

A Hybrid Scheduling Algorithm for Multiclass Production Systems with Setup Times

Y. NARAHARI

Computer Science and Automation
Indian Institute of Science
Bangalore 560 012 - INDIA

R. SRIGOPAL

Computer Science and Automation
Indian Institute of Science
Bangalore 560 012 - INDIA

Abstract

In this article, we propose a hybrid algorithm for scheduling non-preemptive, single operation jobs in a multiclass production environment. The objective is to minimize the sum of the total weighted tardiness and setup costs of the schedule. We believe the problem is NP hard, and we use an efficient suboptimal algorithm based on Lagrangian relaxation and simulated annealing. Our algorithm works well for a variety of scheduling problems.

1 Introduction

The problem of scheduling arises in situations where scarce resources have to be optimally allocated to activities over time. Most scheduling problems belong to the class of NP hard combinatorial optimization problems. Any scheduling methodology should aim to [11]

1. generate efficiently near optimal solutions with measurable performance.
2. perform "what if" analysis to examine the impact of dynamic changes.
3. develop efficient methods for schedule reconfiguration to accommodate these changes.

In the area of discrete activity scheduling, it is generally accepted that a gap exists between scheduling theory and practice. Practical methods react to dynamic changes without the ability to produce good solutions and theoretical methods produce good schedules without the ability to react to dynamic changes. Recently, Luh *et al* [11] and Hoitomt *et al* [6, 7] have developed a Lagrangian relaxation based suboptimal algorithm for scheduling of non-preemptive single/multi operation jobs on parallel identical machines and for job shop scheduling. Their method

performs very well in a wide variety of scheduling situations and is also amenable for carrying out extensive "what-if" analysis. However, their methodology does not take into account setup times and setup costs that are very important in multiclass manufacturing system scheduling. Our present work attempts to extend the scheduling methodology to multiclass production systems comprising parallel identical machines and taking into account setup times and setup costs.

In a multiclass production setting, the jobs are divided into a number of mutually exclusive part types. Setup operations are an important feature of such production environments. A significant setup time is incurred when a machine changes from processing one type of parts to a different type of parts. The setup time generally includes times for fixturing tool changing and preparing the workplace. Thus, a setup cost is incurred, since the setup operations do not contribute to productivity. To minimize the setup times and costs, a batch of products belonging to the same part type is manufactured after a single setup. Large batch sizes on the other hand result in high inventory levels. The economic lot sizing problem (ELSP) [5] addresses this problem of minimizing the sum of inventory and setup costs. The problem is known to be NP hard [10].

The work reported in this paper is an extension of the work by Luh *et al* [11]. The method combines the techniques of *Lagrangian relaxation* and *simulated annealing*. The objective is to minimize the sum of the total weighted tardiness and setup costs (assumed to be a monotonically increasing function of the setup times).

The next section is a survey of relevant literature. It deals with the scheduling of jobs in a single class production environment as described by Luh *et al* [11]. The section summarises the integer programming formulation of the scheduling problem and the solution

methodology. It also briefly reviews optimization using simulated annealing. Section 3 proposes a hybrid methodology to a multiclass production setting with setup times included. Three examples are discussed to test the proposed methodology and detailed numerical results are provided. Section 4 presents conclusions and directions for future work.

2 A Review of Lagrangian Relaxation and Simulated Annealing

2.1 Lagrangian Relaxation

Lagrangian relaxation [2, 3, 4] provides an efficient way of scheduling independent jobs with due dates on identical parallel machines. The special integer programming formulation facilitates the application of the Lagrangian relaxation technique. Decomposition of the dual problem serves to simplify solution at the lower level. The high level problem is solved via a subgradient method. Dynamic changes can easily be accommodated in this approach. In this section, we provide a review of the Lagrangian relaxation technique as applied to scheduling of non-preemptive, single operation jobs on parallel identical machines. The material is mostly taken from [11].

2.1.1 Problem Formulation

An integer programming formulation as described in [11] is a common way to represent a scheduling problem. The following is a static, discrete time, integer programming formulation of the scheduling problem. We shall use the following notation.

- N total number of jobs.
- K time horizon under consideration.
- w_i weight if job i .
- t_i processing time of job i .
- D_i due date of job i .
- M_k number of machines available at time k (assumed to be monotonically nondecreasing in k).
- b_i beginning time of job i .
- c_i completion time of job i .
- δ_{ik} integer variable, equals 1 if job i is active at time k , and 0 otherwise.

T_i tardiness of job $i = \max(0, c_i - D_i)$.

J objective function to be minimized.

Among the above variables, the number of jobs N , time horizon K , weights of jobs $\{w_i\}_{i=1}^N$, time requirements $\{t_i\}_{i=1}^N$, due dates $\{D_i\}_{i=1}^N$ and machine availability $\{M_k\}_{k=1}^K$ are assumed to be given. Also the job processing is non-preemptive so that a contiguous block of time length t_i is needed to process job i . The decision variables are $\{b_i\}_{i=1}^N$. Once the b_i s are selected, $\{c_i\}_{i=1}^N$, $\{T_i\}_{i=1}^N$ and $\{\delta_{ik}\}_{i=1, k=1}^{N, K}$ can easily be derived. The objective function of interest is

$$J = \sum_i w_i T_i \quad (1)$$

A static and deterministic parallel machine scheduling problem can now be formulated as follows.

$$P : \min_{b_i} J = \sum_i w_i T_i \quad (2)$$

subject to capacity constraints

$$\sum_i \delta_{ik} \leq M_k \quad (k = 1, 2, \dots, K) \quad (3)$$

and processing time constraints

$$c_i - b_i + 1 = t_i \quad (i = 1, 2, \dots, N) \quad (4)$$

The single machine sequencing problem can be solved as a weighted bipartite matching problem that is NP hard [10]. Consequently, the parallel machine weighted tardiness problem is also NP hard. The additivity of the objective function facilitates the decomposition approach.

2.1.2 Solution Methodology

Relaxing the capacity constraints (3) using Lagrange multipliers π_k ($k = 1, 2, \dots, K$) to form the relaxed problem,

$$R : \min_{b_i} \left[\sum_i w_i T_i + \sum_k \pi_k \left(\sum_i \delta_{ik} - M_k \right) \right] \quad (5)$$

subject to (4), the dual problem is

$$D : \max_{\pi} L \quad (6)$$

with

$$L = - \sum_k \pi_k M_k + \min_{b_i} \sum_i \left(w_i T_i + \sum_k \pi_k \delta_{ik} \right) \quad (7)$$

subject to

$$\pi \geq 0; \quad (8)$$

This leads to the following decomposed subproblems for each job i (given π).

$$R_i : \min_{1 \leq b_i \leq K-t_i+1} L_i \quad (9)$$

with

$$L_i \equiv \left(w_i T_i + \sum_k \pi_k \delta_{ik} \right) \quad (10)$$

subject to

$$c_i - b_i + 1 = t_i \quad (11)$$

K is assumed to be large enough to complete all the jobs.

For convex programming problems, the maximum of the lag (dual cost) equals the minimum of the original objective function and a saddle point exists. However, there are several difficulties in utilizing this technique for solving discrete variable problems. First, the saddle point may or may not exist and it may be difficult to determine when the algorithm has terminated. Second, even if the dual optimum were obtained, the corresponding schedule at that point may not be feasible. Heuristic adjustment is generally required to ensure that the once relaxed constraints are obeyed. Therefore, the various steps to obtaining a near optimum solution are

1. solving the subproblems,
2. solving the dual problem,
3. constructing a feasible solution, and
4. finding a (sub) optimal solution.

Each of these steps is discussed in [11].

The optimized Lagrangian multipliers π_k are interpreted as a shadow price for using the resource (machine) at k . Therefore, they reflect the sensitivity of the objective function with respect to resource levels. This can be used to provide answers to "what if" questions and to reconfigure an existing schedule when changes occur in resource availability. Thus, Lagrangian relaxation has the ability to react effectively to dynamic changes and at the same time produce good suboptimal schedules.

2.2 Simulated Annealing

Simulated annealing [1, 8] is a powerful algorithm for solving (approximately) combinatorial optimization (CO) problems. Salient features of the algorithm

are its simplicity, generality and applicability to find high quality solutions. The algorithm is based on an intriguing combination of ideas from two different fields – statistical physics and CO. On the one hand, it can be viewed as an algorithm simulating the physical annealing process of solids to their minimum (ground) energy states. On the other hand, it can be considered as a generalization of local search algorithms which play an important role in CO problems. Simulated annealing is a randomization technique that can be mathematically described using Markov chains.

The simulated annealing algorithm starts off with a given initial solution and continuously tries to transform a current solution to its neighbours by applying a generating mechanism and an acceptance criterion. The acceptance criterion allows for deteriorations in a limited way. This is controlled by a control parameter that plays a similar role as temperature in the physical annealing process. Allowing deteriorations makes the simulated annealing algorithm more general than pure local search algorithms in which only strict improvements are allowed. The resulting effect is that the annealing algorithm can escape from a local minimum in order to arrive at a near optimum.

Thus, simulated annealing procedures present a new and promising approach to scheduling problems. A nominal schedule or a set of schedules is varied in a small and generally random way. A probability which is determined by the relative change in schedule cost and control parameter is assigned to the result. This probability is then used to determine which schedule/schedules becomes nominal for the next iteration. As in most other optimization problems there is some degree of enumeration and there is no way to measure the quality of the resulting schedule. The technique has been applied recently by Laarhoven *et al* [9] to job shop scheduling.

3 A Hybrid Scheduling Algorithm

In a multiclass production system, *switchover times* or *setup times* can have a significant effect on the way parts are scheduled. The jobs of a given part type need not be processed together. It is desired to find a schedule that minimizes the sum of the weighted tardiness and switchover costs.

First we determine a range of suitable values of switchovers using considerations such as:

- too many switchovers will make the setup costs dominate over the tardiness costs.

- too few switchovers will make the tardiness costs of delayed jobs substantial.

Several complications arise with the introduction of switchover times. The Lagrangian relaxation technique of Luh *et al* [11] cannot be directly applied because

- For every job j , we now need to evaluate L_{ij}^* , and b_{ij}^* , where i is the part type of the job that was processed immediately before j , ($j = 1, 2, \dots, N$); ($i = 1, 2, \dots, P$) where P is the total number of part types.
- Designing an effective greedy heuristic to arrive at near optimum feasible schedule at the termination of the subgradient algorithm is not easy.

The simple simulated annealing algorithm is not likely to yield good results for parallel machine scheduling because an efficient perturbation operator is not conceivable. To circumvent this, a hybrid approach that makes use of simulated annealing to arrive at a near optimal sequence of setup operations and Lagrangian relaxation to arrive at the schedule of jobs of a part type on the machines is developed. The assumption here is that once the machines are set up for a part type, all jobs belonging to the part type are processed. The following simplifying assumptions are made regarding switchover times and costs:

1. the switchover times are the same for all classes.
2. the setup costs depend only on the setup times and further, are a monotonically increasing function of the setup times.

Defining the state of a machine at time t to be the type of part it is processing at t , the extra data necessary are the initial states of the machines and the time instants at which each machine first becomes available.

First, we describe a method to arrive at the schedule of parts of a particular type on the machines. Let Q denote the total number of machines. An upper bound on the planning horizon K for scheduling jobs belonging to class $l = \sum_{i \in l} t_i$ (if all jobs are scheduled on a single machine). Let v_i denote the time machine i ($i = 1, 2, \dots, Q$) first becomes available (after necessary setup operations). Let the permutation (S_1, S_2, \dots, S_Q) denote the sequence of machines such that $v_{S_1} \leq v_{S_2} \leq \dots \leq v_{S_Q}$. Determine $q = j$ such that $\max_j [v_{S_j} \leq v_{S_1} + K]$; Machines S_{q+1}, \dots, S_Q cannot process any jobs belonging to the part type under consideration. For $k = 1, 2, \dots, K$, form M_k based on v_{S_1}, \dots, v_{S_n} , where $n = 1, 2, \dots, Q$. It is here that

the second assumption regarding setup costs becomes important. If two or more machines become available at the same time, any machine can be chosen for processing the jobs belonging to the part type thus preventing unnecessary enumeration at this stage. Use Lagrangian relaxation to arrive at the schedule of jobs and cost for each n . Each of these tasks is parallelizable. The schedule for which the sum of the setup cost and tardiness cost is minimum is chosen and the availability of the machines and states of the machines are accordingly updated.

To determine the order of the part types, higher level simulated annealing optimization is carried out. The simulated annealing process will give us the order in which to process the part types, taking into account the setup times and setup costs. Having obtained the order of part types, the schedule on each machine and cost is computed using the method discussed in the previous paragraph. It can easily be shown that in the global optimum schedule, jobs belonging to the same part type and having the same processing times and due dates have to be processed in the decreasing order of their weights. These can be reordered to yield a lower cost at the termination of the algorithm.

3.1 Numerical Results

The examples discussed here are adapted from the ones appearing in [11, 6, 7].

3.1.1 Example 1

There are 12 jobs belonging to 4 part types. They are to be scheduled on 2 machines that are available from time instant 1. Initial state of M_1 is given to be A and that of M_2 to be B.

Job Data for Example 1

i	w_i	t_i	D_i	class	i	w_i	t_i	D_i	class
1	2	4	41	A	2	2	4	41	A
3	2	4	61	A	4	2	2	71	B
5	2	2	46	B	6	2	3	31	C
7	2	3	31	C	8	2	3	31	C
9	2	3	36	C	10	2	3	56	C
11	2	1	26	D	12	2	1	61	D

Setup Times and Setup Costs

Job Class	A	B	C	D
Setup Time	40	20	30	10
Setup Cost	200	100	150	50

Schedule after 16 Iterations

M_1	1	2	3	11	12		
M_2	5	4	6	7	8	9	10

The cost of the above schedule is 274 units, out of which setup costs account for 200 units and tardiness costs equal 74 units.

3.1.2 Example 2

There are 25 jobs belonging to 7 part types. They are to be scheduled on 4 machines that are available from time 1. Initial state of M_1 is given to be A, M_2 is B, M_3 is C and M_4 is D.

The cost of the (suboptimal) schedule is 930 units. Tardiness cost is 130 units and the rest are setup costs.

Job Data for Example 2

i	w_i	t_i	D_i	cl	i	w_i	t_i	D_i	cl
1	6	4	21	A	2	2	4	21	A
3	5	4	101	A	4	2	5	61	B
5	8	5	101	B	6	2	8	61	C
7	2	8	41	C	8	5	8	76	C
9	2	8	126	C	10	1	2	61	D
11	2	2	20	D	12	2	2	66	D
13	1	2	101	D	14	6	2	126	D
15	2	6	126	E	16	2	7	61	F
17	2	7	126	F	18	2	7	176	F
19	2	7	76	F	20	2	7	101	F
21	2	7	101	F	22	2	7	151	F
23	2	7	151	F	24	2	3	176	G
25	2	3	76	G					

Setup Times and Setup Costs

Job Class	A	B	C	D	E	F	G
Setup Time	40	50	80	20	60	70	30
Setup Cost	200	250	400	100	300	350	150

Schedule after 49 Iterations

M_1	1	2	3	15			
M_2	4	5	24	25			
M_3	7	6	8	9			
M_4	10	11	12	13	14	16	19
	17	20	21	18	22	23	

3.1.3 Example 3

Eighty nine jobs belonging to 15 part types are to be scheduled on 10 machines. The first five machines are available from the beginning of the planning horizon and the next five are available from time instant 10. Initial states the machines 1, 2, ..., 10 are A, B, ..., J respectively.

The cost of the schedule is 4749 units out of which 1199 units are tardiness costs and the rest are setup costs.

Job Data for Example 3

i	w_i	t_i	D_i	cl	i	w_i	t_i	D_i	cl
1	1	1	1	A	2	9	1	121	A
3	1	1	131	A	4	5	1	11	A
5	1	1	11	A	6	1	1	11	A
7	1	1	-29	A	8	1	1	71	A
9	1	1	71	A	10	1	1	71	A
11	1	1	71	A	12	1	1	71	A
13	1	1	81	A	14	1	1	81	A
15	6	1	21	A	16	1	1	21	A
17	1	1	21	A	18	1	1	51	A
19	9	1	-9	A	20	1	1	-9	A
21	1	1	101	A	22	1	1	101	A
23	1	1	101	A	24	1	1	91	A
25	1	2	-19	B	26	1	2	151	B
27	1	2	151	B	28	1	2	1	B
29	1	2	1	B	30	1	2	131	B
31	1	2	-9	B	32	1	2	-9	B
33	1	2	-9	B	34	1	2	81	B
35	1	2	71	B	36	1	2	71	B
37	1	2	71	B	38	1	2	41	B
39	1	2	11	B	40	1	2	11	B
41	1	2	11	B	42	1	3	251	C
43	1	3	241	C	44	1	3	201	C
45	1	3	21	C	46	1	3	21	C
47	1	3	21	C	48	1	3	21	C
49	1	3	21	C	50	1	3	21	C
51	1	3	21	C	52	1	3	71	C
53	1	3	71	C	54	1	3	91	C
55	1	3	91	C	56	1	3	91	C
57	1	3	51	C	58	1	3	11	C
59	6	3	121	C	60	1	3	101	C
61	1	3	1	C	62	1	3	1	C
63	1	3	31	C	64	1	4	181	D
65	1	4	1	D	66	9	4	11	D
67	1	4	91	D	68	1	4	41	D
69	1	4	151	D	70	6	5	191	E
71	1	5	51	E	72	1	5	51	E
73	16	5	21	E	74	6	5	101	E
75	16	6	71	F	76	1	7	31	G
77	6	8	601	H	78	1	8	111	H
79	1	8	101	H	80	1	9	81	I
81	9	10	91	J	82	1	10	201	J
83	1	10	201	J	84	1	11	111	K
85	6	12	91	L	86	1	15	421	M
87	1	16	241	N	88	1	16	171	N
89	1	20	241	O					

Setup Times and Setup Costs

Class	A	B	C	D	E	F	G	H
Time	10	20	30	40	50	60	70	80
Cost	50	100	150	200	250	300	350	400
Class	I	J	K	L	M	N	O	
Time	90	100	110	120	150	160	200	
Cost	450	500	550	600	750	800	1000	

Schedule after 225 Iterations

M_1	84										
M_2	25	28	29	31	32	33	39	40	41	35	36
	38	37	34	27	30	26					
M_3	61	62	58	48	49	50	45	51	46	47	63
	53	54	55	56	57	43	60	44	42	52	59
M_4	65	66	68	67	69	64	87	88			
M_5	71	72	74	73	70	89					
M_6	75	19	15	4	5	6	20	17	7	1	18
	16	24	8	22	3	13	10	12	9	2	21
	11	23	14								
M_7	76	85									
M_8	78	77	79								
M_9	80	86									
M_{10}	82	81	83								

4 Conclusions

In this work, a new hybrid scheduling algorithm that uses simulated annealing and Lagrangian relaxation has been proposed and tested for multiclass production systems consisting of identical parallel machines. The technique is found to work very well for many examples. However, the two key issues

1. performance evaluation and
2. schedule reconfiguration in the event of dynamic changes

have not been answered. Future work should concentrate on

1. answering questions regarding performance evaluation and schedule reconfiguration in the event of dynamic changes for multiclass production systems.
2. extending the hybrid technique to job shop scheduling.

References

- [1] E. H. L. Aarts and P. J. M. Van Laarhoven, Statistical cooling — A general approach to solve combinatorial optimization problems, *Philips Journal of Research*, 5, 1985, pp 193–226.
- [2] M. L. Fischer, Optimal solution of scheduling problems using Lagrange multipliers, *Operations Research*, 21, 1973, pp 1114–1127.
- [3] M. L. Fischer, Lagrangian relaxation methods for solving integer programming problems, *Management Science*, 27, 1981, pp 1–18.
- [4] M. L. Fischer, A dual algorithm for the one machine scheduling problem, *Mathematical Programming*, 11, 1976, pp 229–251.
- [5] B. Fleishmann, The discrete lot sizing and scheduling problem, *European Journal of Operations Research*, 44, 1990, pp 337–348.
- [6] D. J. Hoitomt, P. B. Luh, E. Max, and K. R. Pattipati, Scheduling jobs with simple precedence constraints on parallel machines, *IEEE Control Systems Magazine*, 10, February 1990, pp 34–40.
- [7] D. J. Hoitomt, P. B. Luh, and K. R. Pattipati, A practical approach to job shop scheduling problems, *IEEE Transactions on Robotics and Automation*, 1993, 9, pp 1–13.
- [8] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, Optimization by simulated annealing, *Science*, 220, 1983, pp 671–680.
- [9] J. M. Peter Van Laarhoven, E. H. L. Aarts, and J. K. Lenstra, Job shop scheduling by simulated annealing, *Operations Research*, 40, 1992, pp 113–125.
- [10] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnoy Kan, and D. B. Shmoys, Sequencing and scheduling — algorithms and complexity, *Logistics of Production and Inventory*, Volume 4, edited by S. C. Graves, A. H. G. Rinnoy Kan, and P. Zipkin, 1989.
- [11] P. B. Luh, D. J. Hoitomt, K. R. Pattipati, and Eric Max, Schedule generation and reconfiguration for parallel machines, *IEEE Transactions on Robotics and Automation*, 6, December 1990, pp 687–696.