Parallel machine scheduling under uncertainty

Adam Kasperski[1], Adam Kurpisz[2], and Paweł Zieliński[2]

[1] Institute of Industrial Engineering and Management, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370, Wrocław, Poland, adam.kasperski@pwr.wroc.pl

[2] Faculty of Fundamental Problems of Technology, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370, Wrocław, Poland, {adam.kurpisz,pawel.zielinski}@pwr.wroc.pl

Abstract. In this paper a parallel machine scheduling problem with uncertain processing times is discussed. This uncertainty is modeled by specifying a scenario set containing K distinct processing time scenarios. The ordered weighted averaging aggregation (OWA) operator, whose special cases are the maximum and Hurwicz criteria, is applied to compute the best schedule. Some new positive and negative approximation results concerning the problem are shown.

Keywords: scheduling, parallel machines, uncertain processing times, OWA operator, robust optimization

1 Introduction and motivation

Scheduling under uncertainty is a wide and important area of operations research and computer science. In real life, the values of some parameters which appear in scheduling problems, for example processing times, are often uncertain. Every possible realization of the parameters which may occur is called a *scenario* and all the scenarios form a *scenario set*. Under uncertainty, no additional information in the scenario set, such as a *probability* or *possibility distribution*, is known (see, e.g., [11]). In the traditional robust approach to scheduling (see, e.g., [8, 9]) we wish to minimize a schedule cost in a worst case, for example its maximum cost over all scenarios. Hence the resulting solution can be very conservative and is appropriate if a decision maker is risk-averse. In this paper we generalize the min-max approach by using the *ordered weighted averaging aggregation operator* (shortly OWA) proposed in [13], which is well known in decision theory. The OWA operator allows a decision maker to take his/her attitude towards risk into account. In particular, the well known Hurwicz criterion under which we minimize a convex combination of the maximal (pessimistic) and the minimal (optimistic) cost is also a special case of the OWA operator.

In this paper we consider the parallel machine scheduling problem denoted as $P||C_{\max}$ in the commonly used Graham's notation. This is one of the basic and most extensively studied scheduling problems (see, e.g., [2]). The classical deterministic $P||C_{\max}$ problem is known to be NP-hard. Therefore, its version with uncertain parameters cannot be easier and is also NP-hard. If an optimization problem turns out to be NP-hard, then we would like to design some efficient approximation algorithms for it. An algorithm $\mathcal{A}$ is a k-*approximation algorithm* for a minimization problem if it runs in polynomial time and for every instance of this problem it returns a solution whose cost is at most $k \cdot OPT$, where OPT is the cost of an optimal solution. If an optimization problem admits a k-approximation algorithm, then we say that it is approximable within k. Sometimes it is possible to design a family of $(1+\epsilon)$-approximation algorithms for $\epsilon > 0$. Such a family is called a *polynomial time approximation scheme* (PTAS) or a *fully polynomial time approximation scheme* (FPTAS) if it is also polynomial in $1/\epsilon$. This is the best one could expect if P≠NP.

In this paper, we show some new positive and negative results concerning the approximation of the $P||C_{\max}$ problem with uncertain processing times and the OWA criterion. In particular, we prove that, contrary to the deterministic problem, the uncertain version of $P||C_{\max}$ with unbounded scenario set cannot have a PTAS if P≠NP. We also show that if the number of machines and scenarios are constant, then the problem with the min-max criterion admits an FPTAS and the problem with the OWA criterion can be solved in a pseudopolynomial time. Finally, we propose a mixed integer programming based approach (MIP) to get an optimal schedule for the Hurwicz criterion. In order to make the presentation clear, we place all the technical proofs in Appendix.

2 Parallel machine scheduling under uncertainty

Let $J = \{1, \dots, n\}$ be a set of jobs which must be processed on m identical parallel machines $M_1, \dots, M_m$. Each machine can process at most one job at a time. In the deterministic case, each job $j \in J$ has a nonnegative processing time p_j. We wish to assign each job to exactly one machine so that the maximum job completion time of the resulting schedule, called a *makespan*, is minimal. This problem is denoted as $P||C_{\max}$ in the commonly used Graham's notation (see, e.g., [2]). It is well known [10] that the deterministic problem with only two machines, is already NP-hard. However, if the number of machines is constant, then the problem admits an FPTAS [12]. On the other hand, if the number of machines is unbounded (it is a part of the input), then the problem becomes strongly NP-hard [4] but it admits a PTAS [1, 7]. The problem has also an efficient $(\frac{4}{3} - \frac{1}{3m})$-approximation algorithm based on a *list scheduling* according to the LPT rule [6].

Suppose that the job processing times are uncertain and $\Gamma = \{S_1, \dots, S_K\}$ contains $K \geq 1$ distinct processing time scenarios. Thus each scenario $S \in \Gamma$ is a vector $(p_1^S, p_2^S, \dots, p_n^S)$ of job processing times, which may appear with a positive but unknown probability. Let Π be the set of all schedules. We denote by $C_{\max}(\pi, S)$ the makespan of schedule $\pi \in \Pi$ under scenario S. We also use $\overline{C}_{\max}(\pi) = \max_{S \in \Gamma} C_{\max}(\pi, S)$ and $\underline{C}_{\max}(\pi) = \min_{S \in \Gamma} C_{\max}(\pi, S)$ to denote the maximal and the minimal makespans of schedule π over all scenarios in Γ, respectively. The OWA of a schedule π with respect to weights $\boldsymbol{w} = (w_1, \dots, w_K)$,

$w_1 + \cdots + w_K = 1$, $w_i \geq 0, i = 1, \ldots, K$, is defined as follows:

$$\mathrm{OWA}_{\boldsymbol{w}}(\pi) = \sum_{i=1}^{K} w_i C_{\max}(\pi, S_{\sigma(i)}), \tag{1}$$

where σ is a permutation of the set $\{1, \ldots, K\}$ such that $C_{\max}(\pi, S_{\sigma(1)}) \geq C_{\max}(\pi, S_{\sigma(2)}) \geq \cdots \geq C_{\max}(\pi, S_{\sigma(K)})$.

In the OWA $P||C_{\max}$ problem, we seek a schedule π minimizing $\mathrm{OWA}_{\boldsymbol{w}}(\pi)$ for a given vector of weights $\boldsymbol{w}$. It can be easily verified that if $w_1 = 1$ and $w_i = 0$ for $i = 2, \ldots, K$, then $\mathrm{OWA}_{\boldsymbol{w}}(\pi) = \overline{C}_{\max}(\pi)$ and the problem reduces to computing a min-max schedule, which is a typical goal in the *robust optimization* (see, e.g., [9]). We denote such a problem as Min-Max $P||C_{\max}$. We get another important case when $w_1 = \alpha$ and $w_K = (1 - \alpha)$ for $\alpha \in [0, 1]$. Then

$$\mathrm{OWA}_{\boldsymbol{w}}(\pi) = (1 - \alpha)\underline{C}_{\max}(\pi) + \alpha\overline{C}_{\max}(\pi) = H_\alpha(\pi)$$

is the well known Hurwicz criterion (see, e.g., [11]), which is a compromise between the best (optimistic) and the worst (pessimistic) case. We will denote the problem with the Hurwicz criterion as Hur $P||C_{\max}$. Obviously, if $\alpha = 1$, then we again get the Min-Max $P||C_{\max}$ problem.

3 Approximation results for the general problem

It is obvious that OWA $P||C_{\max}$, Min-max $P||C_{\max}$ and Hur $P||C_{\max}$ are NP-hard even if $m = 2$ and $K = 1$, which follows from the fact that their deterministic versions with only one scenario and two machines are already NP-hard [10, 4]. Furthermore, it turns out that Min-Max $P||C_{\max}$ is equivalent to the *vector scheduling problem* discussed in [3]. So, all the results obtained in [3] applies to Min-Max $P||C_{\max}$ as well. In particular, if m is unbounded and K is constant, then the problem admits a PTAS but if both m and K are unbounded, then the problem is not approximable within any constant factor unless NP=ZPP [3] (if we use the results presented in [14], the we can state the above hardness approximation result assuming only P≠NP). Since both OWA $P||C_{\max}$ and Hur $P||C_{\max}$ include Min-Max $P||C_{\max}$ as a special case, they are also not approximable within any constant factor if m and K are unbounded unless P=NP.

In this section we wish to investigate the case when the number of machines m is constant (in particular $m = 2$) and the number of scenarios K is unbounded. The following theorem is the main result of this section:

Theorem 1. *If the number of scenarios K is unbounded and $m = 2$, then OWA $P||C_{\max}$ with $w_K = \alpha$, $w_{\lceil K/2 \rceil + 1} = 1 - \alpha$ and $w_i = 0$, for all $i \neq 1, \lceil K/2 \rceil + 1$, $\alpha \in [0, 1]$, is not approximable within $\frac{3\alpha + \frac{3}{2}(1-\alpha)}{2\alpha + \frac{3}{4}(1-\alpha)} - \epsilon$ for any $\epsilon > 0$ unless $P = NP$.*

Proof. See Appendix. $\square$

Theorem 1 applied to the extreme values of α leads to the following corollary:

Corollary 1. *If the number of scenarios K is unbounded and $m = 2$, then* Min-Max $P||C_{\max}$ *and* Hur $P||C_{\max}$ *are not approximable within $3/2 - \epsilon$ and* OWA $P||C_{\max}$ *is not approximable within $2 - \epsilon$ for any $\epsilon > 0$ unless $P = NP$. Hence all these problems do not admit a PTAS.*

The following theorem establishes a positive approximation result for the general problem:

Theorem 2. *For any schedule $\pi \in \Pi$ it holds* $\mathrm{OWA}_{\boldsymbol{w}}(\pi) \leq m \cdot \mathrm{OWA}_{\boldsymbol{w}}(\pi^*)$, *where π^* is an optimal schedule. Hence* OWA $P||C_{\max}$ *is approximable within m.*

Proof. See Appendix. □

Corollary 1 and Theorem 2 allow us to close the approximation gap for the OWA $P||C_{\max}$ problem.

Corollary 2. OWA $P||C_{\max}$ *with two machines ($m = 2$) is approximable within 2 but not approximable within $2 - \epsilon$ for any $\epsilon > 0$ if P≠NP.*

4 Approximation results for Min-Max $P||C_{\max}$

In this section, we discuss the approximation of Min-Max $P||C_{\max}$. Let us first consider the case, when m and K are unbounded. In [3] a very simple $O(\ln Km/\ln\ln Km)$-approximation randomized algorithm and a deterministic (K+1)-approximation algorithm for the *vector scheduling problem* have been proposed. This problem is equivalent to Min-Max $P||C_{\max}$. Hence, both algorithms can also be applied to Min-Max $P||C_{\max}$. The first one simply assigns each job to a machine chosen uniformly at random (see Algorithm 1).

Algorithm 1: Randomized algorithm for Min-Max $P||C_{\max}$.

foreach $j \in J$ **do**
  assign job j to exactly one machine $M_1, \dots, M_m$, with the probability $1/m$;

Notice that the random mechanism in Algorithm 1 can be realized by casting for each job $j \in J$ a symmetric m-faced dice whose face probabilities are $1/m$. Since each job is assigned to exactly one machine, Algorithm 1 returns a feasible schedule. The idea of the second algorithm, proposed in [3], (Algorithm 2) is to apply the classical list scheduling algorithm (see e.g., [6]) to a particular processing time scenario.

The following theorems give approximation bounds for schedules constructed by Algorithm 1 and Algorithm 2:

Theorem 3 ([3]). *Algorithm 1 returns an $O(\ln(Km)/\ln\ln(Km))$-approximate schedule with high probability.*

Algorithm 2: Approximation algorithm for MIN-MAX $P||C_{\max}$.

foreach $j \in J$ **do**
  $p_j^{\widehat{S}} \leftarrow \sum_{S \in \Gamma} p_j^S$ /*Construct an auxiliary scenario $\widehat{S}$ */

/*Apply List Scheduling Algorithm for jobs in J with $p_j^{\widehat{S}}, j \in J$ */
foreach $j \in J$ **do**
  assign j to the machine with the smallest load;

Theorem 4 ([3]). *Algorithm 2 returns a $(K+1)$-approximate schedule.*

Since MIN-MAX $P||C_{\max}$ is a special case of OWA $P||C_{\max}$ with $\alpha = 1$, Theorems 2 and 4 immediately imply MIN-MAX $P||C_{\max}$ is approximable within $\min\{m, K+1\}$. Thus in particular it is approximable within 2 for two machines. There is still and open gap, because we know (see Corollary 1) that the two-machine case is not approximable within $3/2 - \epsilon$ for any $\epsilon > 0$.

We now discuss the case in which both the number of machines m and the number of scenarios K are constant. We first show that the general OWA $P||C_{max}$ problem can be solved in pseudopolynomial time. Let us define a *load vector*:

$$\boldsymbol{L} = (L_{11}, \ldots, L_{m1}, L_{12}, \ldots, L_{m2}, \ldots, L_{1K}, \ldots, L_{mK}),$$

where L_{ik} is the load of machine M_i under scenario S_k for some partial schedule. Let $L_{\max} = \max_{S \in \Gamma} \sum_{j \in J} p_j^S$. Clearly $L_{ik} \leq L_{\max}$ for all $i = 1, \ldots, m$ and $k = 1, \ldots, K$. Therefore, the number of distinct load vectors is bounded by $l = (L_{\max})^{mK}$. We can now use a simple dynamic algorithm to compute an optimal schedule. An idea of that algorithm is illustrated in Fig. 1.

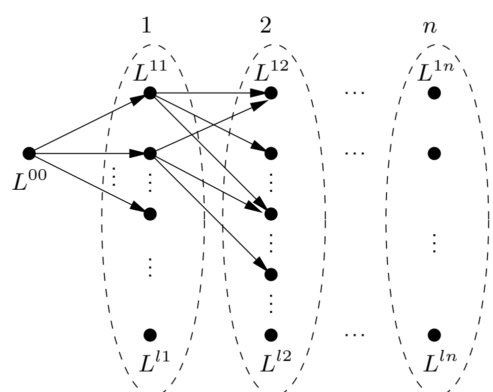

Fig. 1. An illustration of the exact algorithm.

We first build a directed acyclic network composed of node $\boldsymbol{L}^{00} = (0, \ldots, 0)$ and n layers, where the jth layer is composed of nodes $\boldsymbol{L}^{1j}, \ldots, \boldsymbol{L}^{lj}$. The arc between $\boldsymbol{L}^{uj-1}$ and $\boldsymbol{L}^{vj}$ exists if and only if the load vector $\boldsymbol{L}^{vj}$ can be obtained from $\boldsymbol{L}^{uj-1}$ by placing the job j on some machine. This network can be built in $O(nml) = O(nm(L_{\max})^{mK})$ time, since it contains $nl+1$ nodes and each node has at most m outgoing arcs. The last, nth, layer contains the load vectors of at most l complete schedules and the vector corresponding to schedule π with the minimum value of $\mathrm{OWA}_{\boldsymbol{w}}(\pi)$ is optimal. This optimal schedule can be obtained by using simple backward computations in the network constructed. The overall running time of the algorithm is $O(nm(L_{\max})^{mK}) = O(nm(np_{\max})^{mK}) = O(n^{mK+1}m(p_{\max})^{mK})$. Theorem 5 summarizes this result:

Theorem 5. OWA $P||C_{\max}$ *can be solved in* $O(n^{mK+1}m(p_{\max})^{mK})$ *time, which is pseudopolynomial if* m *and* K *are constant.*

Using theorem 5 we get the following result:

Theorem 6. *For any instance of the* Min-max $P||C_{\max}$ *problem with constant* m *and* K *and any* $\epsilon \in (0,1)$*, there is an* $(1+\epsilon)$*-approximation algorithm which runs in* $O(n^{2mK+1}m(1/\epsilon)^{mK})$ *time. Hence,* Min-max $P||C_{\max}$ *with constant* m *and* K *admits and FPTAS.*

Proof. see Appendix. □

5 Some results for Hur $P||C_{\max}$

It is not difficult to see that Hur $P||C_{\max}$ can be represented as follows:

$$\begin{array}{lll} H_\alpha(\pi^*) = \min\ (1-\alpha)u + \alpha v & & \\ \quad \text{s.t.} \quad u \geq C_{\max}(\pi, S_k) - M(1-\delta_k), & k = 1,\dots,K, & \\ \qquad\qquad v \geq C_{\max}(\pi, S_k), & k = 1,\dots,K, & \\ \qquad\qquad \pi \in \Pi, & & \\ \qquad\qquad \sum_{k=1}^{K} \delta_k = 1, & & \\ \qquad\qquad \delta_k \in \{0,1\}, & k = 1,\dots,K, & \end{array} \tag{2}$$

where M is a number greater than all the possible values of $C_{\max}(\pi, S_k)$ for $k=1,\dots,K$. The first K constraints ensure that $u = \underline{C}_{\max}(\pi)$ and the next K constraints set $v = \overline{C}_{\max}(\pi)$. Let $x_{ij} \in \{0,1\}$ and $x_{ij}=1$ if job j is placed on machine M_i. Then problem (2) can be represented as the following mixed integer linear programming problem, which can be solved by some standard off-the-shelf MIP solvers.:

$$\begin{array}{lll} & \min\ (1-\alpha)u + \alpha v & \\ \text{s.t.} & u \geq \sum_{j=1}^{n} x_{ij} p_j^{S_k} - M(1-\delta_k), & i=1,\dots,m, k=1,\dots,K, \\ & v \geq \sum_{j=1}^{n} x_{ij} p_j^{S_k}, & i=1,\dots,m, k=1,\dots,K, \\ & \sum_{i=1}^{m} x_{ij} = 1, & j=1,\dots,n, \\ & \sum_{k=1}^{K} \delta_k = 1, & \\ & \delta_k \in \{0,1\}, & k=1,\dots,K, \\ & x_{ij} \in \{0,1\}, & i=1,\dots,m, j=1,\dots,n. \end{array} \tag{3}$$

Clearly, if $\alpha = 1$, then we get the MIN-MAX $P||C_{\max}$ problem to which all the approximation results shown in Section 4 can be applied. The second extreme case, when $\alpha = 0$, is even easier. Because $H_0(\pi^*) = \min_{\pi \in \Pi} \min_{S \in \Gamma} C_{\max}(\pi, S) = \min_{S \in \Gamma} \min_{\pi \in \Pi} C_{\max}(\pi, S)$, we get an optimal solution by solving K deterministic problems for scenarios $S_1, \ldots, S_K$. Furthermore, each deterministic problem can be approximated within $(4/3-1/(3m))$ by applying the list scheduling algorithm according to LPT rule. In consequence the case $\alpha = 0$ is also approximable within $(4/3-1/(3m))$.

Theorem 2 shows that HUR $P||C_{max}$ with two machines is approximable within 2 even if the number of scenarios is unbounded. On the other hand this problem is not approximable within $3/2-\epsilon$ for any $\epsilon > 0$ (see Corollary 1). So closing this gap is an interesting open problem.

6 Conclusions

In this paper, we have discussed one of the basic scheduling problems denoted as $P||C_{\max}$. We have considered the situation in which the job processing times are uncertain and all the possible vectors of the processing times form a given scenario set. Then an additional criterion is required to choose a solution. We have adopted the well known OWA operator, which is commonly used in decision theory. This operator generalizes the min-max criterion typically used in robust optimization and allows us to model various attitudes towards the risk. The deterministic version of the problem is already NP-hard, even for two machines. So, the problem with uncertain processing times cannot be easier. In fact, we have shown that the general problem with the OWA criterion is hard to approximate even for two machines. However, we have also provided some positive results. In particular, we have strengthen approximation results for the min-max criterion if the number of machines and scenarios are constant. Under this criterion only a randomized approximation algorithm and a simple version of the list scheduling algorithm have been proposed so far. We have shown that in this case the problem with the min-max criterion admits an FPTAS and the problem with the OWA criterion can be solved in a pseudopolynomial time. For the Hurwicz criterion we have proposed a mixed integer programming based approach that gives us an optimal schedule.

There are a number of open problems concerning the considered problem. There are still some gaps between the positive and negative results which should be closed. Also, no approximation algorithm for the general OWA criterion is known. One can try to transform the pseudopolynomial time algorithm into an FPTAS. We can also consider some generalization of the $P||C_{\max}$ problem, for example by assuming that the machines are not identical. We believe that it is an interesting area of further research.

Acknowledgments. The first and the third author of the paper were partially supported by Polish Committee for Scientific Research, grant N N206 492938.

References

1. Alon, N., Azar, Y., Woeginger, G.J., Yadid, T.: Approximation Schemes for Scheduling on Parallel Machines. Journal of Scheduling 1, 55–66 (1998)
2. Brucker, P.: Scheduling Algorithms. Springer Verlag, Heidelberg, 5th edition (2007)
3. Chekuri, C., Khanna, S.: On Multi-dimensional Packing Problems. SIAM Journal on Computing 33, 837–851 (2004)
4. Garey, M.R., Johnson, D.S.: "Strong" NP-Completeness Results: Motivation, Examples, and Implications. Journal of the ACM 25, 499–508 (1978)
5. Garey, M.R., Johnson, D.S.: Computers and Intractability. A Guide to the Theory of NP-Completeness. W. H. Freeman and Company (1979)
6. Graham, R.L.: Bounds on Multiprocessing Timing Anomalies. SIAM Journal of Applied Mathematics 17, 416–429 (1969)
7. Hochbaum, D.S., Shmoys, D.B.: Using dual approximation algorithms for scheduling problems: theoretical and practical results. Journal of the ACM 34, 144–162 (1987)
8. Kouvelis, P., Daniels, R.L., Vairaktarakis, G.: Robust scheduling of a two-machine flow shop with uncertain processing times. IIE Transactions 32, 421–432 (2000)
9. Kouvelis, P., Yu, G.: Robust Discrete Optimization and its applications. Kluwer Academic Publishers (1997)
10. Lenstra, J.K., Rinnooy Kan, A., Brucker, P.: Complexity of Machine Scheduling Problems. In: Hammer, P., Johnson, E.L., Korte, B.H., Nemhauser, G.L. (eds.) Studies in Integer Programming, Annals of Discrete Mathematics, vol. 1, pp. 343–363. North-Holland Publishing Company (1977)
11. Luce, R., Raiffa, H.: Games and Decisions: Introduction and a Critical Survey. Wiley (1957)
12. Sahni, S.K.: Algorithms for Scheduling Independent Tasks. Journal of the ACM 23, 116–127 (1976)
13. Yager, R.R.: On Ordered Weighted Averaging Aggregation Operators in Multi-Criteria Decision Making. IEEE Transactions on Systems, Man and Cybernetics 18, 183–190 (1988)
14. Zuckerman, D.: Linear Degree Extractors and the Inapproximability of Max Clique and Chromatic Number. Theory of Computing 3, 103–128 (2007)

Appendix

Proof of Theorem 1. We show a gap-introducing reduction from the following NAE-3SAT problem, which is known to be NP-hard [5]:

NAE-3SAT: *Input:* A set of Boolean variables $x_1, \ldots, x_q$ and a set of clauses $\mathcal{C} = \{C_1, \ldots, C_r\}$, where each clause contains at most three literals.
Question: Is there a truth assignment to the variables such that each clause in $\mathcal{C}$ has at least one true literal and at least one false literal?

Given an instance of NAE-3SAT, we define for each literal l_i, where $l_i = x_i$ or $l_i = \overline{x}_i$, a job J_{l_i}. For each clause $(l_1 \vee l_2 \vee l_3) \in \mathcal{C}$, we create four *clause scenarios*: S such that $p^S_{l_1} = p^S_{l_2} = p^S_{l_3} = 1$ and three scenarios S', S'', S''', corresponding to all pairs of the literals in the clause, such that $p^{S'}_{l_1} = p^{S'}_{l_2} = \frac{3}{4}$, $p^{S''}_{l_1} = p^{S''}_{l_3} = \frac{3}{4}$,

$p_{l_2}^{S'''} = p_{l_3}^{S'''} = \frac{3}{4}$. The processing times of all the remaining jobs under S, S', S'' and S''' are equal to 0. For each variable x_i, we create *variable scenarios* $\overline{S}$, $\overline{S}'$ such that $p_{x_i}^{\overline{S}} = p_{\overline{x}_i}^{\overline{S}} = \frac{3}{2}$, $p_{x_i}^{\overline{S}'} = p_{\overline{x}_i}^{\overline{S}'} = \frac{3}{4}$ and the processing times of all the remaining jobs under $\overline{S}$ and $\overline{S}'$ are equal to 0. A sample reduction is shown in Table 1. Obviously, $|\Gamma| = K = 4r + 2q$ and $|J| = 2q$ and thus the instance of OWA $P||C_{\max}$ can be constructed in polynomial time. Finally, we set $m = 2$, $w_1 = \alpha$ and $w_{2r+q+1} = 1 - \alpha$, where $\alpha \in [0, 1]$ is fixed, and $w_k = 0$, for all $k \neq 1, 2r + q + 1$.

Table 1. The reduction for $\mathcal{C} = \{(\overline{x}_1 \vee x_2 \vee \overline{x}_3), (x_1 \vee \overline{x}_2 \vee \overline{x}_4), (x_1 \vee x_2 \vee \overline{x}_4), (\overline{x}_1 \vee x_3 \vee x_4)\}$.

	S_1	S_1'	S_1''	S_1'''	S_2	S_2'	S_2''	S_2'''	S_3	S_3'	S_3''	S_3'''	S_4	S_4'	S_4''	S_4'''	$\overline{S}_1$	$\overline{S}_1'$	$\overline{S}_2$	$\overline{S}_2'$	$\overline{S}_3$	$\overline{S}_3'$	$\overline{S}_4$	$\overline{S}_4'$
J_{x_1}	0	0	0	0	1	$\frac{3}{4}$	$\frac{3}{4}$	0	1	$\frac{3}{4}$	$\frac{3}{4}$	0	0	0	0	0	$\frac{3}{2}$	$\frac{3}{4}$	0	0	0	0	0	0
$J_{\overline{x}_1}$	1	$\frac{3}{4}$	$\frac{3}{4}$	0	0	0	0	0	0	0	0	0	1	$\frac{3}{4}$	$\frac{3}{4}$	0	$\frac{3}{2}$	$\frac{3}{4}$	0	0	0	0	0	0
J_{x_2}	1	$\frac{3}{4}$	0	$\frac{3}{4}$	0	0	0	0	1	$\frac{3}{4}$	0	$\frac{3}{4}$	0	0	0	0	0	0	$\frac{3}{2}$	$\frac{3}{4}$	0	0	0	0
$J_{\overline{x}_2}$	0	0	0	0	1	$\frac{3}{4}$	0	$\frac{3}{4}$	0	0	0	0	0	0	0	0	0	0	$\frac{3}{2}$	$\frac{3}{4}$	0	0	0	0
J_{x_3}	0	0	0	0	0	0	0	0	0	0	0	0	1	$\frac{3}{4}$	0	$\frac{3}{4}$	0	0	0	0	$\frac{3}{2}$	$\frac{3}{4}$	0	0
$J_{\overline{x}_3}$	1	0	$\frac{3}{4}$	$\frac{3}{4}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\frac{3}{2}$	$\frac{3}{4}$	0	0
J_{x_4}	0	0	0	0	0	0	0	0	0	0	0	0	1	0	$\frac{3}{4}$	$\frac{3}{4}$	0	0	0	0	0	0	$\frac{3}{2}$	$\frac{3}{4}$
$J_{\overline{x}_4}$	0	0	0	0	1	0	$\frac{3}{4}$	$\frac{3}{4}$	1	0	$\frac{3}{4}$	$\frac{3}{4}$	0	0	0	0	0	0	0	0	0	0	$\frac{3}{2}$	$\frac{3}{4}$

If the answer to NAE-3SAT is *yes*, then according to a satisfying truth assignment, we form a schedule π by assigning the jobs corresponding to the true literals to machine M_1 and the jobs corresponding to the false literals to machine M_2. Let us sort the scenarios in nonincreasing order with respect to the values of the makespans for π, $C_{\max}(\pi, S_{\sigma(1)}) \geq C_{\max}(\pi, S_{\sigma(2)}) \geq \cdots \geq C_{\max}(\pi, S_{\sigma(K)})$. The jobs associated with the contradictory literals are placed on different machines and each clause has at least one truth and at least one false literal, which implies $C_{\max}(\pi, S_{\sigma(2r+q)}) \leq \cdots \leq C_{\max}(\pi, S_{\sigma(1)}) \leq 2$. The same reasoning yields $C_{\max}(\pi, S_{\sigma(2r+q+1)}) = \cdots = C_{\max}(\pi, S_{\sigma(K)}) = \frac{3}{4}$. So, $\mathrm{OWA}_{\boldsymbol{w}}(\pi) \leq 2\alpha + \frac{3}{4}(1 - \alpha)$. On the other hand, if the answer to NAE-3SAT is *no*, then for all schedules π at least two jobs corresponding to the contradictory literals are executed on the same machine, $C_{\max}(\pi, S_{\sigma(1)}) = 3$, $C_{\max}(\pi, S_{\sigma(2r+q+1)}) = \frac{3}{2}$ thus $\mathrm{OWA}_{\boldsymbol{w}}(\pi) = 3\alpha + \frac{3}{2}(1 - \alpha)$, or for at least one clause all three jobs corresponding to this clause are executed on the same machine, $C_{\max}(\pi, S_{\sigma(1)}) = 3$, $C_{\max}(\pi, S_{\sigma(2r+q+1)}) = \frac{3}{2}$ thus $\mathrm{OWA}_{\boldsymbol{w}}(\pi) = 3\alpha + \frac{3}{2}(1 - \alpha)$, where the permutation σ is such that $C_{\max}(\pi, S_{\sigma(1)}) \geq C_{\max}(\pi, S_{\sigma(2)}) \geq \cdots \geq C_{\max}(\pi, S_{\sigma(K)})$. So, OWA $P||C_{\max}$ is not approximable within $\frac{3\alpha + \frac{3}{2}(1-\alpha)}{2\alpha + \frac{3}{4}(1-\alpha)} - \epsilon$ for any $\epsilon > 0$ unless P=NP. □

Proof of Theorem 2 Under each scenario S, it holds $C_{\max}(\pi, S) \leq \sum_{j \in J} p_j^S$ and $C_{\max}(\pi^*, S) \geq \frac{1}{m} \sum_{j \in J} p_j^S$. This implies $C_{\max}(\pi, S) \leq m \cdot C_{\max}(\pi^*, S)$ for all $S \in \Gamma$. Let σ and ρ be two permutations of $\{1, \ldots, K\}$ such that $C_{\max}(\pi, S_{\sigma(1)}) \geq \cdots \geq C_{\max}(\pi, S_{\sigma(K)})$ and $C_{\max}(\pi^*, S_{\rho(1)}) \geq \cdots \geq C_{\max}(\pi^*, S_{\rho(K)})$. We now show that $C_{\max}(\pi, S_{\sigma(i)}) \leq m \cdot C_{\max}(\pi^*, S_{\rho(i)})$ for all $i = 1, \ldots, K$. Suppose by contradiction that this is not the case and $C_{\max}(\pi, S_{\sigma(j)}) > m \cdot C_{\max}(\pi^*, S_{\rho(j)})$ for some $j \in \{1, \ldots, K\}$. Suppose that there is a scenario S such that $S \in \{S_{\sigma(1)}, \ldots, S_{\sigma(j)}\}$ and $S \in \{S_{\rho(j+1)}, \ldots, S_{\rho(K)}\}$. Then it holds

$$C_{\max}(\pi, S) \geq C_{\max}(\pi, S_{\sigma(j)}) > m \cdot C_{\max}(\pi^*, S_{\rho(j)}) \geq m \cdot C_{\max}(\pi^*, S),$$

a contradiction. In consequence, each scenario which is in $\{S_{\sigma(1)}, \ldots, S_{\sigma(j)}\}$ must also be in $\{S_{\rho(1)}, \ldots, S_{\rho(j)}\}$ and $\{S_{\sigma(1)}, \ldots, S_{\sigma(j)}\} = \{S_{\rho(1)}, \ldots, S_{\rho(j)}\}$. Thus, in particular $S_{\rho(j)} \in \{S_{\sigma(1)}, \ldots, S_{\sigma(j)}\}$ and

$$C_{\max}(\pi, S_{\rho(j)}) \geq C_{\max}(\pi, S_{\sigma(j)}) > m \cdot C_{\max}(\pi^*, S_{\rho(j)}),$$

a contradiction. We thus get

$$\mathrm{OWA}_{\boldsymbol{w}}(\pi) = \sum_{i=1}^{K} w_i C_{\max}(\pi, S_{\sigma(i)}) \leq m \sum_{i=1}^{K} w_i C_{\max}(\pi^*, S_{\rho(i)}) = m \cdot \mathrm{OWA}_{\boldsymbol{w}}(\pi^*),$$

which completes the proof. □

Proof of Theorem 6. Let us fix $\epsilon \in (0, 1)$. For each scenario $S \in \Gamma$ define a *scaled scenario* $\widehat{S}$ under which $p_j^{\widehat{S}} = \left\lfloor \frac{n p_j^S}{\epsilon p_{\max}} \right\rfloor$, $j \in J$, where $p_{\max} = \max_{S \in \Gamma, j \in J} p_j^S$. If L_{ik} is the load of machine M_i under S_k for some schedule π, then $\widehat{L}_{ik}$ is the load of machine M_i under $\widehat{S}_k$ for π. Similarly, $\widehat{\overline{C}}_{\max}(\pi)$ is the maximal makespan of π over the set of scaled scenarios. It holds:

$$\frac{\epsilon p_{\max}}{n} p_j^{\widehat{S}} \leq p_j^S \leq \frac{\epsilon p_{\max}}{n} (p_j^{\widehat{S}} + 1).$$

Thus, for each $i = 1, \ldots, m$ and $k = 1, \ldots, K$:

$$\frac{\epsilon p_{\max}}{n} \widehat{L}_{ik} \leq L_{ik} \leq \frac{\epsilon p_{\max}}{n} \widehat{L}_{ik} + \epsilon p_{\max},$$

which implies that for any schedule π:

$$\frac{\epsilon p_{\max}}{n} \widehat{\overline{C}}_{\max}(\pi) \leq \overline{C}_{\max}(\pi) \leq \frac{\epsilon p_{\max}}{n} \widehat{\overline{C}}_{\max}(\pi) + \epsilon p_{\max}. \tag{4}$$

If $\widehat{\pi}$ is an optimal schedule for the scaled scenario set and π^* is an optimal schedule for the original scenario set, then $\widehat{\overline{C}}_{\max}(\widehat{\pi}) \leq \widehat{\overline{C}}_{\max}(\pi^*)$, $\overline{C}_{\max}(\pi^*) = OPT$, $OPT \geq p_{\max}$ and inequalities (4) imply:

$$\overline{C}_{\max}(\widehat{\pi}) \leq \frac{\epsilon p_{\max}}{n} \widehat{\overline{C}}_{\max}(\widehat{\pi}) + \epsilon p_{\max} \leq \frac{\epsilon p_{\max}}{n} \widehat{\overline{C}}_{\max}(\pi^*) + \epsilon p_{\max} \leq (1 + \epsilon) OPT.$$

Hence the cost of $\widehat{\pi}$ is within $(1 + \epsilon)OPT$. We can find $\widehat{\pi}$ by applying the exact pseudopolynomial time algorithm for the scaled scenarios (see Theorem 5). Since the maximal scaled processing time is not greater than n/ϵ, the schedule $\widehat{\pi}$ can be computed in $O(n^{2mK+1} m (1/\epsilon)^{mK})$ time. □