Controlling and Assessing Correlations of Cost Matrices in Heterogeneous Scheduling

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

Controlling and Assessing Correlations of Cost Matrices in Heterogeneous Scheduling

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

FEMTO-ST Institute / CNRS – Université de Franche-Comté / UBFC
25000 Besançon, France
{louis-claude.canon,laurent.philippe,pierre-cyrille.heam}@univ-fcomte.fr

Abstract. This paper 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 is a proximity measure with uniform instances[1]. Cost matrices generated with a novel generation method 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.

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. Considering static scheduling, the problem is deterministic, although complex, because all the data are known a priori. In the case of independent tasks running on a heterogeneous platform and with the objective of minimizing the total execution time [13, 14], the performance[2] of any scheduling algorithm depends on the properties of the input cost matrix and generating input instances is thus a crucial problem in algorithm assessment [5, 7]. In a previous study [8], we have proposed heterogeneity measures and procedures to control this property when generating cost matrices. In particular, we showed that the heterogeneity was previously not properly controlled despite having a significant impact on the relative performance of scheduling heuristics. However, the proposed measures prevent tuning how the machines are related to one another in terms of processing time, i.e., if the execution times are proportional and depend on a task weight and a machine cycle time.

[1] Uniform instances are particular unrelated instances in which each execution time is proportional to the weight of the task and the cycle time of the machine.

[2] The performance of any algorithm for this NP-Hard problem is given by the difference between the obtained total execution time and the minimum one.

In this paper, we propose to investigate a continuum of instances between the uniform case and the unrelated case. The contribution[3] is a measure, the correlation, to explore this continuum, its analysis in existing generation methods and existing studies (Section 3), a new generation method with better correlation properties (Section 4) and its analysis on several heuristics (Section 5) and, last, the confrontation of the correlation to a related measure (Section 6).

2 Related Work

The validation of scheduling heuristics in the literature relies mainly on two generation methods: the range-based and CVB methods. The range-based method [4, 5] 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 is based on the same principle except it uses more generic parameters and a distinct underlying distribution. In particular, the parameters consist of two CV[4] (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:[5] the rows in a submatrix containing a fraction a of the initial rows and a fraction b of the initial columns are sorted.

The shuffling and noise-based methods were later proposed in [7, 8]. 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 relies on a similar principle: it inserts noise in each cost by multiplying it by a random variable with mean one. Both methods require the parameters V_{task} and V_{mach} to set the task and machine heterogeneity. In addition, the amount of noise introduced in the noise-based method can be adjusted through the parameter V_{noise}.

This study focuses on the average correlation between each pair of tasks or machines in a cost matrix. No existing work explicitly considers this property. 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. 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.

[3] These results are also available in the companion research report [6].

[4] The Coefficient of Variation is the ratio of the standard deviation to the mean.

[5] In a consistent cost matrix, any task faster than another task on a given machine will be consistently faster than this other task on any machine.

The TMA (Task-Machine Affinity) quantifies the specialization of a platform [1, 2], i.e., whether some machines are particularly efficient for some specific tasks. This measure proceeds 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 [2] and on both the rows and columns in [1]. 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 [12] 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. See Section 6 for a more thorough discussion on the TMA.

The range-based and CVB methods do not cover the entire range of possible values for the TMA [2]. Khemka et al [12] propose a method that iteratively increases the TMA of an existing matrix while keeping the same MPH and TDH. A method that generates matrices with varying affinities (similar to the TMA) and which resembles the noise-based method is also proposed in [3]. However, no formal method has been proposed for generating matrices with a given TMA.

3 Correlation Between Tasks and Processors

As stated previously, the unrelated model is more general than the uniform model 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.

Correlations 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 ignored. Also, since the correlation is symmetric ($\rho^r_{i,i'} = \rho^r_{i',i}$), it is actually sufficient to only compute half of them. We define the *machine correlation*, ρ_{mach}, analogously on the columns. 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.

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 and the problem can be equivalent to $Q||C_{\max}$ [6, Proposition 1]. 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 and all tasks are identical [6, 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 [6, Proposition 3]. For any other cases, we do not have any relation to another existing problem that is more specific than scheduling unrelated instances.

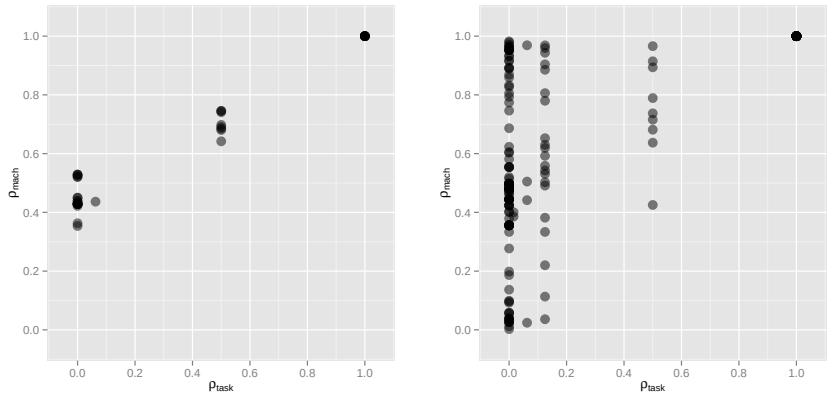
Correlations of Existing Methods Table 1 synthesises the analysis of the asymptotic correlation properties of the range-based, CVB and noise-based methods [6, Propositions 4 to 9].

Table 1. Summary of the asymptotic correlation properties of existing methods.

Method	ρ_{task}	ρ_{mach}
Range-based [4, 5]	$a^2 b$	$\begin{cases} \frac{3}{7} & \text{if } a = 0 \\ b^2 + 2\sqrt{\frac{3}{7}}b(1-b) + \frac{3}{7}(1-b)^2 & \text{if } a = 1 \end{cases}$
CVB [4, 5]	$a^2 b$	$\begin{cases} \frac{1}{V^2_{\text{mach}}(1+1/V^2_{\text{task}})+1} & \text{if } a = 0 \\ b^2 + \frac{2b(1-b)}{\sqrt{V^2_{\text{mach}}(1+1/V^2_{\text{task}})+1}} + \frac{(1-b)^2}{V^2_{\text{mach}}(1+1/V^2_{\text{task}})+1} & \text{if } a = 1 \end{cases}$
Noise-based [8]	$\frac{1}{V^2_{\text{noise}}(1+1/V^2_{\text{mach}})+1}$	$\frac{1}{V^2_{\text{noise}}(1+1/V^2_{\text{task}})+1}$

Correlations in Previous Studies More than 200 unique settings used for generating instances were collected from the literature and synthesized in [8]. For each of them, we computed the correlations using the formulas from Table 1. 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 [8] (missing consistency values were replaced by 0 and the expected value was set to one for the CVB method).

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



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

Fig. 1. Correlation properties (ρ_{task} and ρ_{mach}) of cost matrices used in the literature.

largely unexplored. Additionally, few instances have high task correlation and are thus underrepresented.

Two matrices extracted from the SPEC benchmarks on five different machines are provided in [1]. 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 2. The correlations for these two benchmarks correspond to an area that is not well covered in Figure 1. This illustrate the need for a better exploration of the correlation space when assessing scheduling algorithms.

Table 2. Summary of the properties for two benchmarks (CINT2006Rate and CFP2006Rate).

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

4 Controlling the Correlation

Table 1 shows that the correlation properties of existing methods are determined by a combination of unrelated parameters, which is unsatisfactory. We propose a cost matrix generation method that takes the task and machine correlations as parameters. This method assumes that both these parameters are distinct from one.

Algorithm 1 Combination-based cost matrix generation with gamma distribution

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

1: $V_{\text{col}} \leftarrow \frac{\sqrt{r_{\text{task}}}+\sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}}+\sqrt{1-r_{\text{mach}}}\right)}{\sqrt{r_{\text{task}}}\sqrt{1-r_{\text{mach}}}+\sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}}+\sqrt{1-r_{\text{mach}}}\right)} V$ {Scale variability}
2: **for all** $1 \leq i \leq n$ **do** {Generate base column}
3: $\quad c_i \leftarrow G(1/V_{\text{col}}^2, V_{\text{col}}^2)$
4: **end for**
5: **for all** $1 \leq i \leq n$ **do** {Set the correlation between each pair of columns}
6: $\quad$ **for all** $1 \leq j \leq m$ **do**
7: $\quad\quad e_{i,j} \leftarrow \sqrt{r_{\text{mach}}} c_i + \sqrt{1 - r_{\text{mach}}} \times G(1/V_{\text{col}}^2, V_{\text{col}}^2)$
8: $\quad$ **end for**
9: **end for**
10: $V_{\text{row}} \leftarrow \sqrt{1 - r_{\text{mach}}} V_{\text{col}}$ {Scale variability}
11: **for all** $1 \leq j \leq m$ **do** {Generate base row}
12: $\quad r_j \leftarrow G(1/V_{\text{row}}^2, V_{\text{row}}^2)$
13: **end for**
14: **for all** $1 \leq i \leq n$ **do** {Set the correlation between each pair of rows}
15: $\quad$ **for all** $1 \leq j \leq m$ **do**
16: $\quad\quad e_{i,j} \leftarrow \sqrt{r_{\text{task}}} r_j + \sqrt{1 - r_{\text{task}}} e_{i,j}$
17: $\quad$ **end for**
18: **end for**
19: **for all** $1 \leq i \leq n$ **do** {Rescaling}
20: $\quad$ **for all** $1 \leq j \leq m$ **do**
21: $\quad\quad e_{i,j} \leftarrow \frac{\mu e_{i,j}}{\sqrt{r_{\text{task}}}+\sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}}+\sqrt{1-r_{\text{mach}}}\right)}$
22: $\quad$ **end for**
23: **end for**
24: **return** $\{e_{i,j}\}_{1 \leq i \leq n, 1 \leq j \leq m}$

Algorithm 1 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 task correlation (see the proof of Proposition 1). Propositions 1 and 2 show these two steps generate a matrix with the target correlations for any value of V_{col}.

Proposition 1. *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. Given Lines 7, 16 and 21, any cost, multiplied by $\frac{1}{\mu}\left(\sqrt{r_{\text{task}}}+\sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}}+\sqrt{1-r_{\text{mach}}}\right)\right)$ as it does not change $\rho^r_{i,i'}$, is: $e_{i,j}=\sqrt{r_{\text{task}}}r_j+\sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}}c_i+\sqrt{1-r_{\text{mach}}}G(1/V_{\text{col}}^2,V_{\text{col}}^2)\right)$

Let's focus on the first part of the numerator of $\rho^r_{i,i'}$ (from Equation 2): $\frac{1}{m}\sum_{j=1}^m e_{i,j}e_{i',j} = r_{\text{task}}^2\frac{1}{m}\sum_{j=1}^m r_j^2 + \frac{1}{m}\sum_{j=1}^m\sqrt{r_{\text{task}}}r_j\sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}}c_i+\sqrt{1-r_{\text{mach}}}G(1/V_{\text{col}}^2,V_{\text{col}}^2)\right)+\frac{1}{m}\sum_{j=1}^m\sqrt{r_{\text{task}}}r_j\sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}}c_{i'}+\sqrt{1-r_{\text{mach}}}G(1/V_{\text{col}}^2,V_{\text{col}}^2)\right)+(1-r_{\text{task}})\frac{1}{m}\sum_{j=1}^m\left(\sqrt{r_{\text{mach}}}c_i+\sqrt{1-r_{\text{mach}}}G(1/V_{\text{col}}^2,V_{\text{col}}^2)\right)\times\left(\sqrt{r_{\text{mach}}}c_{i'}+\sqrt{1-r_{\text{mach}}}G(1/V_{\text{col}}^2,V_{\text{col}}^2)\right)$.

The first sum converges to $r_{\text{task}}(1+(1-r_{\max})V_{\text{col}}^2)$ as $m\to\infty$ because r_j follows a gamma distribution with expected value one and standard deviation $\sqrt{1-r_{\max}}V_{\text{col}}$. The second sum 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 $m\to\infty$ and the third sum 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 sum 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 $\left(\sqrt{r_{\text{task}}}+\sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}}c_i+\sqrt{1-r_{\text{mach}}}\right)\right)\left(\sqrt{r_{\text{task}}}+\sqrt{1-r_{\text{task}}}\left(\sqrt{r_{\text{mach}}}c_{i'}+\sqrt{1-r_{\text{mach}}}\right)\right)$ as $m\to\infty$. Therefore, the numerator of $\rho^r_{i,i'}$ converges to $r_{\text{task}}(1-r_{\max})V_{\text{col}}^2$ 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., $G(1/V_{\text{col}}^2,V_{\text{col}}^2)$) is $\sqrt{1-r_{\text{mach}}}V_{\text{col}}$ (resp., V_{col}). Therefore, the standard deviation of e_{ij} is $\sqrt{r_{\text{task}}(1-r_{\text{mach}})V_{\text{col}}^2+(1-r_{\text{task}})(1-r_{\text{mach}})V_{\text{col}}^2}$.

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

Proposition 2. *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 $n\to\infty$.*

The proof of Proposition 2 is similar to the proof of Proposition 1 [6, Proposition 14].

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 CV (Coefficient of Variation) of the costs is V. The proof of Proposition 3 is more direct than the previous ones [6, Proposition 15].

Proposition 3. *When used with the parameters μ and V, the combination-based method generates costs with expected value μ and CV V.*

Note that the correlation parameters may be zero. However, each of them must be distinct from one. If they are both equal to one, a direct method exists by building the unrelated instance corresponding to a uniform instance. 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 scheduling unrelated instances are affected by this proximity.

A subset of the heuristics from [7] were used with instances generated using the combination-based method. The three selected heuristics are based on distinct principles to emphasize how the correlation properties may have different effects on the performance. First, we selected EFT [11, E-schedule] [9, Min-Min], which relies on a greedy principle that schedules first the tasks that have the smallest duration. The second heuristic is an adaptation of LPT [10] for unrelated platforms. Since LPT is a 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). HLPT differs from EFT by considering first the largest tasks instead of the smallest ones based on their minimum cost on any machine. The last heuristic is BalSuff [8], which iteratively balances an initial schedule by changing the allocation of the tasks that are on the most loaded machine. The new machine that will execute it is chosen such as to minimize the increase in the task duration.

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), which is polynomial. Therefore, we expect these heuristics to perform well with these instances.

In the following experiments, we rely on the combination-based method (Algorithm 1) to generate cost matrices. Instances are generated with $n = 100$ tasks and $m = 30$ machines. Without loss of generality, the mean cost μ is one (scaling a matrix by multiplying each cost by the same constant will have no impact on the scheduling heuristics). The cost CV is $V = 0.3$.

For each scenario, we compute the makespan[6] 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 (we use a genetic algorithm that is initialized with all the solutions obtained by other heuristics as in [7] because the problem is NP-Complete and finding the optimal solution would take too much time). The closer to zero, the better the performance. We assume in this study that the reference makespan closely approximates the optimal one.

The heat maps on Figure 2 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). 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. 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

[6] The makespan is the total execution time and it must be minimized.

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 assigned to another tile than the bottommost and leftmost one depending on its actual correlations. Any value outside any tile was discarded when it occurred.

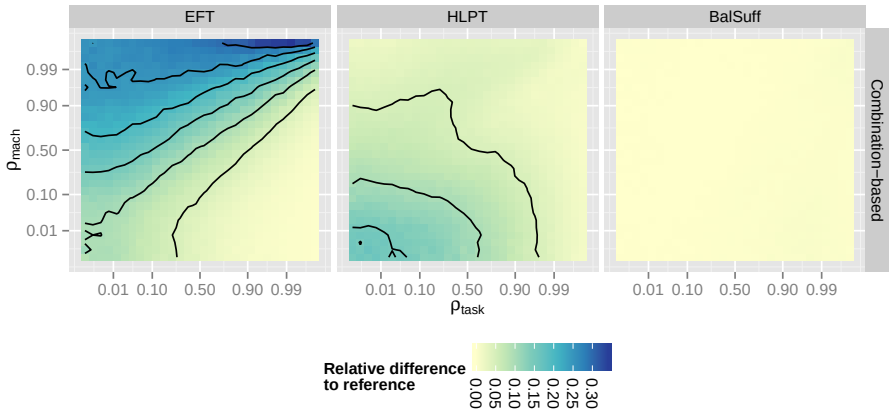


Fig. 2. Heuristic performance with 180 000 instances for the combination-based method. The cost CV V is set to 0.3. The x- and y-axes are in probit scale between 0.001 and 0.999. Each tile represents on average 200 instances. The contour lines correspond to the levels in the legend (0, 0.05, 0.1, ...).

Figure 2 compares the average performance of EFT, HLPT and Balsuff. 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. HLPT achieves the best performance when either correlation is close to one. This is particularly true in the case of the task correlation. HLPT shows however some difficulties when both correlations are close to zero. Finally, BalSuff closely follows the reference makespan except when the task correlation reaches values above 0.5. This is surprising because we could expect any heuristic to have its best performance in the bottom-right part as for EFT. Despite having good performance in this area, this is not the case with BalSuff.

6 Relation to TMA

The TMA is a measure based on the singular values of the normalized inverse cost matrix. We consider the variant in which the normalization is done alternatively on both the rows and columns [1]. The cost matrix is first inverted before being normalized with an iterative procedure. Finally, the result corresponds to the inverse of the ratio between the first singular value and the mean of the other singular values.

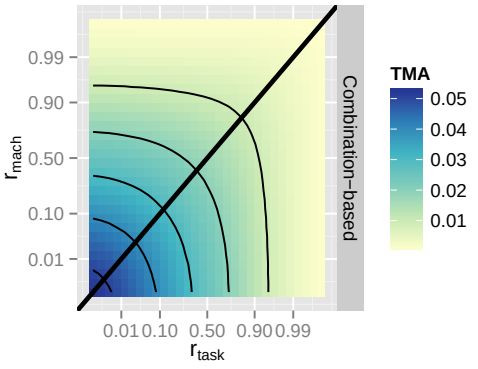


Fig. 3. TMA of the instances used in Figure 2.

Similarly to the correlation, the TMA measures the affinities between the tasks and the machines. TMA values equal to zero means machines are uniform (no affinity) because only the first singular value is non-zero and the rank of the cost matrix is one. Oppositely, TMA values equal to one means tasks and machines have unrelated characteristics (high affinities between tasks and machines) because the cost matrix is orthogonal.

However, the correspondence with the correlation is not systematic. Let $\{e_{i,j}\}_{1\leq i\leq n, 1\leq j\leq n}$ be a cost matrix where $e_{i,j} = \epsilon$ if $i = j$ and $e_{i,j} = w_i b_j$ otherwise (with w_i the weight of task i and b_j the cycle time of machine j). The TMA of this cost matrix converges to one as $\epsilon \to 0$, which suggests a discrepancy from any uniform instance. By contrast, both its task and machine correlations converge to one as $n \to \infty$ and $m \to \infty$ (suggesting a similarity with a uniform instance). Assuming the number of tasks is greater than the number of machines (i.e., $n > m$), each task i must be scheduled on machine i for $1 \leq i \leq m$. The problem is thus equivalent to scheduling the last $n - m$ tasks, each of which has a well-defined weight. This cost matrix corresponds therefore to a uniform instance as indicated by the correlation properties. This contrived example shows that changing a few single values may impact the TMA more profoundly than the correlations. We conclude that the correlations focus on the general consistency across multiple tasks and machines, whereas the TMA stresses the specialization of a few machines for some specific tasks.

Figure 3 depicts the TMA of each of the $2 \times 30^2 \times 200$ instances generated in Section 5. The TMA is strongly associated with the correlations in our settings. Note that it does not reach large values given that its maximum is one, even when the correlations are close to zero.

The TMA is also symmetric relatively to the diagonal slices: it is the same when the task/machine correlations are high/low as when they are low/high. Therefore, some behaviors may not be seen with the TMA. For instance, EFT performance varies mainly relatively to the other diagonal (from the top-left to the bottom-right).

The TMA offers several advantages: its normalization procedure makes it independent from the heterogeneity and like the correlation, it is associated to the performance of the selected heuristics. However, it suffers from several drawbacks. Its value depends on the cost matrix dimension and on the cost CV. Moreover, its normalization procedure makes derivations of analytical results difficult. By contrast, the correlation has no such default but it is not independent from the heterogeneity. Also, the correlation is finer because it consists of two different values, which allow the characterization of behaviors that cannot be seen with the TMA (e.g., for EFT). Nevertheless, the TMA may be more relevant than the correlation in some specific cases. For instance, with small cost matrices, the TMA is more sensitive to individual values that may impact significantly the performance. Devising a SVD-based measure that outperforms the TMA (analytically simpler and independent from the cost matrix dimension and the cost CV) is left for future work.

7 Conclusion

This paper 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 ($R||C_{\max}$) and the closest uniform instance ($Q||C_{\max}$) in which any cost is proportional to the task weight and machine cycle time. We analyzed several generation methods from the literature and designed a new one to see the impact of these properties.

Even though the correlation approximates the distance between uniform and unrelated instances (a unitary correlation does not imply it corresponds to a uniform instance), our proposed generation method shows how some heuristics from the literature are affected. For instance, the closer instances are from the uniform case, the better HLPT, an adaptation of LPT to the unrelated case, performs. Additionally, the need for two correlations (for the tasks and for the machines) arise 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.

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 proposed method could use other distributions as long as the mean and standard deviation are preserved. Second, all formal derivations are in the asymptotic case only. Hence, the proposed results may be less relevant for small instances. Also, the proposed correlation measures and generation method 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 method is 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.

Acknowledgments

Computations have been performed on the supercomputer facilities of the Mésocentre de calcul de Franche-Comté.

References

1. Al-Qawasmeh, A.M., Maciejewski, A.A., Roberts, R.G., Siegel, H.J.: Characterizing task-machine affinity in heterogeneous computing environments. In: IPDPSW (2011)
2. Al-Qawasmeh, A.M., Maciejewski, A.A., Siegel, H.J.: Characterizing heterogeneous computing environments using singular value decomposition. In: IPDPSW (2010)
3. Al-Qawasmeh, A.M., Pasricha, S., Maciejewski, A.A., Siegel, H.J.: Power and Thermal-Aware Workload Allocation in Heterogeneous Data Centers. Transactions on Computers 64(2), 477–491 (2013)
4. Ali, S., Siegel, H.J., Maheswaran, M., Hensgen, D.: Task execution time modeling for heterogeneous computing systems. In: HCW. pp. 185–199. IEEE (2000)
5. Ali, S., Siegel, H.J., Maheswaran, M., Hensgen, D., Ali, S.: Representing task and machine heterogeneities for heterogeneous computing systems. Tamkang Journal of Science and Engineering 3(3), 195–208 (2000)
6. Canon, L.C., Héam, P.C., Philippe, L.: Controlling and Assessing Correlations of Cost Matrices in Heterogeneous Scheduling. Tech. Rep. RR-FEMTO-ST-1191, FEMTO-ST (Feb 2016)
7. Canon, L.C., Philippe, L.: On the Heterogeneity Bias of Cost Matrices when Assessing Scheduling Algorithms. In: Euro-Par. pp. 109–121 (2015)
8. Canon, L.C., Philippe, L.: On the Heterogeneity Bias of Cost Matrices when Assessing Scheduling Algorithms. Tech. Rep. RR-FEMTO-ST-8663, FEMTO-ST (Mar 2015)
9. Freund, R.F., Gherrity, M., Ambrosius, S., Campbell, M., Halderman, M., Hensgen, D., Keith, E., Kidd, T., Kussow, M., Lima, J.D., Mirabile, F., Moore, L., Rust, B., Siegel, H.J.: Scheduling resources in multi-user, heterogeneous, computing environments with SmartNet. In: HCW. pp. 184–199. IEEE (1998)
10. Graham, R.L.: Bounds on Multiprocessing Timing Anomalies. Journal of Applied Mathematics 17(2), 416–429 (1969)
11. Ibarra, O.H., Kim, C.E.: Heuristic Algorithms for Scheduling Independent Tasks on Nonidentical Processors. Journal of the ACM 24(2), 280–289 (Apr 1977)
12. Khemka, B., Friese, R., Pasricha, S., Maciejewski, A.A., Siegel, H.J., Koenig, G.A., 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 5, 14–30 (2014)
13. 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 67(6), 695–714 (2007)
14. Maheswaran, M., Ali, S., Siegel, H.J., Hensgen, D., Freund, R.F.: Dynamic mapping of a class of independent tasks onto heterogeneous computing systems. Journal of Parallel and Distributed Computing 59(2), 107–131 (1999)