

# FIXED SEQUENCE JOB SHOP SCHEDULING USING THE ANALYTIC HIERARCHY PROCESS

David Hauser

Joseph M. Katz Graduate School of Business, 247 Mervis Hall  
University of Pittsburgh, Pittsburgh, PA 15260, U.S.A.

**Abstract** -- When scheduling a set of jobs for a two machine fixed-sequence process, one has various ways for minimizing the makespan. Johnson's algorithm has been shown to be one of the better ways. Complete enumeration or versions of trial and error are costly in time, accuracy, and computer memory. Its disadvantages include an inability to perform sensitivity analysis, the dependency on the jobs themselves, and its limitation to two machines. Most scheduling heuristics center around the minimization of idle-time. This paper discusses a completely different approach by developing an Analytic Hierarchy Process (AHP) model for two machines and several heuristics for  $m > 2$  machines.

**Keywords:** Analytic Hierarchy Process, job shop scheduling, Johnson's algorithm

## 1. Introduction

Scheduling is ordering or prioritization in time. The Analytic Hierarchy Process (AHP) is a fundamental methodology of prioritization [8-10] but so far, it has not been used methodically to set priorities of tasks with respect to time taking into consideration structural requirements. To that end, we set the task to apply the AHP in time scheduling.

Johnson's [5] algorithm finds an optimal sequence of jobs for a two-machine fixed sequence job shop which minimizes makespan. The simplicity of his method encouraged others to generalize his ideas for  $m > 2$  machines, yet without success. Many combinatorial approaches have been presented for solving the problem [1,4,11,12] yet all suffer from the exponential time requirement characteristic of this problem. Since the general scheduling problem is NP-complete, and all NP-complete problems cannot be solved with efficient algorithms, the research focus has been placed on heuristic development [3].

Palmer [7] suggested that higher priority should be assigned to jobs that tend to increase processing time as they move from one machine to the next. Campbell et al. [2] proposed to generate a set of  $m-1$  two-machine problems from the original  $m$ -machine problem, apply the two-machine algorithm of Johnson to each of the  $m-1$  sub-problems, and then employ the schedules which resulted in the lowest overall makespan. Nawaz et al. [6] proposed that a job with larger total processing time should have higher priority in the sequence and have shown that this proposal outperforms the heuristics proposed by Campbell et al.

Given that the general job shop scheduling problem is NP-complete, the best algorithms require exponential time and all of the polynomial time heuristics encounter a decrease in efficiency as the number of jobs and the number of machines grow. However consider the potential power of the AHP. The AHP is a tool which attributes priorities to various elements of a problem. The absolute measurements associated with various aspects of a problem are either converted to  $[0,1]$  ratio scales for the purpose of developing priorities or to express dominance of one element over another. As such, employing the AHP in the job shop scheduling problem is a unique approach. However in order to generalize to the  $m > 2$  case, we will use a part of the strategy central to the Campbell et al. heuristic.

It has been proven that Johnson's algorithm<sup>1</sup> provides an order of jobs that minimizes total makespan if all of the jobs to which the algorithm is applied require two different machines in a fixed order of operation [5]. This same objective can also be met by proper application of the AHP. Consider a two-machine job shop in which all  $m$  jobs first require machine A then later require machine B. From the list of job times for each machine, we construct one list  $R = \{r_i\}$ , to be defined more carefully in a later section, where  $r_i$  is the  $i^{\text{th}}$  element of this list for all  $i$  satisfying  $i=0, \dots, n$ . This list will become the intensity ratings for an AHP model.

Since the AHP is a prioritization tool, each of the jobs to which this model is applied will be prioritized based on characteristics of the two machines and on the times required by the jobs. We will show that the magnitude of the times required are not relevant to the schedule, rather the prioritization of the jobs will be a function of the rank order of the required processing times. Those jobs with the larger synthesized priorities will be scheduled earlier than the jobs with the smaller synthesized priorities. It will also be shown that the rank order of jobs is a schedule with the minimum makespan in the  $m=2$  machines case and provides the near minimum makespan in the  $m > 2$  machines case.

## 2. Development of the Hierarchy

To construct an AHP model, one must first decompose the problem into a goal, a set of criteria, and a list of alternatives [8-10]. The criteria which govern the jobs are the two machines involved in the process. Therefore the upper echelon of the hierarchy is determined by the nature of the problem. Here are the steps considered in the development of the lowest level.

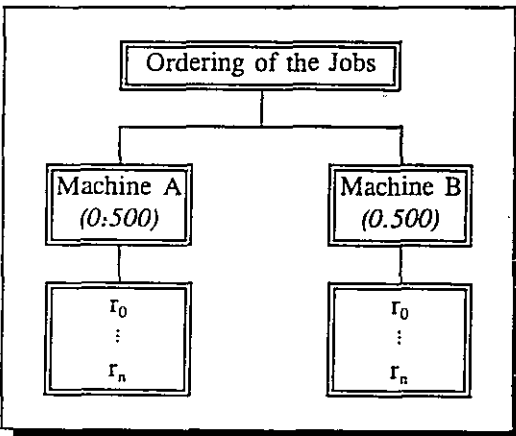


Figure 1 - The AHP Model of Johnson's Algorithm.

the right. The list of all job times for the first machine is  $\{2, 8, 1, 4, 3, 2\}$ . We append to this list the complete set of job times required on the second machine. After we remove the duplicates, we have  $\{2, 8, 1, 4, 3, 5, 9\}$  which is sorted to read  $\{1, 2, 3, 4, 5, 8, 9\}$ . Finally we append a time

First one constructs the list of processing times across all jobs for the first machine. Next, one appends to this set the list of processing times across all jobs for the second machine, and then removes all duplicate elements. Next one sorts the list in ascending order, and then appends a hypothetical job time of  $\infty$ ; this final list will be the intensity ratings for an AHP model. Let  $R = \{r_i\}$  be this ratings list, where  $r_i$  is the  $i^{\text{th}}$  rating of the list for all  $i$  satisfying  $i=0, \dots, n$ , where  $n$  is the index of the element  $\infty \in R$ . See Figure 1.

For an illustration, consider the problem defined by the data on

Job	Processing Times	
	Machine A	Machine B
a	2	3
b	8	2
c	1	4
d	4	5
e	3	9
f	2	2

<sup>1</sup> Consider a two-machine fixed-sequence job shop. If the minimum time for a job appears on the first machine, the corresponding job is scheduled as early as possible else this job is scheduled as late as possible [5].

of  $\infty$  to this sorted result, and thus  $R = \{1, 2, 3, 4, 5, 8, 9, \infty\}$ , where  $i=0$  is the smallest index of this list and  $i=n=7$  is the largest.

### 3. Validity of the Model

The absolute mode of weighting the alternatives in the AHP is the manner in which the weights of each processing time are transferred to the jobs. This approach permits a large number of jobs to be easily scheduled [8]. In the absolute mode, each alternative is measured independently of the others [8-10]. Hence application of the absolute mode to the jobs will guide the schedule as it is based solely on the decreasing order of the priorities of the jobs.

**Definitions:** Let  $R = \{r_i\}$  be the ascending-order non-repetitive list of intensity ratings of all jobs between both machines of a two-machine fixed-sequence process, where  $r_i$  is the  $i^{\text{th}}$  intensity rating for all  $i$  satisfying  $i=0, \dots, n$ , where  $n$  is the index of the element  $\infty \in R$  ( $n = |R| - 1$ ). Let  $A_i = v^i$ ,  $i=0, \dots, n$ ,  $B_i = 1 - v^i$ ,  $i=0, \dots, n-1$ , and  $B_n = 1$ , where  $A_i$  and  $B_i$  are the idealized priorities of  $r_i \in R$   $i=0, \dots, n$  for machines A and B, respectively.

**Theorem:** Given  $R$ , for any  $v \in (0, \frac{1}{2}]$ , equally weighted application of  $A_i$  and  $B_i$  to all jobs of a fixed-order two-machine process yields a rank (schedule) of jobs with minimum makespan.

**Proof:** Johnson's algorithm has been proven to find a job schedule with the minimum makespan for a fixed-sequence two-machine process [5]. By definition of  $R$  ( $r_i < r_{i+1}$ ), we conclude that  $A_i > A_{i+1}$  and  $B_i < B_{i+1}$   $i=0, \dots, n-1$ , which are respectively equivalent to  $v^i > v^{i+1}$  and  $1 - v^i < 1 - v^{i+1}$ ,  $i=0, \dots, n$ , which holds for  $0 < v < 1$ .

Johnson's algorithm examines the times required by each job on both of the machines. Hence the machines act as the criteria governing the objective and determining the job placement within the schedule [5]. Since both machines are regarded as equally important, each must be prioritized equally and thus the priorities of both machines A and B are equal to 0.500.

Any job not yet scheduled cannot interfere with the schedule of jobs already situated, since the algorithm is based on the ascending order of processing times (once a job has been placed at the head of the schedule, all jobs considered thereafter must have times greater than or equal to those already placed). The minimum requirement a job can have of machine A is  $r_0$ . If job  $\alpha$  requires  $r_0$  time on machine A, then job  $\alpha$  must have the largest synthesized priority over *all* other jobs regardless of the additional priority job  $\alpha$  is to acquire from its machine B rating. Since job  $\alpha$ , by definition, must have the greatest synthesized priority, the machine B time of job  $\alpha$  may be the smallest possible value,  $r_1$ , (which corresponds to the smallest possible priority) without altering the assumption that the minimum occurred on machine A. Suppose job  $\beta$  has the next highest priority. The smallest time job  $\beta$  may require without interfering with the schedule of job  $\alpha$  is  $r_1$  for machine A. Since job  $\beta$  cannot have a priority greater than that of job  $\alpha$ , the greatest possible priority of job  $\beta$  must be less than the smallest priority of job  $\alpha$ . Job  $\beta$  could have a requirement of  $r_n$  on B (which corresponds to the greatest priority of all machine B ratings). Thus we know that

$$A_0 + B_1 \geq A_1 + B_n \quad (1)$$

Since the same argument holds (but in reverse fashion) for the jobs placed latest in the schedule,

we know that jobs with the minimum times found on machine B must have priorities lower than jobs considered for scheduling thereafter. Hence we can derive the relation:

$$B_0 + A_1 \leq B_1 + A_n \quad (2)$$

In general, once the  $i^{\text{th}}$  rating has been applied to a job, all jobs considered thereafter have ratings greater than  $r_i$ . Hence constraints (1) and (2) can be respectively generalized as follows:

$$A_i + B_{i+1} \geq A_{i+1} + B_n \quad (3a)$$

$$B_i + A_{i+1} \leq B_{i+1} + A_n \quad (3b)$$

Substituting for  $A_i$  and  $B_i$  into constraint (3a), we have

$$v^i + (1-v^{i+1}) \geq v^{i+1} + (1-v^n) \quad (4)$$

which simplifies to

$$v^i - 2v^{i+1} \geq -v^n \quad (5)$$

and to

$$v^i(2v-1) \leq v^n \quad (6)$$

Since  $v$  is positive, both the right side of (6) and the  $v^i$  term of the left side of (6) are positive. For certain values of  $v$ , the  $(2v-1)$  term of the left side of (6) is negative hence guaranteeing (6) to hold. Therefore consider the following sub-problem derived from (6):

$$2v-1 \leq 0 \quad (7)$$

which holds when

$$v \leq \frac{1}{2} \quad (8)$$

thus the parameter  $v$  as constrained by (3a) and (8) must satisfy  $0 < v \leq \frac{1}{2}$ . To fully justify the definitions of  $A_i$  and  $B_i$ , we must also show that these definitions hold for constraint (3b). Furthermore, constraint (8) which was derived from (3a) must also satisfy (3b). As before, we substitute for  $A_i$  and  $B_i$  into constraint (3b), and obtain

$$(1-v^i) + v^{i+1} \leq (1-v^{i+1}) + v^n \quad (9)$$

which simplifies to

$$-v^i + 2v^{i+1} \leq v^n \quad (10)$$

and to

$$v^i(2v-1) \leq v^n \quad (11)$$

Since  $v$  is positive, both the right side of (11) and the  $-v^i$  term of the left side of (11) are positive. For certain values of  $v$ , the  $(2v-1)$  term of the left side of (11) is negative hence guaranteeing (11) to hold. Thus consider the following sub-problem derived from (11):

$$2v-1 \leq 0 \quad (12)$$

which holds when

$$v \leq \frac{1}{2} \quad (13)$$

For values of  $v$  satisfying  $0 < v \leq \frac{1}{2}$ , relations (4) and (9) are simultaneously satisfied. To disprove the theorem, one must show that there is no value for  $v$  which simultaneously satisfies  $A_i$  and  $B_i$ . Yet we have just shown that  $v$  is in fact constrained and of the reals, and thus the theorem is proven.

#### 4. An Illustration ( $m=2$ )

Suppose a two-machine fixed-sequence job shop is considering the arrangement of 5 different jobs each of which has the technological requirement that they first be processed on machine A and then later processed on machine B. The times required by these jobs are listed in Table I. The ordered list of times between all jobs across both machines are  $\{1,2,5,6\}$ , hence  $r_0 = 1$ ,  $r_1 = 2$ ,  $r_2 = 5$ , and  $r_3 = 6$ . Next, we append a time greater than 6 to this list; we now have  $\{1,2,5,6,\infty\}$ , and thus  $n=4$ . This new list will be applied to the AHP model as intensity ratings. The completed hierarchy for this example can be seen in Figure 2.

The next step is the assignment of priorities of the rating intensities. The priorities are not a function of the values of the intensities. Rather, they are a function of their indices which are based on their orders. Hence, we define the priorities of the rating intensities of the machines based on the equations for  $A_i$  and  $B_i$  defined in terms of the parameter  $v$ , where  $v$  is arbitrarily chosen within the range 0 to  $\frac{1}{2}$  excluding the

Table I - Job Time Requirements.

Job	Machine A	Machine B
a	2	6
b	5	6
c	1	2
d	2	5
e	6	2

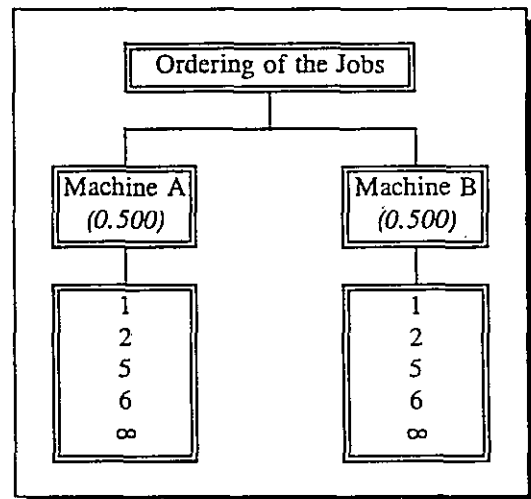


Figure 2 - The AHP Model of Johnson's Algorithm.

lower extreme value. Therefore, we can arbitrarily set  $v = 0.1$ . See Table II. Next we rate the jobs using the appropriate elements of R and synthesize the schedule which is {c,a,d,b,e}. See Table III. For validation, we apply Johnson's algorithm directly and find agreement in the optimal schedule which we note has a makespan of 22 time-units. This proof demonstrated the minimality of the makespan within the AHP model along lines similar to those used by Johnson to derive his classical result.

Note that in addition to producing a rank order, we also have ratio scale priorities for the jobs which can be used to compute expected times of completion of a job relative to the makespan time.

### 5. Generalization to Three Machines and Beyond

Johnson's approach is applicable only to the two machine problem, hence this restriction must also apply to the AHP approach. The algorithm makes no reference to a third machine, hence for one to extend the applicability of this approach, an appropriately justified method for dealing with the additional information and additional constraints must be found.

Consider a three machine fixed-sequence process. If one were to disregard the second machine in a three machine process and use Johnson's algorithm, it is clear that the effects of the second machine are not accounted for. Since the algorithm is concerned with the minimum time on either stage of a two-stage process, the coupling of the first and second machines or the second and third machines will help increase the likelihood that the influence of the second machine will be accounted for in the model.

Both Johnson's algorithm and the AHP approach weight the job times required on the first and second stages. Thus to use this  $m=2$  AHP scheduling approach with an  $m > 2$  machine fixed-sequence process, the  $m$  machine process must be examined as though it were a two machine process. Consider the *coupled-machines* approach.

Suppose we *couple* the processes of several consecutive machines into a single all-encompassing process. In this case, we can reduce  $m$  to  $m' < m$ , hence one can consider the job process to be composed of  $m'$  different stages rather than  $m$  different machines. For example, suppose  $m=4$ . We

Table II - Time Priorities.

Machine 1		Machine 2	
1	= 1.0000	1	= 0.0000
2	= 0.1000	2	= 0.9000
5	= 0.0100	5	= 0.9900
6	= 0.0010	6	= 0.9990
$\infty$	= 0.0001	$\infty$	= 1.0000

Table III - AHP Ratings Mode.

Alternatives	MACH A	MACH B	Total
1 Job c	1	2	0.950
2 Job a	2	6	0.549
3 Job d	2	5	0.545
4 Job b	5	6	0.505
5 Job e	6	2	0.450

Alternatives	.5000	.5000	Total
1 Job c	1.0000	0.9000	0.950
2 Job a	0.1000	0.9990	0.549
3 Job d	0.1000	0.9900	0.545
4 Job b	0.0100	0.9990	0.505
5 Job e	0.0010	0.9000	0.450

Alternatives	MACH A	MACH B	Total
1 Job c	0.5000	0.4500	0.950
2 Job a	0.0500	0.4995	0.549
3 Job d	0.0500	0.4950	0.545
4 Job b	0.0050	0.4995	0.505
5 Job e	0.0005	0.4500	0.450

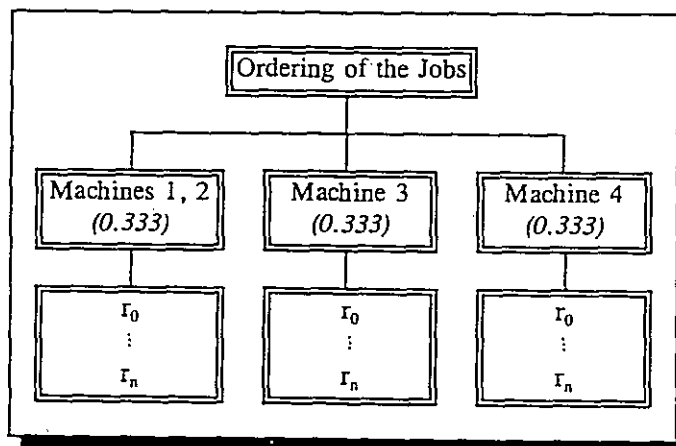


Figure 3 - A partially coupled AHP scheduling model for 4 machines.

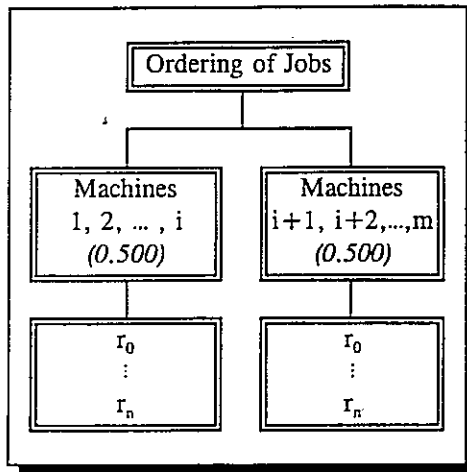


Figure 4 - A fully coupled model.

can couple machines 1 and 2 together and leave machines 3 and 4 apart<sup>2</sup>. See Figure 3. Each of the remaining  $m'$  stages could themselves be coupled by this approach until we have arrived at 2 distinct processing stages. See Figure 4. Each coupling is simply the theoretical operation of the coupled machines as though they were one unit. Thus the amount of time required for a particular job on the coupled unit is greater than or equal to the total time required for that job on each machine coupled together. This approach assumes that there is no unused time during the operation of the coupled unit, when in reality, this coupled unit is simply a group of machines, one or more of which may encounter unutilized time prior to its deployment. Hence this assumption must be accounted for and treated appropriately.

Suppose we apply the coupling approach to  $m$  machines where one coupling is composed of  $i$  machines and the other coupling is composed of  $m-i$  machines. If  $i > m-i$ , then the jobs rated by the coupling with  $i$  machines should have a total time generally larger than the total job times of the coupling with  $m-i$  machines; the converse is also possible. Hence there is a bias toward the coupling of fewer machines (and of lower times). Thus we propose the following solution.

There are  $m-1$  feasible coupling models. If one were to apply the  $m=2$  AHP model to a particular coupling configuration, one would obtain the optimal solution within the domain of that coupling. This local optimum (the result from a coupling model) may not be a global optimum (the result if machine coupling had not taken place). For emphasis, we note that this local optimum assumes there is no unutilized time within the coupled-machines stage. Therefore the time required on this coupled unit by the  $k^{\text{th}}$  job is equal to the sum of the times required on each machine in this coupled unit by the  $k^{\text{th}}$  job. Given that this assumption may not always hold, we consider all  $m-1$  possible couplings and synthesize the  $m-1$  local optimum solutions. These  $m-1$  sets of vectors can be synthesized together by the use of a governing hierarchy. Let  $p_{i,k}$  be the synthesized priority of the  $i^{\text{th}}$  job as found by the  $m=2$  machine AHP model where the first criterion of this 2-machine model has the first  $k$  machines coupled together and the second criterion has the remaining  $m-k$  machines coupled together. See Figure 5. This governing hierarchy considers all local optima by weighting the dominance of each coupling set on the

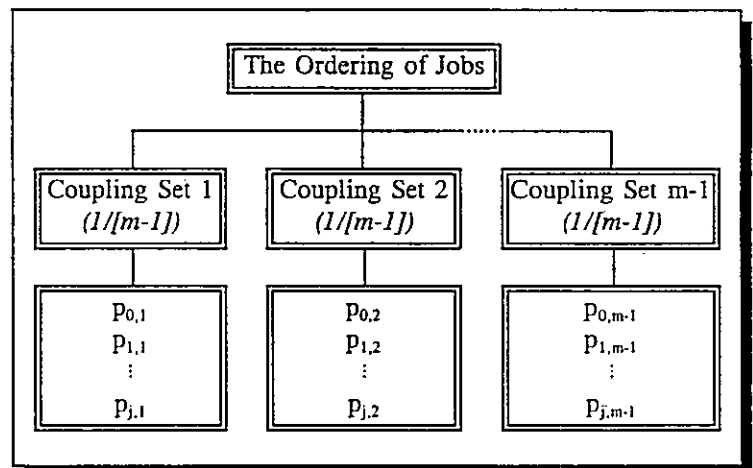


Figure 5 - The Hierarchy of Coupling Sets.

<sup>2</sup> Of course there exists other couplings, such as (1, 2, 3/4), (1, 2/3, 4), (1, 2/3/4), and (1/2/3, 4), but for the purpose of illustration, we show only (1/2, 3, 4).

global schedule. As such, the criteria level of this governing hierarchy is composed of the  $m-1$  coupling sets and the alternatives relative to each of these coupling sets is the local optimum as derived from the 2-machine case where each "machine" is the relevant coupling set. Since the AHP is designed to compile the judgments of a user, each local optimum is judged (and proven) to be the best set of weights (and optimal schedule) in that domain. Thus the AHP used in this manner will synthesize these  $m-1$  local vectors into a single global vector.

The global dominance that each of the  $m-1$  vectors has depends on the dominance of its coupling. It may be that in certain situations, one particular coupling may be more dominant than another and in other situations, this particular coupling may be less dominant than another. Therefore across all situations in general, each coupling should be regarded as equally dominant as any other. Hence for each of the  $m-1$  coupling sets (which are the criteria of Figure 5), we associate a priority of  $1/(m-1)$  thus retaining the required property that the total of the priorities between all criteria is 1. The synthesized priority of the jobs is thus obtained by

$$p_i = \sum_{x=1}^{m-1} \frac{P_{i,x}}{m-1} = \left[ \frac{1}{m-1} \right] \sum_{x=1}^{m-1} p_{i,x} \quad i = 1, \dots, n \quad (14)$$

where  $p_i$  is the synthesized priority of the  $i^{\text{th}}$  job, and  $j$  is the number of jobs to be scheduled. The rank order of these jobs indicates a balance between all feasible local optima. It may be that the synthesized rank as determined by the governing hierarchy is not equal to the rank of any of the local optima. However there are situations when to satisfy the majority, we may need to only partially satisfy each minority to maximize overall coverage [8].

## 6. An Illustration ( $m=6$ )

Consider a fixed-sequence job shop composed of  $m=6$  machines where each job must be processed on each machine for some predetermined time. Further, suppose there are 5 jobs to be scheduled. The times required by the 5 jobs on the 6 machines are indicated in Table IV. The first step in the procedure requires that one determine each of the 5 couplings.

Table IV - Job shop time requirements.

Jobs	Machines					
	1	2	3	4	5	6
a	7	2	5	6	6	4
b	2	6	6	2	7	6
c	8	6	8	8	6	7
d	1	3	3	3	5	5
e	3	5	8	3	9	8

The first unit of the first coupling is composed of machine 1 and the second unit of the first coupling is composed of the sum total of times of machines 2, 3, 4, 5, and 6. See Table V. The ratings list for this local situation is  $\{1, 2, 3, 7, 8, 19, 23, 27, 33, 35, \infty\}$ , where  $i=0$  is the index of the smallest rating, and  $i=n=10$  is the index of the largest. The weight of each rating for the first stage of this coupling set is:  $A_i = v^i$ ,  $i=0, \dots, 10$ ,  $0 < v \leq \frac{1}{2}$ . Similarly, the weights of each of these ratings for the second stage of this coupling set is:  $B_i = 1 - v^i$ ,  $\forall i$ ,  $v$  satisfying  $i=0, \dots, 9$ ,  $0 < v \leq \frac{1}{2}$ , and  $B_{10} = 1$ .

The first unit of the second coupling is composed of the sum of times required on machines 1 and 2, whereas the second unit of the second coupling is composed of the sum of times required on machines 3, 4, 5, and 6. See Table VI. The ratings list for this local situation is  $\{4, 8, 9, 14, 16, 21, 28, 29, \infty\}$ , where  $i=0$  is the index of the smallest rating of this list, and  $i=n=8$  is the index of the largest. The weight of each rating for the first stage of this coupling



is:  $A_i = v^i$ ,  $i=0, \dots, 8$ ,  $0 < v \leq 1/2$ . Similarly, the weight of each rating for the second stage of this coupling set is:  $B_i = 1 - v^i$ ,  $\forall i$ ,  $v$  satisfying  $i=0, \dots, 7$ ,  $0 < v \leq 1/2$ , and let  $B_8 = 1$ . This process is reiterated for the remaining couplings.

For each of these five two-stage processes, we compute the optimal schedule. Since each coupling set is restricted to  $0 < v \leq 1/2$ , we arbitrarily set  $v=0.25$  and compute the 5 local schedules based on this value of the parameter. We note these five optimal schedules are optimal in their respective two-stage domains and are not necessarily optimal in the  $m$ -stage domain. The five local vectors can be found in Table VII, where the local ranks are the local schedules.

The second step in the  $m > 2$  problem is to construct the governing hierarchy. This hierarchy synthesizes together the  $m-1$  vectors. The  $m-1$  criteria correspond to the  $m-1$  coupling sets where each criterion receives a weight of  $1/(m-1) = 0.200$ . Since the goal is to determine a scheduling priority of each job, the 6 jobs are the alternatives. The weights have already been determined by the first step in the process (see Table VII). Finally, we synthesize the priorities of the criteria and alternatives of the governing hierarchy as defined by equation (14). These results are tabulated on the bottom of this page. The AHP derived schedule,  $\{d, e, b, c, a\}$ , requires 56 time-units which can be verified as optimal by enumerating the 720 possible schedules.

## 7. General Observations

Recall that the coupled-machine *notion* relied on the assumption that the amount of time required by a job on a coupled unit is greater than or equal to the sum of times required by a job on each machine represented by the coupled unit. In contrast, however, the coupled machine *approach* assumes that the amount of time required by a job in a coupled unit is equal to the sum of the times required by a job on each machine represented by the coupled unit. This assumption introduces problems when a machine enjoys idle time between the processing of jobs. It may be that the optimal schedule requires machine  $k$  to sit idle for

Table V - Coupling set number 1.

Jobs	Set 1		Set 2					
	1	sum	2	3	4	5	6	sum
a	7	7	2	5	6	6	4	23
b	2	2	6	6	2	7	6	27
c	8	8	6	8	8	6	7	35
d	1	1	3	3	3	5	5	19
e	3	3	5	8	3	9	8	33

Table VI - Coupling set number 2.

Jobs	Set 1			Set 2				
	1	2	sum	3	4	5	6	sum
a	7	2	9	5	6	6	4	21
b	2	6	8	6	2	7	6	21
c	8	6	14	8	8	6	7	29
d	1	3	4	3	3	5	5	16
e	3	5	8	8	3	9	8	28

Table VII - Ideal weights for each coupling set (schedules for each two-stage coupling).

Jobs	Set 1	Set 2	Set 3	Set 4	Set 5
a	0.507	0.531	0.500	0.122	0.469
b	0.531	0.999	0.625	0.477	0.471
c	0.998	0.623	0.998	0.313	0.781
d	0.624	0.507	0.508	0.488	0.488
e	0.625	0.998	0.531	0.377	0.377

Job	Synthesized Results	
	Ideal Priority	Rank Order
a	0.3137	5
b	0.5296	3
c	0.4754	4
d	0.7496	1
e	0.5299	2

Table VIII - Job shop time requirements.

Jobs	Machines					
	1	2	3	4	5	6
a	6	1	7	9	1	6
b	3	1	5	7	7	6
c	1	4	2	6	5	2
d	2	7	2	8	5	2
e	2	2	6	7	6	4

time greater than 0 time units between consecutive jobs  $i$  and  $j$ . This idle time increases the time required on that coupled unit. However, since the machine time (and hence coupled-unit process time) spent idle is unknown, the ratings list contains the minimum time used on a coupled-unit and assumes no dead-time.

To illustrate this problem, consider a modified version of the above example based on the data listed in Table VIII. The derived AHP synthesis for this example is in the table below. The synthesized schedule has a makespan of 53 time-units. After enumerating all possible schedules, we find the

minimum makespan is 51. The AHP derived schedule is in the 93.3 percentile which means that 93.3% of the possible schedules require a makespan equal to or greater than the AHP derived schedule. Thus the following must be asked: since the AHP approach does not guarantee an optimal schedule, how efficient is the heuristic?

Clearly, the larger the *Percentile Index* (PI), the better the heuristic. However, other indices exist which have more informative interpretations. The *Optimality Index* (OI) indicates the probability that the heuristic will give the optimal schedule. Like the PI, the greater the OI, the more valuable the heuristic. Yet the cost of attaining the optimal schedule may outweigh its benefit. It. Thus the *Efficiency Index* (EI) is equal to the probability that the derived schedule is better than the mean makespan. The assurance of "better than expected" may be sufficient in certain problems, and the EI may prove useful.

Job	Synthesized Results	
	Ideal Priority	Rank Order
a	0.4017	4
b	0.6207	1
c	0.5316	3
d	0.3281	5
e	0.5820	2

To compute these indices, all possible schedules must be enumerated and compared to the AHP derived priorities. Simulation was repeated 500 times whereby the number of jobs and machines were randomly generated ranging between 2 and 5, and 2 and 50, respectively. The mean of each index was computed and the results are

given in Table IX.

Each of the indices shows that when  $m=2$ , the AHP derived outcome is optimal in all respects. However, the heuristic decreases in reliability as  $m$  increases. If  $m > 2$ , this same conclusion can be drawn as  $n$  increases. Nonetheless, in each instance, EI clearly indicates that the AHP result is better than average. Likewise, PI demonstrates that the percentile of the derived schedule is quite strong relative to all other possible schedules. OI, which seems to be the weakest indicator of all the indices, indicates that it is more difficult to derive the optimum schedule as  $n$  and  $m$  increase yet in each instance, it is still highly probable that the optimum makespan will be derived. Despite this seemingly pessimistic conclusion, by analyzing EI, we know the makespan of the derived schedule is still better than the expected makespan.

The potential suboptimality of the schedule found with the  $m > 2$  AHP approach is due to the dead-time incurred by some of the machines between jobs. Since this unutilized time is not accounted for in the  $m > 2$  AHP model, the assumption regarding the amount of time a job will require of a coupled unit must hold. Since the optimal solution may have dead time, this assumption may not hold in all situations. Given that dead time can only accumulate between

Table IX - Composite of 500 simulations for different job-shop parameters.

Mean Percentile Index

Number of Jobs	Number of Machines												
	2	3	4	5	6	7	8	9	10	20	30	40	50
2	100	93.5	95.3	97	93.5	94.5	98.5	96.0	96.5	93.5	96.5	96.0	96.0
3	100	94.0	90.7	89.8	94.0	90.7	90.2	90.3	91.2	90.2	90.8	90.0	89.1
4	100	95.0	91.6	92.9	91.7	89.2	88.7	89.2	91.4	89.7	87.7	88.5	84.2
5	100	95.3	94.1	93.2	93.0	93.1	89.8	90.0	90.0	88.8	89.6	88.5	86.2

Mean Optimality Index

Number of Jobs	Number of Machines												
	2	3	4	5	6	7	8	9	10	20	30	40	50
2	100	87.0	88.5	94.0	87.0	89.0	97.0	92.0	93.0	87.0	93.0	92.0	92.0
3	100	77.0	71.0	67.0	77.0	70.0	63.0	66.0	68.0	65.0	62.0	59.1	55.1
4	100	72.0	60.0	55.0	54.0	48.0	39.0	41.0	53.0	40.1	42.1	42.8	52.5
5	100	57.0	36.0	44.0	35.2	29.7	40.1	29.4	22.0	27.9	32.1	35.7	28.3

Mean Efficiency Index

Number of Jobs	Number of Machines												
	2	3	4	5	6	7	8	9	10	20	30	40	50
2	100	87.0	88.5	94.0	87.0	89.0	97.0	92.0	93.0	87.0	93.0	91.0	90.0
3	100	93.0	91.0	91.0	95.0	91.0	94.0	95.0	94.0	91.0	93.4	93.5	91.1
4	100	99.0	93.0	96.0	97.0	94.0	94.0	93.0	97.0	96.8	91.8	93.8	94.7
5	100	97.0	98.0	96.0	99.0	98.4	98.7	96.9	98.5	98.7	95.5	94.6	94.7

machines, there are  $m-1$  possible locations for this unaccounted increase in job-time requirements. Thus as  $m$  increases, so too does the probability that the optimal schedule may utilize unaccounted dead time. As the number of jobs increases, the number of feasible schedules increases, and thus the likelihood also grows that this dead time will occur. Hence, the zero-dead-time assumption becomes more unrealistic as the system grows in size and the overall efficiency of the heuristic will decrease. Nevertheless, within the job-shop domain as illustrated in Table VII, the AHP scheduling heuristic demonstrates a strong measure of "better than average" efficiency and may prove more useful than the costly full-enumeration approach.

## 8. Discussion and Conclusion

The *redundant-coupled-machines* is a modified form of the coupled-machines approach. Consider again the case when  $m=3$ . We could couple the first two machines together as one unit and couple the second two machines together as another unit. We refer to this situation as *redundant* since the second machine is represented in both couplings. The two couplings, "Machines 1 and 2" and "Machines 2 and 3" represent the total amount of time that a job will

require between both machines listed in the respective title. In this manner, the process is also seen as a two-stage process, while the second machine of the three machine process acts as a bridge connecting the first and third stages. We note the redundant-coupled-machines approach is different from the coupled-machines approach because the middle processes are repeated.

This alternate proposal suffers from several problems: the approach may bias the model in favor of the redundant machine; there is a lack of independence between the two couplings; and this approach does not reduce to the  $m=2$  case. Like the (non-redundant) coupled-machines approach, this alternate model suffers from the same dead-time problems. Since unutilized time is not accounted for, the assumption regarding the amount of time a job will require of a coupled unit must hold. The optimal schedule may have dead time, thus this assumption may not hold in all situations. Since dead time can only accumulate between machines, there are  $m-1$  possible locations for this unaccounted increase in job-time requirement. Hence as  $m$  increases, so too does the probability that the optimal schedule may incorporate such unaccounted dead time. As the number of jobs increases, the number of feasible schedules grows, increasing the likelihood that this dead time will occur. Hence, the zero-dead-time assumption becomes more unrealistic as the system grows in complexity and the overall efficiency of this heuristic should decrease.

The fundamental difference between the proposed AHP models the other scheduling strategies suggested in the literature is that the AHP models are based on the priorities of times and machines. In contrast, Johnson's algorithm and most others proposed compare not the importance of various aspects of the jobs; rather they compare the cardinality of the jobs relative to the times required. The dead-time encountered between machines has repercussions on the entire system. Therefore one must consider such feedback and determine the amount of influence it may have. Since the AHP can implement feedback and feedforward in one system in the supermatrix, the proposed AHP scheduling model can be advanced by the inclusion of this dependency. Hence, prioritization proves applicable in fixed-sequence job shop scheduling and since ratio scales can be composed and composition of cardinal scales is sometimes difficult, the ease and applicability of the *coupled-machine governing hierarchy approach* of the AHP may prove more beneficial in the future in the field of scheduling than all of the traditional techniques.

## 9. References

- [1] Ashour, S., "A Brand-and-Bound Algorithm for Flow Shop Scheduling Problems," *AIIE Trans.*, v2, 1970.
- [2] Campbell, H. G., Dudek, R. A., and M. L. Smith, "A Heuristic Algorithm for the n-job m-machine Sequencing Problem," *Management Science*, v16, 1970.
- [3] Dudek, R. A., Panwalkar, S. S., and M. L. Smith, "The Lessons of Flowshop Scheduling Research," *Operations Research*, v40n1, 1992.
- [4] Garey, M. R., Johnson, D. S., and R. Sethi, "Complexity of Flow Shop and Job Shop Scheduling," *Math. Operations Research*, v1, 1976, pp. 117-29.
- [5] Johnson, S. M., "Optimal Two and Three-Stage Production Schedules with Setup Times Included," *Naval Research Logistics Quarterly*, v1, pp 61-68, 1954.
- [7] Nawaz, M., Enscore, E. E., and I. Ham, "A Heuristic Algorithm for the m-machine n-job Flow Shop Sequencing Problem," *Omega*, v11, 1983, pp. 91-5.
- [6] Palmer, D. S., "Sequeing Jobs Through a Multistate Process in the Minimum Total Time-A Quick Method of Obtaining a Near Optimum," *Operational Research Quarterly*, v16, 1965.
- [8] Saaty, T. L., *The Analytic Hierarchy Process*, New York, New York, McGraw-Hill Book Company, Inc., 1980.
- [9] Saaty, T. L., *Multicriteria Decision Making: The Analytic Hierarchy Process*, RWS Publications, Pittsburgh, 1994.
- [10] Saaty, T. L., *Fundamentals of Decision Making and Priority Theory*, RWS Publications, Pittsburgh, 1994.
- [11] Smith, R. D., and R. A. Dudek, "A General Algorithm for Solution of the n-job m-machine Scheduling Problem of the Flow Shop," *Operations Research*, v15, 1967.
- [12] Turner, S., and F. D. Booth, "Comparisons of Heuristics for Flow Shop Scheduling," *Omega*, v15, 1987.