Bachelor Thesis Review

Faculty of Mathematics and Physics, Charles University

Thesis author	Vít Koštejn
Thesis title	Algorithms for Timetabling in Sports
Year submitted	2022
Study program	Computer Science
Specialization	Obecná informatika

Review author	Mgr. Tung Anh Vu	Reviewer
Department	Department of Applied Mathematics	

Summary

This work introduces the Periodic Scheduling with Shared Resources (PSR) problem, which is motivated by the problem of creating startlists in orienteering events. All numbers in the following description of the problem are integers. A *periodic job* is given by a triple of *number of repetitions* p_j, a *required resource* f_j, and a *minimal period* g_j. The goal is to schedule n periodic jobs on m machines with the following constraints. A *schedule* is given by a *start time* S_j for $1 \leq j \leq n$ and a *period* $G_j \geq g_j$. We require that each repetition of a job occurs at times $\{S_j + i \cdot G_j \mid i \in \{0, \ldots, p_j - 1\}\}$. No two jobs using the same resource can be scheduled[1] in the same time step at any point of the schedule. The goal is to find a schedule of minimum *makespan*, i.e., the time of the last completed job.

Theorem 1.4.2 shows that PSR is NP-hard. In light of this result, the author studies approximation algorithms for the problem. They do not manage to design an algorithm for PSR in its generality, but they are successful in developing algorithms for several special cases.

- Theorem 1.5.3 shows a 3-approximation algorithm for the case where g_j's (minimal periods of the jobs) are uniform.
- Lemma 1.6.1 improves the approximation ratio to 2 in the case when g_j's are all equal to one.
- Moreover, Theorem 1.6.3 improves upon Lemma 1.6.1 and shows that the greedy algorithm from the said lemma has an approximation ratio of $\frac{22+\sqrt{3}}{13}$.
- Theorem 1.7.3 shows a 2-approximation algorithm for the case when g_j's are powers of two for the case when there is a single machine. Surprisingly, this result is relevant since the author claims (but defers the proof to the journal publication) that even this case is NP-hard.

Furthermore, the author also provides some lower bounds for the analyzed algorithms.

- Theorem 1.5.1 shows that any algorithm for PSR with a uniform minimal period g_j which sets all periods G_j to g_j has an approximation ratio of at least 2. This shows that analysis of the algorithm from Theorem 1.5.3 can be improved to be a 2-approximation at best.

[1] To different machines, of course.

- Theorem 1.6.2 shows that the greedy algorithm from Lemma 1.6.1 has an approximation ratio of at least $\frac{11}{8}$.

The author also implements the developed algorithms. They experimentally evaluated the implementations on historical data. In addition, the author gives a CSP formulation of the problem and evaluates the performance of a CSP solver. The performance of these algorithms is compared to human-made schedules. As far as I can tell, the test bench and the implementation is correct and well-designed.

Relevance

From my cursory survey of the literature, I agree with the author that PSR has not been studied before.

Periodicity constraints have been studied in the context of operating system schedulers. In this setting however, jobs repeat infinitely many times, as opposed to PSR where the number of repetitions is finite (and given on input).

Resource constraints have also been studied before. The author seems to be aware of the existing literature. The following work may be of interest to the author, since the objective is makespan minimization as well.

> Vitaly A. Strusevich (2021) Approximation algorithms for makespan minimization on identical parallel machines under resource constraints, Journal of the Operational Research Society, 72:9, 2135–2146, DOI: 10.1080/01605682.2020.1772019.

Techniques

The algorithm given by Theorem 1.5.3 is obtained by analyzing a greedy algorithm. This algorithm is not too dissimilar from the classical greedy 2-approximation algorithm for $P||C_{\max}$. An additional case compared to the classical algorithm increases the approximation ratio from 2 to 3.

The algorithm given by Lemma 1.6.1 is not purely a greedy algorithm. While there are unscheduled jobs, it chooses the unscheduled job with the most frequent resource from the pool of unscheduled jobs and schedules it greedily. In other words, the jobs are not processed one by one but are ordered by a priority. A clever observation about the job which determines the makespan lets the author show that the approximation ratio in this case is only 2.

Theorem 1.6.3 further improves the analysis of the greedy algorithm from Lemma 1.6.1 and shows that the approximation ratio is only $\frac{22+\sqrt{3}}{13} \approx 1.83$. The proof of the theorem requires deep insight into the structure of the solution of the greedy algorithm. To me, this is the technical highlight of this work.

Unfortunately, I was not able to verify this proof in full detail. The last paragraph on page 17 and the following one are incomprehensible to me. This is caused by using confusing notation (and an uncharacteristic lack of figures). For an example, I do not follow why "no one of [jobs running at time $t-1$ on machines which do not start by a gray job] are green". Next, it is claimed that "the jobs" would overlap. Which ones? Why is an overlap an issue? (I guess that jobs with the same resource would clash, but I do not see why one job of green shade cannot run) Issues of similar nature occur in the following paragraph.

Theorem 1.6.1 provides a lower bound for the algorithm given by Theorem 1.6.3 and finds an instance where the solution has cost at least $\frac{11}{8}$ times the optimum.

Theorem 1.7.3 is obtained by analyzing a *doubling algorithm*. As far as I can tell, these results mostly follow standard techniques with a mix of insights into the structure of the solution.

Aside from the aforementioned issues, most arguments in the proofs are "morally" correct from my point of view. Their presentation is occasionally lacking, however, and hinders their comprehensibility. As such, I was not able to verify all arguments in detail. I elaborate more on this in the following section.

Even though the used techniques are not completely novel, their contribution to the state of the art should be very much appreciated.

Presentation

The author should be commended for their English. The number of typos is minimal. The number of grammatical errors is within a reasonable threshold. However some "Czech-isms" do appear (see Minor comments).

The typography and presentation is sufficient. There are many places to improve, especially for the journal publication (see Minor comments), but none of the issues prevent the reader from understanding the main message of the text.

Some proofs and algorithm descriptions are overly reliant on figures and pseudocode respectively. My opinion is that textual description should always be provided. For an example, Figure 1.3a is used to prove Theorem 1.5.1 contains a dark gray job with both a period of size 1 and 2. This is not a feasible solution as the period must be a single number. It is possible that these are two different shades of dark gray, but it is not visually distinguishable for me at least.

In the proof of Theorem 1.6.2, colors of jobs had their meaning changed. In Section 1.3, the *color* of a job refers to its consumed resource. Here however, the colors are merely used to refer to groups of jobs. Also at a first glance, it would seem that each job in the figure is actually a single action of a job.[2] A better description of the figure (preferably a textual one) would clear the confusion.

Additional comments

In the introduction of Chapter 2, you claim that the running time of the implementations is limited to 10 minutes due to "the impatience of the Internet user". I never encountered an user who would be willing to wait more than seconds. The standard practice for long running processes is for the server to acknowledge that a process has started, and notify the user (independently of the request session) that the process is finished. Try requesting the copy of your personal data collected by Google for an example; you should receive an email notification roughly the next day that the process is finished.

Overall assessment

The submitted thesis clearly demonstrates that the student has a very solid understanding of the state of the art. They were able to leverage this knowledge to obtain original results. This is far beyond what can be expected from a good Bachelor's thesis. As such, I recommend this thesis to be accepted as a award level Bachelor's thesis and awarded the highest possible grade. I have no doubt that the journal publication of this thesis will be accepted.

[2]That is how jobs are depicted in classical scheduling problems.

Questions for the defence[3]

1. In the proof of Theorem 1.6.2, you claim that "all machines are full till minute $\frac{7}{2}m$ and only jobs of m resources remain to be scheduled." Is this claim true no matter the chosen order of the tied resources (see two sentences before this claim)?
2. In the proof of Theorem 1.6.3, I believe there is an off-by-one error and $t \leq full + 1$ should hold. Consider the input $(100, 1, 1), (100, 2, 1), (1, 1, 1)$ with $m = 2$. From my understanding, the two jobs with the hundred actions are scheduled on machines one and two first. Then the job with one action is scheduled to minute 101. Can you please clarify? As I am not proficient in reading pseudocode.

Overall grade	Excellent
Award level thesis	Yes

Date

Signature

[3]Only provided as it is customary to do so.

Minor comments

In expressions of the form `noun + number` such as Condition (1.2), minute 5, etc., I would not use a definite article before the noun.

The Czech "x se dá udělat z y" should be translated as "x can be (re)constructed from y", especially in the context of mathematics.

I suggest using small capitals font (`\textsc`) for typesetting problem names. The same applies for algorithm names and pseudocode variables, possibly with typewriter font (`\texttt`).

I suggest using emphasis (`\emph`) when introducing terminology. Some examples are going to be in Specific comments.

Sections 1.1 to 1.4 should be combined into the introduction for the journal publication.

Occasionally, you forget to separate lists with commas. $a, b \ldots, c$ should be $a, b, \ldots, c$.

- a
- b
- c

should be

- a,
- b, (and)
- c.

Introduction

The introduction should contain many more citations, which support made claims made by you. For an example, I would add some citations to the second paragraph where you describe what orienteering is.

Section 1.1

I suggest defining what a makespan is.

In Definition 1.1.1, I do not understand why the variables where chosen as they are. I can imagine that p in p_j stands for Czech "počet", but I do not follow the rest. Number of actions could be n_j or c_j for "count", resource could be r_j, and minimal period p_j.

"There is a fixed period of size at most g_j". I think it should be at least g_j.

In Definition 1.1.2, I would say that Condition (1.5) should not be a part of the definition of the problem. I understand that it helps with modelling the problem (as a CSP) and with bounding the cost of the solution, but it would be more suitable as an observation about the problem or the like. The definition should also stress out that each action of a job lasts one minute. This is mentioned in Section 1.3.

Specific comments

The following are just my suggestions and opinions.

- p. 2, second paragraph: "the map with marked control points". Use an indefinite article.
- p. 2, third paragraph: "the 10-years-old girls". Drop the article.

- p. 2, fourth paragraph: I would definitely add many more citations to the claim "While scheduling is a wide area".
- p. 2, fifth paragraph: "To summarize this". Drop this.
- p. 3, fourth paragraph: "The periodic job". Use indefinite article instead.
- p. 3, last paragraph: "there is a fixed period". The word fixed does not seem right in this context. Maybe common would be better.
- p. 4, fourth paragraph: "following way as follows".
- p. 5, -4^{th} paragraph: The parenthesis following "such a long schedule always exists" should be a footnote.
- p. 6, third paragraph: red job should be emphasized.
- p. 6, eighth paragraph: minutes $1, 6, 11, \ldots$ **belong** to the same... instead of belongs.
- p. 6, eighth paragraph: "the different congruence class" should be different congruence classes.
- p. 6, ninth paragraph: I do not think that this was used anywhere.
- p. 6, -3^{rd} paragraph: polynomial **in** the length of the input.
- p. 6, -3^{rd} paragraph: the list misses commas to separate items.
- p. 6, -3^{rd} paragraph: it is unclear what [12] refers to. I would add a sentence at the beginning saying "we describe ... as in [12]".
- p. 7, second paragraph: I would move the part which comments on the proof after the proof.
- p. 7, third paragraph: It is more usual to use recall instead of remind.
- p. 7, third paragraph: bound b should be emphasized.
- p. 7, third paragraph: I would stick citation [3] to Numerical 3D-matching, not after the paragraph.
- p. 8, last paragraph: "the period of each action is less than $\frac{1}{3}b'$. I believe it is **at most** instead of a sharp inequality.
- p. 8, last paragraph: the X'. I would omit the article.
- p. 8, last paragraph: see in Figure 1.2. I would omit **in**.
- p. 8, last paragraph: "k jobs corresponding to x'". I think it should be capital X'.
- p. 9, -4^{th}: These restrictions can be validate**d**.
- p. 10, Theorem 1.5.1. The statement claims that the lower bound is 2, while the preceding sentence claims that the lower bound is $2 - \varepsilon$.
- p. 11, first word: below instead of bellow.

- p. 11, fourth line of the pseudocode: contains Czech.
- p. 11, eighth line of the pseudocode: p_j should be p_i and G_j should be G_i.
- p. 11, Invariant 1.5.2: for the entire run of the Greedy algorithm **it** holds **that**.
- p. 11, proof of Invariant 1.5.2: after its **addition** the variant. . .
- p. 11, proof of Invariant 1.5.2: "denote that . . . ". Omit "that".
- p. 11, proof of Invariant 1.5.2: "could begin" should be could have begun.
- p. 11, last word for **the entire** schedule.
- p. 12, second paragraph: "the conditions are also valid . . . . I would refer to the appropriate line number in the pseudocode.
- p. 12, second paragraph: **I**nvariant 1.5.2.
- p. 13, pseudocode, seventh line: **while** *unused_jobs* $= \emptyset$ should have a $\neq$ sign.
- p. 13, pseudocode: argmax should be typeset with normal text. (`\operatorname*{argmax}` from the `amsmath` package)
- p. 13, -4^{th} paragraph: PRS should be PSR.
- p. 14, last line: it was cause**d** by . . . .
- p. 15, last paragraph: There is a strange long dash (`---`) without spaces surrounding it.
- p. 15, proof of Lemma 1.6.4: $S_{j_1} \geq \cdots \geq S_{j_m}$, use `\cdots` here.
- p. 17, last paragraph: I would clarify that "slots" (I do not think that this an established term) in these paragraphs are the ones without a gray job in minute 0.
- p. 17, last paragraph: no one can be combined into none.
- p. 20, first paragraph: "look at" and "look on"s is view in English. E.g., we can view this schedule. . . .
- p. 20, second paragraph: decreases **to**.
- p. 20, third paragraph: jobs with "the" period 1 are schedule**d**.
- p. 23, second paragraph: typeset $\log_2$ as `\log_2`.
- p. 23, second paragraph: not scheduled job is unscheduled job.