stichting
mathematisch
centrum

MC

AFDELING MATHEMATISCHE BESLISKUNDE BW 145/81 AUGUSTUS
(DEPARTMENT OF OPERATIONS RESEARCH)

M.L. FISHER, B.J. LAGEWEG,
J.K. LENSTRA, A.H.G. RINNOOY KAN

SURROGATE DUALITY RELAXATION FOR JOB SHOP SCHEDULING

Preprint

kruislaan 413 1098 SJ amsterdam

Printed at the Mathematical Centre, 413 Kruislaan, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

1980 Mathematics Subject Classification: 90B35, 90C10

SURROGATE DUALITY RELAXATION FOR JOB SHOP SCHEDULING

M.L. FISHER

The Wharton School, University of Pennsylvania, Philadelphia

B.J. LAGEWEG

Mathematisch Centrum, Amsterdam

J.K. LENSTRA

Mathematisch Centrum, Amsterdam

A.H.G. RINNOOY KAN

Erasmus University, Rotterdam

ABSTRACT

Surrogate duality bounds for the job shop scheduling problem are obtained by replacing certain constraints by their weighted sum and strengthening the aggregate constraint by iterating over all possible weights. The constraints successively considered for this purpose are the capacity constraints on the machines and the precedence constraints determining the machine order for each job. The resulting relaxations are investigated from a theoretical and a computational point of view.

KEY WORDS & PHRASES: *job shop scheduling, branch-and-bound, surrogate duality relaxation, computational experience.*

NOTE: This report has been submitted for publication in a journal.

1. INTRODUCTION

In the general job shop problem, n *jobs* have to be scheduled on m *machines*, subject to both *capacity constraints* expressing that each machine can handle at most one job at a time, and *precedence constraints* reflecting a specified processing order through the machines for the operations of each job. The objective is to find a schedule that minimizes the maximum of the job completion times.

This problem is well known to pose a formidable computational challenge. As shown in Table 1, only two very special cases are solvable in polynomial time and their immediate generalizations are NP-hard. This implies that optimization algorithms have probably to be based on some form of implicit enumeration. We refer to [Lageweg *et al.* 1977] for a review of such attempts and in particular for a systematic survey of the lower bounds proposed for use in branch-and-bound algorithms.

The purpose of this paper is to investigate various *surrogate duality relaxations* [Glover 1968, 1975; Dyer 1980] of the job shop problem. Such a relaxation is obtained by starting from a mathematical programming formulation of the problem and replacing certain constraints by a weighted linear combination of them. This procedure should be carried out in such a way that the relaxed problem is easily solvable and yields a strong lower bound on the optimum solution value for an appropriate choice of weights.

number of machines	number of operations per job	processing time per operation	complexity	reference
2	≤ 2	arbitrary	O(n log n)	[Jackson 1956]
2	arbitrary	arbitrary	unary NP-hard	[Garey *et al.* 1976]
2	≤ 3	arbitrary	binary NP-hard	[Lenstra *et al.* 1977;
3	≤ 2	arbitrary	binary NP-hard	Gonzalez & Sahni 1978]
2	arbitrary	1	O(n log n)	[Hefetz & Adiri 1979]
2	arbitrary	1 or 2	unary NP-hard	[Lenstra & Rinnooy Kan 1979]
3	arbitrary	1	unary NP-hard	[Lenstra & Rinnooy Kan 1979]

Table 1. Summary of complexity results.

Surrogate duality relaxation is closely related to *Lagrangean relaxation* [Geoffrion 1974; Fisher *et al.* 1975; Fisher 1981], in which the weighted and aggregated constraints are removed and added to the objective function. This approach has been applied with great success to several notorious combinatorial optimization problems, such as the traveling salesman problem [Held & Karp 1971], a job shop problem with min-sum objective [Fisher 1973], the generalized assignment problem [Ross & Soland 1975], a single machine scheduling problem [Fisher 1976], and the plant location problem [Cornuejols *et al.* 1977]. It is easily verified, however, that for fixed weights the surrogate duality bound is at least as strong as the Lagrangean bound. Fortunately, in the job shop case the former bound is also as easy to calculate.

Two mathematical programming formulations of the job shop problem are given in Section 2. Surrogate duality relaxations of the capacity constraints and of the precedence constraints are considered in Sections 3 and 4 respectively. Computational experience is presented in Section 5 and some concluding remarks are contained in Section 6.

2. MATHEMATICAL PROGRAMMING MODELS

We shall denote the n *jobs* by $J_1,\ldots,J_n$ and the m *machines* by $M_1,\ldots,M_m$. Job J_j consists of a chain of *operations* $(O_{m_{j-1}+1},\ldots,O_{m_j})$ $(j = 1,\ldots,n)$, where $0 = m_0 < m_1 < \ldots < m_n$. Operation O_u requires a given uninterrupted *processing time* of p_u time units on a given machine M_{μ_u} $(u = 1,\ldots,m_n)$; O_u has to be completed on M_{μ_u} before O_{u+1} can start on $M_{\mu_{u+1}}$ $(u = m_{j-1}+1,\ldots,m_j-1;\ j = 1,\ldots,n)$. Machine M_i can handle no more than one operation at a time $(i = 1,\ldots,m)$.

The problem is to minimize the *maximum completion time* z subject to

(a) the requirement that all operations are performed in the interval $[0,z]$,

(b) the *precedence constraints* among the operations of each job, and

(c) the *capacity constraints* on the machines.

Let x_u denote the *starting time* of operation O_u in a given schedule, and let $x = (x_1,\ldots,x_{m_n})$. Conditions (a) and (b) can now be formulated as

$$x \in X(z)$$

where X(z) is the set of all vectors x for which

$$\begin{aligned} x_u+p_u-x_{u+1} &\le 0 \quad (u = m_{j-1}+1,\ldots,m_j-1;\ j = 1,\ldots,n), \\ x_u+p_u-z &\le 0 \quad (u = 1,\ldots,m_n), \\ x_u &\ge 0 \quad (u = 1,\ldots,m_n). \end{aligned} \tag{1}$$

Condition (c) allows several mathematical formulations. The traditional way [Manne 1960] is as follows. Let T be a given upper bound on the length of an optimal schedule, and for each ordered pair (O_u,O_v) such that $\mu_u = \mu_v$ let y_{uv} be a 0-1 variable with

$$y_{uv} = \begin{cases} 0 & \text{if } O_u \text{ precedes } O_v, \\ 1 & \text{if } O_v \text{ precedes } O_u. \end{cases}$$

Condition (c) can now be formulated as

$$\left.\begin{aligned} x_u+p_u-x_v-Ty_{uv} &\le 0 \\ y_{uv}+y_{vu} &= 1 \\ y_{uv} &\in \{0,1\} \end{aligned}\right\} \text{ for all } (u,v) \text{ with } \mu_u = \mu_v. \tag{2}$$

In an alternative approach [Fisher 1973], let $n_{it}(x)$ denote the number of operations performed on machine M_i at time t for a vector x of starting times:

$$n_{it}(x) = |\{O_u | \mu_u = i, t-p_u < x_u \le t\}|.$$

Condition (c) now amounts to

$$n_{it}(x)-1 \le 0 \quad (i = 1,\ldots,m;\ t = 1,\ldots,T). \tag{3}$$

The aggregation of capacity constraints (2) and (3) will be investigated in Section 3 and the aggregation of precedence constraints (1) in Section 4. We will treat the left-hand side of the aggregate constraint as a new objective function and ask for the smallest z such that its minimum subject to the other constraints is nonpositive. This is obviously equivalent to minimizing z subject to the condition that the relaxed constraints allow a feasible solution.

3. RELAXATION OF THE CAPACITY CONSTRAINTS

We start by considering the first formulation of the capacity constraints. We assign nonnegative weights α_{uv} to the constraints (2) and aggregate them to a single linear constraint. The relaxed problem is then to minimize z subject to

$$x \in X(z),$$

$$\textstyle\sum_{\{u,v\}:\mu_u=\mu_v} [\alpha_{uv}(x_u+p_u-x_v)+\alpha_{vu}(x_v+p_v-x_u)-\alpha_{uv}Ty_{uv}-\alpha_{vu}Ty_{vu}] \le 0, \qquad (2')$$

$$\left.\begin{array}{l} y_{uv}+y_{vu} = 1 \\ y_{uv} \in \{0,1\} \end{array}\right\} \text{ for all } (u,v) \text{ with } \mu_u = \mu_v.$$

For given weights α_{uv} and a given objective value z, let $A(\alpha,z)$ denote the minimum value of the left-hand side of (2') subject to the other constraints. The relaxed problem is then equivalent to finding the smallest z such that $A(\alpha,z) \le 0$.

Consider any pair $\{u,v\}$ with $\mu_u = \mu_v$, and suppose that $\alpha_{uv} \ge \alpha_{vu}$. This assumption implies that there is an optimal solution with $y_{uv} = 1$ and $y_{vu} = 0$. The contribution of $\{u,v\}$ to $A(\alpha,z)$ can now be rewritten as

$$(\alpha_{uv}-\alpha_{vu})(x_u-x_v)+\alpha_{uv}p_u+\alpha_{vu}p_v-\alpha_{uv}T.$$

Since $x_u-x_v \le T-p_u$ and $p_v \le T-p_u$, this contribution is at most equal to

$$(\alpha_{uv}-\alpha_{vu})(T-p_u)+\alpha_{vu}(T-p_u)-\alpha_{uv}(T-p_u) = 0.$$

A similar argument applies if $\alpha_{uv} < \alpha_{vu}$. It follows that $A(\alpha,z) \le 0$.

The smallest z such that $A(\alpha,z) \le 0$ is therefore equal to the smallest $\bar{z}$ for which $X(\bar{z}) \ne \emptyset$, *i.e.*,

$$\bar{z} = \max_j\{\textstyle\sum_{u=m_{j-1}+1}^{m_j} p_u\}.$$

This is a familiar [Charlton & Death 1970; Schrage 1970] and extremely weak lower bound on the length of an optimal schedule.

We therefore turn to the second formulation of the capacity constraints. We assign nonnegative weights β_{it} to the constraints (3) and aggregate them to obtain

$$\sum_{i=1}^{m} \sum_{t=1}^{T} \beta_{it}(n_{it}(x)-1) \le 0. \qquad (3')$$

For given β and z, let $B(\beta,z)$ denote the minimum value of the left-hand side of (3') subject to $x \in X(z)$. As before, we ask for the smallest integer z such that $B(\beta,z) \le 0$.

Since $X(z) \subseteq X(z+1)$, $B(\beta,z)$ is nonincreasing in z. For $z < \bar{z}$, we have $X(z) = \emptyset$ and hence $B(\beta,z) = \infty$.

Calculation of $B(\beta,z)$ for fixed β and all $z \in \{\bar{z},...,T\}$ is carried out by means of *dynamic programming*, as in [Fisher 1976]. Since the contribution of an operation to $B(\beta,z)$ is equal to the sum of the weights associated with its machine over all time units of its execution, $B(\beta,z)$ can be rewritten as

$$B(\beta,z) = \sum_{j=1}^{n} B_j(\beta,z) - \sum_{i=1}^{m} \sum_{t=1}^{T} \beta_{it}$$

where

$$B_j(\beta,z) = \min_{x \in X(z)} \{\sum_{u=m_{j-1}+1}^{m_j} \sum_{t=x_u+1}^{x_u+p_u} \beta_{\mu_u t}\}.$$

Consider a fixed job J_j. For all $u \in \{m_{j-1}+1,...,m_j\}$, let $b_u(z)$ denote the minimum cost of performing the operations $O_{m_{j-1}+1},...,O_u$ in the interval $[0,z]$. Due to constraint (1), O_u has to be performed in the interval $[r_u, T-q_u]$, where

$$r_u = \sum_{v=m_{j-1}+1}^{u-1} p_v, \quad q_u = \sum_{v=u+1}^{m_j} p_v. \qquad (4)$$

Hence, $b_u(z)$ can be calculated by the following recursion:

$$b_{m_{j-1}}(z) = 0 \qquad (z = 0,...,T-q_{m_{j-1}});$$

$$b_u(z) = \begin{cases} \infty & (z = 0,...,r_u+p_u-1) \\ \min\{b_u(z-1), b_{u-1}(z-p_u) + \sum_{t=z-p_u+1}^{z} \beta_{\mu_u t}\} & (z = r_u+p_u,...,T-q_u) \end{cases}$$

$$(u = m_{j-1}+1,...,m_j).$$

We now have

$$B_j(\beta,z) = b_{m_j}(z).$$

It follows that $B(\beta,\bar{z}),...,B(\beta,T)$ can be calculated in $O(m_n T)$ time.

For fixed β, the smallest z such that $B(\beta,z) \leq 0$ yields a valid lower bound $\hat{z}$ with an associated vector $\hat{x}$ of starting times. By maximizing over all possible choices of β, we may improve on this bound. That is, we try to find new weights β' for which $B(\beta',\hat{z}) > 0$, calculate a new lower bound $\hat{z}'$ as the smallest z such that $B(\beta',z) \leq 0$, and repeat.

There are various procedures to obtain β' from β, similar to the *sub-gradient optimization* techniques that are used in the context of Lagrangean relaxation. Here, however, we are only concerned with the *sign* of $B(\beta,z)$ and hence the lower bound is invariant under scalar multiplication of β. Under these circumstances, it can be proved [Minsky & Papert 1969] that the iteration scheme defined by

$$\beta'_{it} = \max\{0, \beta_{it} + \lambda(n_{it}(\hat{x}) - 1)\}$$

for any constant step size $\lambda > 0$ will converge to a β' for which $B(\beta',\hat{z}) > 0$, if such a β' exists. Other iteration procedures are possible as well; we refer to [Lageweg 1982] for details.

We note that β can be initiated in such a way that $\hat{z}$ closely approximates the job shop bound proposed in [Bratley *et al.* 1973], reputedly the best one currently available. More precisely, the latter bound is equal to the maximum solution value over m single machine problems, and when these problems are relaxed by allowing preemption, the resulting bound will be no larger than $\hat{z}$ for a certain choice of β. Again, we refer to [Lageweg 1982] for details.

4. RELAXATION OF THE PRECEDENCE CONSTRAINTS

We next investigate the relaxation of the precedence constraints. We assign nonnegative weights γ_u to the constraints (1) and aggregate them. The resulting problem is to minimize z subject to

$$\sum_{j=1}^{n} \sum_{u=m_{j-1}+1}^{m_j-1} \gamma_u (x_u+p_u-x_{u+1}) \le 0, \tag{1'}$$

$$0 \le x_u \le z-p_u \qquad (u = 1,\dots,m_n), \tag{5}$$

$$\text{the capacity constraint on } M_i \quad (i = 1,\dots,m). \tag{6}$$

For given γ and z, let $C(\gamma,z)$ denote the minimum value of the left-hand side of (1') subject to (5) and (6). The relaxed problem is again equivalent to finding the smallest integer z such that $C(\gamma,z) \le 0$.

It is easily verified that $C(\gamma,z)$ is nonincreasing in z and that $C(\gamma,z) = \infty$ for $z < \bar{\bar{z}}$, where

$$\bar{\bar{z}} = \max_i \{\textstyle\sum_{\mu_u=i} p_u\}.$$

Calculation of $C(\gamma,z)$ for fixed γ and $z \ge \bar{\bar{z}}$ requires the solution of m separate single machine problems:

$$C(\gamma,z) = \sum_{i=1}^{m} C_i(\gamma,z) + \sum_{j=1}^{n} \sum_{u=m_{j-1}+1}^{m_j-1} \gamma_u p_u$$

where

$$C_i(\gamma,z) = \min_{(5),(6)} \{\textstyle\sum_{\mu_u=i} \gamma'_u x_u\}$$

with

$$\gamma'_u = \begin{cases} \gamma_u & (u = m_{j-1}+1;\ j = 1,\dots,n), \\ -\gamma_{u-1} & (u = m_j;\ j = 1,\dots,n), \\ \gamma_u-\gamma_{u-1} & (\text{otherwise}). \end{cases}$$

$C_i(\gamma,z)$ can be calculated by a simple generalization of *Smith's rule* [Smith 1956]: schedule the operations in order of nonincreasing ratios γ'_u/p_u, with the positively weighted operations assigned to an interval starting at time 0 and the negatively weighted ones assigned to an interval finishing at time z. It follows that $C(\gamma,z)$ can be calculated in $O(\sum_{i=1}^{m} n_i \log n_i)$ time, where n_i

is the number of operations performed on M_i $(i = 1,\dots,m)$.

It also follows that, for $z \geq \bar{\bar{z}}$, $C(\gamma,z)$ is a linear function of z:

$$C(\gamma,z) = c+c'z \tag{7}$$

for some constant c and with c' equal to the sum of the negative weights γ'_u. Hence, the smallest integer z such that $C(\gamma,z) \leq 0$ is given by $\hat{z} = \lceil -c/c' \rceil$. This observation allows a comparison between the above approach and Lagrangean relaxation, in which constraint (1') is removed from the problem and its left-hand side is added to the objective function. The Lagrangean bound is given by $\lceil \min_z \{c+(1+c')z\} \rceil$. This value is no larger than $\hat{z}$, and both bounds are equal if the weights are normalized such that $c' = -1$.

We may improve on the lower bound $\hat{z}$ by applying standard subgradient optimization techniques to the Lagrangean problem subject to the normalization constraint on the weights, or by using one of the iteration schemes referred to in the previous section. In both cases, we can make use of the property that the optimal scheduling order determined by Smith's rule does not change as long as the ordering of the ratios γ'_u/p_u remains the same. The latter condition defines linear constraints in weight space demarcating regions that can be traversed in a single step. Unfortunately, serious degeneracy occurs at the border of these regions and a BOXSTEP-like iteration [Marsten *et al.* 1975] would be required to continue from there.

Initial computational experiments indicated that the resulting lower bound would be very weak. This is not surprising, since the precedence constraints (1) are poorly represented by the aggregate constraint (1'). We can obtain a better bound by taking certain implications of (1) explicitly into account in the calculation of $C(\gamma,z)$. In particular, (1) implies that each operation O_u has to be performed between a *release date* r_u and a *deadline* $z-q_u$ (*cf.* (4)), so that (5) can be replaced by

$$r_u \leq x_u \leq z-q_u-p_u \quad (u = 1,\dots,m_n). \tag{5'}$$

To analyze the complexity of the improved lower bound computation, let us distinguish between two types of single machine problems:

- the *feasibility* problem: determine whether $C_i(\gamma,z)$ is finite;
- the *minimization* problem: calculate $C_i(\gamma,z)$.

If we impose the additional constraints (5'), then the feasibility problem and *a fortiori* the minimization problem become unary NP-hard [Lenstra *et al.* 1977]. If, in addition, we allow *preemption* (job splitting), then the feasibility problem can be solved in O(n log n) time [Lageweg *et al.* 1976], but the minimization problem is still unary NP-hard [Labetoulle *et al.* 1979]. In both cases, the linearity property expressed by (7) is lost. It is worth observing that these statements are still true if the release dates are respected only for the positively weighted operations and the deadlines only for the negatively weighted operations.

The strongest possible bound arising from this discussion, incorporating release dates and deadlines and not allowing preemption, dominates the bound due to [Bratley *et al.* 1973] mentioned in the previous section. In fact, the latter bound is equal to the smallest z such that $C(\gamma,z)$ is finite. The computation of either bound requires the solution of several instances of an NP-hard single machine problem, but in view of their small size that is not necessarily disastrous.

5. COMPUTATIONAL EXPERIENCE

We shall restrict ourselves to reviewing several attempts to solve the notorious 10-job 10-machine 100-operation problem from [Fisher & Thompson 1963]. This benchmark problem is not known to have been solved to optimality. We feel that, in spite of this limitation, the results as summarized in Table 2 give a fair representation of the qualities of the four algorithms in question. Below, we briefly describe these algorithms and comment on the results obtained.

All four algorithms are branch-and-bound methods and use a *branching scheme* that generates all *active schedules* [Brooks & White 1965; Florian *et al.* 1971].

Algorithm 1 is the best job shop algorithm published so far. The *lower bound* is given by the maximum solution value over m single machine problems; each of these problems is based on the relaxation of the capacity constraints on all machines but one and is equivalent to minimizing maximum lateness subject to release dates. This bound was proposed in [Bratley *et al.* 1973]; a very efficient method for its computation was given in [McMahon & Florian 1975]. The *search strategy* selects a node with minimum lower bound for further examination. Algorithm 1 was able to solve the 20-job 5-machine 100-operation problem from [Fisher & Thompson 1963] to optimality; it generated 2259 nodes and required 152 CPU seconds. Table 2 gives the results for the 10×10 problem.

Algorithm 2 is an improved implementation of Algorithm 1. Noteworthy points of difference are the following:

Algorithm Reference	1 [McMahon & Florian 1975]	2 [Lageweg 1982]	3 Section 3	4 Section 4
First lower bound	808	808	813	808
Best solution	972	935	1084	1084
Number of nodes	26692	119344	1	1
CPU seconds	698	512	700	1024
Language	FORTRAN	PASCAL	PASCAL	PASCAL
Computer	Cyber 74	Cyber 170-750	Cyber 73-28	Cyber 73-28

Table 2. Results for 10×10 problem.

- *lower bound*: The single machine algorithm from [McMahon & Florian 1975] has been modified as described in [Lageweg *et al.* 1976] and is applied only to a limited number of promising machines, for which a weaker single machine bound takes on the highest values.
- *search strategy*: A recursive search strategy is employed, which is adaptive in the sense that, when a good solution is prespecified or a better one obtained, it starts looking around for improvements in the neighborhood of that solution.
- *upper bound*: A heuristic from [Lageweg *et al.* 1977] tries to find a better solution at four equidistant levels of the search tree.

Further details will be provided in [Lageweg 1982]. Algorithm 2 solved the 20×5 problem after generating 1696 nodes in 26 CPU seconds. Table 2 reports on an application to the 10×10 problem given a solution of value 936; the latter solution was obtained by an alternating sequence of improvements by computer and adjustments by hand.

After this, the results obtained with Algorithms 3 and 4 incorporating the bounds developed in Sections 3 and 4 are somewhat disappointing. Both algorithms apply a weight iteration scheme proposed in [Shor 1968] to increase the *lower bound* and use the heuristic method mentioned above to decrease the global *upper bound*. Table 2 gives the results for the 10×10 problem with respect to the root node of the search tree.

Algorithm 3 implements the second relaxation of the capacity constraints. After 84 ascents, we obtained a lower bound of value 813, which represents at least an improvement over the McMahon-Florian bound. However, the investments in computer time involved did not encourage us to carry on the search beyond the root. In an effort to speed up the computations, we tried to decrease the problem size by scaling the processing times or by allowing nonzero multipliers β_{it} only for certain equidistant values of t, to incorporate second order information in the weight iteration scheme, and to use the final weights from the root node at some or all levels of the tree. None of these attempts was very successful. Another idea might be to use a fully polynomial approximation scheme in solving the relaxed problem during the initial iterations.

Algorithm 4 implements the strongest possible bound resulting from the relaxation of the precedence constraints. For a given choice of weights, this requires the solution of m instances of an NP-hard single machine problem, *viz.* minimizing total weighted completion time subject to release dates and deadlines. In the absence of an efficient method to solve this problem, we resorted to brute force techniques and, for the initial choice of weights, obtained a lower bound of value 808. Previous experiments indicated that the other bounds of this type, disregarding release dates or deadlines or allowing preemption, are rather weak.

6. CONCLUDING REMARKS

The results presented in this paper are so far primarily of theoretical interest. Although our bounds dominate those proposed before, the time required for their computation prohibits the solution of problems of a reasonable size. This confirms once more the inherent intractability of the job shop scheduling problem, and we repeat the all too familiar words of [Conway *et al.* 1967]: "Many proficient people have considered this problem, and all have come away essentially empty-handed. Since this frustration is not reported in the literature, the problem continues to attract investigators who just cannot believe that a problem so simply structured can be so difficult until they have tried it." At least the latter sentence of this quotation has lost some of its validity.

It should be mentioned that the applicability of surrogate duality relaxation to combinatorial optimization problems is of interest by itself. The property that the resulting bounds are superior to those obtained by Lagrangean relaxation might render surrogate duality a promising approach to other problems in the area as well.

ACKNOWLEDGMENTS

The authors gratefully acknowledge stimulating discussions with M.A.H. Dempster. This research was partially supported by NSF Grant ENG-7826500 to the University of Pennsylvania, by NATO Special Research Grant 9.2.02 (SRG.7), and by NATO Research Grant 1575.

REFERENCES

P. BRATLEY, M. FLORIAN, P. ROBILLARD (1973) On sequencing with earliest starts and due dates with application to computing bounds for the ($n/m/G/F_{max}$) problem. *Naval Res. Logist. Quart.* 20,57-67.

G.H. BROOKS, C.R. WHITE (1965) An algorithm for finding optimal or near optimal solutions to the production scheduling problem. *J. Indust. Engrg.* 16,34-40.

J.M. CHARLTON, C.C. DEATH (1970) A method of solution for general machine scheduling problems. *Oper. Res.* 18,689-707.

R.W. CONWAY, W.L. MAXWELL, L.W. MILLER (1967) *Theory of Scheduling*, Addison-Wesley, Reading, MA.

G. CORNUEJOLS, M.L. FISHER, G.L. NEMHAUSER (1977) Location of bank accounts to optimize float: an analytic study of exact and approximate algorithms. *Management Sci.* 23,789-810.

M.E. DYER (1980) Calculating surrogate constraints. *Math. Programming* 19, 255-278.

H. FISHER, G.L. THOMPSON (1963) Probabilistic learning combinations of local job-shop scheduling rules. In: J.F. MUTH, G.L. THOMPSON (eds.) (1963) *Industrial Scheduling*, Prentice-Hall, Englewood Cliffs, NJ, 225-251.

M.L. FISHER (1973) Optimal solution of scheduling problems using Lagrange multipliers, part I. *Oper. Res.* 21,1114-1127.

M.L. FISHER (1976) A dual algorithm for the one-machine scheduling problem. *Math. Programming* 11,229-251.

M.L. FISHER (1981) The Lagrangian relaxation method for solving integer programming problems. *Management Sci.* 27,1-18.

M.L. FISHER, W.D. NORTHUP, J.F. SHAPIRO (1975) Using duality to solve discrete optimization problems: theory and computational experience. *Math. Programming Stud.* 3,56-94.

M. FLORIAN, P. TRÉPANT, G. McMAHON (1971) An implicit enumeration algorithm for the machine sequencing problem. *Management Sci.* 17,B782-792.

M.R. GAREY, D.S. JOHNSON, R. SETHI (1976) The complexity of flowshop and jobshop scheduling. *Math. Oper. Res.* 1,117-129.

A.M. GEOFFRION (1974) Lagrangean relaxation for integer programming. *Math. Programming Stud.* 2,82-114.

F. GLOVER (1968) Surrogate constraints. *Oper. Res.* 16,741-749.

F. GLOVER (1975) Surrogate constraint duality in mathematical programming. *Oper. Res.* 23,434-451.

T. GONZALEZ, S. SAHNI (1978) Flowshop and jobshop schedules: complexity and approximation. *Oper. Res.* 26,36-52.

N. HEFETZ, I. ADIRI (1979) An efficient optimal algorithm for the two-machines, unit-time, job-shop, schedule-length, problem. Operations Research, Statistics and Economics Mimeograph Series 237, Technion, Haifa.

M. HELD, R.M. KARP (1971) The traveling-salesman problem and minimum spanning trees: part II. *Math. Programming* 1,6-25.

J.R. JACKSON (1956) An extension of Johnson's results on job lot scheduling. *Naval Res. Logist. Quart.* 3,201-203.

J. LABETOULLE, E.L. LAWLER, J.K. LENSTRA, A.H.G. RINNOOY KAN (1979) Preemptive scheduling of uniform machines subject to release dates. Report BW 99, Mathematisch Centrum, Amsterdam.

B.J. LAGEWEG (1982) *Combinatorial Planning Models*, Mathematisch Centrum, Amsterdam, to appear.

B.J. LAGEWEG, J.K. LENSTRA, A.H.G. RINNOOY KAN (1976) Minimizing maximum lateness on one machine: computational experience and some applications. *Statist. Neerlandica* 30,25-41.

B.J. LAGEWEG, J.K. LENSTRA, A.H.G. RINNOOY KAN (1977) Job-shop scheduling by implicit enumeration. *Management Sci.* 24,441-450.

J.K. LENSTRA, A.H.G. RINNOOY KAN (1979) Computational complexity of discrete optimization problems. *Ann. Discrete Math.* 4,121-140.

J.K. LENSTRA, A.H.G. RINNOOY KAN, P. BRUCKER (1977) Complexity of machine scheduling problems. *Ann. Discrete Math.* 1,343-362.

A.S. MANNE (1960) On the job shop scheduling problem. *Oper. Res.* 8,219-223.

R.E. MARSTEN, W.W. HOGAN, J.W. BLANKENSHIP (1975) The BOXSTEP method for large scale optimization. *Oper. Res.* 23,389-405.

G. McMAHON, M. FLORIAN (1975) On scheduling with ready times and due dates to minimize maximum lateness. *Oper. Res.* 23,475-482.

M. MINSKY, S. PAPERT (1969) *Perceptions: an Introduction to Computational Geometry*, MIT Press, Cambridge, MA.

G.T. ROSS, R.M. SOLAND (1975) A branch and bound algorithm for the generalized assignment problem. *Math. Programming* 8,91-103.

L. SCHRAGE (1970) Solving resource-constrained network problems by implicit enumeration - nonpreemptive case. *Oper. Res.* 18,263-278.

N.Z. SHOR (1968) The rate of convergence of the generalized gradient descent method. *Cybernetics* 4.3,79-80.

W.E. SMITH (1956) Various optimizers for single-stage production. *Naval Res. Logist. Quart.* 3,59-66.