Research Paper

Volume : 5 | Issue : 1 | JANUARY 2016 • ISSN No 2277 - 8179

Optimal Job Sequence Involving Three or More Machines

Mathematics

KEYWORDS : Minimization , Johnson's algorithm, Lexisearch.

CH. N. ANURADHA	Department of Mathematics, Vasavi College of Engg., Hyderabad, India
V. V. HARAGOPAL	Department of Statistics and Center For Quantative Methods, Osmania University, Hyderabad
SHAHNAZ BATHUL	Department of Mathematics, JNTU, Hyderabad, India

ABSTRACT

The problem of Optimal Job Sequencing (Where processing order of the machines are the same for all jobs and objective is minimizing the makespan) is considered in the general case of M>=3 machines. After reviewing some special cases where the problem is reducible to equivalent two machines case which can be solved by Johnson's algorithm, procedures for effective bound setting and hence effective lexisearch for optimal solutions in case of general problems with no special restrictions are presented. Computational efficiency of these bounding procedures are investigated by solution of randomly generated problems of different sizes.

1.INTRODUCTION :

The problem of Optimal Job Sequencing where process order of the machines are the same for all jobs and objective is to minimize the makespan is easily solved by the well known Johnson's algorithm. However, the problem is a really hard problem when the number of machine's m is greater than 2. Like many other combinatorial optimization problems, Exact solutions appear to be possible only by (implicit) enumeration and search through solution space, in general.

However, under special conditions on the processing times of jobs on the different machines, it has been shown that as m (> 2) can be 'reduced' to an equivalent to machine problem such that the optimal solution sequences of the later are also the optimal solution sequences of the former. As is again true with general combinatorial optimization problems, in this problem also good heuristics exist and one can even

pickup and optimal sequence very 'quickly' but it may take inordinate (relatively) heavy computational effort to recognize that the guessed / proposed / suspected solution on hand is indeed an optimal solution. In that follows, we shall investigate a lexi search procedure which includes effective bound setting which appears to be very efficient. The procedure is essentially the one based on Bhanumurthy (1986).

However, before one goes into these details, a brief 'review' of and comments on the possibility of reducing an m (> 2) machine problem to an equivalent 2 machine problem are in order.

One well known structural constraint is the following:

Let $(t_{ij},\ i = 1,\ n\ ;\ j = 1, 2, 3)$ be the processing times of n jobs on 3 machines written as an $n \times 3$ matrix. Then the problem is reducible to a 2 machines problem with processing times $A_{ij},\ i = 1, n;\ j = 1,2$ *where* $A_{i1} = t_{i1} + t_{i2}$ *and* $A_{i2} = t_{i2} + t_{i3}$,if column 1 of (t) 'dominates' column 2 of (t), j that is, j if $t_{i1} \geq t_{j2}$ *for all* i, j. A similar sufficient condition is that column 3 of t dominates column 2 of (t).

Quite a few further generalization, to the case of more than 3 machines also are reported in the literature (see, for instance szwsarc.w (1977)), the most recently, reported one

perhaps is Bagga and bhambani (1997). However, two interesting properties of the scheduling problem one in the general case and the other in the case of problems reducible to two – machine case seem not to have received the attention they deserve. They are the following:

2.THE PROBLEM:

1. The makespan for any given job sequence is essentially a "critical path " for an activity network. It is the largest of the paths from the start of processing first job in the sequence on the first machine to the completion of the last job in the sequence on the last machine (cf. personal communication from TKB Rao).

Consider the '' reverse image '' of this activity network where the order of the sequence is reversed as also the ordering of the machines. It is obvious that this reverse network has the same critical path as earlier only the direction of 'flow' is reversed.

Hence, these two networks are in a sense duals of one another, having the same value as the makespan for any sequence of the first problem and the makespan of the dual (that is reverse order) sequence of the dual problem.

Hence if π^* $(\alpha_1,. . . .\alpha_n)$ is an optimal sequence for the first problem, the " reverse problem " has $(\alpha_n, \alpha_{n-1},. . . . \alpha_1)$ as an optimal sequence.

To illustrate ; consider.

	1	2	3	4
1	7	5	9	2
2	1	8	6	4
3	6	7	8	10

7_7^*	5_{12}^*	9_{21}	2_{23}
1_8	8_{20}^*	6_{27}	4_{31}
6_{14}	7_{27}^*	8_{35}^*	10_{45}^*

*the critical path is {11, 12, 13, 23, 33, 34} The dual (reverse image) problem is

10_{10}^*	8_{18}^*	7^*_{25}	6_{31}
4_{14}	6_{24}	8_{33}^*	1_{34}
2_{16}	9_{33}	5_{38}^*	7_{45}^*

Volume : 5 | Issue : 1 | JANUARY 2016 • ISSN No 2277 - 8179

And has the critical path {11, 12, 23, 22, 33, 34}. The reverse symmetry is seen by writing the second path in reverse order adding the component "coordinates" j

$$\{11,\ 12,\ 23,\ 22,\ 33,\ 34\}$$

$$\{34,\ 33,\ 22,\ 23,\ 12,\ 11\}$$

$$Component\ wise\ total \qquad \{45,\ 45,\ 45,\ 45,\ 45,\ 45\}$$

From the observation is obvious that the two problems have optimal (that is minimal) makespan as the same value. It therefore, also follows that structural constrains connected with any properties of makespan values also have a "dual interpretation". Since column dominance is non-directional either from top to bottom or bottom to top, dominance is not altered , while left to right ordering of elements in a row gets reversal as right to left in the dual problem.

It now follows that the properties of reducibility of a 3 machines problem into 2 machine if column 1, dominate column 2 also leads to the 'reflected' result that reducibility holds also, if column 3 dominates column 1.

Direct extension of this result to the case of 'more than 3' machines problem is obvious.

3.3-MACHINE PROBLEM:

3. When the 3 machines problem is reducible to 2 machine case, the difference in makespan for the same sequence in the two problem is invariant, (and, infact, is equal to the total of the processing times on the middle machine). (Schwartz 1974).

Bhanumurthy (1986) presents a 'search reduction matrix (SRM) concept that can be very usefull in many sequencing situations. It often happens that between two sequence which differ only in an adjacent pair jobs, one of them is necessarily better than the other in 'Context – free 'sense.

Let($P_\sigma \alpha\beta Q_\sigma \ and \ P_\sigma \beta\alpha Q_\sigma$) be an arbitrary pair of sequence which differ only in the relative ordering of two neighboring jobs $\alpha \ and \ \beta$, under certain conditions only on "the processing times" of the two jobs $\alpha \ and \ \beta$, it will be possible to prove that

$V(P_\sigma \alpha\beta Q_\sigma) \leqq V(P_\sigma \beta\alpha Q_\sigma)$ for all $(P_\sigma\ Q_\sigma)'s$, it follow that if $\alpha, \beta \notin P_\sigma$, then one can exclude $P_\sigma \beta\alpha Q_\sigma$ *from the set of plausible optimal solution*. Then, if at all an optimal solution$P_\sigma \alpha\beta Q_\sigma$ exists, then the set $(P_\sigma \beta\alpha Q_\sigma^1)$ where Q_σ^1 is any subsequently permuting elements of Q_σ is excluded from further consideration for optimality.

An SRM is an $n \times n$ matrix in which the entry P_{ij} takes value ± 1 *or* 0 depending on whether one can establish that $(P_\sigma jiQ_\sigma)$ can not be optimal (+ 1) (and then $P_{ij} = -1$) or whether no such unambiguous conclusion be drawn (0).

A scrutiny of the structure of this matrix can fix the first few or the last few of the elements in an optimal sequence without any further computational effort, search being restricted to only the remaining elements. As Bhanumurthy has shown in the 3 machine case, this reduction can be substantial. Extension of this concept to the case of four and more machines cases is under investigation and will be reported in a subsequent communication. In the next section we discuss the Lexi search approach to 'm' machine problem.

4.The Lexi search approach to m machine sequence problem:- As the concepts of Lexi serach are relatively simple and straight forward – once they are spelled out and illustrated, we shall present an illustration to introduce and explain this approach.

Let us consider the following 4 machine problem with 5 jobs.

T =

jobs\machines	1	2	3	4
a	8	7	3	10
b	1	5	6	8
c	10	2	7	3
d	8	4	3	7
e	6	5	3	11
Colum Total	33	23	22	39

(1)	(2)	(3)
20	13	10
19	14	8
12	10	3
14	10	7
19	14	11

Column (1), (2), (3) are from right to left the cumulative totals of the elements in colum 4, 3 and 2 of 7. The alphabet table rearranges the entries in the columns (1), (2) and (3) in ascending order along with the corresponding job label. Now for the bounds and the search table the computational follows:

T=

	1	2	3	4
a	8	7	3	10
b	1	5	6	8
c	10	2	7	3
d	8	4	3	7
e	6	5	3	11

Alphabet Table

c-12	c-10	c-3
d-14	d-10	d-7
b-19	a-13	b-8
e-19	b-14	a-10
a-20	e-14	e-11

backward

b-1	b-6	b-12
e-6	e-11	e-14
a-8	c-12	d-15
d-8	d-12	a-18
c-10	a-15	c-19

Forward

	33	23	22	39
Total Min.	1	2	3	3

Makespan of any sequence of these 5 jobs can not be less than 33 + 12 = 45, 1+23+10=34, (1+5) + 22 + 3 = 31, (1+5+6) = 39 + 51. Thus, no solution can have a value less than 51, that is minimum possible makespan is at least 51. Though it can be much more. The makespan for the sequence V(a, b, c, d, e) = 57. Hence, maximum "hopable" improvement is 57 – 51 =6. Now the search table is as follows:

	1	2	3	4
a (57)	8_8 +25+12 (45)	7_{15} +16+10 (41)	3_{18} +19+3 (40)	10_{28} +29+0 (57)
ab (57)	1_9 +24+12	5_{20} +11+10	6_{26} +13+3	8_{28} +21+0
abc (57)	10_{19} +14+12	2_{22} +9+10	7_{33} +6+3	3_{39} +18
abcd (57)	8_{27} +6+19	4_{31} +5+14	3_{36} +3+11	7_{46} +11+0
abcde (57)	6_{33} +0+19	5_{38} +0+14	3_{41} +0+11	11_{57} +0+0
b (51)	1_1 +32+12	5_6 +18+10	6_{12} +16+3	8_{20} +31+0(51)
ba (51)	8_9 +24+12	7_{16} +11+10	3_{19} +13+3	10_{30} +21+0
bac (52)	8_{27} +6+14	4_{31} +5+10	3_{34} +3+7	7_{41} +11+0(52)
bacd (52)	6_{33} +0+19	5_{38} +0+14	3_{34} +0+11	11_{52} +0+0(52)
bacde (52)	6_{25} +8+19 (52)			

	1	2	3	4
bad (51)	8_{17} +16+12	4_{21} +7+10	3_{24} +10+3	7_{37} +14+0
badc (51)	10_{27} +6+12	2_{29} +5+10	7_{36} +3+3	3_{40} +11+0
badce (52)	6_{33} +0+19 52*	5	3	11
bade (51)*	6_{23} +10+12	5_{28} +2+10	3_{31} +7+3	11_{48} +3 (51)
badec (51)	10_{33} +0+12	2_{35} +0+10	7_{42} +0+3	3_{51} +0+0 (51)*

Search Over

5.CONCLUSIONS:

An interesting heuristic in this context is the following:

If a job has minimal forward cumulative times and another one say α has minimal backward cumulative times, (the former, any way, is computed as part of constructing constructing the alphabet table), sequences with structure $\alpha\,\sigma\,\beta$ where sigma is a permutation of job set after excluding $\alpha\ and\ \beta$ from it are likely to have makespans much nearer to the optimum than other sequence. In the illustrative example, one has $\alpha \equiv b\ and\ \beta\ as\ c$. Also, after excluding this pair, job 'e' has smallest backward cumulative totals. Hence, a good guess would be the sequence $be\sigma_3\sigma_4 c\ says\ \{beadc\}$. This sequence has the value 51, which turns out to be equal to the bound as well and hence, obviously is one of the optimal solution. Of course this "rule" is only heuristic and and is not one which necessarily gives the optimal sequence.

REFERENCE

1. S.Z.WARC,W. (1974) "Mathematical Aspects of the 3xn Job shop sequencing problem", Nav. Res. log. Quart, Vol 2 : 145-153. | | 2. Johnson, S.M (1954): "Optimal Two and Three stage Production schedules with set up times Included", Nav. Res. Log. Quart . 1 61-68. | | 3. Bhanumurthy,A(1986): "Some Problems in sequencing and scheduling Ph.D thesis", Osmania University. | | 4. Bagga. P.C and Bhambani.A(1997): "Sequencing with restrictions in processing times. Opsearch vol 34: No.2,116-127. | | 5. S.Z.WARC,W,(1977): Special cases of the flow-shop problem. Nav. Res. Log. Quart vol 24 pp 483-492. |