1. Introduction

There is a primary assumption on many flowshop scheduling problems in the literature called permutation in the order of jobs [1]. However, some papers emphasize the significance of non-permutation schedules and their effect on improving the schedule performance measures [2,3,4]. When there are more than three machines, the permutation schedules are no longer dominant [5]. The non-permutation schedules may provide much better solutions in comparison with the permutation schedules, if the problems are subjected to some special assumptions such as missing operations of jobs[a], or have non-regular performance measures [2,6].

Almost all existing researches focus on the permutation flowshop schedules and there is insufficient study dedicated to the non-permutation flowshop scheduling problems in the literature [7,8]. In each m-machine n-job non-permutation flowshop scheduling problem, the total number of feasible schedules tends to $(n!)^m$, and in the permutation case, the number of feasible solutions is reduced to $n!$ [9]. Therefore, modeling and solving the non-permutation problems are harder.

As the problem size increases, finding the best permutation schedule itself becomes quite difficult [10]. Obviously, identifying the best non-permutation sequence is more complex and difficult. In the literature, heuristic methods have usually been used to solve the non-permutation flowshop scheduling problems. Some of them are [5,11,12,13,14]. In this paper, we first present a mixed integer linear programming model to obtain an optimal schedule for the m-machine n-job flowshop scheduling problem with the objective function of the makespan ($F//C_{max}$). An efficient heuristic method is next developed to solve the problem and find a near-optimal solution. An ant colony optimization (ACO) algorithm is also proposed for solving the problem. We have revised the ACO procedure developed by Sadjadi et al. [15] to cover the non-permutation schedules. In order to evaluate the performance of the proposed heuristics, the methods are implemented using some test problems and the results of computational experiments are presented.

2. Assumptions and notations

Assumptions considered in this paper and associated notations are as follows:

1. A job consists of several operations, each to be performed on a specified machine. The processing times of the operations are given.
2. Set-up times between operations are negligible or included in the processing times (sequence-independent setup times).
3. Machines are available at all times.
4. All jobs are available for processing at time zero.
5. No machine can process more than one job at the same time.
6. A job can not be performed by more than one machine at the same time.
7. Preemption is not allowed, i.e. a started operation can not be interrupted during its performance.
8. There are no precedence constraints among the jobs.
9. Machine breakdowns and maintenances do not occur.
10. There is only one of each type of machine.
11. All programming model parameters are deterministic and there is no randomness.

$\boldsymbol{I}$: number of jobs,
$\boldsymbol{K}$**:** number of machines,
$\boldsymbol{i}$**:** denotes ith job; i=1,…,I,
$\boldsymbol{k}$**:** denotes kth machine; k=1,…,K,
$\boldsymbol{t_{ik}}$**:** processing time of job i on machine k,
$\boldsymbol{q_{ijk}}$**:** completion time of job i on machine k which is processed in position j of the machine sequence,
$\boldsymbol{x_{ijk}}$**:** binary variable taking value 1 if job i on machine k is processed in position j and 0 otherwise.

a- The flowshop scheduling problem with missing operations of jobs is sometimes known as flowline-based manufacturing system (FBMS) in the literature.