A Time-Dependent Multiple Criteria Single-Machine Scheduling Problem

Kathrin Klamroth* Margaret M. Wiecek

Department of Mathematical Sciences
Clemson University
Clemson, SC 29634
USA

Abstract

We introduce a non-preemptive single-machine scheduling model with time-dependent multiple criteria. We formulate the problem as a knapsack problem and propose a dynamic-programming-based algorithm to finding all efficient schedules. An illustrative example is enclosed.

*On leave from the Department of Mathematics, University of Kaiserslautern, Kaiserslautern, Germany.

This work was partially supported by ONR Grant N00014-97-1-0784

1 Introduction

Preemptive and non-preemptive single-machine scheduling has been a subject actively researched from the point of view of different formulations and solution approaches. The element of time-dependency has been introduced to single-machine scheduling in various ways. Gascon and Leachman [7] examined a problem in which items to be scheduled on a single machine come with time-dependent demands and proposed a dynamic programming-based algorithm. Sousa and Wolsey [24] examined a time-indexed formulation in which the cost of a job was indexed by time. They proposed a cutting plane/branch-and-bound algorithm based on problem-related valid inequalities. This formulation was also analyzed in depth by van der Akker [26] and van der Akker et al. [27] who focused on polyhedral combinatorics and mixed-integer programming approaches to the problem. Single-machine scheduling with linear processing times of jobs was studied by Konov [16] and Cheng and Ding [4], [5]. The former analyzed the criterion of maximum lateness that makes the problem NP-hard while the latter showed that the makespan problem is strongly NP-complete. An overview on time-dependent scheduling can be found in Alidaee and Womer [1]. Gawiejnowicz [8] surveys more general discrete-continuous scheduling models including time-dependent processing times and other continuous resources.

Independently of time-dependency, researchers extended single objective models and studied bi-criteria single-machine scheduling problems. Typical criteria of interest were given by a combination of flowtime and a measure of tardiness. Among others, Hoogeveen and van de Velde [11] proved that the problem of minimizing total completion time and maximum cost is solvable in polynomial time. Hoogeveen [10] studied the criteria of maximum promptness and maximum lateness while Azizoglu et al. [2] examined the criteria of flowtime and maximum earliness. Gupta et al. [9] studied a problem with customer orders and multiple job classes in which the makespan and carrying costs of customer orders were minimized. Some authors made special assumptions about job processing times to achieve stronger results. For example, unit processing times were studied by Kondakci et al. [15].

In this paper, we propose a non-preemptive single machine scheduling model with time-dependent multiple criteria. According to the authors, such a model has not been studied in the literature before. Specifically, job processing times depend upon the times the jobs have been started and the minimization of the completion time may be one of the criteria of interest. All criteria to be optimized are, in general, monotone functions of time. Scheduling is subject to a machine capacity constraint. The formulation is based upon the classical knapsack problem extended by multiple criteria and time-dependency. Similarly to others (e.g., Sousa and Wolsey [24]) we do not enforce all jobs to be scheduled and therefore the model can be viewed as a selection problem rather than a scheduling problem (see McCormick [23]). Since the model extends the classical knapsack model, it is in general NP-hard (see e.g., Martello and Toth [21]).

Analyzing our model in the context of the multiple criteria knapsack problem (MCKP), we recognize that this problem has already become a known combinatorial optimization problem due to a wide range of applications. Since in many real world applications the preferences of multiple decision makers and/or various objectives have to be incorporated into the model, it is a natural extension of the classical knapsack model (for an overview see e.g., [20, 21]) to consider more than one criterion. Examples may be found in affordability analysis where projects have to be chosen with respect to more then a single criterion or in capital budgeting (see e.g., [3, 28, 19]). Teng and Tzeng [25] applied the multiple criteria multiple constraint knapsack problem (MCMCKP) to transportation investment planning. Multiple criteria knapsack problems were used by Kostreva et al. [17] to deal with relocation issues arising in conservation biology. The concept of time-dependency in the knapsack problem has not been much studied although in some applications the parameters of the problem may change in time. Consider a project selection problem in the presence of budget requirements and earnings that are time-dependent, or a loading problem with loading requests arriving stochastically over time and prices offered accordingly. Random and dynamic change of (single objective) knapsack problem parameters has been recently examined by Kleywegt and Papastavrou [14].

In this paper, we propose a dynamic programming (DP) approach to the proposed time-dependent multiple criteria scheduling problem (TDMCSP) following upon Villarreal and Karwan [29]. They were perhaps the only ones who proposed DP approaches to the MCMCKP. They proposed four approaches: two basic ones, an embedded state approach, and a hybrid approach. The first basic approach was very similar to Nemhauser and Garfinkel's [6] recursive equations (I) developed for the single objective single constraint knapsack problem while the second basic approach was a generalization of the recursive equations (III) developed by the same authors for the same problem [6] and by Ibaraki [12] as model (1). The two other approaches aimed at reducing the computational complexity of the basic approaches.

To develop a solution approach to the TDMCSP, we follow upon the second basic approach of Villarreal and Karwan [29]. While adapting this model to the time-dependent multiple objective case, we modify the time-dependent multiple criteria dynamic programming introduced by Kostreva and Wiecek [18]. They proposed two approaches and algorithms (backward and forward) to solving time-dependent multiple criteria routing problems by means of dynamic programming. The backward approach is computationally more complex but can handle general cost functions while the forward approach, although it requires a monotonicity assumption about the objective functions, is more efficient. Both approaches can be adapted to the TDMCSP.

We apply the forward approach as we believe it is more appealing due to its efficiency while the monotonicity assumption seems to naturally fit many decision making situations. If for some applications, however, this assumption was too constraining, the backward approach could be also modified to handle the TDMCSP.

In Section 2 we formulate the TDMCSP and in Section 3 we present a DP solution approach. We illustrate the algorithm with a tri-criteria example in Section 4. Section 5 concludes the paper.

2 Model formulation

Consider the classical problem of choosing projects over time in which the total associated benefit is to be optimized subject to resource constraints. Only one project can be selected at a time. Due to time-dependency, the selection problem becomes in fact a scheduling problem in which the projects (or jobs) are to be scheduled on a single machine. Motivated by this general problem, Sousa and Wolsey [24] used time-discretization which divides time into periods so that the benefit yielded by and the amount of resource used by each project depend upon the period the project has been selected. They considered the special case in which the resource availability in each period is the same and the amount of resource used by a project in a period is either zero or one. The model allowed to handle deadlines and release times for jobs.

In the model proposed in this paper, time is considered as a continuous variable upon which the benefit yielded by each job depends while the amount of resource used by a job is a positive constant.

Let $S = \{1, ..., n\}$ be an index set where j, $j \in S$, denotes a job. A schedule of jobs is defined as a sequence $x := \{x_r\}_{r=1}^p$ of elements $x_r \in S$, $r = 1, \ldots, p$ satisfying

$$x \in \{\{x_r\}_{r=1}^p \ : \ p \in I\!N, \ \ x_r \in S, r = 1, \ldots, p\}. \tag{1}$$

In a schedule, a job can be repeated or is to be executed only once. Only one job can be processed at the same time and all jobs are released at time 0. Furthermore, we assume that the jobs do not have due dates, and preemptions are not allowed.

Each job $j \in S$ has a weight a_j representing the job's usage of the resource (e.g., machine occupancy, cost). The total weight of a schedule, calculated as the sum of the weights of all jobs in the schedule, is represented by a weight function defined as

$$a(x) := \sum_{r=1}^{p} a_{x_r}, \tag{2}$$

where a_{x_r} is the weight coefficient of the job x_r in the schedule.

The total weight of a schedule cannot exceed a given capacity (budget) constraint induced by the machine

$$a(x) \leq b, \tag{3}$$

which is referred to as a *capacity constraint* (*budget constraint*).

We additionally assume that the weights $a_j, j \in S$, and the capacity b are positive integers. In order to avoid trivial solutions let $0 < a_j \leq b$, $j = 1, \ldots, n$.

A feasible schedule of jobs is consequently a sequence of elements in S such that the total weight of the schedule does not exceed the capacity. Let $X := \{x : a(x) \leq b\}$ be the set of all the feasible schedules of the TDMCSP. Note that due to the fact that all weight coefficients are positive integers, all feasible solutions in X are finite. Namely we get that $p \leq b$ for all $\{x_r\}_{r=1}^p \in X$.

The jobs are evaluated with respect to m benefits they yield. For every $j \in S$, let $c_j(t)$ be a unit vector benefit associated with the job j at time t. Elements $c_j^i(t)$, $i = 1, \ldots, m$, $j = 1, \ldots, n$, are defined to be real-valued functions of time t and are not assumed to be continuous. In particular, let

$c_j^1(t)$, $j = 1, \ldots, n$ be positive functions measuring the processing time of the job j if it has been started at time t. In the context of the project selection problem, $c_j^1(t)$ thus represents the duration of project j, $j = 1, \ldots, n$. Let the other components $c_j^i(t)$, $i = 2, \ldots, m$, $j = 1, \ldots, n$, represent the benefits of interest to the decision maker according to which the jobs are to be scheduled and processed (e.g., earnings, revenue, appreciation).

A benefit of a schedule is calculated as the vector sum of the benefits of all jobs in the schedule:

$$f_i(x) := \sum_{r=1}^{p} c^i_{x_r}(t^r(x)), \qquad i = 1, \ldots, m, \tag{4}$$

where t is a continuous variable, $t \geq 0$, and $t^r(x)$ represents the time at which the r-th job of the schedule x is started and is calculated as

$$\begin{aligned} t^1(x) &= 0, \\ t^{s+1}(x) &= t^s(x) + c^1_{x_s}(t^s(x)), \qquad s = 1, \ldots, p. \end{aligned} \tag{5}$$

In particular, the completion time of a schedule is represented by $f_1(x)$.
We formulate the TDMCSP as:

$$\begin{aligned} &\text{vmax}^* \quad f(x) = [f_1(x), f_2(x), \ldots, f_m(x)]^T \\ &\text{s.t.} \qquad a(x) \leq b. \end{aligned} \tag{6}$$

As we are interested in maximizing the objective functions $f_i(t)$, $i = 2, \ldots, m$ and in minimizing the completion time simultaneously, the operator $vmax^*$ in (6) denotes the maximization of $[-f_1(x), f_2(x), \ldots, f_m(x)]^T$, i.e.

$$\text{vmax}^* \, [f_1(x), f_2(x), \ldots, f_m(x)]^T := \text{vmax} \, [-f_1(x), f_2(x), \ldots, f_m(x)]^T. \tag{7}$$

However, the decision maker may choose not to minimize the completion time but only maximize the other criteria. In this situation the operator $vmax^* = vmax$ is applied only to criteria $[f_2(x), \ldots, f_m(x)]^T$.

Solving (6) is understood as generating its efficient (Pareto) schedules. A feasible schedule $\hat{x} \in X$ is said to be an efficient solution of (6) if there is no other feasible schedule $x \in X$ such that

$$f_1(x) \leq f_1(\hat{x}) \qquad \text{and} \qquad \forall i \in \{2, \ldots, m\} \;\; f_i(x) \geq f_i(\hat{x}) \tag{8}$$

with at least one strict inequality in (8).

Let $\mathcal{X}_e$ denote the set of efficient schedules of (6) and let $\mathcal{Y}_e$ denote the image of $\mathcal{X}_e$ in the objective space, that is $\mathcal{Y}_e = f(\mathcal{X}_e)$, where $f = [f_1, \ldots, f_m]^T$. $\mathcal{Y}_e$ is referred to as the set of nondominated criterion vectors (outcomes) of (6).

3 A dynamic programming approach

In the following it is convenient to consider the TDMCSP with the right-hand-side $k = 1, \ldots, b$ of the capacity constraint which is denoted by k-TDMCSP. A feasible schedule $x := \{x_r\}_{r=1}^p$ of the k-TDMCSP is a sequence of jobs $x_r \in S$, $r = 1, \ldots, p, p \leq k$, so that $\sum_{r=1}^p a_{x_r} = k$ for $k = 1, \ldots, b$. Given two feasible schedules, $x^s = \{x_r\}_{r=1}^s$ of the k_s-TDMCSP and $x^w = \{x_{s+r}\}_{r=1}^w$ of the k_w-TDMCSP, we define the concatenated schedule $\{x^s, x^w\} := \{x_r\}_{r=1}^{s+w}$ that is a feasible schedule of the $(k_s + k_w)$-TDMCSP.

Let a set of states Q be defined as

$$Q := \{q(0), q(1), \ldots, q(b)\},$$

where the initial state is defined to contain only the empty schedule $\{\,\}$ (independently from the weight function a and the objective coefficients)

$$q(0) = \{\{\,\}\}$$

and the state $q(k)$, $k = 1, \ldots, b$, represents all the feasible schedules of the k-TDMCSP, i.e.

$$q(k) := \{\{x_r\}_{r=1}^p \;:\; p \leq k,\;\; x_r \in S, r = 1, \ldots, p,\; \sum_{r=1}^p a_{x_r} = k\}.$$

Since there may occur efficient schedules in all the states, the set of final states Q_F is given by

$$Q_F := \{q(0), q(1), \ldots, q(b)\}.$$

The decision of adding a job $j \in S$ to a schedule of jobs $x \in q(k)$ results in an increase of the right-hand side k by a_j and thus corresponds to a transition of x from the state $q(k)$ to the state $q(k + a_j)$. Observe that with this definition of the states, the original problem is represented as a loop-free sequential decision process, i.e. a process whose states can be indexed from 0 to b, so that a transition from a state $q(k)$ always occurs to a state $q(l)$ such that $k < l$, for any $k, l = 0, 1, \ldots, b$ (see [12]).

Without loss of generality we assume that the system is in the state $q(0)$ at time $t = 0$. We also assume that there is no waiting time between adding two consecutive jobs to a feasible schedule, i.e. once a job j has been processed, another job $\bar{j}$ is being chosen and processed right after. For all $j = 1, \ldots, n$, $c_j^1(t) > 0$ represents the time needed to make a transition from a state $q(k)$ to a state $q(k + a_j)$ for $k = 0, 1, \ldots, b - a_j$ given that the system is in the state $q(k)$ at time t. The arrival time at the state $q(k + a_j)$ is equal to $t + c_j^1(t)$.

The states defined above and the possible transitions between them yield a network whose nodes and arcs are defined by these states and transitions, respectively. This network does not have any circuits since the states and the transitions form a loop-free decision process. Associated with every arc of this network is a vector cost $[c_j^1(t), \ldots, c_j^m(t)]^T$ related to adding the job $j \in S$ to a feasible schedule at time t.

Given the network, we are in the position to apply the forward approach of Kostreva and Wiecek [18]. They developed this approach to find the set of all efficient (shortest) paths from a given source node to every other node in the network whose links carried a time-dependent vector cost. They considered a general network whose every node could be connected to every other node. The costs were assumed to be real-valued positive and monotone increasing functions of time. These assumptions were necessary to establish the principle of optimality for dynamic multiple objective networks.

Given the special structure of our network, we may relax and change some of their assumptions due to the fact that the feasibility constraint of the TDMCSP yields a circuit-free network and that our problem involves maximization rather than minimization. In general, we can allow all the objective functions to be positive and/or negative functions of time no matter whether we pose a maximization or a minimization problem. In both cases the optimal objective function (vector) value will be necessarily bounded as we have a finite number of states (nodes), a finite number of transitions (arcs), and no circuits in the network. However, as we have chosen $c_j^1(t), j = 1, \ldots, n$ to represent the processing time, we require these functions to be positive, while the other functions $c_j^i(t), i = 2, \ldots, m, j = 1, \ldots, n$, representing general criteria of interest may be of any sign. In fact, a transition from a state

$q(k)$ at a given time t_1 to a state $q(l)$ may for some criterion $i, i \in \{2, \ldots, m\}$ involve a cost $c^i_j(t_1) < 0$, which means that adding the job j at time t_1 to a schedule is not beneficial at that time, however, the resulting schedule may still be efficient. Clearly, the same job j could be added at a time $t_2, t_2 \neq t_1$, so that the corresponding cost would be positive, $c^i_j(t_2) > 0$.

The following assumption is necessary to prove a principle of optimality for the TDMCSP.

Assumption 1 *For all $t_1, t_2 \geq 0$, if $t_1 \leq t_2$, then*

(a) $t_1 + c^1_j(t_1) \leq t_2 + c^1_j(t_2)$ for all $j = 1, \ldots, n$, and

(b) $c^i_j(t_1) \geq c^i_j(t_2)$ for all $i = 2, \ldots, m$, and $j = 1, \ldots, n$.

Assumption 1 (a) requires that if a job j is added to a schedule at time t_1 or at a later time t_2, then with the earlier starting time the next job may be considered to be added to the schedule earlier than with the later starting time. We observe that if $c^1_j(t), j = 1, \ldots, n$, are monotone increasing functions of time, then this assumption holds. Assumption 1 (b) simply requires that the other components of the objective functions $c^i_j(t)$ for all $i = 2, \ldots, m$ and $j = 1, \ldots, n$ be monotone decreasing functions of time. In the context of the model this implies that (for example) the benefits generated by a project (job) j decrease in time, or in other words, the later the job is processed in a schedule the less benefits it brings. Both assumptions represent the typical requirement of the project selection problem that the earlier the project is started, the earlier it is completed and the bigger benefits it brings.

Theorem 1 *Principle of Optimality for the TDMCSP.*
Under Assumption 1, an efficient solution sequence of jobs $x^p = \{x_r\}_{r=1}^p$ of the k-TDMCSP completed at time $t^{p+1}(x^p)$ has the property that each solution subsequence $x^s = \{x_r\}_{r=1}^s, 1 \leq s \leq p$ completed at time $t^{s+1}(x^s), t^{s+1}(x^s) \leq t^{p+1}(x^p)$, is an efficient solution sequence of jobs for the $(\sum_{r=1}^s a_{x_r})$-TDMCSP.

Proof:
Let x^p be an efficient solution sequence of the k-TDMCSP completed at time $t^{p+1}(x^p)$. Assume to the contrary that a solution subsequence x^s of the solution sequence x^p, $x^s = \{x_r\}_{r=1}^s, 1 \leq s \leq p$, completed at time $t^{s+1}(x^s)$,

$t^{s+1}(x^s) \leq t^{p+1}(x^p)$, is not an efficient solution sequence of the $(\sum_{r=1}^{s} a_{x_r})$-TDMCSP. Then there exists another solution sequence $y^u = \{y_r\}_{r=1}^{u}$ of the $(\sum_{r=1}^{s} a_{x_r})$-TDMCSP completed at time

$$t^{u+1}(y^u) \leq t^{s+1}(x^s) \tag{9}$$

such that

$$f_1(y^u) \leq f_1(x^s) \qquad \text{and} \qquad \forall i \in \{2, \ldots, m\} \;\; f_i(y^u) \geq f_i(x^s) \tag{10}$$

with at least one strict inequality in (10). Furthermore, we have that $x^p = \{x^s, x^w\}$ is the concatenation of the sequence x^s and some other sequence $x^w = \{x_{s+r}\}_{r=1}^{w}$ that is a solution sequence of the $(k - \sum_{r=1}^{s} a_{x_r})$-TDMCSP. Thus the concatenation $\{y^u, x^w\}$ of y^u and x^w is also a feasible solution sequence of the k-TDMCSP. We get the following objective values for $x^p = \{x^s, x^w\}$ and for $\{y^u, x^w\}$:

$$f(x^p) = f(\{x^s, x^w\}) = f(x^s) + \sum_{r=1}^{w} c_{x_{s+r}}(t^{s+r}(\{x^s, x^w\}))$$

and

$$f(\{y^u, x^w\}) = f(y^u) + \sum_{r=1}^{w} c_{x_{s+r}}(t^{u+r}(\{y^u, x^w\})),$$

where $t^{s+r}(\{x^s, x^w\})$ and $t^{u+r}(\{y^u, x^w\})$, $1 \leq r \leq w$, are the times of completing subsequences of $s+r-1$ and $u+r-1$ jobs in the sequences $\{x^s, x^w\}$ and $\{y^u, x^w\}$, respectively.

Applying Assumption 1 (a) to (9), we obtain

$$t^{u+1}(y^u) + c_j^1(t^{u+1}(y^u)) \leq t^{s+1}(x^s) + c_j^1(t^{s+1}(x^s))$$

for every element j in any feasible solution sequence. In particular, applying this assumption to every element x_{s+r}, $1 \leq r \leq w$, in the sequence x^w concatenating the sequence y^u and the sequence x^s we get

$$t^{u+1}(y^u) + \sum_{r=1}^{w} c_{x_{s+r}}^1(t^{u+r}(\{y^u, x^w\})) \leq t^{s+1}(x^s) + \sum_{r=1}^{w} c_{x_{s+r}}^1(t^{s+r}(\{x^s, x^w\})),$$

which yields

$$t^{u+w+1}(\{y^u, x^w\}) \leq t^{s+w+1}(\{x^s, x^w\}), \tag{11}$$

where $t^{u+w+1}(\{y^u, x^w\})$ and $t^{s+w+1}(\{x^s, x^w\})$ are the times at which the same subsequence of jobs x^w is completed when started at the times $t^{u+1}(y^u)$ and $t^{s+1}(x^s)$, respectively. Clearly, $t^{s+w+1}(\{x^s, x^w\}) = t^{p+1}(x^p)$.

Applying Assumption 1 (b) to (9) and summing over all the elements in the solution sequence x^w that concatenates the sequence y^u and the sequence x^s, we analogously have

$$\sum_{r=1}^{w} c^i_{x_{s+r}}(t^{u+r}(\{y^u, x^w\})) \geq \sum_{r=1}^{w} c^i_{x_{s+r}}(t^{s+r}(\{x^s, x^w\})), \quad i \in \{2, \ldots, m\}. \tag{12}$$

Observe that (11) is equivalent to

$$f_1(\{y^u, x^w\}) \leq f_1(x^p) \tag{13}$$

while (12) implies that

$$f_i(\{y^u, x^w\}) \geq f_i(x^p), \quad i \in \{2, \ldots, m\}. \tag{14}$$

Since at least one inequality in (10) is strict, it must be that the inequality in (13) and (14) corresponding to the same index i, $i \in \{1, \ldots, m\}$, is also strict. This implies that the solution sequence x^p consisting of subsequences x^s and x^w is not efficient.

□

Note that when Assumption 1 is not satisfied, Theorem 1 is in general not true. Consider, for example, a problem where $c^1_j(t)$, $j = 1, \ldots, n$, are decreasing functions of time, and $c^i_j(t)$, $i = 2, \ldots, m$, $j = 1, \ldots, n$ are all increasing functions of time. Then a solution subsequence x^s that is not efficient and that is completed at a later time than another efficient subsequence y^u may still yield an efficient solution if it is concatenated with an appropriate solution subsequence x^w since the later starting time may yield higher benefits in this case.

Let $f(x) := f(\{x_r\}_{r=1}^p)$ be a nondominated criterion vector of the k-TDMCSP corresponding to an efficient schedule completed at time $t^{p+1}(x)$ that can be computed using (5). Let $G(q(k)) = \text{vmax}^*\{f(x) : x \in q(k)\}$ be the set of all the nondominated criterion vectors of the k-TDMCSP.

By Theorem 1, we establish that

$$\begin{aligned}
G(q(0)) &= \{\underline{0}\} \\
G(q(k)) &= \mathrm{vmax}^*\{f(x^{p+1} = \{x_r\}_{r=1}^{p+1}) : x^{p+1} \in q(k)\} \\
&= \mathrm{vmax}^*\{f(x^p = \{x_r\}_{r=1}^{p}) + c_{x_{p+1}}(t^{p+1}(x^p)) : \\
&\qquad f(x^p) \in G(q(k - a_{x_{p+1}})),\ x_{p+1} \in S,\ k - a_{x_{p+1}} \geq 0\}, \\
&\qquad\qquad k = 1, \ldots, b,
\end{aligned}$$

where the operation *vmax** computes the nondominated criterion vectors according to (7) in the set whose every element is a vector sum of a nondominated criterion vector of the $(k - a_{x_{p+1}})$-TDMCSP completed at time $t^{p+1}(x^p)$, and the cost vector $c_{x_{p+1}}(t)$ evaluated at time $t = t^{p+1}(x^p)$ at which the schedule $x^p = \{x_r\}_{r=1}^{p}$ is completed.

Since all the states are final, the set of vector costs of all the nondominated criterion vectors $\mathcal{Y}_e$ is obtained as the vector-maximum of the union of the sets $G(q(k))$, $k = 0, 1, \ldots, b$; i.e.

$$\mathcal{Y}_e = \mathrm{vmax}^* \bigcup_{k=0,1,\ldots,b} G(q(k)).$$

Note that in each step of the recursion two or more nondominated criterion vectors may correspond to efficient solutions given by different schedules built with the same jobs which shows that different objective function (vector) values can be achieved while choosing the same jobs to a schedule but at different times. We discuss this and similar situations in Section 4.

4 Example

We now present a didactic example of the time-dependent scheduling problem with three criteria ($m = 3$) and three jobs/projects ($n = 3$), i.e. $S = \{1, 2, 3\}$. The fixed budget is given by $b = 3$ and the cost coefficients a_j, $j = 1, 2, 3$ of each job are given by

$$a_1 = 1,\ a_2 = 2,\ a_3 = 1.$$

The first criterion defines the processing time of each job if it is added to a schedule x at time t. With respect to the project selection problem this time can be interpreted, for example, as the duration of the corresponding projects. The other two criteria that are maximized could represent the revenue and dual use potential yielded by the projects if they are selected at time t.

The objective vectors $c_j(t)$, $j = 1, 2, 3$ related to each job are defined as

$$c_1(t)=\begin{bmatrix} 2 \\ 40-t^2 \\ 30-2t \end{bmatrix}, \quad c_2(t)=\begin{bmatrix} 4 \\ 80-2t^2 \\ 40-t \end{bmatrix}, \quad c_3(t)=\begin{bmatrix} 2t+1 \\ 40 \\ 20-2t \end{bmatrix}.$$

The resulting TDMCSP has the following form:

$$\begin{array}{ll} \text{vmax}^* & f(x) = [f_1(x), f_2(x), f_3(x)]^T \\ \text{s.t.} & a(x) \leq 3. \end{array} \tag{15}$$

The possible transitions between states for this example problem are represented by the arcs in the network given in Figure 1. The objective vector $c_j(t) = [c_j^1(t), c_j^2(t), c_j^3(t)]^T$ of each transition and the corresponding job j are identified for each arc and denoted by the vector $[j, c_j]$.

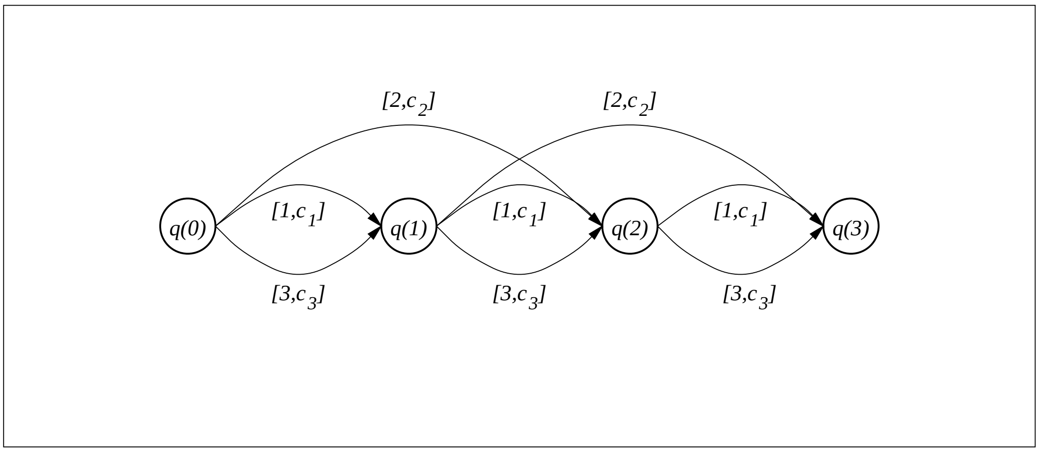

Figure 1: The vertices of this network represent the states of the DP-formulation for the example problem (15).

Applying the recursive equations developed in the previous section we obtain the following sets of nondominated criterion vectors $G(q(k))$, $k = 0, 1, 2, 3$:

$$G(q(0)) = \{\underline{0}\}$$

$$G(q(1)) = \left\{ \begin{bmatrix} 1 \\ 40 \\ 20 \end{bmatrix}, \begin{bmatrix} 2 \\ 40 \\ 30 \end{bmatrix} \right\}$$

$$G(q(2)) = \left\{ \begin{bmatrix} 3 \\ 79 \\ 48 \end{bmatrix}, \begin{bmatrix} 4 \\ 76 \\ 56 \end{bmatrix}, \begin{bmatrix} 4 \\ 80 \\ 40 \end{bmatrix}, \begin{bmatrix} 7 \\ 80 \\ 46 \end{bmatrix} \right\}$$

$$G(q(3)) = \left\{ \begin{bmatrix} 5 \\ 110 \\ 72 \end{bmatrix}, \begin{bmatrix} 5 \\ 118 \\ 59 \end{bmatrix}, \begin{bmatrix} 6 \\ 100 \\ 78 \end{bmatrix}, \begin{bmatrix} 6 \\ 112 \\ 68 \end{bmatrix}, \begin{bmatrix} 10 \\ 119 \\ 62 \end{bmatrix}, \begin{bmatrix} 13 \\ 116 \\ 68 \end{bmatrix}, \begin{bmatrix} 13 \\ 120 \\ 52 \end{bmatrix} \right\}.$$

The set of efficient solutions $\mathcal{X}_e$ and the set of nondominated criterion vectors $\mathcal{Y}_e$ of this example problem can thus be calculated as

$$\begin{aligned} \mathcal{X}_e = & \{ \{\,\}, \{3\}, \{1\}, \{3,1\}, \{1,1\}, \{2\}, \{3,1,1\}, \{3,2\}, \{1,1,1\}, \{1,2\}, \{3,1,3\}, \\ & \{1,1,3\}, \{2,3\} \}, \end{aligned}$$

$$\begin{aligned} \mathcal{Y}_e = & \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 40 \\ 20 \end{bmatrix}, \begin{bmatrix} 2 \\ 40 \\ 30 \end{bmatrix}, \begin{bmatrix} 3 \\ 79 \\ 48 \end{bmatrix}, \begin{bmatrix} 4 \\ 76 \\ 56 \end{bmatrix}, \begin{bmatrix} 4 \\ 80 \\ 40 \end{bmatrix}, \begin{bmatrix} 5 \\ 110 \\ 72 \end{bmatrix}, \begin{bmatrix} 5 \\ 118 \\ 59 \end{bmatrix}, \right. \\ & \left. \begin{bmatrix} 6 \\ 100 \\ 78 \end{bmatrix}, \begin{bmatrix} 6 \\ 112 \\ 68 \end{bmatrix}, \begin{bmatrix} 10 \\ 119 \\ 62 \end{bmatrix}, \begin{bmatrix} 13 \\ 116 \\ 68 \end{bmatrix}, \begin{bmatrix} 13 \\ 120 \\ 52 \end{bmatrix} \right\}. \end{aligned}$$

Note that in this example different criterion vectors are achieved by scheduling the same jobs in a solution sequence at different times. For example, the solution sequences $\{3, 2\}$ and $\{2, 3\}$ with the criterion vectors $[5, 118, 59]^T$ and $[13, 120, 52]^T$ are both efficient. Similarly, the solution sequences $\{3, 1, 1\}$ and $\{1, 1, 3\}$ with the criterion vectors $[5, 110, 72]^T$ and $[13, 116, 68]^T$ are both efficient.

Furthermore we observe that shorter processing times are achieved when only one or two jobs are performed which may not be of high priority to the company.

In order to make a final decision what jobs should be scheduled, the company would have to specify additional preferences. For example, if the preference was to perform three (not necessarily different) jobs rather than only two, four sequences $\{3, 1, 1\}$, $\{1, 1, 1\}$, $\{3, 1, 3\}$ and $\{1, 1, 3\}$ would be the candidates for the final optimal solution. The decision maker would have to choose between the criterion vectors $[5, 110, 72]^T$, $[6, 100, 78]^T$,

$[10, 119, 62]^T$ and $[13, 116, 68]^T$. The vectors show that performing only job 1 yields the smallest revenue but the highest dual use potential and keeps the total processing time quite short. On the other hand, the later job 3 is performed, the later all the jobs are completed while the related revenue and dual use potential stay competitive.

5 Conclusions

We introduced the time-dependent multiple criteria scheduling problem (TDMCSP) and proposed an approach to generate all its efficient solutions. The time-dependency of the model is included in its objective functions while the capacity constraint is assumed to be fixed. The feasible solutions are defined to be schedules of jobs consecutively chosen from a set of jobs at different times. The efficient solutions are found by the proposed dynamic-programming approach using the monotonicity of the objective functions. The model allows to use quite general scheduling criteria personalized to the jobs/projects being scheduled which goes beyond the standard practice in scheduling where criteria such as flowtime and tardiness are typically used.

As in every multiple criteria program, the solutions inform the decision maker about the structure of the efficient and nondominated set of the TDMCSP. The time-dependency makes the solutions even more significant as it equips the decision maker with additional information on mutual relationships among the jobs of the efficient schedule, their order in the schedule with respect to time, and the related objective function values. This information provides the decision maker with deeper insight into the model and may serve as a decision tool while choosing a preferred optimal solution.

The model and the solution approach are part of AMADEuS, an interactive decision tool developed by Klamroth et al. [13]. AMADEuS is based on MATLAB 5.3.0 [22], a software package for numeric computation, data analysis and graphics. The tool generates nondominated criterion vectors for several types of multiple criteria capital budgeting problems and for the TDMCSP, all being variations of the MCKP. The model presented in this paper allows a job to be repeated in a schedule while AMADEuS includes two scenarios for the TDMCSP: schedules with and without jobs' repetitions. The tool is being currently tested so that complete computational results on a family of MCKP-related models will be available in the near future.

The proposed model could be also modified to accommodate additional features. For example, applying lexicographic vector optimization and always minimizing first with respect to the first component of the criterion vector, the shortest-processing-time scheduling rule could be included.

Further research should focus on developing more complex time-dependent models featuring multiple constraints or multiple time periods.

References

[1] B. Alidaee and N. K. Womer. Scheduling with time dependent processing times: Review and extensions. *Journal of the Operational Research Society*, 50:711–720, 1999.

[2] M. Azizoglu, S. K. Kondakci, and M. Koksalan. Bicriteria scheduling: minimizing flowtime and maximum earliness on a single machine. In J. Climaco, editor, *Multicriteria Analysis*, pages 279–288. Springer, Berlin, 1997.

[3] K. Bhaskar. A multiple objective approach to capital budgeting. *Accounting and Business Research*, 9:25–46, 1979.

[4] T. C. E. Cheng and Q. Ding. The complexity of scheduling starting time dependent tasks with release times. *Inform. Process. Lett.*, 65:75–79, 1998.

[5] T. C. E. Cheng and Q. Ding. The time dependent machine makespan problem is strongly NP-complete. *Computers and Operations Research*, 26:749–754, 1999.

[6] R. S. Garfinkel and G. L. Nemhauser. *Integer Pogramming*. John Wiley & Sons, New York, 1972.

[7] A. Gascon and R. C. Leachman. A dynamic programming solution to the dynamic, multi-item, single-machine scheduling problem. *Operations Research*, 36:50–56, 1988.

[8] S. Gawiejnowicz. A brief survey of continuous models of scheduling. *Found. Comput. and Decis. Sci.*, 21:81–100, 1996.

[9] J. N. D. Gupta, J. C. Ho, and J. A. A. van der Veen. Single machine hierarchical scheduling with customer orders and multiple job classes. *Annals of Operations Research*, 70:127–143, 1997.

[10] J. A. Hoogeveen. Minimizing maximum promptness and maximum lateness on a single machine. *Mathematics of Operations Research*, 1:100–114, 1996.

[11] J. A. Hoogeveen and S. L. van de Velde. Minimizing total completion time and maximum cost simultaneously is solvable in polynomial time. *Operations Research Letters*, 5:205–208, 1995.

[12] T. Ibaraki. Enumerative approaches to combinatorial optimization, part ii. *Annals of Operations Research*, 11:343–602, 1987.

[13] K. Klamroth, M. M. Wiecek, and R. L. Hartman. AMADEuS, Affordable MultiAttribute DEcisionS, a decision support tool for multiple criteria capital budgeting. Technical report, Dept. of Math. Sc., Clemson University, Clemson, SC, 1999.

[14] A. J. Kleywegt and J. D. Papastavrou. The dynamic and stochastic knapsack problem. *Operations Research*, 46:17–35, 1998.

[15] S. K. Kondakci, E. Emre, and M. Koksalan. Scheduling of unit processing time jobs on a single machine. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making*, pages 654–660. Springer, Berlin, 1997.

[16] A. Konov. Scheduling problems with linear increasing processing times. In U. Zimmermann, U. Derigs, W. Gaul, R. H. Moehring., and K.-P. Schuster, editors, *Operations Research Proceedings 1996*, pages 208–212. Springer, Berlin, 1997.

[17] M. M. Kostreva, W. Ogryczak, and D. W. Tonkyn. Relocation problems arising in conservation biology. *Computers and Mathematics with Applications*, 37:135–150, 1999.

[18] M. M. Kostreva and M. M. Wiecek. Time dependency in multiple objective dynamic programming. *Journal of Mathematical Analysis and Applications*, 173:289–307, 1993.

[19] W. Kwak, Y. Shi, H. Lee, and C. F. Lee. Capital budgeting with multiple criteria and multiple decision makers. *Review of Quantitative Finance and Accounting*, 7:97–112, 1996.

[20] S. Martello, D. Pisinger, and P. Toth. New trends in exact algorithms for the 0-1 knapsack problem. In J. Barceló, editor, *Proceedings of EURO/INFORMS-97, Barcelona*, pages 151–160, 1997.

[21] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations.* John Wiley & Sons, New York, 1990.

[22] *MATLAB - The Language of Technical Computing.* The Math Works, Inc., 3rd edition, 1999.

[23] S. T. McCormick. Fast algorithms for parametric scheduling come from extensions to parametric maximum flow. *Operations Research*, 47:744–756, 1999.

[24] J. P. Sousa and L. Wolsey. A time-indexed formulation of nonpreemptive single machine scheduling problems. *Mathematical Programming*, 54:353–367, 1992.

[25] J.-Y. Teng and G.-H. Tzeng. A multiobjective programming approach for selecting non-independent transportation investment alternatives. *Transportation Research - B*, 30:291–307, 1996.

[26] J. M. van den Akker. *LP-based solution methods for single machine scheduling problems.* Ph.D. Dissertation,Technische Universiteit Eindhoven, Eindhoven, 1994.

[27] J. M. van den Akker, C. P. M. van Hoesel, and M. W. P. Savelsbergh. A polyhedral approach to single-machine scheduling problems. *Mathematical Programming*, 85:541–572, 1999.

[28] R. Vetschera. Time preferences in capital budgeting - an application of interactive multiobjective optimization. *Methods of Operations Research*, 50:649–660, 1985.

[29] B. Villarreal and M. H. Karwan. Multicriteria integer programming: A (hybrid) dynamic programming recursive approach. *Mathematical Programming*, 21:204–223, 1981.