TILBURG UNIVERSITY
KATHOLIEKE UNIVERSITEIT BRABANT
POSTBOX 90153
5000 LE TILBURG
THE NETHERLANDS

[redacted]

DEPARTMENT OF ECONOMICS

RESEARCH MEMORANDUM

AN O(nlogn) ALGORITHM FOR THE TWO-MACHINE FLOW SHOP PROBLEM WITH CONTROLLABLE MACHINE SPEEDS

C.P.M. van Hoesel

FEW 475

An $O(n\log n)$ algorithm for the two-machine flow shop problem with controllable machine speeds

C.P.M. van Hoesel

Abstract.

An algorithm is developed to solve the two-machine flow shop problem, if machine speeds may vary. This algorithm makes use of an elementary dominance relation to obtain the $O(n\log n)$ running time, which is an improvement on previously developed algorithms. Moreover it is shown that faster algorithms are not possible, even in the case with fixed machine speeds.

1. Introduction

Classical research on machine scheduling concentrates on the issue of sequencing. In this paper we treat a scheduling problem in which, next to the permutation of jobs on the machines also the speeds of the machines are controllable. In particular, we treat the two-machine flow shop problem with controllable machine speeds.

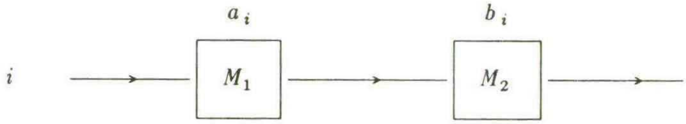
The two-machine flow shop problem, in which the machine speeds are fixed, is well-known to be solvable by Jonhson's algorithm in $O(nlogn)$ time [4] (with n equal to the number of jobs). The two-machine flow shop problem with controllable machine speeds has been introduced by Ishii et al. [3]. They proposed an $O(n^2 logn)$ time algorithm for this problem. This time bound was improved by Van Vliet and Wagelmans [8] to $O(n\sqrt{n})$. The main result of this paper is an $O(nlogn)$ solution method.

Next to the two-machine flow shop problem, other scheduling environments in which the machine speeds are controllable have also been considered. For instance Potts and Van Vliet [6] give a linear time algorithm for the two-machine open shop and Strusevich [7] gives an $O(n^3)$ algorithm for the two-machine no-wait flow shop problem. The above mentioned studies all deal with two-machine environments. Problem instances with more than two machines have been shown to be NP-hard for the case where machine speeds are fixed. Van Vliet [9] discusses a class of algorithms for the general machine flow shop problem with controllable machine speeds for which worst-case bounds are derived.

The sequel is organized as follows. In section 2 we present the problem treated and give some properties on the fixed machine case. In section 3 we treat the case that the first machine can be speeded up. The algorithm to solve the problem is presented together with the datastructures necessary to achieve the time bound. Finally, in section 4 we consider various cost functions which can be considered when speeding up machines. Moreover we show how the algorithm can easily be adapted to the case in which both machines can be speeded up.

2. The two-machine flow shop problem

Two machines, M_1 and M_2, are given on which n jobs have to be processed. Each job $i \in \{1,\ldots,n\}$ has a processing time a_i on M_1 and b_i on M_2. Moreover, processing job i on M_2 can only start if the processing of i on M_1 is finished. Finally job processing should be done unpreempted on both machines:



The objective to be minimized is the makespan C_{max}, i.e., the time on which the last job on M_2 is finished. An optimal schedule can be found where the order of the jobs on both machines is equal, i.e. we can restrict ourselves to permutation schedules, as proved in Johnson [4].

2.1 Johnson's algorithm

An optimal strategy to solve the two-machine flow shop problem is the following:

The jobs are partitioned into two sets L_1 and L_2, where $L_1 := \{i \mid a_i \leq b_i\}$ and $L_2 := \{i \mid a_i > b_i\}$.

The jobs in L_1 are ordered according to increasing processing times on M_1.
The jobs in L_2 are ordered according to decreasing processing times on M_2.

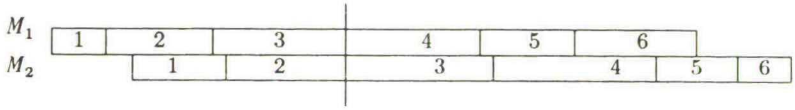
The optimal job permutation consists of first performing the jobs in L_1, in the ordered way and second performing the jobs in L_2 in the ordered way on both machines.

A correctness proof of Johnson's algorithm is easily derived by using a simple exchange argument. It will therefore be omitted here. The complexity of the algorithm is easily seen to be $O(nlogn)$. Partitioning the jobs in L_1 and L_2 can be done $O(n)$ time. Sorting the jobs in L_1 and L_2 takes $O(nlogn)$ time.

Example

i	1	2	3	4	5	6
a_i	2	5	7	8	4	8
b_i	5	6	9	9	3	1

$L_1 = \{1,2,3,4\}$, $L_2 = \{5,6\}$. The optimal schedule is:



It is easily seen that $C_{max} = a_1 + a_2 + a_3 + b_3 + b_4 + b_5 + b_6 = 36$. Job 3 is called the "critical job" here.

2.2 Lower bound for the complexity of the two-machine flow shop problem

We will prove now that Johnson's algorithm cannot be improved upon with respect to worst case behaviour. We do this by showing that sorting is a necessary part of the two-machine flow shop problem.

Take arbitrary integers $a_1, a_2, \ldots, a_n$, where no two are equal. Define $b_i = \min\{a_j | a_j > a_i\}$ for $i = 1, \ldots, n$. If $a_k = \max\{a_i | i = 1, \ldots, n\}$, then $b_k := a_k + 1$. An optimal solution to the thus defined flow shop processes the jobs in order of increasing processing time on M_1. Moreover, it is easily seen that this is the ONLY optimal solution. Therefore finding the optimal schedule amounts to sorting the integers $a_1, a_2, \ldots, a_n$, thus providing a lower bound of $n \log n$ with respect to time complexity.

2.3 Dominance

Now let the jobs be processed according to their numbering, i.e., the first job to be processed is job 1, the second job 2 and so on. The makespan with respect to this schedule can be easily calculated as

$$\max_{1 \leq i \leq n} \left\{ \sum_{j=1}^{i} a_j + \sum_{j=i}^{n} b_j \right\} \tag{2.1}$$

A job for which this maximum is attained is a critical job. It has been shown by Monma and Rinnooy Kan [5] that for any permutation the makespan C_{max} can be calculated using (2.1), where i is the critical job.

In this subsection the elementary concept of dominance will be introduced.

Definition: Let i, j be such that $1 \le i < j \le n$

Job i is said to dominate j if $\sum_{k=i+1}^{j} a_k \le \sum_{k=i}^{j-1} b_k$

Job j is said to dominate i if $\sum_{k=i+1}^{j} a_k \ge \sum_{k=i}^{j-1} b_k$

Notation: i *dom* j and j *dom* i respectively. Moreover, we define i *dom* S for any subset of the jobs, if each job in S is dominated by i. Finally we adopt the convention that i dominates itself, i.e., i *dom* i.

The following propositions are easily proved, directly from the definition:

Proposition 1 Let $i, j \in \{1,\dots,n\}$. Then i *dom* j or j *dom* i or both.

Proposition 2 (transitivity) Let $i,j,k \in \{1,\dots,n\}$.
If i *dom* j and j *dom* k, then i *dom* k.

From Propositions 1 and 2 it follows that for each job i the complete set of jobs can be partitioned (not uniquely) in sets S_i and T_i such that $\forall_{j \in S_i}$: i dominates j and $\forall_{j \in T_i}$: j dominates i. The following property connects the concept of dominance with the critical job:

Proposition 3 i is a critical job if and only if i *dom* $\{1,\dots,n\}$.

3. Speed-up of M_1

If M_1 is speeded up by a factor v, this results in a decrease of all processing times on M_1, with a factor v, i.e. if the original processing time of a job i is a_i, then it becomes a_i/v. We will use the reciprocal of v, in the following. This reciprocal will be denoted by α and it will be called the multiplication factor. Note that $\alpha v=1$.

Defining $C_{max}(\alpha)$ as the makespan of the optimal permutation with respect to the multiplication factor α, it is not hard to see that $C_{max}(\alpha)$ is piecewise linear. It is also monotone non-decreasing in α, but not convex or concave in general.

In this section we derive an algorithm that determines $C_{max}(\alpha)$, by calculating its breakpoints. The running time of the algorithm will be shown to be $O(nlogn)$. Ishii et al. [3] show that these breakpoints can be used to determine optimal machine speeds for various cost functions. See also section 4 on this problem.

We suppose that the jobs are numbered such that

$$\frac{b_1}{a_1} \geq \frac{b_2}{a_2} \geq \dots \geq \frac{b_n}{a_n}$$

Moreover the permutations ρ and σ are determined as follows:

$$a_{\sigma(1)} \leq a_{\sigma(2)} \leq \dots \leq a_{\sigma(n)} \qquad\qquad b_{\rho(1)} \geq b_{\rho(2)} \geq \dots \geq b_{\rho(n)}$$

Note that this amounts to sorting the numbers a_i, b_i and b_i/a_i which takes $O(nlogn)$ time. As a result of the numbering of the jobs, we can determine L_1 and L_2 for a given α simply as $L_1 = \{1,\dots,k\}$ and $L_2 = \{k+1,\dots,n\}$ where k is such that $\frac{b_k}{a_k} \geq \alpha > \frac{b_{k+1}}{a_{k+1}}$. The ordering in L_1 and L_2 now follows from σ and ρ resp.

Although jobs may jump from L_1 to L_2, when α is increased there is a certain monotonicity with regard to the dominance described in section 2. This monotonicity is expressed in the following lemma.

Lemma 3.1

Let two jobs i and j be given. Let i precede j in the optimal schedule for a given α and suppose that j dominates i. When α is increased j remains to dominate i as long as neither job jumps from L_1 to L_2.

Proof. Since j dominates i for α we have:

$$\alpha\left(\sum_{k \in I} a_k + a_j\right) \geq b_i + \sum_{k \in I} b_k \qquad (3.1)$$

Here I consists of the jobs between i and j, in the optimal permutation with respect to α. Raising α increases the left-hand side of (3.1) and will therefore not influence the validity of (3.1). However there may be jobs added and deleted to I when α is raised. Fortunately this happens for any job k exactly when $\alpha a_k = b_k$ so that the contribution to both sides of the inequality sign is equal.

□

Given the "monotonicity" of the dominance relation for pairs of jobs we maintain only the jobs which constitute the "important" dominance relations. Let π describe the optimal permutation with respect to a given α as follows: $\pi(i)$ is the position in the optimal permutation of job i. Determine jobs $i_1, i_2, \ldots, i_R$ such that $\pi(i_r) < \pi(i_{r+1})$ for $r = 1, \ldots, R\text{-}1$ and

I) i_{r-1} dominates i_r $\qquad r = 2, \ldots, R$

II) i_r dominates $\{i \mid \pi(i_{r-1}) < \pi(i) < \pi(i_r)\}$ $\qquad r = 1, \ldots, R \qquad (i_0 := 0)$

From I and II and the transitivity of the dominance relation it follows that these jobs dominate all jobs succeeding them in π. Moreover, these jobs are the only ones with this property. It follows that $\pi(i_R)$ is the last job in the optimal sequence π, i.e. $\pi(i_R) = n$. Moreover, since i_1 *dom* $\{1, \ldots, n\}$ this is a critical job with respect to α. The jobs $i_1, \ldots, i_R$ are called potential critical periods for obvious reasons: when α is raised lemma 3.1 shows that a job that is not potentially critical cannot become critical, until it jumps to L_2 or until i_1 jumps to L_2. This follows directly from lemma 3.1. Before we analyse how "jumping" and "dominance" are handled when α is increased, some parameters are defined:

Definition:

Let α be given:

1) k is chosen such that $L_1 = \{1, \ldots, k\}$ and $L_2 = \{k+1, \ldots, n\}$, i.e. $\frac{b_k}{a_k} \geq \alpha > \frac{b_{k+1}}{a_{k+1}}$.

2) For each pair (i_{r-1}, i_r) we define $\alpha(r)$ as the value of α for which i_r starts to dominate i_{r-1} with respect to π. $\alpha(r)$ is determined as $B(r)/A(r)$ where

$$A(r) = \sum_{i:\pi(i_{r-1}) < \pi(i) \leq \pi(i_r)} a_i \qquad B(r) = \sum_{i:\pi(i_{r-1}) \leq \pi(i) < \pi(i_r)} b_i$$

Note that $\alpha(r) \geq \alpha$ otherwise i_{r-1} *dom* i_r would not be true, contradicting I). Furthermore, since i_1 is a critical job the critical value can be calculated as

$$C_{max}(\alpha) = \alpha A(1) + \sum_{i=1}^{n} b_i - B(1).$$

The following invariant is used, for a given α.

I1) The set of "potential critical jobs" is given by $i_1,\ldots,i_R$ such that
$\pi(i_1)<\pi(i_2)<\ldots<\pi(i_R)$;
i_{r-1} *dom* i_r, $\quad (r=2,\ldots,R)$;
i_r *dom* $\{i|\pi(i_{r-1})<\pi(i)<\pi(i_r)\}$, $\quad (r=1,\ldots,R)$.

I2) $L_1=\{1,\ldots,k\}$; $L_2=\{k+1,\ldots,n\}$ where $k=max\{i|\ \frac{b_i}{a_i}\geq\alpha\}$.

Initially we take $\alpha=0$. Thus $L_1=\{1,\ldots,n\}$; $L_2=\emptyset$ i.e. $k=n$. Moreover $\pi=\sigma$ and $i_r=\sigma^{-1}(r)$ $(r=1,\ldots,n)$.

As a stopping criterion we use $k=0$ <u>and</u> $i_1=\rho^{-1}(n)$. Note that $k=0$ reflects $L_1=\emptyset$ and $i_1=\rho^{-1}(n)$ reflects that the last job is the critical one.

3.1 Description of an iteration

Suppose that I1) and I2) are valid for a given α. Let π be the corresponding optimal permutation. The set of potential critical jobs $\{i_1,\ldots,i_R\}$ will be denoted by J.

Let i_s be such that $s=\arg\min\{\alpha(r)\,|\,r=2,\ldots,R\}$. Raise α to min $\left\{\alpha(s),\ \frac{b_k}{a_k}\right\}$.

If $\alpha=\alpha(s)$, then i_{s-1} is deleted from J and $\alpha(s)$ is recalculated.

If $\alpha=\frac{b_k}{a_k}$ then k moves from L_1 to L_2. The new permutation will be denoted by π'.

First, if $\pi'=\pi$ this amounts to k "jumping" from the last position in L_1 to the first position in L_2. In this case actually nothing happens with respect to I1). I2) is trivially restored.

Second, suppose $\pi'\neq\pi$. Then job k moves from $\pi(k)$ to $\pi'(k)$. As a result each job j with $\pi(k)<\pi(j)\leq\pi'(k)$ moves one place to the left of the permutation: $\pi'(j):=\pi(j)-1$.

Now let t be such that $\pi(i_{t-1}) < \pi(k) \leq \pi(i_t)$. If $\pi(k) < \pi(i_t)$ then $\alpha(t)$ is recalculated. If $\pi(k) = \pi(i_t)$ i.e. $k = i_t$ then i_t is deleted from J and $\alpha(t+1)$ is recalculated. Furthermore, let u be such that $\pi'(i_{u-1}) < \pi'(k) < \pi'(i_u)$. Thus k is placed between the potential critical jobs i_{u-1} and i_u. If i_u dominates k then $\alpha(u)$ is recalculated and nothing else happens. If k dominates i_u, then $\alpha(u)$ is calculated, as well as the speed-up factor for which k starts to dominate i_{u-1}.

Finally k is decreased by one.

3.2 Correctness of the algorithm

By lemma 3.1 it follows that most of the dominance relations mentioned in I1 remain valid. We need only check cases where a job in J becomes dominated by its successor in J ($\alpha = \alpha(s)$) and where a job jumps in between two jobs in J ($\alpha = \frac{b_k}{a_k}$).

Case 1 $\qquad \alpha = \alpha(s)$; i_{s-1} is removed from J.

Then i_s *dom* i_{s-1} and since i_{s-1} *dom* $\{i \mid \pi(i_{s-2}) < \pi(i) < \pi(i_{s-1})\}$ it follows by transitivity (proposition 2) that i_s *dom* $\{i \mid \pi(i_{s-2}) < \pi(i) < \pi(i_{s-1})\}$. Moreover, since for $\alpha = \alpha(s)$ we have i_{s-1} *dom* i_s and i_{s-2} *dom* i_{s-1} we have i_{s-2} *dom* i_s.

Case 2 $\qquad \alpha = \frac{b_k}{a_k}$

If $\pi' = \pi$ then nothing remains to be proved.

If $\pi(k) < \pi(i_t)$ then by lemma 3.1 the dominance relations in I1) with respect to i_t are satisfied.

If $\pi(k) = \pi(i_t)$ i.e. $k = i_t$, it remains to be proved that i_{t+1} dominates $\{i \mid \pi(i_{t-1}) < \pi(i) < \pi(k)\}$. Note that i_{t-1} *dom* i_{t+1}, since i_{t-1} *dom* k and k *dom* i_{t+1} which remains so after k has jumped to L_2, since $\alpha a_k = b_k$. Let l be the successor of k, i.e. $\pi(l) = \pi(k) + 1$. If $l \in L_1$, then trivially, $a_k \leq a_l$. If $l \in L_2$ then $b_k \leq b_l$, since $\pi' \neq \pi$. Moreover, since $l \in L_2$, $b_l \leq \alpha a_l$ and thus $\alpha a_k = b_k \leq b_l \leq \alpha a_l$, which also implies $a_k \leq a_l$. From this it follows directly that l dominates $\{i \mid \pi(i_{t-1}) < \pi(i) < \pi(k)\}$. We are finished now, since i_{t+1} dominates l.

Now let $\pi'(i_{u-1})<\pi'(k)<\pi'(i_u)$. If i_u dominates k, then I1) follows from lemma 3.1. If k dominates i_u we consider the predecessor of k, denoted by l, i.e. $\pi'(l)=\pi'(k)-1$. If $l\in L_2$ then $b_l\geq b_k=\alpha a_k$. If $l\in L_1$ then l is the last job in L_1, since $k\in L_2$. Thus $\pi(k)<\pi(l)$ since $\pi'\neq\pi$ and therefore $a_k\leq a_l$. As $l\in L_1$ we also have $\alpha a_l\leq b_l$ leading to $\alpha a_k\leq\alpha a_l\leq b_l$. Therefore $\alpha a_k\leq b_l$ and this means that l dominates k with respect to π'. Thus, as k dominates i_u we have l dominates i_u. By lemma 3.1 it now follows that $l=i_{u-1}$. This suffices to prove that the invariant is maintained, since $\{i\,|\,\pi'(i_{u-1})<\pi'(i)<\pi'(k)\}=\emptyset$.

3.3 Datastructures

Later it will be shown that the number of iterations is $O(n)$. Therefore the datastructures should be chosen such that the amount of work per iteration is $O(logn)$.

From the previous description of an interation, the reader can easily check the following operations must be performed:

a) For any job k find $i_{r-1},i_r\in J$ such that $\pi(i_{r-1})<\pi(k)\leq\pi(i_r)$.
b) Add/delete a job from J.
c) Calculate $\alpha(r)$ for $i_r\in J$.
d) Find the minimum of $\{\alpha(r)\,|\,i_r\in J\backslash\{i_1\}\}$.

Although π is mentioned we only keep it implicitly in two binary trees T_1 and T_2. These trees also facilitate *c*). Both trees contain n leaves, numbered from 1 to n. A leave numbered $\sigma(i)$ for $i\in L_1$ has label a_i in T_1. The other leaves have label 0. Intermediate nodes in T_1 have a label equal to the sum of the labels of the leaves in its subtree. Analogously in T_2 leaves numbered $\rho(i)$ for $i\in L_2$ have label a_i etc. Now for given j and k the value

$$\sum_{i:\pi(j)<\pi(i)\leq\pi(k)} a_i$$

can be calculated in $O(logn)$ time. Thus $A(i_r)$ for $i_r\in J$ can be calculated in this time. Using a similar structure $B(i_r)$ and therefore $\alpha(r)$ can be calculated in $O(logn)$ time. Finally, updating T_1 and T_2 when a given job k jumps from L_1 to L_2 is also easily seen to take $O(logn)$ time.

A combination of datastructures is used for the execution of a), b) and d). As only 2-3 trees are used we mention the features of this datastructure: the operations SEARCH, ADD and DELETE are supported in $O(log\ n)$ time, n being the number of leaves. A detailed description can be found in [1].

Consider the pairs $(i_r,\alpha(r))$ for $i_r \in J$. One 2-3 tree is used to store the $\alpha(r)$ in an ordered set. J is partitioned in the sets $L_1 \cap J$ and $L_2 \cap J$. In $L_1 \cap J$ the jobs are ordered with the processing times on M_1 as a key i.e. a_{i_r} $(i_r \in L_1 \cap J)$. Analogously in $L_2 \cap J$ the jobs are ordered with processing times on M_2 as a key i.e. b_{i_r} $(i_r \in L_2 \cap J)$.

It is now left to show that the number of iterations is $O(n)$. If $\alpha := \alpha(s)$ then a job is deleted from J. If $\alpha := \frac{b_k}{a_k}$ then a job is deleted from L_1, but a job may be added to J. In either case $2|L_1| + |J|$ is decreased. Since $2|L_1| + J \leq 3n$ at the beginning of the algorithm the bound $O(n)$ is a valid one. Now we have proved the main result:

Theorem 3.2.
$C_{max}(\alpha)$ for $\alpha \in [0,\infty)$ can be determined in $O(nlogn)$ time.

4. Speeding up both machines

In section 3 the speed of M_2 was fixed to 1. However we may introduce a speed-up factor β for this machine as well. It is then asked to minimize a function $f(\alpha,\beta,C_{max})$. However $C_{max}(\alpha,\beta)$ has the same shape for any fixed β compared to $\beta = 1$ as follows from the following formula.

$$C_{max}(\alpha,\beta) = \beta \min_{\pi} \max_{i} \left\{ \delta \sum_{k:\pi(k) \leq i}^{i} a_{\pi(k)} + \sum_{k:\pi(k) \geq i}^{n} b_{\pi(k)} \right\} \qquad \left[\delta = \frac{\alpha}{\beta} \right]$$

Intuitively this is clear: speeding up both machine with the same factor reduces C_{max} with the same factor. Consequently, if we can prove that for fixed β, say $\bar{\beta}$, the function f attains its minimum in a breakpoint of $C_{max}(\alpha,\bar{\beta})$ then we only need to show that for such a breakpoint $(\bar{\alpha},C_{max}(\bar{\alpha},\bar{\beta}))$ the amount of work to calculate $\min_{\varepsilon > 0} \{ f(\varepsilon\bar{\alpha},\varepsilon\bar{\beta},\varepsilon C_{max}(\bar{\alpha},\bar{\beta})) \}$ can be done in $O(logn)$ time. Functions for which this can be done are, for instance, those considered in Ishii et al. [3]:

$$f(\alpha,\beta,C_{max}) = w_1 C_{max}^{q_1} + w_2\alpha^{q_2} + w_3\beta^{q_3} (w_1,w_2,w_3 \text{ positive}, \; q_1,q_2,q_3 \leq -1)$$

Here the minimum can even be determined in constant time.

5. Conclusions

The complexity of the algorithm to determine optimal speeds for the two-machine flow shop scheduling problem has now been reduced to a minimum for most objective functions. A similar result has been proved by Potts and Van Vliet [6] for the two-machine open shop scheduling problem. For the no-wait flow shop the complexity gap lies between $nlogn$ and n^3. The $O(nlogn)$ time bound for the original problem is proved in Gilmore et al. [2], whereas the time bound for the problem with speed-up of machines is given in Strusevich [7]. It is an open problem, whether this gap can be tightened.

Acknowledgement

The author would like to thank Antoon Kolen, Mario van Vliet, Alexander Rinnooy Kan, Albert Wagelmans and Carin Zwaneveld for their comments on earlier drafts of this paper, which greatly simplified the exposition.

References

[1] Aho, A.V., J.E. Hopcroft and J.D. Ullmann, "Data structures and algorithms". Addison Wesley; Series in computer science and information processing (1983).

[2] Gilmore, P.C., E.L. Lawler and D.B. Shmoys, "Well-soved special cases, in: The Travelling Salesman Problem. A Guided Tour of Combinatorial Optimization" (E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys eds.). Wiley, Chichester et al. (1986), pp. 87-143.

[3] Ishii, H., T. Masuda and T. Nishida, "Two machine mixed shop scheduling problem with controllable machine speeds". Discrete Applied Mathematics 17, (1987) pp. 29-38.

[4] Johnson, S.M., "Optimal two- and three stage production schedules with setup times included". Naval Research Logistics Quarterly 1, (1954) pp. 61-68.

[5] Monma C.L. and A.H.G. Rinnooy Kan, "A concise survey of efficiently solvable special cases of the permutation flow shop problem". RAIRO Recherche Operationelle 17 (1983), pp. 105-119.

[6] Potts C.N. and M. Van Vliet, "A note on speeding up machines in a two machine open shop". Technicalreport, Econometric Institute, Erasmus University, Rotterdam, the Netherlands (1991), in preparation.

[7] Strusevich, V.A., "Two machine flow shop scheduling problem with no wait in process: controllable machine speeds". Technicalreport, Econometric Institute, Erasmus University, Rotterdam, the Netherlands (1991), in preparation.

[8] Van Vliet, M. and A.P.M. Wagelmans, "Speeding up machines in a two machine flow shop". Technical report no 9001/A, Econometric Institute, Erasmus University, Rotterdam, the Netherlands (1990).

[9] Van Vliet, M., "Optimization of Manufacturing System Design". Ph.D. thesis, in preparation (1991).

IN 1990 REEDS VERSCHENEN

419 Bertrand Melenberg, Rob Alessie
A method to construct moments in the multi-good life cycle consumption model

420 J. Kriens
On the differentiability of the set of efficient (μ,σ^2) combinations in the Markowitz portfolio selection method

421 Steffen Jørgensen, Peter M. Kort
Optimal dynamic investment policies under concave-convex adjustment costs

422 J.P.C. Blanc
Cyclic polling systems: limited service versus Bernoulli schedules

423 M.H.C. Paardekooper
Parallel normreducing transformations for the algebraic eigenvalue problem

424 Hans Gremmen
On the political (ir)relevance of classical customs union theory

425 Ed Nijssen
Marketingstrategie in Machtsperspectief

426 Jack P.C. Kleijnen
Regression Metamodels for Simulation with Common Random Numbers: Comparison of Techniques

427 Harry H. Tigelaar
The correlation structure of stationary bilinear processes

428 Drs. C.H. Veld en Drs. A.H.F. Verboven
De waardering van aandelenwarrants en langlopende call-opties

429 Theo van de Klundert en Anton B. van Schaik
Liquidity Constraints and the Keynesian Corridor

430 Gert Nieuwenhuis
Central limit theorems for sequences with m(n)-dependent main part

431 Hans J. Gremmen
Macro-Economic Implications of Profit Optimizing Investment Behaviour

432 J.M. Schumacher
System-Theoretic Trends in Econometrics

433 Peter M. Kort, Paul M.J.J. van Loon, Mikulás Luptacik
Optimal Dynamic Environmental Policies of a Profit Maximizing Firm

434 Raymond Gradus
Optimal Dynamic Profit Taxation: The Derivation of Feedback Stackelberg Equilibria

435 Jack P.C. Kleijnen
Statistics and Deterministic Simulation Models: Why Not?

436 M.J.G. van Eijs, R.J.M. Heuts, J.P.C. Kleijnen
Analysis and comparison of two strategies for multi-item inventory systems with joint replenishment costs

437 Jan A. Weststrate
Waiting times in a two-queue model with exhaustive and Bernoulli service

438 Alfons Daems
Typologie van non-profit organisaties

439 Drs. C.H. Veld en Drs. J. Grazell
Motieven voor de uitgifte van converteerbare obligatieleningen en warrantobligatieleningen

440 Jack P.C. Kleijnen
Sensitivity analysis of simulation experiments: regression analysis and statistical design

441 C.H. Veld en A.H.F. Verboven
De waardering van conversierechten van Nederlandse converteerbare obligaties

442 Drs. C.H. Veld en Drs. P.J.W. Duffhues
Verslaggevingsaspecten van aandelenwarrants

443 Jack P.C. Kleijnen and Ben Annink
Vector computers, Monte Carlo simulation, and regression analysis: an introduction

444 Alfons Daems
"Non-market failures": Imperfecties in de budgetsector

445 J.P.C. Blanc
The power-series algorithm applied to cyclic polling systems

446 L.W.G. Strijbosch and R.M.J. Heuts
Modelling (s,Q) inventory systems: parametric versus non-parametric approximations for the lead time demand distribution

447 Jack P.C. Kleijnen
Supercomputers for Monte Carlo simulation: cross-validation versus Rao's test in multivariate regression

448 Jack P.C. Kleijnen, Greet van Ham and Jan Rotmans
Techniques for sensitivity analysis of simulation models: a case study of the CO_2 greenhouse effect

449 Harrie A.A. Verbon and Marijn J.M. Verhoeven
Decision-making on pension schemes: expectation-formation under demographic change

450 Drs. W. Reijnders en Drs. P. Verstappen
Logistiek management marketinginstrument van de jaren negentig

451 Alfons J. Daems
Budgeting the non-profit organization
An agency theoretic approach

452 W.H. Haemers, D.G. Higman, S.A. Hobart
Strongly regular graphs induced by polarities of symmetric designs

453 M.J.G. van Eijs
Two notes on the joint replenishment problem under constant demand

454 B.B. van der Genugten
Iterated WLS using residuals for improved efficiency in the linear model with completely unknown heteroskedasticity

455 F.A. van der Duyn Schouten and S.G. Vanneste
Two Simple Control Policies for a Multicomponent Maintenance System

456 Geert J. Almekinders and Sylvester C.W. Eijffinger
Objectives and effectiveness of foreign exchange market intervention
A survey of the empirical literature

457 Saskia Oortwijn, Peter Borm, Hans Keiding and Stef Tijs
Extensions of the τ-value to NTU-games

458 Willem H. Haemers, Christopher Parker, Vera Pless and Vladimir D. Tonchev
A design and a code invariant under the simple group Co_3

459 J.P.C. Blanc
Performance evaluation of polling systems by means of the power-series algorithm

460 Leo W.G. Strijbosch, Arno G.M. van Doorne, Willem J. Selen
A simplified MOLP algorithm: The MOLP-S procedure

461 Arie Kapteyn and Aart de Zeeuw
Changing incentives for economic research in The Netherlands

462 W. Spanjers
Equilibrium with co-ordination and exchange institutions: A comment

463 Sylvester Eijffinger and Adrian van Rixtel
The Japanese financial system and monetary policy: A descriptive review

464 Hans Kremers and Dolf Talman
A new algorithm for the linear complementarity problem allowing for an arbitrary starting point

465 René van den Brink, Robert P. Gilles
A social power index for hierarchically structured populations of economic agents

IN 1991 REEDS VERSCHENEN

466 Prof.Dr. Th.C.M.J. van de Klundert - Prof.Dr. A.B.T.M. van Schaik
Economische groei in Nederland in een internationaal perspectief

467 Dr. Sylvester C.W. Eijffinger
The convergence of monetary policy - Germany and France as an example

468 E. Nijssen
Strategisch gedrag, planning en prestatie. Een inductieve studie binnen de computerbranche

469 Anne van den Nouweland, Peter Borm, Guillermo Owen and Stef Tijs
Cost allocation and communication

470 Drs. J. Grazell en Drs. C.H. Veld
Motieven voor de uitgifte van converteerbare obligatieleningen en warrant-obligatieleningen: een agency-theoretische benadering

471 P.C. van Batenburg, J. Kriens, W.M. Lammerts van Bueren and R.H. Veenstra
Audit Assurance Model and Bayesian Discovery Sampling

472 Marcel Kerkhofs
Identification and Estimation of Household Production Models

473 Robert P. Gilles, Guillermo Owen, René van den Brink
Games with Permission Structures: The Conjunctive Approach

474 Jack P.C. Kleijnen
Sensitivity Analysis of Simulation Experiments: Tutorial on Regression Analysis and Statistical Design