Available online at www.alphanumericjournal.com

alphanumeric journal

The Journal of Operations Research, Statistics, Econometrics and Management Information Systems

Volume 1, Issue 1, 2013

2013.01.01.OR.03

AN APPLICATION ON FLOWSHOP SCHEDULING*

Sündüs DAĞ†

İstanbul Üniversitesi, İşletme Fakültesi, Üretim Anabilim Dalı, İstanbul

Abstract

Flow shop scheduling problem has been well known as a research field for fifty years. In recent years, researchers have suggested many heuristic procedures to solve this type of problems. Most of these proposed algorithms in flow shop literature were applied to the benchmark problems. Few studies in flow shop literature include a real production application. The aim of this paper is to apply scheduling activity in a real flow shop production line. A cable production line is choosen for the application. All of the jobs are processed with same order which is named as permutational environment. The production line which is composed of eight different machines produces twelve kinds of cable. In other words, the problem size is 12 jobs x 8 machines. The objective of this problem focuses on minimizing total completion time and makespan. An ant colony algorithm is proposed to solve the problem. By changing initial solution of the algorithm, effect on objective function was monitored.

Keywords: Scheduling, Flowshop, Ant colony, Real production environment
Jel Code: C6

Özet

Akış tipi çizelgeleme problemi yaklaşık elli yıldır araştırmacıların fazlasıyla ilgisini çeken bir konu haline gelmiştir. Son yıllarda, bu tip problemlerin çözümüne yönelik birçok meta-sezgisel algoritma önerilmiştir. Çizelgeleme literatürüne bakıldığında, yapılan çalışmalarda geliştirilen algoritmaların kıyaslama problemleri üzerinde denendiği gözlenmiştir. Gerçek üretim problemleri üzerinde yapılan çalışma sayısı çok azdır. Bu çalışmanın amacı, gerçek bir akış tipi üretim hattında çizelgeleme çalışmasının uygulanmasıdır. Uygulama alanı olarak kablo üretim sektöründen bir firma seçilmiştir. Seçilen üretim hattındaki makineler akış tipi üretime uygun bir biçimde sırlanmıştır ve tüm işlerin bu makinelerden geçiş sırası aynıdır. Üretim hattı sekiz makineden oluşur ve bu hatta on iki çeşit kablo üretilmektedir. Problemde amaç, maksimum tamamlanma zamanı ve toplam akış zamanını enküçüklemektir. Problemin çözümü için bir karınca koloni algoritması önerilmiştir. Ayrıca algoritmanın başlangıç çözümü değiştirilerek sonuç üzerindeki etkisi değerlendirilmiştir.

Anahtar Kelimeler: Çizelgeleme, Akış tipi atölye, Karınca koloni algoritması, Gerçek üretim uygulaması
Jel Kodu: C6

** Bu çalışma 14. Uluslararası Ekonometri Yöneylem Araştırması ve İstatistik Sempozyumunda özet bildiri olarak sunulmuştur.*
This paper has been presented at 14th International Symposium on Econometrics Operations Research and Statistics

† skumpas@istanbul.edu.tr

1. Introduction

Flowshop scheduling problem is one of the most studied problem in the scheduling literature. The objective of this problem generally focuses to minimize the makespan. Besides this, total flow time, tardiness, idle time are also considered. First research on flowshop scheduling problem has been done by Johnson (1954). Johnson developed an exact algorithm for n tasks and two-machines flowshop scheduling problem with objective of makespan. After the Johnson’s paper, many exact algorithms and heuristics have been proposed for solving flowshop scheduling problems with different objectives. Ignall and Schrage (1965), Lominicki (1965), Ashour (1970), Mcmahon and Burton (1967), Bansal (1977), Lageweg et al. (1978), Stafford (1988) have been proposed exact solutions for this problem. Exact algorithms are limited by the problem size to solve, as they become impractical for large size problems. When the flow shop scheduling problem enlarges as including more jobs and machines, it becomes a combinatorial optimization problem. Combinatorial optimization problems are in NP-hard problem class, and approximate optimum solutions are preferred for such problems. Several heuristics for the flowshop scheduling problem have been developed by Palmer (1965), Smith and Dudek (1967), Campbell et al. (1970), Gupta (1971), Dannenbring (1977), Nawaz et al. (1983), Hundal and Rajgopal (1988), Widmer and Hertz (1989), Taillard (1990), Ho and Chang (1991), Rajendran and Chaudhuri (1991), Rajendran (1993), Rajendran and Ziegler (1997), Woo and Yim (1998), Lui and Reeves (2001), Framinan and Leisten (2003), Kalczynski and Kamburowski (2007), Li et al. (2009), Rad et al. (2009). In recent years, to obtain better solutions modern metaheuristics have been presented for the flowshop scheduling problem such as simulated annealing (SA), tabu search (TS), genetic algorithms (GA), particle swarm optimization (PSO) and ant colony optimization (ACO). Osman and Potts (1989), Ogbu and Smith (1991), Ishibuchi, Misaki, and, Tanaka (1995), Zegordi, Itoh, and, Enkawa (1995), Wodecki and Bozejko (2002) are well-known studies for SA. Ben-Daya and Al-Fawzan (1998), Grabowski and Pempera (2001), Watson et al. (2002), Grabowski and Wodecki, (2004), Eksioglu, Eksioglu, and, Jain (2008) solved flowshop scheduling problem with TS. Liao, Tseng, and Luarn (2007), Tasgetiren et all. (2007), Jarboui, et al. (2008), Lian, Gu, and Jiao (2008), Kuo et al. (2009), Zhang, Ning, and Ouyang (2010), presented PSO algortihms for flowshop scheduling problem.

Recently, ACO algorithm has become the mostly used technique to solve scheduling problems. The pioneering research has been done by Stutzle (1998). Stutzle (1998) has proposed ACO algorithm, called MMAS, to solve the flowshop scheduling problem with the objective of minimizing the makespan. T’kindt et. al. (2002) have proposed the 2-machine flowshop scheduling problem with the objective of minimizing both the total completion time and the makespan criteria. Rajendran and Ziegler (2004) have developed two ACO algorithms for the the flowshop scheduling problem with the objective of minimizing the makespan and total flowtime of jobs. Ying and Liao (2004) have proposed an ACO algorithm, called ACS, to solve the flowshop scheduling problem with the objective of minimizing the makespan. Yagmahan and Yenisey (2010) have developed a new ACO to minimize makepan and total flowtime of jobs in the flowshop environment.

The aim of this paper is to apply scheduling activity in a real flow shop production line. A cable production line is choosen for the application. All of the jobs are processed with same order which is named as permutational environment. The production line which is composed of eight different machines produces twelve kinds of cable. In other words, the problem size is 12 jobs x 8 machines. An ant colony algorithm is proposed to solve the problem. By changing initial solution of the algorithm, effect on objective function was monitored.

2. Problem Description and Mathematical Formulation

A flowshop production system is defined by more or less continuous and uninterrupted flow of jobs through multiple machines in series. All jobs in flowshop have to follow the same route; in other words, work-flow is unidirectional. The flowshop scheduling problem consists in scheduling n jobs with given processing times on m machines. It is assumed that each job can be processed on only one machine at a time and that each machine can process only one job at a time. Besides, machines are continuously available, all jobs are independent and available for processing at time 0. Setup times are sequence independent and are included in the processing times, or ignored.

The problem is denoted as Fm/prmu/C_{max},ΣF Fm shows machine environment, prmu gives details of processing caracteristics, and C_{max} and ΣF describes the objectives to be minimized.

The objective is to find the job sequence given minimum C_{max} and ΣF values. The notation used in the formulation are as follows:

n total number of jobs to be scheduled

m total number of machines in the flowshop

t_{ij} processing time for job i (i=1,2,..,n) on machine j (j=1,2,.,m)

σ the set of scheduled jobs

C(σ, j) the completion time of partial schedule σ on the j-th machine

C(σi, j) the completion time of job i on machine j when job i is appended to σ.

F flow time

Assuming that each operation is to be performed as soon as possible, for a given sequence of jobs the completion or finishing times of the operations can be found as follows:

The completion times of each job i on the machines are given by

$$C(\sigma_1, 1)= t(\sigma_1, 1) \qquad (1)$$

$$C(\sigma_i, 1)= C(\sigma_{i-1}, 1) + t(\sigma_i, 1) \qquad i=2,\ldots,n \qquad (2)$$

$$C(\sigma_1, j)= C(\sigma_1, j-1) + t(\sigma_1, j) \qquad j=2,\ldots,m \qquad (3)$$

$$C(\sigma_i, j)= \max\{C(\sigma_{i-1}, j), C(\sigma_i, j-1)\} + t(\sigma_i, j) \quad i=2,\ldots,n;\ j=2,\ldots,m \qquad (4)$$

Then the makespan and total flow time can be defined respectively as follows:

$$C_{max}(\sigma)= C(\sigma_n, m) \qquad (5)$$

$$\sum F = \sum_{i=1}^{n} C(\sigma_i, m) \qquad (6)$$

3. Ant Colony Optimization

Ant Colony Optimization (ACO) is an artificial system developed to solve hard combinatorial optimization problems (Stützle and Dorigo 2003). The first ACO was first mentioned by Dorigo's PhD thesis in 1992 with the name Ant System.

The ACO algorithm is developed by the inspiration of ants' ability to find the shortest path between their nests and food sorces. Food search techniques of natural ant colonies have been used for development of this method. The basic principle of the ACO is to follow the trails of a chemical substance which is named as pheromone. While walking, ants excrete pheromone on the ground and follow, in probability, pheromone earlier laid by other ants. A greater amount of pheromone on the path gives an ant a stronger stimulation and thus a higher probability to follow it (Ying and Liao, 2004).

Shorter distance to the destination (i.e., better objective function value) results in greater pheromone level. In other words,the pheromone amount between any two nodes is inversely related to the long of the path.

The first example of such an algorithm is Ant System (AS) developed by Dorigo, Maniezzo, and Colorni (1991a, 1991b, 1996), Colorni, Dorigo, and Maniezzo (1992a, 1992b) for the Traveling Salesman Problem(TSP). Afterwards, several different ACO algorithms are suggested to improve its performance. Here are some of most popular

ISSN 2148-2225
http://www.alphanumericjournal.com/

variations of ACO Algorithms: Elitist Ant System (Dorigo 1992), Ant-Q (Gambardella and Dorigo, 1995), Ant Colony System (ACS) (Gambardella and Dorigo 1996), MMAS (Stutzle and Hoos 1996) and rank-based ant system by (Bullnheimer, Hartl, and Strauss 1999)

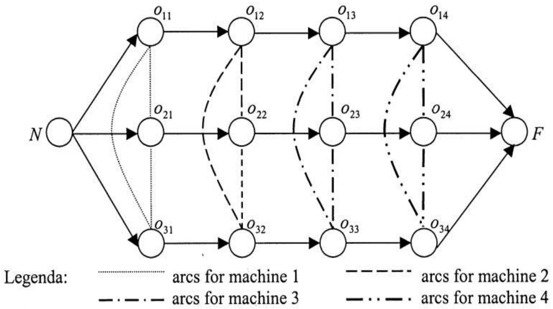
4. Multi-Objective Ant Colony System Algorithm

The first major improvement over the original ant system to be proposed was ant colony system (ACS), introduced by Dorigo and Gambardella (1996) to create a solution for Travelling Salesman Problem. Differs from AS in three points (Dorigo and Gambardella,1997):

- The state transition rule provides a direct way to balance between exploration of new edges and exploitation of a priori and accumulated knowledge about the problem
- Pheromone evaporation and pheremone deposit only takes place on the arcs belonging to the best-so-far tour
- When ants construct a solution a local pheromone updating is applied.

Multi-Objective Ant Colony System Algorithm (MOACSA) are used in this study. MOACSA was proposed for flowshop scheduling problems by Yağmahan and Yenisey in 2010. Small changes have been made in the MOACSA. Initial solution and parameter values are changed according to the problem.

The Fm/prmu/Cmax,ΣF problem can be represented by a disjunctive graph in Figure 1.(Ying and Liao,2010)



N Nest (Starting node)
F Food source or (Final node)
o_{ij} job i on machine j

Figure 1. Representation of disjunctive graph

Fig. 1 gives an instance consist of 4 machines and 3 jobs. machines. In a disjunctive graph, circles represent jobs. Conjunctive arcs (directed arcs) explain precedence constraints among the machines for the same job. Disjunctive arcs (undirected) conform to possible constraints among the jobs on the same machine.

The structure of the MOACSA is given in the following,

1.Initialization: The pheromone trails, the heuristic information and the
parameters are initialized
2. Iterative Procedure:
2.1 A colony of ants determines starting jobs.
2.2 Construct a complete schedule for each ant:
Repeat
Apply state transition rule to select the next processing job
Apply the local updating rule
Until a complete schedule is constructed
2.3 Apply local search process
2.4 Apply the global updating rule
3. Stopping Criteria: If the maximum number of iterations is verified, then STOP; Otherwise go to step 2.

Figure 2. Structure of the MOACSA

4.1. Pheromone trails

The first step of the algorithm is to determine the initial pheromone trails

(τ_0). The initial pheromone trail can be determined either randomly or by an initial solution. In this sudy, firstly, initial pheromone trail is determined randomly. Then, SPT (Jobs with the shortest processing time are scheduled first) and LPT (Jobs with the longest processing time are scheduled first) rules are used for determination of the initial pheromone trail.

Initial pheromone level is calculated by following formulation:

$$\tau_0 = \left[n \cdot \left(C_{\max}(S) \right) + \sum F(S) \right]^{-1} \quad (7)$$

where n is the number of jobs, Cmax(S) is the makespan of the solution and ∑F(S) is the taotal flowtime of the solution for sequence S generated by the SPT or LPT rules.

4.2. Heuristic information

Heuristic information is used in conjuction with the pheromone trails to manage ants' probabilistic solution process. Heuristic information directs ants in the search process for improving computational efficiency and solution accuracy. It is important to use problem specific knowledge. Heuristic algorithms or priority rules can use as heuristic information. The heuristic information used in this study is distance between two jobs determined by SPIRIT (Sequencing Problem Involving a Resolution by Integrated Taboo Search Techniques) rule presented by Widmer and Hertz (1989). According to this rule, the distance between job *i* (*i* = 1, 2, . . . ,*n* U N) and job *u* (*u* = 1, 2,. . . ,*n*) is given by the following equation:

$$d_{ij} = t_{i1} + \sum_{k=2}^{m} (m-k) \cdot \left| t_{ik} - t_{jk-1} \right| + t_{jm} \quad (8)$$

SPIRIT is based on a weighting of the difference between the processing times of jobs. The distance d_{ij} between two jobs is a measure of increase in objective function value if job i scheduled after job j.

For job *i* (*i* = 1, 2,. . . ,*n* U N) and job *u* (*u* = 1, 2,. . . ,*n*) heuristic information is described as follows:

$$\eta(i,u) = \frac{1}{d_{iu}} \quad (9)$$

4.3. Solution Procedure

First, in the initialization step the pheromone trails are initialized, the heuristic information and the parameters are set.

Second, in the iterative process a colony of ants is initially positioned on the starting job. Each ant builds a tour by recurrently applies the state transition rule to select the next job until a complete schedule is built. When constructing a schedule, both pheromone amount and heuristic information is taken into account for determining of the jobs to be selected .

While constructing the schedule, an ant also decreases the amount of pheromone between selected jobs by applying the local updating rule to change other ants schedule. Once all ants have completed their schedules, an adjacent pairwise interchange (API) method is applied to the best schedule to get a better schedule. Afterwards, the global updating rule is applied to increase pheromone between jobs of the best schedule up to the current iteration and decrease pheromone between other jobs. In this way, all the ants will head for a better schedule.

4.4. State transition rule

When building a tour in ACS, an ant k at the current position of node i chooses the next node j to move to by applying the following rule state transition rule:

$$J = \begin{cases} \arg\max_{u \in S_k(i)} \left[\tau(i,u) \right]^{\alpha} \left[\eta(i,u)^{\beta} \right] & \text{if } q \le q_0 \\ J & \text{otherwise} \end{cases} \quad (10)$$

where τ (i; u) is the pheromone trail of edge (i; u), the heuristic desirability η(i,u) =1/d (i,u) is the inverse of the length from node i to node u(d (i; u)), S_k (i) is the set of nodes that remain to be visited by ant k positioned on node i. Besides, α is a parameter which determines the relative importance of pheromone trail (α > 0); β is a parameter which determines the relative importance of heuristic information (β > 0); where *q* is a random number uniformly distributed in [0 .. 1]; q_0 is a parameter ($0 \le q_0 \le 1$) which determines the relative importance of exploitation versus exploration. Additionally, J an operation randomly selected according to a probability distribution, called the random-proportional rule, given in the following equation:

$$P_k(i,j) = \begin{cases} \dfrac{[\tau(i.j)]^{\alpha}[\eta(i,j)]^{\beta}}{\sum_{u \in S_k(i)} [\tau(i.j)]^{\alpha}[\eta(i,j)]^{\beta}} & \text{if } j \in s_k(i) \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

Every time an ant in node i chooses an operation j to move to, it generates a random number q. If $q \le q_0$, then the best job is chosen using the Eq. (10), otherwise the best job is chosen using Eq. (11).

4.5. Local updating rule

While building a solution, ants change their pheromone level between selected jobs by applying the local updating rule of Eq. (12)

$$\tau_{ij} \leftarrow (l - pl) \cdot \tau_{ij} + pl\tau_0 \quad (12)$$

ρl is the local pheromone evaporating parameter (0 < ρl <1).

4.6. Global updating rule

This rule is applied after all ants completed their schedules. The ant which constructed the shortest tour from the beginning of the trial is allowed to deposit pheromone.By means of global updating rule, a greater amount of pheromone trail is left between neighbour jobs of best schedule.The pheromone level is updated by applying the global updating rule of Eq.(13).

$$\tau_{ij} \leftarrow (1 - \rho g).\tau_{ij} + \rho g.\Delta\tau_{ij} \quad (13)$$

$$\Delta\tau_{ij} = \begin{cases} \dfrac{1}{L_b} & if\,(i,j) \in best\,schedule \\ 0 & otherwise \end{cases} \quad (14)$$

ρ*g* is the pheromone evaporating parameter of global updating (0 < ρ*g* < 1). L_b is the objective function value of the best schedule until the current iteration.

4.7. Local search

In some cases, extra steps are needed to improve the quality of the constructed solutions. Performing a local search based on heuristical knowledge to improve the quality of constructed solutions can speed up the the algorithm. Adjacent pairwise interchange method (API) is used for proposed MOACSA. This procedure is obtained by swapping two adjacent jobs.

5. Case Study

In this section, an application of the proposed algoritm for the flow shop -type production system is presented. The production line discussed is composed of eight different machines. Machines in the production line are sequenced in accordance with the flow shop type production system. The sequence of processing a job on all machines is identical and unidirectional for each job. In other words, each machine processes the jobs in the same order.

Machines in the production line perform the following operations respectively:

1. Wire drawing
2. Conducter stranding
3. Insulation
4. Core stranding
5. Filling
6. Armouring
7. Outer sheating
8. Packaging

Twelve kind of cable is produced in the production line. The cable names and characteristics are shown in Table 1.

Table 1- The cable names and characteristics

Name of Cable	Characteristic of Cable
YVZ 3V (3x240/120)	3 Cored-phase cross section 240 mm^2/ neutralcross section 120 mm^2
YVZ 3V (3x240/50)	3 Cored-phase cross section 240 mm^2/ neutralcross section 50 mm^2
YVZ 3V (3x185/95)	3 Cored-phase cross section 185 mm^2/ neutralcross section 95 mm^2
YVZ 3V (3x150/70)	3 Cored-phase cross section 150 mm^2 / neutralcross section 70 mm^2
YVZ 3V (3x120/70)	3 Cored-phase cross section 120 mm^2 / neutralcross section 70 mm^2
YVZ 3V (3x100/50)	3 Cored-phase cross section 100 mm^2 / neutralcross section 50 mm^2
YVZ 3V (3x95/70)	3 Cored-phase cross section 95 mm^2 / neutralcross section 70 mm^2
YVZ 3V (3x95/50)	3 Cored-phase cross section 95 mm^2 / neutralcross section 50 mm^2
YVZ 3V (3x70/35)	3 Cored-phase cross section 70 mm^2 / neutralcross section 35 mm^2
YVZ 3V (3x35/16)	3 Cored-phase cross section 35 mm^2/ neutralcross section 16 mm^2
YVZ 3V (3x25/16)	3 Cored-phase cross section 25 mm^2 / neutralcross section 16 mm^2
YVZ 3V (3x16/10)	3 Cored-phase cross section 16 mm^2 / neutralcross section 10 mm^2

The production is carried out as follows:

Copper comes to the company as electrolytic copper cathode with a 99.7% pureness. Before the production raw material is heated in an oven to 1180 °C and becomes semi-manufactured 8mm copper wire rod. This 8mm package is thinned on the wire drawing machine. Then, thinned wire rod is stranded according to the account of resistance. Stranded wire is insulated with the plastic material. After the this stage, four cored cable insulated is stranded in order to obtain medium-voltage cable. It is covered with the plastic sheating material to become single cable. Then, this single cable is armoured with steel. In the last isolation step, steel armoured cable covered with the PVC material and the final product are packed on the packaging machine.

The problem taken from the company is composed of 12 jobs x 8 machines. Processing time of every job is determined by the employees of the production line. The objective of the study is to obtain best schedule minimizing makespan and total flow time. The problem is solved by MOACSA.

In the previous study, parameter analysis were carried out for the flow shop scheduling problems. The details of the analysis can be found in Dağ (2012). The parameter analysis was made on ten benchmark problems with 20 machines-5 jobs and 20 machines-10 jobs given by Taillard (http://mistic.heig-vd.ch/taillard). The best values of computational analysis for the flow shop scheduling problems with only makespan objective were obtained for α=1, β=0.5, ρl=0.2, ρg=0.1, q_0=0.9 and t_{max} (iteration number)= 1000.

The algoritm is coded in MATLAB 9.0 and implemented on Intel Core i7 1.60 GHz system with the 8 GB DDR3 RAM. The algorithm is repeated with the 10 runs on the problem and the best solution is selected.

Table 2 shows the MOACSA results. Times in the table 1 are given in minute.

Table 2- Results

Criteria	Initial solution rule		
	Random	SPT	LPT
Cmax	2940 min.	2910 min.	2915 min.
∑F	18471 min.	18070 min.	18062 min.

According to the scheduling data taken from the company, makespan value and total flow time value are respectively 3217 min. and 27460 min. Looking at the Table-2, the proposed algorithm provides approximately %10 improvement for makespan and % 30 improvement for total flow time compared with the schedule of company.

Additionally, by changing initial solution of the algorithm, effect on objective function is monitored. When an initial solution (SPT or LPT) is used in the algorithm, solution results are better than the random selection for both makespan and total flow time.

6. Conclusion

This paper presents a real-world scheduling problem. The application is carried out in a well-known cable production company. Production system in the company is arranged in accordance with the flow shop. The problem consists of 12 jobs and 8 machines. The objective of this problem is to minimize both makespan and total flow time.

In recent years, metaheuristic algorithms are proposed to solve this type of problems. In this paper, an ACO algortihm are offered to solve the problem because of showing good performance. The results of the proposed algorithm provides significant improvement for both makespan and total flow time compared with the schedule data taken from the company. The results of the study are able to be used by the company managers for giving direction to the production. As a future research, I plan to make some modifications and improvements on the algorithm and local search method to apply for larger size problems.

References

1. Ashour, S. (1970). An experimental investigation and comparative evaluation of flowshop sequencing techniques. Operations Research, 18, 541–549.
2. Ben-Daya, M., & Al-Fawzan, M. (1998). A Tabu Search Approach for the Flowshop Scheduling Problem. European Journal of Operational Research, 109, 88-95.
3. Bullnheimer, B., Hartl, R.F. & Strauss, C. (1999). A New Rank-based Version of the Ant System: A Computational Study. Central European Journal for Operations Research and Economics, 7, 25–38.
4. Campbell, H.G., Dudek, R.A., & Smith B.L. (1970). A Heuristic Algorithm for the n Job m Machine Sequencing Problem. Management Science, 16, 10-16.
5. Colorni, A., Dorigo, M., & Maniezzo, V. (1992a). Distributed optimization by ant colonies. In F. J. Varela & P. Bourgine (Eds.), Proceedings of the first European conference on artificial life (pp. 134–142). Cambridge: MIT Press.
6. Colorni, A., Dorigo, M., & Maniezzo, V. (1992b). An investigation of some properties of an ant algorithm. In R. Manner & B. Manderick (Eds.), Proceedings of PPSN-II, second international conference on parallel problem solving from nature 509–520).
7. Dag, S. (2012). Optimizatıon of flowshop scheduling problems using heuristic techniques. Istanbul University, Ph.D. Thesis, Istanbul, Turkey: Department of Business Administration (in Turkish).
8. Dannenbring, D.G. (1977). An Evaluation of Flowshop Sequencing Heuristic", Management Science, 23, 1174-1182.
9. Dorigo, M., Maniezzo, V., & Colorni, A. (1991a). Positive feedback as a search strategy. Technical Report, 91-016, Italy: University of Milan.
10. Dorigo, M., Maniezzo, V., & Colorni, A. (1991b). The ant system: An autocatalytic optimizing process. Technical Report, 91-016 (Revised), Italy: University of Milan.
11. Dorigo, M., Maniezzo, V., & Colorni, A. (1996). The ant system: Optimization by a colony of cooperating agents. IEEE Transactions on
12. Systems, Man, and Cybernetics – Part B, 26, 29–41.
13. Dorigo, M., & Gambardella, L.M. (1996). A Study of Some Properties of Ant-Q. In Proceedings of PPSN Fourth International Conference on Parallel Problem Solving From Nature, 656-665.
14. Dorigo, M., & Gambardella, L.M. (1997). Ant Colony System: A Cooperative Learning Approach to the TSP. IEEE Transactions on Evolutionary Computation, 1, 1-24.
15. Eksioglu, B., Eksioglu, S. D., & Jain, P. (2008). A tabu search algorithm for the flowshop scheduling problem with changing neighborhoods, Computers and Industrial Engineering, 54, 0360-8352.
16. Framinan, J.M., & Leisten, R. (2003). An efficient constructive heuristic for flowtime minimisation in permutation flow shops. Omega-International Journal of Management Science 31, 311-317.
17. Gambardella, L. M., & Dorigo, M. (1995). Ant-Q: A reinforcement learning approach to the traveling salesman problem. In A. Prieditis & S. Russell (Eds.), Proceedings of the Twelfth international conference on machine learning (ML-95) (pp. 252–260). Palo Alto: Morgan Kaufman
18. Gambardella, L. M., & Dorigo, M. (1996). Solving symmetric and asymmetric TSPs by ant colonies. In Proceedings of the 1996 IEEE international conference on evolutionary computation (pp. 622–627). Piscataway: IEEE Press
19. Grabowski, J., & Wodecki, M. (2004). A Very Fast Tabu Search Algorithm for The Permutation Flowshop Problem with Makespan Criterion. Computers and Operations Research, 31, s.1891–1909.
20. Grabowski, J., & Pempera, J. (2007). The permutation flow shop problem with blocking. Omega, 35, 302-311.

21. Gupta, J.N.D. (1971). A Functional Heuristic Algorithm For Flow-Shop Scheduling Problem. Operations Research, 22, 39-47.
22. Ho, J.C., & Chang, Y. (1991). A New Heuristic For The n-Job, m-Machine Flowshop Problem. European Journal of Operations Research, 52, 194-202.
23. Hundol, T.S., & Rajgopal, J. (1988). An Extension of Palmer's Heuristic for the Flow shop Scheduling Problem. International Journal of Production Research, 26, 1119-1124.
24. Ignall, E., & Schrage, L. (1965). Application of the branch and bound technique to some flowshop scheduling problems. Operations Research, 13, 400-412.
25. Ishibuchi H., Misaki, S., & Tanaka, H. (1995). Theory and Methodology Modified Simulated Annealing Algorithms for The Flowshop Sequence Problem. European Journal of Operations Research, 81, 388-398.
26. Jarboui,B., Eddaly,M., & Siarry,P. (2009). An estimation of distribution algorithm for minimizing the total flowtime in permutation flowshop scheduling problems. Computers & Operations Research 36, 2638-2646.
27. Johnson, S.M. (1954).Optimal two three-stage production schedule with setup times included. Naval Research Logistics Quarterly, 1, 61-68.
28. Kalczynski, P., & Kamburowski, J. (2008). An improved NEH Heuristic to Minimize Makespan in Permutation Flowshops. Computers and Operations Research, 35, 3001-3008.
29. Lageweg, B. J., Lenstra J., K., & Rinnooy Kan A., H, G. (1978). A general bounding scheme for the permutation flow-Shop problem. Operations Research, 26, 53-67.
30. Li, X.P., Wang,Q., & Wu, C. (2009) Efficient composite heuristics for total flowtime minimization in permutation flow shops. Omega-International Journal of Management Science 37 (1), 155-164.
31. Liu, J., & Reeves, C.R. (2001). Constructive and composite heuristic solutions to the P// Σ Ci scheduling problem. European Journal of Operational Research 132,439 – 452 .
32. Lominicki, A.Z. (1965). A Brunch and bound algorithm for the exact solution of the three-machine scheduling problem. Operational Research Quarterly, 16, 439–452.
33. McMahon, G. B., & Burton, P. (1967). Flowshop scheduling with branch and bound method. Operations Research, 15, 473–481.
34. Nawaz, M., Enscore, J.E., & Ham, İ. (1983). A Heuristic Algorithm For The m-Machine, n-Job Flow-Shop Sequencing Problem. Omega, 11, 91-95.
35. Ogbu, F., & Smith, D. (1990). Simulated Annealing for The permutation Flowshop Problem. Omega, The International Journal of Management Science, 19, 64-67.
36. Osman, H.İ., & Potts, C. (1989). Simulated Annealing for Permutation Flowshop Scheduling. Omega, The International Journal of Management Science, 17, 551-557.
37. Palmer, D.S. (1965). Sequencing jobs through a multi-stage process in minimum total time a quick method of obtaining a near optimum. Operational Research Quarterly, 16, 101-107.
38. Rad, S.F., Ruiz, R. & Boroojerdian, N. (2009). New High Performing Heuristics for Minimizing Makespan in Permutation Flowshops. Omega, 37, 331-345.
39. Rajendran, C., & Chaudhuri D. (1991). An Efficient Heuristic Approach to the Scheduling of Jobs in a Flowshop. European Journal of Operational Research, 61, 318-325.
40. Rajendran, C. (1993). Heuristic algorithm for scheduling in a flowshop to minimize total flowtime. International Journal of Production Economics 29,65–73.
41. Rajendran,C., & Ziegler, H. (1997a). Heuristics for scheduling in a flowshop with setup, processing and removal times separated. Production Planning and Control 8,568–576.
42. Rajendran, C., & Ziegler, H. (2004). Ant-colony algorithms for flowshop scheduling to minimize makespan/total flowtime of jobs. European Journal of Operational Research, 155, 426–438.
43. Smith, R.D., & Dudek, R. A. (1967). A General algorithm for the solution of the n job, m machine sequencing problem of the flowshop. Operations Research 15, 71–82.
44. Stafford, E.F. (1988). On the development of a mixed-integer linear programming model for the standard flowshop. Journal of the Operational Research Society 39, 1163–1174.
45. Stutzle, T., & Hoos, H. (1996). Improving the ant system: A detailed report on MAX-MIN ant system. Technical Report, AIDA-96-12 (Revised version), Darmstadt: Darmstadt University of Technology.
46. Stuetzle,T., (1998). An ant approach for the flow shop problem. In: Proceedings of the 6th European Congress on Intelligent Techniques and Soft Computing (EUFIT _98),vol. 3. Verlag Mainz,Aachen,Germany, 1560–1564.
47. Stutzle, T., & Dorigo, M. (2003). The ant colony optimization metaheuristic: Algorithms, applications, and advances. In F. Glover & G. Kochenberger (Eds.), Handbook of metaheuristics (pp. 251–285). Norwell, MA: Kluwer Academic Publishers.
48. Taillard, E. (1990). Some Efficient Heuristic Methods for The Flowshop Sequencing Problem. European Journal of Operational Research, 47, 65-74.
49. Taşgetiren, M.F., Liang, Y.C., Şevkli M. & Gençyılmaz, G. (2007). A Particle Swarm Optimization Algorithm for Makespan and Total Flowtime Minimization in the Permutation Flowshop Sequencing Problem. European
50. T'kindt, V., Monmarche, N., Tercinet, F., & Laugt, D. (2002). An Ant Colony Optimization Algorithm to Solve a 2-machine Bicriteria Flowshop Scheduling Problem. European Journal of Operational Research,142, 250–257.
51. Watson, J.-P., Barbulescu, L., Whitley, L.D., & Howe, A.E. (2002). Contrasting structured and random permutation flow-shop scheduling problems: Search-space topology and algorithm performance. INFORMS Journal on Computing 14, 98–123
52. Widmer, M., & Hertz, A. (1989). A New Heuristic Method for The Flowshop Sequencing Problem. European Journal of Operational Research, 41, 186–193.

53. Wodecki, M. & Bozejko, W. (2002). Solving the Flow Shop Problem by Parallel Simulated Annealing. In Parallel Processing and Applied Mathematics, 2328, 236–244.
54. Woo, H. S., & Yim, D.S. (1998). A heuristic algorithm for mean flowtime objective in flowshop scheduling. Computers and Operations Research 25, 175–182.
55. Yağmahan, B., & Yenisey M.M. (2010). A Multiobjective Ant Colony System Algorithm for Flowshop Scheduling Problem. Expert system with Aplication, 37, 01361-1368.
56. Ying, K.C. ve Liao, C.J., 2004 An Ant Colony System for Permutation Flowshop Sequencing", Computers and Operations Research, C:31, No:5, s.791-801.
57. Zhang, C., Ning, J., & Ouyang, D. (2010). Hybrid Alternate Two Phases Particle Swarm Optimization Algorithm for Flow Shop Scheduling Problem. Computers and Industrial Engineering, 58, 1-11.
58. Zegordi, S.Y., Itoh, K. & Enkawa, T. (1995). Minimizing Makespan for Flowshop Scheduling by Combining Simulated Annealing with Sequencing Knowledge. European Journal of Operational Research, 85,. 515-531.