On the Heterogeneity Bias of Cost Matrices when Assessing Scheduling Algorithms

Louis-Claude Canon, Laurent Philippe

On the Heterogeneity Bias of Cost Matrices when Assessing Scheduling Algorithms

Louis-Claude Canon
Laurent Philippe
FEMTO-ST / CNRS – Université de Franche-Comté / UBFC
25000 Besançon, France
{louis-claude.canon,laurent.philippe}@univ-fcomte.fr

March 12, 2020

Abstract

Assessing the performance of scheduling heuristics through simulation requires to generate synthetic instances of tasks and machines with well-identified properties. Carefully controlling these properties is mandatory to avoid any bias. We consider the scheduling problem consisting of allocating independent sequential tasks on unrelated processors while minimizing the maximum execution time. In this problem, the instance is a cost matrix that specifies the execution cost of any task on any machine. This paper proposes a measure for quantifying the heterogeneity properties of a cost matrix. An analysis of two classical methods used in the literature reveals a bias in previous studies. A new method is proposed to generate instances with given heterogeneity properties and it is shown that they have a significant impact on several heuristics.

1 Introduction

Leveraging the parallelism of multi-core distributed platforms involves to efficiently schedule applications on several machines [19]. Current studies on performance evaluation can be divided into several categories: formal analysis, experiments, simulations, etc. In the case of simulations, a scheduling strategy is tested in a virtual environment with a given workload. Synthetic instances of workload allow a more general evaluation than specific traces. They are particularly useful for sensitivity analysis [21], which consists in assessing the impact of the instance properties on the algorithms. The lack of control on the instance properties, however, makes it difficult to confront the results of independent studies. For instance, although many papers have compared several scheduling heuristics [9, 10, 13, 20], predicting their performance is still an issue. These problems can be tackled by carefully controlling the instance properties.

We consider the scheduling problem noted $R||C_{\max}$ in $\alpha|\beta|\gamma$ notation [17]. It consists in scheduling n independent sequential tasks on m unrelated machines to minimize the latest task completion time. All tasks are available simultaneously and preemption is not possible. The instance is a *cost matrix* where each element $e_{i,j} \in \mathbb{N}$ is the execution cost of task i on machine j.

This paper provides the following contributions[1]: a statistical description of the use of the range-based and CVB methods in the literature (Section 3); a study of how to quantify the heterogeneity properties of a cost matrix (Section 4); a formal analysis of the range-based and CVB methods and the identification of a bias that impacts several studies (Section 4); a new method with control over heterogeneity properties (Section 5); and, an assessment[2] of the impact of these properties on several heuristics (Section 6).

2 Related Work

The concept of heterogeneity was first introduced in the context of cost matrix by Armstrong [8]. He described the *heterogeneity quadrant* in which cost matrices are divided into four categories depending on

[1] These results are also available in the companion research report [11].

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

their heterogeneity properties regarding tasks and processors: low/low, low/high, high/low, and high/high. However, no method for generating such matrices was proposed.

The range-based and CVB methods were proposed to fill this gap in [5] and then in [6, 7]. However, task and machine heterogeneities were not formally defined and analyzed. The methods were assumed to generate matrices with the expected properties and only validated through some examples.

The limits of these methods were later acknowledged in [4], which proposed to consider the average coefficient of variation[3], skewness and kurtosis of the costs for each task and for each machine. The proposed scheme (based on decision trees) uses these additional information to predict heuristic performance. Despite a wide experimentation plan, the study lacks discussion and interpretation on the relative importance of the considered measures. Additionally, no formal analysis was provided. The exhibited decision trees suggest that the average coefficient of variation plays a significant role, which supports the current work.

The MPH (Machine Performance Homogeneity) is introduced in [3] for capturing the heterogeneity between the machines, while its counterpart for the tasks, the TDH (Task Difficulty Homogeneity), appears in [2]. We discuss them more extensively in Section 4. In addition, the TMA (Task-Machine Affinity) is also defined in [3]: it quantifies the specialisation of the system (i.e., whether some machines are particularly efficient for some specific tasks). Although the three measures are applied to a real benchmark, no method is proposed for generating matrices with given MPH, TDH and TMA.

Friese et al [14] present a method for adding tasks in a given cost matrix while preserving some statistical properties on each column (mean, coefficient of variation, skewness and kurtosis). It ignores the properties on each row however.

A method for generating matrices with varying affinity (similar to the TMA) is proposed in [1]. Khemka et al [18] propose a method for changing the TMA of an existing matrix while keeping the same MPH and TDS. TMA is mentioned to be related to the correlation. Investigating the correlation properties is left for future work. There is also another field of studies dedicated to the generation of matrices with given correlation and covariance matrices [15].

3 Matrix Generations Methods

3.1 Range-based and CVB methods

The most used methods for generating cost matrices are the range-based and the CVB (Coefficient of Variation Based) methods [5–7].

The range-based method generates n vectors of m values that follow a uniform distribution in the range $[1, R_{mach}]$. Each line is then multiplied by a random value that follows a uniform distribution in the range $[1, R_{task}]$.

The CVB method is based on the same principle except it uses parameters that are distinct from the underlying distribution parameters. In particular, it requires two coefficients of variation (V_{task} for the tasks and V_{mach} for the machines) and one mean (μ_{task} for the tasks). The random values follow a gamma distribution whose parameters are computed such that the provided CV (Coefficient of Variation) and mean are respected.

Proposition 1. *When used with parameters V_{task}, V_{mach} and μ_{task}, the CVB method generates costs with expected value μ_{task} and coefficient of variation $\sqrt{V_{task}^2 V_{mach}^2 + V_{task}^2 + V_{mach}^2}$.*

Proof. Each cost is the product of a random variable that follows a gamma law with mean μ_{task} and CV V_{task} and a random variable that follows a gamma law with mean 1 and CV V_{mach}. Therefore, the expected value of the costs is the product of the expected values of both distributions, namely μ_{task}.

The standard deviation of the product of two random variables with means μ_1 and μ_2, and standard deviations σ_1 and σ_2 is $\sqrt{\sigma_1^2\sigma_2^2 + \mu_1^2\sigma_2^2 + \sigma_1^2\mu_2^2}$. With a similar argument as for the expected value we can derive the CV of the costs. □ □

To obtain the CV of the costs with the range-based method, we can replace V_{task} by the CV of the first uniform law, $\frac{\sqrt{12}}{6}\frac{R_{task}-1}{R_{task}+1}$, and V_{mach} by the CV of the second uniform law, $\frac{\sqrt{12}}{6}\frac{R_{mach}-1}{R_{mach}+1}$. This CV remains close to a constant except for low values of R_{task} and R_{mach}. For instance, it is around 0.86

[3] Ratio of the standard deviation to the mean.

when $R_{task} = R_{mach} = 100$ and the asymptotic value is $\frac{\sqrt{7}}{2} \approx 0.88$ when both R_{task} and R_{mach} are large. This is not well-suited to control the heterogeneity of the resulting cost matrix. Also, the asymmetry of this method may lead to different heterogeneity properties for the tasks and for the machines.

3.2 Consistency Extension

Both the previous methods produce cost matrices that may not be representative of realistic settings. For instance, the costs of a given task is not correlated to the costs of another task, which may often be the case in practice. The consistency extension consists in reordering the costs in the generated matrix to have an instance that is closer to the uniform case. Specifically, the rows of a submatrix of an rows and bm columns are sorted. Thus, a machine that is faster for a given task than another machine will likely be also faster for another task. Inconsistent matrices have $a = b = 0$ while consistent matrices have $a = b = 1$ (other matrices are either called semiconsistent or partially consistent).

3.3 Usage in the Literature

We covered the English articles that cite at least one of the references in which the methods were initially presented [5–7] and that were freely available. For each reference, we extracted all the distinct sets of parameters. However, the size was ignored because we only consider asymptotic properties (see Section 4.2).

Some data were not specifically provided. The parameters that could be directly inferred from the article or from similar works are mostly related to missing parameters for the consistency extension (the ones from the cited article were taken). Otherwise, they are treated as missing values. Some articles lack enough information, which prevented any parameter extraction.

On the 160 analysed articles, 78 provide exploitable information on the cost matrix instances. The rest consists of 40 articles with no description, but which refer to instances described in other articles and 42 articles with unclear descriptions or approaches that do not fit the current study. The extracted data are available in [11, Appendix B] and summarized below. While most articles fail to precisely describe the used method, only the range and CV parameters are crucial for reproducing similar instances. In the end, 342 sets of parameters were extracted in 78 articles for a total of 210 unique settings: 37 for the range-based method and 173 for the CVB one.

Figure 1 depicts the values used with both methods. Although there is no clear agreement on which precise parameters are the most relevant, there are some common tendencies. Values for low heterogeneity are usually 10 and 100 for the range-based method and .1, .25 and .3 for the CVB method. Values for high heterogeneity are usually 100, 1e3, 3e3 and 1e5 for the range-based method and .3, .35, .4, .5, .6, .7, .9, 1 and 2 for the CVB method.

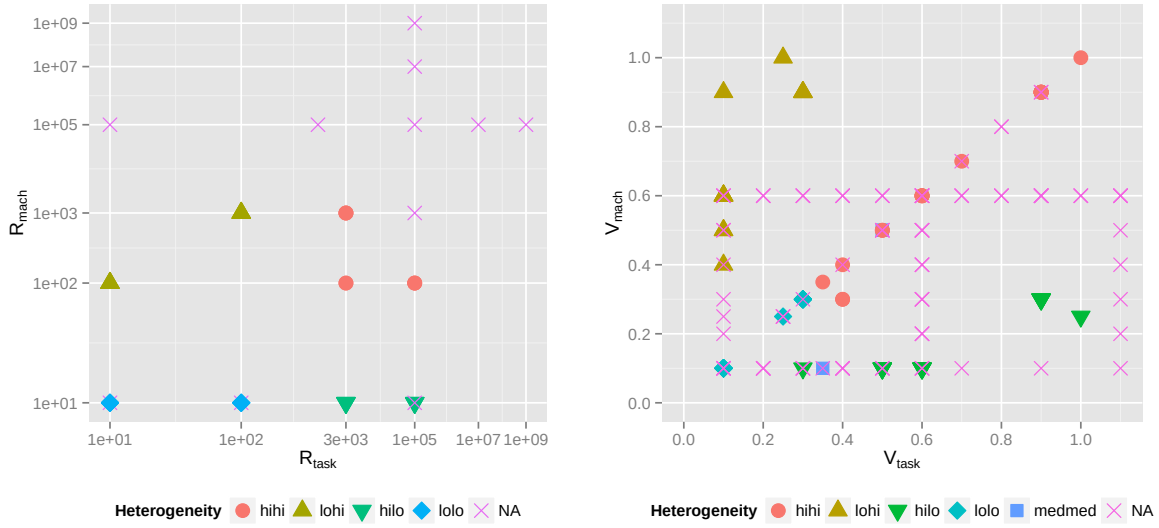
4 Heterogeneity Measures

Assessing the impact of heterogeneity on heuristic performance requires a method for quantifying the heterogeneity of the generated cost matrices.

4.1 TDH and MPH

The closest related measures are the TDH (Task Difficulty Homogeneity) and the MPH (Machine Performance Homogeneity) [2, 3]. The TDH computation consists in computing the difficulty of each task (noted $TD[i]$), sorting all the $TD[i]$ in ascending order and averaging all the ratios between successive $TD[i]$. The measure lies in the interval $(0, 1]$: if it is one, then tasks are all similar; if it is close to zero, then the task heterogeneity is large. The MPH computation is analogous, but for the machine.

These measures have two major shortcomings. First, they are not intuitive (they require to invert costs, to order sums and to average ratios). Also, they do not rely on classical statistical measures, which makes deriving formal results more difficult. Another notable problem is that the resulting values depend on the size of the matrix. In particular, it is close to one when the matrix is large (even if it is generated with the same parameters and has, intuitively, the same characteristics). For instance, if we consider only one machine, the following matrices (cost vectors in this case) have the same TDH: $[1, 2]$



(a) Range-based method

(b) CVB method

Figure 1: Parameters used in the literature. Three points are not shown for the CVB method: $(1.4, 0.4)$, $(1.8, 0.4)$ and $(0.1, 2)$.

and $[0.125, 0.25, 0.5, 1, 2, 4]$. The second vector, however, seems more heterogeneous. As another example, let the minimum TD be 1 and the maximum TD be 100. The TDH is always greater than 0.60 when there are 10 tasks and it is always greater than 0.95 when there are 100 tasks [11, Proposition 1]. This measure is thus relevant only for comparing small cost matrices with similar sizes.

4.2 Intuitive Measures of Heterogeneity

Assuming that the mean of each row represents a task weight, the task heterogeneity may be defined as the CV (Coefficient of Variation) of the means of the rows (noted $V\mu_{task}$). Analogously, the machine heterogeneity may be measured as the CV of the means of the columns (noted $V\mu_{mach}$).

These measures of task and machine heterogeneity has been criticized for small instances [2]. It is argued that the MPH is better than the CV as it is less sensible to outliers. However, we consider asymptotic properties for large matrices in this work because we expect them to hold for small instances. Moreover, in the case of outliers, the CV can be replaced by the quartile coefficient of dispersion, which is a similar standard statistical measure but is more difficult to formally analyse. Finally, the decision trees in [4] suggest that varying this measure has an impact on the heuristics performance and is thus significant.

4.3 Coherence with the Uniform Model

The previous measures do not only rely on intuition, they are also consistent with the expectation when we consider the uniform model. In this model, the cost of executing a task i on a machine j is given by the product of the task weight, w_i, and the inverse of the machine speed, b_j. The concept of task and machine heterogeneity is easy to grasp in the uniform model: it is given by the statistical dispersion of the weights and the speeds, respectively. We assume that the CV of the weights, noted CV_{task}, is a relevant measure of the task heterogeneity. Analogously, the CV of the speeds, noted CV_{mach}, represents the machine heterogeneity.

It is possible to convert an instance of the uniform model in the unrelated model because this last model is more general. The cost matrix is generated by combining both vectors $\{w_i\}_{1 \leq i \leq n}$ and $\{b_j\}_{1 \leq j \leq m}$ such that $e_{i,j} = w_i b_j$. As we know the heterogeneity properties of a uniform instance, we expect our proposed measures for the unrelated model to be consistent when applied on the converted instance, which is indeed the case [11, Proposition 2].

4.4 Heterogeneity of the Range-Based and CVB Methods

We analyse the asymptotic heterogeneity properties of the CVB method with the proposed measures depending on the parameters V_{task} and V_{mach}. An estimator T converges to θ when the expected value of T tends to θ as the number of samples (n and m in our case) tends to ∞.

Proposition 2. *The measure $V\mu_{task}$ of a cost matrix generated using the CVB method with the parameters V_{task} and V_{mach} converges to V_{task} as $n \to \infty$ and $m \to \infty$.*

Proof. This proof assumes that the mean of a set of n samples (called the sample mean) of a random variable with mean μ and standard deviation σ is a random variable with mean μ and standard deviation $\frac{\sigma}{\sqrt{n}}$. Moreover, the CV of a set of n samples (called the sample CV) of a random variable with CV V converges to V as $n \to \infty$.

Let μ_i be the sample mean of the costs on line i. This row is the product of a random variable that follows a distribution with mean μ_{task} and CV V_{task} and m values that follow a distribution with mean one and CV V_{mach}. μ_i is thus also the product of the first random variable and the sample mean of the other m values, which follows a random variable with mean one and CV $\frac{V_{mach}}{\sqrt{m}}$. Therefore, the mean of μ_i is μ_{task} and its CV is $\sqrt{V_{task}^2 \frac{V_{mach}^2}{m} + \frac{V_{mach}^2}{m} + V_{task}^2}$, which tends to V_{task} as $m \to \infty$. Then, the sample CV of all μ_i tends to V_{task} as $n \to \infty$ and $m \to \infty$. □ □

We can also show that $V\mu_{mach}$ converges to $a\sqrt{b}V_{mach}$ as n and $m \to \infty$. Although more technical, the proof is analogous and provided in [11, Proposition 6].

These formal results can be extended to the range-based method by replacing V_{task} by the CV of the first random variable ($\frac{\sqrt{12}}{6}\frac{R_{task}-1}{R_{task}+1}$) and V_{mach} by the CV of the second one ($\frac{\sqrt{12}}{6}\frac{R_{mach}-1}{R_{mach}+1}$). Indeed, the proofs only use the mean and the CV of the random underlying distributions.

In the case of complete consistency (i.e., when $a = b = 1$), $V\mu_{task} = V_{task}$ and $V\mu_{mach} = V_{mach}$, which supports the proposed heterogeneity measures. This special case is due to the fact that consistent cost matrices are closer to uniform instances than inconsistent ones.

The main issue of the CVB method is related to the impact of the consistency parameters on the heterogeneity properties. It biases comparisons of scheduling methods when cost matrices are used with different consistency settings because these matrices will also have different heterogeneity properties. The range-based method presents an even stronger bias as both V_{task} and V_{mach} tends to $\frac{\sqrt{12}}{6}$ as R_{task} and $R_{mach} \to \infty$ (the heterogeneity properties are thus often similar).

4.5 Task and Machine Heterogeneity in Previous Studies

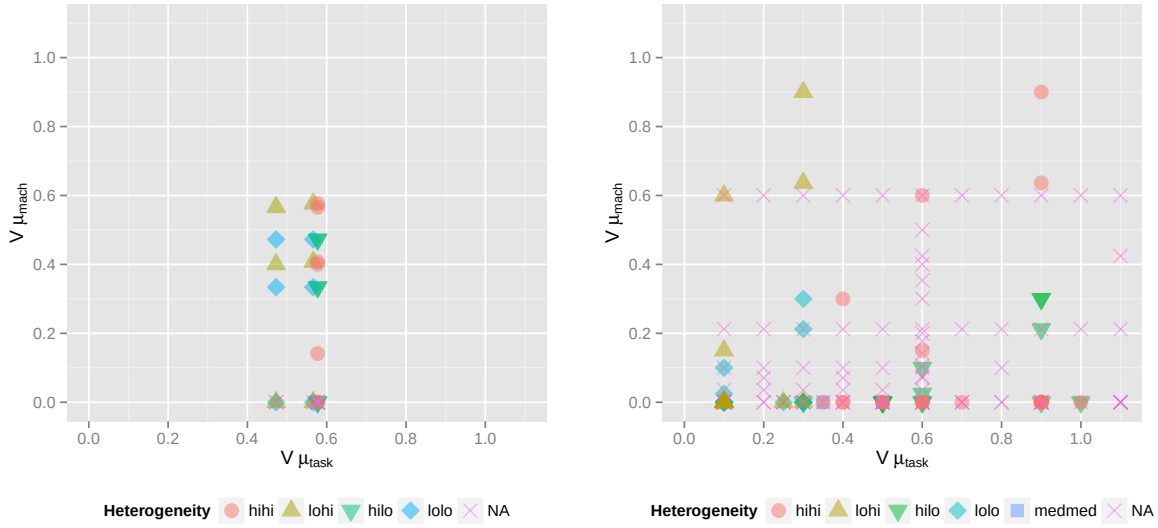
For each of the instances summarized in Section 3, we computed both heterogeneity measures using the previous analysis and the input parameters: R_{task}, R_{mach}, V_{task}, V_{mach}, a and b.

Figures 2 depicts the values for the measures proposed above. The range-based method has a clear bias because many heterogeneity properties have never been obtained. Also, the consistency parameters invalidate the claimed properties of the cost matrices relatively to the heterogeneity quadrant: some *hihi* instances have the same machine heterogeneity as *lolo* instances.

This analysis is also consistent with the observation made in [3] about the fact that the range-based and CVB methods do not cover the entire range of possible values for the MPH.

5 Controlling the Heterogeneity

We are interested in generating cost matrices that have specific heterogeneity properties according to the measures introduced in Section 4. We propose a method that alters a cost matrix generated from uniform instances for which we control the task and machine heterogeneities. This cost matrices have specific properties in terms of consistency and correlation between each row and each column, and the proposed method introduces some randomness in the matrix by shuffling the costs. It first generates the task weights, $\{w_i\}_{1 \leq i \leq n}$, with a gamma distribution with mean one and CV V_{task}, and then the inverse of the machine speeds, $\{b_j\}_{1 \leq j \leq m}$, with a gamma distribution with mean one and CV V_{mach}. The corresponding matrix is computed such that $e_{i,j} = w_i b_j$ before starting the shuffling part. For each cost $e_{i,j}$, another cost $e_{i',j'}$ is selected on a different row and column. The same amount is then removed



(a) Range-based method

(b) CVB method

Figure 2: Heterogeneity properties ($V\mu_{task}$ and $V\mu_{mach}$) of cost matrices used in the literature. Two points are not shown for the CVB method: $(1.4, 0)$ and $(1.8, 0)$.

from these costs and is added to two other costs, $e_{i,j'}$ and $e_{i',j}$ (one that is on the same row as the first cost and on the same column as the second, and another one that is on the same row as the second cost and on the same column as the first). This step preserves the mean of each row and the mean of each column. The heterogeneity properties remain thus the same. The transfered amount is the largest value (in absolute) such that no cost among the four considered costs becomes lower than the minimum one among them (this prevents costs to be arbitrarily low). For instance, if $e_{i,j}$ is the minimum cost (i.e., $e_{i,j} = \min(e_{i,j}, e_{i',j}, e_{i,j'}, e_{i',j'})$), there are two cases: if $e_{i,j'} < e_{i',j}$, then $e_{i,j'}$ becomes the new minimum and the added value to $e_{i,j}$ and to $e_{i',j'}$ is $e_{i,j'} - e_{i,j}$; otherwise, it is $e_{i',j} - e_{i,j}$. This method focuses on preventing costs to be arbitrarily low because it is critical to guarantee positive costs.

Proposition 3. *When used with parameters V_{task} and V_{mach}, the shuffling method generates costs with expected value* 1.

Proof. Costs in the matrix corresponding to the uniform matrix follow a distribution that is the product of two distributions both with mean one. Therefore, the expected value of the costs in the matrix before the shuffling step is also one. The shuffling step do not change the expected value of the costs because the amount that is taken on any cost is given to another cost. □ □

Proposition 4. *The measure $V\mu_{task}$ of a cost matrix generated using the shuffling method with the parameters V_{task} and V_{mach} converges to V_{task} as $n \to \infty$.*

Proof. Analogously to the proof of Proposition 3, the shuffling step has no impact on the mean of each row and each column. The measure $V\mu_{task}$ is thus the same for the final cost matrix as for the intermediate matrix that corresponds to a uniform instance.

The sample CV of the sample means of all rows in this intermediate matrix is equal to the sample CV of the vector $\{w_i\}_{1 \leq i \leq n}$. This last sample CV tends to V_{task} as $n \to \infty$. □ □

An analogous proof relying on the symmetry of the shuffling method shows that $V\mu_{mach}$ converges to V_{mach} as $m \to \infty$.

6 Impact on Scheduling Heuristics

This section assesses the impact of the heterogeneity properties defined in Section 4 on the performance of some classic heuristics. Our intention is not to find the best heuristic but rather to show the impact

Table 1: Summary of the scheduling heuristics for the $R||C_{\max}$ problem.

Name	Ref	Complexity	Remark
Min-min	[10]	n^2m	Earliest finish time
Max-min	[10]	n^2m	Earliest finish time of largest task
GA	[10]	–	Genetic Algorithm
Suff	[12]	n^2m	Task that will suffer most first
HLPT	[16]	$nm + n\log(n)$	Heterogeneous version of LPT
BalSuff	[11]	–	Reconsider allocation on sufferage

of the cost matrix generation method on the performance results. We use classical heuristics from the literature summarized in Table 1. These heuristics are described in [11, Appendix C].

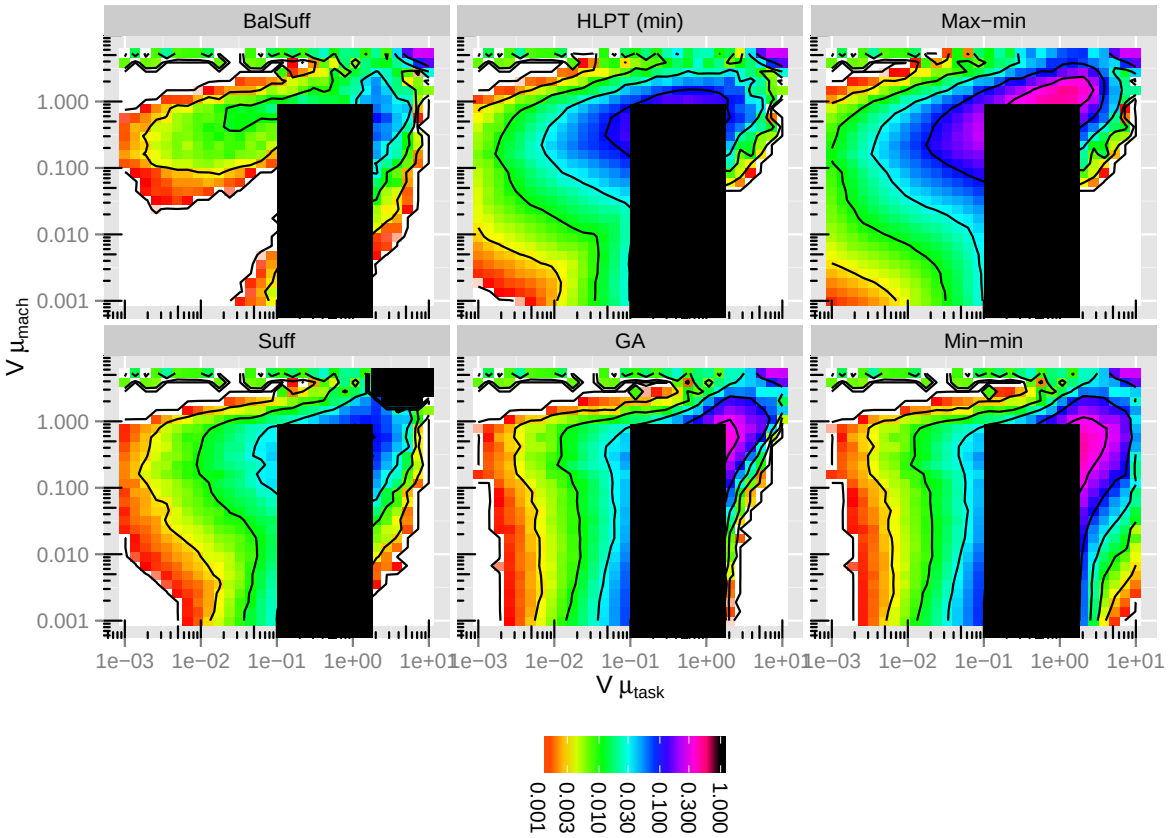


Figure 3: Heuristic performance relatively to the best case with the shuffling method. Values below 0.001 are shown in white and values above 1 are shown in black.

Cost matrices are generated with the shuffling method using $V\mu_{task}$ and $V\mu_{proc}$, each with 30 values exponentially distributed in $[0.001, 10]$. For each pair of parameters, 100 cost matrices are generated with $n = 100$ tasks and $m = 30$ machines. For each scenario, we compute the makespan of each heuristics. We only consider the relative difference from the reference makespan: $|C - C_{\min}|/C_{\min}$ where C is the makespan of a given heuristic and $C_{\min}$ the best makespan we obtained (a genetic algorithm initialized with all the solutions obtained by the other algorithms). The closer to zero, the better the performance.

The results presented on Figure 3 is a heat map of the relative performance for each algorithm. On each figure, we use a logarithmic scale on both axises: the x-axis gives the heterogeneity value for the

tasks ($V\mu_{task}$) while the y-axis gives the heterogeneity value for the machines ($V\mu_{mach}$). The bottom-left area represents almost homogeneous instances, while the top-right area is the most heterogeneous one. The heterogeneity values covered by the range-based and CVB methods in the literature are represented with dark rectangles on each sub-figure. Contour lines correspond to the levels in the legend.

Figure 4 plots the best heuristic depending on the heterogeneity properties. Contour lines show the number of heuristics which performance is closer to the best heuristic than 0.001. For instance, there are at least four heuristics whose relative performances are almost equivalent when task heterogeneity is high. When several heuristics are equivalent for a given tile, the appearing heuristic is the one that is the best the least often. The dark rectangles correspond to the properties covered by the range-based and CVB methods in the literature.

The settings cover a large part of the possible instances for the $R||C_{\max}$ problem. Some areas on the figures may be associated to specific scheduling problems: the $Q|p_i = p|C_{\max}$ problem (top-left area), the $P|p_i = p|C_{\max}$ problem (bottom-left area) and the $P||C_{\max}$ problem (bottom-right area). While the first two problems can be solved in polynomial time, the last problem is NP-complete.

The heat maps suggest that the area where the heterogeneity values are between 0.1 and 1 is more challenging for most heuristics (areas in purple on the heat maps are 30% far from the reference). This is confirmed by Figure 4 where the best heuristic is often far from the second best with these settings. Oppositely, many heuristics are close to the best one when the task heterogeneity is low or high, or when the machine heterogeneity is high. On one hand, execution costs are quite similar when the coefficient of variation is below 0.1. A non-optimal allocation will thus have a lower impact than with higher heterogeneity. On the other hand, most execution costs are close to zero when the coefficient of variation is higher than 1 and bad allocations may be easy to avoid because there are few allocations that are extremely critical while most of them are not. It is thus easier to generate a reasonable schedule. When the machine heterogeneity is low (with medium task heterogeneity), there is often a single best heuristic. This suggests that these settings leads to difficult instances. As mentioned above, this is close to the $P||C_{\max}$ problem. We may conclude that dealing with heterogeneous tasks is more difficult than with heterogeneous machines, which is also supported by the asymmetry of the heat maps.

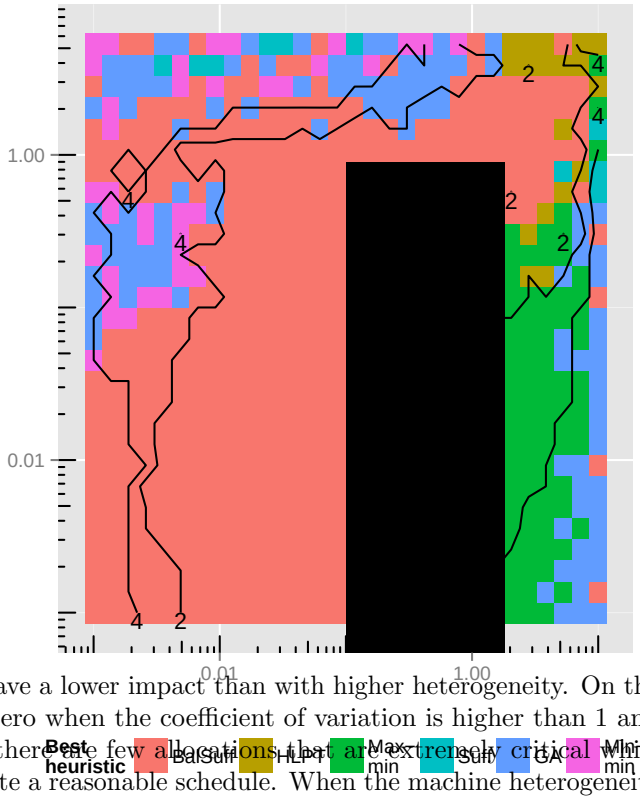


Figure 4: Best heuristic in the average case.

The range-based and CVB generation methods used in the literature could not provide these results due to two factors: the heterogeneity properties of the generated instances have a limited coverage (shown by the dark rectangles) and the erroneous claimed properties of these matrices prevent an unbiased analysis.

This study focuses on the impact of two measures, $V\mu_{task}$ and $V\mu_{proc}$, on the performance of several heuristics. There are however many other properties that could be measured. If we consider the skewness and the kurtosis as in [4], we can think of 4×4 measures for the lines and as many for the columns. The main limitation of this study is to ignore the effect of all these possible measures.

Another limitation is related to the effect of outliers. For large instances, the law of large number applies and the measures proposed in Section 4 correspond to the characteristics of the cost matrices. However, for small instances, we suggest to switch to robust measures such as the median, the interquartile range and the quartile coefficient of dispersion instead of the mean, the standard deviation and the CV, respectively.

7 Conclusion

This study shows that the methods used in the literature for generating cost matrices are biased: the claimed heterogeneity properties of these instances are invalidated by the measures we proposed to quantify them. We also show that the range of instances that has been used are restricted. It is specifically the case for the range-based method that covers only a minor fraction of all the possible settings in terms of heterogeneity. By providing a new cost matrix generation method we show that heuristics for the $R||C_{\max}$ problem have interesting behavior outside this restriction.

In addition to all the possible measures mentioned in Section 6, we plan to analyse other properties, in particular the correlation. It would also be interesting to see if the conclusions hold for some variations of the $R||C_{\max}$ problem such as considering arrival times or online scheduling.

References

[1] Al-Qawasmeh, A., Pasricha, S., Maciejewski, A., Siegel, H.: Power and thermal-aware workload allocation in heterogeneous data centers. TC 64(2), 477–491 (2013)

[2] Al-Qawasmeh, A., Maciejewski, A., Roberts, R.G., Siegel, H.: Characterizing task-machine affinity in heterogeneous computing environments. In: IPDPSW (2011)

[3] Al-Qawasmeh, A., Maciejewski, A., Siegel, H.: Characterizing heterogeneous computing environments using singular value decomposition. In: IPDPSW (2010)

[4] Al-Qawasmeh, A., Maciejewski, A., Wang, H., Smith, J., Siegel, H., Potter, J.: Statistical measures for quantifying task and machine heterogeneities. The Journal of Supercomputing 57(1), 34–50 (2011)

[5] Ali, S.: A comparative study of dynamic mapping heuristics for a class of independent tasks onto heterogeneous computing systems. Ph.D. thesis, Purdue University (1999)

[6] Ali, S., Siegel, H., Maheswaran, M., Hensgen, D.: Task execution time modeling for heterogeneous computing systems. In: HCW. pp. 185–199. IEEE (2000)

[7] Ali, S., Siegel, H., 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)

[8] Armstrong Jr, R.K.: Investigation of effect of different run-time distributions on SmartNet performance. Tech. rep., DTIC Document (1997)

[9] Bardsiri, A.K., Hashemi, S.M.: A Comparative Study on Seven Static Mapping Heuristics for Grid Scheduling Problem. IJSEIA 6(4), 247–256 (2012)

[10] Braun, T.D., Siegel, H., Beck, N., Bölöni, L.L., Maheswaran, M., et al.: A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. JPDC 61(6), 810–837 (2001)

[11] Canon, L.C., Philippe, L.: On the Heterogeneity Bias of Cost Matrices when Assessing Scheduling Algorithms. Tech. rep., FEMTO-ST (Feb 2015)

[12] Casanova, H., Legrand, A., Zagorodnov, D., Berman, F.: Heuristics for scheduling parameter sweep applications in grid environments. In: HCW. pp. 349–363 (2000)

[13] Diaz, C.O., Pecero, J.E., Bouvry, P.: Scalable, low complexity, and fast greedy scheduling heuristics for highly heterogeneous distributed computing systems. The Journal of Supercomputing 67(3), 837–853 (2014)

[14] Friese, R., Khemka, B., Maciejewski, A., Siegel, H.J., Koenig, G., et al.: An analysis framework for investigating the trade-offs between system performance and energy consumption in a heterogeneous computing environment. In: IPDPSW (2013)

[15] Ghosh, S., Henderson, S.G.: Behavior of the norta method for correlated random vector generation as the dimension increases. ACM TOMACS 13(3), 276–294 (2003)

[16] Graham, R.L.: Bounds on Multiprocessing Timing Anomalies. Journal of Applied Mathematics 17(2), 416–429 (1969)

[17] Graham, R.L., Lawler, E.L., Lenstra, J.K., Kan, A.H.G.R.: Optimization and approximation in deterministic sequencing and scheduling: a survey. Annals of Discrete Mathematics 5, 287–326 (1979)

[18] Khemka, B., Friese, Pasricha, Maciejewski, A., Siegel, H., et al.: Utility maximizing dynamic resource management in an oversubscribed energy-constrained heterogeneous computing system. Sustainable Computing: Informatics & Systems (2014)

[19] Leung, J.Y.T. (ed.): Handbook of Scheduling: Algorithms, Models, and Performance Analysis. Chapman & Hall/CCR (2004)

[20] Luo, P., Lü, K., Shi, Z.: A revisit of fast greedy heuristics for mapping a class of independent tasks onto heterogeneous computing systems. JPDC 67(6) (2007)

[21] Saltelli, A., Chan, K., Scott, E.M.: Sensitivity analysis. Wiley New York (2009)