol. 37(1) pp. 18-27.
[37] Chang, J. H.; Chiu, H. N. (2005) *A comprehensive review of lot streaming.* International Journal of Production Research, Vol. 43(8) pp. 1515-1536.
[38] Tseng, C. T.; Liao, C. J. (2008) *A discrete particle swarm optimization for lot-streaming flowshop scheduling problem.* European Journal of Operational Research, Vol. 191(2) pp. 360-373.
[39] Pan, Q. K. et al. (2011) *A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem.* Information sciences, Vol. 181(12) pp. 2455-2468.