Scheduling Imprecise Computations on Parallel Machines with Linear and Non-Linear Error Penalties*

Akiyoshi Shioura † Natalia V. Shakhlevich (speaker) ‡
Vitaly A. Strusevich §

1 Introduction

We consider scheduling problems that have been mainly studied within the body of research on imprecise computation. In these problems computation tasks have to be assigned to parallel processors in such a way that their mandatory parts are fully executed, while their optional parts are processed only if a task can complete before the due date. The remaining optional part is understood as the error of computation for a task. The objectives include minimizing the total weighted error, the maximum weighted error, and various constrained versions, such as minimizing the total error subject to smallest maximum error. Additionally, we also study the problems of minimizing the quadratic error cost function and its various constrained versions.

Unlike the earlier algorithms, which are often applicable to only specific versions of the problem, the new approach we propose uses a common tool based on advanced network flow techniques, namely parametric max-flow in combination with improved algorithms for bipartite networks. It is applicable to a broad range of problems, with linear and non-linear objectives, is easier to justify and analyze, and achieves the time complexity known for solving the feasibility versions of the same problems with fixed processing times.

2 Description of models

Formally, in the imprecise computation model the jobs of set $N = \{1, 2, \ldots, n\}$ have to be processed on parallel machines. For each job $j \in N$, its processing time $p(j)$ is not given in advance but has to be chosen by the decision-maker from a given interval $[l(j), u(j)]$, where $l(j)$ is the duration of the mandatory part, while the remaining part $u(j) - l(j)$ is optional. The value $x(j) = u(j) - p(j)$ is the computation error which affects the accuracy of computation.

Each job $j \in N$ is given a release date $r(j)$ and a deadline $d(j)$. Processing of a job can be preempted and resumed later, possibly on another machine. Typically, the

*This research was supported by the EPSRC funded project EP/J019755/1

†shioura@dais.is.tohoku.ac.jp. Graduate School of Information Sciences, Tohoku University, Sendai, Japan.

‡n.shakhlevich@leeds.ac.uk. School of Computing, University of Leeds, Leeds LS2 9JT, U.K.

§v.strusevich@gre.ac.uk. Department of Mathematical Sciences, University of Greenwich, Old Royal Naval College, Park Row, London SE10 9LS, U.K.

problems of imprecise computation are those of finding a deadline feasible preemptive schedule that minimizes a certain function F that depends on errors, e.g., the total error cost or the maximum error cost. We associate each job j with two unit-costs, $w_T(j)$ and $w_M(j)$. In such a doubly-weighted system of imprecise computation the costs $w_T(j)$ are involved in computing the total error cost, which in the linear case is defined as

$$F_\Sigma = \sum_{j\in N} w_T(j)\, x(j)$$

and in the quadratic case as

$$F_{\text{quad}} = \sum_{j\in N} w_T(j) x(j)^2.$$

Similarly, the costs $w_M(j)$ are involved in computing the maximum error cost, which is defined as

$$F_{\max} = \max\{x(j)/w_M(j)\,|j \in N\}.$$

As far as the machine environment is concerned, we are given m parallel machines. Identical machines have the same speed, so that for a job j with an actual processing time $p(j)$ the total length of the time intervals in which this job is processed in a feasible schedule is equal to $p(j)$. If the machines are uniform, then it is assumed that machine M_i has speed s_i, $1 \le i \le m$.

Depending on the machine environment and the objective function we generically denote the problems under consideration by $\Pi_\beta(\alpha)$, where $\beta \in \{\Sigma, \max, \text{quad}\}$ is the objective function, and $\alpha \in \{P, Q\}$ is the machine system, identical (P) or uniform (Q). For example, $\Pi_\Sigma(P)$ denotes the problem of minimizing the total error cost on identical machines, while $\Pi_{\max}(Q)$ denotes the problem of minimizing the maximum error cost on uniform machines.

As is traditional in the imprecise computation literature, we also look at the constrained problems, which we denote by $\Pi_{\beta'|\beta''}(\alpha)$. For these problems, the objective function $F_{\beta'}$ is minimized in the class of the schedules with the minimum value of $F_{\beta''}$, where $\beta', \beta'' \in \{\Sigma, \max, \text{quad}\}$, $\beta' \neq \beta''$. For example, $\Pi_{\Sigma|\max}(P)$ denotes the problem of finding a schedule on parallel identical machines that minimizes the total error cost among all schedules with the smallest maximum error cost.

Each problem with $p(j) \in [l(j), u(j)]$ can be seen as an extension of the feasibility problem $\Pi(\alpha)$, in which the processing times of all jobs are fixed, i.e., equal to given values $p(j), 1 \le j \le n$. To solve problem $\Pi(\alpha)$ means either to find a feasible schedule for the corresponding machine environment if it exists or to report that such a schedule does not exist.

3 The main result

Theorem 1 *For $\beta \in \{\Sigma, \max, \text{quad}\}$, each problem $\Pi_\beta(P)$ on m identical parallel machines is solvable in $O(n^3)$ time, and each problem $\Pi_\beta(Q)$ on m uniform parallel machines is solvable in $O(mn^3)$ time. The constrained versions $\Pi_{\beta'|\beta''}(P)$ and $\Pi_{\beta'|\beta''}(Q)$, where $\beta', \beta'' \in \{\Sigma, \max, \text{quad}\}$, $\beta' \neq \beta''$, are also solvable in $O(n^3)$ and $O(mn^3)$ time, respectively. These running times meet the best known running times required for solving the feasibility problems $\Pi(P)$ and $\Pi(Q)$, respectively.*

For comparison, the table below lists the complexity estimates of the known approaches.

Identical machines			Uniform machines		
$\Pi_\Sigma(P)$	$O\left(n^4 \log n\right)^*$	[1, 5, 7]	$\Pi_\Sigma(Q)$	$O\left(m^2 n^4 \log mn\right)$	[5]
				$O\left(mn^4\right)$	[6]
$\Pi_{\max}(P)$	$O\left(n^4\right)^*$	[2, 4]	$\Pi_{\max}(Q)$	$O\left(mn^4\right)$	[8]
$\Pi_{\Sigma\mid\max}(P)$	$O\left(n^4\right)^*$	[2, 5]	$\Pi_{\Sigma\mid\max}(Q)$	$O\left(mn^4\right)$	[8]
$\Pi_{\max\mid\Sigma}(P)$	$O\left(n^5\right)^*$ (if all w_T-weights are distinct)	[2, 3]	$\Pi_{\max\mid\Sigma}(Q)$	$O\left(mn^5\right)$ (if all w_T-weights are distinct)	[8]

*after correcting a faulty claim that problem $\Pi(P)$ is solvable in $O(n^2 \log^2 n)$ time

Notice that the quadratic objective F_{quad} has not been studied in the past, while our technique handles it as a slight generalization. The complexity estimate for all versions of the problem involving F_{quad}, even in combination with Σ or max, remains the same as for the feasibility problem $\Pi(\alpha)$.

References

[1] J.Y. Chung, W.-K. Shih, J.W.S. Liu, and D.W. Gillies (1989) Scheduling imprecise computations to minimize total error. *Microproc. Microprog.* 27: 767–774.

[2] K.I.-J. Ho (2004) Dual criteria optimization problems for imprecise computation tasks. In *Handbook of Scheduling: Algorithms, Models and Performance Analysis*, ed. J.Y.-T. Leung (Chapman & Hall/CRC) pp. 35-1 – 35-26.

[3] K.I.-J. Ho, and J. Y.-T. Leung (2004) A dual criteria preemptive scheduling problem for minimax error of imprecise computation tasks, *Int. J. Found. Comput. Sci.* 15: 717–731.

[4] K.I.-J. Ho, J. Y.-T. Leung, and W.-D. Wei (1994) Minimizing maximum weighted error for imprecise computation tasks, *J. Algorithms* 16: 431–452.

[5] J.Y.-T. Leung (2004) Minimizing total weighted error for imprecise computation tasks. In *Handbook of Scheduling: Algorithms, Models and Performance Analysis*, ed. J.Y.-T. Leung (Chapman & Hall/CRC, London), 34-1 – 34-16.

[6] N.V. Shakhlevich and V.A. Strusevich (2008) Preemptive scheduling on uniform parallel machines with controllable job processing times. *Algorithmica* 51**:** 451–473.

[7] W.-K. Shih, J.W.S. Liu, J.-Y. Chung, D.W. Gillies (1989) Scheduling tasks with ready times and deadlines to minimize average error, *ACM SIGOPS Oper. Syst. Review* 23: 14–28.

[8] G. Wan, J.Y.-T. Leung, M.L Pinedo (2007) Scheduling imprecise computation tasks on uniform processors, *Inform. Process. Lett.* 104, 45–52.