Computers Math. Applic. Vol. 19, No. 2, pp. 1–7, 1990
Printed in Great Britain.

0097-4943/90 $3.00 + 0.00

DYNAMIC PROGRAMMING APPROACH TO THE SINGLE-MACHINE SEQUENCING PROBLEM WITH DIFFERENT DUE-DATES

T. C. E. CHENG

Department of Actuarial and Management Sciences, University of Manitoba, Winnipeg, Manitoba R3T 2N2, Canada

(*Received* 2 *December* 1988)

Abstract—Given a set of n jobs each of which is assigned a due-date and all jobs are simultaneously available to be processed on a single machine, the problem is to find the optimal job processing order that minimizes the sum of absolute deviation of job completion times about their respective due-dates. Since this problem has been shown to be NP-complete, we present a DP (dynamic programming) formulation of the problem and apply the principle of optimality of DP to find the optimal solution by implicit enumeration.

INTRODUCTION

The optimal due-date scheduling problem has attracted considerable attention of the scheduling community lately. Both academic researchers and practising managers are thrilled at solving the problem of optimally scheduling jobs to meet their due-dates. This may be attributable to the recent surge in popularity of the concept of just-in-time (JIT) production in the manufacturing industry. The essence of JIT production is to have the right amount of materials of the right quality at the right time in the right place to produce the right quantity of items demanded by the next stage of production. Thus the success of a JIT production system is greatly hinged on the ability of the system to deliver the materials perfectly on time. Any deviation from the required delivery time, be it early or late, will result in poor system performance.

Voluminous amount of research work has been done on the subject of due-date scheduling over the years. Extensive surveys of much of these research results have been conducted by various authors who, among others, include Sen and Gupta (1984), Gupta and Kyparisis (1987) and Cheng and Gupta (1989). Much of the passion the researchers have for this aspect of scheduling research stems from the theoretical challenge associated with solving the difficult due-date scheduling problem and the large extent to which the theoretical results can be applied in real situations. The name of the game in today's business is productivity and competitiveness. Therefore the success of a firm depends to a large extent on its ability to deliver customer orders on time and to minimize waste through adopting such modern production methods as JIT. An excellent treatment of due-date management in the job shop production environment can be found in Ragatz and Mabert (1984).

PROBLEM FORMULATION

We consider in this paper the problem of sequencing n independent jobs on a single machine where each job is assigned a different due-date. The objective is to find the optimal job sequence that minimizes the sum of absolute deviation of job completion times about their respective due-dates. As has been discussed earlier, the relevance of this objective function is apparent in a JIT production environment in which perfect on time delivery is emphasized and both early and late delivery are considered undesirable.

Let $N = \{1, 2, \ldots, n\}$ be a set of n independent jobs simultaneously ready to be sequenced on a single machine which can process no more than one job at a time. Once a job is started on the machine, it will be processed without interruption until completion. Job i requires t_i processing time on the machine and is assigned a due-date d_i, $\forall_i \in N$; both t_i and d_i are assumed to be deterministic

and known before processing begins. Let π be the permutation set of the n jobs and let σ be arbitrarily one of the $n!$ possible job sequences. Also let the subscript $[i]$ denote the job $j \in N$ in position i of σ, then $C_{[i]}$ denotes the completion time of j in σ. The objective function to be minimized can be written as

$$f(\sigma) = \sum_{i=1,\ldots,n} |C_{[i]} - d_{[i]}|. \tag{1}$$

This problem is a special case of the class of general due-date assignment and sequencing problems. The objective of these problems is to find the joint optimal due-date values and the optimal job sequence to minimize a total penalty function depending on the assigned due-dates and the individual job earliness and tardiness values. An efficient algorithm is available in the literature for solving a special case of the problem in which all jobs are assigned a common due-date, i.e. $d_i = d, \forall i \in N$, and d is constrained to be no less than the makespan $\mathrm{MS} = \Sigma_{1 \le i \le n} t_i$. Kanet (1981) presents the original results and algorithm which are later extended by various researchers, including Sundararaghavan and Ahmed (1984), Bagchi *et al.* (1986), Hall (1986), Cheng (1987), Emmons (1987) and Quaddus (1987).

As for the problem with different due-dates, Garey *et al.* (1988) have recently shown that it is NP-complete. Thus it is highly unlikely that a polynomial-bound algorithm can be found. The only viable solution methods available are the use of heuristics and implicit enumeration. While the heuristic approach is fast in obtaining a solution and so is suited for large problems, it does not guarantee to yield the optimal solution. On the other hand, an implicit enumeration method is more time consuming and is generally limited to small- to medium-sized problems, but it is guaranteed that the solution found is optimal. An effective heuristic for the problem addressed in this paper has been presented by Ahmed and Sundararaghavan (1984). We shall present a DP approach to solving the problem in this paper. According to the DP approach, the original problem is first decomposed into a series of smaller problems which are optimally solved in a sequential manner. The optimal solution of the original problem is then constructed from those of the smaller problems following the principle of optimality of DP.

DYNAMIC PROGRAMMING SOLUTION

Held and Karp (1962) are probably the first researchers to apply DP to sequencing problems. Their idea is based on a simple observation that if the objective function of a sequencing problem can be decomposed into a series of objective functions corresponding to a series of smaller problems, then the problem can be solved by the DP technique. We will follow the approach of Held and Karp to decompose our problem so that it can sequentially be solved by DP.

Let $\sigma^* = ([1]^*, [2]^*, \ldots, [n]^*)$ be an optimal job sequence, then the objective function associated with σ^* can be written as

$$f(\sigma^*) = \sum_{i=1,\ldots,n} |C^*_{[i]} - d^*_{[i]}|. \tag{2}$$

For any $k = 1, 2, \ldots, n$, (2) can be decomposed as

$$\begin{aligned} f(\sigma^*) &= \sum_{i=1,\ldots,k} |C^*_{[i]} - d^*_{[i]}| + \sum_{i=k+1,\ldots,n} |C^*_{[i]} - d^*_{[i]}| \\ &= f(\sigma^*_A) + f(\sigma^*_B), \end{aligned} \tag{3}$$

where $\sigma^*_A = ([1]^*, [2]^*, \ldots, [k]^*)$ and $\sigma^*_B = ([k+1]^*, [k+2]^*, \ldots, [n]^*)$.

Let $A = \{[1]^*, [2]^*, \ldots, [k]^*\}$ and $B = \{[k+1]^*, [k+2]^*, \ldots, [n]^*\}$ be two subsets of N. Then the following lemma can be shown to hold.

Lemma 1

If σ^* is an optimal job sequence for N, then the subsequence σ^*_A obtained from decomposition of $f(\sigma^*)$, as determined from (3), is an optimal job sequence for the reduced job-set A.

Proof of Lemma 1. Let σ^* be an optimal job sequence for N. Also let σ^*_A and σ^*_B be the subsequences of σ^* obtained from decomposition according to equation (3) and A and B be the

associated reduced job-sets. Suppose $\sigma_A' \neq \sigma_A^*$ is an optimal sequence for A. It follows that $f(\sigma_A') < f(\sigma_A^*)$. Thus we could construct a new sequence for N by concatenating σ_B^* to σ_A' which will result in an objective function value $f(\sigma_A') + f(\sigma_B^*) < f(\sigma_A^*) + f(\sigma_B^*) = f(\sigma^*)$. However, this is contradictory to the assumption that σ^* is optimal for N. Thus the converse must be true, so σ_A^* must be optimal for A. This completes the proof of the lemma.

We can now employ the result of Lemma 1 to decompose our problem which can then be solved by DP. Let J be some subset of N and $J' = N - J$ be the complement of set J. Define C_J as

$$C_J = \sum_{i \in J} t_i, \tag{4}$$

which is the sum of the processing times of all the jobs in J. Suppose a sequence σ for N has been constructed in which all the jobs in J precede every job in J'. If σ is to be optimal then Lemma 1 requires that the jobs in J must be optimally sequenced, no matter how the jobs in J' are sequenced given that none of these jobs can start earlier than C_J. Now let $F(J)$ be the minimum value of the objective function involving only those jobs in J. Employing Lemma 1 and observing that the last job in J must be completed at C_J, we have

$$F(J) = f(\sigma_J^*) = \min_{\sigma J \in \tau} \left\{ \sum_{i \in J} |C_i - d_i| \right\} = \min_{i \in J} \{ |C_J - d_i| + F(J - \{j\}) \}, \tag{5}$$

where τ is the permutation set of the jobs in J. If the set J contains no jobs, then $J = \varnothing$ and we define $F(\varnothing) = 0$. It follows that the minimum value of the objective function for the full problem is given by

$$F(N) = f(\sigma^*) = \min_{\sigma \in \pi} \left\{ \sum_{i \in N} |C_i - d_i| \right\} = \min_{i \in N} \{ |C_N - d_i| + F(N - \{j\}) \}. \tag{6}$$

The recurrence relations (5) and (6), along with the boundary condition $F(\varnothing) = 0$, enable us to find the minimum value of the objective function for N. We will start with all J containing a single job and use relation (5) to find the minimum value $F(J)$ for each J. Using these values and relation (5), we can find the minimum values of $F(J)$ for all J containing just two jobs; then for all J containing just three jobs; and so on until we eventually find $F(N)$. If we keep track of where the minimum in relation (5) occurred at each stage, we will be able to reconstruct the optimal job sequence σ^* after finding $F(N)$. This is exactly how DP is used to solve our problem. It is clear that Lemma 1 is just the principle of optimality of DP, while relations (5) and (6) are the recurrence relations required by DP to solve a problem in a sequential manner. Formally, the DP formulation of the due-date sequencing problem expressed in terms of standard DP terminology, see, for example, Hastings (1973) and Dreyfus and Law (1977), is as follows. Letting $|X|$ be the cardinality of the set X, we define

Stage: $S = |J|$, number of jobs in a reduced job-set J.
State: $J = \{j : j \in N \text{ such that } |J| = S\}$, a particular combination of S jobs.
Action: $a(S, J, j)$ = schedule job j after all other jobs in J.
Return: $r(S, J, j) = |C_J - d_i|$.
Optimal value function: $f(S, J)$ = minimum value of the objective function for the set J.
Optimal policy function: $P(S, J)$ = state $(J - \{j\})$ at stage $S - 1$, where j is such that $a(S, J, j)$ is optimal.
Recurrence relation: $f(S, J) = \min_{i \in J} \{r(S, J, j) + f(S - 1, J - \{j\})\}$.
Boundary condition: $f(0, \varnothing) = 0$.

Using the boundary condition and the recurrence relation, one can solve the problem recursively starting with $S = 0$ and $J = \varnothing$ until $f(n, N)$ is found which yields the minimum value of the objective function of the problem. The optimal job sequence is then reconstructed from $f(n, N)$ by tracing the stages that are involved in determining the minimum solution. This approach is known as forward DP because it works forward through subsequences of the full problem by considering reduced job-sets J processed first. A backward DP can also be used to solve the problem by

considering J processed last. Such an approach to sequencing problems has been presented by Lawler (1964).

AN EXAMPLE

To illustrate the DP solution method, we use the problem presented in Ahmed and Sundararaghavan (1984). It is a six-job problem with processing times $t_1 = 15$, $t_2 = 27$, $t_3 = 63$, $t_4 = 71$, $t_5 = 86$, $t_6 = 99$ and due dates $d_1 = 74$, $d_2 = 45$, $d_3 = 11$, $d_4 = 2$, $d_5 = 15$ and $d_6 = 14$. The complete DP calculations are shown in Table 1. It is clear that the optimal job sequence σ^* is (3, 1, 2, 4, 5, 6) and the minimum value of the objective function is $f(\sigma^*) = 884$ which are the same as those obtained by Ahmed and Sundararaghavan (1984).

COMPUTATIONAL REQUIREMENTS OF DYNAMIC PROGRAMMING

It is important to recognize the computational requirements of the DP solution method. At each stage S, DP generates $C(n, S)$ different states J, corresponding to the total number of ways of obtaining from N a reduced job-set J of size S, for consideration. Associated with each of the state J at stage S, there are S number of actions $a(S, J, j)$, corresponding to scheduling job j in J after

Table 1. DP solution of the six-job problem

J	$a(S, J, j)$	$r(S, J, j)$	$r(S, J, j) + f(S-1, J-\{j\})$	$F(S, J)$	$P(S, J)$
			Stage 1 ($S = 1$)		
{1}	1	59	59	*	{1}
{2}	2	18	18	*	{2}
{3}	3	52	52	*	{3}
{4}	4	69	69	*	{4}
{5}	5	71	71	*	{5}
{6}	6	85	85	*	{6}
			Stage 2 ($S = 2$)		
{1, 2}	1	32	50	*	{2}
	2	3	62		
{1, 3}	1	4	56	*	{3}
	3	67	126		
{1, 4}	1	12	81	*	{4}
	4	84	143		
{1, 5}	1	87	158		
	5	86	145	*	{1}
{1, 6}	1	40	125	*	{6}
	6	100	159		
{2, 3}	2	45	97	*	{3}
	3	79	97	*	{2}
{2, 4}	2	53	122		
	4	96	114	*	{2}
{2, 5}	2	68	139		
	5	98	116	*	{2}
{2, 6}	2	81	166		
	6	112	130	*	{2}
{3, 4}	3	124	192		
	4	132	184	*	{3}
{3, 5}	3	138	209		
	5	134	186	*	{3}
{3, 6}	3	151	236		
	6	148	200	*	{3}
{4, 5}	4	155	226		
	5	142	211	*	{4}
{4, 6}	4	168	253		
	6	156	241	*	{4}
{5, 6}	5	170	255		
	6	171	242	*	{5}
			Stage 3 ($S = 3$)		
{1, 2, 3}	1	31	128		
	2	60	116	*	{1, 3}
	3	94	144		
{1, 2, 4}	1	39	153		
	2	68	149	*	{1, 4}
	4	111	161		
{1, 2, 5}	1	54	170		
	2	83	228		
	5	113	163	*	{1, 2}

—continued opposite

Table 1—*continued*

J	$a(S, J, j)$	$r(S, J, j)$	$r(S, J, j) + t(S - \{j\}, J - \{j\})$	$F(S, J)$	$P(S, J)$
{1, 2, 6}	1	67	197		
	2	96	221		
	6	127	177	*	{1, 2}
{1, 3, 4}	1	75	259		
	3	138	219	*	{1, 4}
	4	147	331		
{1, 3, 5}	1	90	276		
	3	153	298		
	5	149	205	*	{1, 3}
{1, 3, 6}	1	103	303		
	3	166	291		
	6	163	219	*	{1, 3}
{1, 4, 5}	1	98	309		
	4	170	315		
	5	157	238	*	{1, 4}
{1, 4, 6}	1	111	352		
	4	183	308		
	6	171	252	*	{1, 4}
{1, 5, 6}	1	126	368		
	5	185	310	*	{1, 6}
	6	186	331		
{2, 3, 4}	2	116	300		
	3	150	264		
	4	159	256	*	{2, 3}
{2, 3, 5}	2	131	317		
	3	165	281		
	5	161	258	*	{2, 3}
{2, 3, 6}	2	132	332		
	3	166	296		
	6	163	260	*	{2, 3}
{2, 4, 5}	2	139	350		
	4	182	298		
	5	169	283	*	{2, 4}
{2, 4, 6}	2	152	393		
	3	195	325		
	6	183	297	*	{2, 4}
{2, 5, 6}	2	155	397		
	5	185	315		
	6	186	302	*	{2, 5}
{3, 4, 5}	3	209	420		
	4	218	404		
	5	205	389	*	{3, 4}
{3, 4, 6}	3	222	463		
	4	231	431		
	6	219	403	*	{3, 4}
{3, 5, 6}	3	237	479		
	5	233	433		
	6	234	420	*	{3, 5}
{4, 5, 6}	4	254	496		
	5	241	482		
	6	242	453	*	{4, 5}
			Stage 4 (S = 4)		
{1, 2, 3, 4}	1	102	358		
	2	131	350		
	3	165	314		
	4	174	290	*	{1, 2, 3,
{1, 2, 3, 5}	1	117	375		
	2	146	351		
	3	186	343		
	5	176	292	*	{1, 2, 3}
{1, 2, 3, 6}	1	130	390		
	2	159	378		
	3	193	370		
	6	190	306	*	{1, 2, 3}
{1, 2, 4, 5}	1	125	408		
	2	154	392		
	4	197	360		
	5	184	333	*	{1, 2, 4}
{1, 2, 4, 6}	1	138	435		
	2	167	419		
	4	210	387		
	6	198	347	*	{1, 2, 4}
{1, 2, 5, 6}	1	153	455		
	2	182	492		
	5	212	389		
	6	213	376	*	{1, 2, 5}

—*continued overleaf*

Table 1—*continued*

J	$a(S, J_j)$	$r(S, J_j)$	$r(S, J_j) + f(S-1, J - \{j\})$	$F(S, J)$	$P(S, J)$
{1, 3, 4, 5}	1	161	550		
	3	224	462		
	4	233	438	*	{1, 3, 5}
	5	220	439		
{1, 3, 4, 6}	1	174	577		
	3	237	489		
	4	246	465		
	6	234	453	*	{1, 3, 4}
{1, 3, 5, 6}	1	189	609		
	3	252	262		
	5	248	467		
	6	249	454	*	{1, 3, 5}
{1, 4, 5, 6}	1	197	650		
	4	269	579		
	5	256	508		
	6	257	495	*	{1, 4, 5}
{2, 3, 4, 5}	2	202	591		
	3	236	519		
	4	245	503		
	5	232	488	*	{2, 3, 4}
{2, 3, 4, 6}	2	215	618		
	3	249	546		
	4	258	518		
	6	246	502	*	{2, 3, 4}
{2, 3, 5, 6}	2	230	650		
	3	264	566		
	5	260	520		
	6	261	519	*	{2, 3, 5}
{2, 4, 5, 6}	2	238	694		
	4	268	565		
	5	281	583		
	6	269	552	*	{2, 4, 5}
{3, 4, 5, 6}	3	308	761		
	4	317	737		
	5	304	707		
	6	305	694	*	{3, 4, 5}
		Stage 5 (S = 5)			
{1, 2, 3, 4, 5}	1	188	676		
	2	217	655		
	3	251	584		
	4	260	552		
	5	247	537	*	{1, 2, 3, 4}
{1, 2, 3, 4, 6}	1	201	703		
	2	230	683		
	3	264	611		
	4	273	579		
	6	261	551	*	{1, 2, 3, 4}
{1, 2, 3, 5, 6}	1	216	735		
	2	245	699		
	3	279	655		
	5	275	581		
	6	276	586	*	{1, 2, 3, 5}
{1, 2, 4, 5, 6}	1	260	812		
	2	289	784		
	4	332	708		
	5	319	666		
	6	320	653	*	{1, 2, 4, 5}
{1, 3, 4, 5, 6}	1	296	990		
	3	359	854		
	4	368	822		
	5	355	808		
	6	356	794	*	{1, 3, 4, 5}
{2, 3, 4, 5, 6}	2	337	1031		
	3	371	923		
	4	380	899		
	5	367	869		
	6	368	856	*	{2, 3, 4, 5}
		Stage 6 (S = 6)			
{1, 2, 3, 4, 5, 6}	1	287	1143		
	2	316	1110		
	3	350	1003		
	4	359	927		
	5	346	897		
	6	347	884	*	{1, 2, 3, 4, 5}

The minimum value of the objective function is given by $f(6, N) = 884$. The optimal job sequence is reconstructed from the optimal policy function as follows. $P(6, \{1, 2, 3, 4, 5, 6\}) = \{1, 2, 3, 4, 5\}$, so $[6] = 6$. $P(5, \{1, 2, 3, 4, 5\}) = \{1, 2, 3, 4\}$, so $[5] = 5$; $P(4, \{1, 2, 3, 4\}) = \{1, 2, 3\}$, so $[4] = 4$. $P(3, \{1, 2, 3\}) = \{1, 3\}$, so $[3] = 2$. $P(2, \{1, 3\}) = \{3\}$, so $[2] = 1$. $P(1, \{3\}) = \emptyset$, so $[1] = 3$. Thus $\sigma^* = (3, 1, 2, 4, 5, 6)$.

all other jobs in J, whose return function $r(S, J, j)$ is to be calculated and among which the optimal action is to be identified. Thus the total number of subproblems solved by DP in order to find the optimal solution to the full problem is given by

$$Q(n) = \sum_{1 \leqslant S \leqslant n} C(n, S)S$$
$$= n2^{n-1}.$$

It is clear that $Q(n)$ increases exponentially with the number of jobs n in N. So, from a computational point of view, DP is not an efficient solution method for large problems. However, compared with complete enumeration which requires solving $n!$ problems, the amount of saving in computational efforts due to DP is tremendous. Thus given that n is not too large, DP provides a viable optimization method to solve the due-date sequencing problem.

CONCLUSIONS

We consider in this paper the single-machine sequencing problem with different due-dates. The objective is to find the optimal job sequence which minimizes the sum of absolute deviation of job completion times about their respective due-dates. We present a DP formulation of the problem and apply the principle of optimality of DP to find the optimal solution by implicit enumeration. A numerical example is provided to illustrate the working procedures of the DP solution method. Finally, we analyze the computational requirements of DP and note that it is a far more efficient optimization method than complete enumeration for solving the due-date sequencing problem.

Acknowledgement—This research was supported in part by a grant from the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

Ahmed M. U. and Sundararaghavan P. S. (1984) Minimizing the sum of absolute deviation of job completion times from their due dates in a single machine scheduling problem. *TIMS Southeast Chapter Meeting*, pp. 83–40.

Bagchi U., Sullivan R. S. and Chang Y. L. (1986) Minimizing mean absolute deviation of completion times about a common due date. *Nav. Res. Logist. Q.* **33**, 227–240.

Cheng T. C. E. (1987) Minimizing the average deviation of job completion times about a common due-date: an extension. *Mathl. Modelling* **9**, 13–15.

Cheng T. C. E. and Gupta M. C. (1989) Survey of scheduling research involving due-date determination decisions. *Eur. J. Opl. Res.* **38**, 156–166.

Dreyfus S. E. and Law A. M. (1977) *The Art and Theory of Dynamic Programming*. Academic Press, New York.

Emmons H. (1987) Scheduling to a common due date on parallel uniform processors. *Nav. Res. Logist. Q.* **34**, 803–810.

Garey M. R., Tarjan R. E. and Wilfong G. T. (1988) One-processor scheduling with symmetric earliness and tardiness penalties. *Maths Ops Res.* **13**, 330–348.

Gupta S. K. and Kyparisis J. (1987) Single machine scheduling research. *OMEGA* **13**, 207–227.

Hall N. G. (1986) Single- and multiple-processor models for minimizing completion variances. *Nav. Res. Logist. Q.* **33**, 49–54.

Hastings N. A. J. (1973) *Dynamic Programming with Management Applications*. Butterworths, London.

Held M. and Karp R. M. (1962) A dynamic programming approach to sequencing problems. *J. SIAM* **10**, 196–210.

Kanet J. J. (1981) Minimizing the average deviation of job completion times about a common due date. *Nav. Res. Logist. Q.* **28**, 643–651.

Lawler E. L. (1964) On scheduling problems with deferral costs. *Mgmt Sci.* **11**, 280–288.

Quaddus M. A. (1987) A generalized model of optimal due-date assignment by linear programming. *J. Opl. Res. Soc.* **38**, 353–359.

Ragatz G. L. and Mabert V. A. (1984) A framework for the study of due-date management in job shops. *Int. J. Prod. Res.* **22**, 685–695.

Sen T. and Gupta S. K. (1984) A state-of-the-art survey of static scheduling research involving due-dates. *OMEGA* **12**, 63–76.

Sundararaghavan P. S. and Ahmed M. U. (1984) Minimizing the sum of absolute lateness in single machine and multi machine scheduling. *Nav. Res. Logist. Q.* **31**, 325–333.