Machine Scheduling Problem (MSP)

1. Introduction

There are many definitions for machine scheduling, but the simplest one for understanding is that, the allocation of resources over time to perform a collection of tasks (Baker [13]). Resources and tasks are called machines and jobs respectively and both of them can take many forms. For example: we can consider a computer (or computers) as a machine (or machines) and the programs that are to be run on that computer (or computers) as jobs. Another example: we can consider hospital equipment's as machines and the patients in that hospital as jobs.

Scheduling, generally speaking, means to assign machines to jobs in order to complete all jobs under the imposed constraints. The problem is to find the optimal processing order of these jobs on each machine to minimize the given objective function. There are two general constraints in classical scheduling theory [19]. Each job is to be processed by; at most, one machine at a time and each machine is capable of processing at most one job at a time. A schedule is feasible if it satisfies the two general constraints, and also if it satisfies the various requirements relating to the specific problem type. The problem type is specified by the machine environment, the job characteristics and an optimality criterion.

1.1 MSP Terminology

The function to be maximized or minimized with or without subject to certain constraints is called the **objective function**. A schedule σ for the minimum problem is said to be **feasible** if it satisfies the corresponding constraints. The set of all feasible schedules are called the **set of solutions**.

For many years, scheduling researches focused on **single (objective)** performance measures. In most real world, a scheduling application, with more than one performance measure is of interest. The **multi-criteria** (multi-objective) have received significant attention in recent years [1].

1.2 MSP Notations

We assume that there are n jobs which we denoted by 1,2,…,n, these jobs are to be scheduled on a set of machines that are continuously available from time zero onwards and that can handle only one job at a time. In this thesis we only state here the notation that is used for the single machine, job j (j=1,2, …,n) has:

p_j: which means that it has to processed for a period of length p_j.

d_j: a due date, the date when the jobs should be completed, the completion of job after its due date is allowed, but a penalty is incurred, and when due date is constant for all jobs then called common due date.

$\bar{d}_j$: when the due date absolutely must be met, it is referred to as a deadline.

r_j: a release date of job j, i.e., the earlier time at which the processing of job can begin.

W_j: a weight, denoting the importance of a job j relative to another job.

Now for given sequence of jobs we can compute for job j, j=1,…,n.

- The completion time C_j.
- The lateness $L_j=C_j - d_j$.
- The tardiness $T_j= \max\{C_j - d_j, 0\}$.
- The earliness $E_j= \max\{d_j - C_j, 0\}$.
- The late work $V_j=\min\{T_j , p_j\}$

The following performance criteria appear frequently in the literature [49]. For a given schedule σ we compute:

- $C_{max}(\sigma) = \max_j(C_j)$ (maximum completion time).
- $E_{max}(\sigma) = \max_j(E_j)$ (maximum earliness).
- $L_{max}(\sigma) = \max_j(L_j)$ (maximum lateness).
- $T_{max}(\sigma) = \max_j(T_j)$ (maximum tardiness).
- $V_{max}(\sigma) = \max_j\{V_j\}$ (maximum late work).
- $\sum_j (W_j) C_j(\sigma)$ (total (weighted) completion time).
- $\sum_j (W_j) E_j(\sigma)$ (total (weighted) earliness).
- $\sum_j (W_j) T_j(\sigma)$ (total (weighted) tardiness).

The following sequencing rules and basic concepts are used in this thesis:

SPT rule [85]: Jobs are sequencing in non-decreasing order of p_i, this SPT (shortest processing time) rule is used to minimize ΣC_i for $1/ /\Sigma C_i$ problem.

EDD rule [53]: Jobs are sequencing in non-decreasing order of d_i, this EDD (Earliest due date) rule is used to minimize T_{max} for $1/ /T_{max}$

Minimum Slack Time (MST) rule [21]: Jobs are sequenced in non-decreasing order of slack times (s_j) $s_j=d_j-p_j$, this rule well known for solve the problem $1//E_{max}$.

Definition (1) [53]: The term "**optimize**" in a multi-criteria decision making problem refers to a solution around which there is no way of improving any objective without worsening the other objective.

Definition (2) [46]: A feasible schedule σ is **Pareto optimal** (PO), or non-dominated (efficient) with respect to the performance criteria f and g if there is no feasible schedule π such that both $f(\pi)\leq f(\sigma)$ and $g(\pi)\leq g(\sigma)$, where at least one of the inequalities is strict.

1.3 MSP Classification

A notation which is commonly used to formulate scheduling problem is based on three fields: α / β / γ [45]. In this notation α describes the machine environment, i.e., the structure of the:

- Single machine or multiple machines.
- Identical or different machines.

The field $\beta \in \{pmtn, r_j, prec\}$ indicates certain job characteristics [11]:

- If pmtn is present, then preemptions are allowed; otherwise, no preemptions are allowed.
- If r_j is present, then each job may have different release dates. Otherwise, all jobs arrive at time zero.
- If prec is present, then there is precedence relation $\prec$ among the jobs.

The field $\gamma \in \{f_{max}, \sum f_j\}$ refers to the optimality criterion or the objective, the value which is to be optimized (minimized). There are two types of single criteria, the first is maximum function $f_{max}\in\{C_{max}, L_{max}, T_{max}, E_{max}, V_{max}, V_{max}^{w}, T_{max}^{w}\}$. The second is the sum of functions, where we need to find a summation object $\Sigma f_j \in\{\Sigma C_j, \Sigma E_j, \Sigma T_j, \Sigma V_j, \Sigma U_j\}$ and weighted summation objects $\Sigma W_j fj=\{\Sigma W_j C_j, \Sigma W_j E_j, \Sigma W_j T_j, \Sigma W_j V_j, \Sigma W_j U_j\}$.

1.4 Examples of Scheduling Problem

There are more examples of three fields classification of scheduling problem:

- $1/p_{rec}/\sum W_j U_j$ (is the problem of scheduling jobs with precedence constraints on a single machine to minimize the (weighted) number of tardy jobs)
- $1/r_j/\sum V_j$ (is the problem of scheduling jobs with release dates on one machine to minimize the total late work).

- $1//\sum W_jC_j + E_{max}$ (is the problem of scheduling jobs on a single machine to minimize the sum of the total (weighted) completion times and maximum earliness).