Controlling the Correlation of Cost Matrices to Assess Scheduling Algorithm Performance on Heterogeneous Platforms

Louis-Claude Canon, Pierre-Cyrille Héam, Laurent Philippe

CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2016; **00**:1–21
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cpe

Controlling the Correlation of Cost Matrices to Assess Scheduling Algorithm Performance on Heterogeneous Platforms†

L.-C. Canon[1,2*], P.-C. Héam[2], L. Philippe[2]

1. *Inria, ENS Lyon and University of Lyon, LIP laboratory, Lyon, France*
2. *FEMTO-ST Institute, CNRS, Univ. Bourgogne Franche-Comté, Besançon, France*
E-mail: [louis-claude.canon,pierre-cyrille.heam,laurent.philippe]@univ-fcomte.fr

SUMMARY

Bias in the performance evaluation of scheduling heuristics has been shown to undermine the scope of existing studies. Improving the assessment step leads to stronger scientific claims when validating new optimization strategies. This article considers the problem of allocating independent tasks to unrelated machines such as to minimize the maximum completion time. Testing heuristics for this problem requires the generation of cost matrices that specify the execution time of each task on each machine. Numerous studies showed that the task and machine heterogeneities belong to the properties impacting heuristics performance the most. This study focuses on orthogonal properties, the average correlations between each pair of rows and each pair of columns, which measure the proximity with uniform instances. Cost matrices generated with two distinct novel generation methods show the effect of these correlations on the performance of several heuristics from the literature. In particular, EFT performance depends on whether the tasks are more correlated than the machines and HLPT performs the best when both correlations are close to one. Copyright © 2016 John Wiley & Sons, Ltd.

Received ...

KEY WORDS: Scheduling; Cost Matrix; Correlation; Parallelism; Unrelated; Measure.

1. INTRODUCTION

The problem of scheduling tasks on processors is central in parallel computing science because it supports parts of the grid, computing centers and cloud systems [2]. Many papers [3–7] propose new or adapted scheduling algorithms that are assessed on simulators to prove their superiority. There is however no clear consensus on the superiority of one or another of these algorithms because they are usually tested on different simulators and parameters for the experimental settings. As for all experimental studies, a weak assessment step in a scheduling study may lead to bias in the conclusions (e.g., due to partial results or erroneous/misleading results). By contrast, improving the assessment step leads to a sounder scientific approach when designing new optimization strategies such as scheduling algorithms. In this context, using standardized experimental input data allows being in line with the open science approach because it enforces reproducibility [8].

This article tackles the problem of generating input instances to assess the performance of scheduling algorithms. Several input data impact the performance of such algorithms, among which the characteristics of the tasks and of the execution resources. In the cases when the tasks and their

*Correspondence to: louis-claude.canon@univ-fcomte.fr
†A preliminary version of this work appeared in Euro-Par 2016 [1]. The current article extends it with more detailed proofs in the analysis of existing generation methods, a new generation method and additional experiments and analysis of the behavior of scheduling heuristics.

execution times are deterministic, the performance results of the algorithm directly depend on the input instance. These cases correspond to the offline scheduling case where the algorithm takes a set of tasks and computes the whole schedule for a set of processors or nodes and to the online scheduling case where the algorithm dynamically receives tasks during the system execution and schedules them, one at a time, depending on the load state of execution resources. The performance of any heuristic for these problems is then given by the difference between the obtained optimization criterion (such as the makespan) and the optimal one. Of course, the performance of any scheduling algorithm depends on the properties of the input instance. Generating instances is thus a crucial problem in algorithm assessment [9, 10].

The previous scheduling cases correspond to numerous practical situations where a set of tasks, either identical or heterogeneous, must be distributed on platforms ranging from homogeneous clusters to grids and including semi-heterogeneous platforms such as CPU/GPU platforms [11] but also quasi-homogeneous systems such as clouds. In this context, several practical examples may be concerned by assessing the scheduling algorithm and adapting it depending on the execution resources characteristics, e.g. resource managers for heterogeneous environments as Condor [12], dedicated runtimes as Hadoop [13], batch schedulers or master/slave applications that are publicly distributed on a large variety of platforms [14] and must include a component that chooses where to run each task. In these examples, the choice of the scheduling algorithm is a key point for the software performance.

Three main parallel platform models that specify the instance have been defined: the *identical* case (noted P in the $\alpha|\beta|\gamma$ notation [15]), where the execution time of a task is the same on any machine that runs it; the *uniform* case (noted Q), where each execution time is proportional to the weight of the task and the cycle time of the machine (a common model); and, the *unrelated* case (noted R), where each task execution time depends on the machine. This article focuses on this last case in which an input instance consists in a matrix E where each element $e_{i,j}$ ($i \in T$, the task set and $j \in M$, the machine set) stands for the execution time of task i on machine j. Note that the unrelated case includes the identical and the uniform cases as particular cases. Hence, algorithm assessment for these two cases may also use a matrix as an input instance provided that this matrix respects the problem constraints (i.e., $\forall i \in T, \forall (j,k) \in M^2, e_{i,j} = \alpha_{j,k} \times e_{i,k}$ where $\alpha_{j,k} > 0$ is arbitrary for the uniform case and $\alpha_{j,k} = 1$ for the identical case).

To reflect the diversity of heterogeneous platforms, a fair comparison of scheduling heuristics must rely on a set of cost matrices that have distinct properties. Controlling the generation of synthetic random cost matrix in this context enables an assessment on a panel of instances that is sufficiently large to encompass practical settings that are currently existing or yet to come. In this generation, it is therefore crucial to identify and control the properties that impact the most critically the performance. Moreover, a hyperheuristic mechanism, which automates the heuristic selection, can exploit these properties through machine learning techniques or regression trees [16].

In a previous study [10], we already studied the problem of generating random matrices to assess the performance of scheduling algorithms in the unrelated case. In particular, we showed that the heterogeneity was previously not properly controlled despite having a significant impact on the relative performance of scheduling heuristics. We proposed both a measure to quantify the matrix heterogeneity and a method to generate instances with controlled heterogeneity. This previous work provided observations that are consistent with our intuition (e.g., all heuristics behave well with homogeneous instances), while offering new insights (e.g., the hardest instances have medium heterogeneity). In addition to providing an unbiased way to assess the heterogeneity, the introduced generation method produces instances that lie on a continuum between the identical case and the unrelated case.

In this article, we propose to investigate a more specific and finer continuum between the uniform case and the unrelated case. In the uniform case, each execution time is proportional to the weight of the task and the cycle time of the machine and, in the particular case where all the tasks have the same weight, an optimal solution can be found in polynomial time. By contrast, durations may be arbitrary in the unrelated case and finding an optimal solution is NP-Hard. In practice, however, the execution times may be associated to the task and machine characteristics: heavy tasks are more

DOI: 10.1002/cpe

likely to take a significant amount of time on any machine; analogously, efficient machines are more likely to perform any task quickly. Since unrelated instances are rarely arbitrary, our objective is to determine how heuristics are impacted by the degree at which an unrelated instance is close to a uniform one. In other words, we want to assess how scheduling algorithms respond when the considered tasks or machines are more or less uniform. We use the notion of correlation to denote this proximity (in particular, uniform instances have a correlation of one). This article provides the following contributions:[‡]

- a new measure, the correlation, for exploring a continuum between unrelated and uniform instances (Section 3);
- an analysis of this property in previous generation methods and previous studies (Section 3);
- an adaptation of a previous generation method and a new one with better correlation properties (Section 4);
- and, an analysis of the effect of the correlation on several static scheduling heuristics (Section 5).

The main issue addressed in this paper is the random generation of input instances to assess the performance of scheduling algorithms. It contains several technical mathematical proofs providing the theoretical foundations of the results. However, understanding these proofs is not required to understand the algorithms and the propositions. The reader unfamiliar with the mathematical notions can read the paper without reading the proofs.

2. RELATED WORK

This section first covers existing cost matrix generation methods used in the context of task scheduling. It continues then with different approaches for characterizing cost matrices.

The validation of scheduling heuristics in the literature relies mainly on two generation methods: the range-based and CVB (Coefficient-of-Variation-Based) methods. The range-based method [9, 19] generates n vectors of m values that follow a uniform distribution in the range $[1, R_{\text{mach}}]$ where n is the number of tasks and m the number of machines. Each row is then multiplied by a random value that follows a uniform distribution in the range $[1, R_{\text{task}}]$. The CVB method (see Algorithm 1) is based on the same principle except it uses more generic parameters and a distinct underlying distribution. In particular, the parameters consist of two coefficients of variation[§] (V_{task} for the task heterogeneity and V_{mach} for the machine heterogeneity) and one expected value (μ_{task} for the tasks). The parameters of the gamma distribution used to generate random values are derived from the provided parameters. An extension has been proposed to control the consistency of any generated matrix:[¶] the costs on each row of a submatrix containing a fraction of the initial rows and columns are sorted.

The shuffling and noise-based methods were later proposed in [10, 20]. They both start with an initial cost matrix that is equivalent to a uniform instance (any cost is the product of a task weight and a machine cycle time). The former method randomly alters the costs without changing the sum of the costs on each row and column. This step introduces some randomness in the instance, which distinguishes it from a uniform one. The latter (see Algorithm 2) relies on a similar principle: it inserts noise in each cost by multiplying it by a random variable with expected value one. Both methods require the parameters V_{task} and V_{mach} to set the task and machine heterogeneity.

[‡]The related code, data and analysis are available in [17]. Most of these results are also available in the companion research report [18] and in a conference paper [1].
[§]Ratio of the standard deviation to the mean.
[¶]In a consistent cost matrix, any machine faster than another machine for a given task will be consistently faster than this other machine for any task. Machines can thus be ordered by their efficiency.

DOI: 10.1002/cpe

Algorithm 1: CVB cost matrix generation with the gamma distribution [9, 19]

Input: n, m, V_{task}, V_{mach}, μ_{task}
Output: a $n \times m$ cost matrix

$\alpha_{\text{task}} \leftarrow 1/V_{\text{task}}^2$
$\alpha_{\text{mach}} \leftarrow 1/V_{\text{mach}}^2$
$\beta_{\text{task}} \leftarrow \mu_{\text{task}}/\alpha_{\text{task}}$
for all $1 \leq i \leq n$ **do**
  $q[i] \leftarrow G(\alpha_{\text{task}}, \beta_{\text{task}})$
  $\beta_{\text{mach}}[i] \leftarrow q[i]/\alpha_{\text{mach}}$
  for all $1 \leq j \leq m$ **do**
    $e_{i,j} \leftarrow G(\alpha_{\text{mach}}, \beta_{\text{mach}}[i])$
  end for
end for
return $\{e_{i,j}\}_{1 \leq i \leq n, 1 \leq j \leq m}$

In addition, the amount of noise introduced in the noise-based method can be adjusted through the parameter V_{noise}.

Algorithm 2: Noise-based cost matrix generation with gamma distribution [10]

Input: n, m, V_{task}, V_{mach}, V_{noise}
Output: a $n \times m$ cost matrix

for all $1 \leq i \leq n$ **do**
  $w_i \leftarrow G(1/V_{\text{task}}^2, V_{\text{task}}^2)$
end for
for all $1 \leq j \leq m$ **do**
  $b_j \leftarrow G(1/V_{\text{mach}}^2, V_{\text{mach}}^2)$
end for
for all $1 \leq i \leq n$ **do**
  for all $1 \leq j \leq m$ **do**
    $e_{i,j} \leftarrow w_i b_j \times G(1/V_{\text{noise}}^2, V_{\text{noise}}^2)$
  end for
end for
return $\{e_{i,j}\}_{1 \leq i \leq n, 1 \leq j \leq m}$

Once a cost matrix is generated, numerous measures can characterize its properties. The MPH (Machine Performance Homogeneity) and TDH (Task Difficulty Homogeneity) [21, 22] quantifies the amount of heterogeneity in a cost matrix. These measures present some major shortcomings such as the lack of interpretability [20]. Two alternative pairs of measures overcome these issues [10]: the coefficient of variation of the row means $V\mu_{\text{task}}$ and the mean of the column coefficient of variations μV_{task} for the task heterogeneity (the machine heterogeneity has analogous measures). These properties impact the performance of various scheduling heuristics and should be considered when comparing them.

This study focuses on the average correlation between each pair of tasks or machines in a cost matrix. No existing work considers this property explicitly. The closest work is the consistency extension in the range-based and CVB methods mentioned above. The consistency extension could be used to generate cost matrices that are close to uniform instances because cost matrices corresponding to uniform instances are consistent (machines can be ordered by their efficiency). However, this mechanism modifies the matrix row by row, which makes it asymmetric relatively to the rows and columns. This prevents its direct usage to control the correlation.

The TMA (Task-Machine Affinity) quantifies the specialization of a platform [21, 22], i.e., whether some machines are particularly efficient for some specific tasks. This measure proceeds

DOI: 10.1002/cpe

in three steps: first, it normalizes the cost matrix to make the measure independent from the matrix heterogeneity; second, it performs the singular value decomposition of the matrix; last, it computes the inverse of the ratio between the first singular value and the mean of all the other singular values. The normalization happens on the columns in [21] and on both the rows and columns in [22]. If there is no affinity between the tasks and the machines (as with uniform machines), the TMA is close to zero. Oppositely, if the machines are significantly specialized, the TMA is close to one. Additionally, Khemka et al [23] claims that high (resp., low) TMA is associated with low (resp., high) column correlation. This association is however not general because the TMA and the correlation can both be close to zero.

The range-based and CVB methods do not cover the entire range of possible values for the TMA [21]. Khemka et al [23] propose a method that iteratively increases the TMA of an existing matrix while keeping the same MPH and TDH. A method generating matrices with varying affinities (similar to the TMA) and which resembles the noise-based method is also proposed in [24]. However, no method with analytically proven properties has been proposed for generating matrices with a given TMA.

There is finally a field of study dedicated to the generation of random vectors given a correlation (or covariance) matrix that specifies the correlation between each pair of elements of a random vector [25–28]. The proposed techniques for sampling such vectors have been used for simulation in several contexts such as project management [29] or neural networks [30]. These approaches could be used to generate cost matrices in which the correlations between each pair of rows (resp., columns) is determined by a correlation matrix. However, the correlation between each pair of columns (resp., rows) would then be ignored. In this work, we assume that all non-diagonal elements of the correlation matrices associated with the rows and with the columns are equal.

3. CORRELATION BETWEEN TASKS AND PROCESSORS

As stated previously, the unrelated model (R) is more general than the uniform model (Q) and all uniform instances are therefore unrelated instances. Let $U = (\{w_i\}_{1\leq i\leq n}, \{b_j\}_{1\leq j\leq m})$ be a uniform instance with n tasks and m machines where w_i is the weight of task i and b_j the cycle time of machine j. The corresponding unrelated instance is $E = \{e_{i,j}\}_{1\leq i\leq n, 1\leq j\leq m}$ such that $e_{i,j} = w_i b_j$ is the execution time of task i on machine j. Our objective is to generate unrelated instances that are as close as desired to uniform ones. On the one hand, all rows are perfectly correlated in a uniform instance and this is also true for the columns. On the other hand, there is no correlation in an instance generated with nm independent random values. Thus, we propose to use the correlation to measure the proximity of an unrelated instance to a uniform one.

3.1. Correlation Properties

Let $e_{i,j}$ be the execution time for task i on machine j. Then, we define the *task correlation* as follows:

$$\rho_{\text{task}} \triangleq \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{i'=1, i'\neq i}^{n} \rho^r_{i,i'} \tag{1}$$

where $\rho^r_{i,i'}$ represents the correlation between row i and row i' as follows:

$$\rho^r_{i,i'} \triangleq \frac{\frac{1}{m}\sum_{j=1}^{m} e_{i,j}e_{i',j} - \frac{1}{m}\sum_{j=1}^{m} e_{i,j}\frac{1}{m}\sum_{j=1}^{m} e_{i',j}}{\sqrt{\frac{1}{m}\sum_{j=1}^{m} e^2_{i,j} - \left(\frac{1}{m}\sum_{j=1}^{m} e_{i,j}\right)^2}\sqrt{\frac{1}{m}\sum_{j=1}^{m} e^2_{i',j} - \left(\frac{1}{m}\sum_{j=1}^{m} e_{i',j}\right)^2}} \tag{2}$$

Note that any correlation between row i and itself is 1 and is hence not considered. Also, since the correlation is symmetric ($\rho^r_{i,i'} = \rho^r_{i',i}$), it is actually sufficient to only compute half of them.

DOI: 10.1002/cpe

Similarly, we define the *machine correlation* as follows:

$$\rho_{\text{mach}} \triangleq \frac{1}{m(m-1)} \sum_{j=1}^{m} \sum_{j'=1, j' \neq j}^{m} \rho^{c}_{j,j'} \tag{3}$$

where $\rho^{c}_{j,j'}$ represents the correlation between column j and column j' as follows:

$$\rho^{c}_{j,j'} \triangleq \frac{\frac{1}{n}\sum_{i=1}^{n} e_{i,j}e_{i,j'} - \frac{1}{n}\sum_{i=1}^{n} e_{i,j}\frac{1}{n}\sum_{i=1}^{n} e_{i,j'}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n} e^2_{i,j} - \left(\frac{1}{n}\sum_{i=1}^{n} e_{i,j}\right)^2}\sqrt{\frac{1}{n}\sum_{i=1}^{n} e^2_{i,j'} - \left(\frac{1}{n}\sum_{i=1}^{n} e_{i,j'}\right)^2}} \tag{4}$$

These correlations are the average correlations between each pair of distinct rows or columns. They are inspired by the classic Pearson definition, but adapted to the case when we deal with two vectors of costs.

The following two cost matrix examples illustrate how these measures capture the intuition of the proximity of an unrelated instance to a uniform one:

$$E_1 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 10 \end{pmatrix} \quad E_2 = \begin{pmatrix} 1 & 6 & 10 \\ 2 & 2 & 3 \\ 6 & 3 & 4 \end{pmatrix}.$$

Both correlations are almost one with E_1 ($\rho_{\text{task}} = \rho_{\text{mach}} = 1$), whereas they are close to zero with E_2 ($\rho_{\text{task}} = -0.02$ and $\rho_{\text{mach}} = 0$) even though the costs are only permuted. The first matrix, E_1, may be transformed to be equivalent to a uniform instance by changing the last cost from the value 10 to 9. However, E_2 requires a lot more changes to be equivalent to such an instance. In these examples, the correlations ρ_{task} and ρ_{mach} succeed in quantifying the proximity to a uniform one.

3.2. Related Scheduling Problems

There are three special cases when either one or both of these correlations are one or zero. When $\rho_{\text{task}} = \rho_{\text{mach}} = 1$, then instances may be uniform ones (see Proposition 1) and the corresponding problem can be equivalent to $Q||C_{\max}$ (see [15] for the $\alpha|\beta|\gamma$ notation) for example. When $\rho_{\text{task}} = 1$ and $\rho_{\text{mach}} = 0$, then a related problem is $Q|p_i = p|C_{\max}$ where each machine may be represented by a cycle time (uniform case) and all tasks are identical (see Proposition 2). Finally, when $\rho_{\text{mach}} = 1$ and $\rho_{\text{task}} = 0$, then a related problem is $P||C_{\max}$ where each task may be represented by a weight and all machines are identical (see Proposition 3). For any other cases, we do not have any relation to another existing model that is more specific than R.

Proposition 1
The task and machine correlations of a cost matrix corresponding to a uniform instance (Q) are $\rho_{\text{task}} = \rho_{\text{mach}} = 1$.

Proof
In an unrelated instance corresponding to a uniform one, $e_{i,j} = w_i b_j$ where w_i is the weight of task i and b_j the cycle time of machine j. The correlation between $\{w_i b_j\}_{1 \leq j \leq m}$ and $\{w_{i'} b_j\}_{1 \leq j \leq m}$ is one for all $(i, i') \in [1; n]^2$ because the second vector is the product of the first by the constant $w_{i'}/w_i$. Therefore, $\rho_{\text{task}} = 1$. Analogously, we also have $\rho_{\text{mach}} = 1$. □

The reciprocal is however not true. Consider the cost matrix $E = \{e_{i,j}\}_{1 \leq i \leq n, 1 \leq j \leq m}$ where $e_{i,j} = r_i + c_j$ and both $\{r_i\}_{1 \leq i \leq n}$ and $\{c_j\}_{1 \leq j \leq m}$ are arbitrary. The task and machine correlations are both one, but there is no corresponding uniform instance in this case. The second generation method proposed in this article generates such instances. However, the first proposed method produces cost matrices which are close to uniform instances when both target correlations are high.

For the second special case, we propose a mechanism to generate a cost matrix that is arbitrarily close to a given uniform instances with identical tasks. Let $w_i = w$ be the weight of any task i. In the related cost matrix, $e_{i,j} = w b_j + u_{i,j}$ where $U = \{u_{i,j}\}_{1 \leq i \leq n, 1 \leq j \leq m}$ is a matrix of random values that follows each a uniform distribution between $-\epsilon$ and ϵ. This cost matrix can be seen as a uniform instance with identical tasks with noise.

DOI: 10.1002/cpe

Proposition 2
The task and machine correlations of a cost matrix $E = \{wb_j + u_{i,j}\}_{1 \leq i \leq n, 1 \leq j \leq m}$ tend to one and zero, respectively, as $\epsilon \to 0$ and $n \to \infty$ while the root-mean-square deviation between E and the closest uniform instance with identical tasks (Q and $w_i = w$) tends to zero.

Proof
We first show that $\rho_{\text{task}} \to 1$ and $\rho_{\text{mach}} \to 0$ as $\epsilon \to 0$. Both the numerator and the denominator in Equation 2 tend to $\frac{1}{m}\sum_{j=1}^{m}(wb_j)^2 - (\frac{1}{m}\sum_{j=1}^{m} wb_j)^2$ as $\epsilon \to 0$. Therefore, the task correlation $\rho_{\text{task}} \to 1$ as $\epsilon \to 0$. The numerator in Equation 4 simplifies as $\frac{1}{n}\sum_{i=1}^{n} u_{i,j}u_{i,j'} - \frac{1}{n^2}\sum_{i=1}^{n} u_{i,j}\sum_{i=1}^{n} u_{i,j'}$, while the denominator simplifies as $\sqrt{\frac{1}{n}\sum_{i=1}^{n} u_{i,j}^2 - \left(\frac{1}{n}\sum_{i=1}^{n} u_{i,j}\right)^2} \times \sqrt{\frac{1}{n}\sum_{i=1}^{n} u_{i,j'}^2 - \left(\frac{1}{n}\sum_{i=1}^{n} u_{i,j'}\right)^2}$. This is the correlation between two columns in the noise matrix. This tends to 0 as $n \to \infty$ if the variance of the noise is non-zero, namely if $\epsilon \neq 0$.

We must now show that the root-mean-square deviation (RMSD) between E and the closest uniform instance with identical tasks tends to zero. The RMSD between E and the instance where w is the weight of the task and b_j the cycle time of machine j is $\sqrt{\frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} u_{i,j}^2}$. This tends to zero as $\epsilon \to 0$. Therefore, the RMSD between E and any closer instance will be lower and will thus also tends to zero as $\epsilon \to 0$. □

Proposition 3
The task and machine correlations of a cost matrix $E = \{w_i b + u_{i,j}\}_{1 \leq i \leq n, 1 \leq j \leq m}$ tend to zero and one, respectively, as $\epsilon \to 0$ and $m \to \infty$ while the root-mean-square deviation between E and the closest identical instance (P) tends to zero.

Proof
The proof is analogous to the proof of Proposition 2. □

In Propositions 2 and 3, ϵ must be non-zero, otherwise the variance of the rows or columns will be null and the corresponding correlation undefined.

Note that when either the task or machine correlation is zero, the correlation between any pair of rows or columns may be different from zero as long as the average of the individual correlations is zero. Thus, there may exist instances with task and machine correlations close to one and zero (or zero and one), respectively, that are arbitrarily far from any uniform instance with identical tasks (or identical instance). However, the two proposed generation methods in this article produce cost matrices with similar correlations for each pair of rows and for each pair of columns. In this context, it is therefore relevant to consider that the last two special cases are related to the previous specific instances.

In contrast to these proposed measures, the heterogeneity measures proposed in [20] quantify the proximity of an unrelated instance with an identical one with identical tasks. Depending on the heterogeneity values, however, two of the special cases are shared: uniform with identical tasks (Q and $w_i = w$) when the task heterogeneity is zero and identical (P) when the machine heterogeneity is zero.

3.3. Correlations of the Range-Based, CVB and Noise-Based Methods

We analyze the asymptotic correlation properties of the range-based, CVB and noise-based methods described in Section 2 and synthesize them in Table I. We discard the shuffling method due to its combinatorial nature that prevents it from being easily analyzed. The range-based and CVB methods use two additional parameters to control the consistency of any generated matrix: a and b are the fractions of the rows and columns from the cost matrix, respectively, that are sorted.

In the following analysis, we refer to convergence in probability simply as convergence for concision. Also, the order in which the convergence applies (either when $n \to \infty$ and then when $m \to \infty$, or the contrary) is not specified and may depend on each result.

The proofs of the analysis of the range-based and CVB methods (Propositions 4 to 7) are in the companion research report [18].

DOI: 10.1002/cpe

Proposition 4
The task correlation ρ_{task} of a cost matrix generated with the range-based method with the parameters a and b converges to a^2b as $n \to \infty$ and $m \to \infty$.

Proposition 5
The machine correlation ρ_{mach} of a cost matrix generated with the range-based method with parameter b converges to $\frac{3}{7}$ as $n \to \infty$, $m \to \infty$, $R_{\text{task}} \to \infty$ and $R_{\text{mach}} \to \infty$ if the matrix is inconsistent and to $b^2 + 2\sqrt{\frac{3}{7}b(1-b) + \frac{3}{7}(1-b)^2}$ as $n \to \infty, m \to \infty, R_{\text{task}} \to \infty$ and $R_{\text{mach}} \to \infty$ if $a = 1$.

Proposition 5 assumes that $R_{\text{task}} \to \infty$ and $R_{\text{mach}} \to \infty$ because the values used in the literature (see Section 3.4) are frequently large. Moreover, this clarifies the presentation (the proof provides a finer analysis of the machine correlation depending on R_{task} and R_{mach}).

Proposition 6
The task correlation ρ_{task} of a cost matrix generated with the CVB method with the parameters a and b converges to a^2b as $n \to \infty$ and $m \to \infty$.

Proposition 7
The machine correlation ρ_{mach} of a cost matrix generated with the CVB method with the parameters V_{task}, V_{mach} and b converges to $\frac{1}{V_{\text{mach}}^2(1+1/V_{\text{task}}^2)+1}$ as $n \to \infty$ and $m \to \infty$ if the matrix is inconsistent and to $b^2 + \frac{2b(1-b)}{\sqrt{V_{\text{mach}}^2(1+1/V_{\text{task}}^2)+1}} + \frac{(1-b)^2}{V_{\text{mach}}^2(1+1/V_{\text{task}}^2)+1}$ as $n \to \infty$ and $m \to \infty$ if $a = 1$.

Proposition 8
The task correlation ρ_{task} of a cost matrix generated using the noise-based method with the parameters V_{mach} and V_{noise} converges to $\frac{1}{V_{\text{noise}}^2(1+1/V_{\text{mach}}^2)+1}$ as $m \to \infty$.

Proof
Let's analyze the four parts of Equation 2 (the two operands of the subtraction in the numerator and the two square roots in the denominator). As $m \to \infty$, the first part of the nominator converges to the expected value of the product of two scalars drawn from a gamma distribution with expected value one and CV V_{task}, the square of b_j that follows a gamma distribution with expected value one and CV V_{mach} and two random variables that follow a gamma distribution with expected value one and CV V_{noise}. This expected value is $1 + V_{\text{mach}}^2$. As $m \to \infty$, the second part of the numerator converges to the product of the expected values of each row, namely one. As $m \to \infty$, each part of the denominator converges to the standard deviation of each row. This is $\sqrt{V_{\text{mach}}^2 V_{\text{noise}}^2 + V_{\text{mach}}^2 + V_{\text{noise}}^2}$ because each row is the product of a scalar drawn from a gamma distribution with expected value one and CV V_{task} and two random variables that follow two gamma distributions with expected value one and CV V_{mach} and V_{noise}. This concludes the proof. □

Proposition 9
The machine correlation ρ_{mach} of a cost matrix generated using the noise-based method with the parameters V_{task} and V_{noise} converges to $\frac{1}{V_{\text{noise}}^2(1+1/V_{\text{task}}^2)+1}$ as $n \to \infty$.

Proof
Due to the symmetry of the noise-based method, the proof is analogous to the proof of Proposition 8. □

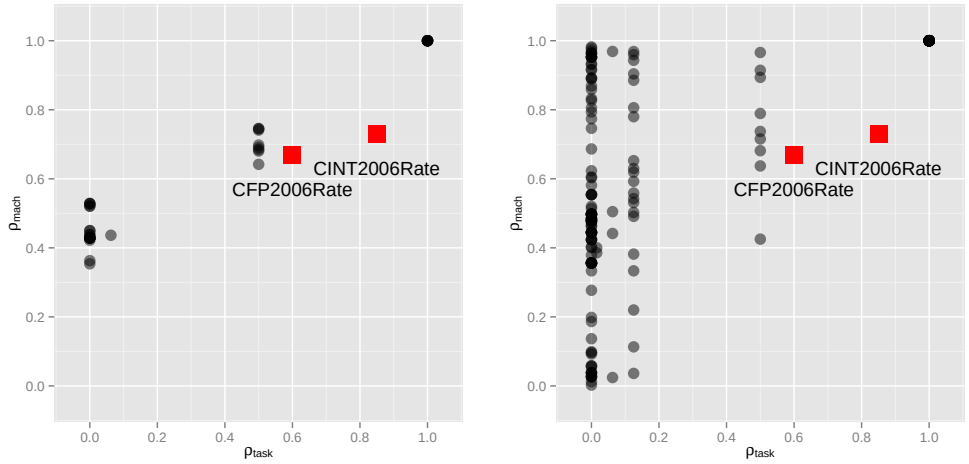
3.4. Correlations in Previous Studies

More than 200 unique settings used for generating instances were collected from the literature and synthesized in [10]. For each of them, we computed the correlations using the formulas from Table I. For the case when $0 < a < 1$, the correlations were measured on a single 1000×1000 cost matrix that was generated with the range-based or the CVB method as done in [10] (missing consistency values were replaced by 0 and the expected value was set to one for the CVB method).

DOI: 10.1002/cpe

Table I. Summary of the asymptotic correlation properties of existing methods (Propositions 4 to 9).

Method	ρ_{task}	ρ_{mach}
Range-based	$a^2 b$	$\begin{cases} \frac{3}{7} & \text{if } a = 0 \\ b^2 + 2\sqrt{\frac{3}{7}} b(1-b) + \frac{3}{7}(1-b)^2 & \text{if } a = 1 \end{cases}$
CVB	$a^2 b$	$\begin{cases} \frac{1}{V_{\text{mach}}^2(1+1/V_{\text{task}}^2)+1} & \text{if } a = 0 \\ b^2 + \frac{2b(1-b)}{\sqrt{V_{\text{mach}}^2(1+1/V_{\text{task}}^2)+1}} + \frac{(1-b)^2}{V_{\text{mach}}^2(1+1/V_{\text{task}}^2)+1} & \text{if } a = 1 \end{cases}$
Noise-based	$\frac{1}{V_{\text{noise}}^2(1+1/V_{\text{mach}}^2)+1}$	$\frac{1}{V_{\text{noise}}^2(1+1/V_{\text{task}}^2)+1}$



(a) Range-based method (b) CVB method

Figure 1. Correlation properties (ρ_{task} and ρ_{mach}) of cost matrices used in the literature (adapted from [1]). The correlations for the SPEC benchmarks belong to an area that is not well covered.

Table II. Summary of the properties for two benchmarks (CINT2006Rate and CFP2006Rate). Both cost matrices are provided in [22].

Benchmark	ρ_{task}	ρ_{mach}	$V\mu_{\text{task}}$	$V\mu_{\text{mach}}$	μV_{task}	μV_{mach}	TDH	MPH	TMA
CINT2006Rate	0.85	0.73	0.32	0.36	0.37	0.39	0.90	0.82	0.07
CFP2006Rate	0.60	0.67	0.42	0.32	0.48	0.39	0.91	0.83	0.13

Figure 1 depicts the values for the proposed correlation measures. The task correlation is larger than the machine correlation (i.e., $\rho_{\text{task}} > \rho_{\text{mach}}$) for only a few instances. The space of possible values for both correlations has thus been largely unexplored. Additionally, few instances have high task correlation and are thus underrepresented. By contrast, the methods proposed below (Algorithms 3 and 4) cover the entire correlation space.

Two matrices extracted from the SPEC benchmarks on five different machines are provided in [22]. There are 12 tasks in CINT2006Rate and 17 tasks in CFP2006Rate. The values for the correlation measures and other measures from the literature are given in Table II. The correlations for these two benchmarks correspond to an area that is not well covered in Figure 1. Hence, instances used in the literature are not representative of these benchmarks and cannot be used to validate scheduling heuristics. This emphasizes the need for a better exploration of the correlation space when assessing scheduling algorithms.

DOI: 10.1002/cpe

4. CONTROLLING THE CORRELATION

Table I shows that the correlation properties of existing methods are determined by a combination of unrelated parameters, which is unsatisfactory. We propose two cost matrix generation methods that take the task and machine correlations as parameters. The methods proposed in this section assume that both these parameters are distinct from one.

4.1. Adaptation of the Noise-Based Method

Algorithm 3: Correlation noise-based generation of cost matrices with gamma distribution for controlling the correlations

Input: n, m, r_{task}, r_{mach}, μ, V
Output: a $n \times m$ cost matrix

1: $N_1 \leftarrow 1 + (r_{\text{task}} - 2r_{\text{task}}r_{\text{mach}} + r_{\text{mach}})V^2 - r_{\text{task}}r_{\text{mach}}$
2: $N_2 \leftarrow (r_{\text{task}} - r_{\text{mach}})^2V^4 + 2(r_{\text{task}}(r_{\text{mach}} - 1)^2 + r_{\text{mach}}(r_{\text{task}} - 1)^2)V^2 + (r_{\text{task}}r_{\text{mach}} - 1)^2$
3: $V_{\text{noise}} \leftarrow \sqrt{\frac{N_1 - \sqrt{N_2}}{2r_{\text{task}}r_{\text{mach}}(V^2+1)}}$
4: $V_{\text{task}} \leftarrow \frac{1}{\sqrt{(1/r_{\text{mach}} - 1)/V_{\text{noise}}^2 - 1}}$
5: $V_{\text{mach}} \leftarrow \frac{1}{\sqrt{(1/r_{\text{task}} - 1)/V_{\text{noise}}^2 - 1}}$
6: **for all** $1 \leq i \leq n$ **do**
7: $\quad w_i \leftarrow G(1/V_{\text{task}}^2, V_{\text{task}}^2)$
8: **end for**
9: **for all** $1 \leq j \leq m$ **do**
10: $\quad b_j \leftarrow G(1/V_{\text{mach}}^2, V_{\text{mach}}^2)$
11: **end for**
12: **for all** $1 \leq i \leq n$ **do**
13: $\quad$ **for all** $1 \leq j \leq m$ **do**
14: $\quad\quad e_{i,j} \leftarrow \mu w_i b_j \times G(1/V_{\text{noise}}^2, V_{\text{noise}}^2)$
15: $\quad$ **end for**
16: **end for**
17: **return** $\{e_{i,j}\}_{1 \leq i \leq n, 1 \leq j \leq m}$

We first adapt the noise-based method by changing its parameters (see Algorithm 3). The objective is to set the parameters V_{task}, V_{mach} and V_{noise} of the original method (Algorithm 2) given the target correlations r_{task} and r_{mach}. Propositions 10 and 11 show that the assignments on Lines 4 and 5 fulfill this objective for any value of V_{noise}. On Lines 7, 10 and 14, $G(k, \theta)$ is the gamma distribution with shape k and scale θ. This distribution generalizes the exponential and Erlang distributions and has been advocated for modeling job runtimes [31, 32].

Proposition 10
The task correlation ρ_{task} of a cost matrix generated using the correlation noise-based method with the parameter r_{task} converges to r_{task} as $m \to \infty$.

Proof
According to Proposition 8, the task correlation ρ_{task} converges to $\frac{1}{V_{\text{noise}}^2(1+1/V_{\text{mach}}^2)+1}$ as $m \to \infty$. When replacing V_{mach} by $\frac{1}{\sqrt{\frac{1}{V_{\text{noise}}^2}\left(\frac{1}{r_{\text{task}}} - 1\right) - 1}}$ (Line 5 of Algorithm 3), this is equal to r_{task}. □

Proposition 11
The machine correlation ρ_{mach} of a cost matrix generated using the correlation noise-based method with the parameter r_{mach} converges to r_{mach} as $n \to \infty$.

DOI: 10.1002/cpe

Proof
Due to the symmetry of the correlation noise-based method, the proof is analogous to the proof of Proposition 10. □

To fix the parameter V_{noise}, we impose a bound on the coefficient of variation of the final costs in the matrix to avoid pathological instances due to extreme variability. This constraint requires the complex computation of V_{noise} on Lines 1 to 3.

Proposition 12
When used with the parameters μ and V, the correlation noise-based method generates costs with expected value μ and coefficient of variation V.

Proof
The expected value and the coefficient of variation of the costs in a matrix generated with the noise-based method are μ and $\sqrt{V_{\text{task}}^2 V_{\text{mach}}^2 V_{\text{noise}}^2 + V_{\text{task}}^2 V_{\text{mach}}^2 + V_{\text{task}}^2 V_{\text{noise}}^2 + V_{\text{mach}}^2 V_{\text{noise}}^2 + V_{\text{task}}^2 + V_{\text{mach}}^2 + V_{\text{noise}}^2}$, respectively [10, Proposition 12]. Replacing V_{task}, V_{mach} and V_{noise} by their definitions on Lines 3 to 5 leads to an expression that simplifies as V. □

Note that the correlation parameters may be zero: if $r_{\text{task}} = 0$ (resp., $r_{\text{mach}} = 0$), then $V_{\text{task}} = 0$ (resp., $V_{\text{mach}} = 0$). However, each of them must be distinct from one. If they are both equal to one, a direct method exists by setting $V_{\text{noise}} = 0$. The distribution of the costs with this method is the product of three gamma distributions as with the original noise-based method.

4.2. Combination-Based Method

Algorithm 4 presents the combination-based method. It sets the correlation between two distinct columns (or rows) by computing a linear combination between a base vector common to all columns (or rows) and a new vector specific to each column (or row). The algorithm first generates the matrix with the target machine correlation using a base column (generated on Line 3) and the linear combination on Line 7. Then, rows are modified such that the task correlation is as desired using a base row (generated on Line 12) and the linear combination on Line 16. The base row follows a distribution with a lower standard deviation, which depends on the machine correlation (Line 10). Using this specific standard deviation is essential to set the target task correlation (see the proof of Proposition 13). Propositions 13 and 14 show these two steps generate a matrix with the target correlations for any value of V_{col}.

Proposition 13
The task correlation ρ_{task} of a cost matrix generated using the combination-based method with the parameter r_{task} converges to r_{task} as $m \to \infty$.

Proof
Let's recall Equation 2 from the definition of the task correlation:

$$\rho^r_{i,i'} \triangleq \frac{\frac{1}{m}\sum_{j=1}^{m} e_{i,j}e_{i',j} - \frac{1}{m}\sum_{j=1}^{m} e_{i,j}\frac{1}{m}\sum_{j=1}^{m} e_{i',j}}{\sqrt{\frac{1}{m}\sum_{j=1}^{m} e_{i,j}^2 - \left(\frac{1}{m}\sum_{j=1}^{m} e_{i,j}\right)^2}\sqrt{\frac{1}{m}\sum_{j=1}^{m} e_{i',j}^2 - \left(\frac{1}{m}\sum_{j=1}^{m} e_{i',j}\right)^2}}$$

Given Lines 7, 16 and 21, any cost is generated as follows:

$$e_{i,j} = \mu\frac{\sqrt{r_{\text{task}}}r_j + \sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}}c_i + \sqrt{1-r_{\text{mach}}}G(1/V_{\text{col}}^2, V_{\text{col}}^2)\right)}{\sqrt{r_{\text{task}}} + \sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}} + \sqrt{1-r_{\text{mach}}}\right)} \tag{5}$$

Let's scale all the costs $e_{i,j}$ by multiplying them by $\frac{1}{\mu}\left(\sqrt{r_{\text{task}}} + \sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}} + \sqrt{1-r_{\text{mach}}}\right)\right)$. This scaling does not change $\rho^r_{i,i'}$. We thus simplify Equation 5 as follows:

$$e_{i,j} = \sqrt{r_{\text{task}}}r_j + \sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}}c_i + \sqrt{1-r_{\text{mach}}}G(1/V_{\text{col}}^2, V_{\text{col}}^2)\right) \tag{6}$$

DOI: 10.1002/cpe

Algorithm 4: Combination-based generation of cost matrices with gamma distribution

Input: n, m, r_{task}, r_{mach}, μ, V
Output: a $n \times m$ cost matrix

$V_{\text{col}} \leftarrow \frac{\sqrt{r_{\text{task}}}+\sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}}+\sqrt{1-r_{\text{mach}}}\right)}{\sqrt{r_{\text{task}}}\sqrt{1-r_{\text{mach}}}+\sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}}+\sqrt{1-r_{\text{mach}}}\right)}V$ {Scale variability}
for all $1 \leq i \leq n$ **do** {Generate base column}
  $c_i \leftarrow G(1/V_{\text{col}}^2, V_{\text{col}}^2)$
end for
for all $1 \leq i \leq n$ **do** {Set the correlation between each pair of columns}
  for all $1 \leq j \leq m$ **do**
    $e_{i,j} \leftarrow \sqrt{r_{\text{mach}}}c_i + \sqrt{1-r_{\text{mach}}} \times G(1/V_{\text{col}}^2, V_{\text{col}}^2)$
  end for
end for
$V_{\text{row}} \leftarrow \sqrt{1-r_{\text{mach}}}V_{\text{col}}$ {Scale variability}
for all $1 \leq j \leq m$ **do** {Generate base row}
  $r_j \leftarrow G(1/V_{\text{row}}^2, V_{\text{row}}^2)$
end for
for all $1 \leq i \leq n$ **do** {Set the correlation between each pair of rows}
  for all $1 \leq j \leq m$ **do**
    $e_{i,j} \leftarrow \sqrt{r_{\text{task}}}r_j + \sqrt{1-r_{\text{task}}}e_{i,j}$
  end for
end for
for all $1 \leq i \leq n$ **do** {Rescaling}
  for all $1 \leq j \leq m$ **do**
    $e_{i,j} \leftarrow \frac{\mu e_{i,j}}{\sqrt{r_{\text{task}}}+\sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}}+\sqrt{1-r_{\text{mach}}}\right)}$
  end for
end for
return $\{e_{i,j}\}_{1\leq i\leq n, 1\leq j\leq m}$

Let's focus on the first part of the numerator of $\rho^r_{i,i'}$:

$$\frac{1}{m}\sum_{j=1}^{m} e_{i,j}e_{i',j} = r_{\text{task}}\frac{1}{m}\sum_{j=1}^{m} r_j^2 \tag{7}$$

$$+ \frac{1}{m}\sum_{j=1}^{m} \sqrt{r_{\text{task}}}r_j\sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}}c_i + \sqrt{1-r_{\text{mach}}}G(1/V_{\text{col}}^2, V_{\text{col}}^2)\right) \tag{8}$$

$$+ \frac{1}{m}\sum_{j=1}^{m} \sqrt{r_{\text{task}}}r_j\sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}}c_{i'} + \sqrt{1-r_{\text{mach}}}G(1/V_{\text{col}}^2, V_{\text{col}}^2)\right) \tag{9}$$

$$+ (1-r_{\text{task}})\frac{1}{m}\sum_{j=1}^{m} \left(\sqrt{r_{\text{mach}}}c_i + \sqrt{1-r_{\text{mach}}}G(1/V_{\text{col}}^2, V_{\text{col}}^2)\right) \times \tag{10}$$

$$\left(\sqrt{r_{\text{mach}}}c_{i'} + \sqrt{1-r_{\text{mach}}}G(1/V_{\text{col}}^2, V_{\text{col}}^2)\right) \tag{11}$$

The first subpart (Equation 7) converges to $r_{\text{task}}(1+(1-r_{\text{mach}})V_{\text{col}}^2)$ as $m \to \infty$ because r_j follows a gamma distribution with expected value one and standard deviation $\sqrt{1-r_{\text{mach}}}V_{\text{col}}$. The second subpart (Equation 8) converges to $\sqrt{r_{\text{task}}}\sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}}c_i + \sqrt{1-r_{\text{mach}}}\right)$ as $m \to \infty$ because the expected value of $G(1/V_{\text{col}}^2, V_{\text{col}}^2)$ is one. The third subpart (Equation 9) converges to $\sqrt{r_{\text{task}}}\sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}}c_{i'} + \sqrt{1-r_{\text{mach}}}\right)$ as $m \to \infty$. Finally, the last subpart (Equations 10 and 11) converges to $(1-r_{\text{task}})\left(\sqrt{r_{\text{mach}}}c_i + \sqrt{1-r_{\text{mach}}}\right)\left(\sqrt{r_{\text{mach}}}c_{i'} + \sqrt{1-r_{\text{mach}}}\right)$ as $m \to \infty$. The second part of the numerator of $\rho^r_{i,i'}$ is simpler and converges to $\left(\sqrt{r_{\text{task}}} + \sqrt{1-r_{\text{task}}}\right.$

DOI: 10.1002/cpe

$\left(\sqrt{r_{\text{mach}}}c_i + \sqrt{1-r_{\text{mach}}}\right)\right)\left(\sqrt{r_{\text{task}}} + \sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}}c_{i'} + \sqrt{1-r_{\text{mach}}}\right)\right)$ as $m \to \infty$. Therefore, the numerator of $\rho^r_{i,i'}$ converges to $r_{\text{task}}(1-r_{\text{mach}})V^2_{\text{col}}$ as $m \to \infty$.

The denominator of $\rho^r_{i,i'}$ converges to the product of the standard deviations of e_{ij} and $e_{i'j}$ as $m \to \infty$. The standard deviation of r_j (resp., $G(1/V^2_{\text{col}}, V^2_{\text{col}})$) is $\sqrt{1-r_{\text{mach}}}V_{\text{col}}$ (resp., V_{col}). Therefore, the standard deviation of e_{ij} is $\sqrt{r_{\text{task}}(1-r_{\text{mach}})V^2_{\text{col}} + (1-r_{\text{task}})(1-r_{\text{mach}})V^2_{\text{col}}}$.

The correlation between any pair of distinct rows $\rho^r_{i,i'}$ converges thus to r_{task} as $m \to \infty$. □

Proposition 14
The machine correlation ρ_{mach} of a cost matrix generated using the combination-based method with the parameter r_{mach} converges to r_{mach} as $n \to \infty$.

Proof
The correlation between any pair of distinct columns $\rho^c_{j,j'}$ is (Equation 4):

$$\rho^c_{j,j'} \triangleq \frac{\frac{1}{n}\sum_{i=1}^{n} e_{i,j}e_{i,j'} - \frac{1}{n}\sum_{i=1}^{n} e_{i,j}\frac{1}{n}\sum_{i=1}^{n} e_{i,j'}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n} e^2_{i,j} - \left(\frac{1}{n}\sum_{i=1}^{n} e_{i,j}\right)^2}\sqrt{\frac{1}{n}\sum_{i=1}^{n} e^2_{i,j'} - \left(\frac{1}{n}\sum_{i=1}^{n} e_{i,j'}\right)^2}}$$

Let's consider the same scaling for the costs $e_{i,j}$ as in Equation 6.
The first part of the numerator of $\rho^c_{j,j'}$ is:

$$\frac{1}{n}\sum_{i=1}^{n} e_{i,j}e_{i,j'} = r_{\text{task}}r_j r_{j'} + 2(1-r_{\text{task}})\frac{1}{n}\sum_{i=1}^{n}\sqrt{r_{\text{mach}}}c_i\sqrt{1-r_{\text{mach}}}G(1/V^2_{\text{col}}, V^2_{\text{col}}) \quad (12)$$

$$+ (1-r_{\text{task}})\frac{1}{n}\sum_{i=1}^{n} r_{\text{mach}}c_i^2 \quad (13)$$

$$+ (1-r_{\text{task}})\frac{1}{n}\sum_{i=1}^{n}(1-r_{\text{mach}})G(1/V^2_{\text{col}}, V^2_{\text{col}})^2 \quad (14)$$

$$+ (r_j + r_{j'})\frac{1}{n}\sum_{i=1}^{n}\sqrt{r_{\text{task}}}\sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}}c_i + \sqrt{1-r_{\text{mach}}}G(1/V^2_{\text{col}}, V^2_{\text{col}})\right) \quad (15)$$

The first subpart (Equation 12) converges to $r_{\text{task}}r_j r_{j'} + 2(1-r_{\text{task}})\sqrt{r_{\text{mach}}}\sqrt{1-r_{\text{mach}}}$ as $n \to \infty$. The second subpart (Equation 13) converges to $(1-r_{\text{task}})r_{\text{mach}}(1+V^2_{\text{col}})$ as $n \to \infty$ because c_i follows a gamma distribution with expected value one and standard deviation V_{col}. The third subpart (Equation 14) converges to $(1-r_{\text{task}})(1-r_{\text{mach}})$ as $n \to \infty$ and the last subpart (Equation 15) converges to $(r_j + r_{j'})\sqrt{r_{\text{task}}}\sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}} + \sqrt{1-r_{\text{mach}}}\right)$ as $n \to \infty$. The second part of the numerator of $\rho^c_{j,j'}$ converges to $\left(\sqrt{r_{\text{task}}}r_j + \sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}} + \sqrt{1-r_{\text{mach}}}\right)\right)\left(\sqrt{r_{\text{task}}}r_{j'} + \sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}} + \sqrt{1-r_{\text{mach}}}\right)\right)$ as $n \to \infty$. Therefore, the numerator of $\rho^c_{j,j'}$ converges to $(1-r_{\text{task}})r_{\text{mach}}V^2_{\text{col}}$ as $n \to \infty$.

The denominator of $\rho^c_{j,j'}$ converges to $(1-r_{\text{task}})\left(r_{\text{mach}}V^2_{\text{col}} + (1-r_{\text{mach}})V^2_{\text{col}}\right)$ as $n \to \infty$ and the correlation between any pair of distinct columns $\rho^c_{j,j'}$ converges thus to r_{mach} as $n \to \infty$. □

Finally, the resulting matrix is scaled on Line 21 to adjust its mean. The initial scaling of the standard deviation on Line 1 is necessary to ensure that the final cost coefficient of variation is V.

Proposition 15
When used with the parameters μ and V, the combination-based method generates costs with expected value μ and coefficient of variation V.

Proof
By replacing with the values of the base row and column on Lines 3 and 12, Equation 5 gives:

$$e_{i,j} = \mu\frac{\sqrt{r_{\text{task}}}G(1/V^2_{\text{row}}, V^2_{\text{row}}) + \sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}}G(1/V^2_{\text{col}}, V^2_{\text{col}}) + \sqrt{1-r_{\text{mach}}}G(1/V^2_{\text{col}}, V^2_{\text{col}})\right)}{\sqrt{r_{\text{task}}} + \sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}} + \sqrt{1-r_{\text{mach}}}\right)}$$

DOI: 10.1002/cpe

The expected value of any cost is thus μ because the expected value of all gamma distributions is one.

The standard deviation of $G(1/V_{\text{col}}^2, V_{\text{col}}^2)$ is V_{col} and the standard deviation of $G(1/V_{\text{row}}^2, V_{\text{row}}^2)$ is $\sqrt{1-r_{\text{mach}}}V_{\text{col}}$. Therefore, the standard deviation of $e_{i,j}$ is:

$$\mu\frac{\sqrt{r_{\text{task}}}\sqrt{1-r_{\text{mach}}}+\sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}}+\sqrt{1-r_{\text{mach}}}\right)}{\sqrt{r_{\text{task}}}+\sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}}+\sqrt{1-r_{\text{mach}}}\right)}V_{\text{col}}$$

Given the assignment on Line 1, this simplifies as μV. The cost coefficient of variation is therefore V. □

As with the correlation noise-based method, the correlation parameters must be distinct from one. Additionally, the final cost distribution is a sum of three gamma distributions (two if either of the correlation parameters is zero and only one if both of them are zero).

Note that the previous propositions give only convergence results. For a given generated matrix with finite dimension, the effective correlation properties are distinct from the asymptotic ones.

5. IMPACT ON SCHEDULING HEURISTICS

Controlling the task and machine correlations provides a continuum of unrelated instances that are arbitrarily close to uniform instances. This section shows how some heuristics for the $R||C_{\max}$ scheduling problem[‖] are affected by this proximity.

5.1. Selected Heuristics

A subset of the heuristics from [20] were used with instances generated using the correlation noise-based and combination-based methods. The three selected heuristics, which are detailed in [18, Appendix E], are based on distinct principles to emphasize how the correlation properties may have different effects on the performance.

First, we selected EFT [34, E-schedule] [35, Min-Min], a common greedy heuristic that computes the completion time of any task on any machine and schedules first the task that finishes the earliest on the corresponding machine. The second heuristic is HLPT [36], an adaptation of LPT [37] for unrelated platforms that is similar to HEFT [38]. HLPT differs from EFT by considering first the largest tasks based on their minimum cost on any machine and assigning it to the machine that finishes it the earliest. Since LPT is an efficient heuristic for the $Q||C_{\max}$ problem, HLPT performs as the original LPT when machines are uniform (i.e., when the correlations are both equal to 1). The last heuristic, BalSuff [36] starts from an initial mapping where the tasks are assigned to their best machines, the ones with their smallest costs. Then, the algorithm iteratively balances the schedule by changing the allocation of the tasks that are on the most loaded machine, i.e. the one that completes its tasks the latest. At each iteration, the algorithm selects a task-machine pair that minimizes the amount by which the task duration increases, its sufferage, and moves the task to the machine provided that the makespan is improved. BalSuff is more sophisticated than the other heuristics but generates excellent solutions.

These heuristics perform identically when the task and machine correlations are arbitrarily close to one and zero, respectively. In particular, sorting the tasks for HLPT is meaningless because all tasks have similar execution times. With such instances, the problem is related to the $Q|p_i = p|C_{\max}$ problem (see Section 3.2), which is polynomial. Therefore, we expect these heuristics to perform well with these instances.

[‖] A well-studied NP-Hard problem [33] in which tasks are independent and the objective is to minimize the total execution time.

DOI: 10.1002/cpe

5.2. Settings

In the following experiments, we rely on the correlation noise-based and combination-based methods (Algorithms 3 and 4) to generate cost matrices. With both methods, instances are generated with $n = 100$ tasks and $m = 30$ machines. Without loss of generality, the cost expected value μ is set to one (scaling a matrix by multiplying each cost by the same constant will have no impact on the scheduling heuristics). Unless otherwise stated, the cost coefficient of variation V is set to 0.3.

For the last two parameters, the task and machine correlations, we use the probit scale. The probit function is the quantile function of the standard normal distribution. It highlights what happens for values that are arbitrarily close to 0 and 1 at the same time. For instance, with 10 equidistant values between 0.01 and 0.9, the first five values are 0.01, 0.04, 0.10, 0.22 and 0.40 (the last five are the complement of these values to one). In the following experiments, the correlations vary from 0.001 to 0.999 using a probit scale.

For each scenario, we compute the makespan** of each heuristic. We then consider the relative difference from the reference makespan: $C/C_{\min} - 1$ where C is the makespan of a given heuristic and $C_{\min}$ the best makespan we obtained. The closer to zero, the better the performance. To compute $C_{\min}$, we use a genetic algorithm that is initialized with all the solutions obtained by other heuristics as in [20], which significantly improves the quality of the generated schedules. Finding the optimal solution would take too much time for this NP-Hard problem. We assume in this study that the reference makespan closely approximates the optimal one.

5.3. Variation of the Correlation Effect

The first experiment shows the impact of the task and machine correlations when the target correlations are the same (see Figure 2). For each generation method and coefficient of variation, 10 000 random instances are generated with varying values for the parameters $r_{\text{task}} = r_{\text{mach}}$ that are uniformly distributed according to a probit scale between 0.001 and 0.999.

In terms of central tendency, we see that the selected heuristics are impacted in different ways when the correlations increase: EFT performance degrades slightly; HLPT performance improves significantly; and, BalSuff performance remains stable except for correlation values above 0.9.

In terms of variance for some given values of correlations, the performance varies moderately. For correlation parameters between 0.01 and 0.1 and a coefficient of variation of 0.3, we generate 1695 instances with the correlation noise-based method. In the case of HLPT, although the average performance stays relatively constant when the correlations vary from 0.01 and 0.1, the relative differences with the best cases were between 0.063 and 0.382. However, the 50% most central of these differences were between 0.148 and 0.200 (see the dark rectangle in Figure 2). Therefore, we may have some confidence in the average performance even though the performance for a single instance may be moderately different from the average one.

5.4. Mean Effect of Task and Machine Correlations

The heat maps on Figures 3 to 5 share the same generation procedure. First, 30 equidistant correlation values are considered between 0.001 and 0.999 using a probit scale (0.001, 0.002, 0.0039, 0.0071, ..., 0.37, 0.46, ..., 0.999). Then, each pair of values for the task and machine correlations leads to the generation of 200 cost matrices (for a total of 180 000 instances). The actual correlations are then measured for each generated cost matrices. Any tile on the figures corresponds to the average performance obtained with the instances for which the actual correlation values lie in the range of the tile. Hence, an instance generated with 0.001 for both correlations may be associated with another tile than the bottommost and leftmost one depending on its actual correlations. Although it did not occur in our analysis, values outside any tile were planned to be discarded.

Figure 3 compares the average performance of EFT, HLPT and BalSuff. The diagonal line corresponds to the cases when both correlations are similar. In these cases, the impact of the

**The makespan is the total execution time and it must be minimized.

DOI: 10.1002/cpe

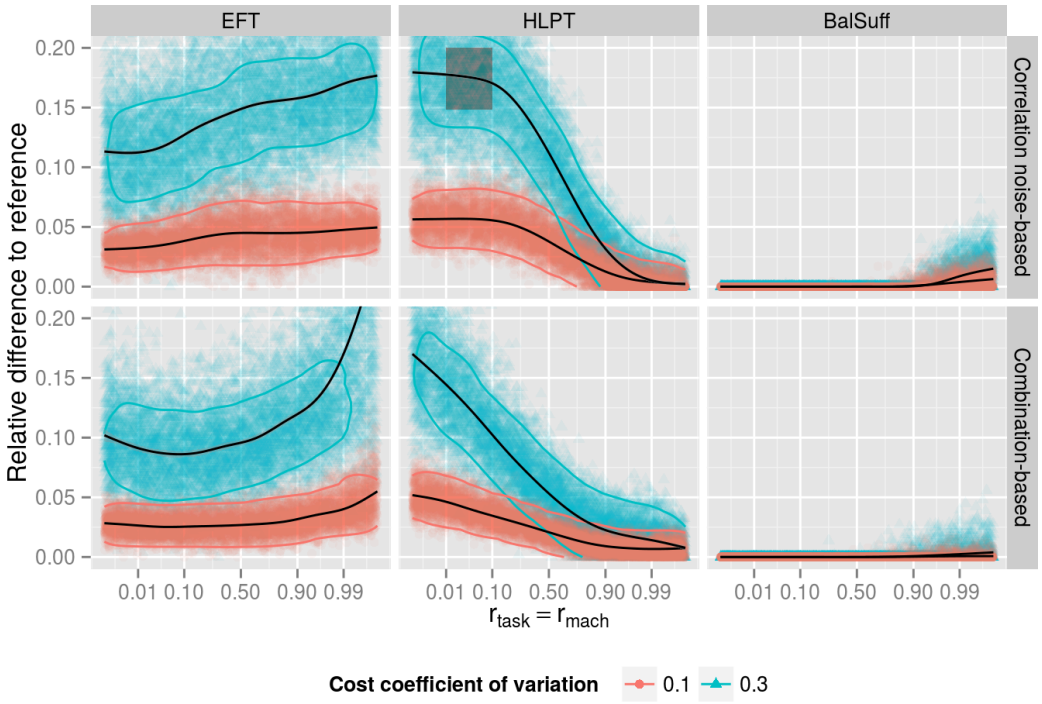


Figure 2. Heuristic performance with 10 000 instances for each pair of generation method and coefficient of variation. The x-axis is in probit scale between 0.001 and 0.999. The central tendency is obtained with a smoothing method relying on the generalized additive model (GAM). The contour lines correspond to the areas with the highest density of points. The dark rectangle corresponds to 50% of the most central values when $0.01 \leq r_{\text{task}} \leq 0.1$ and $V = 0.3$.

correlations on the three heuristics is consistent with the previous observations that are drawn from Figure 2 (see Section 5.3). Despite ignoring the variability, Figure 3 is more informative regarding the central tendency because both correlations vary.

First, EFT performance remains mainly unaffected by the task and machine correlations when they are similar. However, its performance is significantly impacted by them when one correlation is the complement of the other to one (i.e., when $\rho_{\text{task}} = 1 - \rho_{\text{mach}}$, which is the other diagonal). In this case, the performance of EFT is at its poorest on the top-left. It then continuously improves until reaching its best performance on the bottom-right (less than 5% from the reference makespan, which is comparable to the other two heuristics for this area). This is consistent with the previous observation that this last area corresponds to instances that may be close to $Q|p_i = p|C_{\max}$ instances, for which EFT is optimal (see Section 5.1). HLPT achieves the best performance when either correlation is close to one. This is particularly true for the task correlation. HLPT shows however some difficulties when both correlations are close to zero. This tendency was already clearly depicted on Figure 2. Finally, BalSuff closely follows the reference makespan. The iterative nature of this algorithm, which makes it more costly than the other two, allows the generation of high-quality schedules.

5.5. *Effect of the Cost Coefficient of Variation*

Figure 4 shows the effect of the cost coefficient of variation, V, on HLPT performance for five distinct values: 0.1, 0.2, 0.3, 0.5 and 1. All costs are similar when the coefficient of variation is 0.1 (0.90, 0.94, 0.95, 1.07 and 1.14 for instance), whereas they are highly heterogeneous when it is 1 (0.1, 0.2, 0.7, 1.5 and 2.5 for instance).

DOI: 10.1002/cpe

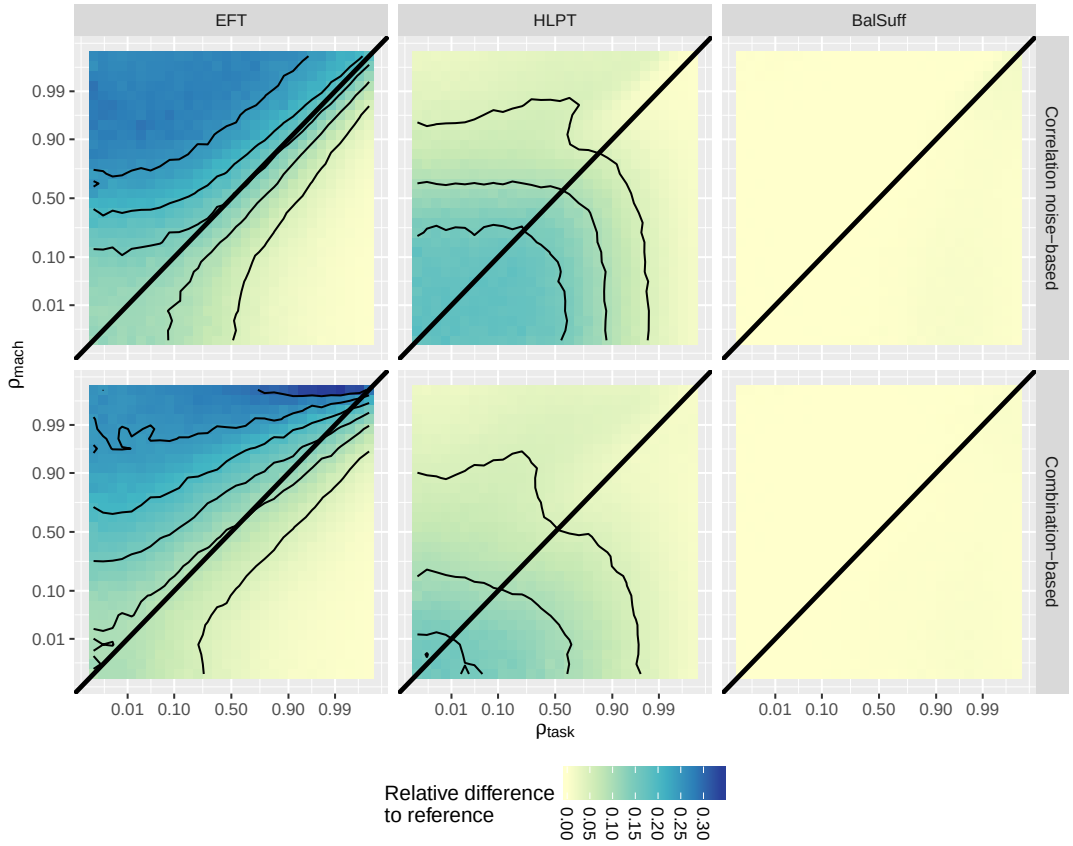


Figure 3. Heuristic performance with 180 000 instances for each generation method. The cost coefficient of variation V is set to 0.3. The x- and y-axes are in probit scale between 0.001 and 0.999. Each tile represents on average 200 instances. The contour lines correspond to the levels in the legend (0, 0.05, 0.1, ...). The diagonal slices correspond to Figure 2.

The behavior of HLPT is similar for most values of V with both generation methods: it performs the worst in the bottom-left area than in the other areas. However, V limits the magnitude of this difference. In particular, the performance of HLPT remains almost the same when $V = 0.1$.

HLPT behaves slightly differently when $V = 1$. At this heterogeneity level, incorrect scheduling decisions may have significant consequences on the performance. Here, HLPT performs the worst for instances for which the task correlation is non-zero and the machine correlation is close to 0. By contrast, it produces near-optimal schedules in the area related to instances of the $P||C_{\max}$ problem (top-left). For these instances, LPT, from which HLPT is inspired, achieves an upper bound of $4/3$, which may explain its efficiency.

5.6. Best Heuristic

Figure 5 depicts the results for the last set of experiments. In addition to the three selected heuristics, two other heuristics were considered: BalEFT [36], which is similar to BalSuff except it selects at each iteration the task that minimizes its earliest finish time, and Max-min [36], which is similar to EFT except it schedules first the task with the largest minimum completion time. Each tile color corresponds to the best heuristic in average over related instances. When the performance of any other heuristic is closer to the best one than 0.001, then this heuristic is considered similar. For instance, if the best heuristic performance is 0.05, then all heuristics with a performance lower than 0.051 are considered similar to the best one. Tiles for which there are at least two similar heuristics (the best one and at least another one) are darker. For instance, this is the case for low task

DOI: 10.1002/cpe

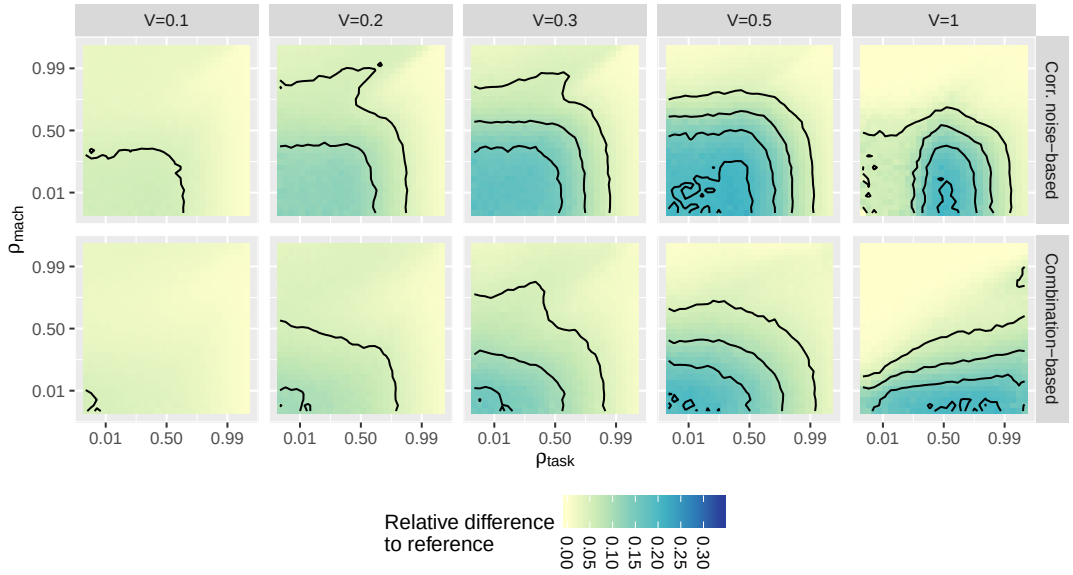


Figure 4. Performance of HLPT with 180 000 instances for each pair of generation method and cost coefficient of variation V. The x- and y-axes are in probit scale between 0.001 and 0.999. Each tile represents on average 200 instances. The contour lines correspond to the levels in the legend (0, 0.05, 0.1, ...).

correlation, high machine correlation and $V = 1$ for which HLPT and Max-min are similar (note that Max-min is never the only heuristic to be the best). The white contour lines show the areas for which there are at least three similar heuristics. When several heuristics are similar for a given tile, the appearing heuristic is the one that is the best the least often (in particular, heuristics are chosen in the reverse order in which they appear in the legend of Figure 5). This highlights the settings for which the worst heuristics are good.

When the cost coefficient of variation is 0.1 or 0.3, the best heuristics are BalSuff and BalEFT. This is expected because they are the most sophisticated and the most costly ones. When $V = 0.1$, BalSuff outperforms BalEFT except for high task and low machine correlations or low task and high machine correlations. In addition, with high task and low machine correlations all tested heuristics behave similarly. The related problem is polynomial and all tested heuristics are optimal for this problem. When $V = 0.3$, BalEFT outperforms BalSuff only for high task and low machine correlations with both generation methods. The case when $V = 1$ is significantly different. BalSuff is almost always the best when the machine correlation is low. For low task and high machine correlations, there are at least two best methods, including HLPT which is the best method when the machine correlation is high. The superiority of HLPT over both BalSuff and BalEFT in this case confirms the results previously pointed out on Figure 4. This behavior, identified by varying the correlations, was not observed when varying the heterogeneity of the costs in [20] and thus illustrates the interest of this new measure when assessing scheduling algorithms.

On both Figures 4 and 5, the behavior of the heuristic performance remains relatively stable except when the cost coefficient of variation is high. The precise impact of large values of V remains to be investigated.

To conclude on the performance of EFT, HLPT and BalSuff: EFT and HLPT perform well in the bottom-right area, which may be because they are optimal for the problem related to this area ($Q|p_i = p|C_{\max}$); HLPT performs also well in this top-left area, which may be because it achieves an upper bound of $4/3$ for the problem related to this area ($P||C_{\max}$); BalSuff performs well everywhere thanks to its costlier approach that balances iteratively the tasks.

The results obtained with both generation methods are not equivalent because for the same correlation values, the generated instances must have different properties depending on the

DOI: 10.1002/cpe

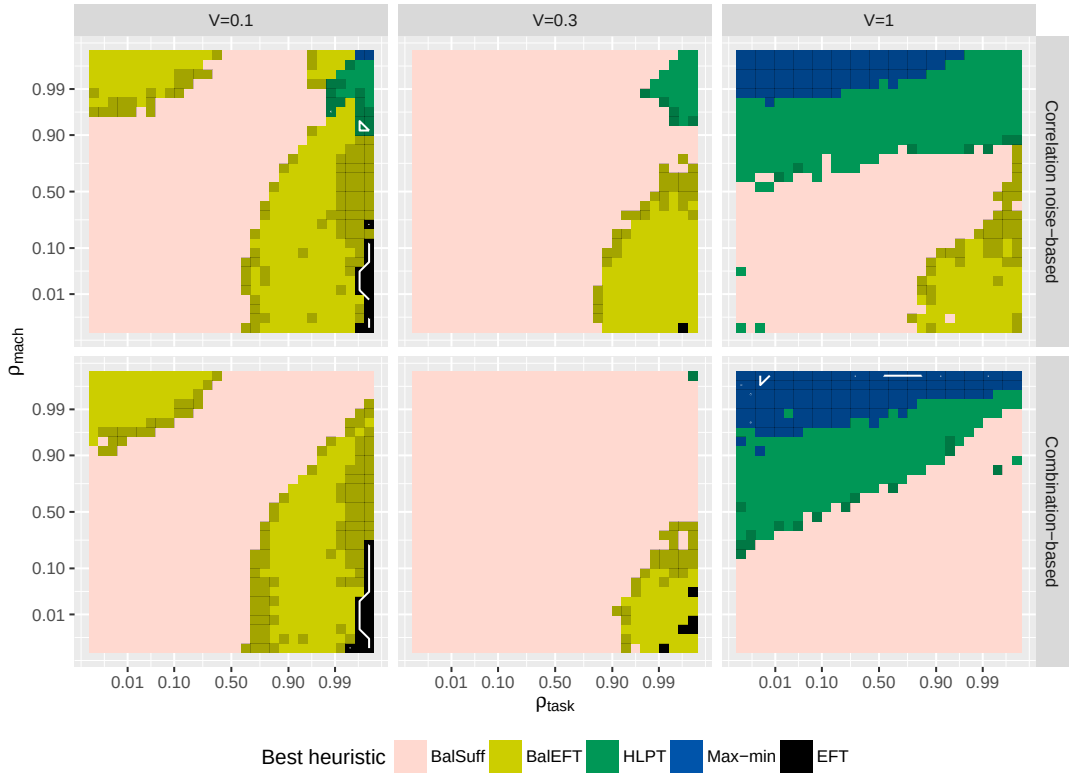


Figure 5. Heuristic with the best average performance with 180 000 instances for each pair of generation method and cost coefficient of variation V (among EFT, HLPT, BalSuff, BalEFT and Max-min). The x- and y-axes are in probit scale between 0.001 and 0.999. Each tile represents on average 200 instances. Tiles with at least two similar heuristics are darker (i.e., there is at least one heuristic with a performance closer to the best heuristic performance than 0.001). Contour lines show the tiles with at least three similar heuristics.

generation process. However, most of the observations in this section are consistent. In particular, the task and machine correlations impact the performance of the heuristics similarly with both generation methods. This shows that controlling this properties when generating cost matrices plays an crucial role. Finally, these two methods should be considered and used as tools to assess the quality of scheduling algorithms and using both will give a better view on the algorithm characteristics and performance considering correlation.

6. CONCLUSION

This article studies the correlations of cost matrices used to assess heterogeneous scheduling algorithms. The task and machine correlations are proposed to measure the similarity between an unrelated instance in which any cost is arbitrary (R) and the closest uniform instance (Q) in which any cost is proportional to the task weight and machine cycle time. We analyzed several generation methods from the literature and designed two new ones to see the impact of these properties. In contrast to instances used in previous studies, the new methods can be used to cover the entire space of possible correlation values (including realistic ones).

Even though the correlation is not a perfect measure for the distance between uniform and unrelated instances (a unitary correlation does not always imply a correspondence to a uniform instance), both proposed generation methods consistently show how some heuristics from the literature are affected. For instance, the closer instances are from the uniform case, the better HLPT,

DOI: 10.1002/cpe

an adaptation of LPT to the unrelated case, performs. Additionally, the need for two correlations (for the tasks and for the machines) arises for EFT for which the performance goes from worst to best as the task and machine correlations go from zero to one and one to zero, respectively. These effects do not depend on the generation method. This shows that both these correlations could enhance a hyperheuristic mechanism that would select a heuristic based on the properties of the instance.

Although the current study highlights the importance of controlling the correlations in cost matrices, it presents some limitations. Overcoming each of them is left for future work. First, results were obtained using the gamma distribution only. However, the two proposed methods could use other distributions as long as the expected value and standard deviation are preserved. Second, all formal derivations are in the asymptotic case only. Hence, the proposed approach must be adjusted for small instances. Also, the proposed correlation measures and generation methods assume that the correlations stay the same for each pair of rows and for each pair of columns: our measures average the correlations and our methods are inapplicable when the correlations between each pair of rows or each pair of columns are distinct. Considering two correlation matrices that define the specific correlations between each pair of rows and each pair of columns would require the design of a finer generation method. Finally, investigating the relation with the heterogeneous properties would require the design of a method that controls both the correlation and heterogeneity properties. A sensitivity analysis could then be used to assess the impact of each of these properties.

ACKNOWLEDGEMENT

We sincerely thank the reviewers for their careful reading and detailed comments. We would like to also thank Stéphane Chrétien and Nicolas Gast for their helpful comments. Computations have been performed on the supercomputer facilities of the Mésocentre de calcul de Franche-Comté.

REFERENCES

1. Canon LC, Héam PC, Philippe L. Controlling and Assessing Correlations of Cost Matrices in Heterogeneous Scheduling. *Euro-Par*, 2016; 133–145.
2. Leung JYT ((ed.)). *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*. Chapman & Hall/CCR, 2004.
3. Maheswaran M, Ali S, Siegel HJ, Hensgen D, Freund RF. Dynamic mapping of a class of independent tasks onto heterogeneous computing systems. *Journal of Parallel and Distributed Computing* 1999; **59**(2):107–131.
4. Luo P, Lü K, Shi Z. A revisit of fast greedy heuristics for mapping a class of independent tasks onto heterogeneous computing systems. *Journal of Parallel and Distributed Computing* 2007; **67**(6):695–714.
5. Munir EU, Li JZ, Shi SF, Zou ZN, Yang DH. MaxStd: A task scheduling heuristic for heterogeneous computing environment. *Information Technology Journal* 2008; **7**(4):679–683.
6. Kołodziej J, Xhafa F. Enhancing the genetic-based scheduling in computational grids by a structured hierarchical population. *Future Generation Computer Systems* 2011; **27**(8):1035–1046.
7. Diaz CO, Pecero JE, Bouvry P. Scalable, low complexity, and fast greedy scheduling heuristics for highly heterogeneous distributed computing systems. *The Journal of Supercomputing* 2014; **67**(3):837–853.
8. Stodden V, Leisch F, Peng RD. *Implementing reproducible research*. CRC Press, 2014.
9. Ali S, Siegel HJ, Maheswaran M, Hensgen D, Ali S. Representing task and machine heterogeneities for heterogeneous computing systems. *Tamkang Journal of Science and Engineering* 2000; **3**(3):195–208.
10. Canon LC, Philippe L. On the heterogeneity bias of cost matrices for assessing scheduling algorithms. *IEEE Transactions on Parallel and Distributed Systems* 2016; doi:10.1109/TPDS.2016.2629503.
11. Agullo E, Beaumont O, Eyraud-Dubois L, Herrmann J, Kumar S, Marchal L, Thibault S. Bridging the gap between performance and bounds of cholesky factorization on heterogeneous platforms. *Parallel and Distributed Processing Symposium Workshop (IPDPSW), 2015 IEEE International*, IEEE, 2015; 34–45.
12. Thain D, Tannenbaum T, Livny M. Distributed computing in practice: the condor experience. *Concurrency and Computation: Practice and Experience* 2005; **17**(2-4):323–356, doi:10.1002/cpe.938. URL http://dx.doi.org/10.1002/cpe.938.
13. Caruana G, Li M, Qi M, Khan M, Rana O. gsched: a resource aware hadoop scheduler for heterogeneous cloud computing environments. *Concurrency and Computation: Practice and Experience* 2016; :n/a–n/adoi: 10.1002/cpe.3841. URL http://dx.doi.org/10.1002/cpe.3841, cPE-15-0439.R3.
14. Anderson DP. Boinc: A system for public-resource computing and storage. *5th International Workshop on Grid Computing (GRID)*, 2004; 4–10.
15. Graham RL, Lawler EL, Lenstra JK, Kan AHGR. Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey. *Annals of Discrete Mathematics* 1979; **5**:287–326.
16. Al-Qawasmeh AM, Maciejewski AA, Wang H, Smith J, Siegel HJ, Potter J. Statistical measures for quantifying task and machine heterogeneities. *The Journal of Supercomputing* 2011; **57**(1):34–50.

17. Canon LC, Héam PC, Philippe L. Controlling and Assessing Correlations of Cost Matrices in Heterogeneous Scheduling Feb 2016, doi:http://dx.doi.org/10.6084/m9.figshare.2858638.v2.
18. Canon LC, Héam PC, Philippe L. Controlling and Assessing Correlations of Cost Matrices in Heterogeneous Scheduling. *Technical Report RR-FEMTO-ST-1191*, FEMTO-ST Feb 2016.
19. Ali S, Siegel HJ, Maheswaran M, Hensgen D. Task execution time modeling for heterogeneous computing systems. *Heterogeneous Computing Workshop (HCW)*, IEEE, 2000; 185–199.
20. Canon LC, Philippe L. On the Heterogeneity Bias of Cost Matrices when Assessing Scheduling Algorithms. *Euro-Par*, 2015; 109–121.
21. Al-Qawasmeh AM, Maciejewski AA, Siegel HJ. Characterizing heterogeneous computing environments using singular value decomposition. *International Parallel & Distributed Processing Symposium Workshops and Phd Forum (IPDPSW)*, IEEE, 2010; 1–9.
22. Al-Qawasmeh AM, Maciejewski AA, Roberts RG, Siegel HJ. Characterizing task-machine affinity in heterogeneous computing environments. *International Parallel & Distributed Processing Symposium Workshops and Phd Forum (IPDPSW)*, IEEE, 2011; 34–44.
23. Khemka B, Friese R, Pasricha S, Maciejewski AA, Siegel HJ, Koenig GA, Powers S, Hilton M, Rambharos R, Poole S. Utility maximizing dynamic resource management in an oversubscribed energy-constrained heterogeneous computing system. *Sustainable Computing: Informatics and Systems* 2014; **5**:14–30.
24. Al-Qawasmeh AM, Pasricha S, Maciejewski AA, Siegel HJ. Power and Thermal-Aware Workload Allocation in Heterogeneous Data Centers. *Transactions on Computers* 2013; **64**(2):477–491.
25. Scheuer EM, Stoller DS. On the Generation of Normal Random Vectors. *Technometrics* 1962; **4**(2):278–281.
26. Cario MC, Nelson BL. Modeling and generating random vectors with arbitrary marginal distributions and correlation matrix. *Technical Report*, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois 1997.
27. Ghosh S, Henderson SG. Behavior of the NORTA method for correlated random vector generation as the dimension increases. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 2003; **13**(3):276–294.
28. Lewandowski D, Kurowicka D, Joe H. Generating random correlation matrices based on vines and extended onion method. *Journal of Multivariate Analysis* 2009; **100**(9):1989–2001.
29. Yang IT. Simulation-based estimation for correlated cost elements. *International Journal of Project Management* 2005; **23**(4):275–282.
30. Macke JH, Berens P, Ecker AS, Tolias AS, Bethge M. Generating spike trains with specified correlation coefficients. *Neural Computation* 2009; **21**(2):397–423.
31. Lublin U, Feitelson DG. The workload on parallel supercomputers: modeling the characteristics of rigid jobs. *J. Parallel Distrib. Comp.* 2003; **63**(11):1105–1122.
32. Feitelson D. Workload modeling for computer systems performance evaluation. *Book Draft, Version 1.0.1* 2014; :1–601.
33. Gary MR, Johnson DS. Computers and intractability: A guide to the theory of np-completeness 1979.
34. Ibarra OH, Kim CE. Heuristic Algorithms for Scheduling Independent Tasks on Nonidentical Processors. *Journal of the ACM* Apr 1977; **24**(2):280–289.
35. Freund RF, Gherrity M, Ambrosius S, Campbell M, Halderman M, Hensgen D, Keith E, Kidd T, Kussow M, Lima JD, *et al.*. Scheduling resources in multi-user, heterogeneous, computing environments with SmartNet. *Heterogeneous Computing Workshop (HCW)*, IEEE, 1998; 184–199.
36. Canon LC, Philippe L. On the Heterogeneity Bias of Cost Matrices when Assessing Scheduling Algorithms. *Technical Report RR-FEMTO-ST-8663*, FEMTO-ST Mar 2015.
37. Graham RL. Bounds on Multiprocessing Timing Anomalies. *Journal of Applied Mathematics* 1969; **17**(2):416–429.
38. Topcuoglu H, Hariri S, Wu My. Performance-effective and low-complexity task scheduling for heterogeneous computing. *IEEE transactions on parallel and distributed systems* 2002; **13**(3):260–274.