Exponentially better than brute force: solving the job-shop scheduling problem optimally by dynamic programming

Research Memorandum 2009-56

Joaquim A.S. Gromicho
Jelke J. van Hoorn
Francisco Saldanha-da-Gama
Gerrit T. Timmer
Exponentially better than brute force: solving the job-shop scheduling problem optimally by dynamic programming

Joaquim A. S. Gromicho
Vrije Universiteit, Amsterdam, The Netherlands & ORTEC, Gouda, The Netherlands
email: jgromicho@feweb.vu.nl

Jelke J. van Hoorn
Vrije Universiteit, Amsterdam, The Netherlands & ORTEC, Gouda, The Netherlands
email: jhoorn@feweb.vu.nl

Francisco Saldanha-da-Gama
Faculdade de Ciências & Centro de Investigação Operacional, Universidade de Lisboa, Portugal
email: fsgama@fc.ul.pt

Gerrit T. Timmer
Vrije Universiteit, Amsterdam, The Netherlands & ORTEC, Gouda, The Netherlands
email: Gerrit.Timmer@ortec.com

Scheduling problems received substantial attention during the last decennia. The job-shop problem is a very important scheduling problem, which is NP-hard in the strong sense and with well-known benchmark instances of relatively small size which attest the practical difficulty in solving it. The literature on job-shop scheduling problem includes several approximation and optimal algorithms. So far, no algorithm is known which solves the job-shop scheduling problem optimally with a lower complexity than the exhaustive enumeration of all feasible solutions. We propose such an algorithm, based on the well-known Bellman equation designed by Held and Karp to find optimal sequences and which offers the best complexity to solve the Traveling Salesman Problem known to this date. For the TSP this means $O(n^22^n)$ which is exponentially better than $O(n!)$ required to evaluate all feasible solutions. We reach similar results by first recovering the principle of optimality, which is not obtained for free in the case of the job-shop scheduling problem, and by performing a complexity analysis of the resulting algorithm. Our analysis is conservative but nevertheless exponentially better than brute force. We also show very promising results obtained from our implementation of this algorithm, which seem to indicate two things: firstly that there is room for improvement in the complexity analysis (we observe the generation of a number of solutions per state for the benchmark instances considered which is orders of magnitude lower than the bound we could devise) and secondly that the potential practical implications of this approach are at least as exciting as the theoretical ones, since we manage to solve some celebrated benchmark instances in processing times ranging from seconds to minutes.

Key words: job-shop scheduling; dynamic programming; complexity analysis

MSC2000 Subject Classification: Primary: 90B35, 90C39; Secondary: 68W40

OR/MS subject classification: Primary: deterministic scheduling, dynamic programming; Secondary: analysis of algorithms

1. Introduction. In a job-shop scheduling problem $n$ jobs have to be processed by $m$ dedicated machines. Each job has to visit all the machines following a specific order. The time each job requires in each machine depends on the job and on the machine and it is assumed to be known in advance. The jobs can not overlap in the machines and no job can be processed simultaneously by two or more machines. Preemption is not allowed. The goal is to schedule the jobs so as to minimize the makespan, which is the maximum of their completion time.

The job-shop scheduling problem is one of the most studied combinatorial optimization problems. Nevertheless, it still remains a very challenging problem to solve optimally. From a complexity point of view, the problem is NP-hard (Lenstra and Rinnooy Kan [21]). Lenstra et al. [22] show that even some ‘simplified’ versions are NP-hard. These include 3 machines and 3 jobs; 2 machines and no more than 3 operations per job (in this case a job may have to visit a machine more than once); 3 machines and no more than 2 operations per job; 3 machines and unitary processing times. Nevertheless, some particular cases of the job-shop scheduling problem are polynomial, such as the problem with 2 machines and no more than 2 operations per job (Jackson [14]) and the problem with 2 machines and unitary processing times (Hefetz and Adiri [12]).

Among the methodologies that have been considered for tackling the job-shop scheduling problem we
can distinguish exact procedures (working for instances with a rather limited size), heuristic procedures and polynomial time approximation algorithms.

To the best knowledge of the authors, all the exact algorithms that have been proposed for the job-shop scheduling problem are branch-and-bound procedures. Without being exhaustive we can cite the works of Applegate and Cook [3], Brucker et al. [4], Carlier and Pinson [5], Lageweg et al. [17] and Martin and Smoys [24].

Many heuristic approaches have been been proposed in the literature for obtaining good quality solutions to the job-shop scheduling problem. A well-known procedure is the so-called shifting bottleneck procedure by Adams et al. [1]. Genetic algorithms were proposed by Croce et al. [6] and Dornendorf and Pesch [7]. Van Laarhoven et al. [30] and Steinhold et al. [28] proposed simulated annealing procedures for obtaining feasible solutions to the problem. A tabu search procedure was proposed in the works by Taillard [29] and Nowicki and Smutnicki [25]. Recently, Zhang et al. [33] proposed a new hybrid approach combining tabu search and simulated annealing, which proved to be very efficient for finding optimal or near-optimal solutions for many benchmark instances of the problem. Rego and Duarte [26] proposed a procedure that combines the basic shifting bottleneck procedure by Adams et al. [1] with a dynamic and adaptive neighborhood search procedure based on the so-called filter-and-fan method.

As far as the polynomial time approximation algorithms are concerned, Shmoys et al. [27] propose a procedure with a performance guarantee $O\left(\frac{\log(m\mu)\log(\min(m\mu,p_{\text{max}}))}{\log\log(m\mu)}\right)$ for a job-shop scheduling problem in which a job may have to return more than once to each machine. $p_{\text{max}} = \max_{ij} p_{ij}$, $m$ is the number of machines and $\mu$ denotes the maximum number of operations over all jobs. When each job has to visit each machine exactly once, the factor $\mu$ can be ignored in the performance guarantee. Goldberg et al. [10] improve the above performance by presenting a polynomial time approximation algorithm with a performance guarantee $O\left(\frac{\log(m\mu)\log(\min(m\mu,p_{\text{max}}))}{\log\log(m\mu)}\right)^2$. For the case in which each job visits at most once each machine the above performance is still improved by Feige and Scheideler [8] who provide an approximation guarantee $O\left(m\mu\log(m\mu)\log\log(m\mu)\right)$. For a fixed number of machines and also for a fixed maximum number of operations per job, Shmoys et al. [27] present a $2 + \epsilon$ approximation algorithm. Jansen et al. [16] improve the previous performance guarantee. Leighton et al. [19] and Leighton et al. [20] propose polynomial time approximation algorithms with constant performance guarantee for the job shop scheduling problem with unitary processing times assuming that each job has to be processed exactly once on each machine.

Unlike the above mentioned complexity results about approximations, little seems to be known about the complexity of optimal algorithms for many NP-hard problems. Woeginger [31] stresses the importance of such algorithms and points out that for many such problems it is possible to do better than ‘brute force’. In particular, for a single machine scheduling problem, an algorithm based on dynamic programming is mentioned. This algorithm follows along the lines of the celebrated work by Held and Karp [13] for the Traveling Salesman Problem, which offers still to this date the best complexity of an optimal algorithm for the TSP (see Woeginger [32]).

In the case of the job-shop scheduling problem, the decision to be made regards the order by which the $n$ jobs will be processed on each of the $m$ machines. Accordingly, a brute-force enumerative algorithm for the problem has worst-case complexity $O\left((n!)^m\right)$, which is lower than the worst-case complexity for branch-and-bound algorithms.

In this paper we present an exact algorithm for the job-shop scheduling problem with a complexity that we can prove to be $O\left(\left(2p_{\text{max}}\right)^{n-1} (n\sqrt{n} + (2p_{\text{max}})^2) (m+1)^n n\right)$, which is exponentially lower than brute-force. The new algorithm is based on dynamic programming. We show that for the job-shop scheduling problem a straightforward application of the Held and Karp equation is not possible because the optimality principle does not hold. Nevertheless, we show that it is possible to redefine the state space and adjust the Belman equation accordingly, in order to recover the optimality principle.

The remainder of the paper is organized as follows. In the next section we present some background which includes notation, definitions and basic properties of the problem. In section 3 we present the new algorithm. Section 4 is devoted to a complexity analysis. In Section 5 some computational tests performed with the new procedure are presented and discussed. The paper ends with a conclusion about the research done.
2. Notation, definitions and basic properties. Consider the job-shop scheduling problem as defined in the previous section. Let $\mathcal{J} = \{j_1, j_2, \ldots, j_n\}$ denote the set of jobs and $\mathcal{M} = \{m_1, m_2, \ldots, m_m\}$ the set of machines. $p_{ij}$ stands for the processing time of job $j \in \mathcal{J}$ on machine $i \in \mathcal{M}$. A job consists of $n$ task operations each of which associated with a specific machine. The sequence of operations for each job $j \in \mathcal{J}$ is denoted by $\pi_j(1), \ldots, \pi_j(m)$ that is, for job $j \in \mathcal{J}$, $\pi_j(i)$ is the $i$-th machine that job $j$ has to visit. $\mathcal{O} = \{o_1, o_2, \ldots, o_{n \times m}\}$ is the set of operations. The first $n$ operations refer to the first operation of each job (in the order of the jobs), operations $o_{n+1}, \ldots, o_{2n}$ concern the second operation of the $n$ jobs, and so on. For an operation $o \in \mathcal{O}$ we denote by $j(o)$ and $m(o)$ the corresponding job and machine, respectively. Note that $j(o_i) = i \mod n$. We denote by $p(o)$ the processing time of operation $o \in \mathcal{O}$. Note that $p(o) = p_{m(o)}j(o)$.

**Definition 2.1** A schedule is a function $\psi: \mathcal{O} \rightarrow \mathbb{N} \cup \{0\}$ such that for each operation $o \in \mathcal{O}$, $\psi(o)$ gives the starting time of operation $o$. A schedule $\psi$ is said to be feasible if:

(i) $\psi(o) \geq 0$, $o \in \mathcal{O}$;

(ii) $\forall o_k, o_l \in \mathcal{O}$ such that $j(o_k) = j(o_l)$ and $k < l$ one has $\psi(o_k) + p(o_k) \leq \psi(o_l)$;

(iii) $\forall o_k, o_l \in \mathcal{O}$ such that $o_k \neq o_l$ and $m(o_k) = m(o_l)$ we have $\psi(o_l) + p(o_l) \leq \psi(o_k) \lor \psi(o_k) + p(o_k) \leq \psi(o_l)$.

The goal in a job-shop scheduling problem is to find the feasible schedule with the minimum makespan that is, the feasible schedule $\psi$ which minimizes $\max_{o \in \mathcal{O}}\{\psi(o) + p(o)\}$. In this study we consider the value 0 for the origin of time.

For every feasible schedule for the job-shop scheduling problem it is possible to associate a sequence of operations where the order of the operations processed on a single machine as well as the order defined for each job is preserved. One example of such type of sequence is obtained by sorting all operations by their starting time in the schedule considered. However, one single sequence of operations defines an infinite number of schedules (e.g. all those obtained by the addition of positive constants to the starting time of all operations). On the other hand, not all the sequences define a feasible solution. In fact, even if a sequence preserves the order of operations given for each job, it may create a dead-lock.

Hereafter, we will be interested only in sequences that correspond to feasible solutions to the problem that is, feasible sequences. For the sake of simplicity, in the remainder of the paper we will omit the word “feasible” every time we refer to a sequence.

Given a sequence of operations of a job-shop scheduling problem, we can obtain the schedule that results from starting all the operations as soon as possible in the order defined by the sequence and keeping feasibility. This is the (unique) schedule which has the minimum makespan among all the schedules that can be associated to the sequence. Hereafter, we will be interested only in such schedules.

Note that different sequences of operations can lead to the same schedule. Figure 1 depicts a feasible solution for a small instance of the job-shop scheduling problem. The schedule associated with this solution can be obtained from the following feasible sequences: $o_1 o_2 o_3 o_4, o_1 o_2 o_4 o_3, o_2 o_1 o_3 o_4$ and $o_2 o_1 o_4 o_3$. This small example makes it clear that there is not a one to one correspondence between feasible schedules and sequences of operations. The following result defines a unique sequence for every schedule.

![Figure 1: An illustrative schedule](image)

**Proposition 2.1** For every feasible solution for the job-shop scheduling problem there is one and only one sequence of operations defining the schedule such that the completion time of the operations along the sequence is non decreasing and in which the order of the machines is increasing for two consecutive operations with equal completion time.

**Proof.** Consider a feasible solution for the job-shop scheduling problem. A sequence of operations featuring the conditions stated is obtained by sorting the operations non-decreasingly in terms of their completion time and for those that have an equal completion time, by sorting them in increasing order.
of the machine associated with them. The total lexicographic order imposed on this sequence assures unicity.

An immediate consequence of the previous proposition is stated in the following corollary.

**Corollary 2.1** For an optimal solution of the job-shop scheduling problem there exists one and only one sequence featuring the conditions stated in proposition 2.1.

With the purpose of building (feasible) sequences (and eventually the optimal sequence) to a job-shop scheduling problem, the operations can be iteratively added to the sequence. Before all the operations are in the sequence, we have an incomplete or partial sequence which we define formally as follows.

**Definition 2.2** A partial sequence is a sequence of operations defined by the first $k$ operations of a sequence for any $k = 0, ..., |O|$.

For a partial sequence $T$ we denote by $C_{\text{max}}(T)$ the corresponding (partial) makespan, which is the maximum completion time for the operations scheduled in the order of the sequence.

**Definition 2.3** If $T_1$ and $T_2$ are two partial sequences involving the same set of operations, we say that $T_2$ dominates $T_1$ if $C_{\text{max}}(T_2) \leq C_{\text{max}}(T_1)$.

**Definition 2.4**

1. An ordered sequence is a sequence ordered as stated in proposition 2.1.

2. An ordered partial sequence is a sequence defined by the first $k$ operations of an ordered sequence for any $k = 0, ..., |O|$.

3. A sequence or a partial sequence not ordered as stated in proposition 2.1 is called unordered.

It is straightforward to conclude that any unordered partial sequence can be converted into an ordered partial sequence with the same makespan. In fact, reordering the operations as stated in proposition 2.1 does not increase the makespan.

**Definition 2.5** Let $S \subseteq O$ denote a subset of operations such that at least one ordered partial sequence can be associated to it. Let $T$ denote an ordered partial sequence that can be associated to $S$.

1. Any set of operations obtained by adding to $S$ one operation that is not in $S$ but such that all the operations that precede it in the same job are already in $S$ is called an expansion of $S$.

2. A partial sequence that is obtained by adding to the end of $T$ one operation $o$ that can be used to expand $S$ is called an expansion of the ordered partial sequence $T$ with $o$. Let $T + o$ denote such expansion.

3. Every partial sequence can be progressively expanded leading to a full sequence of operations thus defining a feasible solution. Such final sequence is called a completion of the initial partial solution.

**Remark 2.1** A subset of operations $S \subseteq O$ has no (ordered) partial sequence associated with it when for some job an operation is in $S$ while one of the preceding operations of the same job is not in $S$.

Hereafter we consider only subsets of operations $S \subseteq O$ such that at least one ordered partial sequence can be associated to them.

It should be noted that the expansion of an ordered partial sequence does not necessarily lead to a new ordered partial sequence. That is the case when the operation selected to join the sequence has an earliest completion time lower than the makespan of the partial sequence.

The job-shop scheduling problem can now be defined as the problem of finding the ordered sequence of operations which leads to the minimum makespan that is, the job-shop shop scheduling problem can be seen simply as a sequencing problem involving all the operations that have to be performed. This motivates the approach that is presented and discussed in the following sections.
3. A Dynamic Programming approach for the job-shop scheduling problem. Held and Karp [13] presented a Dynamic Programming (DP) formulation for sequencing problems, which is famous for its application to the Traveling Salesman Problem. For this particular problem, the resulting Bellman equation is
\[
\begin{align*}
    C(\{u\}, u) &= c_{0u} \\
    C(S, u) &= \min_{v \in S \setminus \{u\}} C(S \setminus \{u\}, v) + c_{vu}
\end{align*}
\]
where \( C(S, u) \) is the cost of the optimal path starting at 0 visiting all nodes in \( S \) and ending in \( u \), and \( c_{vu} \) is the cost of going from \( v \) to \( u \).

As mentioned above, the job-shop scheduling problem can be seen as a sequencing problem. Therefore, one could be tempted to use the Bellman equation above using the dominance criteria given by definition 2.3 and considering only ordered expansions.

However a small example is enough to show that the optimality principle does not hold on ordered (partial) sequences using the dominance concept introduced in the previous section. Figure 2a depicts the optimal solution for an instance of the job-shop scheduling problem with 3 jobs and 3 machines (the data can be easily retrieved from the figure). The ordered sequence defining the solution depicted is \( o_1 \ o_3 \ o_6 \ o_2 \ o_5 \ o_4 \ o_9 \ o_7 \ o_8 \). However, the ordered partial sequence \( \{ o_2 \ o_3 \ o_6 \ o_1 \ o_5 \} \) which leads to the schedule depicted in Figure 2b dominates the partial sequence \( \{ o_1 \ o_3 \ o_6 \ o_2 \ o_5 \} \) from the optimal solution.

![Figure 2: Loosing optimality](image)

In order to use the recursive equation above we propose several adjustments, which allow us to ‘restore’ the optimality principle.

3.1 Restoring the optimality guarantee. Let \( S \subseteq \mathcal{O} \). Denote by \( \Xi(S) \) the set of all the ordered (partial) sequences that can be associated with \( S \) (assuming that there is at least one such sequence that is, \( \Xi(S) \neq \emptyset \)).

Denote by \( \varepsilon(S) \) the set of all operations that can be used to obtain an expansion of \( S \). Note that \( |\varepsilon(S)| \leq n \) because \( S \) can only be expanded with the first unsequenced operation in each job. Note also that if \( |\varepsilon(S)| < n \), there are \( n - |\varepsilon(S)| \) jobs that have all their operations already sequenced.

Denote by \( \eta(T) \) the set of all operations that can be used to obtain an ordered expansion of an ordered partial sequence \( T \). For \( T \in \Xi(S) \) we have that \( \eta(T) \subseteq \varepsilon(S) \). For a partial ordered sequence \( T \in \Xi(S) \) let \( i^* \) denote the machine with the largest completion time (i.e. the machine determining the value \( C_{\max}(T) \)). In case there are multiple such machines, let \( i^* \) denote the highest-numbered machine.

For each \( o \in \varepsilon(S) \) and for each \( T \in \Xi(S) \) define \( \psi(T, o) \) as the earliest starting time for operation \( o \) if this operation is added to the ordered partial sequence \( T \). We have \( o \in \eta(T) \) if and only if
\[
\psi(T, o) + p(o) > C_{\max}(T) \land m(o) \leq i^* \quad \text{or} \quad \psi(T, o) + p(o) \geq C_{\max}(T) \land m(o) > i^*.
\]

Define also
\[
\xi(T, o) = \begin{cases} 
\psi(T, o) + p(o), & \text{if } o \in \eta(T) \\
C_{\max}(T) + p(o), & \text{otherwise.}
\end{cases}
\]
Note that $C_{\text{max}}(T) \leq \xi(T, o) \leq C_{\text{max}}(T) + p(o), \forall o \in \varepsilon(S)$.

$\xi(T, o)$ represents an ‘aptitude’ value for operation $o$ if this operation is used to expand $T$. In case $T + o$ is an ordered partial sequence ($o \in \eta(T)$), $\xi(T, o)$ gives the (minimum) makespan for the operations in $T + o$. Otherwise, $\xi(T, o)$ gives a loose upper bound on this makespan.

Actually $\xi(T, o)$ gives a lower bound of the completion of $o$ in any ordered sequence that starts with $T$ as a subsequence. For $o \in \eta(T)$ it follows directly from the definition. Otherwise, if $o \not\in \eta(T)$ and $o$ is added directly to $T$ then $T + o$ is unordered. To add $o$ to any (ordered) extension $T'$ of $T$ such that $T' + o$ is ordered, an operation $o'$ with $m(o') = m(o)$ should be part of this extension ($T' = T + \ldots + o' + \ldots$). Since $T + \ldots + o' + \ldots$ is ordered we have $\psi(T + \ldots, o') + p(o') \geq C_{\text{max}}(T + \ldots) \geq C_{\text{max}}(T)$. Since $o'$ and $o$ are on the same machine we have that $\psi(T', o) \geq \psi(T + \ldots, o') + p(o') \geq C_{\text{max}}(T)$, thus $\psi(T', o) + p(o) \geq C_{\text{max}}(T) + p(o) = \xi(T, o)$.

Denote by $\lambda(S)$ the set of operations that are the last operation of some job represented in $S$. Note this set only depends on the set $S$ and not on any sequence $T \in \Xi(S)$. Taking into account this definition we always have $|\lambda(S)| \leq n, \forall S \subseteq \mathcal{O}$.

In order to clarify the previous concepts, we consider the instance of the job-shop scheduling problem that was introduced in Figure 2. In particular, for this instance, consider the ordered partial sequence $T = o_1 o_2 o_6 o_2 o_4$ (which leads to the schedule depicted in Figure 3a). In this case we have $C_{\text{max}}(T) = 6, S = \{o_1, o_2, o_3, o_4, o_6\}, \varepsilon(S) = \{o_2, o_7, o_9\}, \lambda(S) = \{o_2, o_4, o_6\}$ and $\eta(T) = \{o_7, o_9\}$. Taking into account that $p(o_5) = 1, p(o_7) = 1$ and $p(o_9) = 3$ we obtain (see Figure 3b):

$$\psi(T, o_5) = 4, \psi(T, o_7) = 6, \psi(T, o_9) = 4 \quad \xi(T, o_5) = 7, \xi(T, o_7) = 7, \xi(T, o_9) = 7.$$
In any case, $\psi(T_1, o) + p(o) \geq \psi(T_2, o) + p(o)$ and thereby $\psi(T_1, o) \geq \psi(T_2, o)$.

This shows that it is possible to add $o$ to $T_2$ starting at time $\psi(T_1, o)$. In this case, we would obtain $C_{\text{max}}(T_2 + o) = \psi(T_1, o) + p(o) = C_{\text{max}}(T_1 + o)$. However, the best starting time for $o$ when the operation is added to $T_2$ is $\psi(T_2, o)$ which can be lower than $\psi(T_1, o)$. In such case we would get $C_{\text{max}}(T_2 + o) < C_{\text{max}}(T_1 + o)$. In any case we get $C_{\text{max}}(T_2 + o) \leq C_{\text{max}}(T_1 + o)$, which proves the result.

Corollary 3.1 In the conditions stated in Proposition 3.1, every ordered completion of $T_1$ is dominated by the same (possibly unordered) completion of $T_2$.

Proof. The result follows by induction using Proposition 3.1.

The previous proposition and its corollary establish a means for comparing two ordered partial sequences namely, in terms of their possible expansions if done using the same operation. We introduce some additional notation. For $S \subseteq O$ and $T_1, T_2 \in \Xi(S)$ when $\xi(T_2, o) \leq \xi(T_1, o), \forall o \in \varepsilon(S)$, we write $T_2 \prec T_1$. The reverse relation will be denoted by $\succ$ and equality by $\equiv$. Note that if $\varepsilon(S) = \emptyset$ then $T_1, T_2 \in \Xi(O)$ and $T_2 \prec T_1$ will be another way to write $C_{\text{max}}(T_2) \leq C_{\text{max}}(T_1)$.

The relation just established between ordered partial sequences allows us to partition the set $\Xi(S)$ into two sets, say $\hat{\Xi}(S)$ and $\Xi(S) \setminus \hat{\Xi}(S)$, such that for every $T_1 \in \hat{\Xi}(S)$, $\nexists T_2 \in \Xi(S)$ such that $T_2 \prec T_1$. We consider the minimal set $\hat{\Xi}(S)$ among all the possibilities that is, if two or more potential elements for being included in $\hat{\Xi}(S)$, say $T_1, T_2, \ldots$, are such that $T_1 \equiv T_2 \equiv \ldots$ then only one is chosen to be included in the set. Any rule can be defined to determine this choice. By construction, $|\hat{\Xi}(O)| = 1$ and it becomes clear that for every possible choice of a sequence $T \in \hat{\Xi}(O)$, $T$ is optimal for the job-shop scheduling problem.

Denote by $\hat{\Xi}(S)$ the set ordered partial sequences $T \in \hat{\Xi}(S)$, such that all subsequences of $T$ (naturally all ordered), are also elements of the set $\hat{\Xi}(\cdot)$ associated with their correspondent sets.

We have finally gathered all the ingredients that allow us to restore the optimality guarantee for the job-shop scheduling problem. The following results assure this.

Proposition 3.2 The set $\hat{\Xi}(O)$ is non-empty.

Proof. Let $T_1 \in \Xi(O)$ be an optimal sequence and suppose that $T_1 \notin \hat{\Xi}(O)$. In this case, there is at least one partial subsequence $T'_1 \in \Xi(S')$ of $T_1$ such that $T'_1 \notin \hat{\Xi}(S')$. Among such sequences, let $T'_1$ be the one with the minimum number of operations. Then $\exists T'_2 \in \hat{\Xi}(S')$ such that $T'_2 \prec T'_1$. Accordingly, we can consider the completion $T_2$ of $T'_2$ with the same operations and in the same order that is necessary to obtain $T_1$ from $T'_1$. We can distinguish the following cases:

I. $T_2$ is ordered. This gives us two cases.
   a) $T_2 \in \hat{\Xi}(O)$. 
   b) $T_2 \notin \hat{\Xi}(O)$. Now we can use the same procedure to find a new optimal solution $T_3$. However, since $T'_2 \in \hat{\Xi}(S')$, the subsequence $T''_2$ of $T_2$ with the minimum number of operations among those such that $T''_2 \notin \hat{\Xi}(S')$ has more operations than $T'_1$.

II. $T_2$ is not ordered. We have that the ordered sequence $T''_2$ corresponding to the schedule defined by this completion satisfies, for all operations $o \in O \setminus S'$, $\psi_{T''_2}(o) \leq \psi_{T_1}(o)$ with strict inequality for at least one of them.

In case Ia we are finished. Case Ib increases the number of operations of the partial subsequence eventually expanded to the optimal sequence found in case Ia. Case II does not necessarily increase the number of operations of the partial subsequence nor decreases it. As the property stated in II. is transitive it cannot include any circular relations. As case Ib increases the number of operations known of the optimal sequence $T \in \hat{\Xi}(O)$ to be found, the procedure of finding $T$ may alternate between cases Ib and II. but is finite and will end in case Ia.

Corollary 3.2 The set $\hat{\Xi}(O)$ contains the optimal sequence to the job-shop scheduling problem.
Define shop scheduling problem. Finally, we conclude that it contains a single element, which is optimal.

The question arising now is how can $\hat{\mathcal{X}}(\mathcal{O})$ be progressively obtained. In order to give an appropriate answer, we define $X(S)$ as the set of all ordered partial sequences $T \in \Xi(S)$ such that $T$ is an expansion of an ordered partial sequence $T'$ where $T = T' + o$ and $T' \in \hat{\mathcal{X}}(S \setminus \{o\})$. More formally, we define

$$X(S) = \bigcup_{o \in \lambda(S)} \bigcup_{T' \in \hat{\mathcal{X}}(S \setminus \{o\}) \text{ with } o \in \eta(T')} T' + o.$$ 

Define $\hat{X}(S)$ in a similar way as done for defining $\hat{\mathcal{X}}(S)$. The following result establishes a relation between $\hat{\mathcal{X}}(S)$ and $\hat{X}(S)$.

**Proposition 3.3** $\hat{\mathcal{X}}(S) \supseteq \hat{X}(S)$.

**Proof.** By definition $\hat{\mathcal{X}}(S) \supseteq \hat{X}(S)$. Let $T \in \hat{\mathcal{X}}(S)$ and suppose $T \notin \hat{X}(S)$ and define $T'$ such that $T = T' + o$. Accordingly, $T' \in \hat{\mathcal{X}}(S')$ where $S' = S \setminus \{o\}$ and $T' \notin \hat{\mathcal{X}}(S')$. Since $T' \notin \hat{\mathcal{X}}(S')$ but $T' \in \hat{\mathcal{X}}(S')$ we conclude that there exists an ordered subsequence $T'' \in \Xi(S'')$ of $T'$ for which $T'' \notin \hat{\mathcal{X}}(S'')$, so we conclude that $T \notin \hat{\mathcal{X}}(S)$ and thereby $\hat{\mathcal{X}}(S) = \hat{X}(S)$. \hfill $\Box$

### 3.3 Dynamic Programming formulation for the job-shop scheduling problem.

We have finally gathered all the necessary elements to propose a Dynamic Programming approach for the job-shop scheduling problem.

Considering $\hat{\mathcal{X}}(S)$ and $\hat{X}(S)$ ($S \subseteq \mathcal{O}$) as defined above, we can introduce the following Bellman equation for the job-shop scheduling problem:

$$\begin{align*}
\hat{\mathcal{X}}(\{o\}) &= \{T\} \text{ where } T = o, \\
\hat{\mathcal{X}}(S) &= \hat{X}(S), \quad S \subseteq \mathcal{O}
\end{align*}$$

Based on this equation we propose Algorithm 3.1 for building the set $\hat{\mathcal{X}}(\mathcal{O})$ that is, to find the optimal solution to the problem.

---

**Algorithm 3.1** Dynamic Programming for the job-shop scheduling problem

*Input:* An instance of the problem with $n$ jobs and $m$ machines

*Assume:* $\hat{X}(S) = \emptyset$ for all $S$

*Output:* A sequence $T$ associated with an optimal schedule to the problem and its makespan $C_{\text{max}}(T)$

for all $o \in \varepsilon(\emptyset)$ do

$X(\{o\}) = \{T\}$ with $T = o$

for $l = 1$ to $n \cdot m$ do // for all sequence lengths

for all $S \subseteq \mathcal{O} : |S| = l$ do

for all $T_1 \in \hat{X}(S)$ do

for all $o \in \eta(T)$ do

$T'_1 = T_1 + o$

if $T'_1 \notin T_2$ for all $T_2 \in \hat{X}(S \cup \{o\})$ then

for all $T_2 \in \hat{X}(S \cup \{o\})$ do

if $T'_1 < T_2$ then

$X(S \cup \{o\}) = \hat{X}(S \cup \{o\}) \setminus \{T_2\}$

$X(S \cup \{o\}) = \hat{X}(S \cup \{o\}) \cup \{T'_1\}$

End if

End for

End if

End for

End for

End for

End for

End for

End for

End for

End for

End for

End for

End for

End for

End for

End for

End for

Let $T$ be such that $\{T\} = \hat{X}(\mathcal{O})$ // $\hat{X} = \hat{\mathcal{X}}(\mathcal{O})$ and $|\hat{\mathcal{X}}(\mathcal{O})| = 1$

return $T$ and $C_{\text{max}}(T)$
3.3 Reducing the state space. In this section we propose a reduction in the state space of our Dynamic Programming approach for the job-shop scheduling problem.

Consider the additional notation \( \varepsilon(S,i) = \{ o \in \varepsilon(S) : m(o) = i \} \). \( \varepsilon(S,i) \) contains all the possible expansions of \( S \) that are associated with machine \( i \) \( (i = 1,\ldots,m) \). Naturally, \( \bigcup_{i=1}^{m} \varepsilon(S,i) = \varepsilon(S) \). The following result holds.

**Proposition 3.4** Let \( T \in \Xi(S) \). If \( \exists i \in \{1,\ldots,m\} \) such that

i) \( \exists o^* \in \varepsilon(S,i) : i \notin \eta(T) \) and

ii) \( \forall o \in \varepsilon(S,i), \xi(T,o) = C_{\text{max}}(T) + p(o) \)

then there exists one optimal solution to the problem such the corresponding ordered sequence does not start with \( T \).

**Proof.** Consider that there is an optimal solution such that the corresponding ordered sequence results from a completion of \( T \). One of the operations in the completion is \( o^* \) that appears after all operations in \( T \). This means that in the ordered sequence associated with the optimal solution, \( \psi(o^*) + p(o^*) \geq C_{\text{max}}(T) \). However, in the conditions of the proposition it is possible to schedule \( o^* \) starting at time \( \psi(T,o) \) ending in time \( \psi(T,o) + p(o^*) \leq C_{\text{max}}(T) \). With this change, it might be possible to schedule earlier the remaining operations in the completion of \( T \). By reordering the sequence thus obtained we obtain a new ordered sequence that does not increase the optimal makespan (so is optimal) and does not have \( T \) as an ordered partial sequence. \( \square \)

**Remark 3.1** For the operations \( o \) that satisfy the second condition of Proposition 3.4 we cannot conclude that \( o \notin \eta(T) \). In fact it can be the case that \( \psi(T,o) = C_{\text{max}}(T) \).

**Proposition 3.5** If we change the definitions of \( \hat{\Xi}(S) \) and \( X(S) \) by removing from these sets every \( T \in \Xi(S) \) satisfying Proposition 3.4, the Bellman equation still gives an optimal solution.

**Proof.** According to the conditions stated in Proposition 3.4 only possible optimal solutions are removed where it is possible to change the schedule by advancing at least a single operation \( o^* \) as defined in Proposition 3.4). For the optimal solution found according to the proof of Proposition 3.2 no operation can be advanced without losing the feasibility of the schedule, so this optimal solution is not removed. \( \square \)

This state space reduction can be easily incorporated in Algorithm 3.1 by checking the conditions of Proposition 3.4 for each \( T_1 \). When these conditions are satisfied do not expand \( T_1 \) any further. Note that \( T_1 \) has to be added to \( X(S) \) to enable it to discard other sequences (when \( T_1 \leq T_2 \)).

4. Complexity analysis In this section we study the complexity of Algorithm 3.1 and present an upper bound on this complexity. In this analysis, we are not taking into account the state space reduction presented in Section 3.3, thereby also giving an upper bound for the complexity of Algorithm 3.1 with the state space reduction.

4.1 The complexity of our algorithm. Considering the main body of Algorithm 3.1 we see that the first two for loops together loop over all sets \( S \subseteq O \). This is at most \( 2^{nm} \). As we will see below, this value can be decreased. Next there is a loop over all elements \( T_1 \in \hat{\Xi}(S) \), after which all expansions \( o \in \eta(T_1) \) are made. Inside these loops we have to determine \( \xi(T_1',o) \) for \( o \in \varepsilon(S \cup \{o\}) \), and we need to compare \( T_1' \) with all \( T_2 \in \hat{\Xi}(S \cup \{o\}) \) which is presented in Algorithm 3.1 as looping twice but can be implemented by looping once over all elements of \( \hat{\Xi}(S \cup \{o\}) \). This leads to a complexity

\[
O \left( 2^{nm} \left| \hat{\Xi}(S) \right| ||\eta(T_1)|| \left( \left| \varepsilon(S \cup \{o\}) \right| + \left| \hat{\Xi}(S \cup \{o\}) \right| \right) \right)
\]

First we take a look into the number of subsets \( S \subseteq O \). Note that not all of these subsets have (feasible) ordered sequences associated with them, in which cases we have \( \Xi(S) = \emptyset \). This is the case when an operation of some job is in the set but one of the preceding operations of the same job is not in the set. In fact, each job impose a sequence of precedence relations which must be respected by an ordered sequence.
This gives us $n$ independent precedence sequences of length $m$. According to Gromicho et al. [11] this leads to a reduction of $(\frac{m+1}{2})^n$ in the possible subsets $S \subseteq O$. So we have at most $(\frac{m+1}{2})^n 2^m = (m+1)^n$ subsets $S \subseteq O$ for which $\Xi(S) \neq \emptyset$.

Furthermore, observe that $|\eta(T_1)| \leq n$ and $|\varepsilon(S \cup \{o\})| \leq n$.

Finally we need to determine $|\tilde{\Xi}(S)|$ and $|\tilde{\Xi}(S \cup \{o\})|$.

Let $c$ be the minimal value of $C_{\text{max}}(T)$ with $T \in \tilde{\Xi}(S)$ and let $T_L$ be the sequence such that $C_{\text{max}}(T_L) = c$. Observe that $\forall o \in \varepsilon(S)$ we have $c \leq \xi(T_L, o) \leq c + p_{\text{max}}$. Now we have $\forall T \in \tilde{\Xi}(S)$ that $C_{\text{max}} < c + p_{\text{max}}$, otherwise $T_L \preceq T$. Hence $\forall o \in \varepsilon(S) \forall T \in \tilde{\Xi}(S)$ we know that $c \leq \xi(T, o) < c + 2p_{\text{max}}$.

Considering the values of $\xi(T, o), \forall o \in \varepsilon(S)$ relative to $c$, we have $0 \leq \xi(T, o) - c < 2p_{\text{max}}$. These values can be represented with a subset of the multiset $S = k_1, \ldots, k_n$, with $k_i = 2p_{\text{max}} - 1$ for $i = 1, \ldots, n$. $S$ consists of $k_i = 2p_{\text{max}} - 1$ copies of $n$ different elements $x_i$, $i = 1, \ldots, n$, where $x_i$ is associated with job $i$. Denote by $\sigma(T)$ the subset associated with $T$. Thus, $\sigma(T) \subseteq S$ is composed by taking $\forall o \in \varepsilon(S)$, $\xi(T, o) - c$ copies of $x_i$, where $i = j(o)$. Now observe that for $T_1, T_2 \in \Xi(S)$, $T_1 \preceq T_2$ if and only if $\sigma(T_1) \subseteq \sigma(T_2)$ with $\sigma(T_1) = \sigma(T_2)$ when $T_1 \equiv T_2$. We conclude that $\forall T_1, T_2 \in \tilde{\Xi}(S)$ we have $\sigma(T_1) \not\subset \sigma(T_2)$ and $\sigma(T_1) \not\supset \sigma(T_2)$.

Before proceeding our analysis, we recall the concept of an Antichain (see Anderson [2] for further details).

**Definition 4.1** An Antichain is a collection of subsets of a set where no two elements of the collection are subsets of each other.

According to this definition, the collection of $\sigma(T)$, with $T \in \tilde{\Xi}(S)$, is an antichain. To make this text self-contained, we recall the following two results on antichains.

**Proposition 4.1** The largest antichain in the collection of all subsets of a multiset is smaller or equal to the largest rank number $N_i$ ($N_i$ the number of elements with rank $i$).

**Proof.** See [2].

**Proposition 4.2** The size of the largest rank for a multiset is equal to size of the middle rank $\lambda$ which is

$$N_\lambda \approx \left(\frac{2}{\pi}\right)^{\frac{1}{2}} \frac{\prod_{i} (k_i + 1)}{\sqrt{\frac{1}{4} \sum_{i} k_i (k_i + 2)}}$$

**Proof.** See [2].

In our case we have $k_i = 2p_{\text{max}} - 1$ for all $i$. Accordingly, we have $|\tilde{\Xi}(S)| \approx \left(\frac{2}{\pi}\right)^{\frac{1}{2}} \frac{(2p_{\text{max}})^n}{\sqrt{n((2p_{\text{max}})^2 - 1)}}$.

Therefore,

$$|\tilde{\Xi}(S)| = O \left(\frac{(2p_{\text{max}})^n}{\sqrt{n((2p_{\text{max}})^2 - 1)}}\right).$$

We can finally state and prove the following result.

**Proposition 4.3** Algorithm 3.1 has complexity

$$O \left((2p_{\text{max}})^{n-1} \left(n\sqrt{n} + (2p_{\text{max}})^2\right) (m+1)^n\right)$$

**Proof.** The value obtained for $|\tilde{\Xi}(S)|$ leads to the following value for the complexity of Algorithm 3.1.

$$O \left(\left(\frac{(2p_{\text{max}})^n}{\sqrt{n((2p_{\text{max}})^2 - 1)}} + n\right) \frac{n(2p_{\text{max}})^n}{\sqrt{n((2p_{\text{max}})^2 - 1)}} (m+1)^n\right)$$
From here we obtain successively

\[
O \left( \left( \frac{(2p_{\text{max}})^n}{\sqrt{n(2p_{\text{max}})^2}} + n \right) \frac{n(2p_{\text{max}})^n}{\sqrt{n(2p_{\text{max}})^2}} (m+1)^n \right)
\]

\[
O \left( \left( \frac{(2p_{\text{max}})^{n-1}}{\sqrt{n}} + n \right) \frac{n(2p_{\text{max}})^{n-1}}{\sqrt{n}} (m+1)^n \right)
\]

\[
O \left( \left( (2p_{\text{max}})^{2n-2} + n\sqrt{n(2p_{\text{max}})^{n-1}} \right) (m+1)^n \right)
\]

\[
O \left( (2p_{\text{max}})^{n-1} \left( \sqrt{n} + (2p_{\text{max}})^2 \right) (m+1)^n \right)
\]

With the analysis above, we were able to give an upper bound on \( |\mathcal{E}(S)| \) with \( S \subseteq \mathcal{O} \). This results in an upper bound on the complexity of Algorithm 3.1. Note, however, that the actual value of \( |\mathcal{E}(S)| \) is much lower and thus we strongly suspect that the actual complexity of our procedure is much lower. This is also evidenced by the computational results presented in section 5. Nevertheless, the value achieved represents already an important breakthrough as we show next.

4.2 Comparison with brute force. The following results holds.

**Proposition 4.4** For a fixed \( p_{\text{max}} \), the Dynamic Programming approached that we propose for the job-shop scheduling problem has a complexity that is exponentially smaller than brute-force in \( n \) and in \( m \).

**Proof.** Using Stirling’s approximation\(^1\) we can evaluate the ratio between the complexity associated with brute-force and the complexity of our procedure. We have:

\[
\frac{(n!)^m}{(2p_{\text{max}})^{n-1} \left( \sqrt{n} + (2p_{\text{max}})^2 \right) (m+1)^n} \approx \frac{\left( \frac{\sqrt{2\pi n}}{n} \left( \frac{e}{n} \right)^n \right)^m}{(2p_{\text{max}})^{n-1} \left( n\sqrt{n} + (2p_{\text{max}})^2 \right) (m+1)^n}
\]

\[
\approx \frac{2p_{\text{max}} \sqrt{2\pi m}}{(2p_{\text{max}})^2 + n\sqrt{n}} \left( \frac{\left( \frac{e}{n} \right)^m}{2p_{\text{max}}(m+1)} \right)^n
\]

This is clearly exponential in \( m \), but is only exponential in \( n \) if \( \frac{(\frac{e}{n})^m}{2p_{\text{max}}(m+1)} > 1 \), or equivalently, if \( (\frac{e}{n})^m > 2p_{\text{max}}(m+1) \). Accordingly, for a fixed \( p_{\text{max}} \) the result holds. \( \square \)

This result represents an important breakthrough because it shows that the new algorithm solves the job-shop scheduling problem with a complexity exponentially lower than ‘brute force’.

5. Computational analysis. In order to evaluate our complexity analysis of the new algorithm, we run some computational tests considering benchmark instances for the job-shop scheduling problem. In particular we considered instances fl06 and la01-ld05. The first instance was first proposed by Fisher and Thompson [9]. The other five were proposed by Lawrence [18]. These benchmark instances (among many others) are available in the OR Library [23].

We implemented the new algorithm including the state space reduction of Section 3.3 with C++ and the tests were performed on a a windows 64 bit machine with a 2.66 GHz CPU and 8 GB memory.

In Table 1 we can observe the results obtained. In this table, the first three columns contain the information defining the instance namely, the name, the number of jobs \( (n) \) and the number of machines \( (m) \). Columns 4 and 5 depict the computational resources requirements for the instances considered. In particular we present the CPU time in seconds for solving each instance as well as the memory required. Columns 6-9 contain the observed and estimated values for the maximal size of \( \mathcal{E}(S) \) (\( \max_S |\mathcal{E}(S)| \)) and for the total number of sequences in the state space \( \left( \bigcup_S \mathcal{E}(S) \right) \).

\(^1n! \approx \sqrt{2\pi n} \left( \frac{e}{n} \right)^n\)
As we can see, the number of sequences needed to solve each instance is much smaller than the estimation obtained in section 4, which gives strong evidence that the bounds presented in the previous section can be improved significantly. For the first instance we were able to run our algorithm without the state space reduction using 2.6 seconds and 8 MB, having values $16$ and $190,592$ for $\max \xi_{\mathcal{S}}$ and $\bigcup S \xi_{\mathcal{S}}$ respectively. According to these values even without state space reduction our complexity analysis is still very conservative.

We should stress here that although using an exponential amount of memory our algorithm can be implemented in such way that only the optimal value is found saving a constant factor in memory. Nevertheless, for the instances that we could run not only the optimal value but also an optimal solution was found.

We present our algorithm as one to find the optimal sequence of operations. From this sequence the optimal solution immediately follows. For example, in the case of instance $ft06$ we obtained the following optimal sequence:

$$o_3 o_1 o_2 o_7 o_9 o_4 o_8 o_6 o_{15} o_{10} o_{12} o_{13} o_5 o_{14} o_{11} o_{21} o_{16} o_{27} o_{18} o_{22} o_{17} o_{19} o_{24} o_{28} o_{20} o_{25} o_{30} o_{36} o_{23} \ o_{26} o_{31} o_{29} o_{32} o_{35} o_{34} o_{33}$$

This sequence corresponds with the solution in Figure 4. This solution has the well-known optimal value of 55 (see for instance [15]).

![Figure 4: Optimal solution of ft06](image-url)
6. Conclusion. In this paper we proposed a Dynamic Programming approach for the classical job-shop scheduling problem. The main achievements of this paper are twofold:

(i) We produce an optimal algorithm with complexity proven to be exponentially lower than exhaustive enumeration, which is to date the best complexity guarantee of an optimal algorithm for the job-shop scheduling problem that we are aware of. Furthermore, our complexity analysis is extremely conservative and our results show strong empirical evidence that they may be improved by a few orders of magnitude.

(ii) Despite the theoretical interest, our algorithm proves to work in practice by solving some moderate benchmark instances.

References


<table>
<thead>
<tr>
<th>Year</th>
<th>Authors</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>2005</td>
<td>Meindert Flikkema</td>
<td>Exploring service development for understanding Schumpeterian innovation in service firms: the deduction of special case criteria, pp. 43</td>
</tr>
<tr>
<td>2005</td>
<td>Enno Masurel</td>
<td>Use of Patent Information: Empirical Evidence from Innovative SME’s, pp. 18</td>
</tr>
<tr>
<td>2005</td>
<td>Rutger Kroon Iris F.A. Vis</td>
<td>Seriously Ceres?, pp. 23</td>
</tr>
<tr>
<td>2005</td>
<td>Lidewey van der Sluis Leon de Caluwé Antonie van Nistelrooij</td>
<td>An explorative analysis of the links between learning behavior and change orientation, pp. 18</td>
</tr>
<tr>
<td>2005</td>
<td>F.A.G. den Butter O.K. van Megchelen</td>
<td>Uitbesteden en innovatie in de bouw Het toenemend belang van de regie- en handelsfunctie, pp. 10</td>
</tr>
<tr>
<td>2005</td>
<td>Adriaan J. van der Welle Frank A.G. den Butter</td>
<td>Arbeidsmarktflexibiliteit in de EMU: een beleidsgericht literatuuroverzicht, pp. 31</td>
</tr>
<tr>
<td>2005</td>
<td>Marina van Geenhuizen Peter Nijkamp</td>
<td>Death of distance and agglomeration forces of firms in the urban E-economy An artificial intelligence approach using rough set analysis, pp. 23</td>
</tr>
<tr>
<td>2005</td>
<td>Iris F.A. Vis Maurice Bakker</td>
<td>Dispatching and layout rules at an automated container terminal, pp. 17</td>
</tr>
<tr>
<td>2005</td>
<td>M. Francesca Cracolici Peter Nijkamp</td>
<td>Attractiveness and Effectiveness of Competing Tourist Areas: A Study on Italian Provinces, pp. 53</td>
</tr>
<tr>
<td>2005</td>
<td>Tibert Verhagen Jaap Boter</td>
<td>The importance of website content in online purchasing across different types of products, pp. 28</td>
</tr>
<tr>
<td>2005</td>
<td>Marina van Geenhuizen Peter Nijkamp</td>
<td>Place-bound versus Footloose Firms in a Metropolitan Area, pp. 17</td>
</tr>
<tr>
<td>2005</td>
<td>Kim A.D. Treur Lidewey E.C. van der Sluis</td>
<td>The benefits of coaching for employees and their organizations, pp. 30</td>
</tr>
<tr>
<td>2005</td>
<td>Wim den Dekker Paul G.W. Jansen Claartje J. Vinkenburg</td>
<td>Dimensions of an individual global mindset, pp. 29</td>
</tr>
<tr>
<td>Year</td>
<td>Authors</td>
<td>Title</td>
</tr>
<tr>
<td>------</td>
<td>----------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>2006-2</td>
<td>Mediha Sahin, Marius Rietdijk, Peter Nijkamp</td>
<td>Ethnic employees’ behaviour vis-à-vis customers in the service sector</td>
</tr>
<tr>
<td>2006-7</td>
<td>Frank A.G. den Butter</td>
<td>The industrial organisation of economic policy preparation in the Netherlands</td>
</tr>
<tr>
<td>2006-8</td>
<td>Evgenia Motchenkova</td>
<td>Cost minimizing sequential punishment policies for repeat offenders</td>
</tr>
<tr>
<td>2006-10</td>
<td>Pieter W. Jansen</td>
<td>Did capital market convergence lower the effectiveness of the interest rate as a monetary policy tool?</td>
</tr>
<tr>
<td>2006-14</td>
<td>M. Francesca Cracolici, Miranda Cuffaro, Peter Nijkamp</td>
<td>Sustainable tourist development in Italian Holiday destination</td>
</tr>
</tbody>
</table>
2006-15  Simonetta Longhi  
         Peter Nijkamp  
Forecasting regional labor market developments under spatial heterogeneity and spatial correlation, 25 p.

2006-16  Mediha Sahin  
         Peter Nijkamp  
         Tüzin Baycan-Levent  
Migrant Entrepreneurship from the perspective of cultural diversity, 21 p.

2006-17  W.J. Wouter  
         Botzen  
         Philip S. Marey  
Does the ECB respond to the stock market? 23 p.

2006-18  Tüzin Baycan-Levent  
         Peter Nijkamp  

2006-19  Ginés Hernández-Cánovas  
         Johanna Koëter-Kant  

2006-20  Miranda Cuffaro  
         Maria Francesca Cracolici  
         Peter Nijkamp  
Economic convergence versus socio-economic convergence in space, 13 p.

2006-21  Mediha Sahin  
         Peter Nijkamp  
         Tüzin Baycan-Levent  
Multicultural diversity and migrant entrepreneurship: The case of the Netherlands, 29 p.
<table>
<thead>
<tr>
<th>Year</th>
<th>Authors</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>2007-2</td>
<td>Daniel Leliefeld, Evgenia Motchenkova</td>
<td>To protect in order to serve, adverse effects of leniency programs in view of industry asymmetry, 29 p.</td>
</tr>
<tr>
<td>2007-5</td>
<td>Bas P. Singer, Bart A.G. Bossink, Herman J.M. Vande Putte</td>
<td>Corporate Real estate and competitive strategy, 27 p.</td>
</tr>
<tr>
<td>2007-7</td>
<td>Stella Flytzani, Peter Nijkamp</td>
<td>Locus of control and cross-cultural adjustment of expatriate managers, 16 p.</td>
</tr>
<tr>
<td>2007-11</td>
<td>Tüzin Baycan-Levent, Peter Nijkamp</td>
<td>Migrant entrepreneurship in a diverse Europe: In search of sustainable development, 18 p.</td>
</tr>
<tr>
<td>2007-13</td>
<td>Miranda Cuffaro, Maria Francesca Cracolici, Peter Nijkamp</td>
<td>Measuring the performance of Italian regions on social and economic dimensions, 20 p.</td>
</tr>
<tr>
<td>Year</td>
<td>Authors</td>
<td>Title</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
2008-1  Maria T. Borzacchiello
        Irene Casas
        Biagio Ciuffo
        Peter Nijkamp

2008-2  Maura Soekijad
        Jeroen Walschots
        Marleen Huysman
        Congestion at the floating road? Negotiation in networked innovation, 38 p.

2008-3  Marlous Agterberg
        Bart van den Hooff
        Marleen Huysman
        Maura Soekijad
        Keeping the wheels turning: Multi-level dynamics in organizing networks of practice, 47 p.

2008-4  Marlous Agterberg
        Marleen Huysman
        Bart van den Hooff

2008-5  Bernd Heidergott
        Haralambie Leahu
        Differentiability of product measures, 35 p.

2008-6  Tibert Verhagen
        Frans Feldberg
        Bart van den Hooff
        Selmar Meents
        Explaining user adoption of virtual worlds: towards a multipurpose motivational model, 37 p.

2008-7  Masagus M. Ridhwan
        Peter Nijkamp
        Piet Rietveld
        Henri L.F. de Groot
        Regional development and monetary policy. A review of the role of monetary unions, capital mobility and locational effects, 27 p.

2008-8  Selmar Meents
        Tibert Verhagen

2008-9  Junbo Yu
        Peter Nijkamp
        China’s prospects as an innovative country: An industrial economics perspective, 27 p

2008-10 Junbo Yu
        Peter Nijkamp
        Ownership, r&d and productivity change: Assessing the catch-up in China’s high-tech industries, 31 p

2008-11 Elbert Dijkgraaf
        Raymond Gradus
        Environmental activism and dynamics of unit-based pricing systems, 18 p.

2008-12 Mark J. Koetse
        Jan Rouwendal
        Transport and welfare consequences of infrastructure investment: A case study for the Betuweroute, 24 p

2008-13 Marc D. Bahlmann
        Marleen H. Huysman
        Tom Elfring
        Peter Groenewegen
        Clusters as vehicles for entrepreneurial innovation and new idea generation – a critical assessment

2008-14 Soushi Suzuki
        Peter Nijkamp
        A generalized goals-achievement model in data envelopment analysis: An application to efficiency improvement in local government finance in Japan, 24 p.

2008-15 Tüzin Baycan-Levent
        External orientation of second generation migrant entrepreneurs. A sectoral
Peter Nijkamp and Mediha Sahin study on Amsterdam, 33 p.

2008-16 Enno Masurel Local shopkeepers’ associations and ethnic minority entrepreneurs, 21 p.


2008-23 Patricia van Hemert and Peter Nijkamp Thematic research prioritization in the EU and the Netherlands: an assessment on the basis of content analysis, 30 p.

<table>
<thead>
<tr>
<th>Title</th>
<th>Authors</th>
<th>Abstract</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Pipelines or global buzz? A micro-level approach towards the knowledge-based view of clusters</td>
<td>Marc D. Bahlmann, Marleen H. Huysman, Tom Elfring, Peter Groenewegen</td>
<td>Global Pipelines or global buzz? A micro-level approach towards the knowledge-based view of clusters, 33 p.</td>
</tr>
<tr>
<td>Between ambition and approach: Towards sustainable knowledge management in development organizations</td>
<td>Julie E. Ferguson, Marleen H. Huysman</td>
<td>Between ambition and approach: Towards sustainable knowledge management in development organizations, 33 p.</td>
</tr>
<tr>
<td>Trapped between antiquity and urbanism. A multi-criteria assessment model of the greater Cairo metropolitan area</td>
<td>Eric de Noronha Vaz, Mário Caetano, Peter Nijkamp</td>
<td>Trapped between antiquity and urbanism. A multi-criteria assessment model of the greater Cairo metropolitan area, 22 p.</td>
</tr>
<tr>
<td>Spatial analysis for policy evaluation of the rural world: Portuguese agriculture in the last decade</td>
<td>Eric de Noronha Vaz, Teresa de Noronha Vaz, Peter Nijkamp</td>
<td>Spatial analysis for policy evaluation of the rural world: Portuguese agriculture in the last decade, 16 p.</td>
</tr>
<tr>
<td>Multitasking in the rural world: Technological change and sustainability</td>
<td>Teresa de Noronha Vaz, Peter Nijkamp</td>
<td>Multitasking in the rural world: Technological change and sustainability, 20 p.</td>
</tr>
<tr>
<td>An operational information systems architecture for assessing sustainable transportation planning: Principles and design</td>
<td>Maria Teresa Borzacchiello, Vincenzo Torrieri, Peter Nijkamp</td>
<td>An operational information systems architecture for assessing sustainable transportation planning: Principles and design, 17 p.</td>
</tr>
<tr>
<td>A decision support system for real estate investment choice</td>
<td>Vincenzo Del Giudice, Pierfrancesco De Paola, Francesca Torrieri, Francesca Pagliari, Peter Nijkamp</td>
<td>A decision support system for real estate investment choice, 16 p.</td>
</tr>
<tr>
<td>IT companies in rough seas: Predictive factors for bankruptcy risk in Romania</td>
<td>Miruna Mazurencu Marinescu, Peter Nijkamp</td>
<td>IT companies in rough seas: Predictive factors for bankruptcy risk in Romania, 13 p.</td>
</tr>
<tr>
<td>Evaluating social science and humanities knowledge production: an exploratory analysis of dynamics in science systems</td>
<td>Patricia van Hemert, Peter Nijkamp, Jolanda Verbraak</td>
<td>Evaluating social science and humanities knowledge production: an exploratory analysis of dynamics in science systems, 20 p.</td>
</tr>
<tr>
<td>Year</td>
<td>Authors</td>
<td>Title</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>2009-15</td>
<td>André de Waal, Karima Kourtit, Peter Nijkamp</td>
<td>The relationship between the level of completeness of a strategic performance management system and perceived advantages and disadvantages, 19 p.</td>
</tr>
<tr>
<td>2009-20</td>
<td>Karima Kourtit, André de Waal, Peter Nijkamp</td>
<td>Strategic performance management and creative industry, 17 p.</td>
</tr>
<tr>
<td>2009-22</td>
<td>Roberta Capello, Peter Nijkamp</td>
<td>Regional growth and development theories revisited, 19 p.</td>
</tr>
<tr>
<td>2009-23</td>
<td>M. Francesca Cracolici, Miranda Cuffaro, Peter Nijkamp</td>
<td>Tourism sustainability and economic efficiency – a statistical analysis of Italian provinces, 14 p.</td>
</tr>
</tbody>
</table>
2009-28 Marco Percoco  
Peter Nijkamp  
Individual time preferences and social discounting in environmental projects, 24 p.

2009-29 Peter Nijkamp  
Maria Abreu  
Regional development theory, 12 p.

2009-30 Tüzin Baycan-Levent  
Peter Nijkamp  
7 FAQs in urban planning, 22 p.

2009-31 Aliye Ahu Gülümser  
Tüzin Baycan-Levent  
Peter Nijkamp  
Turkey’s rurality: A comparative analysis at the EU level, 22 p.

2009-32 Frank Bruinsma  
Karima Kourtit  
Peter Nijkamp  
An agent-based decision support model for the development of e-services in the tourist sector, 21 p.

2009-33 Mediha Sahin  
Peter Nijkamp  
Marius Rietdijk  
Cultural diversity and urban innovativeness: Personal and business characteristics of urban migrant entrepreneurs, 27 p.

2009-34 Peter Nijkamp  
Mediha Sahin  
Performance indicators of urban migrant entrepreneurship in the Netherlands, 28 p.

2009-35 Manfred M. Fischer  
Peter Nijkamp  
Entrepreneurship and regional development, 23 p.

2009-36 Faroek Lazrak  
Peter Nijkamp  
Piet Rietveld  
Jan Rouwendal  

2009-37 Enno Masurel  
Peter Nijkamp  
Bridging the gap between institutions of higher education and small and medium-size enterprises, 32 p.

2009-38 Francesca Medda  
Peter Nijkamp  
Piet Rietveld  
Dynamic effects of external and private transport costs on urban shape: A morphogenetic perspective, 17 p.

2009-39 Roberta Capello  
Peter Nijkamp  
Urban economics at a cross-yard: Recent theoretical and methodological directions and future challenges, 16 p.

2009-40 Enno Masurel  
Peter Nijkamp  
The low participation of urban migrant entrepreneurs: Reasons and perceptions of weak institutional embeddedness, 23 p.

2009-41 Patricia van Hemert  
Peter Nijkamp  

2009-42 Teresa de Noronha Vaz  
Peter Nijkamp  
Knowledge and innovation: The strings between global and local dimensions of sustainable growth, 16 p.

2009-43 Chiara M. Travisi  
Peter Nijkamp  

2009-44 Sander de Leeuw  
Logistics aspects of emergency preparedness in flood disaster prevention, 24 p.
<table>
<thead>
<tr>
<th>Year</th>
<th>Authors</th>
<th>Title</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009-47</td>
<td>Eveline van Leeuwen, Peter Nijkamp</td>
<td>A micro-simulation model for e-services in cultural heritage tourism, 23 p.</td>
<td></td>
</tr>
<tr>
<td>2009-48</td>
<td>Andrea Caragliu, Chiara Del Bo, Peter Nijkamp</td>
<td>Smart cities in Europe, 15 p.</td>
<td></td>
</tr>
<tr>
<td>2009-50</td>
<td>Eric de Noronha Vaz, João Pedro Bernardes, Peter Nijkamp</td>
<td>Past landscapes for the reconstruction of Roman land use: Eco-history tourism in the Algarve, 23 p.</td>
<td></td>
</tr>
<tr>
<td>2009-51</td>
<td>Eveline van Leeuwen, Peter Nijkamp, Teresa de Noronha Vaz</td>
<td>The Multi-functional use of urban green space, 12 p.</td>
<td></td>
</tr>
<tr>
<td>2009-52</td>
<td>Peter Bakker, Carl Koopmans, Peter Nijkamp</td>
<td>Appraisal of integrated transport policies, 20 p.</td>
<td></td>
</tr>
<tr>
<td>2009-55</td>
<td>Bas Straathof, Gert-Jan Linders</td>
<td>Europe’s internal market at fifty: Over the hill? 39 p.</td>
<td></td>
</tr>
</tbody>
</table>