JOB SHOP SCHEDULING WITH TWO JOBS AND NONREGULAR OBJECTIVE FUNCTIONS[1]

ALESSANDRO AGNETIS

Dipartimento di Ingegneria dell'Informazione, Università di Siena, via Roma 56, 53100 Siena, Italy.
agnetis@dii.unisi.it

PITU B. MIRCHANDANI

Systems and Industrial Engineering Dept., The University of Arizona, Tucson, AZ 85721, USA.
pitu@SIE.Arizona.EDU

DARIO PACCIARELLI

Dipartimento di Informatica e Automazione, Università di Roma Tre, Via della Vasca Navale 79, 00146 Roma, Italy.
pacciare@dia.uniroma3.it

ANDREA PACIFICI

Dipartimento di Informatica, Sistemi e Produzione, Università di "Tor Vergata", Via di Tor Vergata 110, 00133 Roma, Italy.
pacifici@disp.uniroma2.it

ABSTRACT

We consider the job shop scheduling problem with two jobs. We consider a broad class of non-regular, quasi-convex functions of the completion time of the two jobs. We show that the optimal solution, for this class of objective functions, can be computed in $O(r \log r + \log H)$ time, where r is the number of operation pairs using the same machine, and H is the maximum operation processing time.

RÉSUMÉ

Nous considérons le problème de job shop scheduling avec deux jobs. L'objectif est la minimisation de non-régulière quasi convexe fonctions de les temps des achèvement de les deux jobs. Le solution optimal pour ça classe de fonctions peut être calculée dans un temps $O(r \log r + \log H)$, où r c'est le nombre d'opération paires utilisant les mêmes machine, et H c'est les durée de la maximum opération.

1. INTRODUCTION

This paper deals with the job shop problem when two jobs have to be performed in the shop. The most efficient solution algorithm for this problem is Brucker's algorithm [4], which addresses the case in which the objective is the minimization of the makespan. Sotskov [7] has given a polynomial algorithm for the more general case in which one wants to minimize an arbitrary regular (i.e., nondecreasing) objective function in the completion times of the two jobs. In this paper we extend this analysis to include quasi-convex, nonregular objective functions. Unlike classical results on the job shop with two jobs, the optimal schedule may not be semi-active. Both the sum and the maximum of these two objective functions are considered as performance criteria. Our approach consists in first finding a set of nondominated solutions, i.e., schedules which are Pareto-optimal from the viewpoint of the two jobs, and then searching for an overall optimum among these solutions. Both steps are carried out in polynomial time, as long as all processing times are integers.

[1]Recd. Jly. 99, Revd. Feb. 00, Acc. Jly. 00.

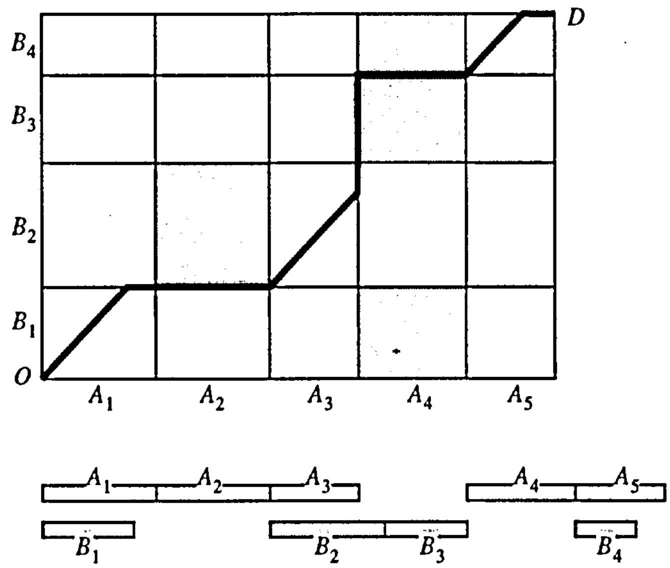

Figure 1: Grid representation of a two-jobs job shop, a path, and the corresponding Gantt chart.

The problem of scheduling two jobs has several applications in the robotics-FMS area (see for instance [1]). In fact, it models any situation in which two processes, each consisting of a sequence on nonpreemptive tasks, compete for using a set of shared resources. The problem is therefore to solve the possible conflicts in the most profitable way. The case of nonregular objective functions is of interest in just-in-time production, in which jobs should not be completed too early, to avoid downstream storage. In some cases, early completions must be avoided for technological reasons, as in the steelmaking-continuous casting production process. In fact, if the liquid steel is to be delivered at a certain time t for casting, it must retain a certain temperature, which may go lost if it is delivered too much time before t [6].

The plan of the paper is as follows. In Section 2, the notation and the preliminary definitions are introduced. In Section 3, we show that only a limited number of schedules need to be generated in order to find a global optimum. These schedules are conveniently represented by means of the cartesian scheme illustrated in Section 4. In Section 5 we concentrate on the aspects of line search related to the class of objective functions considered here. Finally, in Section 6, some conclusions are drawn.

We will make use of the grid representation used by several researchers [3, 8, 5, 4, 2] for the job shop problem with two jobs. For this reason we briefly recall it here. Let us consider two jobs J_A and J_B, both available for processing at time $t = 0$. Jobs J_A and J_B consist of operations $\{A_1, \ldots, A_{n_A}\}$ and $\{B_1, \ldots, B_{n_B}\}$ respectively. The operations of each job must be performed in sequence. Each operation requires a certain machine and a given processing time. Operation A_i (B_j) requires time p_{Ai} (p_{Bj}). We denote by $T(A)$ and $T(B)$ the total processing times of the two jobs respectively. If the operations

A_i and B_j require the same machine, they cannot be done in parallel, i.e., they form an *incompatible pair.* Completion time of job J_A (J_B) will be denoted by C_A (C_B).

Consider the two-axes plane, the horizontal axis corresponding to job J_A and the vertical to J_B. On each axis, the operations of the respective job are indicated by segments, the lengths being proportional to the corresponding durations. Parallel lines to these segments result in a *grid* in the plane (see, e.g., Figure 1). Let O be the origin of the axes and D the point $(T(A), T(B))$, that is, the upper-right point of the grid. On the grid, the rectangles corresponding to incompatible pairs will be referred to as *obstacles*, and we will denote the obstacle by means of the corresponding incompatible pair. In the pictorial representation of the grid the obstacles are shaded.

Any feasible schedule of the two jobs can be represented on the grid by a path from O to D consisting of horizontal, vertical and diagonal (45°) segments. The diagonal path segment in a rectangle implies that both operations are performed concurrently (using different machines). Clearly, it is not feasible to take a diagonal path through an obstacle. Horizontal (vertical) segments correspond to time periods during which only J_A (J_B) is processed, and the other job is waiting for J_A (J_B) to release the machine. In this case, a *conflict* occurs between two incompatible operations.

The upper-left and lower-right vertices of obstacle (A_i, B_j) will be referred to as $NW(A_i, B_j)$ and $SE(A_i, B_j)$ respectively. Note that a path hitting the obstacle (A_i, B_j) passes through either $NW(A_i, B_j)$ or $SE(A_i, B_j)$.

The makespan corresponding to a feasible path is given by $T(A)$ plus the total length of the vertical segments of the path (or, equivalently, by $T(B)$ plus the total length of the horizontal segments.) Hence, in order to minimize the makespan, the path must go diagonally whenever possible.

2. PROBLEM FORMULATION AND MINIMUM-SPAN SCHEDULES

In the problem addressed in this paper, each job has an associated *quasi-convex* function $f_i(C_i)$, $i = A, B$, of its completion time (A function $f(x) : \Re \to \Re$ is quasi-convex if for each $x, y \in \Re, \lambda \in [0,1]$, $f(\lambda x + (1-\lambda)y) \leq \max\{f(x), f(y)\}$.) We indicate with d_A and d_B the globally minimum point for $f_A(C_A)$ and $f_B(C_B)$ respectively. Typically d_A and d_B can be interpreted as due dates, and $f_i(C_i)$ as earliness/tardiness costs.

Let the cost pair associated with a schedule σ be $(f_A(C_A), f_B(C_B))$, and the cost pair associated with a schedule σ' be $(f_A(C'_A), f_B(C'_B))$. The cost pair $(f_A(C_A), f_B(C_B))$ *dominates* the cost pair $(f_A(C'_A), f_B(C'_B))$ if $f_A(C'_A) \geq f_A(C_A), f_B(C'_B) \geq f_B(C_B)$, and at least one of the two inequalities is strict.

We define a *nondominated schedule* as one for which there is no other schedule dominating it.

The problems we consider can be stated as:

Problem 1 *Given two jobs J_A and J_B and respective quasi-convex cost functions $f_A(C_A)$ and $f_B(C_B)$, find a feasible schedule such that $f_A(C_A) + f_B(C_B)$ is minimum.*

Problem 2 *Given two jobs J_A and J_B and respective quasi-convex cost functions $f_A(C_A)$ and $f_B(C_B)$, find a feasible schedule such that* $\max\{f_A(C_A), f_B(C_B)\}$ *is minimum.*

It is seen easily that in both problems an optimal schedule is among the set of nondominated schedules. We define the *span* of a schedule as the time between the beginning of the first operation and the end of the last operation of the schedule. A *minimum span schedule* is a schedule whose span is minimum. A *minimum makespan schedule* is a minimum span schedule where the first operation starts at time $t = 0$.

For any given schedule, we call *offset* the difference k between the completion times of the two jobs:

$$k = C_B - C_A. \tag{1}$$

A k-offset schedule is a schedule having offset k. A minimum span k-offset schedule is a k-offset schedule whose span is minimum. A minimum makespan k-offset schedule (from now on, k-mms) is a minimum span k-offset schedule where the first operation starts at time $t = 0$. We denote by $C_A(k)$ and $C_B(k)$ the completion times of the two jobs in a k-mms. Any minimum span k-offset schedule can be obtained by postponing a k-mms, that is, by increasing the starting time of all operations by the same amount. Also, we can always increase $|k|$ by postponing the last operation of the last completed job.

With reference to the grid representation of the problem, given a feasible k-offset schedule, the point D_k in which the corresponding path encounters the edge of the grid for the first time will be called *meeting point*. When $k > 0$, $D_k = (T(A), T(B) - k)$ lies on the right edge of the grid at distance k from point D (see point D_{k_q} in Figure 2). When $k < 0$, $D_k = (T(A) + k, T(B))$ lies on the upper edge of the grid, at distance $|k|$ from point D (see point D_{k_p} in Figure 2). Hence, a shortest path among those having meeting point D_k, represents a k-mms.

In the next section we show that in order to find the nondominated schedules, all we need is to compute a k-mms for $O(r)$ values of k, where r is the number of incompatible pairs.

3. SPECIAL VALUES OF k: BREAKPOINTS AND JUMPS

Consider a value of k, and the corresponding meeting point D_k on the grid. Let us start drawing a 45° line towards south-west, and stop the first time that either an obstacle (A_i, B_j) or the edge of the grid is hit. In the former case, we say that (A_i, B_j) is the *last obstacle* associated with k, and denote it as $\Omega(k)$. Note that if the path corresponding to a k-offset schedule passes through $NW(\Omega(k))$ or $SE(\Omega(k))$, no other conflicts occur after the last obstacle. If starting from D_k and going south-west, the edge of the grid is hit, a k-mms exists without conflicts, and we associate the *void last obstacle* to k.

A particular role is played by those values of k for which $\Omega(k)$ changes, i.e., $\Omega(k-\epsilon) \neq \Omega(k+\epsilon)$, for some arbitrarily small $\epsilon > 0$. We call *breakpoints* these values of k, and we refer to the two obstacles as $\Omega(k^-)$ and $\Omega(k^+)$ respectively. Hence, the breakpoints $k_1 < k_2 < \ldots$ partition $\Re$ into a set I of intervals $I_p = (k_p, k_{p+1})$ such that the same last obstacle is associated with all the values of k in the same interval.

The next proposition shows a simple but important property of k-mms schedules, when k is a breakpoint.

Proposition 1 *Consider a schedule σ in which the jobs have completion times C_A and C_B. There exists a breakpoint $\bar{k}$, such that $C_A(\bar{k}) \leq C_A$, and $C_B(\bar{k}) \leq C_B$.*

Proof

Given the schedule σ, and the completion times C_A and C_B, let $k = C_B - C_A$. Clearly if σ is not a k-mms there will be a k-mms such that $C_A(k) \leq C_A$ and $C_B(k) \leq C_B$. If k-mms is not a semi-active schedule we can make it semi-active by shifting operations backward, possibly changing its offset to $\hat{k}$, thus obtaining $C_A(\hat{k}) \leq C_A$ and $C_B(\hat{k}) \leq C_B$. From the last conflict ahead, the resulting schedule is therefore associated to a path $\hat{P}$ going from either $NW(\Omega(\hat{k}^+))$ or $SE(\Omega(\hat{k}^-))$ to $D_{\hat{k}}$ with a 45° line, without horizontal or vertical segments. Hence, $\hat{k}$ is a breakpoint, and the thesis follows. •

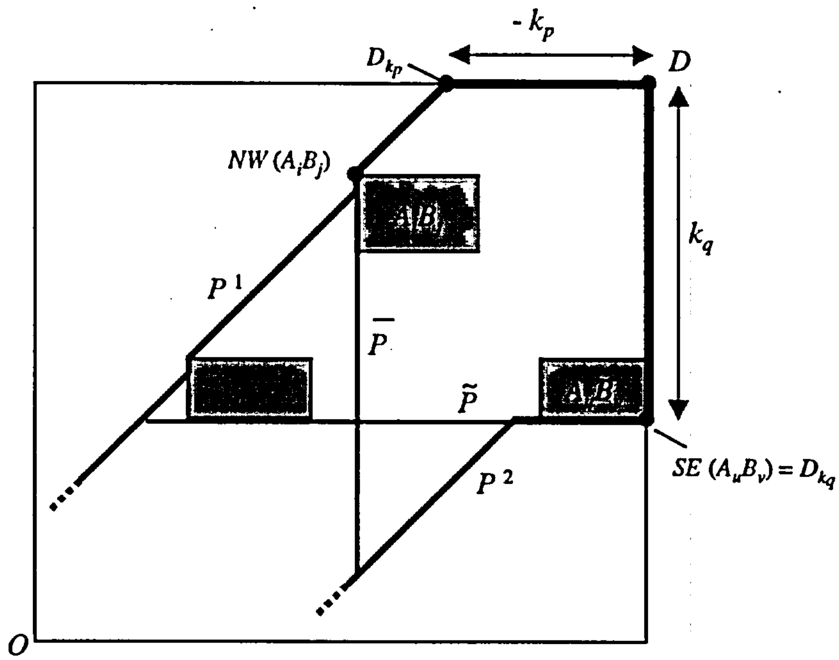

Figure 2: Paths P^1, P^2, $\bar{P}$ and $\tilde{P}$ in the proof of Theorem 1.

Some breakpoints deserve special attention. Let $I_{p-1} = (k_{p-1}, k_p) \in I$, with $k_{p-1} \geq 0$, and let $(A_u, B_v) = \Omega(k)$, for all $k \in I_{p-1}$. If $u = n_A$ (i.e., the last obstacle borders with the right edge of the grid), k_p is called *positive jump*. Let $I_q = (k_q, k_{q+1}) \in I$, with $k_{q+1} \leq 0$, and let $(A_i, B_j) = \Omega(k)$, for all $k \in I_q$. If $j = n_B$ (i.e., the last obstacle borders with the upper edge of the grid), k_q is called *negative jump*. Finally, consider $I_s = (k_s, k_{s+1})$ with $k_s < 0$ and $k_{s+1} > 0$. If $\Omega(k) = (A_{n_A}, B_{n_B})$ for all $k \in I_s$, then k_s and k_{s+1} are called *negative jump* and *positive jump* respectively. Note that in this case no k-offset schedules exist for $k_s < k < k_{s+1}$. In other words, a breakpoint $k_p > 0$ is a positive jump if $\Omega(k_p^-)$ borders with the right edge of the grid, and breakpoint $k_q < 0$ is a negative jump if $\Omega(k_q^+)$ borders with the upper edge of the grid.

The next theorem shows the relation between the values of the job completion times in minimum makespan k-offset schedules, for different values of k, when k is a breakpoint.

Theorem 1 *Let k_p and k_q be two breakpoints, with $k_q > k_p$. The following inequalities hold:*

(*i*) *If k_p and k_{p+1} are not both positive jumps then $C_B(k_p) \leq C_B(k_q)$.*

(*ii*) *If k_{q-1} and k_q are not both negative jumps then $C_A(k_p) \geq C_A(k_q)$.*

Proof

For a given path on the grid, the value of C_B is given by $T(B)$ plus the total length of the *horizontal* segments. Consider the two shortest paths on the grid corresponding to k_p-mms and k_q-mms, and call them P^1 and P^2 respectively. (Figure 2 shows the case in which k_p is a negative breakpoint and k_q is a positive jump.)

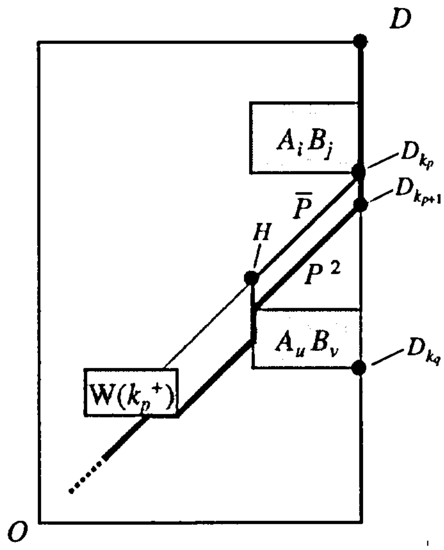

Figure 3: Path $\bar{P}$in the proof of Theorem 1 when k_p is a positive jump.

We first consider statement (i). Let (A_i, B_j) be the last obstacle met by path P^1. Assume first that k_p is not a positive jump. If P^1 passes through $NW(A_i, B_j)$ (i.e., $(A_i, B_j) = \Omega(k_p^+)$, as in Figure 2), we call $\bar{P}$ the path following P^2 up to the completion of operation A_{i-1}, then going vertically until it meets P^1 and henceforth following P^1 until D. If, on the other hand, P^1 passes through $SE(A_i, B_j)$ (i.e., $(A_i, B_j) = \Omega(k_p^-)$), call $\bar{P}$ the path following P^2 up to the completion of operation A_i, then going vertically until it meets P^1, at point $SE(A_i, B_j)$, and henceforth following P^1 until D. Suppose now that k_p is a positive jump, and k_{p+1} is not. Since k_{p+1} is not a jump, $\Omega(k_p^+)$ does not border with the right edge of the grid, and hence no obstacle intersects the segment from D_{k_p} to the point $H = (T(A) - A_{n_A}, T(B) - k_p - A_{n_A})$. In this case, we call $\bar{P}$ the path following P^2 up to the completion of operation A_{n_A-1}, then going vertically up to point H, henceforth going north-east at 45° to reach point D_{k_p} and finally going vertically until D (Figure 3). Note that in all the cases considered $\bar{P}$ is a k_p-offset schedule.

Let $\bar{C}_B$ be the value of C_B associated with the path $\bar{P}$. Since, up to meeting point D_{k_p}, $\bar{P}$ does not contain any more horizontal segments than P^2, $\bar{C}_B$ is not greater than $C_B(k_q)$. On the other hand, $\bar{P}$ defines a k_p-offset schedule, and hence $\bar{C}_B$ is greater than or equal to $C_B(k_p)$. This implies $C_B(k_p) \leq C_B(k_q)$.

By symmetrical arguments, it is possible to prove statement (ii). Let (A_u, B_v) be the last obstacle met by path P^1. Assume first that k_q is not a negative jump. If P^2 passes through $NW(A_u, B_v)$ (i.e., $(A_u, B_v) = \Omega(k_q^+)$), we call $\bar{P}$ the path following P^1 up to the completion of operation B_v, then going horizontally until it meets P^2 and henceforth following P^2 until D. If, on the other hand, P^2 passes through $SE(A_u, B_v)$ (i.e., $(A_u, B_v) = \Omega(k_q^-)$), call $\bar{P}$ the path following P^1 up to the completion of operation B_{v-1}, then going horizontally until it meets P^2 and henceforth following P^2 until D. Suppose now that k_q is a negative jump, and k_{q-1} is not. Since k_{q-1} is not a jump, $\Omega(k_q^-)$ does not border with the upper edge of the grid, and hence no obstacle intersects the segment from D_{k_q} to the point $H' = (T(A) - k_q - B_{n_B}, T(B) - B_{n_B})$. In this case, we call $\bar{P}$ the path following P^1 up to the completion of operation B_{n_B-1}, then going horizontally up to point H', henceforth going north-east at 45° to reach point D_{k_q} and finally going horizontally until D. Note that in all the cases considered $\bar{P}$ is a k_q-offset schedule. Let $\bar{C_A}$ be the value of C_A associated with the path $\bar{P}$. Since, up to meeting point D_{k_q}, $\bar{P}$ does not contain any more vertical segments than P^1, $\bar{C_A}$ is not greater than $C_A(k_p)$. On the other hand, $\bar{P}$ defines a k_q-offset schedule, and hence $\bar{C_A}$ is greater than or equal to $C_A(k_q)$. This implies $C_A(k_q) \leq C_A(k_p)$. •

A straightforward but important consequence of Theorem 1 is expressed by the following corollary.

Corollary 1 *Let k_p and k_{p+1} be two consecutive breakpoints, $k_p < k_{p+1}$, with k_p not a negative jump and k_{p+1} not a positive jump. Then, there are no feasible schedules having completion times C_A and C_B such that $C_A < C_A(k_p)$ and $C_B < C_B(k_{p+1})$.*

Proof

By contradiction, assume that a schedule σ exists having completion times C_A and C_B such that $C_A < C_A(k_p)$ and $C_B < C_B(k_{p+1})$. From Proposition 1, there must be a breakpoint $\bar{k}$, such that $C_A(\bar{k}) \leq C_A$, and $C_B(\bar{k}) \leq C_B$. Hence, $\bar{k}$ differs from k_p and k_{p+1}.

If $\bar{k} > k_{p+1}$, from Theorem 1 we have $C_B(k_{p+1}) \leq C_B(\bar{k})$, and therefore $C_B < C_B(\bar{k})$. If $\bar{k} < k_p$, from Theorem 1 we have $C_A(k_p) \leq C_A(\bar{k})$, and therefore $C_A < C_A(\bar{k})$. In both cases a contradiction follows. •

As long as the conditions of Corollary 1 hold, for $k_p \leq k \leq k_{p+1}$, no k-mms can have both C_A and C_B smaller than $C_A(k_p)$ and $C_B(k_{p+1})$ respectively. On the other hand, it can be easily shown that a k-mms with $C_A = C_A(k_p)$ and $C_B < C_B(k_{p+1})$ can be obtained from a k_p-mms by delaying operation B_{n_B}. (See Figure 4(a).) Similarly, a k-mms with $C_B = C_B(k_{p+1})$ and $C_A < C_A(k_p)$ can be obtained from a k_{p+1}-mms by delaying operation A_{n_A}. In any case, delaying these operations does not cause any conflict because $\Omega(k_p^+)$ $(= \Omega(k_{p+1}^-))$ is the last obstacle in any k-mms for $k_p < k < k_{p+1}$, and it does not border with the edge of the grid.

The situation is somewhat different when conditions of Corollary 1 do not hold.

Proposition 2 *Let k_p and k_{p+1} be two consecutive breakpoints, $k_p < k_{p+1}$.*

(*i*) *If k_{p+1} is a positive jump, then $C_A(k) = C_A(k_p)$ for $k_p \leq k < k_{p+1}$.*

(*ii*) *If k_p is a negative jump, then $C_B(k) = C_B(k_{p+1})$ for $k_p < k \leq k_{p+1}$.*

Proof

(*i*) Consider a positive jump k_{p+1}. It is possible to obtain a k-mms, $k_p \leq k < k_{p+1}$, with $C_A = C_A(k_p)$ and $C_B < C_B(k_{p+1})$, by delaying operation B_{n_B} in a k_p-mms. On the other hand, let $(A_{n_A}, B_j) = \Omega(k_{p+1}^-)$. In any schedule in which A_{n_A} is performed before B_j, the offset is at least $\sum_{h=j}^{n_B} p_{Bh} = k_{p+1}$. So if we want a k-mms with $k_p \leq k < k_{p+1}$, the path must pass through $NW(A_{n_A}, B_j)$, but the smallest value of C_A in a schedule passing through $NW(A_{n_A}, B_j)$ is $C_A(k_p)$. (See Figure 4(b).) Part (*ii*) follows from symmetrical considerations. •

In conclusion, in order to compute a k-mms for any value of k, it is sufficient to consider only the minimum makespan k-offset schedules when k is a breakpoint. This information can be obtained in $O(r \log r)$ (where r is the number of obstacles) by easily adapting the grid scanning procedure performed by Brucker's algorithm.

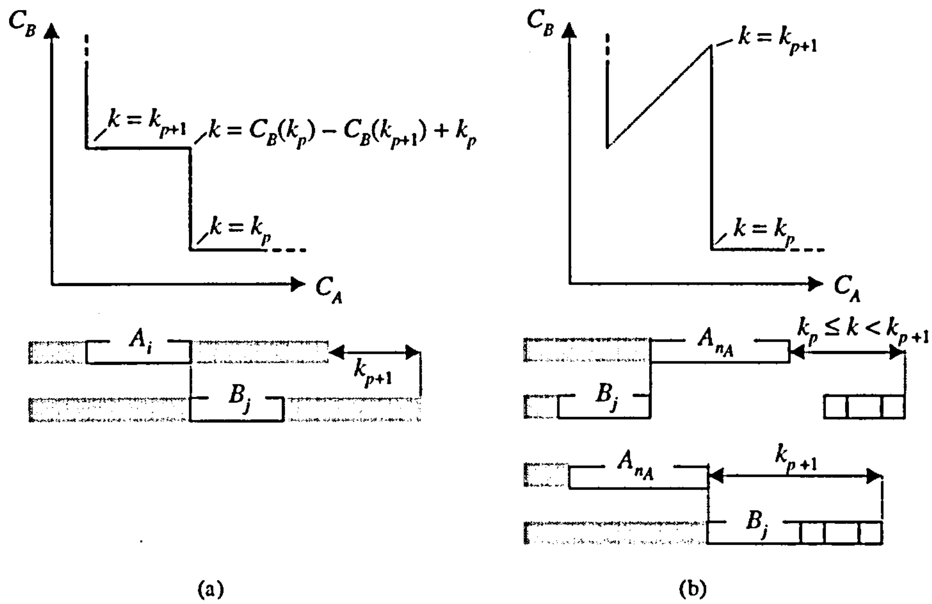

Figure 4: $C_A(k)$ and $C_B(k)$ between two consecutive breakpoints (a) and between a breakpoint and a positive jump (b).

Note that for any breakpoint k, a k-mms is semi-active. In fact, in such a schedule no task can be started earlier. For all the other values of k, any k-mms is not semi-active, since the path on the grid from the last obstacle to the meeting point includes a horizontal or a vertical segment.

It is important to observe that the property expressed in Theorem 1 (and hence Corollary 1) does not hold for consecutive jumps.

Remark 1 *Let k_p and k_{p+1} be two consecutive positive jumps, $k_{p+1} > k_p$. Then, $C_B(k_{p+1})$ may actually be smaller than $C_B(k_p)$. (A symmetrical result holds if k_p is a negative jump.)*

The following example illustrates Remark 1.

Example 1 *Consider the following jobs J_A and J_B, where (p_{Xi}, M_j) denotes that the i-th operation for job J_X requires processing time p_{Xi} on machine M_j:* $J_A = (7, M_1)$ $(30, M_3)$ $(33, M_2)$ $(15, M_1)$, $J_B = (5, M_3)$ $(9, M_1)$ $(8, M_3)$ $(5, M_2)$ $(6, M_3)$ $(6, M_2)$ $(1, M_1)$ $(5, M_3)$ $(4, M_2)$ $(5, M_1)$ $(1, M_2)$ $(3, M_1)$ $(5, M_2)$ $(7, M_1)$.

It is easy to verify that $k_p = 7$ and $k_q = 21$ are consecutive jumps and $C_B(k_q) = 124 < C_B(k_p) = 128$. (See Figure 5.)

4. NONDOMINATED SCHEDULES IN THE (C_A, C_B) PLANE

Recalling our definition of the scheduling problem (Section 2), in what follows d_A and d_B will be referred to as due dates, and $f_i(C_i)$ as earliness/tardiness costs. Clearly, the penalty for being tardy is nondecreasing with completion time and the penalty for being early is nonincreasing with completion time. This leads us to define the following

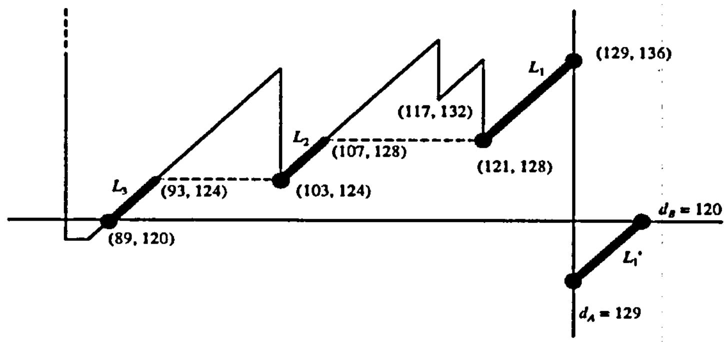

Figure 5: Set of nondominated points for Example 1 with $d_A = 129$ and $d_B = 120$.

zones in the (C_A, C_B) plane:

$$\begin{aligned} Z_I &= \{(C_A, C_B) | C_A < d_A, C_B < d_B\}, \\ Z_{II} &= \{(C_A, C_B) | C_A \leq d_A, C_B \geq d_B\}, \\ Z_{III} &= \{(C_A, C_B) | C_A \geq d_A, C_B \leq d_B\}, \\ Z_{IV} &= \{(C_A, C_B) | C_A > d_A, C_B > d_B\}. \end{aligned}$$

Hereafter, we characterize the points on the (C_A, C_B) plane which correspond to nondominated schedules. Let an *ideal schedule* be such that both jobs finish exactly at their due dates d_A and d_B. Obviously, an ideal schedule exists if and only if a $(d_B - d_A)$-mms exists and has length less or equal to $\max\{d_A, d_B\}$. In what follows we will assume that an ideal schedule does not exist.

Proposition 3 *Let $(\tilde{C}_A, \tilde{C}_B)$ be a point on the (C_A, C_B) plane corresponding to a feasible schedule $\tilde{\sigma}$. Then, $\tilde{\sigma}$ dominates any schedule corresponding to points (C_A, C_B) in the following cases:*

1. *$C_A \leq \tilde{C}_A, C_B \leq \tilde{C}_B$, and $(\tilde{C}_A, \tilde{C}_B)$ is in zone Z_I,*
2. *$C_A \leq \tilde{C}_A, C_B \geq \tilde{C}_B$, and $(\tilde{C}_A, \tilde{C}_B)$ is in zone Z_{II},*
3. *$C_A \geq \tilde{C}_A, C_B \leq \tilde{C}_B$, and $(\tilde{C}_A, \tilde{C}_B)$ is in zone Z_{III},*
4. *$C_A \geq \tilde{C}_A C_B \geq \tilde{C}_B$, and $(\tilde{C}_A, \tilde{C}_B)$ is in zone Z_{IV}.*

4.1 Zone Z_I

This zone does not contain nondominated schedules. In fact, a feasible schedule having (with C_A, C_B) as completion times $(C_A, C_B \in Z_I)$ is dominated by a schedule having either $C_A = d_A$ or $C_B = d_B$ and the same offset. This is obtained by simply shifting both jobs forward by $\min\{d_A - C_A, d_B - C_B\}$. Hence, Z_I does not contain nondominated schedules.

4.2 Zone Z_{IV}

In this zone, both jobs are late and hence objective functions are regular. From Corollary 1 all nondominated schedules in Z_{IV} are among k-mms's corresponding to breakpoints or jumps.

4.3 Zones Z_{II} and Z_{III}

Let now consider Zone Z_{II}. Obviously, a symmetrical discussion holds for Z_{III} by exchanging the role of the two jobs J_A and J_B. Let k^0 be the smallest offset such that there are feasible schedules in Z_{II} and let k^1 be the smallest breakpoint such that $k^1 \geq k^0$.

From Corollary 1, it follows that, if k^1 is a breakpoint and not a jump, then any k^1-mms dominates all the schedules with offset $k > k^1$ in Z_{II}. On the other hand, a k^1-mms is itself dominated by any schedule corresponding to the point $(d_A, C_B(k^1))$ on the (C_A, C_B) plane (note that $C_B(k^1) - d_A = k^0$).

Suppose now k^1 is a jump. Let us first examine the case $k^1 < 0$ (i.e., $C_A(k^1) > C_B(k^1)$). In this case, since an ideal schedule does not exist, by delaying A_{n_A} in a k^1-mms we can get a feasible schedule corresponding to the point $(d_A, C_B(k^1))$ on the (C_A, C_B) plane. This schedule is a k^0-mms and dominates all other schedules in zone Z_{II}. Let now $k^1 > 0$. We first show that in this case $k^1 = k^0$. Let $\hat{k}$ be the largest breakpoint smaller than k^1. By definition of k^1, it follows that $C_A(\hat{k}) > d_A$. From Proposition 2 we have that $C_A(k^0) = C_A(\hat{k})$, and therefore k^0 would not lie in zone Z_{II}, a contradiction. (See point (121, 128) in Figure 5).

We next observe that the point $(d_A, C_B(k^1))$ does not correspond to a feasible schedule (of course unless $C_A(k^1) = d_A$). In fact, since the path on the grid corresponding to a k^1-mms passes below an obstacle (A_{n_A}, B_j), we cannot increase C_A without increasing C_B by the same amount.

Consider now the segment L_1 having endpoints $(C_A(k^1), C_B(k^1))$ and $(d_A, d_A + k^1)$ (briefly, $L_1 = [(C_A(k^1), C_B(k^1)), (d_A, d_A + k^1)]$). Indeed, all the points of L_1 lying in Zone Z_{II} correspond to nondominated schedules (obtained simply by moving k^1-mms forward).

Recalling Remark 1, there *may* be other nondominated schedules with offset $k > k^1$. In fact, let $\tilde{k}$ be the smallest breakpoint such that $\tilde{k} > k^1$. We may have other jumps, at $k = k^2, k^3, \ldots, k^q$ with $k^1 < k^2 < k^3 < \ldots < k^q < \tilde{k}$ such that $C_B(k^1) > C_B(k^2) > \ldots > C_B(k^q)$. We define the semi-open segments $L_i = [(C_B(k^i) - k^i, C_B(k^i)), (C_B(k^{i-1}) - k^i, C_B(k^{i-1}))$, for $i = 2, \ldots, q$. In Figure 5 we represent the points corresponding to k-mms's in the (C_A, C_B) plane for k increasing from k^1 to k^q. From Proposition 2, as k increases between two jumps, C_A remains constant. Hence, Proposition 3 implies that the portions of segments $L_1, \ldots, L_q$ lying in zone Z_{II} correspond to all the nondominated schedules in zone Z_{II}.

4.4 The nondominated set

From the discussion carried out in the previous sections, it turns out that the offset of any nondominated schedule belongs to one of the following sets of values:

1. breakpoints,
2. values k such that $C_A(k) = d_A$ or $C_B(k) = d_B$.

Note that the schedules corresponding to the first and the third set are k-mms, and therefore a single schedule is associated with each value of k. This does not hold in general for the second set of values. In fact, when k is a jump, all the points of a

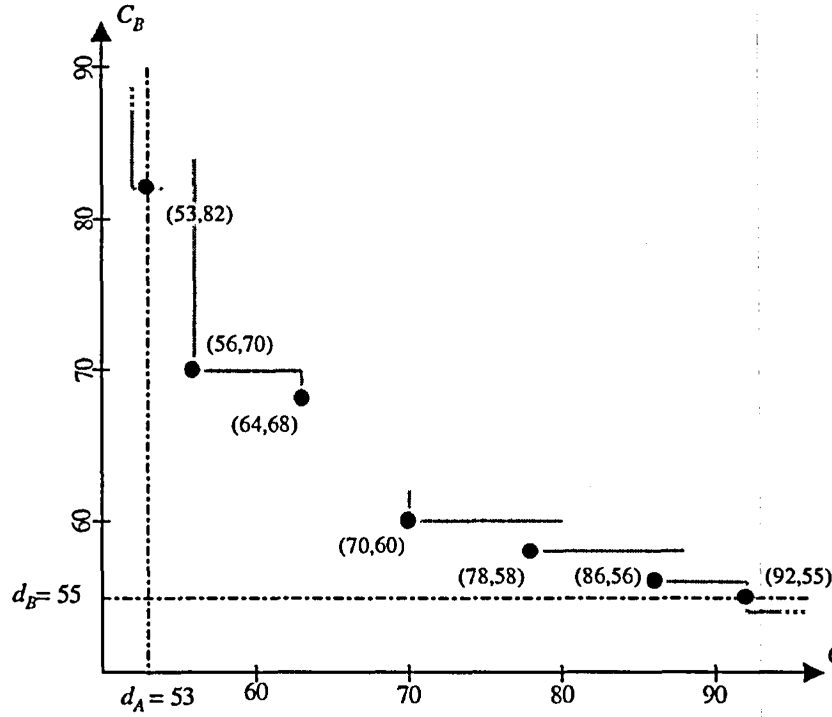

Figure 6: Set $\mathcal{N}$ when $d_A = 53$ and $d_B = 55$.

segment in zone Z_{II} or Z_{III} may correspond to nondominated schedules, all having the same offset k (recall Section 4.3). We call E the union of the above three sets.

Example 2 *Consider the following jobs J_A and J_B: $J_A = \{(10, M_2), (12, M_3), (6, M_1), (4, M_2), (6, M_1), (6, M_2), (8, M_1)\}$;*
$J_B = \{(4, M_3), (6, M_1), (4, M_2), (2, M_3), (8, M_2), (2, M_3), (6, M_1), (4, M_3), (6, M_2), (4, M_4), (4, M_3), (4, M_1)\}$.

Breakpoints are k = -38, -10, 10, 14, 30 with corresponding points $(C_A(k), C_B(k))$ $P_1 = (92, 54)$, $P_2 = (70, 60)$, $P_3 = (60, 70)$, $P_4 = (56, 70)$, $P_5 = (52, 82)$. Jumps are k = -30, -20, -8, 4, 28 with corresponding points $P_6 = (86, 56)$, $P_7 = (78, 58)$, $P_8 = (70, 62)$, $P_9 = (64, 68)$, $P_{10} = (54, 82)$. Note that no feasible schedule exists having offset $-8 < k < 4$.

Let us determine the set $\mathcal{N}$ of points corresponding to nondominated schedules for different values of the due dates d_A and d_B. Recall that k^0 is the minimum offset such that there is a feasible schedule in Z_{II}. Symmetrically, let $\underline{k}^0$ be the maximum offset such that there is a feasible schedule in Z_{III}.

$d_A = 53$, $d_B = 55$. In this case, $\mathcal{N}$ includes points P_2, P_4 (corresponding to breakpoints) and P_6, P_7, P_9 (corresponding to jumps). Moreover, $\mathcal{N}$ contains the points (92, 55), which dominates P_{10} and (53, 82), which dominates P_1. These two points lie on the borders Z_{II}/Z_{IV} and Z_{III}/Z_{IV} respectively. Note that $\mathcal{N}$ consists of a discrete set of points. In Figure 6, the shaded line indicates the points corresponding to k-mms, for k varying from $-\infty$ to $+\infty$.

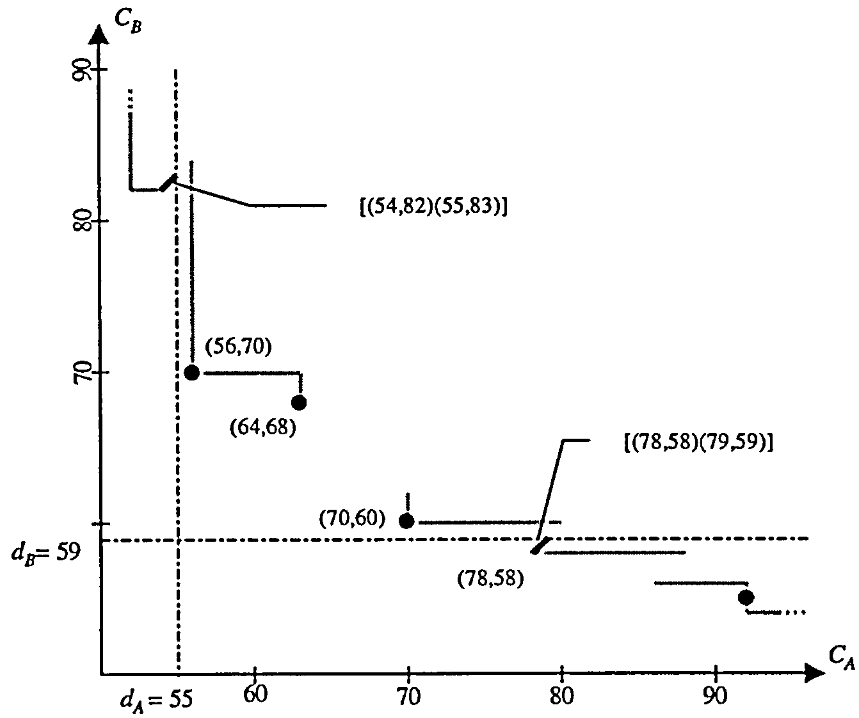

Figure 7: Set $\mathcal{N}$ when $d_A = 55$ and $d_B = 59$.

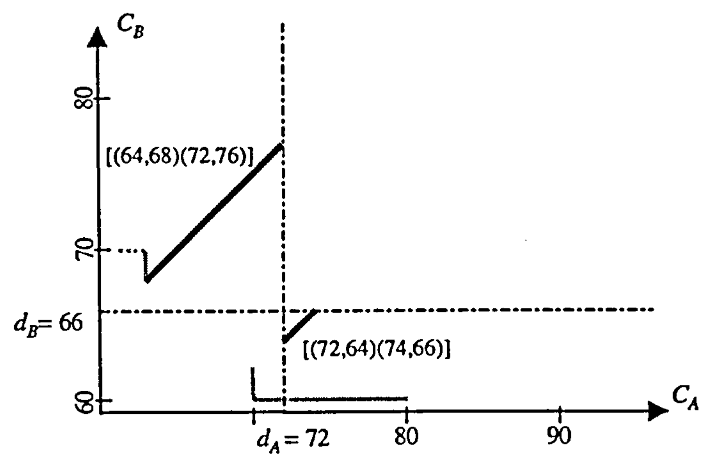

Figure 8. Set $\mathcal{N}$ when $d_A = 72$ and $d_B = 66$.

$d_A = 55$, $d_B = 59$. In this case, $\mathcal{N}$ includes points P_2, P_4 (breakpoints) and P_9 (jump). Moreover, $\mathcal{N}$ contains the segments $[(54, 82), (55, 83)]$ and $[(78, 58), (79, 59)]$ corresponding to minimum span k-offset schedules for $k = k^0 = 28$ and $k = \underline{k}^0 = -30$, respectively. (See Figure 7.)

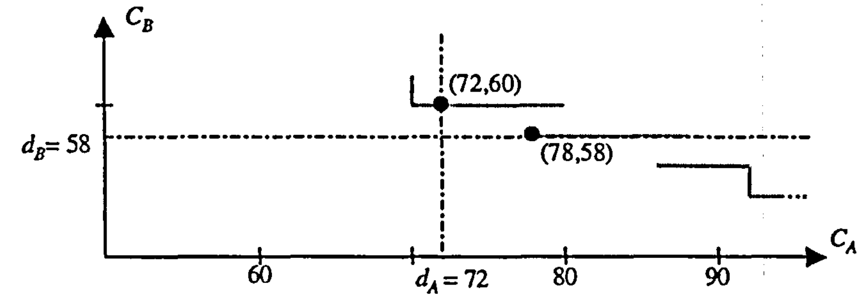

Figure 9: Set $\mathcal{N}$ when when $d_A = 72$ and $d_B = 58$.

$d_A = 72$, $d_B = 66$. Here, point P_8 falls in Z_I and all the other breakpoints and jumps in $Z_{II} \cup Z_{III}$. The set $\mathcal{N}$ only consists of segments. Namely, $[(64,68),(72,76)]$ in Z_{II}, and $[(72,64),(74,66)]$ in Z_{III}, corresponding to minimum span k-offset schedules for $k = k^0 = 4$ and $k = k^0 = -8$, respectively. Note that $(72,64)$ dominates P_8.(See Figure 8.)

$d_A = 72$, $d_B = 58$. In this case all the breakpoints and jumps fall in $Z_{II} \cup Z_{III}$. The set $\mathcal{N}$ only consists of two points, i.e., P_7 and point $(72,60)$ for which $k = k^0 = -12$ and $C_A(-12) = d_A$. Observe that all the points of the segment $[(78,58),(88,58)]$ correspond to k-mms's, with $-30 \le k \le -20$ but P_7 dominates them all. (See Figure 9.)

A different case is Example 1 in Section 2, letting the due dates be $d_A = 129$ and $d_B = 120$. Figure 5 shows the points corresponding to nondominated schedules. We have $k^0 = k^1 = 7$ corresponding to point $(121,128)$ in the (C_A, C_B) plane. The nondominated schedules in Z_{II} correspond to three segments, $L_1 = [(121,128),(129,136)], L_2 = [(103,124),(107,128))$ and $L_3 = [(89,120),(93,124))$. Notice that L_3 is obtained by removing from segment $[(87,118),(93,124))$ the portion belonging to zone Z_I. Also, note that the jump $(117,132) \in Z_{II}$ is dominated (by (121,128)). In accordance with Theorem 1, observe that the point (125,110), corresponding to the jump $k = -15$, is located south-east of the jump (121,128) for $k = 7$. The jump $k = -15$ lies in zone Z_I, and determines the segment $L_1' = [(129,114),(135,120)]$ of nondominated points in zone Z_{III}. The point (125,85) corresponding to breakpoint $k = -40$ (not shown in the figure) lies in zone Z_I and is dominated by the lower endpoint of segment L_1'. There are no breakpoints or jumps in zone Z_{IV}.

5. MINIMIZATION OVER NONDOMINATED SCHEDULES

In this section we consider the minimization of the cost functions defined in Problem 1:

$$F(C_A, C_B) = f_A(C_A) + f_B(C_B)$$

and Problem 2:

$$F(C_A, C_B) = \max\{f_A(C_A), f_B(C_B)\}.$$

Here we assume that C_A and C_B can take on only integer values.

If an ideal schedule exists, it is optimal for both problems. In the remainder of the section we assume that such a schedule does not exist. Clearly, we can limit ourselves

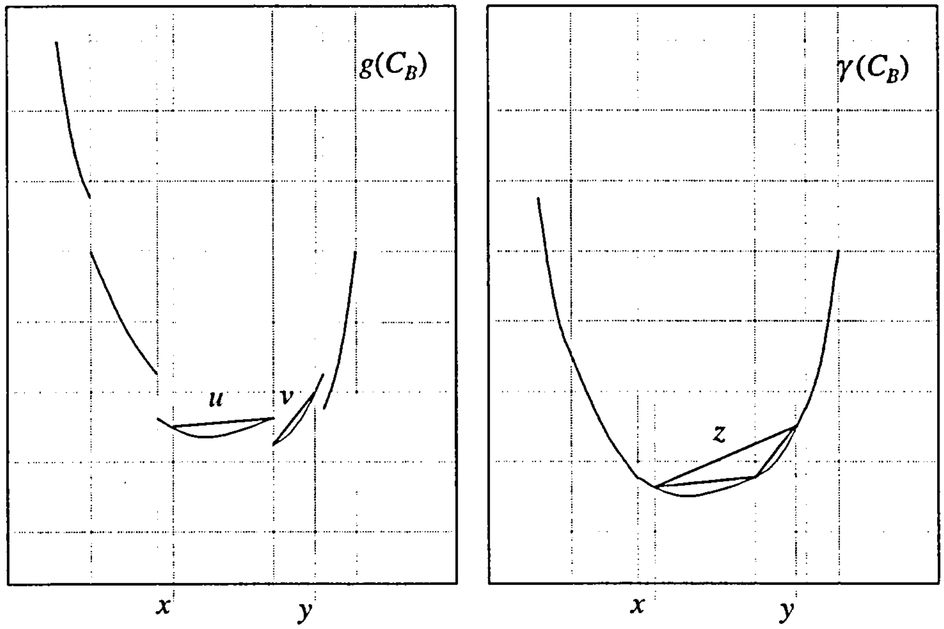

Figure 10: Curves $g(C_B)$ and $\gamma(C_B)$

to search for an optimal schedule among the nondominated schedules $\mathcal{N}$ for the given pair $(f_A(C_A), f_B(C_B))$.

As we showed in Section 4.4, the set $\mathcal{N}$ in general consists of some discrete points (in zone Z_{IV}) plus a set of line segments (in zones Z_{II} and Z_{III}). Once the function $F(C_A, C_B)$ has been evaluated at the discrete points of $\mathcal{N}$ in zone Z_{IV}, we must compute the minimum of $F(C_A, C_B)$ on the line segments of zones Z_{II} and Z_{III}. This means that a one-dimensional nonlinear optimization problem must be solved *for each segment.*

On the other hand, the relevant special case in which $f_A(C_A)$ and $f_B(C_B)$ are convex can be solved very efficiently. For this case we illustrate how to efficiently find the minimum of $F(C_A, C_B)$ on the segments of zone Z_{II}; an analogous discussion holds for zone Z_{III}.

Recall from Section 4.3 that we may have segments of nondominated points L_1, L_2, $\ldots, L_q$ with corresponding offset values $k^1, k^2, \ldots, k^q$. We need to optimize over these segments.

Note that C_B varies continuously between $C_B(k^q)$ and $d_A + k^1$ (see Figure 5), while C_A has discontinuities when jumping from a segment to the next. Moreover, observe that for each C_B such that $C_B(k^q) \leq C_B \leq d_A + k^1$, there is a unique C_A such that $(C_A, C_B) \in \mathcal{N}$. Hence, we can express $F(C_A, C_B)$ in $\mathcal{N}$ in terms of C_B only, namely, let $g(C_B) = F(C_A, C_B)$ with $(C_A, C_B) \in L_1 \cup L_2 \cup \ldots \cup L_q$.

Note that, even if $f_A(C_A)$ and $f_B(C_B)$ are convex, the function $g(C_B)$ is in general non-convex due to the discontinuities of F when jumping from a segment to the next (see Figure 10). Nevertheless, we next show how to efficiently minimize $g(C_B)$.

More precisely, for $C_B = C_B(k^i)$, $i = 2, \ldots, q$, the function g has a discontinuity. Let α_i be the value of the gap occurring at k^i, i.e.

$$\alpha_i = f_A(C_B(k^i) - k^{i+1}) - f_A(C_B(k^i) - k^i). \tag{2}$$

Let $\gamma(C_B)$ be the function obtained from g as follows: for $C_B(k^1) \leq C_B \leq d_A + k^1$, $\gamma(C_B) = g(C_B)$; for $C_B(k^i) \leq C_B \leq C_B(k^{i-1})$, $i = 2, \ldots, q$, $\gamma(C_B) = g(C_B) - \sum_{h=1}^{i-1} \alpha_h$. The function $\gamma(C_B)$ is obtained from g removing the gaps, thus obtaining a continuous curve (see Figure 10).

Theorem 2 *The function $\gamma(C_B)$ is convex for $C_B \in [C_B(k^q), d_A + k^1]$.*

Proof

We need to show that for any two values x and y such that $C_B(k^q) \leq x \leq y \leq d_A + k^1$, $\gamma(\lambda x + (1-\lambda)y) \leq \lambda\gamma(x) + (1-\lambda)\gamma(y)$ for each $\lambda \in [0,1]$. If $C_B(k^i) \leq x \leq y \leq C_B(k^{i-1})$ for some i, i.e., x and y belong to the same segment, the thesis follows from the convexity of F. Let us therefore consider the case in which x and y belong to different segments. We can limit ourselves to the case in which such segments are consecutive, i.e., $C_B(k^{i+1}) \leq x \leq C_B(k^i) \leq y \leq C_B(k^{i-1})$ for some i. Consider the cords u and v from $\gamma(x)$ to $\gamma(C_B(k^i))$ and from $\gamma(C_B(k^i))$ to $\gamma(y)$ respectively (see Figure 10). Clearly, since γ is convex in the interior of each segment, γ lies below u and v. If we denote by z the cord from $\gamma(x)$ to $\gamma(y)$, we need to show that γ lies below z. Due to the convexity of F, one has

$$\frac{\gamma(C_B(k^i)) - \gamma(x)}{C_B(k^i) - x} \leq \frac{\gamma(y) - \gamma(C_B(k^i))}{y - C_B(k^i)}$$

which implies that the cords u and v lie below z. •

Hence, the following can be shown easily:

Corollary 2 *The minimum of $g(C_B)$ is attained either for $C_B(k^i)$, $i = 1, \ldots, q$ or for a value of C_B that minimizes $\gamma(C_B)$.*

As a consequence of this result, we can find the value C_B that minimizes γ by means of a simple binary search over the interval $C_B(k^q) \leq C_B \leq d_A + k^1$.

Actually, we can restrict the search to only two consecutive segments. In fact, let us first compute the value of γ at all the jumps, and let $C_B(k^{i^*})$ correspond to the breakpoint which attains the minimum (this takes $O(n_B)$ time). Due to the convexity of γ, the minimum of $\gamma(C_B)$ is either in the segment $[C_B(k^{i^*}), C_B(k^{i^*-1}))$ or in $[C_B(k^{i^*+1}), C_B(k^{i^*}))$. Therefore the binary search needs to be conducted over these two intervals only.

Let us briefly discuss the complexity of the binary search. Let H_B indicate the maximum processing time of job J_B, that is $H_B = \max_i\{p_{Bi}\}$. Observing that the length of a segment L_i cannot exceed $H_B + p_{A,n_A}$, the width of the interval which must be searched in order to find the minimum of $\gamma(C_B)$ does not exceed $H_B + p_{A,n_A}$. Assuming that the value of $\gamma(C_B)$ for any given C_B can be computed in constant time, the minimum can be found in time $O(\log(H_B + p_{A,n_A}))$. Similarly, the width of any interval that needs to be searched in zone Z_{III} does not exceed $H_A + p_{B,n_B}$, where $H_A = \max_i\{p_{Ai}\}$, and the minimum can be found in time $O(\log(H_A + p_{B,n_B}))$.

In conclusion, the computational complexity to find an optimal schedule is $O(r \log r + \log H_A + \log H_B)$, which is polynomial (although not strongly polynomial) in the size of the problem instance.

Note that the whole discussion carried out for F sum of convex functions is still valid as long as the following properties hold:

$$\min_{C_A} F(C_A, \bar{C}_B) = F(d_A, \bar{C}_B) \text{ for all } \bar{C}_B$$

(C_A, C_B)	g	γ
(129, 136)	256	256
(121, 128)	64.8	64.8
(107, 128)	66.2	64.8
(103, 124)	66.2	64.8

Table 1:

$$\min_{C_B} F(\bar{C}_A, C_B) = F(\bar{C}_A, d_B) \text{ for all } \bar{C}_A$$

In particular, this is the case of Problem 2, with f_A, f_B being general quasi-convex functions.

Example 3 *Consider the two jobs from Example 1. Let the functions $f_A(C_A)$ and $f_B(C_B)$ be defined as:*

$$f_A(C_A) = \begin{cases} 12(107 - C_A) + 2.2 & C_A \leq 107 \\ 0.1(129 - C_A) & 107 \leq C_A \leq 129 \\ 12(C_A - 129) & C_A \geq 129 \end{cases}$$

$$f_B(C_B) = \begin{cases} 10(120 - C_B)^2 & C_B \leq 120 \\ (C_B - 120)^2 & C_B \geq 120 \end{cases} \tag{3}$$

and consider $F(C_A, C_B) = f_A(C_A) + f_B(C_B)$

Since there are no breakpoints in zone Z_{IV}, to find an optimal schedule we must search along the segments L_1, L_2, L_3 in Z_{II} and L_1' in Z_{III} (see Figure 5).

In the set $L_1 \cup L_2 \cup L_3$, C_B ranges between 120 and 136. In order to get γ, we need to compute the values α_1 and α_2. From (2), we have

$$\alpha_1 = f_A(107) - f_A(121) = 1.4$$
$$\alpha_2 = f_A(93) - f_A(103) = 120$$

We then compute the values given in Table 1:

Since γ did not decrease moving from the point (121, 128) to the point (103, 124), we can conclude that the minimum of γ occurs at some $C_B \in [124, 128) \cup [128, 136]$ and, hence, we need not compute g and γ at other candidate points in zone Z_{II} (i.e., points (93, 124) and (89, 120)). This minimum occurs at $C_B^* = 126$, yielding $g(C_B^*) = 62.2$ and $\gamma(C_B^*) = g(C_B^*) - \alpha_1 = 62.2 - 1.4 = 60.8$, and we can conclude that the optimal schedule in zone Z_{II} is the one corresponding to the point (105, 126). Notice that this schedule is *not* a minimum makespan k-offset schedule, but it is a minimum span k-offset schedule (with $k = 21$).

We now consider the line segment $L_1' = [(129, 114), (135, 120)]$ in zone Z_{III}. In this case, since there are no jumps, we can directly minimize g on this segment:

$$F(C_A, C_B) = 12(C_A - 129) + 10(120 - C_B)^2$$

becomes, since $k = -15$,

$$g(C_B) = 12(C_B - 114) + 10(120 - C_B)^2.$$

This function has its minimum at $C_B' = 119.4$, where $g(C_B') = 70.8$.

Comparing the best schedules in zones Z_{II} and Z_{III}, we conclude that the optimal solution is obtained at the point (105, 126).

6. CONCLUSIONS

In this paper we analyzed the complexity of the job shop problem with two jobs and a rather general class of objective functions. The complexity of the problem is substantially the same of the classical makespan minimization problem, as long as we accept a term which is linear in the number of bits required to encode the problem instance.

Further research in this area will address both theoretical extensions and practical applications. More theoretical work is needed to address even more general objective functions, e.g. functions depending also on the completion time of intermediate operations.

ACKNOWLEDGEMENTS

Prof. P.B.Mirchandani was partially supported by the Istituto di Analisi dei Sistemi e Informatica (IASI-CNR) – Roma. Prof. A. Agnetis, Dr. A. Pacifici, and Dr. D. Pacciarelli were partially supported by the MURST 5% project MSSD.

BIBLIOGRAPHY

[1] Agnetis, A., Lucertini, M. and Nicolò, F., Flow management in Flexible Manufacturing Cells with Pipeline Operations, *Management Science*, 1993, 39 (3) , 294–306.

[2] Agnetis, A. and Oriolo, G., The Machine Duplication Problem in a Job Shop with Two Jobs, *International Transactions on Operational Research*, 1995, 1 (2), 45–60.

[3] Akers, S.B. and Friedman, J., A non-numerical approach to production scheduling problems, *Operations Research*, 1955, 3 (4), 429–442.

[4] Brucker, P., An efficient algorithm for the Job-Shop problem with Two Jobs, *Computing*, 1988, 40, 353–359.

[5] Hardgrave, W.W. and Nemhauser, G.L., A geometric model and a graphical algorithm for a sequencing problem, *Operations Research*, 1963, 11(6), 889–900.

[6] Lixin Tang, Jiyin Liu, Aiying Rong, and Zihou Yang, A mathematical programming model for scheduling steemaking-continuous casting production, *European Journal of Operational Research*,2000, 120, 423–435.

[7] Sotskov, Y.N., The complexity of shop scheduling problems with two or three jobs, European Journal of Operational Research, 1991, 53, 322–336.

[8] Szwarc, W., Solution of the Akers-Friedman scheduling problem, *Operations Research*, 1960, 8(6), 782–788.

Alessandro Agnetis was born in Rome in 1963. He got his Laurea Degree in Electronic Engineering in 1987 and his Ph.D. in 1992 from the University of Rome "La Sapienza". In 1994 he became Assistant Professor in Industrial Automation in the same University, and since 1998 he is Associate Professor of Operations Research at the University of Siena. His research interests are mainly focused on combinatorial optimization models for production planning and scheduling. He published papers on several scientific journals, including Management Science, IEEE Transactions on Robotics and Automation, Discrete Applied Mathematics, European Journal of Operational Research.

Pitu Mirchandani, Salt River Project Professor of Technology, Public Policy and Markets, has joint appointments in Departments of Systems & Industrial Engineering and Electrical & Computer Engineering at the University of Arizona. He is also the Director of the ATLAS (Advanced Traffic and Logistics Algorithms and Systems) Research Center. He has BS/MS degrees in Engineering from UCLA and a ScD degree in Operations Research from MIT. His expertise includes logistics (scheduling, location, and routing), games and equilibria, and design of real-time decision/control systems, with application interests in transportation and spatial distribution of services and goods. He has over 70 publications including two books on Location Theory.

Dario Pacciarelli is assistant professor at Roma Tre University, Department of Computer Science and Automation. He received the Ph.D. in Operations Research from the University of Rome "La Sapienza" in 1996. He co-authored several publications in various refereed journals including European Journal of Operational Research, Discrete Applied Mathematics, IIE Transactions. Research interests include the analysis and the design of heuristics and exact algorithms, and the complexity analysis of algorithms and problems. Recent applications work is on routing and scheduling problems for manufacturing industries and rail companies.

Andrea Pacifici is assistant professor at the University of Rome "Tor Vergata", Department of Computer Science, Systems and Production. He got the Laurea degree in Electronic Engineering in 1993, and a Ph.D. in Operations Research in 1997, both at the University of Rome "La Sapienza" He co-authored several papers in the fields of Logistics, Production Planning and Scheduling.