Using BAB to find Optimal Solution for Multi-Objective Function with Setup Time

Hanan Ali Cheachen, Ghufran Khalil Joad

Department of Mathematics, College of Sciences, Al-Mustansireyah University, Baghdad, Iraq
Email: ghufrankalil@gmail.com

Abstract. In this paper we studied, the problem of scheduling jobs on a single machine to minimize the multiple objective function and family setup time. This objective function is (total discount completion time and maximum tardiness respectively) which formulated as \( \frac{1}{\sum_{j=1}^{n} \sum_{f=1}^{p} (1 - e^{-\gamma_j}) + T_{\text{max}}} \). for solving this problem, we derived a lower bound to be used in a branch and bound algorithm. We also proposed heuristic method in order to get an upper bound (near optimal solution). The proposed is number of dominance rules are considered to reduce the number of branches in the search tree. The genetic algorithm (GA) is used to obtain one of the upper bounds is used. The computational results are calculated by coding (programming) the algorithms using (MATLAB) and the final results up to (17) product (jobs) in a reasonable time are introduced by tables and added at the end of the research.

1. Introduction
Scheduling problems are encountered in all kinds of systems, since it is necessary to organize or distribute the work between many entities. There are many definitions for the scheduling problems and among these definitions, it can be quoted the following; (Holger. H. Hoos 2005) [9] defined the scheduling problems accordingly next:

"Given the group from j tasks which has at exist processed by the group from M resources, while the object is for locate a schedule which is a chart from jobs to machines and processing times subject to optimization objectives criteria".

In many industries, the decision to manufacture multiple products on common resources results in the need for changeover and setup activities, symbolize costly disruptions to production processes. Therefore, setup reduction is an important feature of the persistent improvement program of any manufacturing, and even service, organization. It is even more critical if an organization expects to respond to changes like shortened lead times, smaller lot sizes, and big quality standards. Every scheduler should understand the basics of setup reduction and be able to recognize the tolerable improvements [14].

Many practical scheduling problems involve sequencing number of jobs divided into several families, by a machine setup time. Most of these problems are NP-hard ((non-deterministic polynomial-time hardness) is the defining property of a class of problems that are informally "at least as hard as the hardest
problems in NP” even without set up times and thus are difficult to solve to optimality. Several scheduling problems in making and advocacy organizations can be formulated as single aperiodic problems at task classes (families), for example, Bruno and Downey[3], Monma and Potts [17] and Chen [4] described a computer system application, a collection of jobs need to be executed on a computer in which every task has a diversion time, a due date, and a request for a compiler to be resident in the memory of the computer. When the appropriate compiler is resident in the computer, then the job will start its processing immediately; otherwise the setup time incurred for bringing the relative compiler into the computer’s memory. So, the setup time reckon only on the time of loading the compiler for the current job but not on the previous job.

In conventional scheduling problems, it is logical and necessary to consider scheduling problems with setup times. In many concrete situations, the setup times are treating either sequence independent or sequence dependent. In the one case, the setup times are ordinarly added to the job processing times however in the two case, the setup time into the job actually being scheduled depends on the beforehand one or ones already scheduled [25].

Group scheduling problems, where each jobs of the self-family must be scheduled jointly, have attracted abundant researchers due to their frequent real-life occurrence. Many manufacturers have implemented the concept of Group Technology (GT) in order to lessen setup costs, lead times, work-in-process inventory costs, and tools handling costs. Group Technology consists of dividing the sum group of jobs into several subsets, called families, where a family is a subset of jobs that have similar requirements in terms of tooling and setups [11].

Setup operations ensure acquisition material, positioning work in process tools, resumption tooling, cleanup, setting the desired jigs and supply, adjusting tools, and test material [22]. In the problem down study we do not treat the Group Technology (GT) supposition; that is, a family of jobs is not necessarily processed as a single batch [13]. Therefore, a family of jobs could be split into multiple nonconsecutive batches in an optimal sequence and all of the batches endure a setup time.

They explained that regard in scheduling problems that patronize setup times (costs) as split up start in the mid-1960s. Scheduling problems together sequence-dependent setup times are oftentimes across in many production regulation and factory applications.

1.1. Literature Reviews:
The objective function is to minimize discounted total completion time. Following the three-field symbols plot offered by Graham et.al [8]. We declare this problem as: $1/s_f \sum_{f=1}^{F} \sum_{j=1}^{n_f} (1 - e^{-r c f})$ where $r \in (0, 1)$ is a discount rate by SPT rule. According to the importance of the scheduling problems, there were a plethora of the specialists whose produced a fine number of the work papers. In the year (1984), Rothkopf and smith [19] work on the sum discounted weighted completion time which is represented accordingly $1/\sum w_j (1-e^{-r c f})$ they generalized the study to cost function and they showed that this problem can exist dissolve optimally as P-kind in polynomial time by the weighted discounted shorter processing time (WDSPT) rule (i.e. scheduling the jobs in order of non-decreasing ratio $w_j e^{-r p_j}/1-e^{-r p_j}$ The problem $F_n/\sum w_j (1-e^{-r c f})$ is considered as an NP-hard problem by Wang et. Al (2006). In (2012), Lin Li et al. (2013) [15] have studied the problem of the sum weighted discounted total completion time and they showed that the heuristics according to the conformable problems wanting education ware. In (2016), studied the problem $1/\sum w_j (1-e^{-r c f}) + r_{max}$, the maximum tardiness $1/T_{max}$ problem is minimized by Earliest Due Date (EDD) rule [10]. The earliest of these woks was done by Sidney [12].

The interesting in the scheduling problems which deal with the separating setup times (costs) began in the middle of the 60s. As Luh et.al (1998), where they declare that “understanding the sequence-dependent setups is yet confirm to be away of being complete, mostly the scheduling to the congress due dates”. So, considering the tardiness is an important matter while the due dates are met. Many authors have worked on
problems in which involving the due dates and sequence subsidiary setup times in order that minimize maximum of tardiness. Baker and Scudder (1990) [2] represented a treatise of penalties (earliness and tardiness) problems while in (2000). Fry et. al. (1987) [6], Garey et. al. (1988) [7], Davis and Kanet (1993) [5] and Yano & Kim (1991) [24] advanced timetabling procedures for finding the optimal beginning times of tasks with inserted idle times. Davis and Kanet (1993) and Kim and Yano (1994) offer a new procedure using branch & bound (B&B) for the problem. Later, Sourd (2005) and Schaller (2007) improved it. In production systems, there occur great accounts of tardiness for several jobs which rise some hardness. As a genuine enforcement, every job in an institution line should be convey at a proven due date. If a task has great tardiness, then the else tasks will not be masterful, resulting in an imponderable in the institution line. So, the term the tardiness ought for approximate to zero. One of the criteria that can satisfy this aim is minimizing the maximum tardiness which is presented as \( T_{\text{max}} \). The aim function value would be equal to zero in the optimal state. If whatever tardiness exist, then the value of the objective function will be larger than zero. That aim function was one worked on by Amin-Nayeri and Moslehi (2000) [1] as a problem of single-machine scheduling case. An algorithm was proposed by Tavakkoli-Moghaddam et al (2005) for obtaining the better potential amount of an idle insert in a sequence, in (2009), Mahnam & Moslehi have worked on single machine problem with morphologic release times for the same objective. As mentioned before, a great number of studies have been accomplished on the scheduling problems which established on the guesstimate of sequence subordinate (independent) setup times, only no study has been convey yet with the same objective.

The remainder of the paper is organized as follows. In Section 2, problem formulation for the problem under study. In Section 3, we explain the heuristic methods used. In Section 4, review the dominance rule. In Section 5, we show the design of instances and report the computational results.

2. Problem Formulation:

Considering a group \( N \) of \( n \) jobs (\( 1 \leq n \leq N \)) that are divided into \( F \) different families, in our paper we consider \( F=2,4,6 \) and 8. Each family \( f \) (\( 1 \leq f \leq F \)) consists of \( n_f \) jobs (where \( \sum_{f=1}^{F} n_f = n \) and it is appropriate to label the jobs as \((1,f),(2,f),...,(n_f,F)\) which are available to be processed on the machine at time zero [16]. The jobs of the self-family may have different processing times \( p_{jf} \) which is the processing time of job \( j \) from the family \( f \), and they can be processed one after one without request any setup times between them. If the machine moves from one family to the other; we would say from the family \( f \) to the family \( g \) then the setup time is required. Every job will have a due date \( d_{jf} \) which is the due date of job \( j \) from the family \( f \). The setup time is sequence – independent (i.e. rely only on the family that is about to start), so that it is denoted by \( sf \). If the first job to be processed in the sequence belongs to family \( f \), then the setup at time zero is \( s_{0f} \) [20]. The objective is to minimize the by the notation of Graham et al. [8] as \( 1/sf/\sum_{f=1}^{F} (1-e^{-rc_{jf}})+T_{\text{max}} \).

Suppose the processing order \( \sigma = (\sigma(1),...,(n)) \), a vector \( (S_{\sigma(1)},...,S_{\sigma(n)}) \) of corresponding setup time is easily constructed. The setup time request immediately before the processing of job \( \sigma(i) = (i = 1,...,n) \) is given by:

\[
S_{\sigma(1)} = \begin{cases} 0 & \text{if } i > 1, \sigma(i-1) \in f \text{ and } \sigma(i) \in g, f \neq g ; f, g \in F. \\
\sigma_{gf} & \text{otherwise}
\end{cases}
\]

Where \( \sigma_{gf} \) is positive integer constant:

\[
\text{Min } Z=\min_{f=1}^{F} \sum_{j=1}^{n_f} (1-e^{-rc_{jf}})+T_{\text{max}}
\]
3. Heuristic Method

It is well famous that the calculation can be miniature by using a heuristic to law as an upper bound on the optimal solution previous to the enforcement of branch and bound method. Since as problem max $1/\sum_{f=1}^{F}\sum_{j=1}^{r_f}(1-e^{-\alpha_{cj}f})+T_{max}$ is NP-hard and subsequently the presence of a polynomial time algorithm for returns an optimal solution is away. Therefore, improve quick heuristic algorithms submission near optimal solution is of great attention. For our problem we proposed heuristic method, the minimum value is used to provide an Upper Bound (UB) [16].

3.1 The Branch and Bound method:

As an exact method, the (BAB) method is used for searching at the optimality or the area that near of it by setting a number of upper bounds as initial solutions to start from then we put the one with the minimum value, and a lower bound to reduce the searching space. the heuristic method is obtained by applying the following:

3.1.1. $(U_1)$: This upper bound is acquired b stratify the shortest processing time (SPT) rule (i.e. separation the jobs within each family in order of $(d_1 \leq d_2 \leq \cdots \leq d_n)$).

3.1.2. $(U_2)$: This upper bound is obtained by applying the earliest due date (EDD) rule (i.e. separation he jobs within each family in order of $(d_1 \leq d_2 \leq \cdots \leq d_n)$).

3.1.3 $(U_3)$: Obtain by Genetic Algorithm (GA):

Genetic algorithms (GA) are a type of search algorithms and an optimization, for returns optimal solutions to computationally intricate problems. The basic ideas for this method were developed by (Holland (1975), Goldberg and Holland (1988)) during their investigations on how to build computing machines that are capable of learning [21].

The genetic algorithm (GA) is founded on principle "survival of the fittest" for Darwin’s (1859), in another meaning it's founded on the basics of genetics and natural selection. So, it is normal the concepts of Genetic Algorithm are directly derived from biological science.

Genetic algorithm:

Step 1: Formation a primary population of (50) chromosomes, we take any solutions arranged by SPT rule and the second is according EDD rule, the three is according bees’ algorithm, the four-weed algorithm while the remain of them are randomly the rest of the solutions are random order.

Step 2: Evaluate the objective (fitness) function for each chromosome and select the better five chromosomes (i.e. the five chromosomes with the minimum values) to generate the new population.

Step 3: Generate the new population by mating (i.e. applying crossover and mutation) each chromosome from step 2 with the whole four initial chromosomes, and every parent chromosome will produce 8 children chromosomes and add 5 chosen solutions, selected from the population so the resulting new population will consist of 50 new chromosomes.

Step 4: If the termination criterions are met, then go to step 5, else go to step 2.

Step 5: End.
In this subsection, we will introduce the main upper bound (UB) of the problem P which it is obtained by:
UB = \min \{U_1, U_2, U_3\}.

3.2 The Lower Bound (LB):
The lower bound for the problem (p) is instituted decomposing (P) of two sub-problems to obtain the a Lower Bound LB for problem (p), where

\[
\begin{align*}
\min F_1 &= \min_{\sigma(j)} \left\{ \left(1 - e^{-r \sigma(j)}\right) \right\} \\
\text{subject to:} & \quad C_{jff} \geq P_{jff} \quad j = 1, \ldots, n_f; f = 1, \ldots, F \\
& \quad 0 < r < 1 \\
& \quad d_{jff} > 0, P_{jff} > 0 \quad j = 1, \ldots, n_f; f = 1, \ldots, F
\end{align*}
\]

\[
\begin{align*}
\min F_2 &= \min_{\sigma(j)} \{T_{\max}(\sigma)\} \\
\text{subject to:} & \quad C_{jff} \geq P_{jff} \quad j = 1, \ldots, n_f; f = 1, \ldots, F \\
& \quad T_{\max} = \max \{0, C_{jff} - d_{jff}\} \quad j = 1, \ldots, n_f; f = 1, \ldots, F \\
& \quad d_{jff} > 0, P_{jff} > 0 \quad j = 1, \ldots, n_f; f = 1, \ldots, F
\end{align*}
\]

Algorithm (LB):
Step 1: In un-scheduling arrange the jobs by using (SPT) rule and adding setup time for the first job from jobs un-sequence if the not least family job of sequence.
Step 2: Computation the amount of cost function of F1 for the problem p1.
Step 3: Re-arrange the jobs by using (EDD) rule and adding setup time for the first job from jobs un-sequence if the not least family job of sequence.
Step 4: Computation the amount of cost F2 for the problem p2.
Step 5: Sum cost the functions (i.e total cost the problem P1 and the problem P2).
Step 6: We repeat the solutions 5 times to get the solutions and compare with the old solutions.
Step 7: Go back to step 2.

Theorem (3.1) [17]: If F1, F2 and F are the minimum rate of P1, P2 and p problems respectively, then we have that F1 + F2 ≤ F.

4. Dominance Rule
Dominance rules generally describe if a node can be discarded before its lower bound is counted. Obviously, dominance rules are especially beneficial when a node can be terminating which has a lower bound that is least than the optimum solution. Some of dominance rules are useful for minimization of the total of discounted completion time and maximum tardiness.

Theorem (3.1): consider two jobs (J_{ff} and J_{kf}) from the same family, the job J_{ff} must precede the job J_{kf} at least in one optimal sequence if P_{ff} ≤ P_{kf}, d_{ff} ≤ d_{kf} and cost (w’) ≥ cost (w)

Where the symbols in which used in this theorem mean as follows:

\[
S \quad S' \\
B \quad s_f \quad P_{ff} \quad P_{kf} \quad W \quad A \\
B \quad s_f \quad P_{kf} \quad P_{ff} \quad W' \quad A
\]
B: The set of jobs in which are scheduled before job j and job k and the jobs before jobs j and k are from different family.

A: The set of jobs in which are scheduled after job j and k and different from another family.

COST (B): The total cost of \( w_j (1 - e^{-r c_j}) + T_{max} \) for jobs in set B.

COST (A): The total cost of \( w_j (1 - e^{-r c_j}) + T_{max} \) for jobs in set A.

W: The group of the jobs in which are scheduled between job j and job k.

COST (W): The total cost of \( (1 - e^{-r c_w}) + T_{max} \) for the jobs of set in sequence S.

COST \( (w') \): The total cost of \( (1 - e^{-r c_w}) + T_{max} \) for the jobs of set \( w' \) in sequence \( S' \).

t: The completion time of the last job in set B.

c: The starting time of the one job in the set A.

proof: Let \( s_1 = \sigma_1 i j \sigma_2 \) and \( s_2 = \sigma_1 j i \sigma_2 \) where \( \sigma_1 \) and \( \sigma_2 \) are disassemble subsequence and supposed to be the completion time of \( \sigma_1 \), we examine variation on the in \( \Delta_i(t) \), where \( \Delta_i(t) = f_i(t) - f_j(t) \) with following:

for \( S_1 = \sigma_1 i j \sigma_2 \)
\[
(1 - e^{-r c_i}) + (1 - e^{-r c_j}) = (1 - e^{-r(T+P_i)}) + (1 - e^{-r(T+P_j)}) \quad \ldots (1)
\]

For \( s_2 \)
\[
(1 - e^{-r c_i}) + (1 - e^{-r c_j}) = (1 - e^{-r(T+P_i)}) + (1 - e^{-r(T+P_j)}) \quad \ldots (2)
\]

For (1) and (2) we get
\[
(1 - e^{-r(T+P_i)}) + (1 - e^{-r(T+P_j)}) - (1 - e^{-r(T+P_i)}) - (1 - e^{-r(T+P_j)}) = e^{-rT} - e^{-r(T+P_i)} + e^{-r(T+P_j)} + e^{-r(T+P_j)}
\]
\[
= e^{-rT} (e^{-P_j} - e^{-0(t+P_j)}) + e^{-P_j} + e^{-P_i} + e^{-P_i}
\]
\[
= e^{-rT} (e^{-P_j} - e^{-P_i}) < 0 \quad \ldots (3)
\]

And for \( S_1 \)
\[
T_{max} = \max \{T, T_i, T_j\}
\]
where \( T_i = \max \{0, c_i - d_i\} = \max \{0, T + P_i - d_i\} \)
\[
T_j = \max \{0, c_j - d_j\} = \max \{0, T + P_j - d_j\}
\]
For schedule \( S' \)
\[
T_{max}' = \max \{T', T_j', T_i'\}
\]
\[
T_i' = \max \{0, c_i - d_i\} = \max \{0, T + P_i - d_i\}
\]
\[
T_j' = \max \{0, c_j - d_j\} = \max \{0, T + P_j - d_j\}
\]
\[
\therefore c_i' \geq d_i', d_j \leq d_j \text{ and } T = T' \text{ then } T_i' \geq T_j', T_j' \geq T_i \text{ and } T_i' \geq T_j
\]
So \( T_{max}' \geq T_{max} \quad \ldots (4) \)

From (3) and (4) we get
\[
i < j \text{ in optimal solution for problem (P).}
\]

5. Experimental Results: In this part, the results are advertising in the table. Table include the comparison results of the (BAB), ILB and UB for the problem.

1- The Problems Instances: The problems were created randomly, and for all job \( j \), where \( j \in \{1, n\} \), with family \( f = \{2, 4, 6, 8\} \) processing times and setup times are randomly created integer from uniform divide m in \([1, 10]\). While the due date \( d_j \) is uniformly generated in the period \([(1 - TF - RDD), (1 - TF + RDD)]\), where \( TP \) is the total processing times of jobs. TF is the tardiness factor, and \( RDD \) is the range prorated of the due dates. TF has values \( \{0.1, 0.2, 0.3, 0.4\} \) and RDD has values \( \{0.8, 1.0, 1.2\} \).

2- Computational results: The calculation results are presented in tables and his table award the results (i.e. optimal values by (BAB), the upper bound of the problem, the initial lower bound and the pursuance time).
For each n, 5 problems and family (2, 4, 6, 8) are examined, and a stopping condition is 1800 seconds as the most the pursuance time. The code which used in the table is:

- $n$: The number of jobs.
- $F$: The number of setup times families.
- $Ex$: The number of examples.
- BAB: The branch and bound method.
- UB: Indicates that upper bound.
- ILB: Indicates that initial lower bound.
- $N$: The number of the nodes.
- Time: The pursuance time.

The Table of Results: In Table, the results of applying (BAB, UB, ILB, the number of the nodes and the execution time) are showed for $n = [5, 17]$ jobs with the family $f = \{2,4,6,8\}$. For each $n$ and there are 5 different examples are tested. These results showed for $n = \{5,6,7,8,9\}$ of using (BAB and UB) are equal, and the stopping condition is 1800 seconds as the maximum execution time. The symbol (/) in some examples denotes on time exceeded the specified time.

| N  | F  | Ex | BAB       | UB       | ILB       | N  | Time  |
|----|----|----|-----------|-----------|-----------|----|-------|
| 5  | 2  | 1  | 18.5606*  | 18.5606*  | 16.4116   | 35 | 0.0985|
|    |    | 2  | 24.3825*  | 24.3825*  | 21.6727   | 32 | 0.0228|
|    |    | 3  | 12.9510*  | 12.9510*  | 9.7290    | 41 | 0.0128|
|    |    | 4  | 24.5334*  | 24.5334*  | 22.1294   | 33 | 0.0047|
|    |    | 5  | 21.9118*  | 21.9118*  | 19.0831   | 33 | 0.0113|
| 4  | 1  | 1  | 42.1817*  | 42.1817*  | 32.9880   | 122| 0.0507|
|    |    | 2  | 27.0042*  | 27.0042*  | 17.1679   | 146| 0.0087|
|    |    | 3  | 26.7009*  | 26.7009*  | 17.4495   | 140| 0.0096|
|    |    | 4  | 13.2115*  | 13.2115*  | 5.1824    | 27 | 0.0032|
|    |    | 5  | 28.9000*  | 28.9000*  | 19.1987   | 116| 0.0072|
|    |    | 1  | 35.1482*  | 35.1482*  | 20.5844   | 164| 0.0093|
|    |    | 2  | 36.2341*  | 36.2341*  | 21.7908   | 164| 0.0117|
|    |    | 3  | 18.8498*  | 18.8498*  | 8.1347    | 48 | 0.0033|
|    |    | 4  | 42.6685*  | 42.6685*  | 28.3811   | 194| 0.0094|
|    |    | 5  | 25.8115*  | 25.8115*  | 11.2327   | 98 | 0.0054|
| 6  | 1  | 1  | 26.9330*  | 26.9330*  | 11.9709   | 170| 0.0084|
|    |    | 2  | 30.6073*  | 30.6073*  | 16.0076   | 134| 0.0069|
|    |    | 3  | 31.8215*  | 31.8215*  | 17.1607   | 128| 0.0066|
|    |    | 4  | 42.0722*  | 42.0722*  | 27.6636   | 170| 0.0082|
|    |    | 5  | 39.1439*  | 39.1439*  | 24.5159   | 206| 0.0096|
| 8  | 1  | 1  | 21.1585*  | 21.1585*  | 18.4053   | 148| 0.0077|
|    |    | 2  | 9.1939*   | 9.1939*   | 6.5942    | 39 | 0.0031|
|    |    | 3  | 17.2970*  | 17.2970*  | 14.7113   | 52 | 0.0038|
|    |    | 4  | 27.4871*  | 27.4871*  | 25.0249   | 67 | 0.0042|
|    |    | 5  | 25.2247*  | 25.2247*  | 22.6674   | 80 | 0.0045|
|    |    | 1  | 32.5588*  | 32.5588*  | 22.4516   | 353| 0.0153|
|   |   |   |   |   |   |
|---|---|---|---|---|---|
|   |   |   |   |   |   |
| 6 |   | 38.5210* | 38.5210* | 28.9720 | 399 | 0.0165 |
|   |   | 44.9892* | 44.9892* | 35.6464 | 473 | 0.0197 |
|   | 4 | 28.9647* | 28.9647* | 19.4253 | 171 | 0.0081 |
|   | 5 | 30.7474* | 30.7474* | 20.4030 | 415 | 0.0182 |
| 6 |   | 27.6416* | 27.6416* | 8.6658 | 276 | 0.0119 |
|   | 2 | 29.3259* | 29.3259* | 10.8443 | 227 | 0.0101 |
|   | 3 | 37.7544* | 37.7544* | 20.3164 | 331 | 0.0147 |
|   | 4 | 43.5263* | 43.5263* | 22.7986 | 703 | 0.0282 |
|   | 5 | 36.9614* | 36.9614* | 15.7632 | 655 | 0.0268 |
| 8 |   | 44.6377* | 44.6377* | 24.1822 | 797 | 0.0316 |
|   | 2 | 47.8967* | 47.8967* | 26.9530 | 997 | 0.0416 |
|   | 3 | 40.8785* | 40.8785* | 20.4764 | 509 | 0.0206 |
|   | 4 | 43.6828* | 43.6828* | 22.7105 | 1093 | 0.0431 |
|   | 5 | 33.4795* | 33.4795* | 12.5409 | 975 | 0.0388 |
| 7 |   | 25.9323* | 25.9323* | 22.6103 | 323 | 0.0139 |
|   | 2 | 24.9601* | 24.9601* | 21.9569 | 354 | 0.0153 |
|   | 3 | 41.2338* | 41.2338* | 38.7519 | 277 | 0.0120 |
|   | 4 | 44.1869* | 44.1869* | 41.6218 | 500 | 0.0204 |
|   | 5 | 16.2182* | 16.2182* | 12.9773 | 377 | 0.0159 |
| 6 |   | 39.9456* | 39.9456* | 30.4067 | 655 | 0.0256 |
|   | 2 | 47.6493* | 47.6493* | 37.9050 | 1345 | 0.0511 |
|   | 3 | 44.1861* | 44.1861* | 34.5721 | 1123 | 0.0429 |
|   | 4 | 34.5836* | 34.5836* | 24.6628 | 380 | 0.0156 |
|   | 5 | 34.8067* | 34.8067* | 25.2638 | 1642 | 0.0634 |
| 8 |   | 54.0482* | 54.0482* | 26.3170 | 4560 | 0.1786 |
|   | 2 | 74.0625* | 74.0625* | 46.5566 | 7940 | 0.3086 |
|   | 3 | 56.9049* | 56.9049* | 29.0227 | 7388 | 0.2873 |
|   | 4 | 45.6390* | 45.6390* | 17.0008 | 3944 | 0.1526 |
|   | 5 | 67.9602* | 67.9602* | 40.1515 | 7940 | 0.3101 |
| 7 |   | 54.0482* | 54.0482* | 26.3170 | 4560 | 0.1786 |
|   | 2 | 74.0625* | 74.0625* | 46.5566 | 7940 | 0.3086 |
|   | 3 | 56.9049* | 56.9049* | 29.0227 | 7388 | 0.2873 |
|   | 4 | 45.6390* | 45.6390* | 17.0008 | 3944 | 0.1526 |
|   | 5 | 67.9602* | 67.9602* | 40.1515 | 7940 | 0.3101 |
| 8 |   | 39.3047* | 39.3047* | 36.2483 | 1035 | 0.1026 |
|   | 2 | 32.9801* | 32.9801* | 30.5515 | 161 | 0.0080 |
|   | 3 | 30.1949* | 30.1949* | 27.6777 | 193 | 0.0173 |
|   | 4 | 32.4527* | 32.4527* | 28.1092 | 2629 | 0.0999 |
|   | 5 | 23.5777* | 23.5777* | 18.1505 | 682 | 0.0260 |
| 4 |   | 63.6788* | 63.6788* | 54.0788 | 2831 | 0.0814 |
|   | 2 | 43.1333* | 43.1333* | 33.1364 | 3616 | 0.1246 |
|   | 3 | 52.5488* | 52.5488* | 42.7529 | 1625 | 0.0495 |
|   | 4 | 44.4976* | 44.4976* | 34.4244 | 1629 | 0.0477 |
|   |   |   |   |   |   |
|---|---|---|---|---|---|
|   | 8 | 6 | 5 | 39.7246* 39.7246* 29.4441 1592 0.0485 |
| 1 | 58.3657* 58.3657* 37.8256 18661 0.5651 |
| 2 | 60.3769* 60.3769* 40.0932 20688 0.6345 |
| 3 | 48.8810* 48.8810* 27.9300 18291 0.5503 |
| 4 | 61.0708* 61.0708* 39.3523 41328 1.2641 |
| 5 | 71.8359* 71.8359* 51.2241 39521 1.2040 |
|   | 8 |   | 1 | 76.0324* 76.0324* 40.2616 30211 0.9324 |
| 2 | 79.0159* 79.0159* 42.6754 50921 1.6005 |
| 3 | 63.2340* 63.2340* 27.3275 37841 1.1738 |
| 4 | 53.3537* 53.3537* 16.8087 27994 0.8522 |
| 5 | 63.7790* 63.7790* 27.6042 28343 0.8429 |
|   | 9 |   | 2 | 35.5739* 35.5739* 33.3521 1039 0.0399 |
| 2 | 43.9143* 43.9143* 41.1679 640 0.0257 |
| 3 | 43.5595* 43.5595* 40.9257 650 0.0261 |
| 4 | 39.0451* 39.0451* 36.3061 398 0.0163 |
| 5 | 42.4435* 42.4435* 39.1557 6647 0.2566 |
|   | 6 |   | 1 | 69.6771* 69.6771* 49.5267 66718 2.5635 |
| 2 | 63.2888* 63.2888* 41.7853 13679 5.2996 |
| 3 | 71.0934* 71.0934* 50.4783 166932 6.5950 |
| 4 | 61.8668* 61.8668* 40.7659 64116 2.4804 |
| 5 | 53.9612* 53.9612* 33.6253 20382 0.7663 |
|   | 8 |   | 1 | 73.1080* 73.1080* 36.5982 427986 17.9807 |
| 2 | 82.3050* 82.3050* 46.9104 206138 8.0750 |
| 3 | 51.3196* 51.3196* 14.8969 104848 4.0594 |
| 4 | 62.9748* 62.9748* 27.4156 205247 8.2472 |
| 5 | 61.7406* 61.7406* 25.2577 124853 4.8405 |
|   | 10 | 4 | 1 | 32.1529 32.6858 28.2019 26546 1.2040 |
| 2 | 49.1367* 49.1367* 46.6967 625 0.0290 |
| 3 | 59.9532* 59.9532* 55.6443 33608 1.2992 |
| 4 | 36.4785* 36.4785* 33.8726 1454 0.0648 |
| 5 | 70.7035* 70.7035* 68.4968 1667 0.0746 |
|   | 4 |   | 1 | 65.3664* 65.3664* 55.9742 89437 3.7450 |
| 2 | 45.1994* 45.1994* 33.6098 24784 0.9298 |
| 3 | 57.3485 58.4518 47.4566 22350 0.83637 |
| 4 | 51.4936 52.1497 41.3122 14961 0.5623 |
| 5 | 54.0759 55.3678 43.7756 57626 2.1895 |
|   | 6 |   | 1 | 70.3132* 70.3132* 49.0888 65733 2.4590 |
| 2 | 69.8474* 69.8474* 53.2522 40111 1.5037 |
|   | 3 | 45.3040* | 45.3040* | 25.7807 | 19711 | 0.8183 |
|---|---|----------|----------|---------|-------|--------|
|   | 4 | 56.4069  | 57.5568  | 35.4515 | 237702| 10.3467|
|   | 5 | 78.3835  | 78.7416  | 57.5741 | 1051895| 42.8273|
| 8 | 1 | 87.1547* | 87.1547* | 51.1975 | 2835873| 116.2090|
|   | 2 | 79.0112* | 79.0112* | 43.1233 | 1372528| 52.9967 |
|   | 3 | 102.1067*| 102.1067*| 66.3797 | 3652045| 114.3312|
|   | 4 | 70.2734  | 70.4697  | 36.1511 | 1099925| 34.6518 |
|   | 5 | 83.6773* | 83.6773* | 46.8479 | 2081456| 64.7658 |
| 11| 2 | 72.7247  | 72.7784  | 69.6376 | 54330  | 2.0814 |
|   | 2 | 39.3959* | 39.3959* | 34.2215 | 127133 | 3.9015 |
|   | 3 | 38.9179* | 38.9179* | 34.5143 | 115629 | 3.7752 |
|   | 4 | 49.4398* | 49.4398* | 47.1363 | 6208   | 0.1749 |
|   | 5 | 48.0935* | 48.0935* | 44.8642 | 42993  | 1.3023 |
| 6 | 1 | 70.4671* | 70.4671* | 48.1267 | 2325512| 71.9499|
|   | 2 | 75.6740* | 75.6740* | 50.6689 | 491939 | 15.4453|
|   | 3 | 72.2324* | 72.2324* | 47.8844 | 1980559| 60.3902|
|   | 4 | 51.0511  | 51.4002  | 25.3716 | 116135 | 10.2895|
|   | 5 | 37.2267* | 37.2267* | 37.9900 | 335154 | 10.2895|
| 8 | 1 | 97.4213* | 97.4213* | 61.7241 | 24932007| 790.5983 |
|   | 2 | 89.9661  | 89.9986  | 54.7351 | 7156930| 221.6080|
|   | 3 | 70.7659* | 70.7659* | 37.7120 | 308297 | 8.7755  |
|   | 4 | 81.0335* | 81.0353* | 44.6617 | 10707313| 332.0386|
|   | 5 | 88.1852  | 88.2298  | 52.4954 | 20100367| 629.5126|
| 12| 2 | 69.7388* | 69.7388* | 66.9561 | 4790   | 0.1333 |
|   | 2 | 45.0493* | 45.0493* | 42.8396 | 4731   | 0.1636 |
|   | 3 | 73.0304* | 73.0304* | 70.3810 | 19182  | 0.5517 |
|   | 4 | 57.4469* | 57.4469* | 54.9613 | 3241   | 0.0923 |
|   | 5 | 64.2999  | 66.0278  | 61.5916 | 516546 | 16.9135|
| 4 | 1 | 83.7581  | 85.7997  | 74.3406 | 284430 | 8.4155 |
|   | 2 | 68.5505  | 68.7074  | 56.1298 | 714645 | 21.4935|
|   | 3 | 53.4514  | 54.7839  | 43.3238 | 80309  | 2.6660 |
|   | 4 | 77.7415  | 77.8793  | 67.0835 | 384637 | 12.1894|
|   | 5 | 57.7770  | 57.7853  | 45.5842 | 1611786| 49.0465|
| 6 | 1 | 75.8596  | 78.0521  | 53.6569 | 2183638| 66.1950 |
|   | 2 | 96.0855  | 96.4472  | 75.6365 | 3251373| 96.9198 |
|   | 3 | 81.6870  | 82.8484  | 58.5520 | 4589232| 143.1084|
|   | 4 | 80.5243  | 83.2103  | 57.5970 | 3537741| 107.3939|
|   | 5 | 83.6811  | 83.8422  | 56.6534 | 16530539| 513.1630|
| Page | Column 1 | Column 2 | Column 3 | Column 4 | Column 5 | Column 6 |
|------|----------|----------|----------|----------|----------|----------|
| 8    | /        | /        | /        | /        | 1800.0002|
| 2    | /        | /        | /        | /        | 1800.0002|
| 3    | 81.8674  | 81.8756  | 45.1758  | 29915490 | 928.4289 |
| 4    | 74.8196  | 74.9842  | 39.3528  | 21910849 | 679.1056 |
| 5    | /        | /        | /        | /        | 1800.0001|
| 13   | 2        | 61.3626* | 61.3626* | 59.1709  | 27046    | 0.7790   |
| 2    | 80.9849* | 80.9849* | 78.9649  | 38836    | 12.446   |
| 3    | 64.5674* | 64.5674* | 61.4131  | 139847   | 4.3214   |
| 4    | 54.1981* | 54.1981* | 50.1627  | 681122   | 23.1814  |
| 5    | 78.0532* | 78.0532* | 75.8878  | 31634    | 1.0388   |
| 4    | 1        | 69.5527  | 70.1975  | 59.2890  | 939319   | 27.5206  |
| 2    | 66.0666  | 66.2078  | 55.2306  | 1121060  | 32.2618  |
| 3    | 80.2413  | 80.3736  | 70.1658  | 1031227  | 29.8601  |
| 4    | 73.9158* | 73.9158* | 60.4966  | 8852525  | 271.8963 |
| 5    | 71.9607  | 73.9708  | 57.0006  | 3246256  | 100.8326 |
| 6    | 1        | 75.3635* | 75.3635* | 53.8647  | 24181181 | 742.1095 |
| 2    | 77.6531  | 79.1087  | 51.1334  | 29364516 | 905.5780 |
| 3    | 106.0635 | 106.6233 | 85.1235  | 36294738 | 1100.5083|
| 4    | 82.5116  | 82.9162  | 51.1359  | 12053771 | 367.0831 |
| 5    | 43.4232* | 43.4232* | 38.3040  | 10440    | 0.3046   |
| 14   | 2        | 43.0003  | 46.0277  | 39.5110  | 374868   | 11.0523  |
| 2    | 56.7608  | 57.2849  | 51.9202  | 4749284  | 146.1543 |
| 3    | 61.8356* | 61.8356* | 58.4435  | 17923712 | 577.6754 |
| 4    | 43.4232* | 43.4232* | 38.3040  | 10440    | 0.3046   |
| 5    | 72.4318  | 72.8279  | 69.4523  | 5680774  | 176.8153 |
| 6    | 1        | 79.1435  | 79.2923  | 69.2907  | 4782722  | 147.6929 |
| 2    | 92.7653  | 92.9048  | 82.7127  | 2996032  | 92.7637  |
| 3    | /        | /        | /        | /        | 1800.0001|
| 4    | 57.1401  | 58.3414  | 44.4973  | 8551879  | 261.1641 |
| 5    | 67.3696  | 68.2621  | 55.1997  | 4560399  | 137.2345 |
| 15   | 2        | 54.6286  | 54.3219  | 50.7152  | 22140644 | 684.8232 |
| 2    | 60.5660  | 63.2054  | 56.4984  | 135263   | 3.9229   |
| 3    | /        | /        | /        | /        | 1800.0001|
| 4    | 63.0105  | 63.1903  | 58.1535  | 37516308 | 1196.1326|
| 5    | 56.5249  | 59.2365  | 52.5119  | 2069623  | 60.8069  |
| 4    | 1        | 87.4431  | 87.4490  | 77.2440  | 24280207 | 760.6983 |
| 2    | /        | /        | /        | /        | 1800.0001|
| 3    | 63.6930  | 63.9398  | 52.6858  | 6218487  | 184.9003 |
6. Conclusions: -
1- By using branch and bound method done problem solution.
2- Get three the upper and one lower bound.
3- The upper bound using the genetic hybridization algorithm its results were equal to the optimal solution in part of the results. except in case of job (n=14,15,16,17) where there are some examples failed because they take a long executing time (i.e. more than 1800 seconds)

References:
[1] M R Amin-Nayeri and G Moslehi 2000 Optimal algorithm for single machine sequencing to minimize early/tardy cost *J Esteghlal. Isfahan University of Technology* 1 pp 35-48
[2] R Baker and G D Scudder 1990 Sequencing with earliness and tardiness penalties: A review *J OPI Res* 38 pp 22–36
[3] J Bruno and P Downey 1978 Complexity of task sequencing with deadlines, setup times and changeover costs *SIAM J Computer* 7 pp 393-404
[4] B Chen B 1993 A better heuristic for preemptive parallel machine scheduling with batch setup times *SIAM J Computer* 22 pp1303-1318
[5] J S Davis and J J Kanet 1993 Single-machine scheduling with early and tardy completion costs *Nav Res Logist* 40 pp 85-101 tardiness. *Math Opens Res* 13: 330–348.
[6] T D Fry 1987 Armstrong RD and Blackstone JH. Minimizing weighted absolute deviation in single machine scheduling *IIE Trans* 9 pp 445-450
[7] M R Garey, R E Trajan and G T Wilfong 1988 One-processor scheduling with symmetric earliness and tardiness *Math Opens Res* 13 pp 330-348
[8] R L Graham, Lawler, J k Leenstra and K Ringo A H G 1979 Optimization and Approximation in Deterministic Sequence and Schedule *A Survey Annals of Discrete Mathematics* 5
[9] H H Holger and S Thomas 2005 Germany Stochastic Local Search *Foundations and Applications* ISBN:1-55860-872-9. Elsevier Incorporation

| 4 | 58.4665 | 60.7093 | 46.9668 | 3061590 | 90.8379 |
| 5 | 67.5888 | 68.5348 | 55.9355 | 52819043 | 1603.3751 |
| 16 | 2 | 1 | 53.0390* | 53.0390* | 49.0552 | 1409289 | 3061590 | 90.8379 |
| 2 | 69.7273 | 72.6159 | 67.4610 | 1199218 | 39.5166 |
| 3 | 118.3106* | 118.3106* | 115.7161 | 47436616 | 1431.6475 |
| 4 | 70.4269 | 73.1534 | 67.5838 | 1270041 | 39.5166 |
| 5 | / | / | / | 1800.0001 |
| 4 | 1 | 67.0088 | 69.7353 | 52.2008 | 44184610 | 1356.1336 |
| 2 | / | / | / | 1800.0003 |
| 3 | / | / | / | 1800.0001 |
| 4 | / | / | / | 1800.0001 |
| 5 | 74.4973 | 75.4270 | 58.5070 | 47436616 | 1431.6475 |

| 17 | 2 | 1 | 107.5272 | 109.8345 | 104.3674 | 28490250 | 936.4970 |
| 2 | 68.7261 | 70.3712 | 63.7914 | 259586 | 7.9549 |
| 3 | / | / | / | 1800.0002 |
| 4 | / | / | / | 1800.0003 |
| 5 | 56.9582 | 59.8469 | 52.4789 | 31334657 | 1013.1670 |
[10] J R Jackson 1955 Scheduling a production line to minimize maximum tardiness Research Report 43, Management Science Research Project, University of California, Los Angeles, USA

[11] N D JATINDER, Y GUPTA and SAMARN CHANTARAVARAPAN 2008 Single machine group scheduling with family setups to minimize total tardiness 46 (6) 15

[12] J B Sidney 1977 Optimal single-machine scheduling with earliness and tardiness penalties Operations Research 25 (1) pp 62-69

[13] J B Wang and J J Wang 2014 Single machine group scheduling with time dependent processing times and ready times Information Sciences 275 pp 226-231

[14] J R Montoya-Torres, Ferrari and Gonzalez-Solano 2010 Production scheduling with sequence dependent setup and job release time Nor 163 pp 260-269

[15] L Lin, L, Y Sheng-Wu, W Yu-Bin, H Yunshan and J Ping 2013 Single machine scheduling jobs with a truncated sum-of processing-times-based learning effect Int J Adv Manu Technol 67 pp 261-267

[16] H I Manal 2011 Scheduling Job Families with Setups on A single Machine 2011 Al-Mustansiriyah University 22 (3)

[17] C L Monma and C N Potts 1989 On the complexity of scheduling with batch setup times Opera. Res. 37 pp 798-804

[18] A m Ramadhan 1998 Single machine scheduling using branch and bound techniques M.Sc. Thesis University of Al-Mustansiriyah College of Science Department of Mathematics

[19] W E Smith 1956 Various optimizer for single stages production Naval Research Logistics Quarterly 3 pp 59-66

[20] K Steinhöfel, A Albrecht and C K Wong 2003 An experimental analysis of local minima to improve neighbourhood search Computers and Operations Research 30 pp 2157-2173

[21] R J M Vaessens, E H L Aarts and J K lenstra 1998 A Local Search Template Computers and Operations Research 25 pp 969-979

[22] V J Vinicius and E C A José 2016 ILS Heuristics for the Single-Machine Scheduling Problem with Sequence-Dependent Family Setup Times to Minimize Total Tardiness 36570-900

[23] J P Walser 1999 Integer optimization by local search: a domain-independent approach Springer-Verlag

[24] C Yano and Y Kim 1991 Algorithms for a class of single machine weighted tardiness and earliness problems Eur J Opf Res 52 pp 161–178

[25] Y Yuling, L Chinyao and L Wen-Yi 2015 Single Machine Scheduling with Time-Dependent Learning Effect and Non-Linear Past-Sequence-Dependent Setup Times 3 pp 10-15