European Journal of Operational Research 000 (2017) 1–14

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Match-up scheduling of mixed-criticality jobs: Maximizing the probability of jobs execution

Yasmina Seddik[a,*], Zdenek Hanzálek[a,b]

[a] FEE, Czech Technical University in Prague, Karlovo namesti 13, Prague 12135, Czech Republic
[b] CIIRC, Czech Technical University in Prague, Karlovo namesti 13, Prague 12135, Czech Republic

ARTICLE INFO

Article history:
Received 28 April 2015
Accepted 20 March 2017
Available online xxx

Keywords:
Scheduling
Mixed-criticality
Match-up
Release dates
Deadlines

ABSTRACT

This paper deals with a mixed-criticality scheduling problem: each job has a criticality level depending on its importance. In addition, each job has a finite set of possible processing times, and a known probability for each of them. Every job must be processed between its release date and its deadline. Moreover, each job has a weight corresponding to its payoff. This problem has applications in single machine scheduling of real time embedded systems scheduling, production and operating theaters.

We propose a model that takes all the possible processing times of a job into account. An offline multilevel schedule is computed such that safety rules are satisfied, in every situation. This is achieved by allowing the rejection of low criticality jobs when higher criticality jobs need longer processing time, at runtime. The runtime schedule is matched-up again with the offline schedule after such deviations from the offline schedule. The offline multilevel schedule optimizes a non-regular criterion aiming to maximize the average weighted probability of jobs execution (i.e., the total expected payoff).

Such a problem is strongly NP-hard. We first study the problem where the sequence of jobs is fixed: we show its complexity and provide a MILP formulation. For the case with two levels of criticality, we provide a dynamic programming algorithm. Finally, we propose a Branch and Bound method for the general problem (i.e., without a fixed job sequence).

1. Introduction

In classical scheduling, processing times are often assumed to be deterministic and known in advance, which is not always true in reality. It is well known that determining the exact processing times of jobs is a very difficult problem, namely due to the events occurring at runtime. Considering best case (i.e., shortest possible) processing times can lead to a schedule that is dense at runtime and that makes efficient usage of the resource; however deadline constraints may be violated when the actual processing time is longer. On the other hand, considering worst case processing times allows the computation of a schedule that meets safety constraints, but it leads to a schedule that is sparse at runtime, i.e., to a waste of resources.

In mixed-criticality scheduling, first introduced by Vestal (Vestal, 2007), jobs with different criticality levels are distinguished. Considering different criticality levels ensures the satisfaction of safety constraints of high criticality jobs with very high probability, as well as an efficient resource usage (via a schedule that avoids useless idle times and maximizes the (weighted) probability of executed jobs within their time window (defined by a release date and a deadline)). The underlying idea is simple: high criticality jobs are not scheduled side by side, instead they are interleaved with low criticality ones, that can be rejected when the high criticality job processing time is longer at runtime execution. Therefore, a single multilevel schedule is computed offline, which includes several alternatives of the execution that are decided at runtime.

1.1. Motivation example 1 – in-vehicle communications

Our model is motivated by safety-critical applications, such as autonomous cars, using so called time-triggered communication networks, where the nodes have synchronized clocks and messages are transmitted at moments defined by the offline schedule (see the static segment of FlexRay protocol (Dvorak & Hanzalek, 2016) used in the automotive industry or the Isochronous Real Time of a Profinet protocol (Hanzalek, Burget, & Sucha, 2010) used in industrial automation, for example). The schedule is repeated

* Corresponding author.
E-mail addresses: yasminaseddik@gmail.com, yasmina.seddik@fel.cvut.cz (Y. Seddik), hanzalek@fel.cvut.cz (Z. Hanzálek).

http://dx.doi.org/10.1016/j.ejor.2017.03.054

periodically, since the functionalities of the car (steering, trajectory planning, engine control loops, computer vision, chassis stabilization, navigation, entertainment, etc.) are periodic. For simplicity, we consider that all jobs have the same period, therefore we can omit the multi-periodic nature of the problem and we can concentrate on one period only (as in the Profinet case). Sensing, computation and actuation performed by the nodes are executed at specific moments within the period, which represent the release date (i.e., availability of the data on the transmitter side) and the deadline (i.e., latest moment when the data is needed on the receiver side) for every message transmitted on the network. Time-triggered communication is characterized by complete determinism, and is, hence, particularly easy to verify and have certified. However, the traditional paradigm offers limited flexibility: once the schedule is computed (prior to runtime), it is not possible to modify it in response to events that may have occurred during runtime execution.

Instead, communications reliability may be increased by message retransmission if the original message was corrupted. The need for retransmission is rare, but it leads to a prolongation of the communication jobs at runtime. Jobs are nonpreemptive, since the particular structure of the messages does not allow resuming their sending after preemption. For this application, the criticality level of a job corresponds to its maximum number of possible (re)transmissions. For example, the Automotive Safety Integrity Level (ASIL) given by ISO 26262 defines four criticality levels, and the DO178-B avionics standard, used by the Federal Aviation Administration, defines five criticality levels. Let us consider the following jobs sharing one resource (i.e., the communication channel) and having three different levels of criticality:

- Jobs with high criticality (three transmissions: criticality level 3) are used for safety-related functionalities, such as steering and braking
- Jobs with medium criticality (two transmissions: criticality level 2) are used for mission-related functionalities (their failure or malfunction may prevent a goal-directed activity from being successfully completed, for example an autonomous car will not reach a desired destination), such as combustion engine control or navigation system;
- Jobs with low criticality (one transmission: criticality level 1) are used for infotainment functionalities, such as a CD player.

In this example the criticality levels are consecutive integers, a situation that does not necessarily occur. For instance, high criticality jobs could be allowed four transmissions: their criticality level would be 4 and there would be zero jobs with criticality level 3.

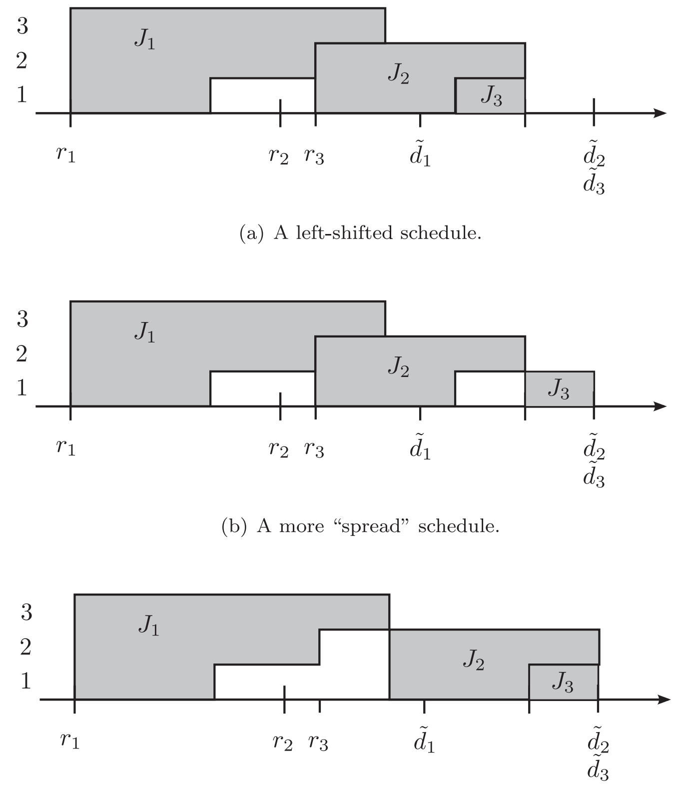
A solution of the scheduling problem is given by a three-level schedule:

- Level 1 considers the best-case processing times (i.e., a single transmission of a message) of all jobs,
- Level 2 omits low criticality jobs and it considers two transmissions of medium criticality and high criticality messages,
- Level 3 includes high criticality jobs only, each of them represents a message transmitted three times.

Each job is constrained by its release date and deadline. Three different feasible mixed-criticality schedules with three jobs on one resource are shown in Fig. 1. When no retransmission occurs at runtime, the schedule is executed on level 1. The objective is to maximize the weighted probability of jobs execution (detailed in Section 2.2).

1.2. Motivation example 2 – production with subcontracting

Let us consider a production system where a single centralized machine controls all the processes of a workshop. An example of a high criticality job would be related to an important customer whose orders are crucial for the workshop's future, they are subject to the customer's audit and they cannot be subcontracted to an external company. A medium criticality job should be performed in-house, but it can be subcontracted. Low criticality jobs can be subcontracted without any impact on the workshop's reputation. A schedule needs to be robust with respect to the prolongation of processing times, by subcontracting lower criticality jobs when higher criticality jobs need more resource time. The objective is to minimize the subcontracting expenses.



(a) A left-shifted schedule.

(b) A more "spread" schedule.

(c) Another "spread" schedule.

Fig. 1. Three different feasible mixed-criticality schedules of high criticality job J_1, medium criticality job J_2 and low criticality job J_3. Release dates are denoted by r and deadlines by $\tilde{d}$. The vertical axis represents the three levels of the schedule.

1.3. Motivation example 3 – operating theater

Let us consider a single operating theater which is dedicated to the provision of surgical operations under the uncertainty of their duration (Denton, Miller, Balasubramanian, & R.Huschka, 2010). A cardiovascular operation represents a high criticality job whose duration is not fully deterministic. Nevertheless, the cardiovascular operation needs to be completed even though it implies the rejection of some medium criticality job (such as a hip replacement surgery) or low criticality job (such as a plastic surgery operation). All jobs are constrained by release dates, representing the availability of medical checkups, and deadlines, representing their expiration time. A rejected job can be considered in some future schedules, but in such a case it requires a new medical checkup. The objective is to maximize the revenue of the operating theater.

1.4. General description

In general, the purpose of the mixed-criticality scheduling framework is to manage interactions between higher and lower

Please cite this article as: Y. Seddik, Z. Hanzálek, Match-up scheduling of mixed-criticality jobs: Maximizing the probability of jobs execution, European Journal of Operational Research (2017), http://dx.doi.org/10.1016/j.ejor.2017.03.054