Operations Research Letters 51 (2023) 1–2

Contents lists available at ScienceDirect

Operations Research Letters

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

A historical note on the complexity of scheduling problems

Jan Karel Lenstra [a,*], Vitaly A. Strusevich [b], Milan Vlach [c]

[a] Centrum Wiskunde & Informatica, Amsterdam, the Netherlands
[b] Sherwood Road, Welling, Kent, UK
[c] Charles University, Prague, Czech Republic

ARTICLE INFO

Article history:
Available online 21 November 2022

Keywords:
Partition
Scheduling
Computational complexity
NP-hardness

ABSTRACT

In 1972 E.M. Livshits and V.I. Rublinetsky published a paper in Russian, in which they presented linear-time reductions of the partition problem to a number of scheduling problems. Unaware of complexity theory, they argued that, since partition is not known to have a simple algorithm, one cannot expect to find simple algorithms for these scheduling problems either. Their work did not go completely unnoticed, but it received little recognition. We describe the approach and review the results.

1. Complexity avant la lettre

In 1972 a paper titled "On the comparative complexity of some discrete optimization problems" appeared in a publication of the Ukrainian Academy of Sciences [10]. The authors were E.M. Livshits and V.I. Rublinetsky. They define the "stones problem", which is an optimization version of the partition problem, and reduce it to a number of scheduling problems involving a single machine, identical parallel machines and flow shops. They were evidently unaware of the contemporaneous development of complexity theory by Cook, Karp and Levin and do not cast their results in the framework of Turing machines, polynomial-time computation and completeness. Instead, their approach is entirely pragmatic: no simple solution method is known for the stones problem, and when they "find stones" in another problem, they do not waste time in looking for a simple solution for that problem either. They do not define when an algorithm is "simple". Similarly, they call their reductions "simple" just because these need about as many operations as there are stones.

The paper received little attention. Tanaev and Shkurba [12] list the paper among their references but do not cite it; their book does not deal with issues of complexity theory. The *Mathematical Reviews* published an abstract of the paper in 1978. The two volumes written by Tanaev and co-authors in the 1980's [13] [14] give due credit to Livshits and Rublinetsky or, from here on, L&R; their results are referred to as NP-hardness proofs. English translations of these books were published in 1994 but their distribution in the West has remained modest. We have found four papers that cite L&R; they all originate from the scheduling community in Minsk. The historical accounts by Trakhtenbrot [15] and Johnson [3] do not mention L&R.

Although the approach is informal and the repertoire is limited, L&R wrote a pioneering paper. It is an original and independent attempt to explain the intrinsic difficulty of combinatorial problems. L&R were probably the first researchers who systematically applied the concept of problem reduction in scheduling theory. The purpose of this note is to give them the recognition they deserve. We will summarize their work below. A translation of their paper is available on the website elementsofscheduling.nl.

We dedicate this paper to the memory of Gerhard Woeginger. Computational complexity was one of his prime loves. He liked elegant mathematical arguments as much as unexpected discoveries about their historical origin.

2. Stones and schedules

L&R say that problem P *reduces* to problem Q if for any instance I_P of P one can find an instance I_Q of Q and for any solution to such an instance I_Q one can find a solution to the corresponding instance I_P such that the set of optimal solutions to I_Q is mapped onto the set of optimal solutions to I_P. This definition is somewhat more complex than that of Karp [4], who considers problems with a yes-no answer and requires that I_P is a yes-instance if and only if I_Q is a yes-instance. While Karp uses *polynomial bounded* as a mathematical equivalent of *easy*, L&R require that the transformation from P to Q and the recoding of solutions to Q back to solutions to P are *computationally simple*, an intuitive notion that is left undefined. In their examples, the reductions are linear.

* Corresponding author.
E-mail addresses: jkl@cwi.nl (J.K. Lenstra), sv02pri@gmail.com (V.A. Strusevich), vlach@ktiml.mff.cuni.cz (M. Vlach).

https://doi.org/10.1016/j.orl.2022.11.006
0167-6377/

The *stones problem* is the problem of arranging n stones in two piles of maximal similarity by weight. More formally, given n positive numbers $a_1, \ldots, a_n$, find a subset $S \subset \{1, \ldots, n\}$ for which $|\sum_{j \in S} a_j - \sum_{j \in \{1,\ldots,n\}-S} a_j|$ has minimum value. L&R note that no simple algorithm is known for the solution of this problem, although, as a special case of the knapsack problem, it can be solved by enumerative methods. Indeed, dynamic programming solves the stones problem in $O(n \sum_{j=1}^{n} a_j)$ time [7].

L&R reduce the stones problem to the following scheduling problems. We use the notation for scheduling problems introduced by Graham et al. [2].

(1) $P2||T_{\max}$, the problem of minimizing maximum tardiness on two identical parallel machines. They set the due dates equal to zero and consider, in fact, the makespan problem $P2||C_{\max}$, which is identical to the stones problem.

(2) $1|r_j|T_{\max}$, minimize maximum tardiness on a single machine subject to release dates. The reduction was reinvented for the lateness problem $1|r_j|L_{\max}$ by Lenstra et al. [8].

(3) $P2||\sum w_j C_j$, minimize total weighted completion time on two identical parallel machines. The reduction is much simpler than that of Bruno et al. [1] and, again, reinvented in [8].

(4) $1|r_j|\sum w_j C_j$, minimize total weighted completion time on a single machine subject to release dates. Lenstra et al. [8] claim strong NP-hardness for $1|r_j|\sum C_j$, with the unweighted objective, but do not specify the reduction.

(5) $1||\sum w_j T_j$, minimize total weighted tardiness on a single machine. A (complex) strong NP-hardness proof due to Garey & Johnson is quoted by Lawler [6]; cf. [8].

(6) $1||\sum w_j U_j$, minimize the weighted number of late jobs on a single machine. L&R use a splitting job; Karp [4] does not in a reduction that is as lean as possible.

(7) $F3||C_{\max}$, minimize makespan in a three-machine flow shop, and three variants in with the first machine M_1, or the third machine M_3, or both M_1 and M_3 have an infinite capacity and can handle any number of jobs at the same time. In the case of machines with infinite capacity, the processing times on M_1 can be viewed as release times and those on M_3 as delivery times. Hence, the three variants are identical to $F2|r_j|C_{\max}$, $F2||L_{\max}$ and $1|r_j|L_{\max}$, respectively; the last of these is also known as the "head-body-tail problem" [5]. L&R reproduce a reduction of the stones problem to the head-body-tail problem given by Livshits in 1969 [9] and state that it will also work for $F3||C_{\max}$ and the two other variants mentioned above.

We note that the earlier paper by Livshits [9] deals with a resource constrained project management problem. He presents an approximation algorithm and gives an (again quite early) analysis of its absolute error. At the end of the paper he formulates a problem equivalent to the stones problem, emphasizes that it is the same as what we call $P2||C_{\max}$, and provides a reduction to a network model which is in fact the head-body-tail problem. This initial step towards the more systematic application of reductions by L&R may qualify as the first NP-hardness proof in the scheduling literature.

(8) $F3||C_{\max}$ where now the second machine M_2 has an infinite capacity. In this case, the processing times on M_2 can be viewed as transportation times or minimum delays in between M_1 and M_3. If one restricts attention to schedules in which each machine processes the jobs in the same order, which has become traditional in the flow shop literature, then an optimal solution can be found in $O(n \log n)$ time [11]. However, in the two-machine flow shop with delays it may be advantageous that a job passes another one in between M_1 and M_3. L&R give an ingenious reduction. Much later, Wenci Yu [16] gave a strong NP-hardness proof for the restricted case of unit processing times on M_1 and M_3 and arbitrary delays.

The reductions given by L&R are correct, modulo some minor inaccuracies in the proofs. They introduce techniques that have become standard, e.g., using a splitting job to separate the sets S and $\{1, \ldots, n\} - S$ and putting the weights equal to the processing times in problems with a weighted minsum objective. Their work is original and central to the development of scheduling theory.

Acknowledgement

We are grateful to Eugene Levner, who sent the paper by L&R [10] to the first author in April 1978, and to Yakov Shafransky, who supplied the earlier paper by Livshits [9].

References

[1] J.L. Bruno, E.G. Coffman Jr., R. Sethi, Scheduling independent tasks to reduce mean finishing time, Commun. ACM 17 (1974) 382–387.
[2] R.L. Graham, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, Optimization and approximation in deterministic sequencing and scheduling: a survey, Ann. Discrete Math. 5 (1979) 287–326.
[3] D.S. Johnson, A brief history of NP-completeness, 1954–2012, in: Documenta Mathematica; Extra Volume: Optimization Stories; 21st International Symposium on Mathematical Programming, 2012, pp. 359–376.
[4] R.M. Karp, Reducibility among combinatorial problems, in: R.E. Miller, J.W. Thatcher (Eds.), Complexity of Computer Computations, Plenum Press, New York, 1972, pp. 85–103.
[5] B.J. Lageweg, J.K. Lenstra, A.H.G. Rinnooy Kan, Minimizing maximum lateness on one machine: computational experience and some applications, Stat. Neerl. 30 (1976) 25–41.
[6] E.L. Lawler, A 'pseudopoloynomial' algorithm for sequencing jobs to minimize total tardiness, Ann. Discrete Math. 1 (1977) 331–342.
[7] E.L. Lawler, J.M. Moore, A functional equation and its application to resource allocation and sequencing problems, Manag. Sci. 16 (1969) 77–84.
[8] J.K. Lenstra, A.H.G. Rinnooy Kan, P. Brucker, Complexity of machine scheduling problems, Ann. Discrete Math. 1 (1977) 343–362.
[9] E.M. Livshits, Estimates of accuracy of approximate solutions to some optimization problems in project management, in: Proceedings of the First Winter Workshop on Mathematical Programming 3, Moscow, 1969, pp. 477–497 (in Russian).
[10] E.M. Livshits, V.I. Rublinetsky, On the comparative complexity of some discrete optimization problems, Numerical Mathematics and Computer Technology, vol. 3, Physics-Technology Institute for Low Temperatures, Academy of Sciences of the Ukrainian SSR, Kharkov, 1978, pp. 78–85 (in Russian). Reviewed in Math. Rev. 56 (17808) (1978) 2309, Translation in English on elementsof-scheduling.nl.
[11] L.G. Mitten, Sequencing n jobs on two machines with arbitrary time lags, Manag. Sci. 5 (1959) 293–298.
[12] V.S. Tanaev, V.V. Shkurba, Introduction to Scheduling Theory, Nauka, Moscow, 1975 (in Russian).
[13] V.S. Tanaev, V.S. Gordon, Y.M. Shafransky, Scheduling Theory; Single-Stage Systems, Nauka, Moscow, 1984 (in Russian). Updated and revised translation in English Kluwer, Dordrecht, 1994.
[14] V.S. Tanaev, Y.N. Sotskov, V.A. Strusevich, Scheduling Theory; Multi-Stage Systems, Nauka, Moscow, 1989 (in Russian). Updated and revised translation in English Kluwer, Dordrecht, 1994.
[15] B.A. Trakhtenbrot, A survey of Russian approaches to perebor (brute-force search), Ann. Hist. Comput. 6 (1984) 384–400.
[16] Wenci Yu, J.A. Hoogeveen, J.K. Lenstra, Minimizing makespan in a two-machine flow shop with delays and unit-time operations is NP-hard, J. Sched. 7 (2004) 333–348.