# Scheduling periodic tasks in a hard real-time environment

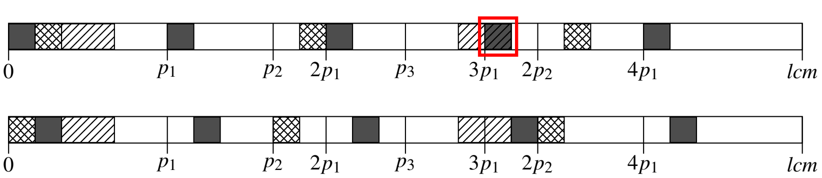
Friedrich Eisenbrand[1], Nicolai Hähnle[1], Martin Niemeier[1],
Martin Skutella[2], José Verschae[2], and Andreas Wiese[2]

[1] EPFL, Lausanne, Switzerland
[2] TU Berlin, Germany

**Abstract** We consider a real-time scheduling problem that occurs in the design of software-based aircraft control. The goal is to distribute tasks $\tau_i = (c_i, p_i)$ on a minimum number of identical machines and to compute offsets $a_i$ for the tasks such that no collision occurs. A task $\tau_i$ releases a job of running time $c_i$ at each time $a_i + k \cdot p_i$, $k \in \mathbb{N}_0$ and a collision occurs if two jobs are simultaneously active on the same machine.
We shed some light on the complexity and approximability landscape of this problem. Our main results are as follows: (i) We show that the minimization problem cannot be approximated within a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$. (ii) If the periods are harmonic (for each $i, j$ one has $p_i \mid p_j$ or $p_j \mid p_i$), then there exists a 2-approximation for the minimization problem and this result is tight, even asymptotically. (iii) We provide asymptotic approximation schemes in the harmonic case if the number of different periods is constant.

## 1 Introduction

The motivation for this research comes from a real-word combinatorial optimization problem that was communicated to us by our industrial partner, a major avionics company. The aircraft designers need to schedule highly critical periodic control tasks with a predictable and static scheduling policy such that preemption and dynamic effects are avoided. The model that is used in this context is as follows. One is given *tasks* $\tau_1, \ldots, \tau_n$ where each task $\tau_i = (c_i, p_i)$ is characterized by its *execution time* $c_i \in \mathbb{N}$ and *period* $p_i \in \mathbb{N}$. The goal is to assign the tasks to identical machines and to compute offsets $a_i \in \mathbb{N}_0$ such that no collision occurs. A task $\tau_i$ generates one *job* with execution time $c_i$ at every time unit $a_i + p_i \cdot k$ for all $k \in \mathbb{N}_0$. Each job needs to be processed immediately and non-preemptively after its generation on the task's machine. A collision occurs if two jobs are simultaneously active on one machine.




The picture above shows three tasks $\tau_1 = (1,6), \tau_2 = (1,10)$ and $\tau_3 = (2,15)$. The upper part shows an infeasible assignment of offsets $(a_1 = 0,\ a_2 = 1,\ a_3 = 2)$ whereas

Dagstuhl Seminar Proceedings 10071
Scheduling
http://drops.dagstuhl.de/opus/volltexte/2010/2534

the lower part shows a feasible assignment of offsets ($a_1 = 1,\ a_2 = 0,\ a_3 = 2$). Notice that the schedule repeats after the least-common multiple (*lcm*) of the periods.

In the single machine context and with unit execution times, this problem was studied by Wei and Liu [WL83] who called the problem of computing offsets the *periodic maintenance problem*. Baruah et al. [BRTV90] and independently [BNBNS02,Bha98] show that the periodic maintenance problem is NP-hard in the strong sense.

Here we are interested in the corresponding *machine minimization* problem, i.e., we want to find the minimum number of identical machines on which the tasks can be distributed in a feasible way. We refer to this problem as the *periodic maintenance minimization problem*. Korst et al. [KALW91,KAL96] studied this problem and show independently from [BRTV90] and [BNBNS02,Bha98] that it is NP-hard in the strong sense. Sometimes even moderately sized real-word instances turn out to be unsolvable with state-of-the-art integer programming approaches. One feature that the easier instances share is that, with only few exceptions, their tasks have harmonic periods, i.e., for each pair of tasks $\tau_i, \tau_j$ one has $p_i \mid p_j$ or $p_j \mid p_i$. Thus, one question that arises is whether instances with this divisibility property are easier to solve than general instances. It turns out that the difference is drastic.

*Our contribution* is a rigorous account on the complexity and approximability landscape of the above described machine-minimization problem. Our results are summarized in Table 1. The details are presented in [EHN$^+$10]. The main results are as follows:
▶ We prove that, for any $\varepsilon > 0$, it is NP-hard to approximate the periodic maintenance minimization problem within a factor of $n^{1-\varepsilon}$, i.e., that the trivial approximation algorithm is essentially tight. This explains the difficulty of moderately sized instances without harmonic periods from a theoretical viewpoint. The result is achieved by a reduction from COLORING that relies on basic number-theoretic results like the Chinese Remainder Theorem and the Prime Number Theorem. We remark that the reduction has been given independently in [BNBNS02,Bha98]. The hardness result also holds under resource augmentation.

▶ We show that the periodic maintenance minimization problem with harmonic periods allows for a 2-approximation algorithm. Furthermore, we show that this is tight, even asymptotically. It is remarkable that a simple variant of First-Fit can be analyzed to be a 2-approximation algorithm. The analysis differs however considerably from the simple analysis that shows that First-Fit for BIN-PACKING yields a 2-approximation. The novel concept that we use is the one of a witness for certain groups of machines. These witnesses prove that these groups are heavily loaded. If a witness for a certain group of machines is missing, the instance can be separated into two independent sub-instances. This allows for an analysis by induction.

▶ Even though the 2-approximation result for the case of harmonic periods is tight, we show that a stronger restriction leads to an asymptotic PTAS: If the number of different periods $k$ is constant, we have an efficient algorithm with approximation guarantee $(1+\varepsilon)OPT + k$, for any constant $\varepsilon > 0$. The basic approach follows the ideas of the classical APTAS for BIN-PACKING of Fernandez de la Vega and Luecker [FdlVL81]. The more complicated nature of the periodic maintenance problem, however, requires several interesting and nontrivial extensions of the techniques such as a more sophisticated rounding procedure and advanced structural insights into the solutions to the

| arbitrary periods | | |
|---|---|---|
| period lengths $k$ | algorithms | hardness results |
| $k$ arbitrary | $2OPT + k - 1$ | $n^{1-\varepsilon} OPT$ |
| $k$ constant | $\left(\frac{3}{2} + \varepsilon\right) OPT + k$ | $\left(\frac{3}{2} - \varepsilon\right) OPT + k - 1$ |

| harmonic periods | | |
|---|---|---|
| period lengths $k$ | algorithms | hardness results |
| $k$ arbitrary | $2OPT$ | $(2 - \varepsilon) OPT + o(OPT)$ |
| $k$ constant | $(1 + \varepsilon) OPT + k$ | $\left(\frac{3}{2} - \varepsilon\right) OPT + k - 1$ |
| $q_k/q_1$ constant | $(1 + \varepsilon) OPT + 1$ | $(2 - \varepsilon) OPT$ |

**Table 1.** The approximability landscape of the periodic maintenance minimization problem. Here $q_k$ and $q_1$ denote the largest and smallest period length, respectively.

periodic maintenance problem. This helps to enumerate the solution space to find a template that can be turned into a solution with the desired approximation guarantee.

## References


[Bha98] R. Bhatia. *Approximation Algorithms for Scheduling Problems*. PhD thesis, University of Maryland, 1998.

[BNBNS02] Amotz Bar-Noy, Randeep Bhatia, Joseph Naor, and Baruch Schieber. Minimizing service and operation costs of periodic scheduling. *Math. Oper. Res.*, 27(3), 2002.

[BRTV90] S. Baruah, L. Rousier, I. Tulchinsky, and D. Varvel. The complexity of periodic maintenance. *Proceedings of the International Computer Symposium*, 1990.

[EHN+10] F. Eisenbrand, N. Hähnle, M. Niemeier, M. Skutella, J. Verschae, and A. Wiese. Scheduling periodic tasks in a hard real-time environment. In *Proceedings of ICALP 2010*, Lecture Notes in Computer Science. Springer, 2010. To appear. Full paper at http://disopt.epfl.ch/webdav/site/disopt/shared/PM_EHNSVW10_report.pdf.

[FdlVL81] W. Fernandez de la Vega and G. S. Lueker. Bin packing can be solved within $1 + \varepsilon$ in linear time. *Combinatorica*, 1:349–355, 1981.

[KAL96] J. Korst, E. Aarts, and J. K. Lenstra. Scheduling periodic tasks. *INFORMS Journal on Computing*, 8:428–435, 1996.

[KALW91] J. Korst, E. Aarts, J. K. Lenstra, and J. Wessels. Periodic multiprocessor scheduling. In E. H. L. Aaarts, J. van Leeuwen, and M. Rem, editors, *PARLE '91 Parallel Architectures and Languages Europe*, volume 505 of *Lecture Notes in Computer Science*, pages 166–178. 1991.

[WL83] W. D. Wei and C. L. Liu. On a periodic maintenance problem. *Operations Research Letters*, 2:90–93, 1983.