جمهورية العراق
وزارة التعليم العالي
والبحث العـلـــمي
جـامــعـة ديــالى
كــليــة الــعــلـوم
قسم علوم الرياضيات

تصغيردالة متعددة الاهداف على مسألة جدولة ماكنة واحدة

رسالة مقدمة الى مجلس كلية العلوم \ جامعة ديالى

وهي جزء من متطلبات نيل درجة الماجستير في علوم الرياضيات

من قبل

ندى محسن خليل

أشراف

أ.م .د.عدويه علي محمود النعيمي

ربيع الثاني ١٤٤٣ العراق تشرين الثاني ٢٠٢١

CHAPTER ONE
MACHINE SCHEDULING PROBLEM

1.1 Introduction

It can be recognized through the literature that there are many contributions have dealt with the problem of machine scheduling. This fact was built since the mathematical formulation can be validated in accordance with machines assignment. However, the tasks (Jobs) throughout such algorithms over the entire time for optimizing performance criteria are validated here approximately and sometimes are exactly.

More precisely, the literature of this field have shown two common trends in term of scheduling research which are:

- Deterministic: in which, all the problems that have to be solved are involving certain known parameters.
- Stochastic: in which, some of the problems that have to be solved are involving certain un-known parameters.

1.1.1 Machine Scheduling

Many machines are normally modeled through the implementation of deterministic scheduling and large range of data are involved. In addition, such approach is used to formulate the problems through a specified time using the desired optimum way. The schedule of implicit assumption is usually implemented immediately after its development.

Finally, it can be deduced from the surveyed past experience that this approach is not preferred in many manufacturing environments since it requires many efforts and didn't give the desired accuracy for some cases

[14].

1.1.2 Concepts of Scheduling

Assume n is the number of jobs that is needed to be scheduled which can be done from time zero of machine starting onwards and assuming that this machine can do one job in one time. However, the following notations are assumed throughout the present study which is devoted for single machine jobs j (j=1,…,n)

p_j : processing time for job j.

r_j : Release date , i.e the earliest time at which P_j begins.

d_j: Due date for job j.

In addition, for a known defined jobs in the problem, the following can be computed:

- C_j: The completing time.
- F_j: The flow Time = $C_j - r_j$.
- L_j: The Lateness =$C_j - d_j$.
- T_j: The Tardiness = $\max\{C_j - d_j, 0\}$.
- E_j: The Earliness= $\max\{d_j - C_j, 0\}$.
- U_j: The Unit penalty $= 1$ if $C_j > d_j$ and $U_j = 0$ if $C_j \le d_j$.

For a known scheduling δ, the following performance criteria appears in frequent times in the literature [15].

1. $C_{max}(\delta) = \max_j (C_j)$ Maximum completion time.
2. $E_{max}(\delta) = \max_j (E_j)$ Maximum earliness.
3. $L_{max}(\delta) = \max_j (L_j)$ Maximum lateness.
4. $T_{max}(\delta) = \max_j (T_j)$ Maximum tardiness.
5. $\sum W_j C_j(\delta)$= total (weighted) completion time.

6. $\sum(W_j)E_j(\delta)$ = total (Weighted) earliness.
7. $\Sigma(W_j)T_j(\delta)$ = total (Weighted) tardiness.
8. $\Sigma(W_j)U_j(\delta)$ = total (Weighted) number of tardy jobs. excluding $E_{\max}$ and $\Sigma(W_j)E_j$ all such criteria are legitimate, i.e, implementing idle time into the schedule will not reduce the value of the objective function.

1.1.3 The Problem Classification

Three fields use the common notations that are widely utilized for formulating scheduling problems: $\alpha/\beta/\gamma$[6]. In such notations, $\boldsymbol{\alpha}$ defines the environment of machine, i.e. the structure of the

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

The restrictions of the scheduling problem and other processing conditions are defined by the field β. Among the constraints that can exist ([16], [17]):

- Is the preemption possible or not, which refers to the possibility of reassuming / interruption for processing.
- Are restrictions like (release date, due date, setup times, etc.) known and defined or not, and if such restrictions are deterministic or stochastic.
- Are the jobs arrival of the problem is fixed or dynamic.

Moreover, the quality of scheduling which is characterized by the criterion (γ) includes:

- Min. total completing period or make span $C_{\max}$.
- Max. Earliness $E_{\max} = \max(E_j)$ for $j = 1, \dots, n$.
- Max. Tardiness $T_{\max} = \max(T_j)$ for $j = 1, \dots, n$. etc.

1.1.4 Scheduling Examples

The following are some examples for three fields regarding scheduling classification.

- The $1/rj/\Sigma W_i C_i$ is a minimizing problem for weighted completion time of one machine that is subjected to non- trivial release dates.
- The $1//\Sigma W_i C_i + T_{\max}$ is the problem for finding jobs processing orders for one machine for minimizing the summation “total weighted completion time” and the maximum Tardiness.
- The F2//Cmax is the problem of minimizing maximum completion time on two machine flow shop.

1.1.5 Problem kinds of Machine Scheduling

In fact, there are many common trends for classifying due to the wide spectrum of such problems. However, the proposed styles through the literature depending upon wide range of dimensions [18].

Moreover, the number of the available machines as well as the organizing nature of these machines can be considered as good example for these dimensions (see figure (1.1)).

In addition, the sequencing problems of single machine can be also considered as representative example. In these problems, all jobs must be executed on a single machine and no two jobs can be completed at the same time. There are four classes of job shop scheduling problems [19]:

1.Single Machine Scheduling: The single machine scheduling problem involves scheduling a set of jobs to a single resource. This is accomplished by determining a sequence that includes each job and it assigns the jobs to its source. Each job can be given a priority, ready time, processing time and due date. The value of the performance measure can be computed on the base of this information and the sequence of jobs. This problem grows in complicity at an exponential rate as the number of jobs to be Scheduling increases.

2. Flow Shop Scheduling: There are n jobs that have to be processed in each of the m machine i.e., each job consists of m steps or operations. The processing of each job is carried out in the same sequence through the processing stages, i.e., from the first to the last machine the processing stages. The problem is to find the sequence in which the jobs should be processed so that the given objectives are achieved.

3. Job Shop Scheduling: This is a general case of the flow shop scheduling problem, in which the sequencing of each job through the machines is not necessarily identical. As in a flow shop, there are also n jobs consisting of m operation within each job are predefined and fixed, for example $J_m/d_j/C_{max}$, denotes a job shop configuration in which all jobs have a due date and the objective is to minimize the maximum completion time.

4.Open shop scheduling: The open shop is a more general case of the job shop scheduling problem. As before, there are n jobs consisting of m steps to be processed in m machines. The sequencing of each job through the machine is different and finding the optimal sequence for each of the n jobs is also part of the problem. Since the sequence of steps within each job has to be determined in addition to the jobs processing schedule.

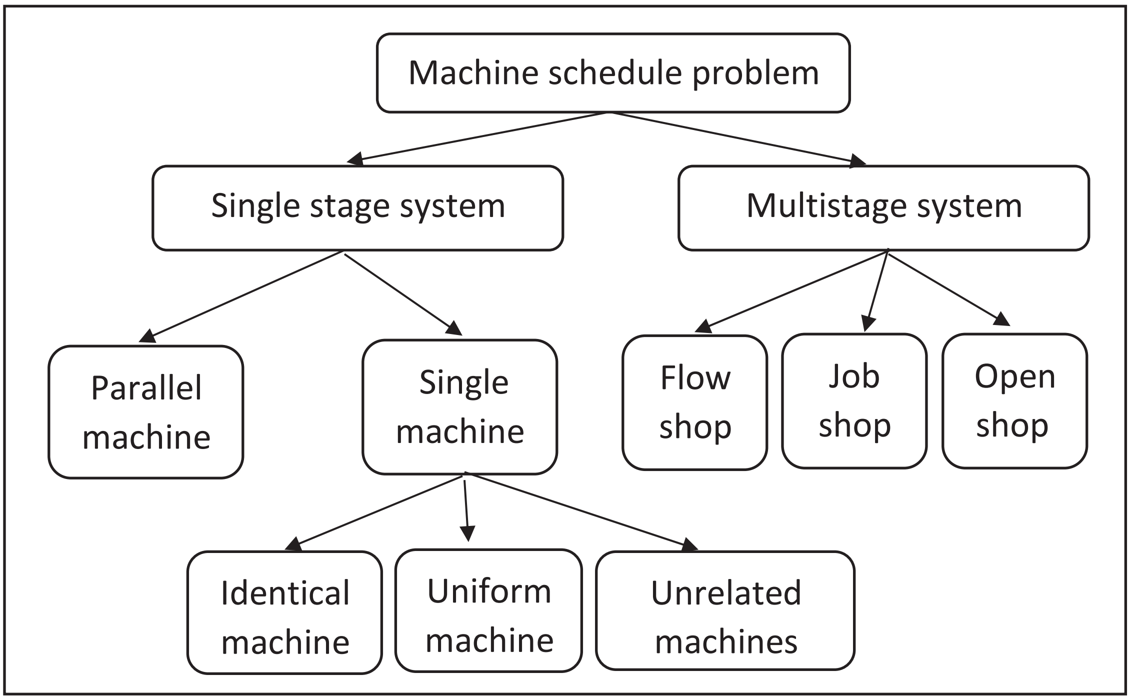

Figure (1.1) classification of machine schedule problems [1].

1.2 Solution Approaches

1.2.1 Basic Rules and Main Results for MSP

The following are the most famous contributions that proposed to find solutions to the problem of scheduling:

1. The shortest processing time or (SPT) which is also known as (Smith rule). In such rule, the jobs of the problem are sequenced in non – decreasing order of p_j. However, such rule is used usually to solve the problem $1//\sum C_j$[20]. In addition, this sequencing can include the weight ratio to solve the problem $1//\sum W_j C_j$ in non-decreasing p_j/w_j.
2. The earliest due date (EDD rule) . In this rule, the jobs of the problem are sequenced in non – decreasing order of d_j and can solve the $1//L_{max}$ problem to $1//T_{max}$ problem [21].

3. **The** longest processing time or (LPT). In such rule, the jobs of the problem are also sequenced in non – increasing order of p_j [22]. Moreover, this rule can minimize $\sum_j E_j$ to $1/ C_j \leq d_j/\sum_j E_j$ problem.
4. The minimum slake time or (MST). In this rule, the jobs of the problem are sequenced in non – decreasing order (with respect to their slack times). In this approach, the slack times can be defined as $s_j = d_j - P_j$ and the problem of $1//E_{max}$ problem can be solved concerning ready time set at zero in single machine environment [5].

1.2.2 Mathematical Programming

There are some mathematical programming techniques utilized for solving the combinatorial optimization problem(COP), these techniques are also used for scheduling problems [23].

A (mixed) programming may be utilized for solving several scheduling challenges; in that case, standard integer programming solution procedure can be used. Both Conway et al. [22] and Baker and Scharge [24] discuss integer programming formulation of scheduling problems.

Dynamic programming (DP) method is implicit enumeration technique and it can be applied to any optimization problem that at a stage is found by solving the derived recurrence relation that is made for this problem [23]. There are some difficulties for this method, one of them is the difficulty of finding a good way for brake down problem into stages so that a convenient computation is rather large, which means that the computation grows to exponential rate with increasing in the size of problem.

Moreover, there are several kinds of problems that can be solved by combinatorial optimization. One of the most well – known ones is the branch and bound (BAB) method which can be used for solving scheduling problems which can be considered as a standard context for combinatorial optimization [25].

This method is the typical example of the implicit enumeration approach, which can find an optimal solution by systematically examining subset of feasible solution. The procedure is usually described by means of search tree with nodes that correspond to these subsets. From each node for a partially complete solution there grows a number of new branches which replace the original one by set of new smaller problems that are mutually exclusive. We have two common types of branching:

1- The forward branching, that is the jobs are sequenced one by one from the beginning.

2- The backward branching, that is, the jobs are sequenced one by one from the end.

To minimize an objective function Z, for a particular scheduling problem, the BAB method successively partitions the problem into subsets by using a branching procedure and computers bound by using a lower bounding procedure. By these procedures we exclude the subsets which are found not to include any optimal solution. This eventually leads to at least one optimal solution. The bounding procedure is used to calculate a lower bound LB on the solution to each generated sub problem. For each node we calculate a LB which is the cost of the scheduling jobs (depending on the objective function and the cost of the unscheduled jobs (depending on the derived lower board)). This node has a value LB greater than or equal to the upper bound UB the upper bound is usually defined as the minimum of the values of all feasible solutions currently found. Then this node is dominated, and we choose one of the remaining nodes that have the least LB. If the branching ends at a complete sequence of jobs, this sequence is evaluated and if its value is less than the current UB, this UB is reset to take that value. We repeat the procedure until all nodes have been considered, that is, LB > UB for all nodes in the search tree. A feasible solution with this UB is an optimal solution for this problem.

In BAB procedure one can introduce dominance rules (if possible) to specify whether a node can be eliminated before computing its LB which reduce the computation time by ignoring the calculations of the dominated nodes and their successors.

1.2.3 The Heuristic Method

It's obvious from the above section, that the solving of specific problem by computational requirement might obligate more time than is habitual taking for large problems. Indeed, there is no surety to find a fast solution in comparatively small matter. Occasionally we utilize a heuristic scheduling instead of optimal schedule, so, we can obtain near optimal solution. Heuristic method defined by Reeves [26] as follows: a heuristic is a technique which seeks good (i.e. near optimal) solution at a reasonable computational cost without being able to guarantee either feasibility or optimality, or even in many cases to state how close to optimality a particular feasible solution.

1.3 Multi-Criteria Scheduling

Many researchers have been working on multiple criteria scheduling with the majority of work being on bi-criteria scheduling. Using two criteria usually makes the problem more realistic than using a single criterion. One criterion can be chosen to represent the manufacturer's concerns while the other could represent consumer's concern. There are several papers that review the multiple criteria scheduling literature. Nagar et al. (1995) [3], and T'kindt and Billaut (1999) [27] review the problem in its general form, whereas Lee and Vairaktarakis (1993) [28] review a special version of the problem, where one criterion is set to its best possible value and the other criterion is tried to be optimized under this restriction. Hoogeveen (1992) [6] studied a number of bi-criteria scheduling problems. Most real-world optimization problems have several, often conflicting objectives. Therefore, the optimum for a multiobjective problem is typically not a single solution. It is a set of solutions that trade-off between objectives.

There are three types of multiple criteria problems that can be identified:

The first of these types of problems involves identifying all sequences that minimize a first objective. One of these sequences that minimize a second objective is chosen as the optimal sequence for that problem. Suppose that we have selected the two performance criteria, say f and g that we want to take into account. If f is more important than g, then the approach is to find the optimum value with respect to criterion f say $f*$, and choose from among the set of optimum schedules for f the one that performs best on g, such an approach is called hierarchical optimization or lexicographical optimization, which is denoted by Lex (f, g) in the third field of the $\alpha/\beta/y$ notation scheme. This approach is called hierarchical approach [4] .

The second of these multiple criteria problems, when the criteria are weighted differently, is an objective function that can be defined as the sum of weighted functions and transforms the problems in to a single criterion scheduling problem. This approach is called simultaneous optimization, along with the third type of multiple criteria problems, also in these multiple criteria problems, both criteria are going to be considered as equally important. The problem is to find a sequence that dose well on both objectives. To solve this problem, the main idea behind it, is that we select from the set of solutions a subset that contains efficient solutions.

Theorem $(\mathbf{1.3.1})[\mathbf{7}]$:

If the composite objective function $\mathrm{F}(\mathrm{f},\mathrm{g})$ is linear, then there exists an extreme schedule that minimizes F.

المستخلص

قدمنا في هذة الرسالة مسألة جدولة متعددة المقاييس على ماكنة واحدة. المقاييس الثلاثة هي وقت الاتمام الكلي $\sum C_j$ ،أعظم تبكير E_{max} وأعظم تأخير L_{max} .

في هذه الدراسة نهدف الى ايجاد الجدول الامثل والتقريبي للمهام j , j = 1,2,… ,n. لتقليل دالة الهدف متعدد المقاييس $\sum_{j=1}^{n} C_j + E_{max} + L_{max}$.

لأستخراج الحل الامثل، قدمنا خوارزمية التقيد والتفرع (BAB) ،هذه الخوارزمية تستخدم قيد أدنى (LB) يعتمد على خاصية تجزئة المسألة متعددة المقاييس. التجارب الحسابية لخوارزمية التقيد والتفرع(BAB) أعطيت على مجموعة كبيرة من مسائل الاختبار.

خوارزمية التقيد والتفرع(BAB) تحتاج وقت أطول للبحث عن الحل الامثل، لذالك نستخدم خوارزميات البحث المحلية للحصول على حلول تقريبية التي تكون قريبة من الحل الامثل وبزمن أقل.

تم تطبيق أثنين من خوارزميات البحث المحلية وهي Descent method (DM) Simulated annealing (SA) and لأيجاد الحلول التقريبية للمسألة.

لغرض تقييم فعالية اساليب الحل ، تم مقارنة خوارزميات (SA), (DM) مع خوارزمية BAB. قدمنا استنتاجات لهذه الخوارزميات بالأعتماد على النتائج حسابية.