Developing a Ranking Problem Library (RPLIB) from a data-oriented perspective

Paul E. Anderson1 · Amy N. Langville2 · Kathryn E. Pedings-Behling2 · Brandon Tat1 · Charlie Ward1

Received: 4 May 2022 / Revised: 7 September 2022 / Accepted: 7 September 2022 / Published online: 23 September 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
We present an improved library for the ranking problem called RPLIB. RPLIB includes the following data and features. (1) Real and artificial datasets of both pairwise data (i.e., information about the ranking of pairs of items) and feature data (i.e., a vector of features about each item to be ranked). These datasets range in size (e.g., from small \( n = 10 \) item datasets to large datasets with hundreds of items), application (e.g., from sports to economic data), and source (e.g., real versus artificially generated to have particular structures). (2) RPLIB contains code for the most common ranking algorithms such as the linear ordering optimization method and the Massey method. (3) RPLIB also has the ability for users to contribute their own data, code, and algorithms. Each RPLIB dataset has an associated JSON model card of additional information such as the number and set of optimal rankings, the optimal objective value, and corresponding figures.

Keywords Ranking problem library · Linear ordering problem · Integer programming · Linear programming · Rankability · Artificial data · Massey rankings · Colley rankings

Amy N. Langville
langvillea@cofc.edu
Paul E. Anderson
pander14@calpoly.edu
Kathryn E. Pedings-Behling
kathryn@behling.org

1 Department of Computer Science and Software Engineering, California Polytechnic State University, San Luis Obispo, CA, USA
2 Department of Mathematics, College of Charleston, Charleston, SC 29401, USA
1 Introduction

Constructing a ranking of items from a set of pairwise comparisons is a common problem with applications in a wide range of fields including sports, economics, social science, and many others. A ranking problem, also known as a linear ordering problem, can be defined as a dataset of items to be ranked. Such a dataset could include features of the items or pairwise comparisons between items. The current approach to producing rankings often focuses on selecting the single ranking best supported by the observed data but fails to communicate the ranking’s uncertainty. In prior work (Anderson et al. 2021b, 2019a, 2021a, 2020; Cameron et al. 2020), we introduced the idea of the rankability of a ranking dataset. We tied rankability to the presence, number, and properties of multiple optimal solutions and thereby quantified a dataset’s inherent ability to produce a meaningful ranking of its items.

Rankability and the linear ordering problem (LOP) are two related but distinct problems. The LOP can be defined as a problem of finding an acyclic tournament with maximum weight in a given weighted complete directed graph. The Rankability problem is very new, proposed in 2019, and does not yet have a standard definition. At least half a dozen definitions for a dataset’s rankability exist. For example, in prior work (Anderson et al. 2019a), our team created a definition of rankability that extends the LOP to find not just the maximum weight but also the number and set of acyclic tournaments that achieve that maximum weight. Our team has created a few other definitions of rankability (Anderson et al. 2021b, 2021a, 2020) and research teams at Maynooth University and Oxford University have added several more (Cameron et al. 2020; McJames et al. 2022; Hu et al. 2022). One hope for RPLIB is that it will create a standard definition of rankability as well as standard benchmark measures.

Often progress in a particular field has been sparked by public datasets and libraries, e.g., ImageNet (Deng et al. 2009), the Linear Ordering Library (Martí 2009), and kaggle.com/. For example, TSPLIB (Reinelt 2017) provides researchers of the traveling salesman problem (TSP) a unified place and problem instances by which to compare their algorithm’s speed, scalability, and solutions and to compete for titles such as the largest TSP solved or fastest runtime. RPLIB, our Rankability Problem Library, was motivated by such libraries.

We hope RPLIB will spark similar research for the ranking community and its instances will serve as comparative benchmarks. By describing RPLIB here, we also aim to encourage the ranking community to attack some of the open questions we pose in Sect. 5. RPLIB tools, datasets, documentation, and more can be found at https://rplib.birg.dev/.
2 Related work

RPLIB builds on the older existing ranking library LOLIB, the Linear Ordering Library (Martí 2009). LOLIB is an excellent repository of ranking data, but its datasets are static and in need of update. For example, an LOLIB text file contains just three pieces of information: the problem size, the optimal objective value for the linear ordering optimization, and one optimal ranking. In prior work Anderson et al. (2021a, 2021b), we discovered that 98% of the 50 instances in the LOLIB IO folder (Grötschel et al. 1984) actually have more than one optimal ranking, yet just one is stored in the LOLIB text file. Unlike LOLIB, RPLIB algorithms produce and the library stores, in its novel JSON model card (Sect. 3.4.1), all (or many if the user terminates the algorithm early) alternate optimal rankings associated with an instance. This is a novel contribution of RPLIB. Figure 1 summarizes several ways in which RPLIB extends LOLIB.

Another new contribution in this paper, in Sect. 5, is the extension of the definition of rankability from three measures to four measures. We use RPLIB instances and algorithms to demonstrate the additional benefits of this fourth rankability measure.

3 RPLIB

The RPLIB, released under the MIT open source license, contains data, algorithms, code, and web applications for user contributions. These are described in the following subsections.

| Feature                                      | LOLIB | RPLIB |
|----------------------------------------------|-------|-------|
| Gives an optimal value                       | ✓     | ✓     |
| Provides multiple optimal solutions          |       | ✓     |
| Gives results from multiple ranking methods  |       | ✓     |
| Ability for the community to submit datasets |       | ✓     |
| Interactive components available for use and download |       | ✓     |

Fig. 1 A comparison of the features available in LOLIB versus those in RPLIB
3.1 Datasets

Each dataset in RPLIB has a unique ID number so that researchers can clearly cite, find, and reference datasets. RPLIB has three types of unprocessed data: (1) static real data ranging from college basketball scores, features associated with U.S. colleges, organized and standardized versions of LOLIB instances, which include economic input-output instances as well as special artificial instances; (2) dynamic artificial data that creates rankings with specific properties; and (3) user-contributed data.

The RPLIB website contains additional details beyond those in this paper. See https://rplib.birg.dev/.

3.1.1 Static real data

RPLIB contains datasets from several real ranking applications.

- Game data from seasons of U.S. college basketball, including the March Madness tournament.
- Economic input-output matrices from Japan’s economic sectors for 1995 and 2005 (Kondo 2014).
- Economic input-output matrices that were originally hosted within the linear ordering libraries, LOLIB and XLOLIB. These datasets have been updated and organized in RPLIB.
- Special collections from LOLIB and XLOLIB. These datasets were also cleaned and organized when moving them to RPLIB.
- US News & World Report data on features associated with liberal arts colleges.

This collection can be filtered, searched, and exported for research purposes. For example, to find within RPLIB’s hundreds of instances, instances from LOLIB where the problem size $n$ is between $30 \leq n \leq 35$, use the filter command as shown in Fig. 2.

To find all instances pertaining to March Madness basketball that contain greater than 1000 optimal rankings, use the filter command as shown in Fig. 3.

![Fig. 2](image-url) This screenshot shows a powerful feature of RPLIB, the ability to filter the hundreds of problem instances to find instances satisfying certain conditions. This example finds all instances from the LOLIB collection that have $30 < n < 35$. 
Developing a Ranking Problem Library (RPLIB) from a…

3.1.2 Dynamic artificial data

RPLIB also has the ability for users to generate artificial instances that have desired structure. An RPLIB user has the ability to tune various parameters, such as the size of the instance or the amount of noise added to a matrix. These artificial datasets are useful for theoretical studies, particularly for the open questions posed in Sect. 5. Clicking on the “Artificial Structured Datasets” tab on the left menu of the RPLIB homepage opens a Google Colaboratory Notebook.

Fig. 3 This screenshot shows all March Madness instances that contain greater than 1000 optimal rankings

Fig. 4 The screenshot above shows how to create an artificial dataset in the Colab Notebook section of RPLIB. Specifically, this creates a 5 × 5 empty matrix where 20% of the data is replaced with random entries between the lowerbound of 2 and the upperbound of 4
Figure 4 is screenshot from the RPLIB Google Colaboratory Notebook. In this example, the command `emptyplusnoise(5, 20, 2, 4)` creates a $5 \times 5$ empty matrix where 20% of the data is replaced with random entries between the lower-bound of 2 and the upperbound of 4.

Currently the following structures of artificial data are available. These artificial datasets were first described in our prior work on rankability (Anderson et al. 2019a).

- empty matrix (i.e., matrix of all zeros)
- empty matrix plus a matrix with random noise
- a completely connected matrix
- a completely connected matrix minus a matrix with random noise
- perfect ranking matrix in hillside form plus random noise
- perfect ranking matrix in dominance form plus random noise
- cyclic matrix
- matrix structured so that it has $c!$ multiple optimal rankings with variations in the top, middle, or bottom of the ranking. See Sect. 4.
- ability to turn weighted matrices into unweighted matrices
- ability to add or remove a percentage of entries in the matrix (i.e., edges in the associated graph)
- ability to generate a $D$ matrix by simulating individual games with a probability of upset

### 3.1.3 User-contributed data

RPLIB makes it easy for users to contribute their own data, either real static data or artificial structured data. There are two ways to make a contribution. (1) An RPLIB user can follow the “Contribute Dataset” link on the RPLIB homepage, which opens a form that prompts the user for their name and email, the name and description of the dataset, the type of data, and the data itself. Contributions will be reviewed by the RPLIB team prior to being added to the library. (2) Alternatively, RPLIB users can create artificial datasets via RPLIB’s Colab tool and then follow the link within that tool to directly submit their generated data. See Fig. 5. Contributed data is licensed under the standard MIT open source license.

Ranking research has continued for decades, whereas rankability research has been recent. We invite researchers from these communities to add methods, data, and code to RPLIB. For example, rankability researchers from the U.S. and Europe have created interesting artificial datasets and new methods (Cameron et al. 2020; McJames et al. 2022).

### 3.2 Dataset processing

In this section, we use a few examples to explain the Processed Data section of RPLIB. Figure 6 contains screenshots taken directly from RPLIB and highlights the difference between unprocessed and processed data. This example is
Developing a Ranking Problem Library (RPLIB) from a…

for the 2002 season of March Madness basketball. Screenshot A is the unprocessed data. Each row contains data for one game that season, showing the point score information. For example, row 1 indicates that Arizona scored 71 points, beating Maryland who scored 67 points. Screenshot B shows a powerful feature of RPLIB—the ability to search and filter data. In this example, we filtered this unprocessed instance to find all games between Cincinnati and Charlotte during that 2002 season. Contrast this with Screenshot C, which shows a subset of the (alphabetically ordered) processed data, a square matrix of pairwise comparisons. An empty cell means the two teams have not played each other. Non-empty cell \((i, j)\) contains the number of times team \(i\) beat team \(j\) in that matchup.

Our next example is the US News & World Report data, which is used to rank U.S. colleges. RPLIB contains a subset of this data, specifically Liberal Arts colleges. Figure 7 shows both the unprocessed and processed data. Figure 7A is the unprocessed data, which are data associated with various features (e.g., graduation rate, retention rate) for each college. Once again, the data is processed to create a square matrix of pairwise comparisons, as shown in Fig. 7B.

We invite the community to add new transformations that convert unprocessed data to processed data. This can be done with a standard GitHub pull request. Once verified, such transformations can be run on all datasets through RPLIB’s infrastructure for automated analysis (Fig. 8).
The library contains implementations of two classes of ranking methods:

1. Optimization: The linear ordering method (Marti and Reinelt 2011) and its hillside variant (Langville et al. 2011; Anderson et al. 2019a, 2021a, 2020, 2019b) are available. These methods output an optimal value, a collection of multiple optimal rankings (if any exist), the diameter of the set of multiple optimal rankings found, the solutions nearest to and farthest from the centroid of the optimal face, a matrix $X^*$ (Anderson et al. 2021a) containing rank-ordered information about pairs of items, and a measure indicating the location of the indecision among the set of optimal rankings.

Fig. 6 These screenshots are taken directly from RPLIB. Screenshot A is the unprocessed data for the 2002 March Madness basketball season. Screenshot B shows a powerful feature of RPLIB—the ability to search and filter data. In this example, we filtered the unprocessed data to find all instances of games between Cincinnati and Charlotte. Screenshot C is a subset of the (alphabetically ordered) processed data, which creates a square matrix of pairwise comparisons. Empty cells mean the teams have not played each other. Non-empty cell $(i,j)$ is the number of times team $i$ beat team $j$ in that matchup.

3.3 RPLIB methods

The library contains implementations of two classes of ranking methods:
Developing a Ranking Problem Library (RPLIB) from a…

2. Linear Algebra: The Massey (Massey 1997) and Colley (Colley 2002) linear systems are available. These methods output the rating vector, its associated ranking vector, and a matrix $Y^*$ similar to the $X^*$ mentioned above, where $Y^*(i,j)$ gives information about the certainty that item $i$ is ranked above item $j$. The rating vector can be used to create a pseudo-optimal set of rankings.

3.4 RPLIB outputs

3.4.1 RPLIB model card

Because ranking is a common task in machine learning, we adopt the standard documentation from that field, namely storing and visualizing the output as a model card (Mitchell et al. 2019). An RPLIB instance contains a JSON file and a Google Colab Notebook that generates images associated with that instance. We intentionally chose JSON as the format for RPLIB data. JSON (JavaScript Object Notation) format is favored for its readability, nesting features, interoperability, and library support in modern programming languages. Because tools that convert JSON to other languages are common, our choice of JSON increases the usability of RPLIB (Fig. 8). The JSON file contains some (or all) of the following items, which are

Fig. 7 These screenshots are taken directly from RPLIB. Screenshot A is the unprocessed data for 2002 US News and World Report Liberal Arts College data. Screenshot B shows a powerful feature of RPLIB—the ability to search and filter data. In this example, we filtered the unprocessed data and compared the data from Amherst and Bowdoin Colleges. Screenshot C is a subset of the (alphabetically ordered) processed data, which creates a square matrix of pairwise comparisons. Entry $(i, j)$ is the number of features out of the six total where college $i$ outperformed college $j$. 
displayed in our RPLIB tool as a model card. The Colab Notebook with images is described in the next section.

- Dominance data. This can be either a square matrix $D$ of dominance relations where $d_{ij}$ is non-zero if $i$ dominates $j$ or a rectangular matrix of features for each item.
- Problem size $n$, i.e., the number of items to be ranked.
- Optimal objective value. As described in Anderson et al. (2019a), many ranking and rankability methods optimize an objective function.
- Optimal ranking(s).
- Diameter of the set of optimal rankings, i.e., the distance (e.g., Kendall tau) between the two farthest optimal rankings.
- Farthest (by Kendall tau) pair of optimal rankings.
- Closest (by Kendall tau) pair of distinct optimal rankings.
- Ranking closest to the centroid of all optimal rankings.
- Ranking farthest from the centroid of all optimal rankings.
- full set $P$ of all optimal rankings for some instances with small $n$. 

**Fig. 8** A JSON Model Card. In this instance with dataset_id 363, there are 1,095 optimal rankings with an objective value of 204. Clicking on a JSON item that has an arrow, such as D, expands to show more information. D is a $65 \times 65$ matrix. The first row corresponds to the team labeled Gonzaga and this can be further expanded to reveal the data for the 65 entries on this row of the D matrix. Similarly, expanding the centroid_solution displays a vector of 65 elements, showing that the team with ID 0 is in rank position 6, team with ID 1 is in rank position 55, and so on.
Developing a Ranking Problem Library (RPLIB) from a…

3.4.2 RPLIB figure outputs

Each RPLIB instance can be loaded in a Google Colab Notebook (see the top left of Fig. 9) and associated methods run, which generate associated measures (top right of Fig. 9), and figures, such as a pixel plot of nonzero elements in a reordered $X(r, r)^*$ matrix and a spaghetti plot between the farthest pair of optimal rankings (see the bottom of Fig. 9).

4 Example use of RPLIB: refining the definition of rankability

In prior work, Anderson et al. define rankability $r$ as a function of three measures, $k$, the distance of $D$ from a perfectly rankable data matrix, $|P|$, the number of optimal rankings, and $\tau$, the diameter (i.e., the distance between the two farthest rankings) of the set of optimal rankings (Anderson et al. 2021b, 2019a, 2021a, 2020, 2019b).
Mathematically, they define $r = f(k, |P|, \tau)$ and show that rankability $r$ is correlated with predictability on the ranking task of predicting outcomes in the March Madness college basketball tournament. In other words, March Madness outcomes are straightforward and follow the rankings in more rankable years and are wild for less rankable years. Figure 10 shows the $X^*$ matrices for two years of March Madness, 2002 and 2008. This figure plots the fractional elements in $X^*$, which measure the locations of disagreements among the optimal rankings in the set $P$ of optimal rankings. Notice that for 2002, there is little disagreement about the teams that belong at the top of the ranking, while there is much more disagreement about the bottom of the ranking. The 2008 year shows more disagreement in the top of the ranking, certainty in the middle of the ranking, and some disagreement in the bottom. With respect to March Madness, we expect the 2008 year to have some upsets in later rounds of the tournament, when two top teams face each other. In this section, we quantify the uncertainty in locations of the ranking by creating a fourth rankability measure, called $\delta$, which we define and describe in the remainder of this section.

RPLIB provides a standardized approach for generating and archiving structured artificial data. We demonstrate the importance of archiving structured artificial data for the research community and the utility of RPLIB by expanding the definition of rankability from three measures to four. Mathematically, we test whether the following expanded definition of rankability is useful: $r = f(k, |P|, \tau, \delta)$, where $\delta$ is a measure of the location of indecision in multiple optimal rankings. In most ranking applications, one is most interested in the top-$k$ part of the ranking.

Thus, we use RPLIB to generate four artificial matrices from the Special class, $D_1$, $D_2$, $D_3$, and $D_4$, designed to have particular structure. Users have the ability in the RPLIB Colab Notebook to create matrices in the Special class that have a desired number of multiple optimal solutions. For example, when the user selects beginning and ending row indices of 6 and 10, respectively, this function creates a
Developing a Ranking Problem Library (RPLIB) from a dominance matrix that has $(10 - 6 + 1)! = 5! = 120$ multiple optimal rankings. In this way, we generated the four dominance matrices $D_1$, $D_2$, $D_3$, and $D_4$ and their corresponding $X^*$ matrices shown in Fig. 11.

Fig. 11 These four $D$ matrices (left) and their corresponding $X^*$ matrices (right) have the same $k$, $p = |P|$, and $r$ scores, but different $\beta$ scores.
Ranking researchers prefer indecision, if it occurs, to appear toward the bottom of the ranking. This distinction is not captured by the existing rankability measures \( k \), \( p \), and \( \tau \). Thus, we propose a fourth rankability measure \( \beta \), defined as follows, which uses the number and location of fractional entries of the \( X^* \) matrix from the linear ordering model to penalize fractional entries that are further away from the diagonal and lower in the ranking. Errors, ambiguity, or indecision in the bottom of rankings are less concerning than the same thing at the top of the ranking; the weighted definition of \( \beta \) takes this into account.

We need to define several matrices in order to build \( \beta \): (1) \( X^*_b \) is a binary matrix where \( X^*_b(i,j) = 1 \) if \( X^*(i,j) \) is fractional and 0, otherwise, (2) \( B_n \) is an \( n \times n \) banded matrix where \( B_n(i,j) = |j - i| \), and (3) \( W_n \) is an \( n \times n \) matrix of weights where all entries in row \( i \) of \( W_n \) are \( \frac{1}{i} \). Once these matrices are created, \( \beta \) is built by summing and then normalizing entries in the Hadamard (component-wise) product of these three matrices \( X^*_b, B_n, \) and \( W_n \). First, the sum is held in the scalar \( \hat{\beta} = \sum_{i=1}^{n} \sum_{j=1}^{n} X^*_b(i,j) \cdot B_n(i,j) \cdot W_n(i,j) \), where \( \cdot \) represents the Hadamard product. This is normalized; \( \beta = \frac{\hat{\beta}}{\beta_{\text{max}}} \), where \( \beta_{\text{max}} = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{|j-i|}{i} \). Finally, one more normalization produces \( \beta \): \( \beta = \frac{\beta}{\text{of fractional entries in} X^*} \).

This new rankability measure \( \beta \) works as intended. Remember that lower scores are better. See the \( \beta \) scores for \( D_1, D_2, D_3, \) and \( D_4 \) in Fig. 11. See Fig. 10 for the \( \beta \) scores for the 2002 and 2008 years of March Madness. Thus, \( \beta \) is capturing something that the other measures do not.

As shown in Fig. 11, the three prior metrics \( (k, p, \tau) \) do not capture the location of the multiple optimal solutions within the rankings (i.e., they do not capture position specific ambiguity in the rankings). Our proposed metric \( \beta \) does take the position specific information into account.

## 5 Open questions

Our hope is that RPLIB will facilitate the study of open questions in the field of ranking. A few open questions follow.

- **Instance Difficulty.** What makes an instance of the ranking problem difficult, i.e., which instances require a long time to find one (or all) optimal ranking(s)? Are there properties of the input matrix that indicate the difficulty of a ranking instance?
- **Inverse Problem.** Another open ranking problem is the inverse problem. Suppose you are given an optimal objective value and set of optimal rankings. Does there exist an input matrix associated with this output data? Under what conditions is the input matrix unique?
- **Rankability Score.** Anderson et al. (Anderson et al. 2021b, 2019a, 2021a) define the related ranking problem, the rankability problem. We hope RPLIB will advance open questions in that new field. For example, Anderson et al. defined
rankability as a function of several measures such as the four measures described in Section 4. How to combine these measures to create a scalar rankability score is an open question.

- **Improving Rankability.** What data can be collected to improve the rankability of a given dataset?

### 6 Future work

In version 2.0 of RPLIB, we plan an interactive tool that enables users to analyze their own data using algorithms from the library. We also plan an interactive tool for one particular dataset, the US News and Report data on rankings of colleges from their features. We envision a tool that parents, students, and guidance counselors can use to personalize rankings of colleges for an individual student based on their preference and weights of features.

### 7 Conclusion

RPLIB is a library for researchers of the ranking (or linear ordering) problem. The library is a searchable database of hundreds of instances of ranking problems from both real and artificial data with algorithms, code, and an interactive notebook. We invite the ranking community to use the library, give feedback for future versions, and contribute work.

**Acknowledgements** We thank California Polytechnic State University students Marisa Aquilina, Ethan Goldfarb, and Jackson Waschura for their contributions to RPLIB.

**References**

Anderson P, Chartier T, Langville A (2019a) The rankability of data. SIAM J Math Data Sci 1:121–143
Anderson P, Chartie T, Langville A, Pedings-Behling K (2019b) IGARDS technical report https://igards.github.io/research/IGARDS_Technical_Report_November_2019.pdf
Anderson P, Chartier T, Langville A, Pedings-Behling K (2020) Revisiting rankability of unweighted data technical report. https://igards.github.io/research/Revisiting_Rankability_Unweighted_Data.pdf
Anderson P, Chartier T, Langville A, Pedings-Behling K (2021a) The rankability of weighted data from pairwise comparisons. Foundations of data science (2639-8001-2019-0-29), pp 1–26
Anderson PE, Chartier TP, Langville AN, Pedings-Behling KE (2021b) Fairness and the set of optimal rankings for the linear ordering problem. Optim Eng 23:1289–1317
Cameron TR, Langville AN, Smith HC (2020) On the graph Laplacian and the rankability of data. Linear Algebra Appl 588:81–100
Colley WN (2002) Colley’s bias free college football ranking method: the colley matrix explained. Princeton University, Princeton
Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) ImageNet: a large-scale hierarchical image database. In: CVPR09
Grötschel M, Jünger M, Reinelt G (1984) A cutting plane algorithm for the linear ordering problem. Oper Res 32(6):1195–1220
Hu R, Chau SL, Huertas J, Sejdinovic D (2022) Explaining preferences with shapley values. arXiv
https://doi.org/10.48550/arXiv.2205.13662
Kondo Y (2014) Triangulation of input-output tables based on mixed integer programs for inter-temporal
and inter-regional comparison of production structures. J Econ Struct 3(1):1–19
Langville AN, Pedings K, Yamamoto Y (2011) A minimum violations ranking method. Optim Eng
13(2):349–370
Martí R (2009) Linear Ordering Problem Instances. http://grafo.etsii.urjc.es/optsicom/lolib/, online;
accessed 10 June 2021
Martí R, Reinelt G (2011) The linear ordering problem: exact and heuristic methods in combinatorial
optimization. AMS
Massey K (1997) Statistical models applied to the rating of sports teams. Bachelor's thesis, Bluefield
College
McJames N, Malone D, Mason O (2022) A supervised learning approach to rankability. Tech Rep arXiv:
2203.07364, Hamilton Institute, Maynooth University
Mitchell M, Wu S, Zaldívar A, Barnes P, Vasserman L, Hutchinson B, Spitzer E, Raji ID, Gebru T (2019)
Model cards for model reporting. In: Proceedings of the conference on fairness, accountability, and
transparency, pp 220–229
Reinelt G (2017) Tsplib https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps
and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the
author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article
is solely governed by the terms of such publishing agreement and applicable law.