ing.grid preprints

DATASET DESCRIPTOR

Job and Operation Entropy in Job Shop Scheduling: A Dataset

Marco Kemmerling [1]
Maciej Combrzynski-Nogala [1]
Aymen Gannouni [1]
Anas Abdelrazeq [1]
Robert H. Schmitt [1]

1. Chair of Production Metrology and Quality Management & Institute for Information Management in Mechanical Engineering (WZL-MQ/IMA), RWTH Aachen University, Aachen.

Date Submitted:
2023-08-16

Keywords:
job shop problem, entropy, dataset, reinforcement learning, combinatorial optimization

Data availability:
Data can be found here: https://doi.org/10.5281/zenodo.8288814

Software availability:
Software can be found here: https://git.rwth-aachen.de/jobshop/entropy

Abstract. The job shop problem is a highly practically relevant NP-hard problem, which has and continues to receive considerable attention in the literature. Approaches to the problem are typically benchmarked on publicly available datasets containing sets of problem instances. These problem instances are usually generated by some mechanism involving randomisation of instance properties or by maximising instance difficulty, but do not explicitly address properties such as product mix. Product mix, or more generally, diversity in jobs and operations, can be highly variable across different use cases and may affect the suitability of different scheduling methods. We generate a dataset explicitly varying this property by formalising the concept of diversity. To this end, we measure the diversity of jobs and operations in job shop instances using the Shannon entropy and generate instances with specific values of entropy. While our interest is specifically in learning-based approaches to scheduling, the generated instances can serve as a common basis to investigate the impact of instance diversity on a wider variety of different scheduling methods.

1 Introduction

Job shop scheduling has been an area of research with origins going back to at least 1956 [1]. Due to the NP-hardness of the problem, simple heuristics are often used to solve the problem in practice. Recently, the application of reinforcement learning is increasingly investigated for job shop scheduling as well [2]–[4]. In many cases, reinforcement learning essentially learns scheduling or dispatching heuristics. While reinforcement learning can derive scheduling heuristics for the general setting, one of its promises is in learning tailor-made heuristics, i.e. heuristics that are designed to perform specifically well on problems typically encountered on one specific shop floor, rather than in the general job shop scheduling problem. Such tailor-made heuristics would have to rely on the exploitation of some characteristic problem structure in these specific settings.

The structure of a given job shop problem, or a set of problems, is defined by three different objects: machines, jobs, and operations. A given problem instance, or a set of problem instances, can feature more or less commonality or diversity in these objects. Maximum diversity means that all jobs and operations are unique, while repeated jobs and operations decrease diversity. This conception of diversity captures an important practical aspect of job shops: the product mix. Some job shops may tailor their operations to the production of a more narrow range of products, while others may produce a wider mix of products.

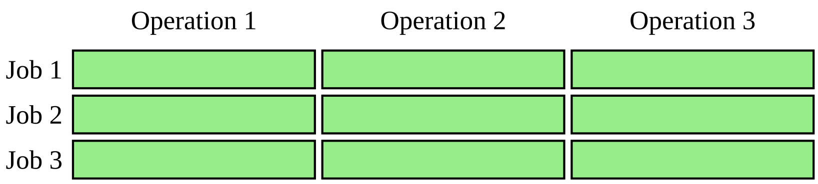
The question then arises whether reinforcement learning agents can learn scheduling heuristics that perform especially well for specific degrees of diversity in jobs and operations. For instance, it may be that scheduling heuristics that perform especially well in the face of low diversity can be found. Further, it is not clear what the impact of this kind of diversity on the solution quality as achieved by (non-exact) methods is. To understand the impact of the degree of diversity in jobs and operations on the performance of reinforcement learning and scheduling approaches in general, we generate job shop problem instances and datasets with varying degrees of diversity. As a foundation for this generation, we formalize different measures of diversity in job shops based on the *Shannon entropy* [5].

Benchmark datasets for the job shop problem have been proposed in the past, but not with a focus on varying diversity. Existing benchmarks such as the well-known Taillard instances [6] instead aim to create instances that are, by some measure, as difficult as possible. With the advent of learning-based scheduling approaches, diversity becomes an increasingly interesting property for the reasons described above. Since our motivation in generating datasets centering around the concept of diversity is thus clearly in studying its impact on learning-based scheduling methods, we will often argue from this perspective in the remainder of this document. The introduced concepts are nevertheless relevant for scheduling methods in general and hence of interest to the operations research community as a whole.

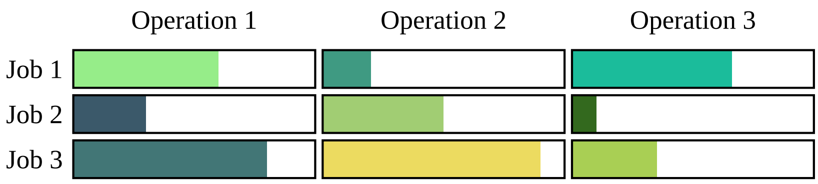
In the remainder of this dataset descriptor, we first give a description of the diversity measures we propose, followed by a description of our generated data, and a detailed description of the procedure used to generate said data. Experiments using the generated data are out of scope for this dataset descriptor and will be carried out in future works.

2 Job & Operation Entropy

A job shop problem instance consists of a set of jobs $\mathcal{J}$, each composed of a set of operations $\mathcal{O}_j \subset \mathcal{O}$, where $\mathcal{O}$ is the set of all operations in the problem instance. Each operation $o \in \mathcal{O}$ has to be processed on a certain machine $m_o \in \mathcal{M}$ for a given duration d_o. The operations of a job are subject to precedence constraints, i.e. they need to be processed in a certain order. Solving such an instance means scheduling all operations in $\mathcal{O}$, i.e. determining when each operation is processed, such that no precedence constraints are violated and only one operation is scheduled on a given machine at a time [7]. For simplicity, we assume that each job $j \in \mathcal{J}$ has the same number of operations $|\mathcal{O}_j| = \frac{|\mathcal{O}|}{|\mathcal{J}|}$, i.e. the operations of the instance are equally divided between all jobs. We further assume that the number of machines equals the number of operations for each job $|\mathcal{M}| = |\mathcal{O}_j|$, and that each machine has a unique machine type represented by an integer. The size of a given instance can then be described as $|\mathcal{J}| \times |\mathcal{O}_j|$, e.g. 6×6, 10×10, and so on. Diversity can be measured either on a job level or on an operation level, and describes



(a) Minimum intra-instance operation entropy. Every operation has the same processing time and required machine type.



(b) Maximum intra-instance operation entropy. Every operation is unique, i.e. has a unique combination of processing time and machine type.

Figure 1: Illustration of intra-instance operation entropy extrema. Each operation is represented by a rectangle, where the color of each rectangle indicates the required machine type, while the processing time is indicated by the amount the rectangle is filled. Note that for illustrative purposes, we have violated the assumption that the number of machines equals the number of operations per job.

how many unique jobs or operations are present within a given collection of jobs and how their frequencies are distributed. The concept of diversity of jobs and operations will be formalized in the following.

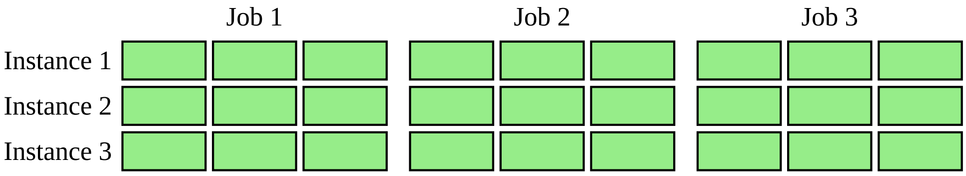
2.1 Intra-Instance Operation Entropy

We begin by focusing our attention on the operation level within a single problem instance. Here, we view two operations as identical if both their processing times and their required machine types are equal. Diversity of operations is then a measure of how many operations within the instance are identical to other operations within the instance, and how many operations are unique. In other words, is a scheduling algorithm expected to repeatedly encounter a smaller number of unique operations, or is it expected to schedule a larger number of unique operations, such that each individual unique operation is encountered less frequently. Figure 1 gives an illustration of examples with minimum and maximum diversity. Between these extremes is a continuum of examples with varying degrees of diversity.

We can formalize this measure of diversity by measuring the frequency of each operation in the instance and calculating the Shannon entropy based on the collected frequencies. By doing so, we essentially view a problem instance as a discrete probability distribution, so that each unique operation corresponds to an event, and the frequency of this operation in the problem instance corresponds to the probability of the event:

$$H(\mathcal{O}) := -\sum_{o \in \mathcal{O}} P(o) \log_{|\mathcal{O}|} P(o) \tag{1}$$

We term the resulting value the *intra-instance operation entropy*. Since the base of the logarithm is chosen as $|\mathcal{O}|$, the value will be 0 for minimum diversity, and 1 for maximum diversity. Note



(a) Minimum inter-instance operation entropy. Every operation in the whole dataset consisting of three instances is the same.

(b) Maximum inter-instance operation entropy. Every operation in the whole dataset is unique.

(c) Minimum inter-instance job entropy. Every job in the dataset is the same. The operations within it *may* vary, but do not necessarily have to.

(d) Maximum inter-instance job entropy. Every job in the dataset is unique, i.e. consists of a unique combination of operations. Individual operations can occur multiple times in the dataset.

Figure 2: Illustration of inter-instance operation and job entropy extrema in a dataset consisting of three instances, each having three jobs, which are each composed of three operations. Each operation is displayed as a rectangle and grouped horizontally with the other operations of the job. Colors represent the required machine type while the processing time is indicated by the amount the corresponding rectangle is filled.

that the intra-instance operation entropy is a property of a problem instance, not a property of a probability distribution from which a problem instance may be sampled.

Intuitively, this intra-instance operation entropy has some connection to the difficulty of a given problem instance. With minimum entropy, every operation is identical and the order of scheduling does not matter at all. Such a minimum entropy problem can hence be considered easy since even random scheduling would lead to an optimal solution. With maximum entropy, the number of unique operations equals the total number of operations. Since every operation is unique, decisions have to be considered more carefully to arrive at good solutions. Note that the notion of difficulty we use here is not about the NP-hardness of the problem, but instead asks the question: how close can a given non-exact method be expected to come to the optimal solution for a given problem? In other words, what is the expected optimality gap of a method for a specific problem instance. The larger it is, the more difficult the instance would be considered to be. While the extremum with minimum entropy provides a clear example of an easy instance, how an instance's difficulty relates to its entropy between the extremes of minimum and maximum entropy remains to be investigated experimentally.

2.2 Inter-Instance Operation Entropy

While the operation entropy described above may be of interest in characterizing single problem instances, testing the ability of reinforcement learning agents to learn tailor-made heuristics requires a view that goes beyond single problem instances. A problem instance may for example be considered one day's worth of jobs in a given shop floor, or some other unit of time. An agent would have to learn to solve not just a single problem instance, but ever new instances as they occur during the daily operations of the shop floor. A specific job shop may produce similar jobs over time, thereby leading to problem instances not entirely different from each other, but sharing some commonalities. To train and test a reinforcement learning agent, we need a collection, or a dataset of such problem instances.

The concept of intra-instance operation entropy can be adapted for this purpose by considering not only the operations of a single instance, but the operations of a whole dataset. The calculation in Equation (1) hence stays the same, merely the meaning of $\mathcal{O}$ is expanded. We call the resulting measure the *inter-instance operation entropy*.

2.3 Intra-Instance & Inter-Instance Job Entropy

The concepts introduced in the two previous subsections can easily be applied to jobs instead of operations. We consider two jobs identical if they consist of the same sequence of operations with identical processing times and required machine types. The *intra-instance job entropy* can then be defined by:

$$H(\mathcal{J}) := -\sum_{j \in \mathcal{J}} P(j) \log_{|\mathcal{J}|} P(j) \tag{2}$$

Similarly, the *inter-instance job entropy* can be defined by considering the set of all jobs in a dataset, rather than all jobs in the problem instance. The extrema of inter-instance job and operation entropy are illustrated in Figure 2.

2.4 Dataset Description

Based on the concepts described above, we generate a number of different datasets. The purpose of the generated datasets is to test scheduling approaches for different settings of job and operation entropy. We generate datasets concerning different levels of inter-instance operation entropy, intra-instance operation entropy, and inter-instance job entropy, as summarized in Table 1. Intra-instance job entropy is not considered here, as the total number of jobs within single instances is typically too small to generate meaningful variation.

By default, we generate 1000 problem instances for each combination of entropy value and problem size. One exception to this are the inter-instance operation entropy datasets, as these require a large set of unique operations to generate datasets of certain entropy levels. This number of required unique operations grows with the number of problem instances. As the uniqueness of an operation is defined by its machine type and processing time, and the number of possible machine types depends on the problem size, the main avenue of generating large sets of unique operations is by defining large ranges of admissible processing times. If the ranges become too wide, the differences between short and long operations become unrealistic. To keep these differences within sensible bounds, we limit the number of instances in the inter-instance operation entropy datasets to 500.

Entropy type	Entropy Values	Dataset size	$\lvert\mathcal{J}\rvert = \lvert\mathcal{O}\rvert$
inter-instance operation	[0.2, 0.3, …, 0.8]	500	[6, 7, …, 15]
intra-instance operation	[0.2, 0.3, …, 0.8]	1000	[6, 7, …, 15]
inter-instance job	[0.2, 0.3, …, 0.8]	1000	[6, 7, …, 15]

Table 1: Overview of the generated datasets characterized by different entropy measures, entropy values, and sizes.

Each entropy dataset is generated to show different levels of diversity as measured by entropy values between 0.2 and 0.8 at 0.1 increments. The dataset size defines the number of instances for each entropy value. For each entropy value, multiple different instance sizes given by $\lvert\mathcal{J}\rvert \times \lvert\mathcal{O}\rvert$ are considered. The full list of generated datasets can be found in Table 2.

3 Data Generation

In the previous sections, we have described our generated data and how we measure its characteristics. In the following, we describe *how* we generate datasets with certain target entropy properties.

As described previously, operation and job entropy are descriptions of the underlying probability distributions of operations and jobs, respectively. To generate job shop instances and datasets with a certain target entropy, we therefore generate a probability distribution with this target entropy and then simply sample from the probability distribution to generate our data.

To generate such probability distributions, denoted as $\mathscr{P}$, we essentially use gradient descent. For ease of modelling and implementation, we define a simple neural network using a single, fully connected hidden layer. The input and output layers have equal dimensions, i.e. the network receives a tensor filled with the scalar value 1 as input and returns a modified distribution matching the desired entropy. This network is not trained to generalise and its weights are only optimised to generate one specific probability distribution. The choice of using a neural network is hence simply due to ease of modelling and the convenience of modern automatic differentiation frameworks. More elegant and efficient ways of generating similar datasets can certainly be devised, but optimising the generation procedure would be ill-spent effort, since it is only executed once to generate our datasets.

The loss function used to train the neural network is composed of the following two terms:

1. The mean squared error between the entropy of the produced probability distribution and the target entropy.
2. A regularization term, defined as a squared difference between the mean of the current probability distribution values and the maximum within them.

The first term allows the network to find a distribution that matches the required entropy, and the regularization term makes sure that they are distributed more uniformly.

The entropy optimizer algorithm follows the following steps:

1. Define the *output size*, which is dependent on the type of the entropy dataset. It defines the maximum size of the operation or job pools, which are sets of unique operations and jobs from which specific operations and jobs for individual instances are sampled subsequently.
2. Run the optimization network for *max episodes*, or until the desired precision is reached, and the uniform validity condition is met. That is defined by the fraction of the distributions with values above the mean.
3. After training each network, filter out values below the frequency threshold, compare the current output's entropy with the best ones, and replace it if necessary.

3.1 Inter-instance job entropy dataset

To generate a dataset with a target entropy for the set of all jobs, it is necessary to determine the entropy probability distribution, which is obtained for the size $|\mathscr{J}| \times |\mathscr{D}|$, where $|\mathscr{D}|$ represents the dataset size. By leveraging the values within $\mathscr{P}$, a job pool is constructed to accommodate the entire dataset. However, a challenge arises due to the rounding of the multiplication between the elements of $\mathscr{P}$ and the dataset size, resulting in an insufficiently sized job pool.

To address this issue while minimizing potential effects on entropy, the pool is augmented by incorporating the least frequent jobs. This ensures that the job pool matches the required size while preserving the desired entropy characteristics. To accomplish this, the frequency counts of jobs within the pool are examined, and the least frequent jobs are identified. These jobs are appended to the pool to compensate for the discrepancy in size.

Once the job pool is created, it is randomly partitioned into $|\mathscr{D}|$ instances, each of the size of $|\mathscr{J}| \times |\mathscr{O}|$.

3.2 Intra-instance operation entropy dataset

To generate a dataset that maintains the target entropy at the instance level, the dataset does not need to be generated all at once. The entropy probability distribution $\mathscr{P}$, is determined for a size equivalent to $|\mathscr{J}| \times |\mathscr{O}|$. Based on this distribution, the operations pool is created.

It is important to note that in order to obtain a set of unique operations, the product of the number of operations and the maximum operation duration should exceed the size of the entropy distribution list. This criterion ensures that there are enough distinct operations for the pool.

Once the pool of operations is created, it is shuffled to introduce more randomness. The pool is then divided into different jobs within an instance. This procedure is repeated until the desired dataset size is reached.

3.3 Inter-instance operation entropy dataset

To generate a dataset with a target entropy for the set of all operations, it is required to determine the entropy distribution list, which is optimized for the size $|\mathscr{J}| \times |\mathscr{O}| \times |\mathscr{D}|$. As a consequence, the size of $\mathscr{P}$ is much larger compared to other dataset types. To ensure that it is possible to create an operation pool consisting of unique operations, the maximum operation duration is increased to 2083 units.

The size of the operation pool is adjusted to fix any rounding issues that may arise from the multiplication of the distribution and the pool size. After the operation pool is created, it is randomly shuffled, and the pool is divided into individual jobs, which are then grouped into instances.

4 Conclusion

We have formalized the property of diversity of jobs and operations in job shop problem instances by introducing the concepts of intra-instance operation entropy, measuring the diversity of operations within single problem instances, inter-instance operation entropy, measuring the diversity of operations within a whole set of problem instances, as well as the similar concepts of intra- and inter-instance job entropy. Based on these concepts, we have devised a method to generate problem instances matching a given target entropy and used it to generate a wide range of different instances belonging to multiple datasets.

We believe our generated datasets are a step towards more research on the effect of job structure in learning-based and traditional scheduling approaches. We hypothesize that reinforcement learning is especially useful in cases of relatively-low inter-instance entropy. In such cases, reinforcement learning may be able to learn tailor-made heuristics exploiting the problem characteristics as measured by the inter-instance entropy, whereas traditional methods need to be able to cope with general scheduling problems. If this hypothesis can be confirmed experimentally, future research will further examine whether combining learning-based methods with planning procedures such as in neural Monte Carlo tree search [8] can compensate for higher entropy levels.

While this is the main motivation behind the generation of our datasets, we can further envision them being used as the basis for curriculum learning approaches [9], where the entropy of instances could be gradually increased during training to vary the problem difficulty. Finally, investigating the impact of operation and job entropy on traditional scheduling methods may be able to deepen the understanding of the impact on job structure on different kinds of potential solutions.

5 Acknowledgements

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy—EXC-2023 Internet of Production—390621612.

6 Roles and contributions

Marco Kemmerling: Conceptualization, Methodology, Writing – original draft, Software, Visualization

Maciej Combrzynski-Nogala: Methodology, Writing – original draft, Software

Aymen Gannouni: Writing - Review & Editing

Anas Abdelrazeq: Writing - Review & Editing

Robert H. Schmitt: Project administration, Funding

References

[1] S. B. Akers Jr, "A Graphical Approach to Production Scheduling Problems," *Operations Research*, vol. 4, no. 2, pp. 244–245, 1956.

[2] V. Samsonov, M. Kemmerling, M. Paegert, *et al.*, "Manufacturing Control in Job Shop Environments with Reinforcement Learning," in *Proceedings of the International Conference on Agents and Artificial Intelligence: ICAART 2021*, vol. 2, 2021, pp. 589–597.

[3] M. Kemmerling, V. Samsonov, D. Lütticke, *et al.*, "Towards Production-Ready Reinforcement Learning Scheduling Agents: A Hybrid Two-Step Training Approach Based on Discrete-Event Simulations," *ASIM Fachtagung Simulation in Produktion und Logistik*, Jan. 2021.

[4] C. Zhang, W. Song, Z. Cao, J. Zhang, P. S. Tan, and X. Chi, "Learning to Dispatch for Job Shop Scheduling via Deep Reinforcement Learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1621–1632, 2020.

[5] C. E. Shannon, "A Mathematical Theory of Communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[6] E. Taillard, "Benchmarks for Basic Scheduling Problems," *European Journal of Operational Research*, vol. 64, no. 2, pp. 278–285, 1993.

[7] M. L. Pinedo, *Scheduling*. Springer, 2012, vol. 29.

[8] M. Kemmerling, D. Lütticke, and R. H. Schmitt, "Beyond Games: A Systematic Review of Neural Monte Carlo Tree Search Applications," *arXiv preprint arXiv:2303.08060*, 2023.

[9] C. Waubert de Puiseau, H. Tercan, and T. Meisen, "Curriculum Learning In Job Shop Scheduling Using Reinforcement Learning," in *Proceedings of the Conference on Production Systems and Logistics: CPSL 2023*, vol. 4, 2023, pp. 34–43.

Appendix

Name	Entropy type	$\lvert\mathcal{D}\rvert$	Entropy	$\lvert\mathcal{J}\rvert \times \lvert\mathcal{O}\rvert$	Optimizer output
inter-op-500-6x6-02	inter-instance operation	500	0.2	36	18000
inter-op-500-6x6-03	inter-instance operation	500	0.3	36	18000
inter-op-500-6x6-04	inter-instance operation	500	0.4	36	18000
inter-op-500-6x6-05	inter-instance operation	500	0.5	36	18000
inter-op-500-6x6-06	inter-instance operation	500	0.6	36	18000
inter-op-500-6x6-07	inter-instance operation	500	0.7	36	18000
inter-op-500-6x6-08	inter-instance operation	500	0.8	36	18000
inter-op-500-7x7-02	inter-instance operation	500	0.2	49	24500
inter-op-500-7x7-03	inter-instance operation	500	0.3	49	24500
inter-op-500-7x7-04	inter-instance operation	500	0.4	49	24500
inter-op-500-7x7-05	inter-instance operation	500	0.5	49	24500
inter-op-500-7x7-06	inter-instance operation	500	0.6	49	24500
inter-op-500-7x7-07	inter-instance operation	500	0.7	49	24500
inter-op-500-7x7-08	inter-instance operation	500	0.8	49	24500
inter-op-500-8x8-02	inter-instance operation	500	0.2	64	32000
inter-op-500-8x8-03	inter-instance operation	500	0.3	64	32000
inter-op-500-8x8-04	inter-instance operation	500	0.4	64	32000
inter-op-500-8x8-05	inter-instance operation	500	0.5	64	32000
inter-op-500-8x8-06	inter-instance operation	500	0.6	64	32000

Name	Entropy type	$\lvert\mathcal{D}\rvert$	Entropy	$\lvert\mathcal{J}\rvert \times \lvert\mathcal{O}\rvert$	Optimizer output
inter-op-500-8x8-07	inter-instance operation	500	0.7	64	32000
inter-op-500-8x8-08	inter-instance operation	500	0.8	64	32000
inter-op-500-9x9-02	inter-instance operation	500	0.2	81	40500
inter-op-500-9x9-03	inter-instance operation	500	0.3	81	40500
inter-op-500-9x9-04	inter-instance operation	500	0.4	81	40500
inter-op-500-9x9-05	inter-instance operation	500	0.5	81	40500
inter-op-500-9x9-06	inter-instance operation	500	0.6	81	40500
inter-op-500-9x9-07	inter-instance operation	500	0.7	81	40500
inter-op-500-9x9-08	inter-instance operation	500	0.8	81	40500
inter-op-500-10x10-02	inter-instance operation	500	0.2	100	50000
inter-op-500-10x10-03	inter-instance operation	500	0.3	100	50000
inter-op-500-10x10-04	inter-instance operation	500	0.4	100	50000
inter-op-500-10x10-05	inter-instance operation	500	0.5	100	50000
inter-op-500-10x10-06	inter-instance operation	500	0.6	100	50000
inter-op-500-10x10-07	inter-instance operation	500	0.7	100	50000
inter-op-500-10x10-08	inter-instance operation	500	0.8	100	50000
inter-op-500-11x11-02	inter-instance operation	500	0.2	121	60500
inter-op-500-11x11-03	inter-instance operation	500	0.3	121	60500
inter-op-500-11x11-04	inter-instance operation	500	0.4	121	60500
inter-op-500-11x11-05	inter-instance operation	500	0.5	121	60500

Name	Entropy type	$\|\mathscr{D}\|$	Entropy	$\|\mathscr{J}\| \times \|\mathscr{O}\|$	Optimizer output
inter-op-500-11x11-06	inter-instance operation	500	0.6	121	60500
inter-op-500-11x11-07	inter-instance operation	500	0.7	121	60500
inter-op-500-11x11-08	inter-instance operation	500	0.8	121	60500
inter-op-500-12x12-02	inter-instance operation	500	0.2	144	72000
inter-op-500-12x12-03	inter-instance operation	500	0.3	144	72000
inter-op-500-12x12-04	inter-instance operation	500	0.4	144	72000
inter-op-500-12x12-05	inter-instance operation	500	0.5	144	72000
inter-op-500-12x12-06	inter-instance operation	500	0.6	144	72000
inter-op-500-12x12-07	inter-instance operation	500	0.7	144	72000
inter-op-500-12x12-08	inter-instance operation	500	0.8	144	72000
inter-op-500-13x13-02	inter-instance operation	500	0.2	169	84500
inter-op-500-13x13-03	inter-instance operation	500	0.3	169	84500
inter-op-500-13x13-04	inter-instance operation	500	0.4	169	84500
inter-op-500-13x13-05	inter-instance operation	500	0.5	169	84500
inter-op-500-13x13-06	inter-instance operation	500	0.6	169	84500
inter-op-500-13x13-07	inter-instance operation	500	0.7	169	84500
inter-op-500-13x13-08	inter-instance operation	500	0.8	169	84500
inter-op-500-14x14-02	inter-instance operation	500	0.2	196	98000
inter-op-500-14x14-03	inter-instance operation	500	0.3	196	98000
inter-op-500-14x14-04	inter-instance operation	500	0.4	196	98000

Name	Entropy type	$\lvert\mathscr{D}\rvert$	Entropy	$\lvert\mathscr{J}\rvert \times \lvert\mathscr{O}\rvert$	Optimizer output
inter-op-500-14x14-05	inter-instance operation	500	0.5	196	98000
inter-op-500-14x14-06	inter-instance operation	500	0.6	196	98000
inter-op-500-14x14-07	inter-instance operation	500	0.7	196	98000
inter-op-500-14x14-08	inter-instance operation	500	0.8	196	98000
inter-op-500-15x15-02	inter-instance operation	500	0.2	225	112500
inter-op-500-15x15-03	inter-instance operation	500	0.3	225	112500
inter-op-500-15x15-04	inter-instance operation	500	0.4	225	112500
inter-op-500-15x15-05	inter-instance operation	500	0.5	225	112500
inter-op-500-15x15-06	inter-instance operation	500	0.6	225	112500
inter-op-500-15x15-07	inter-instance operation	500	0.7	225	112500
inter-op-500-15x15-08	inter-instance operation	500	0.8	225	112500
intra-op-1000-6x6-02	intra-instance operation	1000	0.2	36	36
intra-op-1000-6x6-03	intra-instance operation	1000	0.3	36	36
intra-op-1000-6x6-04	intra-instance operation	1000	0.4	36	36
intra-op-1000-6x6-05	intra-instance operation	1000	0.5	36	36
intra-op-1000-6x6-06	intra-instance operation	1000	0.6	36	36
intra-op-1000-6x6-07	intra-instance operation	1000	0.7	36	36
intra-op-1000-6x6-08	intra-instance operation	1000	0.8	36	36
intra-op-1000-7x7-02	intra-instance operation	1000	0.2	49	49
intra-op-1000-7x7-03	intra-instance operation	1000	0.3	49	49

Name	Entropy type	$\|\mathscr{D}\|$	Entropy	$\|\mathscr{J}\| \times \|\mathscr{O}\|$	Optimizer output
intra-op-1000-7x7-04	intra-instance operation	1000	0.4	49	49
intra-op-1000-7x7-05	intra-instance operation	1000	0.5	49	49
intra-op-1000-7x7-06	intra-instance operation	1000	0.6	49	49
intra-op-1000-7x7-07	intra-instance operation	1000	0.7	49	49
intra-op-1000-7x7-08	intra-instance operation	1000	0.8	49	49
intra-op-1000-8x8-02	intra-instance operation	1000	0.2	64	64
intra-op-1000-8x8-03	intra-instance operation	1000	0.3	64	64
intra-op-1000-8x8-04	intra-instance operation	1000	0.4	64	64
intra-op-1000-8x8-05	intra-instance operation	1000	0.5	64	64
intra-op-1000-8x8-06	intra-instance operation	1000	0.6	64	64
intra-op-1000-8x8-07	intra-instance operation	1000	0.7	64	64
intra-op-1000-8x8-08	intra-instance operation	1000	0.8	64	64
intra-op-1000-9x9-02	intra-instance operation	1000	0.2	81	81
intra-op-1000-9x9-03	intra-instance operation	1000	0.3	81	81
intra-op-1000-9x9-04	intra-instance operation	1000	0.4	81	81
intra-op-1000-9x9-05	intra-instance operation	1000	0.5	81	81
intra-op-1000-9x9-06	intra-instance operation	1000	0.6	81	81
intra-op-1000-9x9-07	intra-instance operation	1000	0.7	81	81
intra-op-1000-9x9-08	intra-instance operation	1000	0.8	81	81
intra-op-1000-10x10-02	intra-instance operation	1000	0.2	100	100

Name	Entropy type	$\vert\mathscr{D}\vert$	Entropy	$\vert\mathscr{J}\vert \times \vert\mathscr{O}\vert$	Optimizer output
intra-op-1000-10x10-03	intra-instance operation	1000	0.3	100	100
intra-op-1000-10x10-04	intra-instance operation	1000	0.4	100	100
intra-op-1000-10x10-05	intra-instance operation	1000	0.5	100	100
intra-op-1000-10x10-06	intra-instance operation	1000	0.6	100	100
intra-op-1000-10x10-07	intra-instance operation	1000	0.7	100	100
intra-op-1000-10x10-08	intra-instance operation	1000	0.8	100	100
intra-op-1000-11x11-02	intra-instance operation	1000	0.2	121	121
intra-op-1000-11x11-03	intra-instance operation	1000	0.3	121	121
intra-op-1000-11x11-04	intra-instance operation	1000	0.4	121	121
intra-op-1000-11x11-05	intra-instance operation	1000	0.5	121	121
intra-op-1000-11x11-06	intra-instance operation	1000	0.6	121	121
intra-op-1000-11x11-07	intra-instance operation	1000	0.7	121	121
intra-op-1000-11x11-08	intra-instance operation	1000	0.8	121	121
intra-op-1000-12x12-02	intra-instance operation	1000	0.2	144	144
intra-op-1000-12x12-03	intra-instance operation	1000	0.3	144	144
intra-op-1000-12x12-04	intra-instance operation	1000	0.4	144	144
intra-op-1000-12x12-05	intra-instance operation	1000	0.5	144	144
intra-op-1000-12x12-06	intra-instance operation	1000	0.6	144	144
intra-op-1000-12x12-07	intra-instance operation	1000	0.7	144	144
intra-op-1000-12x12-08	intra-instance operation	1000	0.8	144	144

Name	Entropy type	$\|\mathcal{D}\|$	Entropy	$\|\mathcal{J}\| \times \|\mathcal{O}\|$	Optimizer output
intra-op-1000-13x13-02	intra-instance operation	1000	0.2	169	169
intra-op-1000-13x13-03	intra-instance operation	1000	0.3	169	169
intra-op-1000-13x13-04	intra-instance operation	1000	0.4	169	169
intra-op-1000-13x13-05	intra-instance operation	1000	0.5	169	169
intra-op-1000-13x13-06	intra-instance operation	1000	0.6	169	169
intra-op-1000-13x13-07	intra-instance operation	1000	0.7	169	169
intra-op-1000-13x13-08	intra-instance operation	1000	0.8	169	169
intra-op-1000-14x14-02	intra-instance operation	1000	0.2	196	196
intra-op-1000-14x14-03	intra-instance operation	1000	0.3	196	196
intra-op-1000-14x14-04	intra-instance operation	1000	0.4	196	196
intra-op-1000-14x14-05	intra-instance operation	1000	0.5	196	196
intra-op-1000-14x14-06	intra-instance operation	1000	0.6	196	196
intra-op-1000-14x14-07	intra-instance operation	1000	0.7	196	196
intra-op-1000-14x14-08	intra-instance operation	1000	0.8	196	196
intra-op-1000-15x15-02	intra-instance operation	1000	0.2	225	225
intra-op-1000-15x15-03	intra-instance operation	1000	0.3	225	225
intra-op-1000-15x15-04	intra-instance operation	1000	0.4	225	225
intra-op-1000-15x15-05	intra-instance operation	1000	0.5	225	225
intra-op-1000-15x15-06	intra-instance operation	1000	0.6	225	225
intra-op-1000-15x15-07	intra-instance operation	1000	0.7	225	225

Name	Entropy type	$\vert\mathcal{D}\vert$	Entropy	$\vert\mathcal{J}\vert \times \vert\mathcal{O}\vert$	Optimizer output
intra-op-1000-15x15-08	intra-instance operation	1000	0.8	225	225
inter-job-1000-6x6-02	inter-instance job	1000	0.2	36	6000
inter-job-1000-6x6-03	inter-instance job	1000	0.3	36	6000
inter-job-1000-6x6-04	inter-instance job	1000	0.4	36	6000
inter-job-1000-6x6-05	inter-instance job	1000	0.5	36	6000
inter-job-1000-6x6-06	inter-instance job	1000	0.6	36	6000
inter-job-1000-6x6-07	inter-instance job	1000	0.7	36	6000
inter-job-1000-6x6-08	inter-instance job	1000	0.8	36	6000
inter-job-1000-7x7-02	inter-instance job	1000	0.2	49	7000
inter-job-1000-7x7-03	inter-instance job	1000	0.3	49	7000
inter-job-1000-7x7-04	inter-instance job	1000	0.4	49	7000
inter-job-1000-7x7-05	inter-instance job	1000	0.5	49	7000
inter-job-1000-7x7-06	inter-instance job	1000	0.6	49	7000
inter-job-1000-7x7-07	inter-instance job	1000	0.7	49	7000
inter-job-1000-7x7-08	inter-instance job	1000	0.8	49	7000
inter-job-1000-8x8-02	inter-instance job	1000	0.2	64	8000
inter-job-1000-8x8-03	inter-instance job	1000	0.3	64	8000
inter-job-1000-8x8-04	inter-instance job	1000	0.4	64	8000
inter-job-1000-8x8-05	inter-instance job	1000	0.5	64	8000
inter-job-1000-8x8-06	inter-instance job	1000	0.6	64	8000
inter-job-1000-8x8-07	inter-instance job	1000	0.7	64	8000
inter-job-1000-8x8-08	inter-instance job	1000	0.8	64	8000
inter-job-1000-9x9-02	inter-instance job	1000	0.2	81	9000
inter-job-1000-9x9-03	inter-instance job	1000	0.3	81	9000
inter-job-1000-9x9-04	inter-instance job	1000	0.4	81	9000
inter-job-1000-9x9-05	inter-instance job	1000	0.5	81	9000
inter-job-1000-9x9-06	inter-instance job	1000	0.6	81	9000
inter-job-1000-9x9-07	inter-instance job	1000	0.7	81	9000
inter-job-1000-9x9-08	inter-instance job	1000	0.8	81	9000
inter-job-1000-10x10-02	inter-instance job	1000	0.2	100	10000
inter-job-1000-10x10-03	inter-instance job	1000	0.3	100	10000
inter-job-1000-10x10-04	inter-instance job	1000	0.4	100	10000
inter-job-1000-10x10-05	inter-instance job	1000	0.5	100	10000
inter-job-1000-10x10-06	inter-instance job	1000	0.6	100	10000
inter-job-1000-10x10-07	inter-instance job	1000	0.7	100	10000
inter-job-1000-10x10-08	inter-instance job	1000	0.8	100	10000
inter-job-1000-11x11-02	inter-instance job	1000	0.2	121	11000
inter-job-1000-11x11-03	inter-instance job	1000	0.3	121	11000
inter-job-1000-11x11-04	inter-instance job	1000	0.4	121	11000
inter-job-1000-11x11-05	inter-instance job	1000	0.5	121	11000

Name	Entropy type	$\|\mathscr{D}\|$	Entropy	$\|\mathscr{J}\| \times \|\mathscr{O}\|$	Optimizer output
inter-job-1000-11x11-06	inter-instance job	1000	0.6	121	11000
inter-job-1000-11x11-07	inter-instance job	1000	0.7	121	11000
inter-job-1000-11x11-08	inter-instance job	1000	0.8	121	11000
inter-job-1000-12x12-02	inter-instance job	1000	0.2	144	12000
inter-job-1000-12x12-03	inter-instance job	1000	0.3	144	12000
inter-job-1000-12x12-04	inter-instance job	1000	0.4	144	12000
inter-job-1000-12x12-05	inter-instance job	1000	0.5	144	12000
inter-job-1000-12x12-06	inter-instance job	1000	0.6	144	12000
inter-job-1000-12x12-07	inter-instance job	1000	0.7	144	12000
inter-job-1000-12x12-08	inter-instance job	1000	0.8	144	12000
inter-job-1000-13x13-02	inter-instance job	1000	0.2	169	13000
inter-job-1000-13x13-03	inter-instance job	1000	0.3	169	13000
inter-job-1000-13x13-04	inter-instance job	1000	0.4	169	13000
inter-job-1000-13x13-05	inter-instance job	1000	0.5	169	13000
inter-job-1000-13x13-06	inter-instance job	1000	0.6	169	13000
inter-job-1000-13x13-07	inter-instance job	1000	0.7	169	13000
inter-job-1000-13x13-08	inter-instance job	1000	0.8	169	13000
inter-job-1000-14x14-02	inter-instance job	1000	0.2	196	14000
inter-job-1000-14x14-03	inter-instance job	1000	0.3	196	14000
inter-job-1000-14x14-04	inter-instance job	1000	0.4	196	14000
inter-job-1000-14x14-05	inter-instance job	1000	0.5	196	14000
inter-job-1000-14x14-06	inter-instance job	1000	0.6	196	14000
inter-job-1000-14x14-07	inter-instance job	1000	0.7	196	14000
inter-job-1000-14x14-08	inter-instance job	1000	0.8	196	14000
inter-job-1000-15x15-02	inter-instance job	1000	0.2	225	15000
inter-job-1000-15x15-03	inter-instance job	1000	0.3	225	15000
inter-job-1000-15x15-04	inter-instance job	1000	0.4	225	15000
inter-job-1000-15x15-05	inter-instance job	1000	0.5	225	15000
inter-job-1000-15x15-06	inter-instance job	1000	0.6	225	15000
inter-job-1000-15x15-07	inter-instance job	1000	0.7	225	15000
inter-job-1000-15x15-08	inter-instance job	1000	0.8	225	15000

Table 2: Table listing detailed information about generated datasets. Each dataset's name is composed of the following information: the type of entropy considered, the size of the dataset, the size of the instances within it, and the entropy level. The optimizer output is the size of the output layer of the neural network that finds the probability distribution for a given target entropy. The larger the optimizer output is, the more unique operations will be generated.