ELSEVIER

Fuzzy Sets and Systems 121 (2001) 523–535

FUZZY sets and systems

www.elsevier.com/locate/fss

# Fuzzy scheduling with application to real-time systems

Marin Litoiu[a], Roberto Tadei[b,*]

[a]IBM Toronto Lab, 1150 Eglinton East, Toronto, Ontario, Canada
[b]Politecnico di Torino, Dip. di Automatica e Informatica, Corso Duca degli Abruzzi 24, 10129 Torino, Italy

Received 12 January 1999; received in revised form 10 August 1999; accepted 20 September 1999

**Abstract**

Task scheduling is a main activity in the design of real-time systems (RTS). It assures both functionality and safety of such systems. RTS can be modeled as a set of periodic tasks that must be completed before specific deadlines. In this paper, we investigate the fuzzy scheduling models on RTS and the main methodologies that solve these models. Thus, we present general periodic task scheduling models with fuzzy deadlines and fuzzy processing times; scheduling algorithms based on optimal assignment of the priorities; and a more general framework for designing RTS, Rate Monotonic Scheduling Theory, that includes the scheduling algorithms. A case study will illustrate the use of the theory. © 2001 Elsevier Science B.V. All rights reserved.

*Keywords:* Periodic scheduling; Fuzzy deadlines; Fuzzy processing times; Heuristics

## 1. An overview of fuzzy scheduling

### 1.1. The general scheduling model

Two different research communities, operational research and computer science, have examined scheduling problems from their own perspectives. The former focuses on off-line techniques, the latter on dynamic scheduling. Also the problems are different from a resources, time granularity and metrics point of view. In spite of these differences, the abstract problems have much in common.

The general scheduling model consists of a system of tasks ($\Gamma$), which should be sequenced on a set of resources ($\boldsymbol{R}$) under a set of constraints and a set of performance cost functions.

*Resources*: In the most cases, resources are machines or processors:

$$\boldsymbol{R} = \{R_1, R_2, \ldots, R_m\}.$$

They may be identical or have different processing speeds. In the most general case, there are additional resources as primary or secondary memory, input/output devices, etc.

*Tasks*: A system of tasks can be described by

$$\Gamma = (\boldsymbol{T}, \leqslant, [p_{ij}], \{R_j\}, \{I_j\}, \{d_j\}, \{t_{0j}\})$$

such that:

- $\boldsymbol{T} = \{T_1, T_2, \ldots, T_n\}$ is the set of tasks, where a task is an executable section of a program or a production activity;

* Corresponding author. Tel.: +39-11-5647032; fax: +39-11-5647099.

*E-mail addresses:* marin@ca.ibm.com (M. Litoiu), tadei@polito.it (R. Tadei).


PII: S0165-0114(99)00176-1

- $\leqslant$ is a partial order relation defined on $\boldsymbol{T}$, which specifies the sequential constraints. $T_i \leqslant T_j$ means $T_i$ must be executed before $T_j$;
- $[p_{ij}]$ is the matrix of processing times. $p_{ij}$ is the time required by task $T_j$ on the resource $R_i$. When the processors are identical, $p_j$ is the processing time of task $T_j$;
- $R_j = [R_1(T_j), \ldots, R_m(T_j)]$, $1 \leqslant j \leqslant n$, denotes the quantity of resources of type $R_i$ required by task $T_j$;
- $I_j$ is the period of task $T_j$. *Periodic tasks* are those invoked regularly; they have a finite fixed period. For *non-periodic tasks*, the periods $I_j = \infty$. Tasks that are invoked more than once but at irregular moments are called *aperiodic*;
- $d_j$ is the *due date* associated with task $T_j$. The task *should* be finished after $d_j$ time units since its invocation. If the task *must* be completed before or at $d_j$, the due date is a *deadline*. For tasks with no due date, $d_j = \infty$;
- $t_{0j}$ is the moment of the first invocation for task $T_j$.

*Scheduling constraints*: Using constraints, we define general scheduling conditions such as:

- Non-preemptive scheduling: a task cannot be interrupted once it begins execution.
- Preemptive scheduling: a task execution can be interrupted.
- Priority lists.

*Performance cost functions*: The performance criteria are cost functions defined with respect to due dates and *completion times* of the tasks (completion time is the moment when the execution of a task is completed).

### 1.2. Fuzzy scheduling

So far, fuzzy sets have been involved in scheduling under two aspects:

- *Modeling*. Although scheduling implies determinism, the imprecision was introduced in the general scheduling model. It deals with fuzzy constraints, fuzzy due dates and fuzzy processing time.
- *Solving deterministic models*. Approximate reasoning based on fuzzy logic was used to build simple solvers [4,8] or expert systems [16,17].

#### 1.2.1. Fuzzy due dates

Deterministic models with fuzzy due dates have been proposed in several papers. Thus, Litoiu and Tadei [9] have proposed a periodic model with linear fuzzy due dates. The same linear fuzzy due dates with a non-periodic model have been used by Ishii et al. [6]. Usually, in these approaches, the due dates have a finite support and a satisfaction function is associated to the due date of each task, as shown in Fig. 1. The satisfaction of a task is 0 when the completion time of a task is greater than $b$ and is 1 for values of completion time less than $a$. Values of completion time within the interval $[a\ \ b]$ give satisfaction in $(1\ \ 0)$ interval. The scheduling should maximize the satisfaction over all tasks and under supplementary constraints [6] or over all task invocations as in the periodic model [9,10]. The problem of finding the optimal scheduler is transformed in a set of deterministic models by mapping satisfaction values in deterministic due dates, as shown by the arrows in Fig. 1.

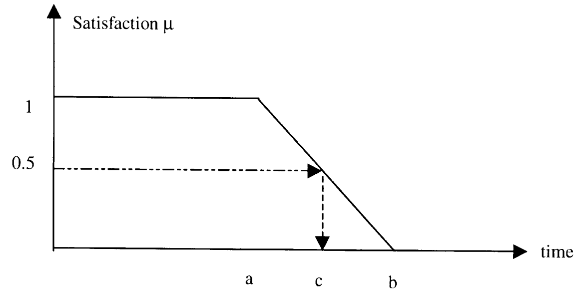


Fig. 1. Linear fuzzy due dates.

#### 1.2.2. Fuzzy constraints

Slany [14] has extended the term constraints over performance cost functions and resources. Expressed as soft barriers, the constraints allow the application of fuzzy concepts. Each constraint becomes a criterion in multi-criteria decision making framework. Deterministic scheduling models built from the support of fuzzy constraints are then compared to these fuzzy criteria. The overall performance function is computed by applying aggregation techniques and ranking methods on the criteria.

#### 1.2.3. Fuzzy processing times

As stated in [2], fuzziness can represent preference or uncertainty. As far as the processing times in real-time systems are concerned, we must recognize that they are uncertain variables. So, it becomes very natural for this kind of systems to consider fuzzy